

Министерство сельского хозяйства Российской Федерации

Российский государственный аграрный университет –  
МСХА имени К.А. Тимирязева

# **ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ В АПК**

Коллективная монография

Москва  
РГАУ – МСХА имени К.А. Тимирязева  
2025

УДК 338.43.02:681.518:63:004

ББК 40.72:65.32:16.6

Ц 74

**Рецензенты:**

Профессор кафедры прикладной информатики института экономики и управления АПК  
ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева, кандидат технических наук, профессор  
**Потапов Б.В.**

Доцент кафедры информатики и вычислительной техники института автоматизации и  
информационных технологий «Самарского государственного технического университета,  
кандидат экономических наук, доцент **Малина А.Б.**

**Информационные системы и технологии в АПК:** коллективная монография /  
Л.И. Хоружий, А.В. Уколова, М.К. Джикия, Б.Ш. Дашиева, С.О. Семенова, А.Ю.  
Ц 74 Ветошкин, А.Д. Титов, Д.А. Титова, Д.В. Быков, А.Н. Сивова [Электронный ресурс]

ISBN 978-5-9675-2123-2

В монографии представлены исследования преподавателей, аспирантов и магистров  
кафедры статистики и кибернетики ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева в области  
цифровизации сельского хозяйства.

Монография может представлять интерес для преподавателей, научных работников и  
сотрудников органов управления агропромышленным комплексом, специалистов сельского  
хозяйства, аспирантов и студентов, обучающихся по направлению «Информационные системы и  
технологии».

© Хоружий Л.И., 2025 © Уколова А.В., 2025  
© Джикия М.К., 2025 © Дашиева Б.Ш., 2025  
© Семенова С.О., 2025 © Ветошкин А.Ю., 2025  
© Титов А.Д., 2025 © Титова Д.А., 2025  
© Быков Д.В., 2025 © Сивова А.Н., 2025  
© ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева,  
2025

## СОДЕРЖАНИЕ

<b>ПРЕДИСЛОВИЕ .....</b>	<b>10</b>
<b>Глава 1. Цифровая платформа для популяризации и развития рынка органической сельскохозяйственной продукции .....</b>	<b>13</b>
1.1 Теоретические основы органического сельского хозяйства и методология проектирования информационных систем.....	13
1.2. Современное состояние и перспективы развития рынка органической продукции в России и мире.....	23
1.3. Проектирование и реализация цифровой платформы для органической продукции .....	48
Библиографический список к главе 1 .....	75
<b>Глава 2. Информационная система мониторинга показателей устойчивого развития сельского хозяйства .....</b>	<b>85</b>
2.1 Теоретико-методологические основы мониторинга целей устойчивого развития в аграрном секторе.....	85
2.2. Расширение системы индикаторов ЦУР №2 «Ликвидация голода»: уровень самообеспечения и землепользование .....	95
2.3. Проектирование и разработка информационной системы на основе парсинга и визуализации данных.....	100
Библиографический список к главе 2 .....	126
<b>Глава 3. Программный сервис для определения заболеваний растений с применением технологий Big Data и компьютерного зрения .....</b>	<b>138</b>
3.1. Обзор технологий: Big Data, компьютерное зрение и веб-разработка в агротехнологиях .....	138
3.2. Процессы диагностики заболеваний сельскохозяйственных культур: от традиционных к автоматизированным .....	144
3.3. Архитектура и реализация сервиса на основе сверточных нейронных сетей (на примере яблони).....	159
Библиографический список к главе 3 .....	176
<b>ПРИЛОЖЕНИЯ .....</b>	<b>178</b>

## ПРЕДИСЛОВИЕ

Монография представляет собой комплексное исследование цифровых решений в сфере устойчивого развития сельского хозяйства. В ней рассматриваются ключевые направления современной агротехнологической трансформации: информационная поддержка органического производства, мониторинг целей устойчивого развития и автоматизированная диагностика фитосанитарного состояния сельскохозяйственных культур. Авторы надеются, что данное издание будет полезно как специалистам агропромышленного комплекса, так и разработчикам информационных систем, а также исследователям, занимающимся вопросами устойчивого сельского хозяйства, анализа больших данных и применения искусственного интеллекта в аграрной сфере.

Цифровизация сельского хозяйства сегодня выходит далеко за рамки автоматизации отдельных процессов. Она становится фундаментальной основой для реализации принципов устойчивого развития, обеспечения продовольственной безопасности, повышения доверия потребителей к продукции и эффективного реагирования на риски, связанные с заболеваниями растений. Внедрение современных IT-решений от платформ агрегации данных и систем мониторинга показателей до моделей компьютерного зрения, позволяет не только оптимизировать производственные цепочки, но и формировать новые модели взаимодействия между производителями, потребителями, наукой и государством.

Указ Президента Российской Федерации № 309 от 7 мая 2024 года, регулирующий направления развития экономики, подчёркивает особую роль цифровой трансформации в обеспечении продовольственной независимости и экологической устойчивости аграрного сектора. В этих условиях задача создания интеллектуальных, прозрачных и адаптивных информационных систем приобретает стратегическое значение.

Монография подготовлена коллективом кафедры статистики и кибернетики ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева на основе исследований

магистрантов, обучающихся по направлению подготовки «Информационные системы и технологии».

Глава 1 «Цифровые платформы для популяризации и развития рынка органической сельскохозяйственной продукции» подготовлена кандидатом экономических наук, доцентом кафедры статистики и кибернетики Дашиевой Б.Ш., ассистентом кафедры статистики и кибернетики Ветошкиным А.Ю., ассистентом кафедры статистики и кибернетики Семеновой С.О. В работе предложена архитектура интернет-платформы, объединяющей функции информационного агрегатора, маркетплейса и аналитической системы, направленной на повышение осведомлённости потребителей и поддержку отечественных производителей.

Глава 2 «Информационная система мониторинга показателей устойчивого развития сельского хозяйства» разработана доктором экономических наук, профессором, директором Института экономики и управления АПК Хоружий Л.И., кандидатом экономических наук, доцентом, и.о. кафедры статистики и кибернетики Уколовой А.В., ассистентом кафедры статистики и кибернетики Титовым А.Д., ассистентом кафедры статистики и кибернетики Титовой Д.А. Исследование включает расширение системы индикаторов ЦУР №2 «Ликвидация голода» за счёт показателей самообеспечения и рационального землепользования, а также создание автоматизированной системы сбора, обработки и визуализации данных на основе парсинга открытых источников.

Глава 3 «Программный сервис для определения заболеваний растений с применением технологий Big Data и компьютерного зрения» подготовлена кандидатом экономических наук, доцентом кафедры статистики и кибернетики Джикией М.К., старшим преподавателем кафедры статистики и кибернетики Быковым Д.В., выпускницей магистратуры Сивовой А.Н. В работе представлена компонентная архитектура системы, основанной на сверточных нейронных сетях, обеспечивающей раннюю диагностику фитопатологий и интеграцию с системами мониторинга агроэкосистем.

Монография может представлять интерес для преподавателей, научных работников, сотрудников органов управления агропромышленным комплексом, практикующих специалистов в области сельского хозяйства, информационных технологий и устойчивого развития, а также для аспирантов и студентов профильных направлений подготовки.

## **Глава 1. Цифровая платформа для популяризации и развития рынка органической сельскохозяйственной продукции**

### **1.1 Теоретические основы органического сельского хозяйства и методология проектирования информационных систем**

Современный агропромышленный комплекс стоит перед вызовом обеспечения растущего населения планеты продовольствием без ущерба для экосистем. В этом контексте органическое сельское хозяйство представляет собой альтернативную модель использования имеющихся сельскохозяйственных ресурсов, которая будет направлена на гармонизацию отношений между человеком и природой. Такие методы, в отличие от распространенных, использующих синтетические удобрения и пестициды, базируются на принципах замкнутого цикла производства, где отходы одного процесса становятся ресурсами для другого.

Органическое сельское хозяйство использует методы, восходящие еще к традиционным земледельческим агропрактикам, однако сегодня они применяются во взаимодействии с научными достижениями, такими как, применение микробных препаратов для защиты растений или точное земледелие с использованием IoT-технологий.

Согласно определению Международной федерации движения экологического развития, органическое сельское хозяйство – целостная система производства, которая поддерживает здоровье экосистем, почв и людей, опираясь на экологические процессы, биологическое разнообразие и природные циклы, а не на использование вредных ресурсов. [105] Органическое сельское хозяйство сочетает традиционные методы, инновационные технологии и научные достижения для повышения устойчивости окружающей среды и обеспечения справедливых отношений во всей производственно-сбытовой цепочке. В основе органического сельского хозяйства по IFOAM лежат 4 принципа: принцип здоровья, принцип экологии, принцип справедливости и принцип заботы. Каждый из них говорит не только о производственных стандартах, но также делает акцент

на устойчивости, этике и социальной ответственности, что немаловажно учитывать, говоря об органической продукции, поскольку при популяризации органического производства нужно в равной степени продвигать ценности экологичного потребления.

Имея представление об органическом сельском хозяйстве с точки зрения философских оснований, далее разберемся, в сущности, исследуемого термина, а также в особенностях его производства. Согласно федеральному закону РФ № 280-ФЗ «Об органической продукции», органическая продукция – экологически чистая сельскохозяйственная продукция, сырьё и продовольствие, произведённые в соответствии с требованиями настоящего закона, исключая применение синтетических пестицидов, химических удобрений, ГМО, гормонов роста и антибиотиков (за исключением разрешённых перечнем), с обязательной сертификацией на всех этапах производства [77] Продукт считается органическим только при наличии сертификата, подтверждающего соответствие стандартам на всех этапах – от производства до упаковки.

Однако, органическое сельское хозяйство регулируется не только национальными, но и международными стандартами, которые также регулируют требования к производству экологически чистой продукции и сопутствующим этому процессам. К наиболее значимым относятся стандарты Европейского союза, США и Codex Alimentarius. Европейские нормы определяют органическое производство, как устойчивая система земледелия, сочетающая лучшие экологические практики, сохранение биоразнообразия и природных ресурсов, а также высокие стандарты благополучия животных [99]. Европейские стандарты имеют более жесткие нормы по содержанию сторонних примесей (не более 0.9% для ГМО), регламентируют обязательное создание буферных зон между органическими и традиционными полями (для предотвращения загрязнения) и детализированные правила импорта органической продукции. Американский стандарт USDA Organic устанавливает, что органическая продукция должна производиться без использования синтетических удобрений, осадков сточных вод (sewer sludge), ионизирующего излучения и ГМО. Ключевые отличия от



российских норм включают разрешение применения некоторых синтетических веществ из Национального списка, а также отсутствие обязательного требования к севообороту, которое присутствует в российском законодательстве (статья 8 ФЗ № 280-ФЗ). Международный стандарт Codex Alimentarius определяет органическое сельское хозяйство как систему, основанную на минимизации использования внешних ресурсов и отказе от синтетических материалов. Этот стандарт служит ориентиром для многих национальных систем сертификации.

Обратимся также к Межгосударственному стандарту по продукции органического производства ГОСТ-33980-2016, в котором указаны такие пункты, как правила производства, переработки, маркировки и реализации, изучаемой продукции. Стоит отметить, что данный стандарт был разработан Национальным фондом защиты потребителей и Комитетом Государственной Думы по аграрным вопросам. Данный контроль качества был принят Межгосударственным советом по стандартизации, метрологии и сертификации и за принятие проголосовали следующие страны: Беларусь, Казахстан, Киргизия, Россия и Таджикистан. Немаловажно указать, что данный стандарт был разработан в соответствии с рекомендациями Codex Alimentarius «Руководство по изготовлению, переработке, маркировке и реализации органических продуктов питания». Разработка данного стандарта направлена на решение ключевых задач современного органического рынка. Прежде всего, он призван обеспечить надежную защиту потребителей от недобросовестных практик, исключая возможность введения в заблуждение при приобретении продукции. Одновременно стандарт создает равные условия для производителей, добросовестно соблюдающих установленные требования, защищая их от нечестной конкуренции со стороны поставщиков, предлагающих продукцию, не соответствующую критериям органического производства, помимо этого целью является поддержка развития органических продовольственных программ по производству и обороту органической продукции, преследующих сохранение окружающей среды. Особенности производства органической продукции являются:

- В органическом производстве запрещается применять генно-модифицированные организмы и продукцию, в виде пищевых продуктов, кормов, пищевых добавок, средств защиты растений, семян и посадочного материала;

- Производство, на котором осуществляют производство органической продукции, должно быть расположено вдали от источников загрязнения окружающей среды, также объектов промышленной деятельности и территорий ведения сельского хозяйства. Производственный участок, предназначенный для ведения органического хозяйства, должен быть отделен от других производственных объектов, которые не соответствуют требованиям данного стандарта;

- Необходимо тщательно очистить оборудование перед производством органической продукции, если оно было задействовано ранее при ведении производства, не относящегося к органическому;

- Необходимо осуществлять документирование производства органической продукции, включающее такую информацию, как ежегодный план выращивания растений и севооборота, информацию об используемых сортах растений, разрешенных удобрениях и средствах защиты растений, в отношении продукции растениеводства. Стоит указать, что в отношении оборота органической продукции необходимо предоставлять отчет о происхождении, типах, составе использовании приобретенной и реализованной продукции;

- В органическом растениеводстве используют методы обработки почвы, направленные на сохранение ее естественного сложения, предотвращения развития деградационных процессов и поддержания разнообразия экосистем;

В первую очередь стоит рассмотреть понятие информационной системы, а также познакомиться с ее функциями. Информационная система — совокупность содержащейся в базах данных информации и обеспечивающих её обработку информационных технологий и технических средств. [46]

Основными элементами этого определения являются:

- базы данных, как организованный набор информации;

- информационные технологии – программное обеспечение и алгоритмы обработки данных;

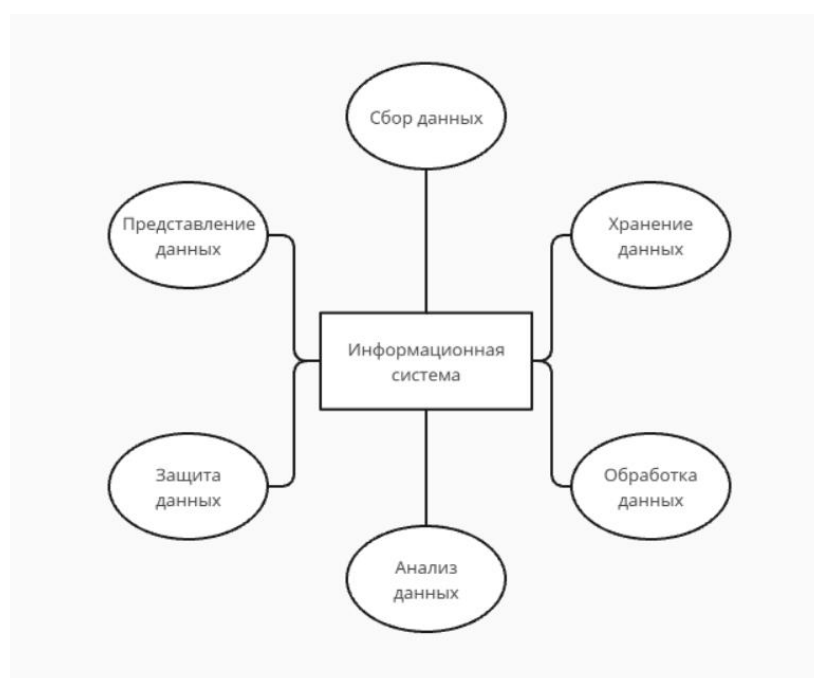
- технические средства – устройства, обеспечивающие работу системы.

Информационные технологии развиваются особенно быстро сегодня. В связи с этим возникает потребность в усовершенствовании процедуры управления и повышении качества администрирования в различных учреждениях и организациях, в особенности информационными ресурсами. Информационная система, в данном контексте, представляет собой комплекс взаимосвязанных компонентов, которые совместно обеспечивают сбор, хранение, обработку, анализ и распространение информации для поддержки различных видов деятельности.

Во многом это достигается благодаря информационным системам поддержки. Анализ собранной и постоянно обновляемой информации может вызвать проблемы, связанные с необходимостью качественной обработки большого количества данных и их мониторингом. Благодаря этому типу систем поддержки аналитическая деятельность и различные инструменты управления наиболее активно используются для принятия более объективных управленческих решений.

Одной из ключевых функций информационной системы является извлечение данных из различных источников и преобразование в единый структурированный формат, для дальнейшего использования, обработки, анализа. Информационная система накапливает большие объемы данных, следовательно, не менее важной функцией является организация хранения данных и обеспечение эффективного доступа к информации, т.е. использование инструментов для быстрого поиска и извлечения необходимых сведений. Особое значение имеет функция обработки и анализа информации, реализующаяся с помощью встроенных аналитических инструментов, направленная на выявление связей, зависимостей, тенденций и т.д. Сюда же можно отнести и интеллектуальные механизмы для поддержки принятия решений. Автоматизация бизнес-процессов, как функция ИС, столь же важна, поскольку система может взять на себя выполнение рутинных операций, как, например, обработка транзакций, формирование отчетов, отправка уведомлений,

что значительно повышает эффективность работы и снижает вероятность ошибок. Интеграция с внешними системами и сервисами существенно расширяет функциональные возможности системы. Информационная система также выполняет коммуникационную функцию, обеспечивая взаимодействие между пользователями. Она предоставляет инструменты для обмена информацией, совместной работы и распространения знаний. Обеспечение информационной безопасности – неотъемлемая часть функционирования ИС. Система реализует механизмы защиты данных, направленные на обеспечение конфиденциальности, целостности и доступности информации (Рисунок 1.1).



**Рисунок 1.1 – Основные функции ИС**

Таким образом, современная информационная система представляет собой многофункциональный инструмент, который не только обеспечивает хранение и обработку данных, но и поддерживает аналитическую деятельность, автоматизирует бизнес-процессы, способствует взаимодействию пользователей и гарантирует безопасность информации. Эти функции делают информационные системы незаменимыми в различных сферах деятельности.

В состав компонентов информационной системы можно включить три основные группы.

Средства информационно аналитического обеспечения, служащие для интеллектуальной и логической поддержки работы системы, в состав которых входит:

1. Интеллектуальная компонента: алгоритмы анализа данных, системы поддержки принятия решений, обработка естественного языка.
2. Логическая компонента: бизнес-правила, логика взаимодействия модулей, валидация данных.
3. Технические средства: оборудование и сети.

Эти компоненты образуют ядро информационно-аналитического обеспечения, которое позволяет пользователям достигать поставленных целей и решать задачи управления в соответствии с намеченными направлениями своей основной деятельности (Рисунок 1.2).



**Рисунок 1.2 – Компоненты средств ИС**

Основные компоненты информационной системы, обеспечивающие ее функционирование:

1. Аппаратное обеспечение, которое включает серверную инфраструктуру, клиентские устройства и сетевое оборудование, обеспечивающее коммуникацию между компонентами.

2. Программное обеспечение, состоящее из системного ПО, прикладного ПО, а также специализированных систем, как например СУБД или системы аналитики.

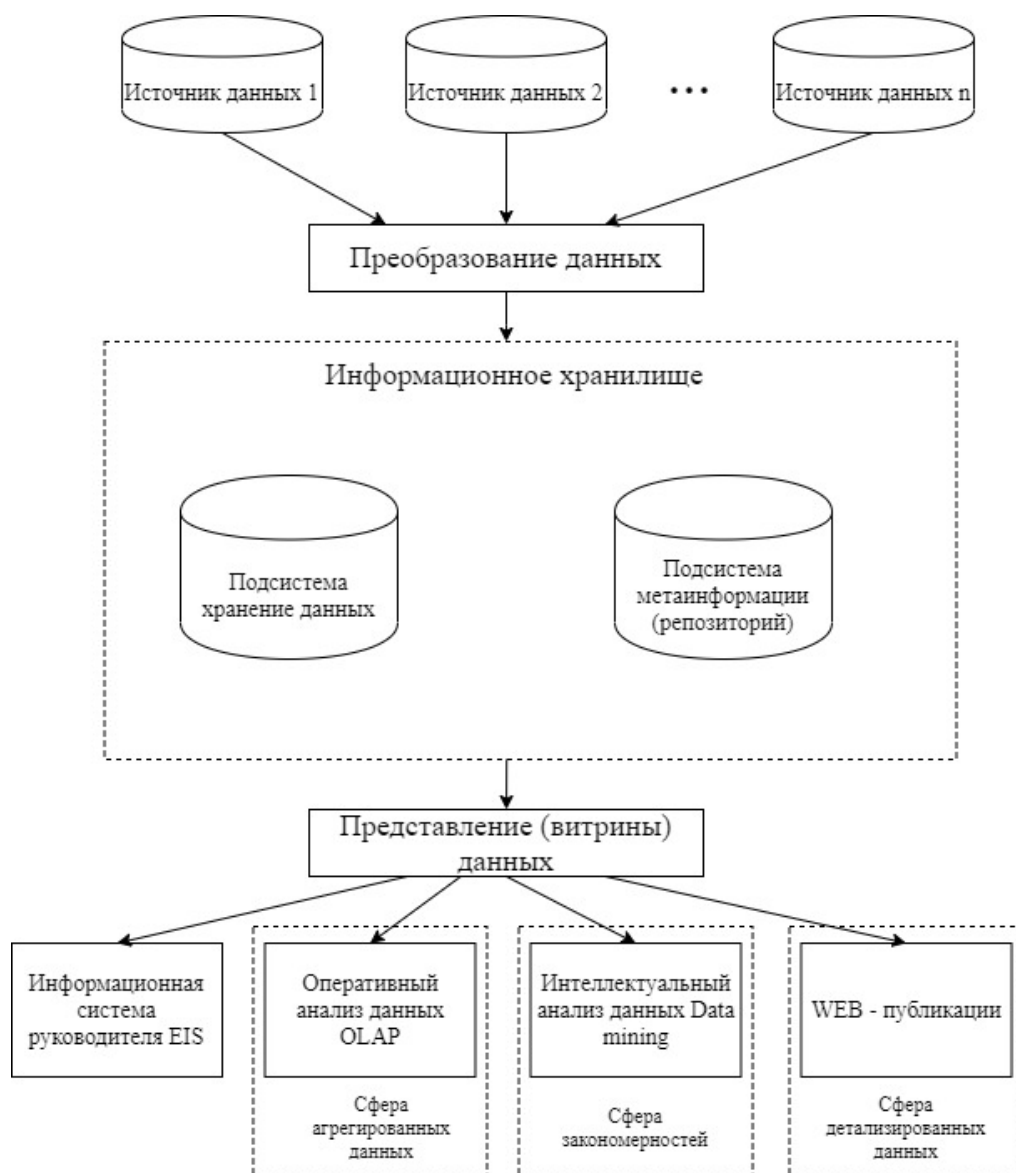
3. Информационные ресурсы – данные структурированные (базы данных, каталоги) или неструктурированные (документы, изображения), а также метаданные, содержащие описания структуры данных и их взаимосвязи.

4. Организационные компоненты, содержащие регламенты, инструкции и процедуры, методики и стандарты работы.

Заключительная группа является абстрактной, поскольку, отвечает лишь за взаимодействие компонентов. Все указанные выше компоненты должны образовывать единую экосистему, в которой технические средства являются физической основой для работы информационной системы. Программные решения, т.е. программное обеспечение реализует бизнес-логику и интерфейсы. Информационные ресурсы содержат актуальные данные для обработки и снабжают ими информационную систему. Организационные компоненты регламентируют процессы работы.

Рассмотрим базовую схему информационно-аналитической системы, построенной на основе хранилища данных (Рисунок 1.3). В определенных реализациях отдельные компоненты этой схемы часто отсутствуют, однако в большинстве случаев она универсальна и подходит для большинства проектов, связанных с обработкой и анализом информации.

На данной схеме представлена концептуальная модель информационно-аналитической системы интернет-платформы в классическом виде, однако ее можно интерпретировать под модель, ориентированную на популяризацию потребления и развитие производства органической сельскохозяйственной продукции, добавив к ней описание и тем самым адаптировать ее компоненты под специфику нужной платформы.



**Рисунок 1.3 – Структура информационной системы (ИС)**

Чтобы сделать ее более релевантной теме, нужно конкретизировать ключевые элементы:

1. Источники данных:

- Данные о производителях органической продукции (адреса, сертификаты, ассортимент).
- Данные о продуктах (названия, категории, описание, сроки хранения, сертификация).
- Данные о пользователях (покупателях и фермерах): профиль, история покупок, предпочтения.
- Отзывы и рейтинги от пользователей.

- Внешние данные: погода, урожайность регионов, рынки сбыта, государственные реестры органических хозяйств и т.д.
2. Преобразование данных (этап очистки, нормализации, объединения):
- Нормализация названий продуктов и категорий.
  - Объединение данных из разных источников (например, от фермеров и магазинов).
  - Геолокационная обработка (привязка к регионам, маршрутам доставки).
  - Обогащение данными (добавление информации о сезонности, сертификации).
3. Информационное хранилище (центральное хранилище структурированных данных и метаданных, обеспечивающее единое пространство для анализа).
4. Представление данных (реализуется через разные форматы):
- EIS для руководства - аналитика по объемам производства, динамика потребления, география продаж.
  - OLAP для детального анализа - анализ спроса, сезонных колебаний, сравнение между регионами.
  - Data Mining для прогнозирования - прогнозирование спроса, выявление предпочтений пользователей, рекомендательные системы.
  - WEB-интерфейсы для взаимодействия с пользователями - интерактивные карты с фермерами, каталоги продуктов, рейтинг производителей.

Данная система позволяет эффективно управлять данными, поддерживать принятие решений и способствовать развитию устойчивого органического сельского хозяйства.



## **1.2. Современное состояние и перспективы развития рынка органической продукции в России и мире**

Мировой опыт показывает, что проблемы обеспечения устойчивого развития сельского хозяйства всегда были и остаются до сих пор самыми важными в экономике каждой страны. В связи с этим многие страны стали существенно пересматривать концепции традиционной технологии возделывания сельскохозяйственных земель и выращивания сельскохозяйственной продукции не только с позиции природопользования, но и улучшения экономической ситуации в этой отрасли. В современном обществе происходит реальное осознание протекающих в мире экологических проблем и потрясений. В условиях глобальных экологических вызовов растет осознание необходимости перехода к более устойчивым моделям. Существует множество факторов, которые негативно влияют как на сохранение окружающей среды, так и на качество и безопасность продуктов питания, и как следствие на состояние здоровья нынешнего и будущего поколений. Популярность продуктов питания органического сельского хозяйства растет с каждым годом во всем мире. Органическое питание становится одним из самых динамично развивающихся сегментов продовольственного рынка.

По данным Organic Trade Association (ОТА), мировой рынок органических продуктов в 2025 году демонстрирует рекордные темпы роста, при этом в США органический сектор впервые с 2020 года опережает общий рост продовольственного рынка. Европейский Союз также продолжает укреплять позиции в сфере органического сельского хозяйства: по данным Eurostat, в 2025 году объём органической продукции увеличился на 8,4% по сравнению с предыдущим годом, при сохранении среднегодового роста выше 7% за последние пять лет. Прежде, чем рассмотреть, как обстоит ситуация с органической продукцией в России, интересно посмотреть, что происходит на этом же рынке в других странах по всему миру.

Согласно последнему изданию «The World of Organic Agriculture – Statistics and Emerging Trends 2025», подготовленному Research Institute of Organic

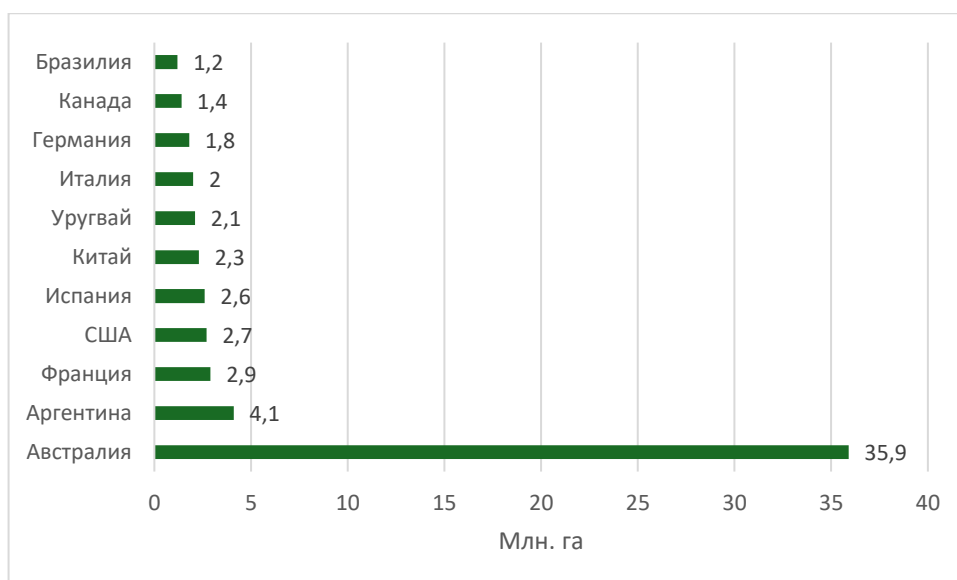
Agriculture (FiBL) и IFOAM – Organics International , к 2025 году статистика по органическому сельскому хозяйству доступна уже в 182 странах, что на 14 стран больше, чем в 2012 году.

По данным на 2023 год общая площадь органических сельскохозяйственных земель в мире составляет 78,7 млн га. Регионы с наибольшими площадями органического земледелия расположены в Океании (36,2 млн га, 46% мирового объёма) и Европе (19,1 млн га, 24%). В Латинской Америке находится 10,5 млн га (13%), за которой следуют Азия (8,1 млн га, 10%), Северная Америка (3,9 млн га, 5%) и Африка (2,4 млн га, 3%).

**Таблица 1.1 – Страны с наибольшими площадями органических с.-х. земель**

Страна/Регион	Площадь земель, используемых для органического сельского хозяйства (млн га)	Площадь земель, используемых для органического сельского хозяйства (% от мирового объёма)
Австралия	35,9	45,6
Аргентина	4,1	5,2
Франция	2,9	3,7
США	2,7	3,4
Испания	2,6	3,3
Китай	2,3	2,9
Уругвай	2,1	2,7
Италия	2,0	2,5
Германия	1,8	2,3
Канада	1,4	1,8
Бразилия	1,2	1,5
Великобритания	0,5	0,6
Россия	0,4	0,5
Украина	0,3	0,4
Казахстан	0,3	0,4
Беларусь	0,2	0,3
Остальные страны	16,9	21,5
ИТОГО:	78,7	100

По данным FiBL & IFOAM за 2023 год, 1,6% всех сельскохозяйственных земель в мире сертифицированы как органические.



**Рисунок 1.4 – Страны с наибольшей площадью органических с.-х. земель**

Как видно из представленных данных, мировой лидер по площади органических сельскохозяйственных земель - Австралия (35,9 млн га), что составляет 45,6% от общемирового объема. За ней следуют Аргентина (4,1 млн га, 5,2%) и Франция (2,9 млн га, 3,7%). Согласно анализу динамики, за последнее десятилетие площадь органических земель в мире увеличилась более чем на 30 млн га, при этом наиболее значительный рост наблюдается в Океании (преимущественно за счет Австралии) – прирост более 18 млн га, Европе на 7.6 млн га и Латинской Америке на 3,9 млн га. Как показывают расчеты 70% всех органических земель сосредоточены в 10 странах-лидерах. Австралия сохраняет абсолютное лидерство, обеспечивая почти половину мирового объема органических земель (45,6%), при этом основная часть приходится на обширные сертифицированные пастбища. Европейские страны также демонстрируют устойчивый рост. Франция, Италия, Испания входят в топ-8 стран по площади органических земель. Китай (2,3 млн га) и Бразилия (1,2 млн га) постепенно увеличивают свою долю на мировом органическом рынке.

Примечательно, что Европейские страны лидируют по доле органических земель в общей структуре сельхозугодий (до 25% в отдельных странах), тогда как глобальный показатель составляет около 1,6%. США, являясь крупнейшим потребителем органической продукции, по площади органических земель

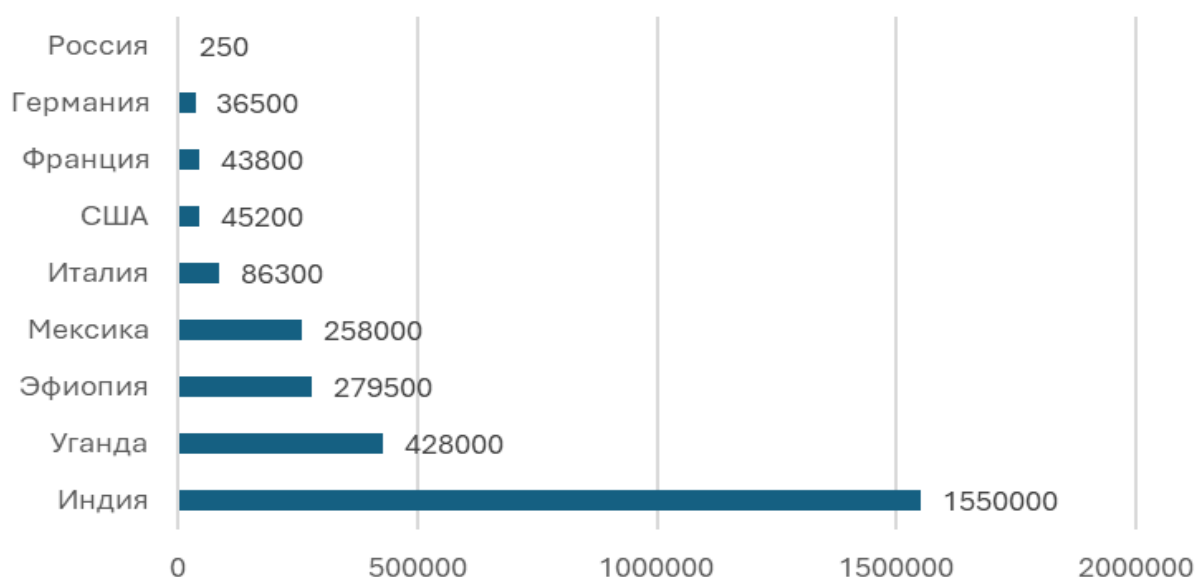
занимают 4-е место. Россия и страны СНГ пока имеют скромные показатели, суммарно около 1,2 млн га (1,5 % от мирового объема).

**Таблица 1.2 – Развитие органических с.-х. земель по странам в 2019-2023 гг., млн. га**

Страна/Регион	2019	2020	2021	2022	2023	Δ 2022-2023 (млн га)	Δ 2022- 2023 (%)
Австралия	35.6	35.7	35.7	35.7	35.9	+0.2	+0.6%
Аргентина	3.6	3.9	4.0	4.0	4.1	+0.1	+2.5%
Франция	2.3	2.5	2.7	2.8	2.9	+0.1	+3.6%
США	2.2	2.3	2.5	2.6	2.7	+0.1	+3.8%
Испания	2.2	2.4	2.5	2.5	2.6	+0.1	+4.0%
Китай	1.9	2.1	2.2	2.2	2.3	+0.1	+4.5%
Уругвай	1.8	1.9	2.0	2.0	2.1	+0.1	+5.0%
Италия	1.8	1.9	1.9	2.0	2.0	0.0	0.0%
Германия	1.5	1.6	1.7	1.8	1.8	0.0	0.0%
Канада	1.2	1.3	1.3	1.4	1.4	0.0	0.0%
Бразилия	1.0	1.1	1.1	1.2	1.2	0.0	0.0%
Великобритания	0.4	0.4	0.5	0.5	0.5	0.0	0.0%
Россия	0.3	0.3	0.4	0.4	0.4	0.0	0.0%
Украина	0.2	0.3	0.3	0.3	0.3	0.0	0.0%
Казахстан	0.2	0.2	0.3	0.3	0.3	0.0	0.0%
Беларусь	0.1	0.2	0.2	0.2	0.2	0.0	0.0%
Остальные страны	15.2	15.8	16.5	16.8	16.9	+0.1	+0.6%
Итого	71.5	73.9	75.3	76.4	78.7	+2.3	+3.0%

В 2023 году общая площадь органических сельскохозяйственных земель в мире достигла 78,7 млн га, что на 2,3 млн га (3,0%) больше, чем в 2022 году (76,4 млн га). Этот стабильный прирост характеризуется планомерным расширением площадей стран-лидеров. В качестве причин устойчивой динамики можно выделить постепенный переход хозяйств на органические методы, что в том числе требует периода конверсии, когда производство работает по всем требованиям органик-стандартов, но пока без сертификата. Сбалансированное развитие рынка потребления, стабильный спрос на органическую продукцию. Отсутствие резких изменений в сельскохозяйственной политике ведущих стран. Ключевые тенденции сводятся к годовому приросту на уровне 2-3 млн га, стабильному развитию в данной отрасли европейских стран и плавной положительной динамике лидирующих стран.

По данным на 2024 год (отчет FiBL & IFOAM основан на статистике за 2022–2023 гг.) существует более 3,7 миллионов производителей сертифицированной органической сельскохозяйственной продукции.



**Рисунок 1.5 – Страны с наибольшим количеством органических с.-х. производителей**

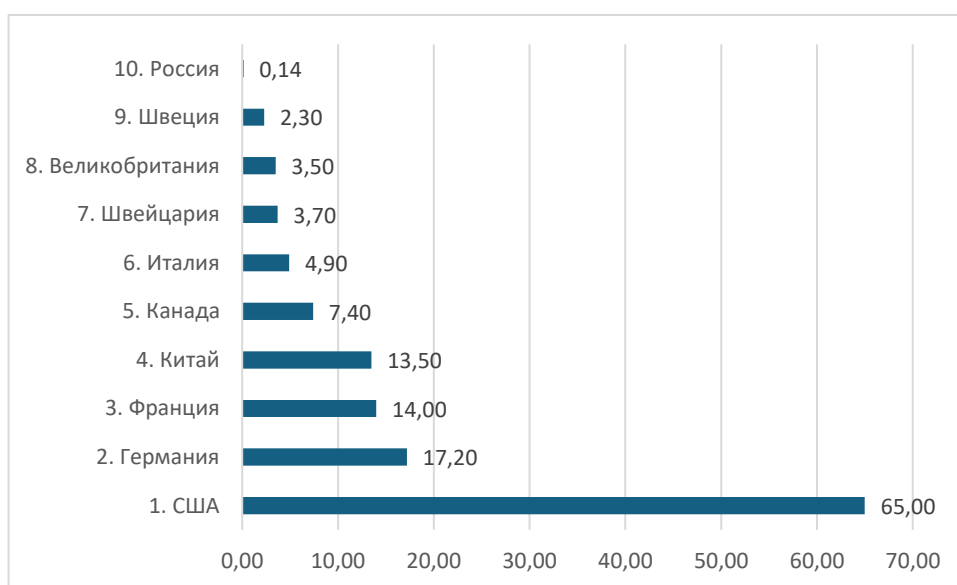
Лидерами являются развивающиеся страны (Индия, Уганда) за счет мелких фермеров, но с низкой долей сертифицированных земель, это достигается благодаря групповой сертификации (сертификат на кооператив из мелких фермеров) или объединению в экспортные кластеры мелких хозяйств, а также низкая стоимость сертификации. В добавок ко всему фермеры включаются в органическое земледелие часто «по традиции» ведения работ таким образом. Европа и США имеют меньше производителей, но более крупные технологичные предприятия, больший объем площадей, а также большую добавленную стоимость.

В России доля органических производителей крайне мала, поскольку каждое хозяйство проходит процесс сертификации отдельно, длительным является период конверсии, требования – высокие, необходимо соответствовать строгим нормам. В добавок в России довольно низкая осведомленность потребителей о преимуществах органических продуктов, а стоимость продукции высока, поэтому подобный переход часто может быть просто невыгодным.

Рынок органических продуктов – один из самых динамично развивающихся в мире. С 1999 по 2023 год он вырос более чем в 10 раз (с 13,1 до 147,8 миллиардов евро). По мнению экспертов рынок органической сельхоз продукции продолжит расти и к 2030 году может составить до 5% от мирового рынка.

Наблюдается и прогнозируемый устойчивый рост глобального рынка. По данным за 2022-2023 годы, объем розничных продаж органических продуктов в мире увеличился на 8 %, достигнув приблизительно 150 миллиардов долларов. Лидерами по объему розничной реализации остаются США с 65 млрд долларов (44,7 %), Германия – 17,2 млрд долларов (11,9%) и Франция – 14 млрд долларов (9,4%). При этом Китай демонстрирует значительный прирост (11,6%), что свидетельствует о растущем спросе на органическую продукцию. По количеству потребляемой продукции на душу населения европейские страны существенно опережают всех в мире.

Следует отметить, что 70% мирового объема органического производства по-прежнему сосредоточено в 10 странах-лидерах, при этом наблюдается постепенное увеличение доли развивающихся рынков, особенно в Азии и Латинской Америке.



**Рисунок 1.6 – Объем розничной реализации продукции органического сельского хозяйства в странах мира, млрд. долларов**

В целом анализ состояния мирового рынка продукции органического сельского хозяйства показал, что наибольшее развитие он получил в ведущих экономиках мира (в основном в странах ЕС и США), на которые в 2020 году

приходится 2/3 всех площадей мировых сельскохозяйственных угодий и 96% объема розничной реализации органической продукции. Развивающиеся страны сами органическую продукцию практически не потребляют и фактически выполняют функцию поставщиков на мировой рынок органической продукции.

В таблице 2.3 рассмотрены основные каналы, по которым реализуется органическая продукция в западных странах. Огромным преимуществом органического рынка является верность потребителей продукции, несмотря на любые кризисы. Вопреки всем проблемам питающийся правильно человек старается не возвращаться к обычным продуктам. По мнению экспертов, органический сектор удержал клиентов в 2013-15 г. и в 2022-23 г., потеряв часть выручки только из-за снижения среднего чека, несмотря на проблемы, как политического, правового, так и экономического характера.

**Таблица 1.3 – Каналы реализации органической продукции в западных странах**

Страна	Доля продаж по различным каналам реализации, %				
	Супермаркеты	Органические супермаркеты	Специализированные магазины	Прямые продажи производителей	Другие каналы (лавки, общественное питание, интернет-магазины)
Дания	90	-	4	6	-
Франция	45	26	12	12	5
Италия	86,5	-	13,5	-	-
Великобритания	72,3	-	15	2,8	9,9
США	54	39		7	

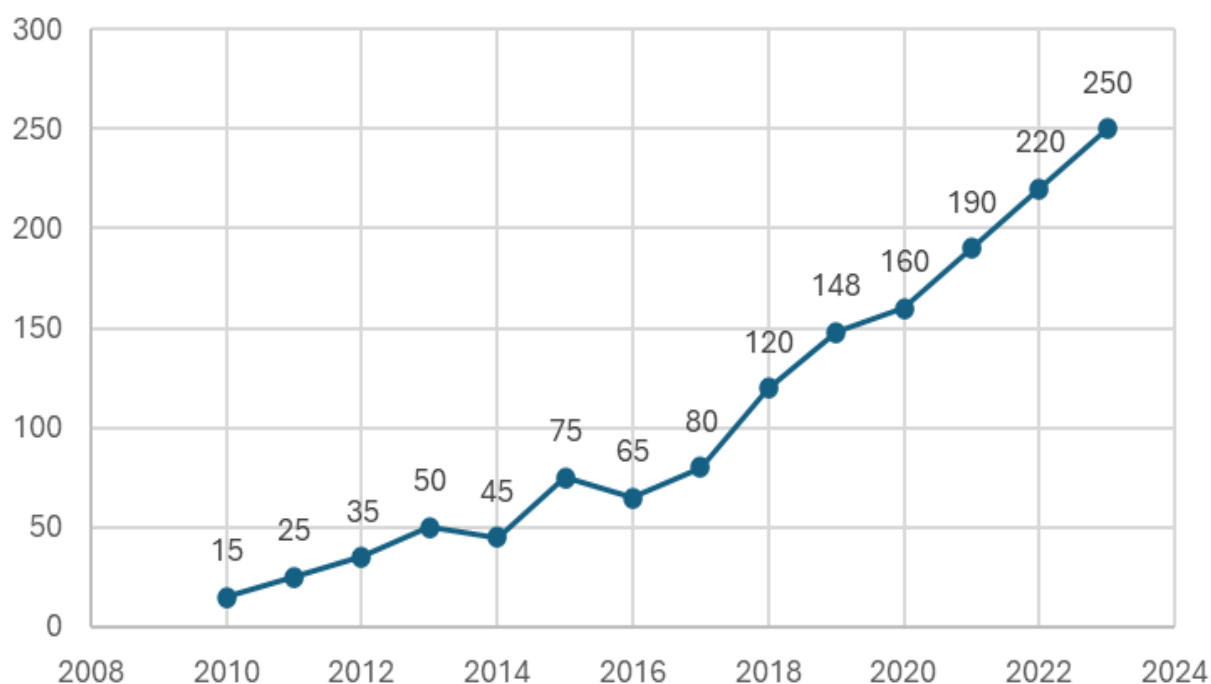
Наиболее значительным каналом реализации органической продукции в рассмотренных странах являются супермаркеты, доля которых составляет в общей структуре продаж от 45 до 90%. Во Франции и США также популярны органические супермаркеты и специализированные магазины – их доля там достигает 38-39%, а в других рассмотренных странах от 4 до 15%. Также во

Франции широко практикуются прямые продажи органической продукции в фермерских магазинах, на рынках и т.д. (12%). В Великобритании относительно высокий удельный вес занимают прочие каналы реализации (9,9%), среди которых наиболее популярна Интернет-торговля.

Положительным фактором развития российского рынка органической продукции является сложившаяся культура её потребления, что создаёт основу для роста внутреннего спроса. После вступления России в ВТО в 2012 году появились предпосылки для развития сильного производства, способного конкурировать с мировыми производителями. Принятие в 2020 году закона № 280-ФЗ "Об органической продукции" создало правовую основу для сертификации экологически чистых производств, что открыло перспективы превращения России в крупного производителя и экспортёра такой продукции. Однако, вследствие обстановки на мировой арене, ожидаемый рост производства для экспорта продукции столкнулся с рядом ограничений, которые включают в себя санкционное давление и, как следствие, изменение рынков сбыта.

По данным IFOAM и Национального органического союза (НОС) текущие показатели развития сектора остаются скромными, в 2023 году объем российского рынка органической продукции составил \$200–250 млн, что соответствует 0,3–0,5% общего рынка продовольствия. Для сравнения, в 2020 году этот показатель составлял лишь \$140–160 млн (0,2% рынка), причем основная часть продукции тогда была преимущественно импортирована из Германии. Поэтому, несмотря на скромную долю, темпы роста остаются высокими – 15–20% в год, что связано с импортозамещением и растущим спросом на локальную экопродукцию, а также, что говорит нам о значительном потенциале.





**Рисунок 1.7 – Развитие рынка продукции органического сельского хозяйства в России в 2010-2023 гг., млн. долларов**

Первоначальные темпы после зарождения рынка роста на 40-60% в 2011-2013 годах связаны с вступлением в ВТО и растущему интересу к «премиум-органик» сегменту. Спад объема рынка в денежном эквиваленте в 2014 году связан с девальвацией рубля. 2015-2016 годы характеризуются сначала контрсанкциями и «бумом» локальных производителей, переходу к импортозамещению, а затем экономическим кризисом и падению спроса на товары «премиум» сегмента. В 2017 стабилизируется курс рубля. 2018 год связан с законом №280 ФЗ «Об органической продукции и о внесении изменений в отдельные законодательные акты Российской Федерации». В дальнейшие годы на экономику огромное влияние оказала пандемия, во время и после которой возрос спрос на интернет-продажи и здоровый образ жизни. В 2022 году – переориентация экспорта в Азию, в 2023 – рекордное число сертифицированных производителей. Как итог – 2014-2016 – «роллеркостер», связанный с внешними шоками, резкие и непредсказуемые колебания. 2018-2023 – устойчивый рост, принятие ФЗ, положительные прогнозы на дальнейшее производство.

Структура рынка претерпела существенные изменения, связанные с геополитическими событиями, после 2022 года. Если раньше из ЕС

импортировалось до 70% рынка органической продукции, то сейчас наблюдается активное их замещение отечественными товарами. Количество сертифицированных производителей по сравнению с 82 в 2020 году выросло до более чем 150 в 2023 году по данным Россельхознадзора. По итогам 2023 года число выданных сертификатов выросло на 14% достигнув 385. Для анализа рынка часто важнее первый показатель – число производителей, равный по последним данным 177. Для оценки разнообразия продукции полезнее данные о количестве сертификатов. Основными направлениями экспорта стали Китай, ОАЭ, Казахстан и Беларусь, что говорит и об изменении географии экспорта.

Внутренний спрос на органическую продукцию остается ограниченным. Исследования АБ-Центра (2023) показывают, что органические продукты в России стоят в 2-5 раз дороже обычных. Основными потребителями являются жители Москвы, Санкт-Петербурга и других крупных городов с доходами выше среднего. В условиях экономической нестабильности часть покупателей перешла на более дешевые аналоги, однако премиальный сегмент сохраняет устойчивость. По данным «Комкон», 52% москвичей готовы переплачивать за безопасные продукты, что указывает на потенциал роста спроса. Однако в этом вопросе есть серьезные расхождения. Согласно опросу НОС (2023), только 35% россиян уверены в подлинности маркировки «органик», что свидетельствует о проблеме доверия. Многие из потребителей не видят разницы между продуктами с маркировкой «эко», «био» и «фермер», что говорит нам о низкой покупательской осведомленности. Покупатели имеют поверхностное или искаженное представление об органической продукции, путая ее с товарами без сертификации, что приводит к недоверию или необоснованным переплатам за несоответствующий товар. Низкая осведомленность о строгих стандартах производства и значении официальной маркировки, в том числе, сдерживает рост рынка. Ограниченность розничных каналов также негативно влияет на рост массового спроса. Однако положительное влияние на распространение органической продукции оказывают онлайн-продажи, которые делают продукты доступнее. Барьером для приобретения органической продукции является ее высокая цена, в России она

стоит в 2-5 раз дороже обычной, что сужает аудиторию. Если постоянные ее потребители готовы платить такую стоимость, то привлечь новых потребителей становится непросто.

**Таблица 1.4 - Сравнительная характеристика цен на основные продукты в эко-магазинах и в крупных торговых сетях в 2024г., руб.**

Магазин / Категория	Говядина (1 кг)	Свинина (1 кг)	Курица (1 кг)	Молоко (1 л)	Яйца (10 шт.)	Картофель (1 кг)	Лук (1 кг)	Морковь (1 кг)
Азбука Вкуса Organic	1 450	1 100	750	290	420	190	140	150
ВкусВилл Organic	1 300	950	700	270	390	170	130	140
Утконос Bio	1 400	1 000	720	280	400	180	135	145
Средняя цена в эко-магазинах	1 383	1 017	723	280	403	180	135	145
Пятёрочка	550	420	240	85	95	45	35	40
Магнит	580	450	260	90	100	50	40	45
Дикси	540	410	230	80	90	40	30	38
Лента	600	470	280	95	110	55	45	50
Средняя цена в сетевых магазинах	568	438	253	88	99	48	38	43
Превышение цены (раз)	2,4	2,3	2,9	3,2	4,1	3,8	3,6	3,4

На данный момент потребительский сегмент эко-продуктов – это люди с высоким уровнем достатка, для которых переплата за продукты в 2-8 раз не влияет кардинальным образом на семейный бюджет. Не случайно «Био-маркет» расположен на Рублевском шоссе, а «Глобус-Гурмэ» открыл свой рынок фермерской продукции в деревне Жуковка. Таким образом, насыщенность рынка эко-продуктов для состоятельных жителей Москвы и Подмосковья постоянно растет, что может свидетельствовать о наличии значительного спроса на эту продукцию. [59] Продавцам, ищущим возможность развиваться на российском

рынке, рано или поздно придется столкнуться с вопросом о расширении сегмента покупателей, и искать пути решения проблемы выхода на рынок для потребителей со средним достатком.

Экспортный потенциал российского органического сельского хозяйства оценивается в \$50-70 млн по итогам 2023 года, экспорт вырос на 15-17%. Ключевыми экспортными товарами стали зерновые (лён, гречиха), дикоросы (грибы, ягоды, мёд), а также рыба и морепродукты. После ухода западных конкурентов российские производители начали активно осваивать рынки Китая, ОАЭ, Беларуси и Казахстана. Однако развитию экспорта мешают такие факторы, как отсутствие взаимного признания сертификатов с ЕС и США, логистические сложности и конкуренция с производителями из Восточной Европы. В глобальном контексте доля России на мировом рынке органики остаётся незначительной - менее 0,1% при общем объёме рынка \$250 млрд в 2023 году по данным FiBL. Для сравнения, рынок ЕС оценивается в \$60 млрд, США - \$70 млрд, Китая - \$15 млрд [94]. При этом Россия обладает уникальными конкурентными преимуществами в сегменте дикоросов (лесные ягоды, грибы), где спрос продолжает расти.

Перспективы развития сектора до 2030 года зависят от мер государственной поддержки. По базовому сценарию рынок может вырасти до \$500 млн (0,8% продовольственного рынка), а при благоприятных условиях - до \$1 млрд. Для реализации этого потенциала необходимо упрощение сертификации для малых фермеров, стимулирование спроса через программы информирования и просвещения и развитие кооперативов для снижения цен.

Российский рынок органической продукции – это перспективное направление. Оно демонстрирует устойчивый рост, однако его масштабы пока не сопоставимы с мировыми лидерами. При совокупном влиянии положительных факторов отрасль производства органической сельхоз продукции должна начать привлекать возможных производителей. Ключевыми драйверами должны стать импортозамещение, экспорт в Азию и господдержка. Однако без решения проблем доверия потребителей и ценовой доступности органическое сельское хозяйство рискует остаться нишевым сегментом для ограниченной аудитории.

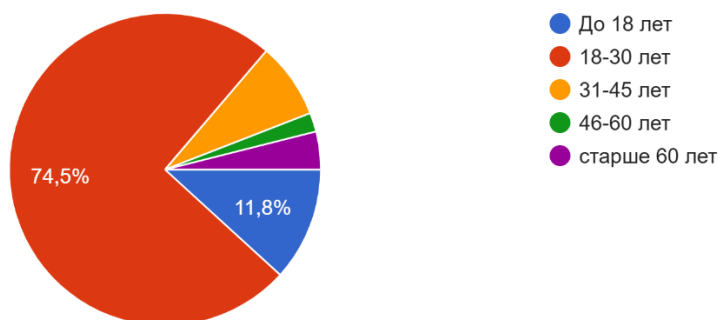
По словам специалистов, Россия отстает от всего мира в сфере экопродукции и экоуслуг на 15-20 лет. Россия имеет высокий потенциал помимо прочего по рыбе и морепродуктам, меду, лесным грибам, ягодам и орехам. Экопродукция может со временем стать одним из растущих секторов национального рынка. Главное, чтобы этому сегменту, производителям и поставщикам, работающим в нем, было уделено достаточное внимание как со стороны государства, так и бизнес-сообщества. [49]

Для выявления потребности российских потребителей в создании информационной интернет-платформы был проведен социальный опрос на тему органической продукции. Результаты анализа анкетирования послужили обоснованием для реализации проекта.

В результате опроса был оценен уровень осведомленности и доверия потребителей к органической продукции, были выявлены мотивы и барьеры для ее покупки, а также определена потенциальная потребность и желаемые функции единой информационной платформы для органической продукции.

В данной главе предлагается ознакомиться с результатами проведенного опроса. В опросе принимали участие мужчины и женщины разных возрастных категорий. 74.5% опрошенных, т.е. большая часть, находится в возрасте от 18 до 30 лет, 11.8% - до 18 лет, оставшиеся 13.7% находятся в иных возрастных диапазонах.

Ваш возраст  
51 ответ

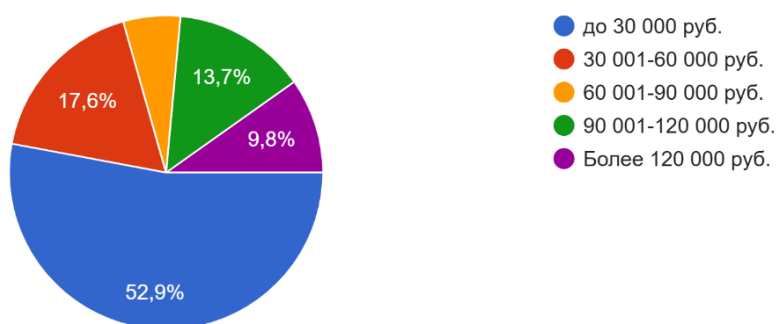


**Рисунок 1.8 – Возраст участников**

В ходе опроса также был выявлен уровень образования участников и их средний ежемесячный доход. Большинство опрошенных имеют высшее образование, 27.5% среднее образование и лишь 13.7% среднее специальное.

Цена является одним из ключевых барьеров для органической продукции, поэтому доход — важный фактор. Большинство опрошенных имеют доход до 30 000 руб., на основании ответов о возрасте, вероятно, это школьники, студенты и люди пенсионного возраста (Рисунок 1.8). Доход от 30 001 – 60 000 руб. имеют 17.6% опрошенных, 13.7% - 90 001 -120 000 руб., 9.8% - более 120 000 руб. Меньшинство опрошенных людей имеют доход в диапазоне 60 001 – 90 000 руб.

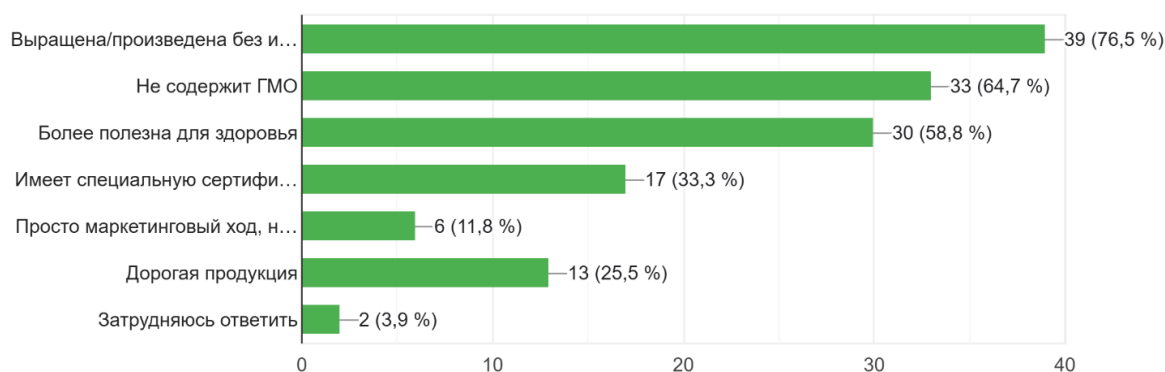
Ваш ежемесячный доход  
51 ответ



**Рисунок 1.9 – Ежемесячный доход**

Следующим шагом необходимо было выявить уровень осведомленности граждан об органической продукции. Почти 70% людей слышали об органической продукции, но не очень разбираются, 23.5% - хорошо знакомы с этим понятием и 7.8% слышат об этом впервые. Следующий вопрос позволил понять, насколько точно респонденты понимают суть органики, выявить распространенные заблуждения или ассоциации.

Что для Вас означает "органическая продукция"? (Выберите все подходящие варианты)  
51 ответ



**Рисунок 1.10 – Сущность органической продукции**

Для большинства опрошенных органическая продукция: выращена/произведена без использования химических удобрений и пестицидов, не содержит ГМО, более полезна для здоровья. Имеет специальную сертификацию, так ответили 33.3%, 25.5% ассоциируют органическую продукцию с высокой ценой, 11.8% считают, что органическая продукция, это просто маркетинговый ход и всего лишь 3.9% затруднились ответить на данный вопрос.

Результатами ответа на вопрос о частоте приобретения товаров органического производства стали – 49 % опрошенных, которые покупают органику иногда, 27.5% не покупают практически никогда, 13.7% - часто (несколько раз в месяц), 9.8% покупают данную продукцию регулярно. Поскольку неясен точный уровень осведомленности потребителей, невозможно сделать корректный вывод о том, насколько часто действительно люди покупают органическую продукцию, однако, явно прослеживается, что люди хотя бы несколько раз в месяц, но стремятся ее покупать, а также то, что у этого сегмента есть регулярные потребители.

Выявим ключевые драйвера для покупки органической продукции.



**Рисунок 1.11 – Мотивация к покупке**

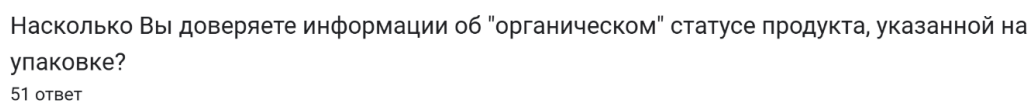
Большинство людей привлекают такие факторы, как забота о здоровье и лучший вкус/качество органической продукции. Также, 23.5% покупают данную продукцию случайно, 21.6% доверяют определенному бренду/производителю, 15.7% – покупают по рекомендациям знакомых и 19.6% не покупают данную продукцию вовсе.

Выясним также и барьеры к покупке, которые может помочь преодолеть информационная платформа. Главной проблемой является высокая цена, а также отсутствие данной продукции в привычных магазинах. Для 29.4% потребителей не понятна разница между органической продукцией и обычной, для 23.5% людей барьером является отсутствие достаточной информации о продукте, 19.6% не знают где её найти либо просто не интересуются этой темой и только 13.7% не доверяют маркировке «органический».



**Рисунок 1.12 – Препятствия для покупки**

Следующим выясненным обстоятельством стало то, что большинство людей затруднились ответить на вопрос о том, доверяют ли они информации об органическом статусе продукции, 23.5% доверяют и 19.6% не доверяют. Низкий уровень доверия указывает на потребность в верифицированных источниках.



**Рисунок 1.13 – Доверие к маркировке**

Далее были выявлены конкретные информационные потребности, которые разрабатываемая платформа должна удовлетворять. Для большинства



опрошенных, является важным возможность подтверждения органического сертификата, 49-51% людей хотели бы видеть информацию о фермере/производителе, а также результаты лабораторных анализов, которые могут подтвердить отсутствие пестицидов и т.д., 45.1% людей хотели бы видеть детальную информацию процесса выращивания/производства, около 27.5% потребителям интересны отзывы других потребителей и только 7.8% достаточно информации на упаковке.

Также большинство опрошенных (74,5%) считают необходимым создание единой информационной платформы об органической продукции. В функциональные возможности респонденты определили – возможность заказа органической продукции (покупки с доставкой на дом); базу данных всех сертифицированных органических продуктов и производителей; возможность обратной связи - задать вопрос производителю или органу сертификации, а также возможность ознакомиться с новостями из мира органического сельского хозяйства. Еще одним из встречающихся предложений была реализация функции сканирования QR-кода на упаковке для мгновенного получения информации о продукте и его происхождении, однако такая возможность информационной системы требует улучшения государственной политики в сфере органического сельского хозяйства. Необходимо создание единого формата маркировки для органической продукции, с возможностью ее сканирования. Такое решение имеет место в перспективе развития проектируемой информационной системы.

Информационные платформы органической продукции – это сервисы, предоставляющие информацию о продукции, произведенной без использования синтетических химических веществ, удобрений и других вредных веществ. Они позволяют потребителям получить информацию о различных видах органической продукции, проверить сертификаты качества и найти проверенных производителей [10].

Создание и развитие информационных платформ для органической продукции является ключевым фактором для обеспечения прозрачности, доверия

потребителей и развития рынка. Эти платформы играют важную роль в соединении производителей, дистрибьюторов, ритейлеров и конечных потребителей.

Выделим ключевые особенности информационной платформы для органической продукции [56].

Централизованное управление сертификацией и соответствием. Платформа должна выступать как единый источник верифицированной информации о сертификации всех участников и продуктов. Должен присутствовать единый реестр сертификатов, который позволит автоматически отслеживать сроки действия, статусы (действующий, приостановлен, отозван) органических сертификатов для каждого участника (фермы, заводы, склады) и каждого продукта. Также, имеет место быть, автоматизированная проверка соответствия, которая заключается в том, чтобы система автоматически проверяла, соответствует ли используемое сырье, производственные процессы и поставщики органическим стандартам на каждом этапе цепочки, а в случае нахождения несоответствий, требуется единый механизм фиксации, который позволит отслеживать и устранять нарушения органических стандартов по всей цепочке. Необходимо, чтобы была предусмотрена возможность для органов загружать и обновлять данные о сертификации напрямую в платформу, а также проводить удаленные аудиты и проверки через доступ к данным.

Для того, чтобы гарантировать неизменяемость и доступность истории продукта «от поля до прилавка», следует обеспечить сквозную и верифицируемую прослеживаемость по всей цепочке. Например, это можно сделать с помощью использования распределенного реестра для записи каждого этапа (посев, сбор, транспортировка, переработка, упаковка, продажа). Благодаря, блокчейн-технологии, это обеспечит неизменяемость и прозрачность данных, так как каждая транзакция будет криптографически защищена и видна всем участникам.

Платформа должна предоставлять полную информацию, с возможностью отследить не только партию, но и конкретное поле, дату сбора, использованные органические удобрения/биопрепараты, условия хранения и транспортировки.

Интерактивные карты или графики, показывающие весь путь продукта, также будут актуальны для данной задачи.

Платформа как хаб для коммуникации и координации между всеми участниками органической экосистемы обеспечивает многостороннее взаимодействие и управление цепочкой поставок [12]. Следовательно, необходимо предусмотреть единую точку входа: для фермеров, переработчиков, дистрибьюторов, ритейлеров и сертификационных органов. Каждый участник цепочки должен иметь возможность координировать объемы производства, заказы, поставки, логистику, учитывая органические требования. Целесообразно организовать электронный документооборот (счета, накладные, сертификаты качества) в рамках платформы, а также разработать функционал для управления логистикой, где можно отслеживать перемещения органической продукции с учетом требований к раздельному хранению и транспортировке.

Предоставление потребителям прямого и легкого доступа к подтвержденной информации о продукте, позволит обеспечить прозрачность и прямое взаимодействие. Это можно сделать с помощью QR-кодов / NFC-меток, уникальных идентификаторов на упаковке, сканирование которых ведет на веб-страницу платформы с полной историей продукта (от поля до прилавка), информацией о фермере, сертификатах, использованных методах. Для более повышенного интереса и спроса на продукт, в его историю можно включить рассказ о происхождении, людях, стоящих за продуктом, и применяемых органических практиках. К тому же, ключевым моментом являются механизмы обратной связи для потребителей, благодаря которым, они могли бы оставлять отзывы, задавать вопросы, сообщать о проблемах. Прямой доступ к верифицированным данным значительно повышает доверие потребителей к органическому статусу продукта.

Платформа должна быть гибкой, чтобы адаптироваться к разным масштабам бизнеса и новым требованиям. В этом могут помочь открытые API для интеграции с существующими ERP, WMS, CRM системами участников [12]. Целесообразно иметь микро-сервисную архитектуру, которая позволит добавлять новые функции или модифицировать существующие без влияния на всю систему. К тому же,

платформа должна быть масштабируемой, способной обрабатывать растущий объем данных и количество пользователей по мере развития органического рынка.

Аналитика и Big Data для экосистемы являются важным аспектом при создании платформы, ведь они позволяют собирать и анализировать агрегированные данные со всей цепочки для получения ценных инсайтов [38]. Благодаря анализу данных о производстве, запасах, заказах и продажах для оптимизации планирования, мы можем прогнозировать спрос и предложение. Также, необходимо выявлять наиболее эффективные маршруты и методы транспортировки, что будет способствовать оптимизации логистики. Анализ рисков поможет выявить потенциально узкие места или проблемы в цепочке поставок. К тому же, важной функцией является автоматическая генерация отчетов о соблюдении стандартов, объемах производства и реализации органической продукции.

Для того, чтобы гарантировать целостность, конфиденциальность и доступность данных в многопользовательской среде, требуются технологии обеспечения доверия и безопасности. Одной из таких технологий является блокчейн, поскольку помимо прослеживаемости, он обеспечивает децентрализованное хранение данных, что затрудняет их подделку. Помимо этого, можно использовать смарт-контракты, которые удобны тем, что условия сделок выполняются автоматически (например, оплата при подтверждении доставки и органического статуса). Ключевым моментом обеспечения безопасности является защита конфиденциальной информации, для этого необходимо использовать шифрование данных. Также, следует строго разграничивать права доступа к данным для разных типов пользователей.

Платформа для органической продукции, таким образом, выходит за рамки простого учета и контроля, становясь инструментом для построения доверия, повышения прозрачности и эффективности всей органической экосистемы.

Платформы могут быть классифицированы по их основному назначению и целевой аудитории:

1. Платформы органов по сертификации и государственных регуляторов.

Указанные платформы предоставляют официальную информацию о сертифицированных органических фермах, производителях и стандартах. Основной задачей данных платформ является подтверждение подлинности и регулирование.

В качестве примеров информации можно привести списки сертифицированных операторов, стандарты органического производства, правила импорта/экспорта.

## 2. Электронные коммерческие площадки и онлайн-магазины.

Довольно часто, данные площадки и магазины продают органические продукты с подробной информацией о продукте, его сертификации и происхождении. Основная идея заключается в прямых продажах, что также удобно для потребителя.

Информация, с которой покупатель может ознакомиться, включает в себя: описание продукта, сертификационные знаки, отзывы, цены, условия доставки.

## 3. B2B-платформы (Business-to-Business).

B2B-платформы соединяют органических производителей с оптовыми покупателями, дистрибьюторами и переработчиками [45]. Таким образом, благодаря B2B-платформам, осуществляется оптовая торговля, поиск партнеров, оптимизация цепочек поставок.

Информация, которую предоставляют данные платформа содержит в себе: каталоги оптовых товаров, объемы поставок, сертификаты, условия сотрудничества.

## 4. Информационные порталы и базы данных некоммерческих организаций/отраслевых ассоциаций.

Указанные платформы собирают и распространяют информацию о рынке органической продукции, исследованиях, новостях и событиях. Это необходимо для развития отрасли и осведомленности интересующихся лиц.

Данные порталы и базы могут включать в себя: отчеты о рынке, статистику, новости, статьи, каталоги производителей.

## 5. Потребительские приложения и агрегаторы.

Определенные приложения помогают потребителям находить органические продукты в магазинах, сканировать этикетки для проверки сертификации, сравнивать цены. Информирование конечного потребителя является важным аспектом, так как это порождает доверие к приобретаемой продукции, что также важно для развития отрасли.

Например, данные приложения могут указывать местоположение магазинов с органической продукцией, покупатель может осуществлять проверку сертификации по штрихкоду, получать информацию о составе продукта, указывать свои диетические предпочтения.

Ключевая информация, которую предоставляют большинство платформ [103]:

- **Сертификация:** Указание органа по сертификации, номер сертификата, срок действия, используемый стандарт (например, USDA Organic, EU Organic, Bio Suisse, ГОСТ 33980-2016 для России).

- **Происхождение:** Страна, регион, иногда конкретная ферма или производитель.

- **Методы производства:** Описание органических практик, отсутствие пестицидов, ГМО, антибиотиков и т.д.

- **Свойства продукта:** Состав, пищевая ценность, аллергены, срок годности.

- **Устойчивость:** Дополнительные экологические или социальные метрики (например, углеродный след, справедливая торговля).

- **Рыночные данные:** Цены, доступность, тенденции.

- **Отзывы и рейтинги:** Мнения других потребителей.

К основным функциям информационных платформ органической продукции можно отнести следующее:

1. Сведения о продуктах.

Платформы содержат описание органических продуктов, их состав, происхождение и производственные методы.

2. Проверка сертификатов:

Позволяют убедиться, что продукт соответствует стандартам органического производства и имеет действующий сертификат соответствия.

### 3. Поиск поставщиков:

Информационные платформы помогают найти проверенных производителей и поставщиков органической продукции.

### 4. Актуальная информация:

Платформы регулярно обновляются, чтобы обеспечить потребителей самой свежей информацией о рынке органических продуктов.

### 5. Образование:

Также некоторые платформы предоставляют информацию о пользе органических продуктов и правилах выбора качественной органической продукции.

Далее, рассмотрим информационные платформы мирового уровня.

К глобальным и международным платформам относятся:

IFOAM – Organics International (ifoam.bio): ведущая международная организация по органическому сельскому хозяйству. Предоставляет обширную информацию о стандартах, политике, исследованиях и рынке органической продукции по всему миру. Ежегодно публикует подробный отчет "The World of Organic Agriculture" совместно с FiBL.

Organic-Bio.com: международный каталог и поисковая система для органических продуктов, производителей и дистрибьюторов. Позволяет искать продукты по категориям, странам и типам сертификации.

Ecosert: один из крупнейших в мире органов по органической сертификации. Данный веб-сайт предоставляет информацию о стандартах, процессах сертификации и имеет базу данных сертифицированных операторов по всему миру.

В Европе можно выделить следующие известные платформы:

EU Organic Farming: официальный портал Европейского Союза по органическому сельскому хозяйству. Предоставляет информацию о регулировании, логотипе ЕС, статистике и базе данных сертифицированных операторов.

Bio Suisse: ведущая швейцарская органическая организация с очень строгими стандартами. Их сайт содержит базу данных сертифицированных производителей и продуктов, а также обширную информацию о швейцарском органическом рынке.

Soil Association: крупнейший орган по органической сертификации в Великобритании. Предоставляет списки сертифицированных ферм и предприятий, а также информацию для потребителей о преимуществах органической продукции.

Популярные платформы Северной Америки:

USDA National Organic Program: официальный портал Министерства сельского хозяйства США. Содержит базу данных сертифицированных органических предприятий (Organic Integrity Database), стандарты, правила и образовательные ресурсы.

Organic Trade Association: Ведущая бизнес-ассоциация, представляющая органический сектор в Северной Америке. Предоставляет рыночные данные, новости, информацию о политике.

Thrive Market: крупный онлайн-ритейлер органических и натуральных продуктов в США, предлагающий широкий ассортимент товаров с подробной информацией о сертификации и происхождении.

Платформы Азии, Австралии, Океании и Африки:

China Organic: национальный портал Китая по органической продукции, предоставляющий информацию о сертификации, стандартах и рынке.

India Organic: портал Совета по содействию экспорту сельскохозяйственной и переработанной пищевой продукции (APEDA) Индии, предоставляющий информацию о национальной органической сертификации и экспорте.

Australian Certified Organic (ACO): крупнейший орган по органической сертификации в Австралии. Их сайт предоставляет базу данных сертифицированных операторов, стандарты и образовательные материалы.

African Organic Network (AfrONet): сеть, объединяющая органические инициативы в Африке, предоставляет информацию о развитии органического сектора на континенте.



В России развитие информационных платформ для органической продукции активно началось после вступления в силу Федерального закона от 03.08.2018 № 280-ФЗ "Об органической продукции и о внесении изменений в отдельные законодательные акты Российской Федерации" с 1 января 2020 года. Этот закон установил четкие правила для органического производства и создал основу для государственной системы сертификации и контроля.

Ключевые платформами и ресурсами в России являются:

1. Единый государственный реестр производителей органической продукции (Роскачество) – это главная официальная платформа [56]. Роскачество является уполномоченным органом, который ведет и поддерживает Единый государственный реестр производителей органической продукции. В этом реестре содержится информация обо всех российских компаниях и фермерских хозяйствах, которые прошли сертификацию на соответствие национальному органическому стандарту (ГОСТ 33980-2016) и имеют право маркировать свою продукцию российским органическим знаком (белый лист на зеленом фоне).

Информация в данном реестре содержит название и контактные данные производителя, номер и срок действия сертификата, наименование сертифицирующего органа, виды сертифицированной продукции. Потребители могут легко проверить подлинность органической продукции, отсканировав QR-код на упаковке, который ведет к записи в этом реестре, благодаря этому Роскачество помогает обеспечить максимальную прозрачность и доверие к российскому органическому рынку.

2. В России действует ряд органов по сертификации, аккредитованных на право проведения органической сертификации. Их сайты также содержат информацию о процессе сертификации, стандартах и иногда списки сертифицированных клиентов.

В качестве примеров, к таким сайтам можно отнести: "Органик Сертификация", "Органик-Тест", "Эко-контроль" и др.

3. Многие российские онлайн-магазины и крупные ритейлеры имеют выделенные разделы для органической продукции. Например, «ВкусВилл» - сеть

магазинов здорового питания, активно работающая с органическими поставщиками. На их сайте и в приложении часто указывается органический статус продуктов. «Азбука Вкуса» - премиум-ритейлер, предлагающий широкий ассортимент импортной и российской органической продукции. Ozon, Wildberries, SberMarket: крупнейшие маркетплейсы, где продавцы могут размещать органические продукты, указывая соответствующие сертификаты.

4. Сайты отраслевых ассоциаций и некоммерческих организаций, такие как союз органического земледелия: одна из ведущих общественных организаций, объединяющая производителей, переработчиков и потребителей органической продукции. На их сайте можно найти новости отрасли, аналитические материалы, информацию о выставках и образовательных программах.

Фермерские кооперативы и интернет-магазины фермерской продукции: многие небольшие фермерские хозяйства объединяются для прямых продаж, создавая собственные сайты или используя платформы-агрегаторы, где часто указывается, что продукция выращена без использования агрохимикатов, хотя не всегда имеет официальную органическую сертификацию.

5. Многие экологические и ЗОЖ-порталы, а также деловые СМИ, освещают тему органической продукции в России, публикуя новости, аналитику и интервью с участниками рынка.

### **1.3. Проектирование и реализация цифровой платформы для органической продукции**

В рамках проектирования информационной системы одним из важнейших шагов является четкое определение требований к архитектуре, функциональным возможностям и технологиям. Данный раздел будет формулировать техническое задание для ИС, которое послужит основой для последующей реализации платформы. Этот раздел должен содержать четкое описание требований к системе, включая цели, функциональные и нефункциональные характеристики, а также технические условия реализации.

## 1. Введение:

### 1.1. Наименование проекта:

«Интернет-платформа (веб-сайт) для популяризации потребления и развития производства органической сельскохозяйственной продукции»

### 1.2. Назначение и область применения:

Агрегация и структурирование актуальной информации об органическом сельском хозяйстве (источник, аккумулирующий актуальную информацию об органической продукции, как для потребителя, так и с целью коммерциализации);

Создание маркетплейса для продажи органической продукции;

Обеспечение взаимодействия между потребителями, производителями и научным сообществом.

## 2. Требования к программе:

### 2.1. Требования к функциональным характеристикам

ИС должна включать следующие модули:

#### 2.1.1. модуль агрегации данных (парсинг и анализ информации)

- автоматический сбор данных из внешних источников (научные статьи, новостные порталы, государственные реестры органических производителей).

- фильтрация и классификация информации.

- регулярное обновление базы данных.

#### 2.1.2 модуль маркетплейса

- каталог органической продукции, с возможностью фильтрации (по категориям, регионам или сертификации)

- личный кабинет пользователя, где будет отображена история заказов, текущая корзина, избранные категории и пр.

- система отзывов и рейтингов.

#### 2.1.3 административный модуль

- управление контентом, в т.ч. добавление или редактирования новостей, изменения каталога товаров;

- мониторинг работы информационной системы (модулей ИС)

#### 2.1.4 модуль аналитики и рекомендаций

- анализ поведения пользователей для персонализации контента
- рекомендация товаров и предложений релевантных для пользователя

## 2.2. Требования к надежности и безопасности

### 2.2.1 Защита данных пользователей – шифрование персональных данных.

Все персональные данные пользователей, включая платежную информацию, должны храниться в зашифрованном виде (с использованием алгоритмов AES-256 или выше), передача данных должна осуществляться по защищенным протоколам (HTTPS, TLS 1.2+), помимо этого должны быть реализованы механизмы защиты от атак Man-in-the-Middle.

Требования к защите должны соответствовать стандартам GDPR для обработки персональных данных европейских пользователей, ФЗ-152 «О персональных данных» для российских пользователей, обработка платежных данных должна быть сертифицирована согласно стандарту безопасности данных индустрии платежных карт PCI DSS.

Что касается управления доступом – должна быть организована двухфакторная аутентификация для административного доступа, журналирование всех действий, связанных с персональными данными, а также механизмы обнаружения аномальных действий.

### 2.2.2 Отказоустойчивость системы – резервное копирование базы данных, автоматическое восстановление после сбоев.

Резервное копирование включает incremental backup (резервное копирование, позволяющее сохранять только последние изменения, внесенные с последнего сохранения) для всех критически важных данных. Полное еженедельное резервное копирование. Политика хранения – 30 дней для ежедневных, 12 месяцев для еженедельных бэкапов.

Обеспечение восстановления не позднее, чем через 1 час для критических компонентов, с максимально допустимой потерей данных не более 15 минут. Системе должен быть обеспечен круглосуточный мониторинг состояния,

оповещения при достижении критических порогов, а также система прогнозирования нагрузки.

#### 2.2.3 Аппаратная и программная надежность.

Аппаратная надежность будет определяться требованиями к серверному оборудованию, а также за счет резервирования мощностей. Они включают в себя избыточное использование всех критических компонентов, использование серверов уровня Tier III+ (с гарантированным uptime 99,9%).

Программная надежность заключается в использовании исключительно лицензионного ПО, его регулярного обновления, контроле целостности программных компонентов. Помимо этого, необходимо разработать актуальную документацию по восстановлению, поддерживать ее актуальность, регулярно отрабатывать аварийные ситуации, а также проводить нагрузочные тесты, тестирование на отказоустойчивость.

#### 2.2.4 Ролевая модель, с целью ограничения доступа.

Введение ролевой модели подразумевает разделение на уровни доступа пользователей системы.

- Администратор – полный доступ ко всем функциям.
- Модератор – управление контентом и пользователями.
- Контент-менеджер – наполнение каталога.
- Аналитик – доступ к статистике.
- Пользователь – базовые функции ИС.

С целью безопасности необходимым будет логирование всех административных действий, регулярные проверки журналов доступа, интеграция с SIEM-системами для анализа угроз.

#### 2.2.5 Организационно-технические мероприятия.

К организационно-техническим мероприятиям относится обязательно обучение сотрудников по информационной безопасности, регламенты реагирования на внештатные ситуации, план восстановления после аварий, регулярные проверки на соответствие требованиям, в т.ч. физических помещений.

### 3. Условия эксплуатации системы:

### 3.1. Требования к технической инфраструктуре.

3.1.1. Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики:

- Температурный режим в серверных помещениях: +18...+24°C
- Влажность: 40-60%
- Обеспечение бесперебойного электропитания (ИБП + дизель-генератор)
- Защита от пыли и прямого солнечного излучения

#### 3.1.2 Требования к серверному оборудованию:

- Процессор: современные многоядерные решения (Intel Xeon Silver/Gold или AMD EPYC)
- Оперативная память: 64-128 ГБ DDR4 ECC (в зависимости от нагрузки)
- Хранение данных:
  - SSD-накопители для системного ПО и БД (минимум 512 ГБ)
  - HDD/SSD-массивы для хранения резервных копий (от 2 ТБ)
- Сетевое оборудование: гигабитные адаптеры с резервированием каналов

#### 3.1.3 Программное обеспечение.

Кадровое обеспечение реализации проекта представляет собой перечень должностных лиц заказчика, в обязанности которых будет вменена реализация функционала ИС.

- Операционные системы:
  - Серверная часть: Linux (Ubuntu LTS/CentOS)
  - Клиентская часть: Windows 10/11 или современные дистрибутивы Linux

#### 3.1.4 Требования к персоналу

- **Системный администратор** (1 штатная единица):

- Образование: высшее техническое (IT, прикладная информатика)
- Обязательные навыки:
  - Администрирование Linux/Windows-серверов
  - Работа с системами мониторинга (Zabbix, Prometheus)
  - Настройка и обслуживание СУБД (PostgreSQL/MySQL)
  - Обеспечение информационной безопасности
- Дополнительные требования:
  - Опыт работы с облачными платформами (AWS, Yandex Cloud)
  - Знание технологий контейнеризации (Docker, Kubernetes)
- **Технический специалист поддержки** (1 штатная единица):
  - Обязанности:
    - Первичная обработка обращений пользователей
    - Мониторинг работоспособности системы
    - Ведение технической документации
  - Требования:
    - Базовые знания SQL и работы с API
    - Опыт работы с системами тикетинга (Jira, Redmine)
- **Разработчик/DevOps-инженер** (по необходимости):
  - Поддержка и развитие функционала системы
  - Автоматизация процессов развертывания и мониторинга

#### 4. Требования к программной документации:

Состав программной документации должен включать в себя:

- 4.1. техническое задание;
- 4.2. программу и методики испытаний;
- 4.3. руководство пользователя;

4.4. текст кода.

## 5. Стадии и этапы разработки системы

Данный раздел определяет последовательность работ по созданию ИС, включая ключевые этапы с описанием действий, указанием сроков и ответственных исполнителей.

5.1. Стадия проектирования – формализация требований и проектирование архитектуры:

- Анализ требований, в т.ч. изучение предметной области, определение функциональных возможностей.
- Проектирование системы, в т.ч. разработка технической архитектуры, создание инфологической модели, визуальное моделирование, прототипирование.
- Формирование ТЗ – формулирование требований к функционалу, безопасности, надежности. Определение стека технологий.

5.2 Стадия разработки – создание рабочей версии системы:

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;
2. разработка программной документации;
3. испытания программы.

Разработка программы включает в себя Backend-разработку (реализацию API, разработку модуля парсинга данных, разработку модуля рекомендательной системы настройку базы данных);

Frontend-разработку (верстку интерфейса, интеграцию с API, адаптацию под мобильные устройства);

Интеграцию и тестирование компонентов (связку фронтенда и бэкенда).

5.3 Стадия тестирования и отладки системы.

- Функциональное тестирование с проверкой всех сценариев использования, тестирование API.



- Нагрузочное тестирование, оптимизация производительности.
- Тестирование информационной безопасности – пентест и исправление уязвимостей.

5.4 Стадия внедрения и сопровождения системы – запуск и поддержка системы.

### 5.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;
2. определение и уточнение требований к техническим средствам;
3. определение требований к программе;
4. определение стадий, этапов и сроков разработки программы и документации на неё;
5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение методики испытаний;
2. проведение приемо-сдаточных испытаний;
3. корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

### 6. Порядок контроля и приемки:

Приемо-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки. Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком Программы и методик испытаний. Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

#### 6.1. Виды испытаний

- Функциональное тестирование
- Нагрузочное тестирование
- Тестирование безопасности

#### 6.2 Процесс проведения испытаний

- Испытания проводятся в тестовой среде, максимально приближенной к production
- Длительность испытательного периода: 14 календарных дней
- Тестирование выполняется совместно представителями Заказчика и Исполнителя
- Для облачных решений предоставляется доступ к staging-окружению

#### 6.2. Общие требования к приемке работы

На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи программы в эксплуатацию.

Инфологическое проектирование – этап создания концептуальной модели системы, в которой должны быть отражены основные понятия и взаимодействия, связанные с информационной системой. На этом этапе определяется, как данные будут собираться, храниться и обрабатываться; как пользователи будут взаимодействовать с системой; какие сущности и процессы составляют основу системы. Цель инфологического проектирования – визуализировать логику работы системы до начала программирования, чтобы избежать ошибок архитектуры.

Проектируемая информационная система представляет собой комплекс взаимосвязанных компонентов, обеспечивающих реализацию информационных

процессов в сфере органического сельского хозяйства, основу системы составляют четыре ключевых модуля, функционирующих, как единый механизм.

Платформа представляет собой уникальное, для данного сегмента в нашей стране, сочетание информационно-аналитического ресурса и маркетплейса органической продукции, создавая устойчивую экосистему для всех участников рынка. Ключевой особенностью является то, что система реализует два взаимосвязанных, но самостоятельных направления работы: формирование базы знаний об органическом сельском хозяйстве и возможность торговых операций экологически чистой продукцией.

Информационно-аналитический блок системы выполняет функцию интеллектуального аккумулятора данных из различных научных и новостных источников. Используя алгоритмы интеллектуального анализа и обработки естественного языка, данный модуль выделяет ключевые термины и понятия, классифицирует данные по тематическим направлениям, устанавливает взаимосвязи между различными материалами, формируя базу знаний и рекомендации для пользователей на ее основе.

Торговый блок платформы представляет собой полный цикл функционала маркетплейса. ИС предоставляет пользователям доступ к каталогу сертифицированной органической продукции с инструментами для поиска и сравнения товаров, возможность проведения безопасных платежей и информацию о поставщиках. Особенностью системы является глубокая интеграция информационного и торгового блока. Это означает, что покупатели имеют возможность изучать историю фермерских хозяйств, получать информацию о технологиях производства, а также ознакомиться с подтверждающими сертификатами. Производителям, в свою очередь, будет доступна возможность публикации своих достижений, презентации своих технологий и инноваций, участия в тематических обсуждениях.

Административный модуль системы обеспечивает модерацию контента и проверку достоверности, дает возможность управления правами доступа, формирования отчетов, настройки бизнес-процессов. Модуль аналитики реализует

персонализацию контента, работу рекомендательных систем, анализ рыночных тенденций, прогнозирование.

Функционирование системы достигается за счет взаимодействия всех компонентов через единую платформу данных, что обеспечивает согласованность информации, быстрый доступ к актуальным данным, гибкость в настройке и масштабировании. Таким образом, платформа создает уникальную среду, где потребителям будет доступно не только приобретение органической продукции, но и получение информации о ее происхождении, технологиях производства, а также общая информация из мира органического сельского хозяйства, достижениях и новшествах в этой отрасли.

Бизнес-процесс организации состоит из последовательности действий (подпроцессов), направленных на получение заданного результата, ценного для организации.

Рассмотрим бизнес-процесс работы пользователя в модуле ИС автоматизированной группировки экономических субъектов. Работа этого процесса начинается с того, что пользователь заходит на веб-сайт под своим логином и паролем, если аккаунт ранее был создан, в противном случае стоит воспользоваться формой регистрации. Далее пользователю открывается главная страница сайта, воспользовавшись интерфейсом можно сразу перейти на интересующие вкладки. После выбора интересующего раздела пользователь имеет возможность перейти на сайт партнера для получения полной актуальной информации, а также для связи напрямую с поставщиком.

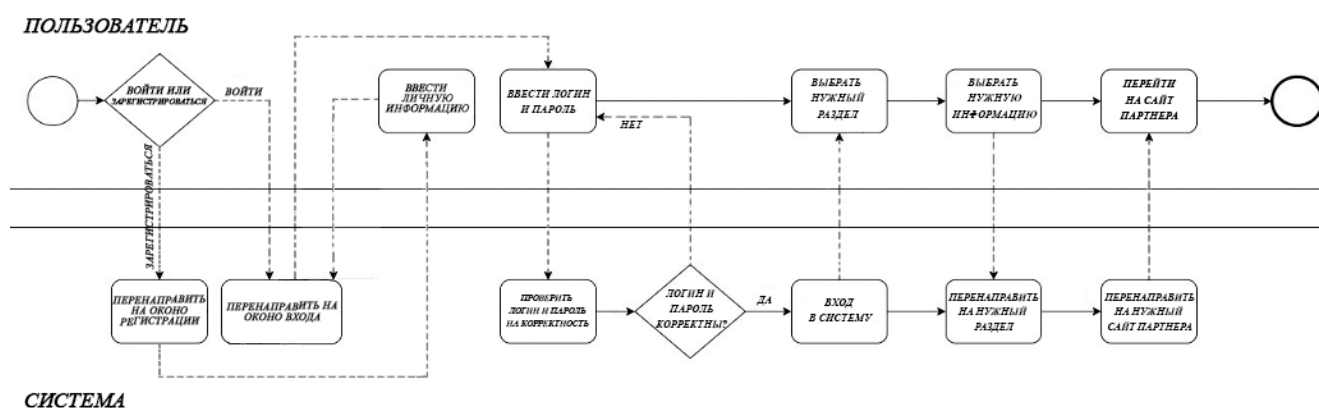
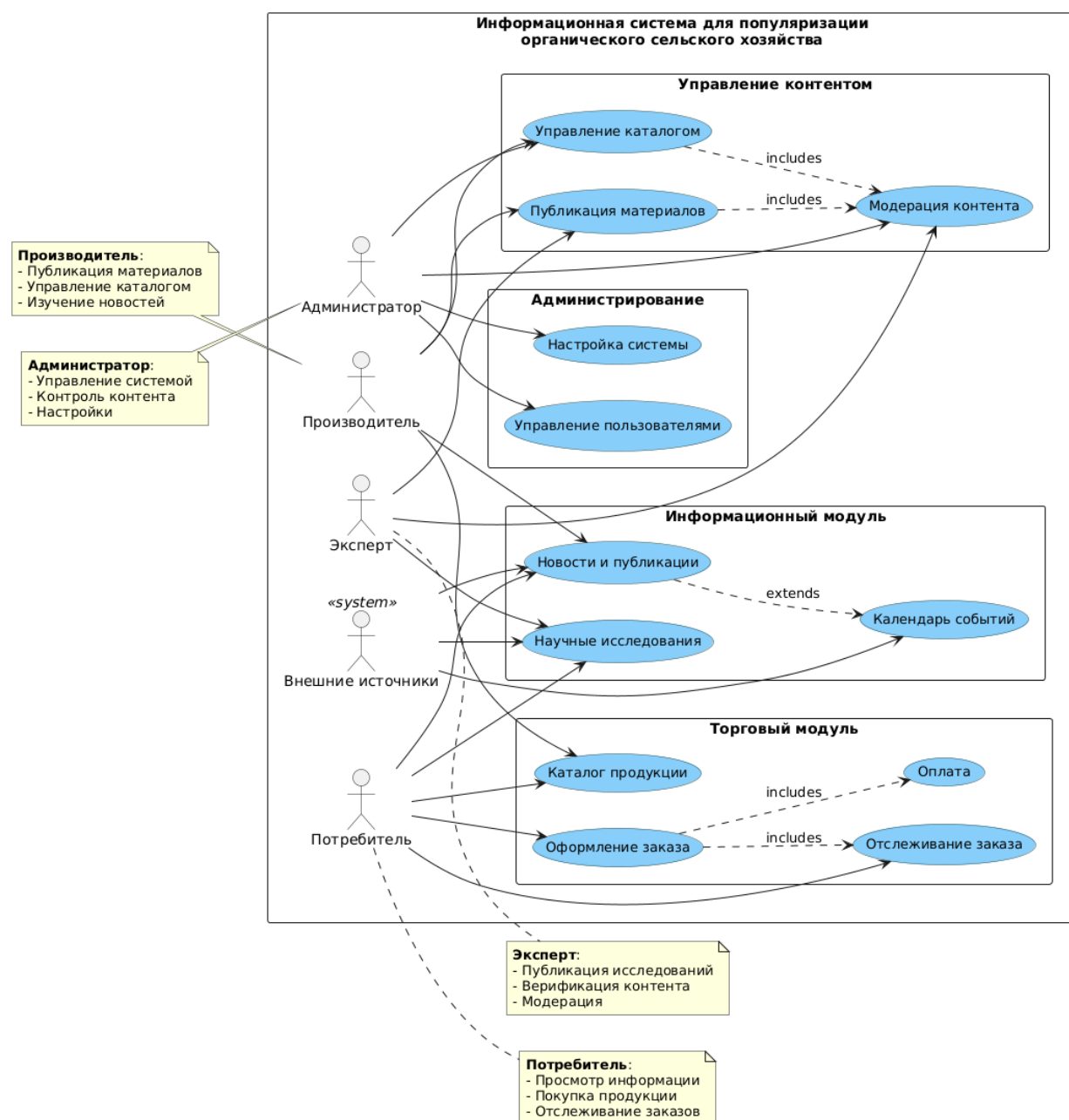


Рисунок 1.14 – Модель бизнес-процесса работы пользователя

В качестве графического представления для выделения основных функций системы воспользуемся диаграммой вариантов использования.

Диаграмма вариантов использования – это диаграмма, которая показывает набор вариантов использования и действующих лиц, а также их связи. [20]

Диаграмма, представленная на рисунке 1.15 представляет визуальную модель ИС для популяризации органического сельского хозяйства, отображающую взаимодействие различных участников с платформой. В центре системы – прямоугольник, объединяющий все варианты использования, сгруппированные в четыре логических модуля.



**Рисунок 1.15 – Диаграмма вариантов использования**

Основные акторы расположены по периметру и связаны стрелками с соответствующими функциями. Администратор системы осуществляет управление ее компонентами – контролирует работу пользователей, параметры платформы, весь контент. Эксперты отвечают за качество контента – публикуют научные исследования, проверяют качество поступающей информации. Внешние источники предоставляют системе новости, информацию о произошедших и предстоящих событиях. Производители также участвуют в наполнении системы – они публикуют материалы о своей деятельности, управляют информацией о

товарах, а также имеют возможность знакомиться с актуальными новостями отрасли. Потребители взаимодействуют с платформой через просмотр информации, изучение каталога товаров и возможностью заказать выбранный товар, с функциями оплаты и отслеживания товара.

Диаграммы деятельности – это графическое представление последовательности действий или процессов в системе, диаграмма деятельности представляет собой блок-схему, которая показывает, как поток управления переходит от одной деятельности к другой. Т.е. она показывает, как выполняются бизнес-процессы, потоки управления – ветвления или параллельные действия, а также участников процесса. С помощью диаграммы деятельности мы описываем последовательность действий для пользователя, необходимую для достижения поставленной цели (Рисунок 1.16).



**Рисунок 1.16 – Диаграмма деятельности парсера**

На рисунке 1.16 представлена диаграмма деятельности для процесса публикации новостей в системе. Для данной ИС важно отразить взаимодействие экспертов с модулем, отвечающим за парсинг данных. Парсер собирает сырые данные. Эксперты выявляют скрытые взаимосвязи, между исследованиями,

формируют аналитические обзоры, объясняют сложную терминологию для конечного пользователя. Т.е. парсер собирает новые материалы, система автоматически удаляет явные дубликаты, классифицирует по темам и пр., эксперт одобряет или отклоняет материал, корректирует его, добавляет комментарии, после этого возможна публикация.

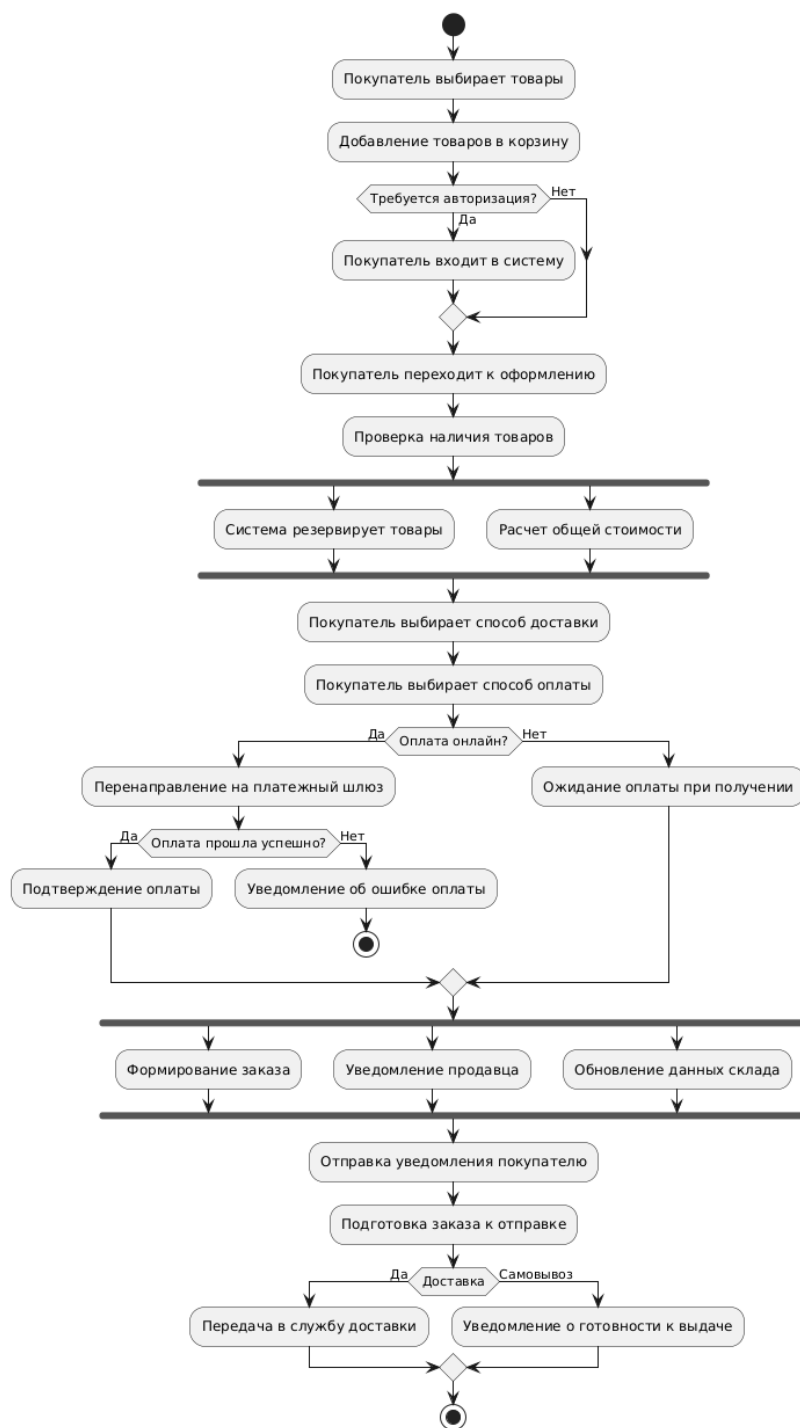


Рисунок 1.17 – Диаграмма деятельности пользователя в системе



Находясь в каталоге товаров, пользователь выбирает интересующие его позиции, добавляет их в корзину, если он не авторизован, ему потребуется ввести логин и пароль, система проверяет правильность ввода. Если данные верны корректно, то покупатель переходит к оформлению заказа, если данные введены некорректно, то пользователь должен повторить попытку ввода. Система проверяет количество товаров, находящихся в наличии, резервирует их, выставляет клиенту счет общей стоимости. Покупатель выбирает способ доставки и оплаты, далее система перенаправляет его на нужную страницу. Если оплата произведена, формируется заказ, продавец получает уведомление, данные склада обновляются. Заказ формируется и готовится к отправке.

Для реализации информационной системы был выбран современный технологический стек, обеспечивающий эффективную разработку всех функциональных модулей. Основой серверной части стал фреймворк Django на языке Python, который был выбран благодаря своей надежности, высокой скорости разработки и наличию встроенных решений для типовых задач. Django предоставляет готовую административную панель для управления контентом, что значительно ускоряет процесс наполнения системы данными. Для организации API использован Django REST Framework, позволяющий создать удобный интерфейс взаимодействия между клиентской и серверной частями системы.

Клиентская часть реализована с использованием библиотеки React.js, которая обеспечивает высокую производительность пользовательского интерфейса за счет виртуального DOM и компонентного подхода. В сочетании с фреймворком Bootstrap это позволяет создавать адаптивные и удобные интерфейсы, одинаково хорошо работающие на различных устройствах. Для хранения данных выбрана СУБД PostgreSQL, обладающая высокой надежностью и подходящая для работы с большими объемами структурированной информации.

Особое внимание было уделено инструментам для сбора и обработки данных. Библиотека BeautifulSoup позволяет эффективно парсить информацию с различных веб-ресурсов, автоматически обновляя базу знаний системы. Дополнительно в системе реализован модуль рекомендаций на основе анализа пользовательских

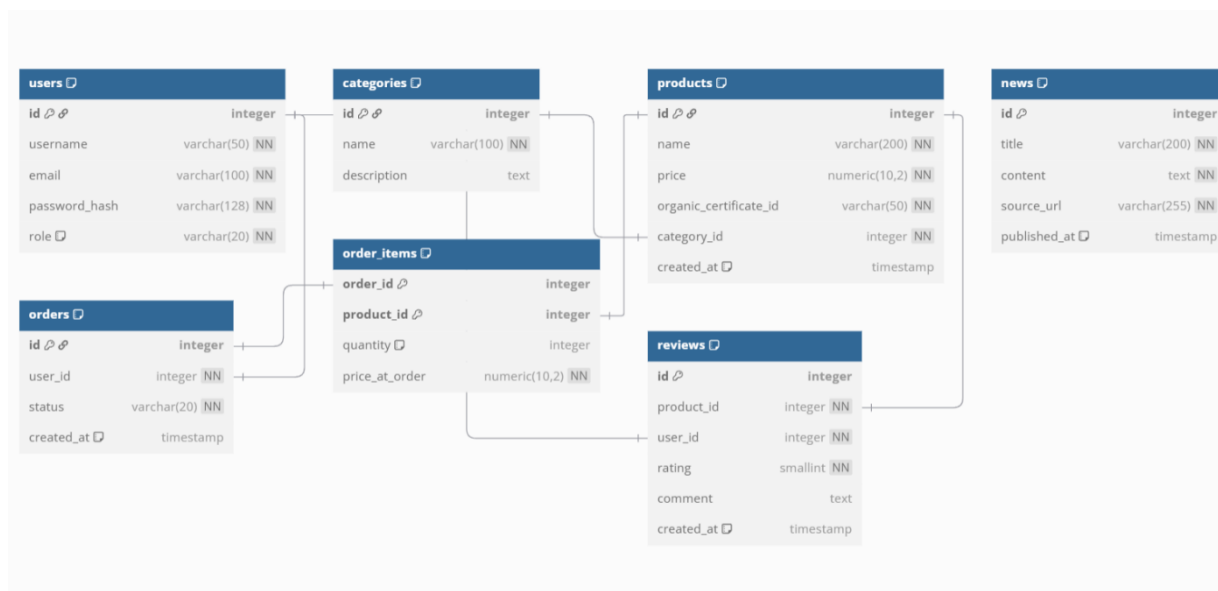
предпочтений и характеристик товаров, использующий алгоритмы машинного обучения из библиотеки Scikit-learn. В качестве среды разработки использован Visual Studio Code - современный редактор кода с широкими возможностями кастомизации и поддержкой всех необходимых технологий.

Выбранный стек технологий обладает рядом ключевых преимуществ:

- обеспечивает высокую скорость разработки благодаря готовым решениям для типовых задач;
- позволяет создавать надежные и производительные системы;
- имеет хорошую масштабируемость для будущего расширения функционала;
- поддерживается большим сообществом разработчиков;
- хорошо документирован и имеет множество обучающих материалов.

Такой набор инструментов оптимально подходит для реализации всех компонентов системы: от модуля сбора и обработки данных до пользовательского интерфейса маркетплейса, обеспечивая при этом высокое качество и удобство дальнейшего сопровождения проекта.

Современные информационные системы, обладающие сложной структурой и широкими функциональными характеристиками, требуют четкого разделения компонентов и модулей для возможности обеспечения масштабируемости, редактирования и исправления ошибок. Разрабатываемая ИС сочетает электронную коммерцию, агрегацию контента, рекомендательную систему, что указывает на ее сложную структуру, поэтому к ней была применена трехуровневая архитектура, которая включает: уровень данных, уровень бизнес-логики и презентационный уровень. Такой подход поможет эффективно распределять задачи между подсистемами, а также может обеспечить гибкость для дальнейшего развития платформы.



**Рисунок 1.18 – ER-диаграмма**

Уровень данных отвечает за хранение, управление и доступ к данным. Он реализован на PostgreSQL – надежной реляционной СУБД, которая обеспечивает целостность и безопасность информации. В базе данных хранится информация и сведения о пользователях, товарах, заказах, отзывах и новостях (рисунок 1.18).

Уровень бизнес-логики построен на Django и Django REST Framework и в нем реализуются основные функциональные возможности приложения. Django REST Framework обеспечивает удобную работу с данными через объектно-реляционное отображение, валидацию входящих запросов и гибкую настройку API для фронтенда. Здесь же реализованы ключевые возможности платформы – парсинг данных из внешних источников с помощью библиотеки BeautifulSoup, рекомендательная система, анализирующая предпочтения пользователей с использованием Scikit-learn, обработка заказов и управление правами доступа для пользователей системы.

Уровень представления отвечает за взаимодействие с пользователем и отображение информации. Реализован на React.js — одним из самых популярных фреймворков для создания динамических пользовательских интерфейсов.

После проектирования архитектуры и структуры базы данных следует перейти непосредственно к реализации системы с использованием выбранного

стека технологий. Backend системы разработан с использованием Django и Django REST Framework (DRF). Он обеспечивает:

- Работу с базой данных через ORM
- REST API для фронтенда
- Аутентификацию пользователей
- Парсинг новостей
- Логику рекомендаций

На фрагменте кода, представленном на рисунке был импортирован базовый модуль Django для работы с моделями БД. Первая модель описывает категории товаров с двумя основными полями – `name` и `description`. Для удобства также был реализован метод `str`, который определяет, как будет отображаться название категории (например, «Овощи» вместо `Category object (1)`). Вторая модель представляет непосредственно товары, содержащиеся в каталоге. Помимо прочего в ней существует поле `created_at`, которое автоматически фиксирует дату создания записи. Эти модели формируют основу структуры данных приложения, обеспечивая хранение информации о товарах, их категоризацию и связь с пользователями-продавцами. Использование `PROTECT` для категорий предотвращает потерю данных при случайном удалении, а `CASCADE` для пользователей автоматически очищает связанные товары при удалении аккаунта продавца.

```

1  from django.db import models
2  from django.contrib.auth import get_user_model
3
4  User = get_user_model()
5
6  class Category(models.Model): 1 usage
7      """Модель категорий товаров"""
8      name = models.CharField(max_length=100, unique=True)
9      description = models.TextField(blank=True)
10
11     def __str__(self):
12         return self.name
13
14     class Product(models.Model):
15         """Модель Товаров с органической сертификацией"""
16         name = models.CharField(max_length=200)
17         price = models.DecimalField(max_digits=10, decimal_places=2)
18         organic_certificate_id = models.CharField(max_length=50, unique=True)
19         category = models.ForeignKey(Category, on_delete=models.PROTECT)
20         seller = models.ForeignKey(User, on_delete=models.CASCADE, related_name='products')
21         created_at = models.DateTimeField(auto_now_add=True)
22
23     def __str__(self):
24         return f"{self.name} (ID: {self.organic_certificate_id})"

```

**Рисунок 1.19 – Модуль Django для работы с моделями БД**

В коде, представленном ниже реализован универсальный парсер новостей, который работает по следующему алгоритму – сначала происходит инициализация списка источников для парсинга, где каждый источник-партнер (информация собирается исключительно из источников, согласных на сотрудничество с нашей ИС) задается в виде словаря с параметрами URL-адреса, HTML-тега и класса элемента, содержащего новости. Основная функция `parse_organic_news` последовательно обрабатывает каждый источник из этого списка.

```

1  import requests
2  from bs4 import BeautifulSoup
3  from .models import NewsItem
4
5
6  def parse_organic_news():
7      sources = [
8          {'url': 'https://example-organic-news.com', 'tag': 'div', 'class': 'news-item'},
9      ]
10
11     for source in sources:
12         response = requests.get(source['url'])
13         soup = BeautifulSoup(response.text, 'html.parser')
14
15         for item in soup.find_all(source['tag'], class_=source['class']):
16             title = item.find('h3').text
17             content = item.find('p').text
18             NewsItem.objects.update_or_create(
19                 title=title,
20                 defaults={'content': content, 'source_url': source['url']}
21             )

```

**Рисунок 1.18 – Модуль парсинга данных**

Парсер ищет все элементы по соответствующим заданным параметрам тега и класса, перебирая их в цикле, для каждого источника. Для каждого найденного новостного элемента осуществляется извлечение заголовка из первого встретившегося тега h3 и основного текста из первого тега p внутри этого элемента. Полученные данные сохраняются в базу данных. Этот код представляет собой базовый каркас парсера, который можно адаптировать под различные новостные сайты путем изменения параметров в списке sources.

Новость становится видимой для пользователя только после проверки ее администратором, опубликованные новости появляются в соответствующих разделах сайта. Под каждой новостью указывается ее источник, с активной ссылкой на оригинальную публикацию. Дабы обеспечить безопасность в юридических аспектах стоит также ограничить частоту запросов, а также реализовать возможность оперативного удаления контента по запросу правообладателя. Модель, используемая для парсинга контента из конкретного источника представлена в приложении.

```
class OrganicProductRecommender:
    def __init__(self):
        self.vectorizer = TfidfVectorizer(
            stop_words=['organic', 'bio', 'eco'],
            ngram_range=(1, 2),
            max_features=5000
        )
        self.scaler = MinMaxScaler()
        self.feature_weights = {
            'purchase_count': 0.35,
            'discount': 0.25,
            'rating': 0.2,
            'novelty': 0.1,
            'text_similarity': 0.1
        }
```

**Рисунок 1.19 – Класс OrganicProductRecommender**

Этот код определяет класс OrganicProductRecommender, который предназначен для рекомендательной системы органических продуктов. Он учитывает наиболее значимые для возможной рекомендации товаров факторы.

Система учитывает популярность, скидки, рейтинг и новизну товара, назначая каждому фактору свой вес. Числовые данные нормализуются.

```
88     def get_popular_products(self, num=8): 1 usage
89         """Рекомендации популярных товаров для главной страницы, в т.ч. для не авторизованных пользователей"""
90         cache_key = f'popular_organic_products_{num}'
91         cached = cache.get(cache_key)
92
93         if cached:
94             return json.loads(cached)
95
96         products = self._get_product_data()
97         features = self._prepare_features(products)
98
99         # Вычисление популярности
100         popularity_scores = []
101         for i, p in enumerate(products):
102             score = (
103                 p.purchase_count * self.feature_weights['purchase_count'] +
104                 (5 if p.is_discounted else 0) * self.feature_weights['discount'] +
105                 p.avg_rating * self.feature_weights['rating'] +
106                 (1 if (timezone.now() - p.created_at).days < 30 else 0) * self.feature_weights['novelty']
107             )
108             popularity_scores.append((i, score))
109
110         # Топ-N товаров
111         top_indices = sorted(popularity_scores, key=lambda x: x[1], reverse=True)[:num]
112         result = [products[i] for i, _ in top_indices]
113
114         cache.set(cache_key, json.dumps(result), 3600)
115         return result
```

### Рисунок 1.20 – Функция get\_popular\_products

Функция get\_popular\_products реализует систему рекомендаций популярных товаров, сочетающую статистический анализ и бизнес-логику для органического каталога. Для каждого товара рассчитывается интегральный показатель популярности по специальной формуле, где учитываются взвешенные вклады четырех ключевых факторов. После расчета показателей для всех товаров происходит их сортировка по убыванию рейтинга популярности и выбор топ-8 позиций. Такие рекомендации особенно полезны новым не зарегистрированным пользователям, впервые попавшим на страницу сайта.

Функция get\_hybrid\_recommendations реализует интеллектуальную систему персонализированных рекомендаций, которая комбинирует несколько подходов для максимально релевантного подбора товаров в личном кабинете пользователя. Система получает общий список популярных товаров, затем, если пользователь имеет историю просмотров, анализируются последние просмотренные товары, для каждого из которых подбираются схожие по характеристикам позиции, используя

функцию `get_content_based_recommendations` (см. код в приложении). Все полученные результаты объединяются, из списка удаляются дубликаты, после чего происходит финальное ранжирование.

```
136 def get_hybrid_recommendations(self, user=None, num=10):
137     """Персонализированные рекомендации"""
138     # Популярные товары
139     popular = self.get_popular_products(num // 2)
140
141     # Персонализированные (если пользователь авторизован)
142     personalized = []
143     if user and hasattr(user, 'product_views'):
144         viewed_products = user.product_views.all()[:3]
145         for p in viewed_products:
146             personalized.extend(
147                 self.get_content_based_recommendations(p.product_id, num // 3)
148             )
149
150     # Смешивание результатов
151     recommendations = list(set(popular + personalized))
152     return sorted(
153         recommendations,
154         key=lambda x: (
155             x.purchase_count * 0.4 +
156             (1 if x.is_discounted else 0) * 0.3 +
157             x.avg_rating * 0.2 +
158             (1 if (timezone.now() - x.created_at).days < 30 else 0) * 0.1
159         ),
160         reverse=True
161     )[:num]
```

**Рисунок 1.21 – Функция `get_content_based_recommendations`**

Таким образом, были рассмотрены и описаны примеры реализации основных функциональных возможностей информационной системы, с полный код программных модулей интернет-платформы представлен в приложении к работе.

Рассмотрим получившийся пользовательский интерфейс и оценим возможности, разработанной ИС. При входе на сайт пользователь попадает на главную страницу. На ней мы видим интуитивно понятное меню для возможности перехода на страницу интересующего раздела (навигация по категориям, кнопка входа в личный кабинет), а также ленту последних новостей и каталог наиболее популярных товаров.



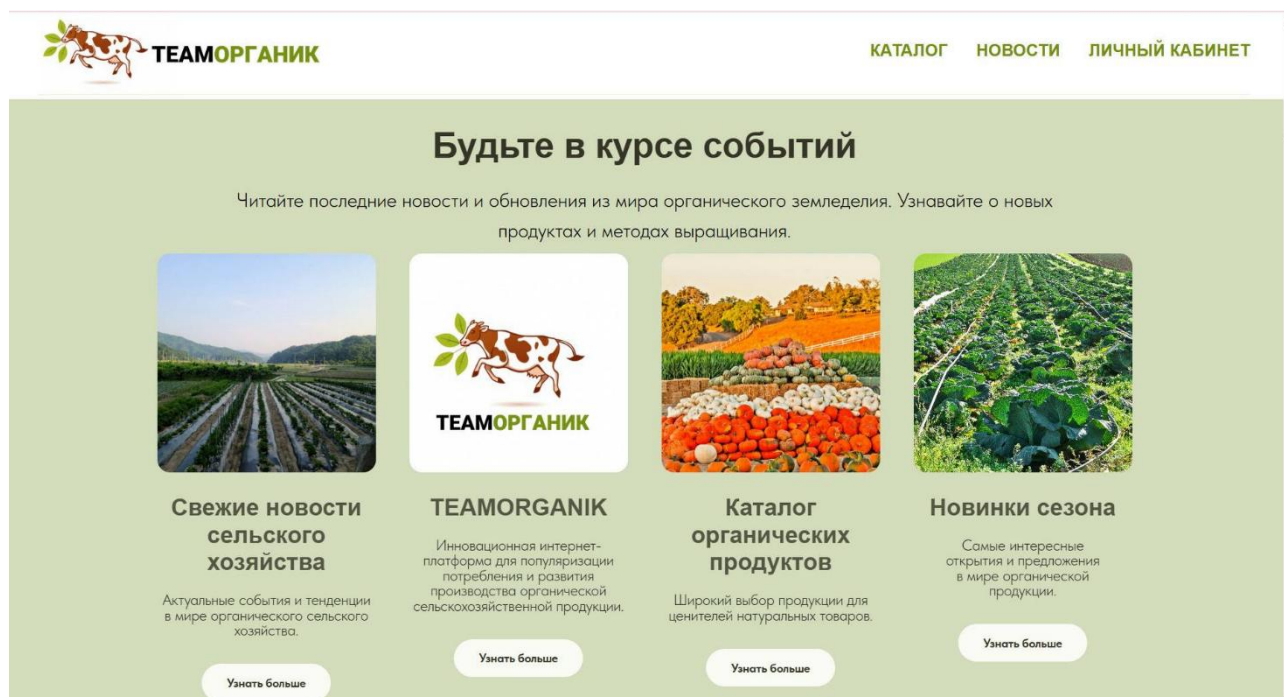


Рисунок 1.22 – Главная страница

На главной странице пользователь может познакомиться с потенциальными партнерами информационной системы. Сразу после новинок из раздела платформа предлагает познакомиться с флагманами в мире популяризации и реализации органической сельхоз продукции, которые может порекомендовать teamorganik (Рисунок 1.23).



Рисунок 1.23 – Главная страница

Каталог представляет собой сетку с изображениями товаров, названиями, ценами. Реализована возможность поиска и фильтрации. Карточка товара содержит детальное описание товара, его характеристики, фото, отзывы пользователей и возможность добавить товар в корзину.

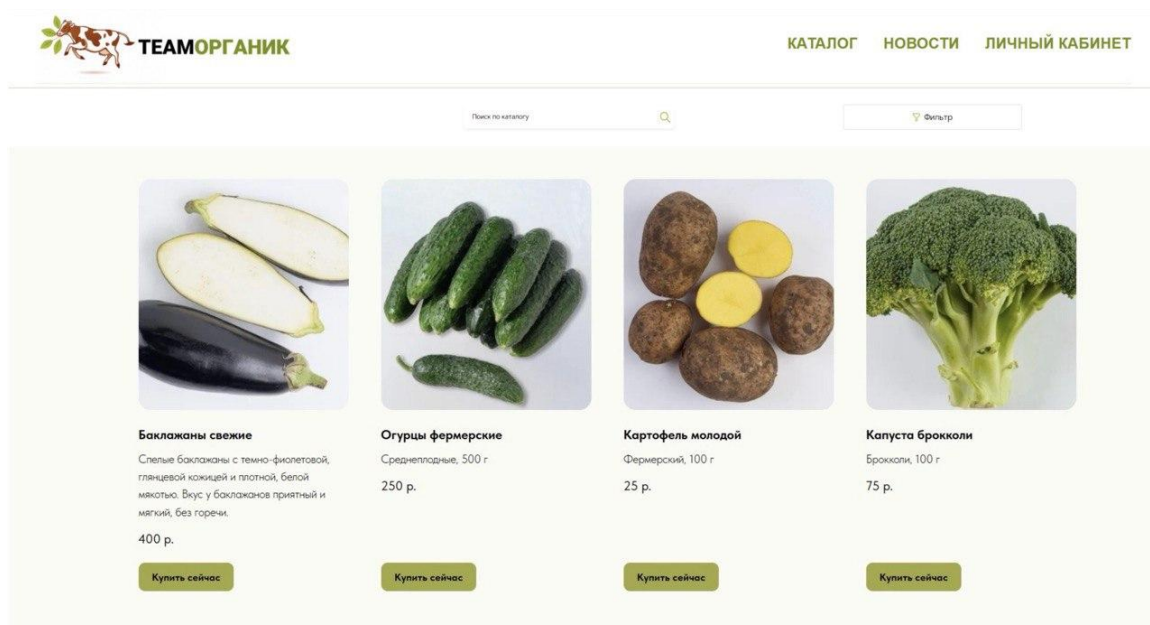


Рисунок 1.24 – Страница «Каталог»

Новостной блок частично был представлен на главной странице, к ленте новостей можно перейти через командную панель.

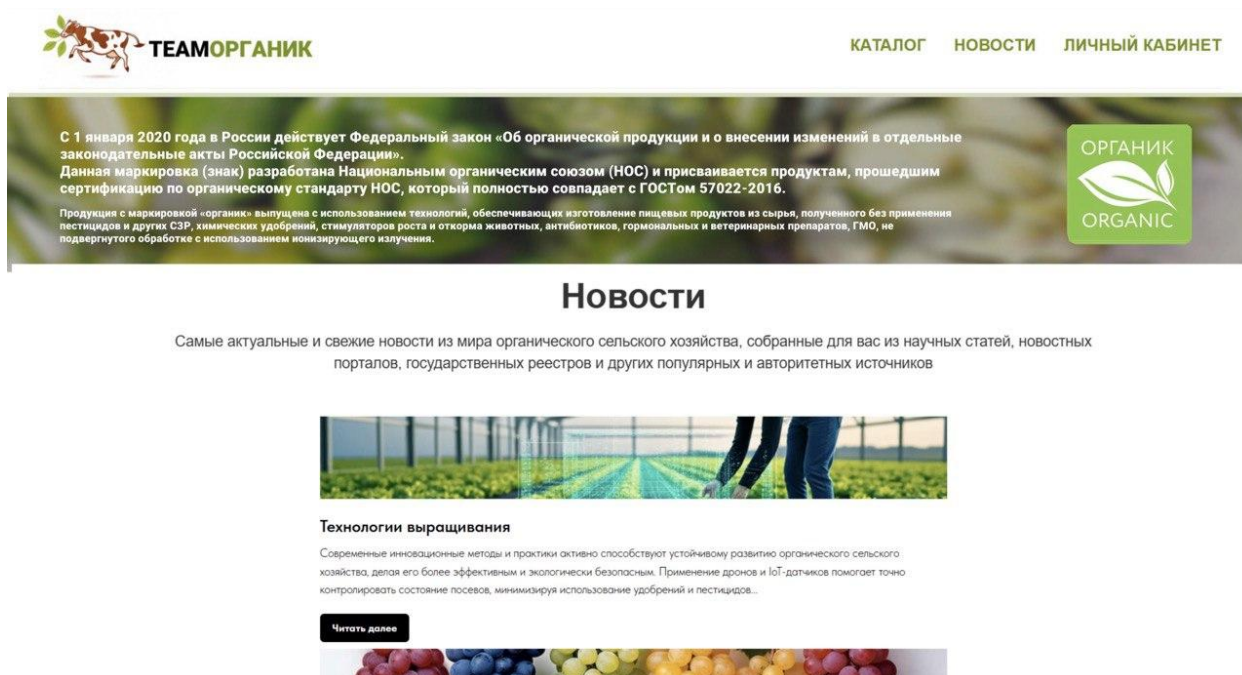


Рисунок 1.25 – Страница «Новости»

Он содержит список всех новостей, для каждой новости есть главное изображение и анонс, а также кнопка «читать далее», нажав на которую можно перейти к странице новости, где будет ее полный текст, дата публикации, источник (при наличии).

Сведения для связи находятся в нижней части каждой страницы, там содержится контактная информация. На ней также есть форма для обратной связи, которая будет полезна и покупателям, и потенциальным партнерам (рис.3.15).

Реализованы также:

- Окно регистрации (авторизации) с формой для входа, личный кабинет, содержащий личную информацию зарегистрированного пользователя с возможностью редактирования.
- Корзина, в которой можно найти список выбранных товаров, суммарную стоимость, кнопка «Оформить заказ».
- Детали заказа – информация о его статусе, сумме, дате и адресе доставки.
- Административная панель.

TEAMORGANIK

КАТАЛОГ НОВОСТИ ЛИЧНЫЙ КАБИНЕТ

Купить сейчас Купить сейчас Купить сейчас Купить сейчас

### Свяжитесь с нами

Вы можете связаться с нами по телефону или электронной почте. Мы всегда рады ответить на ваши вопросы.

8 906 389-06-13  
s.semenova@rgau-msha.ru

Telegram WhatsApp Messenger

Ваш адрес электронной почты

Ваше полное имя

Ваши комментарии

Связаться

**Рисунок 1.26 – Контактная информация**

Проведённое исследование позволило достичь следующих результатов:

- Анализ предметной области: изучены сущность органической продукции, её стандарты и нормативная база, а также современное состояние мирового и российского рынков. Выявлены ключевые тенденции, проблемы и перспективы развития органического сельского хозяйства в России.

- Проектирование информационной системы: разработана архитектура веб-сайта, включающая модули агрегации данных, маркетплейса, аналитики и администрирования. Используются современные технологии (Django, React.js, PostgreSQL), обеспечивающие высокую производительность и безопасность.

- Реализация платформы: создан функциональный прототип, позволяющий пользователям получать актуальную информацию об органической продукции, взаимодействовать с производителями и совершать покупки. Особое внимание уделено удобству интерфейса и персонализации рекомендаций.

Научная ценность заключается в систематизации знаний об органическом сельском хозяйстве и разработке методики создания информационных систем для этой отрасли. Практическая значимость проявляется в предоставлении инструмента для популяризации органической продукции, что способствует развитию рынка, повышению осведомлённости потребителей и поддержке отечественных производителей.

Перспективы развития проекта:

- Расширение функционала платформы за счёт интеграции с государственными реестрами и сертификационными системами.

- Развитие мобильного приложения для увеличения доступности сервиса.

- Внедрение технологий искусственного интеллекта для более точных рекомендаций и анализа данных.

Стоит также отметить, что разработанная интернет-платформа является важным шагом в решении задач информационного обеспечения рынка органической продукции. Её внедрение будет способствовать устойчивому развитию сельского хозяйства, повышению качества жизни населения и укреплению экологической безопасности страны.

## Библиографический список к главе 1

1. Аварский Н. Д. и др. Рынок органической продукции России: современное состояние и потенциал развития // Экономика сельского хозяйства России. – 2014. – №. 5. – С. 29-37.
2. Аварский Н. Д., Таран В. В. Методические и практические аспекты развития инфраструктуры рынка органической продукции России //УПРАВЛЕНИЕ ИННОВАЦИОННЫМ РАЗВИТИЕМ АГРОПРОДОВОЛЬСТВЕННЫХ СИСТЕМ НА НАЦИОНАЛЬНОМ И РЕГИОНАЛЬНОМ УРОВНЯХ. – 2022. – С. 19-28.
3. Алтухов А.И. Отечественная аграрная экономическая наука: вопросов больше, чем ответов // Экономика сельскохозяйственных и перерабатывающих предприятий. – 2017. – № 7. – С. 2-11.
4. Астахова Т. Н., Романова А. А. О разработке информационной системы документооборота сельскохозяйственного предприятия //Новые информационные технологии в образовании. – 2018. – С. 70-73.
5. Белокопытова Л. Е., Крючков Г. Г. ВЛИЯНИЕ ИНТЕРНЕТ-МАРКЕТИНГА НА ЭФФЕКТИВНОСТЬ ПРОИЗВОДСТВА ПРОДУКЦИИ АПК //Научное обозрение: теория и практика. – 2015. – №. 4. – С. 104-108.
6. Белякова З. Ю. Современные правовые формы обеспечения производства и оборота органической продукции //Техника и технология пищевых производств. – 2018. – Т. 48. – №. 3. – С. 140-151.
7. Беркетова Л. В., Акишев Н. С. Европейский рынок экологически чистых продуктов и способы их продвижения //Траектории развития. – 2019. – С. 37-47.
8. Беспяхотный Г. В. Проблемы стратегического планирования развития сельского хозяйства / Г. Беспяхотный // Экономика сельского хозяйства России. – 2013. – №7 – 8. – С. 9 – 15.
9. Бирюкова Т. В. Органическая продукция: основные перспективы развития потребительских предпочтений //Образование и право. – 2020. – №. 4. – С. 409-412.

10. Блиновская Я.Ю. Введение в информационные системы: Учебное пособие / Я.Ю. Блиновская, Д.С. Задоя. -М.: НИЦ ИНФРА-М, 2013. – 112 с.
11. Боровиков А.Д. Анализ производственно-хозяйственной деятельности / А.Д. Боровиков // Научная дискуссия: вопросы экономики и управления – 2017. – 19-24 с.
12. Водяхо, А.И. Архитектурные решения информационных систем / А.И. Водяхо – Санкт-Петербург: Лань, 2017. – 356 с.
13. Волкова Г. А. ПОПУЛЯРНЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ //Агропромышленный комплекс: состояние, проблемы, перспективы. – 2022. – С. 467-470.
14. Воронина Н. П. Государственная поддержка органического сельского хозяйства //Сельское хозяйство. – 2023. – №. 1. – С. 44-50.
15. Голенищев Э.П., Клименко И.В. Информационное обеспечение систем управления. – Ростов-на-Дону: Феникс, 2010.
16. Горбатов А. В. Развитие рынка органической продукции в России //Фундаментальные исследования. – 2016. – №. 11-1. – С. 154-158.
17. Горбатов А. В. Развитие рынка органической продукции в России //Фундаментальные исследования. – 2016. – №. 11-1. – С. 154-158.
18. Григорьев М.В. Проектирование информационных систем: учеб.пособие для вузов / М.В.Григорьев, И.И.Григорьев; Тюменский государственный университет. - М.: Издательство Юрайт, 2017. - 318 с.
19. Дмитриев Я. В. РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ ПРОДАЖИ СЕЛЬСКОХОЗЯЙСТВЕННОЙ ТЕХНИКИ //КОНКУРС ЛУЧШИХ СТУДЕНЧЕСКИХ РАБОТ: сборник статей XXI. – 2024. – С. 6.
20. Евглевская Т. А., Позднякова А. Р. Интернет-маркетинг, как инструмент развития предприятий АПК //Инновационная деятельность в модернизации АПК. – 2017. – С. 33-36.
21. Елешев Р. Е., Балгабаев А. М., Салыкова А. С. Перспективы органического земледелия в Казахстане: состояние и пути дальнейшего развития

//Известия Национальной Академии наук Республики Казахстан. – 2017. – №. 5. – С. 41.

22. Емельянова Н.З. Проектирование информационных систем. – М.: Форум, 2009.

23. ЕМИСС государственная статистика / [Электронный ресурс]. – Режим доступа: <https://www.fedstat.ru>

24. Ермолаев А. Н. Web-технологии как средство совершенствования бюджетного процесса //Вестник Чувашского университета. – 2011. – №. 1. – С. 371-375.

25. Ерофеев Д. В., Рыбакова Т. М. О вопросах сертификации органической продукции. – 2023.

26. Жилина, Е. В. Оценка конкурентоспособности участников продуктового регионального рынка / Е. В. Жилина // Региональная экономика: теория и практика. – 2014. – № 15. – С. 51–57.

27. Затонский А.В. Информационные технологии: разработка информационных моделей и систем: Учеб.пособие. - М.: РИОР: ИНФРА-М, 2014. - 344 с.

28. Землянский, А. А. Информационные системы: учебное пособие / А.А. Землянский; М-во сельского хоз-ва Российской Федерации, Российский гос. аграрный ун-т - МСХА им. К.А. Тимирязева, Учетно-финансовый фак., Каф. прикладной информатики. - Москва: РГАУ-МСХА им. К. А. Тимирязева, 2010. - 173 с.

29. Зинченко А.П. Сельскохозяйственные предприятия: экономико-статистический анализ / А.П. Зинченко. – М.: Финансы и статистика, 2012. – 160 с.

30. Зинченко А.П. Сравнительная оценка регионов России с неблагоприятными условиями ведения сельского хозяйства / А.П. Зинченко, А.В. Уколова, В.В. Демичев // Экономика сельского хозяйства России. – 2013. – №2. – С. 20 – 26.

31. Зинченко А.П. Статистика: учебник / А.П. Зинченко – М.: Издательство РГАУ – МСХА имени К. А. Тимирязева, 2013. – 368 с.



32. Зинченко, А. П., Трутнева Н. Ю. Анализ производства и доходов сельского хозяйства региона в системе национального счетоводства / А. П. Зинченко, Н. Ю Трутнева/ – М.: Изд-во РГАУ-МСХА, 2012. – 160 с.

33. Зинченко, А.П. О развитии информационно-методологического обеспечения экономико-статистического анализа аграрного сектора экономики / А.П. Зинченко // Вопросы статистики. – 2016. – № 1. – С. 9 – 12.

34. Зинченко, А.П. Статистическое наблюдение в сельском хозяйстве: Учебное пособие / А.П. Зинченко. – М.: Изд-во РГАУ-МСХА имени К.А. Тимирязева, 2010. – 24 с.

35. Иванова А. Д. BRAND PORTFOLIO MANAGEMENT OF THE INTERNATIONAL COMPANY IN MODERN ECONOMIC SITUATION. HENKEL BEAUTY CARE DEPARTMENT ANALYSIS //Научное издание. – 2023. – С. 155.

36. Информатика и основы информационных технологий: учебное пособие / В.Д. Шеповалов, С.Д. Шустиков. – Москва: Росинформагротех, 2018. – 200 с.

37. ИОНОВ А. М. RUSSIAN ECONOMIC BULLETIN //RUSSIAN ECONOMIC BULLETIN Учредители: ИП Ключева ММ. – 2024. – Т. 7. – №. 2. – С. 367-372.

38. Ирышков М. М., ВАТОЛКИНА Е. А., КОРОТКОВА Е. С. Проектирование информационной системы интернет-магазина //МОЛОДОЙ УЧЕНЫЙ Учредители: ООО" Издательство Молодой ученый". – 2021. – №. 51. – С. 24-27.

39. Кагирова М.В., Романцева Ю.Н., Семенова С.О. Расширение сегмента потребителей органической продукции сельского хозяйства с использованием информационных технологий// International agricultural journal. 2023. №4, 1067-1086.

40. Кирюшин П. А. Факторы экологически устойчивого развития и "зеленой" экономики в России //Вестник Московского университета. Серия 6. Экономика. – 2019. – №. 1. – С. 122-138. (1)

41. Коваленко В. В. Проектирование информационных систем. – 2021.



42. Козлова О. А. Теория и методология формирования рынка органической продовольственной продукции на основе холистического маркетинга //Автореф. дисс. д-ра экон. наук: спец. – 2011.

43. Колесов Ю.Б., Сениченков Ю.Б. Моделирование систем. Объектно-ориентированный подход. – СПб.: Питер, 2013.

44. Корпоративные информационные системы: учебное пособие. В 2 ч. Ч. 1. Проблемы внедрения и использования / Градусов Д. А., Шутов А. В., Градусов А. Б.; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир: Изд-во ВлГУ, 2014. – С. 70 – 75.

45. Косолапова, М.В. Комплексный экономический анализ хозяйственной деятельности: Учебник / М.В. Косолапова, В.А. Свободин. – М.: Дашков и К, 2016. – 248 с.

46. Кострова Ю. Б., Мартынушкин А. Б. Проблемы развития рынка органической продукции в РФ //Вестник Мичуринского государственного аграрного университета. – 2020. – №. 1. – С. 252-255.

47. Кручинина В. М. Государственное регулирование рынка органической продукции в России //Вестник Воронежского государственного университета инженерных технологий. – 2017. – Т. 79. – №. 2 (72). – С. 296-305.

48. Ксенофонтова, О.Л. Опыт зарубежных стран по созданию и функционированию кластеров: модельный подход / О.Л. Ксенофонтова // Современные наукоемкие технологии. Региональное приложение.-Иваново, 2015. - №2. - 36-42 с.

49. КУЗНЕЦОВ А. В. ОПТИМИЗАЦИЯ РОССИЙСКИХ ВНЕШНИХ ИНВЕСТИЦИОННЫХ СВЯЗЕЙ В УСЛОВИЯХ УХУДШЕНИЯ ОТНОШЕНИЙ С ЕС //ГЕОГРАФИЯ. – 2014. – №. 10. – С. 66-82.

50. Лящук Ю. О. Государственное регулирование рынка экологически чистой и органической продукции //Материалы IV Международной научной конференции «Донецкие чтения. – 2019. – С. 300-303.

51. Мельников В.П. Информационное обеспечение систем управления. – М.: Академия, 2010.

52. Михайлова С. С. РАЗРАБОТКА БЛОКЧЕЙН-ПЛАТФОРМЫ ДЛЯ ТОРГОВЛИ СЕЛЬСКОХОЗЯЙСТВЕННОЙ ПРОДУКЦИЕЙ //Проблемы экономики и юридической практики. – 2024. – Т. 20. – №. 2. – С. 174-184.

53. ОРГАНИЧЕСКАЯ П., САНКЦИОННОЕ Д. Современные технологии и экономика производства органической продукции. – 2023.

54. Органическое сельское хозяйство в России – [Электронный ресурс]: <https://grainrus.com/novosti-kompanii/articles/rynok-organicheskoy-selkhozproduktsii-v-rossii/> (дата обращения: 07.04.2025)

55. Официальный сайт Министерства сельского хозяйства Российской Федерации – [Электронный ресурс]: <http://opendata.mcx.ru/opendata/> (дата обращения: 07.04.2025)

56. Официальный сайт Роскачества – [Электронный ресурс]: <https://rskrf.ru/> (дата обращения: 02.04.2025)

57. Папцов А. Г. и др. Стратегические направления развития рынка органической продукции России. – 2020.

58. Печенкина В. В., Егоров А. Ю. Рынок органической агропродукции //Экономика сельского хозяйства России. – 2012. – №. 8. – С. 50-59.

59. Пивоваров В. Ф. и др. Нормативно-правовое обеспечение рынка органической продукции (в мире, странах ЕАЭС, России) //Овощи России. – 2021. – №. 1. – С. 5-19.

60. Пирогов В. Ю. Информационные системы и базы данных: организация и проектирование. – БХВ-Петербург, 2009.

61. Полуэктова Н. Р. Современные тенденции в развитии корпоративных информационных систем // БизнесИнформ. 2013. №3. – С. 227-230.

62. Полянин А. В., Остроухова Д. К. Основные теоретические аспекты предпринимательской деятельности в современных условиях //Деловой вестник предпринимателя. – 2022. – №. 1 (7). – С. 199-206.

63. Савицкая, Г. В. Анализ хозяйственной деятельности предприятий АПК / Г.В. Савицкая. – М.: ИНФРА-М, 2017. – 656 с.

64. Саменбетова Д. С. Поддержка органического сельского хозяйства как приоритет структурных преобразований аграрных отраслей экономики регионов стран ЕАЭС //ББК 65.04 С69.

65. Семенова С.О. Разработка WEB-приложения для реализации натуральных кормов // Научный сетевой журнал «Столыпинский вестник» №8/2023.

66. Скоропанова Л., Шпак А. Органическое сельское хозяйство в контексте мировых и региональных тенденций //Наука и инновации. – 2016. – Т. 10. – №. 164. – С. 15-18.

67. Соколова А. П., Сухарева О. А. Инновационная активность предприятий АПК Российской Федерации: тренды и возможности роста //Естественно-гуманитарные исследования. – 2023. – №. 45 (1). – С. 217-222.

68. Статья Р. Новые контексты органического сельского хозяйства //Аграрная наука. – 2019. – №. 3. – С. 10-11.

69. Сухомлинов А.И. Разработка информационных систем. – М.: Проспект, 2015.

70. Тарасов С.Б., Викторов А.Д. Кластеры – основа эффективного развития экономики региона // Инновации. 2007. № 2. 4–7 с.

71. Узун, В. Я. Адаптация аграрной политики к требованиям ВТО / В. Я. Узун // Экономика сельскохозяйственных и перерабатывающих предприятий. – 2014. – №4. – С. 20 – 24.

72. Учетно-информационное обеспечение целевых программ // Экономика сельскохозяйственных и перерабатывающих предприятий. – 2013. – № 1. – С. 1-4.  
<http://elibrary.ru/item.asp?id=18840793>

73. Ушачев, И. Г. Экономические проблемы развития АПК России и пути их решения / И. Г. Ушачев, А. И. Алтухов// Вестник Российской академии сельскохозяйственных наук. – 2013. – № 1. – С. 5 – 7

74. Урпебай А., Нургулжанова А. Н. РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ДИСЦИПЛИНЕ" БАЗЫ ДАННЫХ" С ИСПОЛЬЗОВАНИЕМ

WEB-ТЕХНОЛОГИИ //Инновационные технологии на транспорте: образование, наука, практика. – 2018. – С. 296-301.

75. Федеральная служба государственной статистики / [Электронный ресурс]. – Режим доступа: <http://www.gks.ru>

76. Федеральный закон от 03.08.2018 № 280-ФЗ "Об органической продукции и о внесении изменений в отдельные законодательные акты Российской Федерации" // Собрание законодательства РФ. – 2018. – № 32. – Ст. 5133.

77. Федеральный закон от 27.07.2006 № 149-ФЗ "Об информации, информационных технологиях и о защите информации" // Собрание законодательства РФ. – 2006. – № 31 (1 ч.). – Ст. 3448.

78. Федеральный закон от 29.12.2006г. № 264-ФЗ «О развитии сельского хозяйства» (ред. от 29.07.2018) [Электронный ресурс]. – Режим доступа: <http://www.consultant.ru>.

79. Фёдоров Н.В. Проектирование информационных систем на основе современных CASE-технологий / Н.В. Фёдоров – М.: Изд-во МГИУ, 2008. – 345 с.

80. Хетагуров Я.А. Проектирование автоматизированных систем обработки информации и управления. – М.: Бином, 2015.

81. Хорошилов А.В. Предметно-ориентированные экономические информационные системы. – М.: Финансы и статистика, 2011.

82. Цены, финансы и воспроизводство в сельхозорганизациях / Н.А. Борхунов, О.А. Родионова // АПК: Экономика, управление.– 2013.– № 2.– С. 30 – 37. <http://elibrary.ru/item.asp?id=18969899>

83. Чупина И. П., Воронина Я. В. Возрождение российского рынка органической продукции //Аграрный вестник Урала. – 2019. – №. 4 (183). – С. 96-100.

84. Швиденко О. М. Внедрение Интернет-маркетинга в сельскохозяйственных предприятиях //Международный научно-производственный журнал " Экономика АПК". – 2014. – №. 1. – С. 129-135.

85. Шинакова Н. Ю. СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ МАРКЕТИНГА ДЛЯ СЕЛЬСКОХОЗЯЙСТВЕННЫХ И ПЕРЕРАБАТЫВАЮЩИХ

ПРЕДПРИЯТИЙ АПК //ИННОВАЦИОННЫЙ ПУТЬ РАЗВИТИЯ  
ПРЕДПРИЯТИЙ АПК. – 2017. – С. 169-175.

86. Щетинина И.В. Перспективы развития агропромышленных кластеров в России // Экономика сельскохозяйственных и перерабатывающих предприятий. М., 2019, №3, с. 51-55

87. Экономика предприятия (организации) АПК: Учебник / Р.Г. Ахметов, А.В. Голубев, Р.С. Гайсин, А.Е. Шибалкин и др. М.: Издательство РГАУ-МСХА имени К.А. Тимирязева, 2013. – 618 с.

88. Abiri R. et al. Application of digital technologies for ensuring agricultural productivity //Heliyon. – 2023. – Т. 9. – №. 12.

89. Attaran M. Digital technology enablers and their implications for supply chain management //Supply Chain Forum: An International Journal. – Taylor & Francis, 2020. – Т. 21. – №. 3. – С. 158-172.

90. Bolfe É. L. et al. Precision and digital agriculture: Adoption of technologies and perception of Brazilian farmers //Agriculture. – 2020. – Т. 10. – №. 12. – С. 653.

91. Deichmann U., Goyal A., Mishra D. Will digital technologies transform agriculture in developing countries? //Agricultural Economics. – 2016. – Т. 47. – №. S1. – С. 21-33.

92. European Commission. Regulation (EU) 2018/848 on organic production and labelling of organic products. - Brussels: EU Publications, 2018. - 92 p. - URL: <https://eur-lex.europa.eu> (date accessed: 01.07.2024).

93. FiBL. Organic Farming and Digital Technologies [Electronic resource] / Research Institute of Organic Agriculture. - Frick, 2022. - URL: <https://www.fibl.org> (date accessed: 01.07.2024).

94. Green, R. User Experience Design for E-Commerce Platforms / R. Green. - Sebastopol: O'Reilly, 2022. - 287 p.

95. Lavuri R. et al. Green factors stimulating the purchase intention of innovative luxury organic beauty products: Implications for sustainable development //Journal of Environmental Management. – 2022. – Т. 301. – С. 113899.

96. Moscovici D. et al. Consumer preferences for organic wine-Global analysis of people and place //Journal of Cleaner Production. – 2022. – Т. 368. – С. 133215.

97. Mushi G. E., Di Marzo Serugendo G., Burgi P. Y. Digital technology and services for sustainable agriculture in Tanzania: A literature review //Sustainability. – 2022. – Т. 14. – №. 4. – С. 2415.

98. Regulation (EU) 2018/848 of the European Parliament and of the Council of 30 May 2018 on organic production and labelling of organic products // EUR-Lex : [official website]. – URL: <https://eur-lex.europa.eu/eli/reg/2018/848/oj> (дата обращения: 28.04.2025)

99. Rotz S. et al. The politics of digital agricultural technologies: a preliminary review //Sociologia ruralis. – 2019. – Т. 59. – №. 2. – С. 203-229.

100. Sanders, R. E-Commerce Platform Development for Agricultural Products / R. Sanders. - Hoboken: Wiley, 2022. - 324 p.

101. Thompson, S. Digital Marketing Strategies for Organic Products / S. Thompson // Journal of Agricultural Marketing. - 2023. - Vol. 12. - No. 2. - P. 45-59

102. Trendov N. M., Varas S., Zeng M. Digital technologies in agriculture and rural areas. – FAO;, 2019..

103. USDA. Organic Industry Survey 2023: Market Analysis and Trends. - Washington: U.S. Department of Agriculture, 2023. - 78 p. - URL: <https://www.ers.usda.gov> (date accessed: 01.07.2024).

104. USDA. (2023). Organic Market Overview [Electronic resource]. – URL: <https://www.ers.usda.gov/topics/natural-resources-environment/organic-agriculture/organic-market-overview/> (дата обращения: 28.04.2025).

105. Von Briel F., Davidsson P., Recker J. Digital technologies as external enablers of new venture creation in the IT hardware sector //Entrepreneurship Theory and Practice. – 2018. – Т. 42. – №. 1. – С. 47-69.

106. Willer H., Trávníček J., Schlatter B. The world of organic agriculture. Statistics and emerging trends 2024. – 2024.

Willer, H. The World of Organic Agriculture: Statistics and Emerging Trends / H. Willer, J. Lernoud. - Frick: FIBL, 2023. - 178 p

## **Глава 2. Информационная система мониторинга показателей устойчивого развития сельского хозяйства**

### **2.1 Теоретико-методологические основы мониторинга целей устойчивого развития в аграрном секторе**

Концепция устойчивого развития подразумевает необходимость сбалансированного роста, который учитывает не только экономические интересы, но и экологические и социальные. Это означает, что развитие экономики должно происходить таким образом, чтобы не наносить ущерба окружающей среде и природным ресурсам, а также обеспечивать социальную справедливость и равенство возможностей для всех членов общества.

Непосредственно к целям устойчивого развития относится набор из 17 глобальных целей (Таблица 2.1) для содействия экономическому социальному и экологическому развитию на планете. Они были приняты всеми государствами-членами ООН, включая Россию, в качестве инструмента для улучшения благосостояния людей и окружающей среды на период до 2030 года и далее.

**Таблица 2.1 – Цели устойчивого развития**

<b>№ п/п</b>	<b>Наименование целей устойчивого развития</b>
<i>1</i>	<i>2</i>
1	Ликвидация нищеты
2	Ликвидация голода
3	Хорошее здоровье и благополучие
4	Качественное образование
5	Гендерное равенство
6	Чистая вода и санитария
7	Недорогостоящая и чистая энергия
8	Достойная работа и экономический рост
9	Индустриализация, инновация и инфраструктура
10	Уменьшение неравенства

*Продолжение таблицы 2.1*

<i>1</i>	<i>2</i>
11	Устойчивые города и население планеты
12	Ответственное потребление и производство
13	Борьба с изменениями климата
14	Сохранение морских экосистем
15	Сохранение экосистем суши
16	Мир, правосудие и эффективные институты
17	Партнерство в интересах устойчивого развития

Успех деятельности по осуществлению выполнения Целей устойчивого развития зависит от разработанных странами стратегий, планов и программ в области устойчивого развития, так как цели не являются юридически обязательными. Страны несут основную ответственность за проведение на национальном, глобальном и региональном уровнях последующей деятельности и обзора прогресса в реализации этих целей и задач к 2030 году [54].

Одним из основных вызовов сегодня остаётся необходимость увеличения объёмов производства сельскохозяйственной продукции в условиях ограниченных ресурсов и изменяющегося климата. По оценкам ФАО, к 2050 году население планеты достигнет 9,7 миллиардов человек, что потребует увеличения мирового производства продовольствия примерно на 60%. При этом площади пашни уже почти достигли своего предела: около 38% территории суши используется в сельскохозяйственных целях. Дальнейшая распашка новых земель приведёт к разрушению естественных экосистем, что негативно скажется на глобальном климате. Поэтому стратегическим направлением должно стать не увеличение площадей, а повышение продуктивности существующих сельскохозяйственных угодий за счёт внедрения инновационных технологий и рационального управления ресурсами.

На основе информации, опубликованной на сайте Росстата для контроля выполнения этой цели, выделены 14 показателей. Из них 6 показателей уже



разрабатываются и публикуются, 1 показатель в процессе разработки и 7 показателей планируются к разработке (Таблица 2.2) [19].

**Таблица 2.2 – Система показателей Цели устойчивого развития №2  
«Ликвидация голода»**

Наименование цели устойчивого развития	Показатели
Ликвидация голода	<p>Разрабатываются:</p> <p>доля домохозяйств, указавших при оценке своего материального положения на нехватку денег на еду;</p> <p>уровень умеренного или острого отсутствия продовольственной безопасности населения;</p> <p>распространенность задержки роста среди детей в возрасте до 5 лет;</p> <p>индекс производства продукции сельского хозяйства в сопоставимых ценах к предыдущему году;</p> <p>доля животных отечественной репродукции, используемых для целей сельскохозяйственного производства на территории Российской Федерации.</p>
	<p>В процессе разработки:</p> <p>качество генетических ресурсов растительного и зоологического происхождения, предназначенных для производства продовольствия и сельского хозяйства, которые хранятся на специальных объектах либо среднесрочного, либо долгосрочного хранения.</p>

	<p>Планируются к разработке:</p> <p>объем производства на производственную единицу в разбивке по классам размера предприятий фермерского хозяйства / скотоводства / лесного хозяйства;</p> <p>средний доход мелких производителей продовольственной продукции в разбивке по полу и статусу принадлежности к коренным народам;</p> <p>доля площади сельскохозяйственных угодий, на которых применяются продуктивные и неистощительные методы ведения сельского хозяйства;</p> <p>индекс ориентированности на сельское хозяйство, определяемый по структуре государственных расходов;</p> <p>совокупный приток официальных средств в сельское хозяйство;</p> <p>субсидирование экспорта сельскохозяйственной продукции;</p> <p>показатель ценовых аномалий на рынке продовольствия.</p>
--	---

В России проблема повышения продуктивности существующих сельскохозяйственных угодий особенно актуальна. Несмотря на значительные площади сельскохозяйственных угодий, уровень их использования остается ниже потенциального. Например, по данным Росстата, около 20% пашни не используется по различным причинам, включая отсутствие инвестиций, старение техники и слабую логистическую инфраструктуру. Это свидетельствует о том, что текущая система мониторинга и оценки сельскохозяйственного производства не позволяет в полной мере выявлять резервы повышения эффективности. Следовательно, необходимо расширить набор используемых показателей, чтобы более точно оценивать степень реализации потенциала сельскохозяйственных угодий и их вклад в обеспечение продовольственной безопасности страны.

Существующая система показателей не всегда позволяют получить полное представление о реальных возможностях страны в обеспечении продовольственной безопасности. Например, высокие объемы производства определённых культур могут маскировать дефицит других, важных для

полноценного питания. Также отсутствует информация о степени соответствия производимой продукции фактическому спросу населения.

Важно отметить, что устойчивое развитие сельского хозяйства невозможно без учета климатических факторов. Сельское хозяйство не только испытывает на себе последствия глобального потепления, но и само вносит значительный вклад в выбросы парниковых газов. Согласно оценкам ФАО, около 24% всех антропогенных выбросов CO<sub>2</sub> связано с деятельностью АПК. Основные источники выбросов — это метан от животноводства, оксид азота от применения удобрений и вырубка лесов под пашню. Поэтому одной из стратегических задач является переход к низкоуглеродному сельскому хозяйству, которое будет использовать возобновляемые источники энергии и применять современные технологии.

Еще одним направлением устойчивого развития является обеспечение справедливой торговли и доступности продовольствия. Несмотря на то, что в мире производится достаточно пищи для всего населения, более 800 миллионов человек страдают от недоедания. Это связано с неравномерным распределением ресурсов, логистическими проблемами и высокими затратами на транспортировку и хранение продуктов. Для решения этих задач необходимо развивать локальное производство, сокращать продовольственные потери и укреплять продовольственные цепочки. Также важно поддерживать программы государственной помощи и школьного питания, особенно в удаленных и социально незащищенных регионах.

Устойчивое развитие сельского хозяйства невозможно без участия государства. Государственная политика должна быть направлена на создание правовой, финансовой и инфраструктурной базы, которая будет способствовать внедрению устойчивых практик. Это включает в себя предоставление субсидий на экологически чистые технологии, развитие системы сертификации устойчивого производства, модернизацию ирригационных систем и проведение просветительских программ среди сельских жителей. Также необходимы изменения в законодательстве, направленные на защиту земель сельскохозяйственного назначения от застройки и недопущение их деградации.

Реализация принципов устойчивого развития в сельском хозяйстве — это не просто вопрос экономической эффективности, а необходимое условие выживания человечества в условиях растущего населения и ограниченных ресурсов планеты. Именно поэтому предлагается расширить систему показателей устойчивого развития сельского хозяйства.

Как было проанализировано ранее, данные, необходимые для расчета показателей ЦУР, собираются множеством различных организаций: Росстатом, Министерством сельского хозяйства Российской Федерации, региональными органами власти, а также международными организациями, такими как ФАО ООН, ВПП и другими. Каждая из этих организаций имеет собственные методологии сбора данных, форматы представления и периодичность предоставления информации. Это приводит к тому, что данные могут быть несопоставимыми или противоречивыми, особенно при анализе долгосрочных тенденций или сравнении регионов.

Кроме того, возникает проблема временной актуальности данных. Для оперативного управления и принятия решений требуется информация, которая позволяет отслеживать изменения в реальном времени или близком к нему.

Не менее важной является проблема пространственной детализации информации. Все данные размещаются, в зависимости от места публикации, с разной детализацией: только по РФ, по РФ и субъектам РФ. Иногда, в зависимости от методологии, данные публикуются с включением перечня новых субъектов или для обеспечения сопоставимости данных в разрезе федеральных округов, действовавших в 2010-е годы.

Отдельную группу проблем составляют несогласованность и отсутствие стандартов представления данных. Отсутствие унифицированного формата хранения и передачи информации затрудняет её интеграцию в единую информационную систему. Например, данные о производстве зерна могут быть представлены в разных единицах измерения (тонны, центнеры), с разной градацией по культурам (например, пшеница, рожь, ячмень) или с разбивкой по категориям хозяйств (крестьянские, сельскохозяйственные организации, ЛПХ). Такие

различия требуют предварительной обработки, стандартизации и согласования, прежде чем данные можно будет использовать для анализа или прогнозирования.

С учётом всех перечисленных проблем становится очевидной необходимость внедрения автоматизированных решений, способных значительно повысить эффективность сбора и обработки данных. Одним из наиболее перспективных подходов является разработка специализированного парсера.

**Парсинг** – это процесс автоматического сбора информации для последующего преобразования и структурирования. Тогда парсер – это специальная программа или автоматический скрипт, выполняющий процесс сбора и обработки информации.

В каждом парсере используются свои подходы, алгоритмы и технические решения. Но есть у них и общие черты, так как непосредственно в процессе парсинга сложно придумать что-то нестандартное, особенно, если речь идёт об анализе web-страниц.

Алгоритм работы почти любого web-парсера выглядит примерно так:

1. Скрипт обращается к конкретному адресу страницы (по заданному URL, к слову, список URL на парсинг может формироваться автоматически, на основе адресов, извлечённых с исходной/стартовой страницы). Чтобы увеличить шансы на одобрение автоматического ответа, парсер может представляться браузером или ботом поисковых машин.
2. После успешного коннекта скрипт получает HTML-код страницы.
3. Далее HTML-код разбирается на теги, внутри страницы выделяются нужные участки или блоки. Для целей более качественного анализа HTML-кода многие парсеры могут иметь встроенный браузер.
4. Выделенные данные копируются в специальную внутреннюю базу (обычно это SQL-формат, но могут использоваться и любые другие).
5. Из внутренней базы данных информация может быть выгружена в другие форматы: CSV, XML, JSON, YAML и пр.
6. Собранные данные хранятся на локальном диске или в облачном хранилище (в зависимости от типа программы).

Первый и самый значимый плюс парсинга – это автоматизация процесса сбора данных. Те же самые данные можно собирать и вручную, но это дорого, долго и потенциально влечёт за собой массу ошибок из-за человеческого фактора.

Для написания парсера, как и другой программы, можно использовать любой язык программирования. Согласно статистике GitHub за 2023 год (Рисунок 2.1) Python занимает лидирующую позицию в списке наиболее используемых языков программирования. Далее располагаются Java, Go, JavaScript. [88]

# Ranking	Programming Language	Percentage (YoY Change)
1	Python	17.380% (-0.537%)
2	Java	11.770% (+0.488%)
3	Go	10.000% (+0.566%)
4	JavaScript	9.949% (+0.074%)
5	C++	9.656% (-0.048%)
6	TypeScript	7.460% (-0.847%)
7	PHP	5.480% (+0.209%)
8	C	4.576% (+0.334%)
9	Ruby	4.533% (-0.103%)
10	C#	3.437% (+0.167%)

**Рисунок 2.1 – Рейтинг языков программирования по популярности**

Помимо перечисленных популярных языков также следует рассмотреть язык R из-за его принадлежности к статистическому анализу данных.

Для того, чтобы определиться с языком для разработки парсера, необходимо проанализировать плюсы и минусы, решаемые задачи с помощью этих инструментов (Таблица 2.3).

**Таблица 2.3 – Сравнение языков программирования**

Язык программирования	Плюсы	Минусы
Python	<ul style="list-style-type: none"> <li>- большое количество фреймворков;</li> <li>- понятный синтаксис облегчит разработку и сопровождение кода;</li> <li>- большое количество онлайн-ресурсов и готовых решений, что позволит ускорить процесс разработки;</li> <li>- кроссплатформенность.</li> </ul>	<ul style="list-style-type: none"> <li>- менее производительный по сравнению с Java при работе с большими объемами данных;</li> <li>- динамическая типизация (часто приводит к ошибкам в коде, что потенциально увеличивает время на отладку).</li> </ul>
Java	<ul style="list-style-type: none"> <li>- большое количество фреймворков;</li> <li>- высокая скорость выполнения программ;</li> <li>- строгая типизация помогает избежать ошибок на этапе компиляции.</li> </ul>	<ul style="list-style-type: none"> <li>- синтаксис сложнее Python и JavaScript, требует больше кода для выполнения тех же задач;</li> <li>- программы часто требуют больше оперативной памяти.</li> </ul>
Go	<ul style="list-style-type: none"> <li>- высокая производительность за счет компиляции в машинный код;</li> <li>- понятный синтаксис и встроенная поддержка параллельного программирования;</li> <li>- подходит для создания высоконагруженных сетевых приложений.</li> </ul>	<ul style="list-style-type: none"> <li>- меньше готовых решений по сравнению с Java и Python.</li> </ul>
JavaScript	<ul style="list-style-type: none"> <li>- большое количество фреймворков;</li> <li>- асинхронность упрощает работу с сетевыми запросами;</li> <li>- универсальность (frontend, backend).</li> </ul>	<ul style="list-style-type: none"> <li>- возникают проблемы с совместимостью версий пакетов и управлением зависимостей;</li> <li>- асинхронность может усложнять чтение и отладку кода.</li> </ul>
R	<ul style="list-style-type: none"> <li>- существуют фреймворки для парсинга, мощные инструменты для анализа и визуализации данных;</li> <li>- широко используется для анализа статистических данных.</li> </ul>	<ul style="list-style-type: none"> <li>- больше подходит для небольших скриптов, а не для разработки крупных систем;</li> <li>- медленнее по сравнению с Java.</li> </ul>

Таким образом, сравнивая приведенные языки программирования, можно сделать следующие выводы:

- Python является лучшим выбором для большинства проектов благодаря своей простоте, большому количеству фреймворков и существующих готовых решений;
- JavaScript хороший выбор для асинхронных операций;
- Java подойдет для высокопроизводительных систем с большой нагрузкой;
- с помощью Go будет проще создать сетевое приложение;
- R идеален с точки зрения анализа и визуализации статистических данных.

Анализируя минусы обозначенных языков программирования, следует остановиться на Python, так как объем данных не колоссально большой, время на отладку заложено в жизненном цикле системы. Фреймворки BeautifulSoup, Scrapy, Pandas, Numpy, Matplotlib и Seaborn идеально подойдут для создания парсера, очистки и преобразовании данных, а также их визуализации.

Таким образом использование парсера позволит снизить влияние человеческого фактора, исключить задержки при сборе информации и повысить точность и достоверность данных. Кроме того, автоматизация этого процесса создаст основу для последующего внедрения машинного обучения и прогнозирования, что особенно важно в условиях быстро меняющейся внешней среды и необходимости оперативного реагирования на кризисные явления в аграрной сфере. Разработка парсера станет важным шагом на пути к созданию полноценной информационной системы, ориентированной на поддержку принятия управленческих решений в сфере продовольственной безопасности и устойчивого развития сельского хозяйства.



## **2.2. Расширение системы индикаторов ЦУР №2 «Ликвидация голода»: уровень самообеспечения и землепользование**

Анализ текущей системы индикаторов ЦУР выявил недостаточную полноту и репрезентативность используемых показателей. Официальные индикаторы, закреплённые за Целью №2, включают такие параметры, как уровень недоедания, доля недоношенных детей, коэффициент потерь и пищевых отходов, а также степень устойчивости сельскохозяйственного производства. Однако данные показатели не всегда позволяют получить всестороннюю картину состояния продовольственной безопасности и устойчивости сельского хозяйства в конкретных регионах. Например, высокие значения уровня продовольственной обеспеченности могут скрывать дефицит отдельных продуктов, необходимых для полноценного питания, или игнорировать особенности локальных рынков и цепочек поставок.

Для контроля продовольственной безопасности можно использовать показатель уровня самообеспечения основными продуктами питания в разрезе регионов Российской Федерации и по стране в целом. Уровень самообеспечения определяется относительно отрасли сельского хозяйства и отрасли промышленного производства. Данный показатель разрабатывается Росстатом, начиная с 1990 года, и отражает в какой мере собственное производство способно удовлетворить потребности населения или так называемое «внутреннее потребление» страны или ее региона.

Что касается отрасли сельского хозяйства, уровень самообеспечения страны или региона по отдельным видам сельскохозяйственной продукции определяется как процентное соотношение производства соответствующих видов сельскохозяйственной продукции к потреблению их на территории страны или региона.

Внутреннее потребление включает в себя производственное и личное потребление, потери продукции и переработку на непищевые цели.

Переработка на непищевые цели включает, например, объемы продукции, израсходованные предприятиями и организациями на выработку медбиопрепаратов, использованные для научно-исследовательских целей [46].

Так же уровень самообеспечения по отдельным видам сельскохозяйственной продукции публикуется в Доктрине продовольственной безопасности, Государственной программе развития сельского хозяйства и в том числе на основе значений данного показателя формируется политика относительно субсидирования и стимулирования регионов Российской Федерации.

Данный показатель может использоваться в целях контроля обеспеченности товарами сельскохозяйственного и промышленного производства в отдельных регионах и по России в целом и показывать, насколько страна зависит от импорта продуктов питания.

Если соотнести уровень самообеспечения по отдельным видам сельскохозяйственной продукции с пороговыми значениями уровня самообеспечения по основным видам продовольствия, разрабатываемыми Министерством сельского хозяйства Российской Федерации и публикуемыми в Доктрине продовольственной безопасности, то, в конечном счете, можно делать вывод о том, насколько возможно полное обеспечение населения продуктами питания.

Динамика изменения уровня самообеспечения основными продуктами питания по Российской Федерации приведена в таблице 2.4.

**Таблица 2.4 – Уровень самообеспечения основными продуктами питания по Российской Федерации, процентов**

Продукт	2019	2020	2021	2022	2023	Пороговые значения, согласно Доктрине продовольственной безопасности
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
Зерно	155,6	165,6	148,3	191,4	173,5	Не менее 95%
Мясо и мясопродукты	97,4	100,1	99,7	101,8	101,7	Не менее 85%

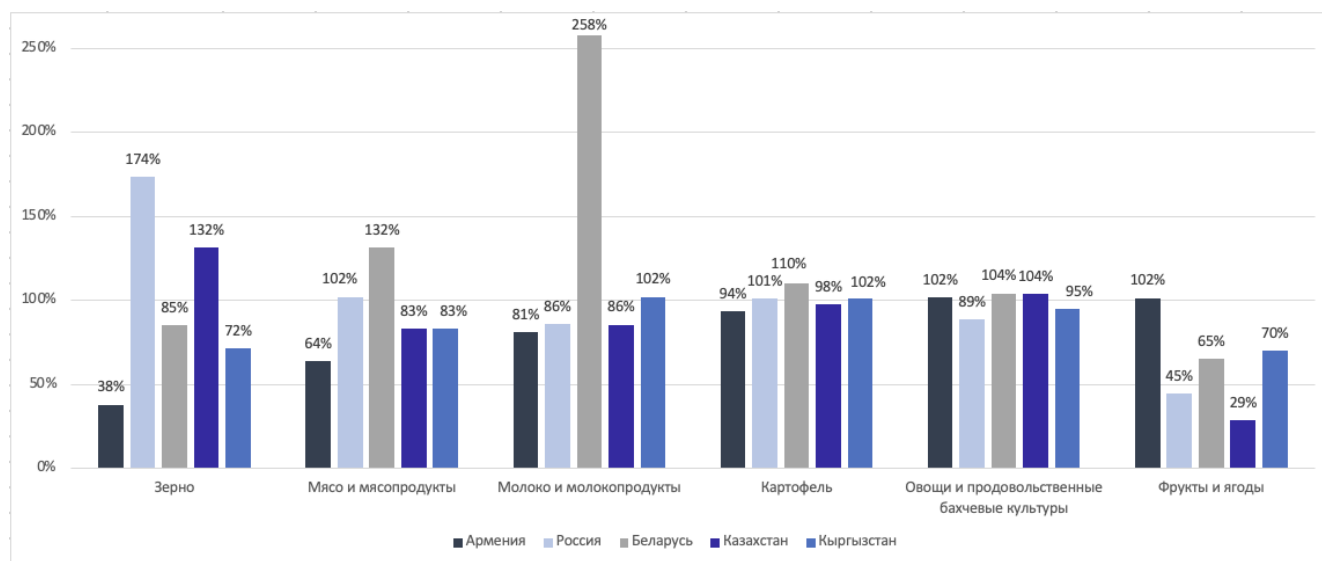
Молоко и молокопродукты	83,9	84,0	84,30	85,7	86,0	Не менее 90%
Картофель	95,1	89,2	88,7	94,5	101,0	Не менее 95%
Овощи и продовольственные бахчевые культуры	87,7	86,3	86,5	88,5	89,1	Не менее 90%
Фрукты и ягоды	40,2	42,4	44,4	47,3	44,6	Не менее 60%

Приведенные данные позволяют сделать вывод о том, что продовольственная независимость устойчиво наблюдается по зерну, мясу и мясопродуктам. Однако, по молоку и молокопродуктам, овощам и продовольственным бахчевым культурам, фруктам и ягодам показатели самообеспечения за период с 2019 по 2023 гг. не достигают пороговых значений уровня самообеспечения по основным видам продовольствия. Значение показателей по картофелю за последние 5 лет превышают пороговое лишь в 2019 и 2023 гг.

Проанализировав уровень самообеспечения основных продуктов питания по государствам – членам Евразийского экономического союза (ЕАС) за 2023 г. (рисунок 2), можно сделать следующие выводы:

- Россия и Казахстан являются наиболее устойчивыми государствами – членами ЕАС относительно уровня самообеспечения по зерну (174% и 132% соответственно);
- самый высокий уровень самообеспечения по мясу и мясопродуктам наблюдается в Беларуси (132%), в Армении, напротив, имеет место самый низкий уровень – 64%;
- наиболее устойчивая продовольственная независимость по молоку и молокопродуктам наблюдается в Беларуси (258%), наиболее неустойчивая – в Армении (81%);
- уровень самообеспечения по картофелю в странах-членах ЕАС находится в пределах от 94% (Армения) до 110% (Беларусь);

- все государства – члены ЕАС имеют близкое к 100% значение уровня самообеспечения по овощам и продовольственным бахчевым культурам. Отклонения составляют от 11 пунктов в России (89%) до 4 пунктов в Беларуси и Казахстане (104%);
- Россия и Казахстан не достигли даже 50% уровня самообеспечения по фруктам и ягодам. Эту планку преодолели Армения (102%), Кыргызстан (70%) и Беларусь (65%).

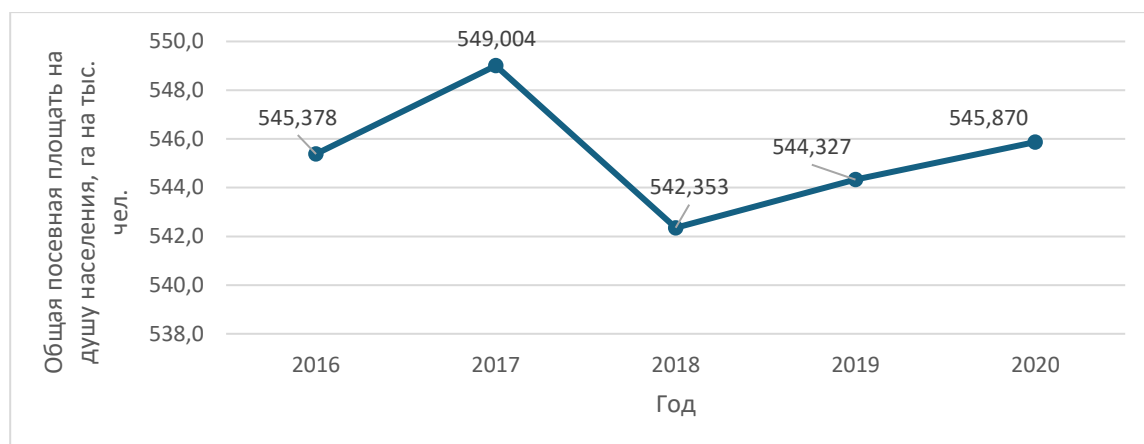


**Рисунок 2.2 – Уровень самообеспечения основными продуктами питания, 2023 г.**

В Канаде, Индии и Франции, например, разрабатывается показатель «Общая численность населения, страдающего от умеренной или острой нехватки продовольствия». В Грузии – «Число недоедающих в стране». Тем не менее, приведенные абсолютные показатели не имеют глубокой статистической значимости, так как, взглянув на представленные данные, невозможно сразу определить, насколько критична нехватка продовольственных товаров.

Еще одним показателем для контроля исполнения цели устойчивого развития «Ликвидация голода» может являться «Площадь сельскохозяйственных угодий на душу населения». Росстатом собираются абсолютные показатели «Посевная площадь сельскохозяйственных угодий» и «Численность населения». По относительному показателю можно оценить обеспеченность населения

площадями, необходимыми для возделывания сельскохозяйственных культур (Рисунок 2.3).



**Рисунок 2.3 – Расчет показателя по Российской Федерации**

Приведенные данные по посевным площадям на душу населения указывают на резкий скачок в 2017 г. (549,004), резкое снижение в 2018 г. (542,353) и постепенное увеличение показателя в 2018-2020 гг. Резкие скачки в 2017 и 2018 гг. отражаются также в расчете индекса производства продукции сельского хозяйства, который в 2017 г. равнялся 102,9%, а в 2018 г. – 99,8% (индекс производства продукции животноводства и растениеводства 103,3% и 98,5% соответственно). При этом изменения численности населения в рассматриваемые периоды были незначительными (сокращение на 11,8 тыс. чел. в 2018 г. по сравнению с 2017 г.), а посевная площадь сельскохозяйственных культур сократилась на 1,2% (снижение на 983 тыс. га в 2018 г. по сравнению с 2017 г.).

С учетом проведенного анализа системы показателей для Цели устойчивого развития №2 «Ликвидация голода» предложено добавить следующие признаки: уровень самообеспечения по отдельным видам сельскохозяйственной продукции; площадь сельскохозяйственных угодий на душу населения (Таблица 2.5).

**Таблица 2.5 – Предложения по системе показателей Цели устойчивого развития №2 «Ликвидация голода»**

Наименование цели устойчивого развития	Система показателей
Ликвидация голода	<ul style="list-style-type: none"> <li>• Уровень умеренного или острого отсутствия продовольственной безопасности населения, по шкале восприятия отсутствия продовольственной безопасности.</li> <li>• Доля домохозяйств, указавших при оценке своего материального положения на нехватку денег на еду.</li> <li>• Распространенность задержки роста среди детей в возрасте до пяти лет (среднеквадратичное отклонение от медианного показателя роста к возрасту ребенка в соответствии с нормами роста детей, установленными Всемирной организацией здравоохранения).</li> <li>• Индекс производства продукции сельского хозяйства в сопоставимых ценах к предыдущему году.</li> <li>• Доля животных отечественной репродукции, используемых для целей сельскохозяйственного производства на территории Российской Федерации.</li> <li>• <i>Уровень самообеспечения по отдельным видам сельскохозяйственной продукции.</i></li> <li>• <i>Площадь сельскохозяйственных угодий на душу населения.</i></li> </ul>

\*курсивом выделены предложения автора

### **2.3. Проектирование и разработка информационной системы на основе парсинга и визуализации данных**

Как уже было отмечено, показатели ЦУР рассчитываются и используются в первую очередь государственными органами управления. Поэтому разработанную информационную систему планируется внедрять, в частности, в Росстат.

Переходя к заинтересованным лицам, стоит отметить следующих:

1. Пользователи – население, заинтересованные федеральные органы управления, ФАО ООН.

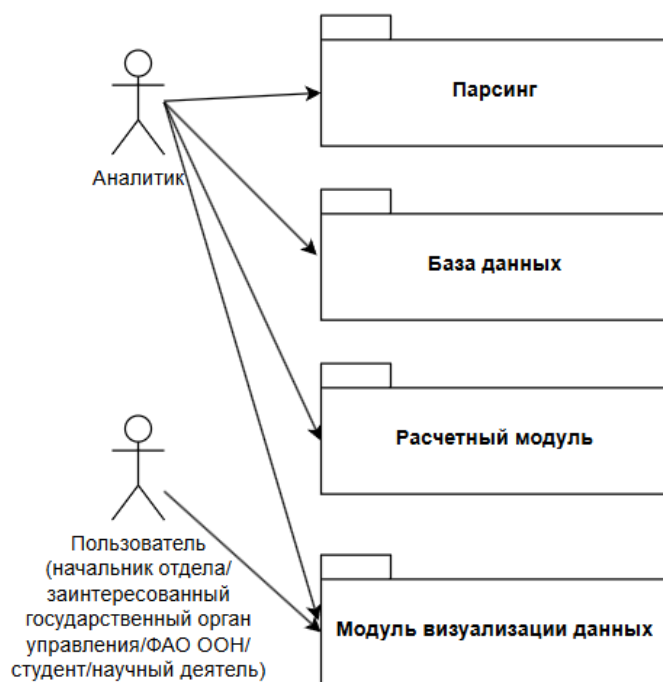
2. Заказчик – государственный орган управления в лице Росстата.
3. Дополнительно: руководители административных органов управления, министерства.

Структура потенциальной информационной системы представлена в таблице 2.6.

**Таблица 2.6 – Структура ИС**

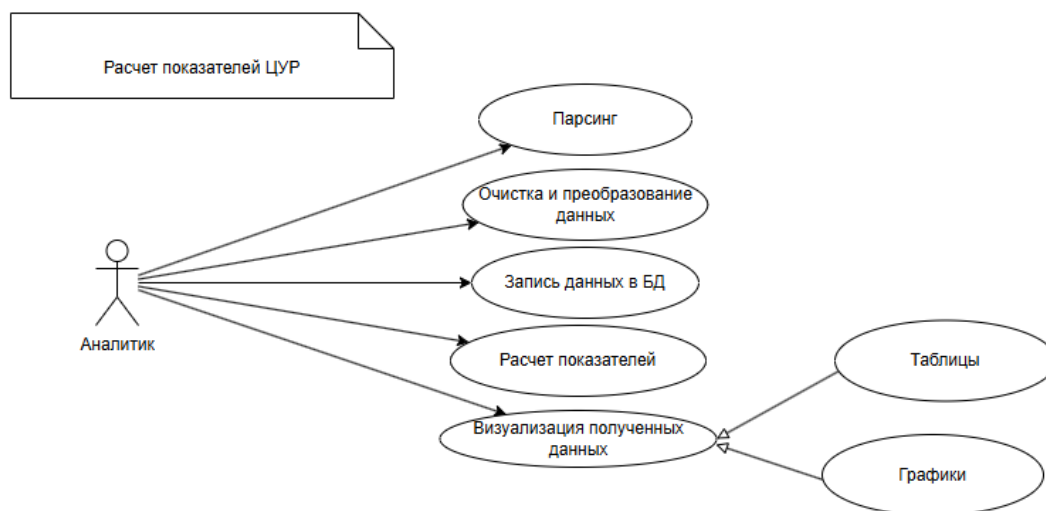
Модуль	Описание
Парсинг	Сбор необходимых данных с сайта Росстата.
База данных	Проверка валидности данных, их очистка, преобразование и запись в базу данных.
Расчетный модуль	Расчет показателей ЦУР по встроенным алгоритмам.
Модуль визуализации данных	Вывод данных в форме таблиц или графиков.

После определения заинтересованных лиц и приведенной ранее структуре информационной системы была построена диаграмма подсистем в модели вариантов использования (рисунок 2.4).



**Рисунок 2.4 – Подсистемы в модели вариантов использования**

Далее был выбран актант аналитик и построена диаграмма вариантов использования (Рисунок 2.5) для последующего определения спецификации.



**Рисунок 2.5 – Диаграмма вариантов использования**

Таким образом, в модуле расчета показателей ЦУР будут реализованы переход из представленного модуля в модуль парсинга, очистка и преобразование данных, запись информации в базу данных, расчет показателей и переход в модуль визуализации полученных данных в виде таблиц или графики.

При работе с данными и расчете показателей ЦУР вариант использования описывает шаги, которые выполняет аналитик для запуска модуля парсинга



данных с сайта Росстата, их последующей очистки и. преобразования, записи в базу данных, расчета показателей ЦУР и запуска модуля визуализации результатов. В качестве актанта выступает аналитик.

Предварительные условия: аналитик аутентифицировался в системе, сайт Росстата доступен, необходимые данные доступны для скачивания.

Постусловия: данные успешно собраны, очищены, преобразованы и записаны в БД. Показатели ЦУР рассчитаны и представлены в виде графиков, диаграмм или таблиц. Аналитик получил уведомление об успешном завершении всех операций.

Основной поток событий при запуске модуля запуска парсинга данных:

1. Аналитик нажимает кнопку «Запустить парсинг».
2. Система запрашивает у аналитика параметры парсинга (регионы, годы, показатели и т.д.).
3. Аналитик вводит необходимые параметры и подтверждает запуск.
4. Система запускает процесс парсинга данных с сайта.
5. По завершении парсинга система уведомляет аналитика об успехе операции.

Основной поток при очистке и преобразовании данных:

1. После завершения парсинга аналитик нажимает кнопку «Очистить и преобразовать данные».
2. Система предлагает аналитику выбрать правила очистки и преобразования данных.
3. Аналитик выбирает нужные правила и подтверждает операцию.
4. Система очищает и преобразует данные согласно выбранным правилам.
5. По завершении парсинга система уведомляет аналитика об успехе операции.

Основной поток при записи данных в БД:

1. После успешной очистки и преобразования данных аналитик нажимает кнопку «Записать данные в БД».

2. Система проверяет наличие необходимых прав у аналитика для записи данных.
3. Если права имеются, система записывает данные в БД.
4. По завершении парсинга система уведомляет аналитика об успехе операции.

Основной поток при расчете показателей ЦУР:

1. После записи данных в БД аналитик нажимает кнопку «Рассчитать показатели ЦУР».
2. Система производит расчет показателей ЦУР на основе введенных данных.
3. По завершении парсинга система уведомляет аналитика об успехе операции.

Основной поток при визуализации полученных данных:

1. После успешного расчета показателей ЦУР аналитик нажимает кнопку «Визуализировать».
2. Система предлагает аналитику выбрать типы графиков и диаграмм или вид выходной таблицы.
3. Аналитик выбирает нужные типы визуализации и подтверждает операцию.
4. Система создает графики и диаграммы или таблицы на основе рассчитанных данных.
5. По завершении парсинга система уведомляет аналитика об успехе операции.

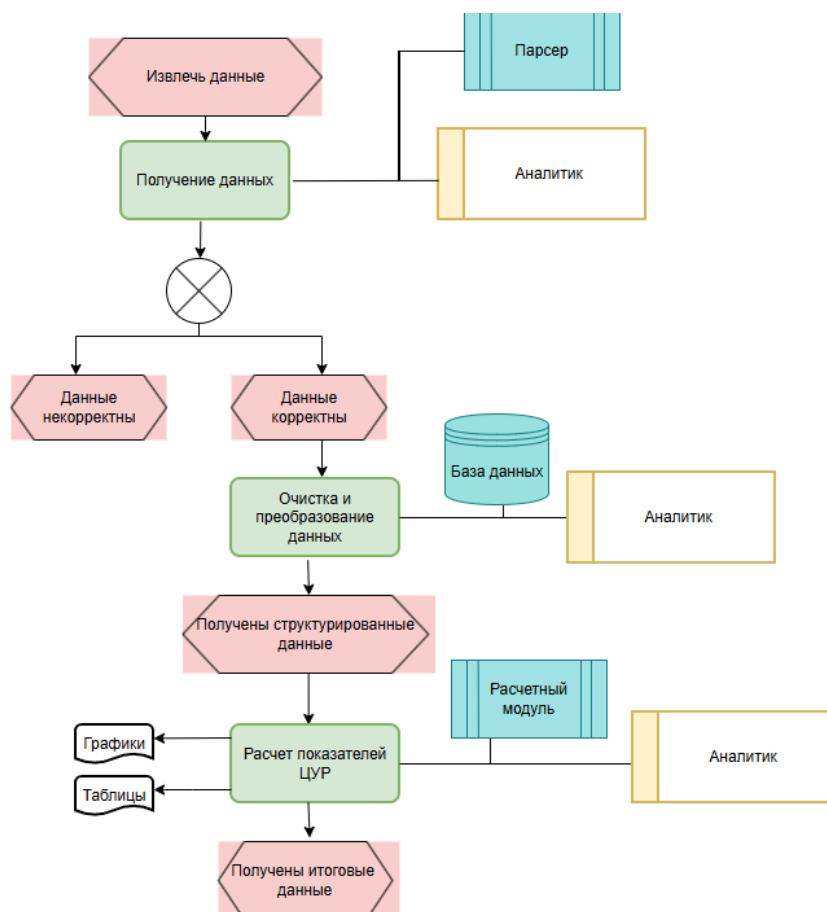
Альтернативные потоки:

1. Если сайт недоступен или произошла ошибка при парсинге, система уведомляет аналитика о проблеме и предлагает повторить попытку позже.
2. Если аналитик выбрал неверные правила очистки и преобразования, система сообщает об ошибке и предлагает выбрать правильные примеры.

3. Если у аналитика нет достаточных прав для записи данных в БД, система уведомляет его об этом и предлагает обратиться. К администратору системы.
4. Если при расчете показателей возникает ошибка (например, нехватка данных), система уведомляет аналитика и предлагает проверить данные или изменить формулы.

Если при визуализации возникают ошибки (например, отсутствие данных для выбранного типа графика), система уведомляет аналитика и предлагает выбрать другой тип визуализации.

Для построения цепочки процесса, управляемой событиями, была выбрана модель eEPC (рисунок 2.6).



**Рисунок 2.6 – Диаграмма eEPC**

Процесс, показанный на диаграмме, строится следующим образом:

- наступает событие «Извлечь данные»;

- функция «Получение данных» выполняется аналитиком с помощью парсера;
- если данные корректны, то процесс продолжается;
- аналитиком выполняется функция «Очистка и преобразование данных» с записью информации в базу данных;
- наступает событие «Получены структурированные данные»;
- аналитик выполняет функцию «Расчет показателей ЦУР» с помощью расчетного модуля и получает документы «Графики», «Таблицы»;
- завершающим событием является «Получены итоговые данные».

В качестве нотации визуализации бизнес-процессов было принято решение использовать IDEF, в частности модель IDEF3 (приложение А, рисунок 2.4):

1. Сначала выполняется сбор информации с сайта Росстата.
2. Далее данные обрабатываются и проверяются.
  - 2.1. Если данные достоверны, то происходит очистка данных (замена точек на запятые, пустые значения заменяются на нули и прочее).
  - 2.2. Если данные не достоверны, то запускается обработка исключения недопустимых значений.
3. На следующем этапе данные записываются в базу данных.
4. Из базы данных можно произвести расчет показателей ЦУР по встроенным алгоритмам.
5. После расчета итоговые значения показателей можно визуализировать в виде:
  - 5.1. таблиц;
  - 5.2. графиков.

Для визуализации логики работы проектируемой информационной системы была создана диаграмма IDEF0 (приложение А, рисунок 2.5):

1. Для функционального блока «Сбор данных с сайта Росстата» в качестве входных данных определены URL-адреса.
2. Для функционального блока «Преобразование и очистка данных» в качестве входных данных определены извлеченные на прошлом этапе данные, в

качестве управляющего механизма выступают средства python и SQL, на выходе создаётся база данных.

3. Для функционального блока «Расчет показателей ЦУР» в качестве входных данных определены структурированные данные из БД, в качестве управляющего механизма выступают средства python и SQL, на выходе – рассчитанные показатели ЦУР.

Для функционального блока «Визуализации и отчётность» в качестве входных данных определены рассчитанные на прошлом этапе показатели ЦУР, в качестве управляющего механизма выступают средства python, на выходе пользователь получает графики, отчеты.

Разрабатывая модуль парсига, было принято решение хранить глобальные настройки в качестве констант (рисунок 2.7).

```
BASE_DIR = os.getcwd() + r'\Dasha_vkr\SDG' # Корневая директория проекта
URLS_FILE = BASE_DIR + r'\urls.txt' # Путь к файлу с URL-ами для парсинга
DOWNLOAD_DIR = BASE_DIR + r'\downloads' # Основная папка для сохранения файлов

USER_AGENT = ("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) "
              "Chrome/134.0.0.0 YaBrowser/25.4.0.0 Safari/537.36")
COOKIE = ("redirect_cookie=1; federalnaya_sluzhba_gosudarstvennoy_statistiki_session=IqiSKptdfVjSv44NXH9vg5Zw1rQz001Snrv134e; "
          "sp_test=1; _ym_uid=1749279528691242495; _ym_d=1749279528; _ym_isad=2; _ym_visorc=w; sputnik_session=174927952804512")

CURRENT_INDICATORS = ['Статистическое приложение к ежегоднику, 2024',
                       'Уровень самообеспечения основными продуктами питания по Российской Федерации и субъектам Российской Федерации',
                       'Посевные площади сельскохозяйственных культур по Российской Федерации (по категориям хозяйств)',
                       'Численность постоянного населения в среднем за год']
MODES = ['🌐 Парсинг показателей', '📄 Просмотр данных', '📊 Расчет и Визуализация']

os.makedirs(DOWNLOAD_DIR, exist_ok=True)
```

**Рисунок 2.7 – Добавление констант**

Были созданы переменные для хранения пути к файлу с URL ссылками, к папке для сохранения файлов, User-Agent браузера для имитации реального пользователя и Cookie для аутентификации.

Для реализации модуля парсига были выбраны следующие библиотеки:

1. requests, предназначенная для работы с HTTP-запросами;
2. os – для взаимодействия с операционной системой, а именно с файлами и папками;
3. модуль BeautifulSoup библиотеки bs4, необходимый для парсинга HTML страниц;

4. модуль `urllib.parse` библиотеки `urljoin` – для корректного объединения URL ссылок;
5. `re`, предназначенная для очистки названий файлов;
6. модули `USER_AGENT`, `COOKIE`, `DOWNLOAD_DIR` библиотеки `config` для импорта настроек браузера.

Далее была создана функция для корректного сохранения файла, а именно для замены недопустимых символов в имени файла (рисунок 2.8).

```
def clean_filename(name): 2 usages
    return re.sub( pattern: r'[\<>:"/\|?*\\x00-\\x1F]', repl: "_", name.strip())
```

### Рисунок 2.8 – Функция для корректного сохранения файла

На рисунке 2.9 представлена функция для парсинга указанного URL с извлечением ссылки на файл, где на входе подаётся URL, а на выходе – список словарей с ключами 'title' и 'url'.

```
def get_files_from_url(url):
    headers = {
        "User-Agent": USER_AGENT,
        "Cookie": COOKIE
    }

    try:
        response = requests.get(url, headers=headers, verify=False, timeout=10)
        response.raise_for_status()

        soup = BeautifulSoup(response.text, "html.parser")
        file_items = soup.select(".document-list__item--row")

        files = []
        for item in file_items:
            link = item.select_one("a[href]")
            title = item.select_one(".document-list__item-title")

            if link and title:
                full_url = urljoin(url, link["href"])
                files.append({
                    "title": clean_filename(title.get_text(strip=True)),
                    "url": full_url
                })
        return files
    except Exception as e:
        print(f"❌ Ошибка загрузки страницы {url}: {e}")
        return []
```

### Рисунок 2.9 – Функция, извлекающая ссылки на файлы

В представленной функции в рамках тестирования отправляется GET-запрос с отключенной проверкой SSL и проверяется успешность выполнения этой операции. Далее проводится парсинг HTML-страницы и поиск блоков с файлами по CSS-селектору. После, в цикле, извлекается ссылка и заголовок файла, формируется, так называемый, абсолютный URL, файл добавляется в список и очищается название.

Далее была добавлена функция, которая парсит все URL из списка (Рисунок 2.10).

```
def parse_all_urls(urls):
    results = {}
    for url in urls:
        results[url] = get_files_from_url(url)
    return results
```

**Рисунок 2.10 – Функция для парсинга URL**

Для скачивания файла по указанному URL была создана ещё одна функция, отраженная на рисунке 2.11.

```
def download_file(file, folder):
    headers = {
        "User-Agent": USER_AGENT,
        "Cookie": COOKIE
    }

    try:
        response = requests.get(file["url"], headers=headers, stream=True, verify=False, timeout=30)
        response.raise_for_status()

        filename = f"{file['title']}.xlsx"
        path = os.path.join(folder, filename)

        with open(path, "wb") as f:
            for chunk in response.iter_content(chunk_size=1024):
                if chunk:
                    f.write(chunk)
            print(f"✅ Скачан: {path}")
        return True
    except Exception as e:
        print(f"❌ Ошибка загрузки файла {file['title']}: {e}")
        return False
```

**Рисунок 2.11 – Функция загрузки файла**

В этой функции проводится попытка запроса с потоковой загрузкой. Далее проверяется статус и формируется путь, имя файла. После файл записывается по указанному ранее пути на диск. При успешном выполнении функции выводится надпись «Скачан», в противном случае – «Ошибка загрузки файла».

На последнем этапе создания парсера были созданы функции сохранения и загрузки (Рисунок 2.12).

```
def save_results_to_json(data, filename):  
    import json  
    with open(filename, "w", encoding="utf-8") as f:  
        json.dump(data, f, ensure_ascii=False, indent=2)  
  
def load_results_from_json(filename):  
    import json  
    with open(filename, "r", encoding="utf-8") as f:  
        return json.load(f)
```

**Рисунок 2.12 – Функции сохранения и загрузки**

Первая функция предназначена для сохранения результатов парсинга в формате JSON, вторая – для загрузки результатов парсинга из этого формата.

Для реализации модуля преобразования и хранения данных были использованы следующие фреймворки:

1. sqlite3, как основная библиотека для создания, заполнения и вывода данных из таблиц БД;
2. pandas, необходимая для создания датафреймов;
3. re для создания регулярных выражений;
4. os для работы с файловой системой;
5. xlrd, openxl – для чтения файлов MS Excel форматов xls иxlsx;
6. json, предназначенная для создания, сохранения и открытия файлов json.

В первую очередь был создан класс, в котором содержатся все функции работы с БД. После создания переменной для соединения БД, списка для хранения отдельных показателей и загрузки файла с конфигом, была реализована функция создания таблиц «Показатели» и «Значения» (Рисунок 2.13).



```

def create_tables(self): 1 usage
    self.cur.execute("""
        CREATE TABLE IF NOT EXISTS Показатели (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            Наименование_показателя TEXT UNIQUE
        )
    """)
    self.cur.execute("""
        CREATE TABLE IF NOT EXISTS Значения (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            показатель_id INTEGER,
            год INTEGER,
            значение REAL,
            FOREIGN KEY (показатель_id) REFERENCES Показатели(id),
            UNIQUE (показатель_id, год)
        )
    """)
    self.conn.commit()
    self.close_connection()

```

**Рисунок 2.13 – Функция создания таблиц «Показатели» и «Значения»**

В таблице «Показатели» будут храниться данные об наименованиях показателей, а в таблице «Значения» - коды, года и значения показателей.

Далее была необходима функция, которая будет очищать значения ячеек скачанных Excel файлов от звёздочек и преобразовывать текст в числа – менять запятые на точки (Рисунок 2.14).

```

@staticmethod 1 usage
def clean_value(value):
    if isinstance(value, str):
        cleaned = value.strip().replace( __old: "*", __new: "" )
        try:
            if "." in cleaned:
                return float(cleaned.replace( __old: ",", __new: "." ))
            else:
                return int(cleaned)
        except ValueError:
            return cleaned
    return value

```

**Рисунок 2.14 – Функция преобразования данных**

Символы «\*» встречаются почти в каждом файле, скаченном с сайта Росстата. Ими обозначают сноски к показателям.

В ходе записи данных в БД была выявлена проблема с чтением показателя «Уровень умеренного или острого отсутствия продовольственной безопасности населения, по шкале восприятия отсутствия продовольственной безопасности».

Существует два уровня представления данных – «умеренный и острый уровень» и «острый (тяжелый) уровень». Оба уровня записываются в одной ячейке, следовательно, необходима функция для разделения ячейки на две для последующей записи данных в БД (Рисунок 2.15).

```
def process_complex_row(row):
    first_col = row.iloc[0]
    if isinstance(first_col, str) and "умеренного или острого" in first_col.lower():
        values = []
        for col in row.index[1:]:
            cell_value = row[col]
            if isinstance(cell_value, str):
                parts = [v.strip() for v in cell_value.split(";") if v.strip()]
                if len(parts) >= 2:
                    if isinstance(parts[0], str):
                        parts = [p.split()[0].replace(__old: ',', __new: '.') for p in parts]
                    values.append(parts)
                else:
                    values.append([cell_value.strip()] * 2)
            else:
                values.append([cell_value] * 2)

        new_rows = []

        new_row1 = row.copy()
        new_row1.iloc[0] = "Уровень умеренного отсутствия продовольственной безопасности населения"
        for i, col in enumerate(row.index[1:]):
            new_row1[col] = values[i][0]
        new_rows.append(new_row1)

        new_row2 = row.copy()
        new_row2.iloc[0] = "Уровень острого отсутствия продовольственной безопасности населения"
        for i, col in enumerate(row.index[1:]):
            new_row2[col] = values[i][1]
        new_rows.append(new_row2)

    return pd.DataFrame(new_rows)
else:
    return pd.DataFrame([row])
```

**Рисунок 2.15 – Функция разделения ячеек для последующей записи данных в БД**

После приведения таблиц к третьей нормальной форме, была создана функция, объединяющая датафрейм с данными по показателям ЦУР (Рисунок 2.16).

```

def merge_dataframes(self):
    all_years = set()
    for df in self.single_DFS:
        for col in df.columns[1:]:
            if str(col).isdigit():
                all_years.add(int(col))
    sorted_years = sorted(all_years)

    cleaned_dfs = []
    for df in self.single_DFS:
        df = df.copy()
        year_columns = {}
        for col in df.columns[1:]:
            if str(col).isdigit():
                year_columns[col] = int(col)
        df.rename(columns=year_columns, inplace=True)
        cleaned_dfs.append(df)

    melted = []
    for df in cleaned_dfs:
        id_vars = ['Наименование_показателя']
        value_vars = [col for col in df.columns if isinstance(col, int)]
        melted_df = df.melt(id_vars=id_vars, value_vars=value_vars, var_name='Год', value_name='Значение')
        melted.append(melted_df)

    full_melted = pd.concat(melted, ignore_index=True)

    merged_wide = full_melted.pivot(
        index='Наименование_показателя',
        columns='Год',
        values='Значение'
    )

    merged_wide = merged_wide.reindex(sorted_years, axis=1)

    merged_wide.reset_index(inplace=True)

    self.temp_DF = merged_wide

```

**Рисунок 2.16 – Функция объединения данных**

В представленной функции в первую очередь создан цикл для получения полного списка уникальных лет из всех датафреймов, хранящих показатели ЦУР. Далее колонки переименовываются, датафреймы объединяются и сортируются года в необходимом порядке. Таким образом, все годы объединяются в одну таблицу и, если значение отсутствует, то ячейка определяется, как None.

Для получения файлов из указанной папки по заданной конфигурации была создана функция, представленная на рисунке 2.17.

```

def load_files_from_folder(self):

    for config in self.xl_config:
        filename = config["filename"]
        file_path = os.path.join(r'/Users/darya/PycharmProjects/Dasha_vkr/SDG/data', filename)

        if not os.path.isfile(file_path):
            print(f"Файл не найден: {file_path}")
            continue

        sheet_name = config["sheet_name"]
        use_cols = config.get("use_cols")
        indicator_name = config.get("indicator_name")
        header_row = config.get("header_row", 0)
        data_start_row = config.get("data_start_row", 1)
        selected_rows = config.get("selected_rows")

        self.read_xl(file_path=file_path,
                     sheet_name=sheet_name,
                     indicator_name=indicator_name,
                     use_cols=use_cols,
                     header_row=header_row,
                     data_start_row=data_start_row,
                     selected_rows=selected_rows)

```

### Рисунок 2.17 – Функция загрузки файлов из папки

После создания переменной, хранящей путь к файлам, необходимо получить данные из конфига.

Следующим шагом требуется прочитать данные из файла MS Excel с помощью функции, представленной на рисунке 2.18.

```

def read_xl(self, file_path: str,
            sheet_name: str,
            indicator_name: str,
            use_cols: list = None,
            header_row: int = 0,
            data_start_row: int = 1,
            selected_rows: list = None):

    df = pd.read_excel(file_path, sheet_name=sheet_name, header=header_row,
                      skiprows=data_start_row - 1 if data_start_row > header_row + 1 else 0,
                      usecols=use_cols)

    if selected_rows is not None:
        df = df.iloc[selected_rows]

    if indicator_name != '':
        df.iloc[0, 0] = indicator_name

    df.columns = [str(re.sub(r'[\.\*]| r', '_', col).strip("_")) if isinstance(col, str) else col for col in df.columns]
    df.columns = ['Наименование_показателя'] + list(df.columns)[1:]

    for col in df.columns:
        df[col] = df[col].apply(self.clean_value)

    complex_rows_dfs = []
    for _, row in df.iterrows():
        complex_rows_dfs.append(self.process_complex_row(row))
    df = pd.concat(complex_rows_dfs, ignore_index=True)

    self.single_DFS.append(df)

```

### Рисунок 2.18 – Функция чтения Excel файла

В представленной функции указывается:

- путь к файлу
- наименование листа, с которого необходимо взять данные;
- наименование показателя, под которым он запишется в БД;
- порядковые номера столбцов, из которых необходимо взять информацию;
- номера строк, из которых необходимо взять информацию.

Финальным этапом необходимо создать функции инициализации запросов к БД (Рисунок 2.19).

```
def get_indicators(self): 1 usage

    self.open_connection()
    self.cur.execute("""SELECT Наименование_показателя FROM Показатели""")
    indicators = [row[0] for row in self.cur.fetchall()]
    self.close_connection()
    return indicators

def get_ind_val(self, val): 3 usages

    self.open_connection()

    query = """SELECT год, значение FROM Значения
                WHERE показатель_id = (SELECT id FROM Показатели
                                     WHERE Наименование_показателя = ?)"""

    df = pd.read_sql_query(query, self.conn, params=(val, ))
    self.close_connection()
    return df
```

**Рисунок 2.19 – Функции инициализации запросов к БД**

Функция `get_indicators` предназначена для инициализации запроса к БД для извлечения всех наименований показателей, а функция `get_ind_val` для инициализации запроса к БД для отбора информации по конкретному наименованию показателя.

Для реализации интерфейса, модуля расчета и визуализации были выбраны следующие фреймворки:

1. `streamlit`, как основная библиотека для создания веб-интерфейса;
2. `plotpy` – для создания графиков;
3. `shutil`, которая хранит методы для удаления полной ветки директории со всеми вложенными файлами;

4. time, необходимая для реализации задержек между операциями;
5. pandas, os, json – для работы с датафреймами, файлами и результатами парсинга.

В первую очередь был добавлен модуль парсинга. Для начала необходимо прочитать ссылки из текстового документа (Рисунок 2.20).

```
try:
    with open(URLS_FILE, "r", encoding="utf-8") as f:
        urls = [line.strip() for line in f if line.strip()]
except FileNotFoundError:
    st.error("❌ Файл urls.txt не найден")
    st.stop()
```

**Рисунок 2.20 – Чтение ссылок**

Далее был создан заголовок страницы и добавлен контейнер, который разбивается на 3 части: отображение подзаголовка, кнопку для запуска парсинга и отображение сообщения об успешном выполнении парсинга (Рисунок 2.21).

```
st.header("🌐 Загрузка файлов с сайта rosstat.gov.ru", divider=True)

with st.container(border=True):
    col1, col2, col3 = st.columns((1, 1, 1))
    col1.subheader("🌐 Парсинг")

    if col2.button("🔍 Запуск парсера"):
        with st.spinner("🔄 Загрузка данных с сайта..."):
            results = parse_all_urls(urls)
            st.session_state.parsed_results = results
            save_results_to_json(results, BASE_DIR + "/all_parsed_results.json")
            col3.success(f"✅ Обработано {len(results)} URL")
```

**Рисунок 2.21 – Создание кнопок и вывод сообщений в модуле парсинга**

Также был добавлен прогресс-бар, обработка всех выбранных URL через цикл, выбор файлов для скачивания и сохранение этого выбора. После завершения этого этапа пользователю будет выводиться сообщение об успешном завершении парсинга.

Следующим шагом была реализована кнопка запуска скачивания данных в файловую систему (Рисунок 2.22).

```
if st.session_state.selected_files_by_url and st.button("📄 Скачать выбранные файлы"):
    with st.spinner("🔄 Идет загрузка файлов..."):
        total_files = sum(len(files) for files in st.session_state.selected_files_by_url.values())
        downloaded_count = 0
```

**Рисунок 2.22 – Кнопка запуска скачивания файлов**

Было принято решение добавить прогресс-бар, логику загрузки файлов и вывод сообщения об успешной загрузке. При этом был предусмотрен вывод оповещения пользователя, если такой файл ранее скачивался.

После создаётся контейнер для отображения скачанных файлов в папке, хранящей загруженные файлы (Рисунок 2.23).

```
placeholder = st.empty()
with placeholder.container(border=True):
    if os.path.exists(DOWNLOAD_DIR) and len(os.listdir(DOWNLOAD_DIR)) > 0:
        st.subheader("📁 Загруженные файлы")
        for root, dirs, files in os.walk(DOWNLOAD_DIR):
            for file in files:
                st.write(f"- {file}")
```

**Рисунок 2.23 – Контейнер для отображения скачанных файлов**

Последним этапом была реализована кнопка для записи скачанной информации в БД и удаления уже ненужных файлов (Рисунок 2.24).

```
if st.button('Запись в БД'):
    placeholder.empty()
    placeholder.info('Начинаю обрабатывать скачанные файлы')
    st.session_state.my_db.load_files_from_folder()
    time.sleep(1)
    placeholder.success('Успешно')
    time.sleep(1)
    placeholder.info('Начинаю обработку данных')
    st.session_state.my_db.merge_dataframes()
    time.sleep(1)
    placeholder.success('Успешно')
    time.sleep(1)
    placeholder.info('Нормализуем структуру')
    st.session_state.my_db.update_normalized_table()
    time.sleep(1)
    placeholder.success('Успешно')
    time.sleep(1)
    placeholder.info('Удаляю лишнее')
    shutil.rmtree(DOWNLOAD_DIR)
    os.remove(BASE_DIR + "/all_parsed_results.json")
    placeholder.success("Процесс загрузки прошел успешно")
```

**Рисунок 2.24 – Кнопка записи данных в БД**

После вывода сообщения о начале обработки данных запускается инициализация метода БД для загрузки файлов. Далее – инициализация метода БД для обработки загруженных файлов, нормализации информации и записи данных в таблицы. По окончании этого этапа выводится сообщение с успешным завершением и удаляются директории с скачанными файлами, временным файлом json с ссылками на скачивание, наименованиями.

Далее была создана функция для работы с модулем расчета и визуализации. Аналогичным образом был отображен заголовок и инициализирован контейнер, в который выводятся абсолютные показатели для дальнейшего расчёта (Рисунок 2.25).

```
st.header('Расчет и визуализация показателей', divider=True)

with st.container(border=True):
    col1, col2 = st.columns(2, border=True)

    with col1:
        centered_text('Посевные площади сельскохозяйственных культур по РФ, тыс. га', tag='h3')
        divided_df = st.session_state.my_db.get_ind_val('Посевные площади сельскохозяйственных культур по РФ')
        col1.dataframe(divided_df)

    with col2:
        centered_text('Численность постоянного поселения РФ, чел.', tag='h3')
        divider_df = st.session_state.my_db.get_ind_val('Численность постоянного поселения РФ')
        col2.dataframe(divider_df)
```

**Рисунок 2.25 – Создание блоков для расчета показателя**

В первой колонке отображается информация по показателю «Посевные площади сельскохозяйственных культур», во второй – «Численность постоянного населения».

Следующим этапом предстоит расчёт относительного показателя «Площадь сельскохозяйственных угодий на душу населения» и визуализация результатов (Рисунок 2.26).

```
if st.button('Произвести расчет показателя "Площадь сельскохозяйственных угодий на душу населения, га на тыс.чел."'):
    with (st.container()):
        col1, col2 = st.columns(2)

        merged_df = pd.merge(divided_df, divider_df, on='год', suffixes=('_divided', '_divider'))

        merged_df['значение'] = merged_df['значение_divided']*1000 / (merged_df['значение_divider']/1000)

        result_df = merged_df[['год', 'значение']]

        col1.dataframe(result_df)

        result_df['значение'] = result_df['значение'].apply(lambda x: f'{x:.2f}')

        fig = plotly.plot(result_df, 'line', x='год', y='значение', markers=True)
        fig.update_traces(text=result_df['значение'], textposition="top center")

        fig.update_layout(
            hovermode='x unified',
            xaxis_title='Год',
            yaxis_title='Значение',
            showlegend=False
        )
        col2.plotly_chart(fig)
```

**Рисунок 2.26 – Кнопка расчета показателя «Площадь сельскохозяйственных угодий на душу населения»**



В представленном фрагменте объединяются датафреймы, содержащие абсолютные показатели, по годам, вычисляется относительный показатель. С целью читаемости показателя было принято решение перевести исходные данные, чтобы в результате получить показатель в га на тыс. чел. Далее был создан финальный датафрейм, отображены таблица с рассчитанным показателем и график

Последним этапом реализации интерфейса была создана основная функция программы, в которой прописан переход между всеми страницами приложения (Рисунок 2.27).

```
def main():

    st.set_page_config(page_title="Мониторинг ЦУР", layout="wide")
    st.sidebar.header('Режимы работы')

    mode = st.sidebar.selectbox('Выберите из списка',
                                MODES,
                                index=None,
                                placeholder='Раскройте список')

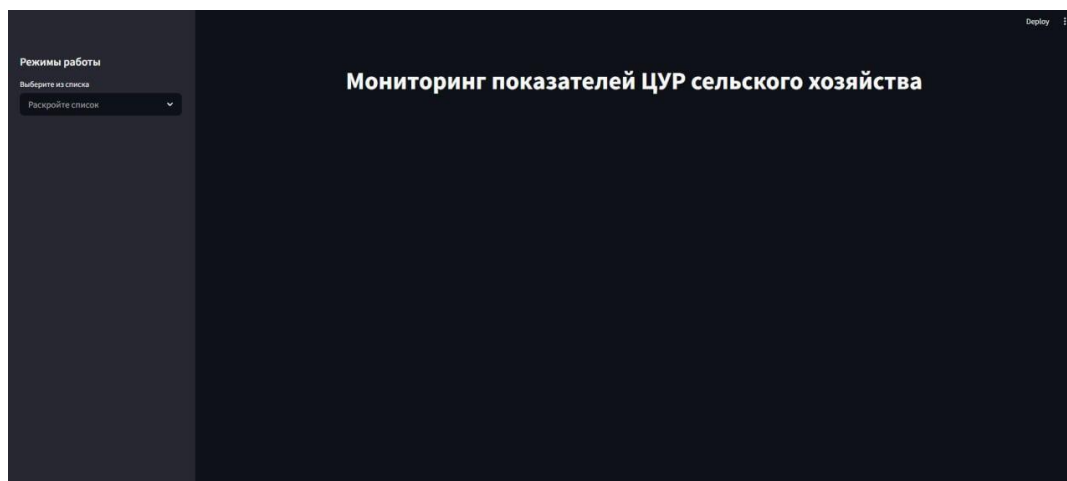
    centered_text("Мониторинг показателей ЦУР сельского хозяйства")

    if mode == '🌐 Парсинг показателей':
        parsing_page()
    elif mode == '📊 Просмотр данных':
        showing_page()
    elif mode == '📈 Расчет и Визуализация':
        calc_page()
```

**Рисунок 2.27 – Функция main**

В представленной функции настраивается отображение вкладки браузера, создаются заголовок боковой панели и главный заголовок для всех страниц. Также инициализируется всплывающий список в боковой панели для реализации выбора страницы.

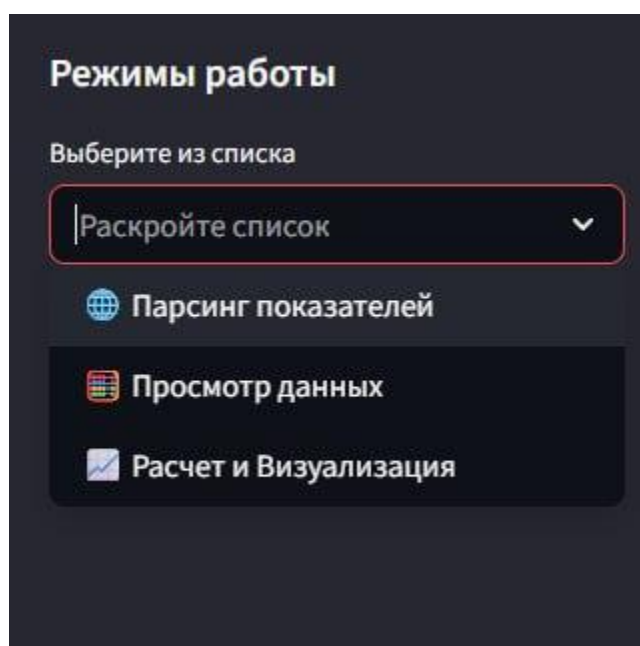
При запуске программы открывается веб-страница с боковой панелью приложения (Рисунок 2.28).



**Рисунок 2.28 – Стартовая страница программы**

На боковой панели расположен выпадающий список (Рисунок 2.29) с выбором модуля (режима работы). Есть возможность выбрать один из следующих модулей:

1. Парсинг показателей;
2. Просмотр данных;
3. Расчёт и Визуализация.



**Рисунок 2.29 – Выбор модуля**

Первый режим позволяет запустить парсинг показателей с официального сайта Федеральной службы государственной статистики.

Второй режим позволяет просмотреть информацию по каждому показателю из базы данных, загруженных в первом режиме.

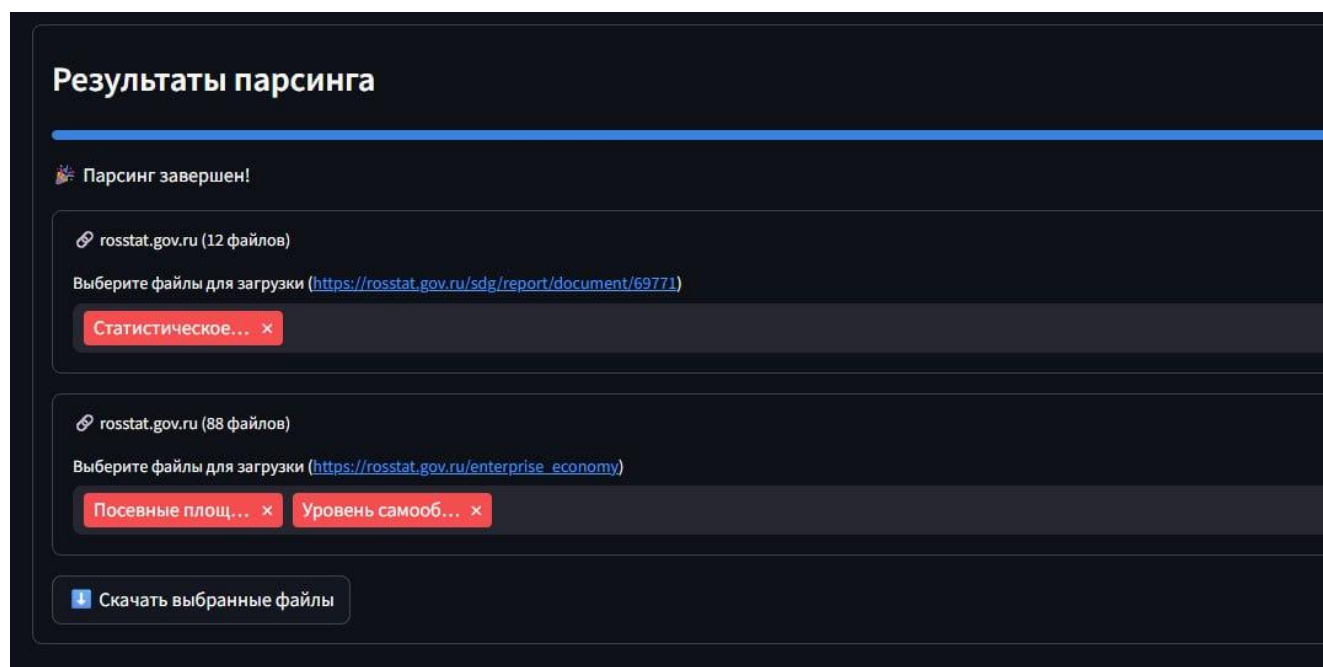
Третий режим позволяет рассчитать и визуализировать предлагаемый для внесения в систему показателей ЦУР индикатор «Площадь сельскохозяйственных угодий на душу населения».

Выбрав первый режим работы программы, пользователь нажимает кнопку «Запуск парсера», а интерфейс, в свою очередь, отображает результаты загрузки данных и сообщение о количестве обработанных ссылок (Рисунок 2.30).



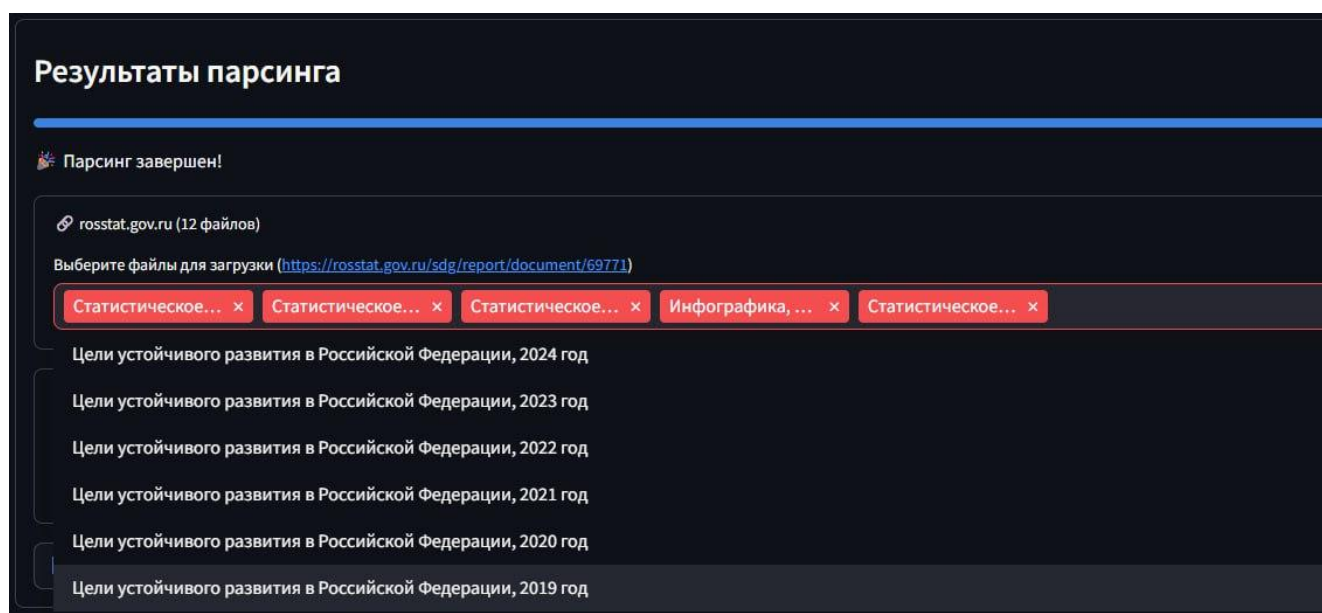
**Рисунок 2.30 – Запуск парсера**

По завершении парсинга в графическом интерфейсе строятся столько новых отдельных блоков, сколько ссылок было обработано на прошлом этапе (Рисунок 2.31).



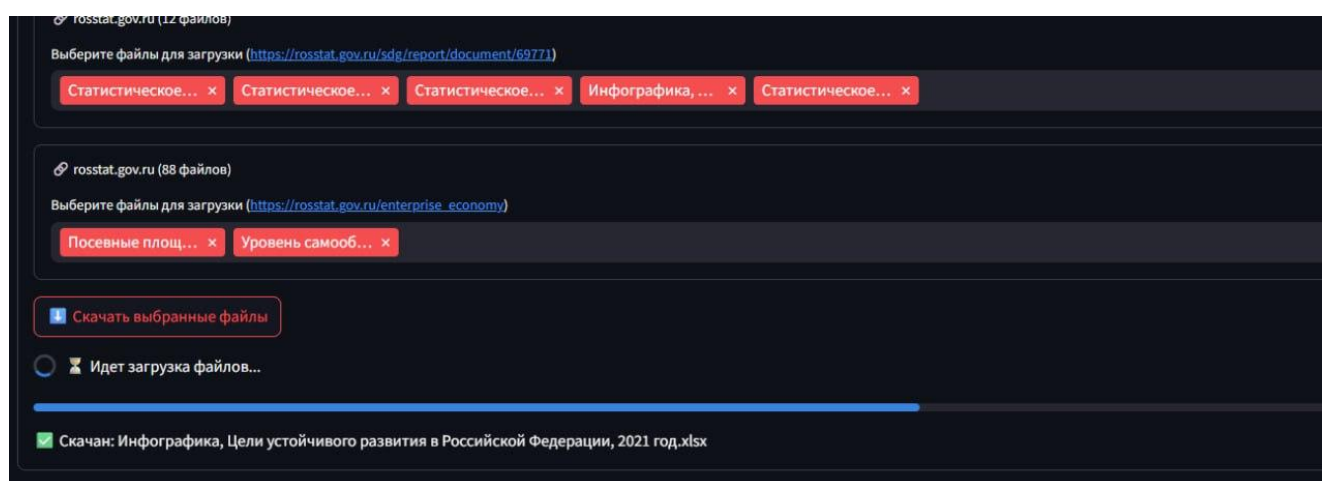
**Рисунок 2.31 – Результат обработки ссылок для парсинга**

Каждый блок предоставляет возможность пользователю выбрать лишь интересующие его файлы с показателями. Для этого достаточно раскрыть список, как показано на рисунке 2.32.



**Рисунок 2.32 – Раскрывающийся список показателей**

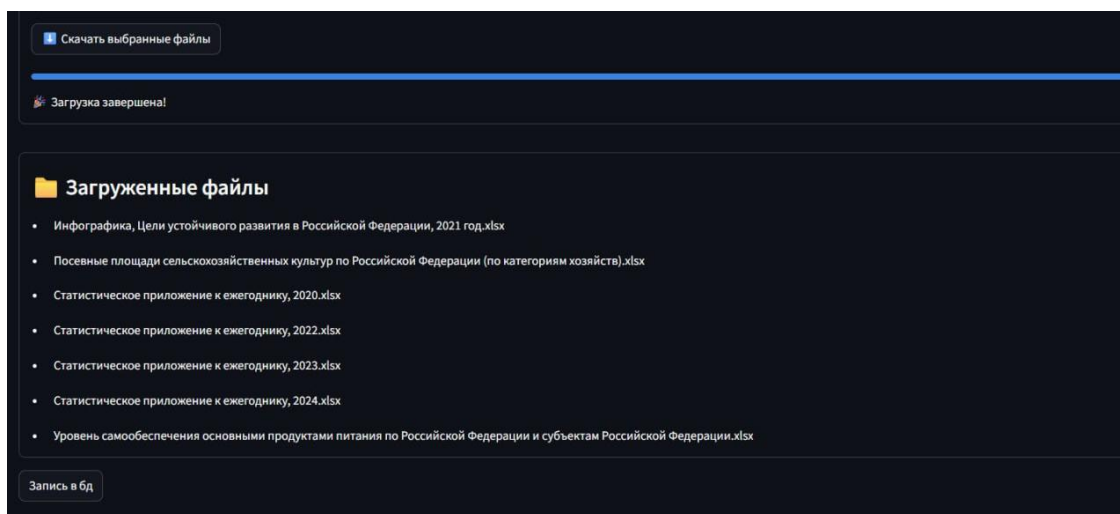
Далее пользователь имеет возможность скачать файлы, нажав на кнопку «Скачать выбранные файлы». В интерфейсе отображается процесс загрузки каждого из выбранных файлов с выводом информационного сообщения о завершении скачивания каждого отдельного файла (Рисунок 2.33).



**Рисунок 2.33 – Процесс скачивания файлов**

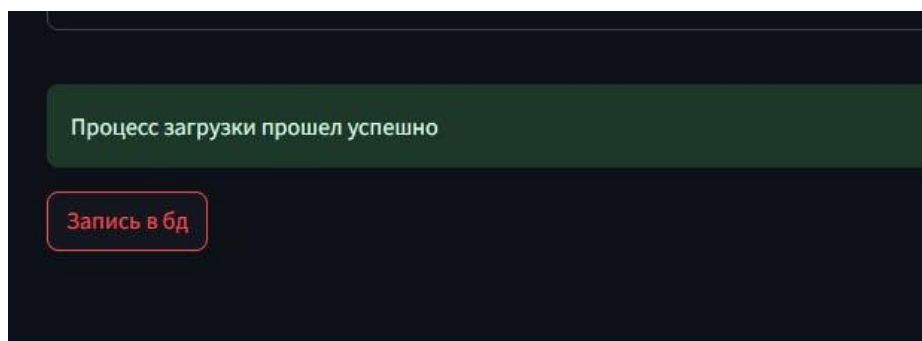
Если любой из файлов не был скачан, то в графическом интерфейсе отображается соответствующее оповещение. Также программой учитывается ситуация, в которой при неоднократном нажатии на кнопку скачивания, дублирующие файлы пропускаются.

После успешного скачивания отображается содержимое загрузочного каталога для уверенности пользователя в загрузке всей необходимой информации (Рисунок 2.34).



**Рисунок 2.34 – Каталог загруженных файлов**

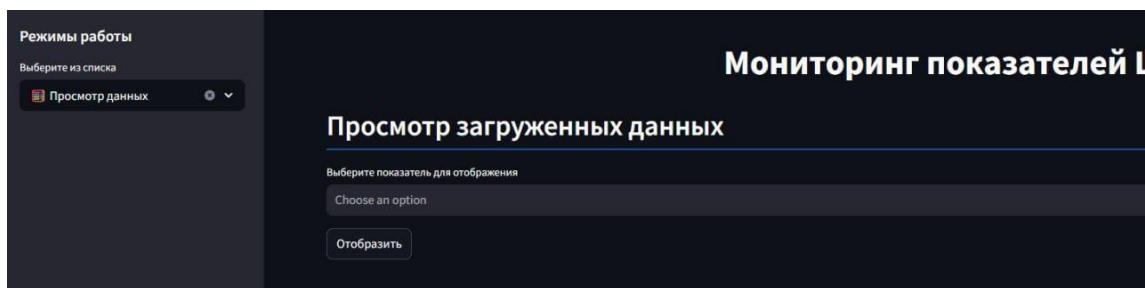
По нажатии на кнопку «Запись в бд» извлекается и преобразуется информация из скачанных файлов с последующей записью в базу данных (Рисунок 2.35).



**Рисунок 2.35 – Запись в базу данных**

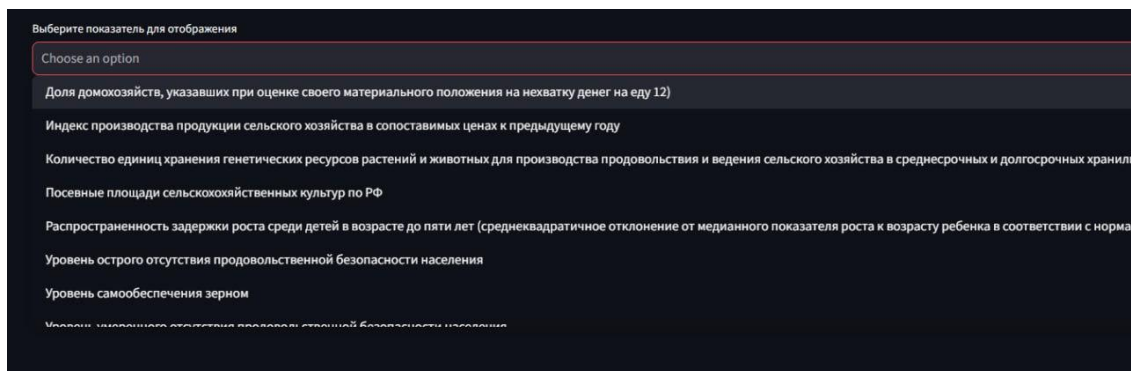
По окончании записи директория с обработанными файлами удаляется, а на экран выводится сообщение об успешном завершении процесса загрузки данных.

Выбрав второй режим работы программы, пользователю необходимо выбрать индикатор из выпадающего списка загруженных показателей. Далее пользователь нажимает кнопку «Отобразить» (Рисунок 2.36).



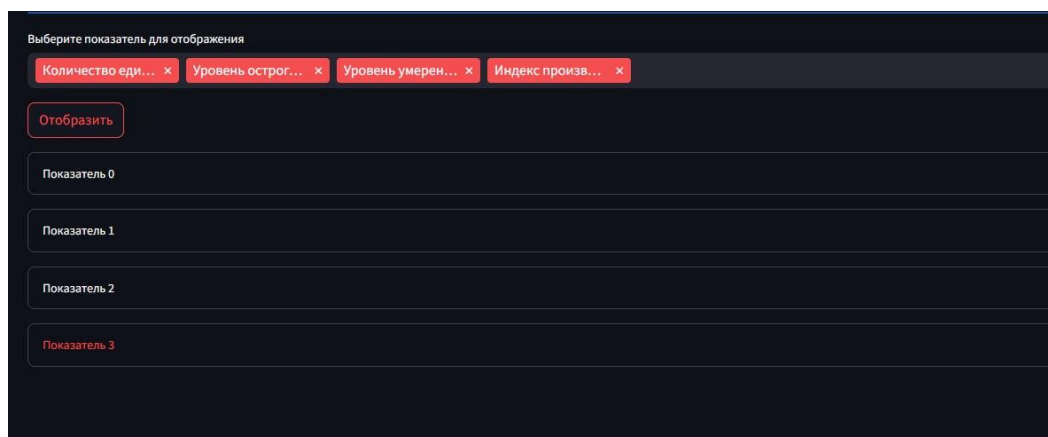
**Рисунок 2.36 – Просмотр загруженных данных**

Выпадающий список показателей представлен на рисунке 2.37.



**Рисунок 2.37 – Выпадающий список показателей для отображения**

При этом, есть возможность выбрать и отобразить несколько показателей (Рисунок 2.38).



**Рисунок 2.38 – Выбор нескольких показателей для отображения**

Таким образом, выбрав необходимые показатели и нажав на кнопку «Отобразить», пользователь получит таблицы, включающие в себя наименование и значение показателя за определенные заранее года (Рисунок 2.39).

Показатель 2

Уровень умеренного отсутствия продовольственной безопасности населения

	год	значение
0	2020	5.7
1	2021	4.6
2	2022	4.1
3	2023	2.1

Показатель 3

Индекс производства продукции сельского хозяйства в сопоставимых ценах к предыдущему году

	год	значение
0	2019	104.3
1	2020	101.3
2	2021	99.3
3	2022	111.3
4	2023	100.2

**Рисунок 2.39 – Отображение нескольких показателей**

Выбрав третий режим работы программы, пользователю предоставляется возможность рассчитать актуальные данные по обновленной системе показателей ЦУР №2 «Ликвидация голода», а именно показатель «Посевные площади сельскохозяйственных культур на душу населения» (Рисунок 2.40).

Режимы работы

Выберите из списка

Расчет и визуализация

# Мониторинг показателей ЦУР сельского хозяйства

## Расчет и визуализация показателей

### Посевные площади сельскохозяйственных культур по РФ, тыс. га

год	значение	
0	2017	80046.696
1	2018	79633.739
2	2019	78869.336
3	2020	75947.997
4	2021	80426.702

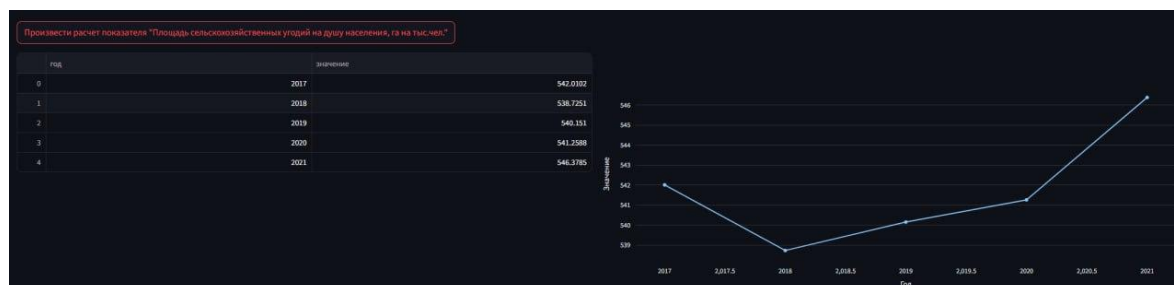
### Численность постоянного населения РФ, чел.

год	значение	
0	2017	147688545
1	2018	147818888
2	2019	147899994
3	2020	147707517
4	2021	147317903
5	2022	146713743
6	2023	146299106

Произвести расчет показателя "Площадь сельскохозяйственных угодий на душу населения, га на тыс.чел"

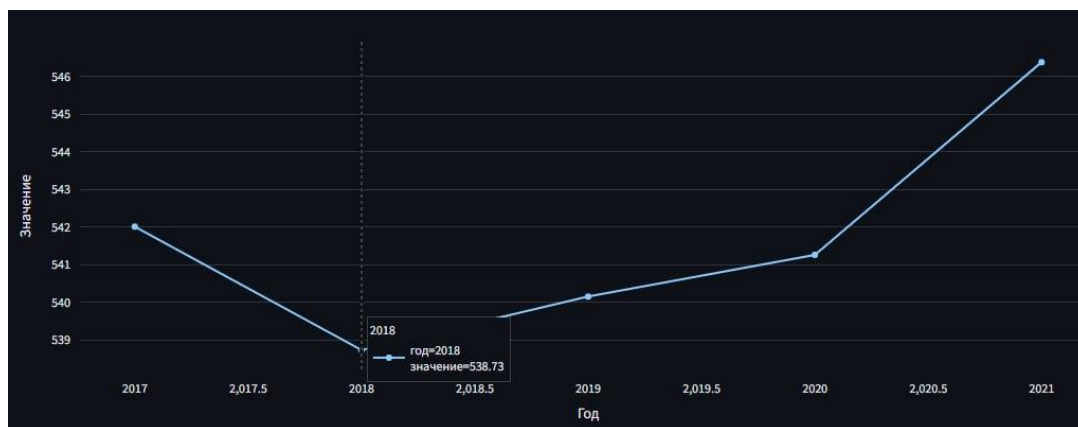
**Рисунок 2.40 – Расчет показателя «Посевные площади сельскохозяйственных культур на душу населения»**

После нажатия на кнопку «Произвести расчет показателя» интерфейс предоставит актуальные расчетные данные и визуализирует их (Рисунок 2.41).



**Рисунок 2.41 – Отображение расчетного показателя**

Также у пользователя есть возможность навести курсором на конкретный маркер, отражающий значение показателя (Рисунок 2.42).



**Рисунок 2.42 – Визуализация расчетного показателя**

В этом случае интерфейс выведет окно с подробной информацией за этот период в формате [год, значение].

В процессе планирования разработки созданы функциональные и инфологические модели проектирования, подробно иллюстрирующие работу и возможности информационной системы.

Система состоит из трёх связанных между собой модулей. Первый модуль позволяет запустить парсинг показателей с официального сайта Федеральной службы государственной статистики. Второй модуль позволяет просмотреть информацию по каждому показателю из базы данных, загруженных в первом модуле. Третий модуль позволяет рассчитать и визуализировать предлагаемый для внесения в систему показателей ЦУР индикатор «Площадь сельскохозяйственных угодий на душу населения».

## **Библиографический список к главе 2**

1. Алиев, Х. Р. Модели оценки стоимости информационных систем в методологиях разработки программного обеспечения / Х.Р. Алиев. - М.: Синергия, 2020. - 569 с.

2. Анализ международной практики внедрения цифровизации в агропромышленный комплекс национальных и наднациональных экономик, на примере стран с традиционно развитым сельским хозяйством : Аналитические материалы / М. Ю. Архипова, М. В. Кагирова, А. В. Уколова [и др.]. – Москва : Научный консультант, 2021. – 118 с.



3. Анциферова О. Ю., Петрова Л. М. Уровень самообеспечения основными продуктами питания-ключевой критерий продовольственной безопасности страны //Московский экономический журнал. – 2021. – №. 11. – С. 714-721.
4. Балдин К., Уткин В. Информационные системы в экономике. – Litres, 2022.
5. Батищев, В. И. Методы адаптивного формирования информационных систем анализа состояния сложных технических объектов / В.И. Батищев. - М.: Синергия, 2018. - 159 с.
6. Белопухова В. В., Пастухова Л. В. ОСОБЕННОСТИ ВЫПОЛНЕНИЯ ДРК МЕЖДУНАРОДНО-ПРАВОВЫХ ОБЯЗАТЕЛЬСТВ В СФЕРЕ ЛИКВИДАЦИИ ГОЛОДА И НЕДОЕДАНИЯ // Океанский менеджмент. – 2025. – №. 1С (29). – С. 27-30.
7. Бендерская О. Б. Система показателей комплексной оценки устойчивости функционирования предприятия и ее апробация. – 2021.
8. Бикетова С. В. АНАЛИЗ СТРАТЕГИЙ РОССИЙСКИХ КОМПАНИЙ ПО ДОСТИЖЕНИЮ ЦЕЛЕЙ УСТОЙЧИВОГО РАЗВИТИЯ //ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ УСТОЙЧИВОГО РАЗВИТИЯ ПРОМЫШЛЕННОСТИ В XXI ВЕКЕ: ОТ ТЕОРИИ К ПРАКТИКЕ. – 2023. – С. 24-28.
9. Бобылёв С. Н. Новые модели экономики и индикаторы устойчивого развития // Экономическое возрождение России. – 2019. – №. 3 (61). – С. 23-29.
10. Буч Г., Якобсон И., Рамбо Д. Введение в UML от создателей языка. – Litres, 2022.
11. Быков, Д. В. Кластерный анализ на основе многомерных средних по результатам ВСХП-2016 с использованием Python / Д. В. Быков, А. В. Уколова // Бухучет в сельском хозяйстве. – 2022. – № 12. – С. 834.
12. Вандер Дж.Плас. Python для сложных задач. Наука о данных и машинное обучение. Питер, 2018. ISBN 978-5-446-10914-2
13. Воройский, Ф. С. Основы проектирования автоматизированных библиотечно-информационных систем / Ф.С. Воройский. - М.: ФИЗМАТЛИТ, 2023. - 384 с.

14. Гаджиева, А. Г. Цифровизация и занятость: роль отраслей сектора услуг / А. Г. Гаджиева // Инновации. - 2018. - № 2. - С. 61-70
15. Гома Х. UML. Проектирование систем реального времени, распределенных и параллельных приложений. – Litres, 2022.
16. Гордиенко Е. П. Обзор технологий, методов и инструментальных систем разработки программного обеспечения //Актуальные проблемы и перспективы развития транспорта, промышленности и экономики России (ТрансПромЭк 2021). – 2021. – С. 23-29.
17. Горшкова, Д. А. Анализ системы показателей целей устойчивого развития / Д. А. Горшкова // Материалы II национальной научной конференции "Современные направления статистических исследований", Москва, 26 декабря 2023 года. – Москва: Российский государственный аграрный университет, 2024. – С. 122-130. – EDN APKNRM.
18. Гумерова Г. Р., Мансурова Т. Г. Проектирование и разработка компонентов информационной системы образовательного учреждения на примере проекта автоматизации работы специалиста по научно-исследовательской работе. – 2022.
19. Данные по показателям ЦУР // Федеральная служба государственной статистики URL: <https://rosstat.gov.ru/sdg/data/goal2> – Дата обращения: 03.06.2025.
20. Демичев, В. В. Статистическое исследование инвестирования в сельское хозяйство России в условиях реализации государственных программ : монография / В. В. Демичев, В. В. Маслакова. – Иркутск : ООО "Мегапринт", 2017. – 162 с.
21. Дроздов И. С. Место продовольственной безопасности в системе национальной безопасности России в контексте целей в области устойчивого развития ООН //Агропродовольственная политика России. – 2024. – №. 4. – С. 10-16.
22. Ерохин, В. В. Безопасность информационных систем. Учебное пособие / В.В. Ерохин, Д.А. Погоньшева, И.Г. Степченко. - М.: Флинта, Наука, 2018. - 184 с.
23. Затонский, А. В. Информационные технологии. Разработка информационных моделей и систем. Учебное пособие / А.В. Затонский. - М.: Инфра-М, РИОР, 2018. - 344 с.

24. Захариади Э. В., Авруцкая С. Г. Внедрение целей устойчивого развития зарубежными и российскими компаниями //Вестник российского химико-технологического университета имени ДИ Менделеева: Гуманитарные и социально-экономические исследования. – 2022. – №. 13-4. – С. 144-155.

25. Зюкин Д. А. и др. Состояние продовольственной безопасности России в контексте самообеспечения ключевыми видами продуктов //Вестник НГИЭИ. – 2023. – №. 4 (143). – С. 99-111.

26. Икромов Х. Х. Создание информационной системы //Universum: технические науки. – 2021. – №. 2-1. – С. 21-22.

27. Информационно-аналитическое обеспечение устойчивого развития сельского хозяйства / М. В. Кагирова, В. В. Демичев, Ю. Н. Романцева [и др.]. – Москва : Российский государственный аграрный университет - МСХА им. К.А. Тимирязева, 2023. – 307 с.

28. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем / Э.Р. Ипатова, Ю.В. Ипатов. - М.: Флинта, 2021. - 256 с.

29. Исаенко Е. В., Тарасова Е. Е., Исаенко А. В. Оценка достижения целей устойчивого развития предприятиями общественного питания потребительской кооперации //Вестник Белгородского университета кооперации, экономики и права. – 2021. – №. 3. – С. 9-22.

30. Искра Е. А., Нелюбина Ю. А., Свиридова И. И. сравнительный анализ нотаций ARIS, IDEF, BPMN 2.0 и" ФИСОМ" при описании бизнес-процессов //Российские регионы в фокусе перемен. – 2019. – С. 472-476.

31. Использование нотации eEPC для графического описания бизнес-процессов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/143273/>, свободный – Дата обращения: 21.12.2024.

32. Карминский, А.М. Методология создания информационных систем / А.М. Карминский. - М.: ИНФРА-М, 2018. - 193 с.

33. Кислицын Е. В. Современные технологии разработки программного обеспечения : учеб. пособие / Е. В. Кислицын, М. А. Панов. - Екатеринбург : Изд-во УрГЭУ, 2021. - 176 с.

34. Колос Н. В., Банчук Г. Г. Методические аспекты анализа бизнес-процессов хозяйствующих субъектов //Современные исследования основных направлений технических и общественных. – 2017. – С. 476.

35. Корреляционно-регрессионный анализ влияния экономических факторов на урожайность пшеницы / В. И. Хоружий, Д. В. Быков, А. В. Уколова, А. Г. Ибрагимов // Бухучет в сельском хозяйстве. – 2024. – № 8. – С. 557-571.

36. Крылов С. И. Развитие методологии анализа в сбалансированной системе показателей. – 2021.

37. Куприянов Д. В. Информационное и технологическое обеспечение профессиональной деятельности : учебник и практикум / Д. В. Куприянов ; Финансовый ун-т при Правительстве Рос. Федерации. - Москва : Юрайт, 2017. - 255 с.

38. Ланьшина Т. А. Опыт локализации и внедрения Целей устойчивого развития в странах–лидерах в данной сфере //Вестник международных организаций: образование, наука, новая экономика. – 2019. – Т. 14. – №. 1. – С. 207-224.

39. Лыгина Н. И., Пудич А. С. Исследование правильности и эффективности средств парсинга информации на веб-ресурсах //Инновационная наука. – 2017. – №. 3-1. – С. 59-67.

40. Любанович Б.. Простой Python. Современный стиль программирования. Питер, 2019, 480 с. ISBN 978-5-446-11054-4

41. Мазур Е. В., Бабешко В. Н. Парсинг как источник получения данных для анализа //Перспективное развитие науки, техники и технологий. – 2019. – С. 161-164.

42. Мальцев А. А., Мальцева В. А. Цифровизация экономики в контексте реализации целей устойчивого развития: обзор ключевых экспертных докладов 2019 г //Вестник международных организаций: образование, наука, новая экономика. – 2020. – Т. 15. – №. 4. – С. 189-195.

43. Маслакова, В. В. Статистический анализ эффективности инвестирования в сельское хозяйство регионов России / В. В. Маслакова // АПК: экономика, управление. – 2018.

44. Маслакова, В. В. Статистический анализ эффективности инвестирования в развитие сельского хозяйства в России / В. В. Маслакова, В. В. Демичев. – Москва : Общество с ограниченной ответственностью "Научный консультант", 2021. – 194 с.

45. Мельник Л. Г. и др. Устойчивое развитие: теория, методология, практика. – 2019.

46. Методические указания по составлению годовых балансов продовольственных ресурсов // Постановление Федеральной службы государственной статистики от 25 декабря 2006 г. № 82. URL: [https://rosstat.gov.ru/storage/mediabank/met-bal\\_prod.rar](https://rosstat.gov.ru/storage/mediabank/met-bal_prod.rar) – Дата обращения: 10.11.2024.

47. Министерство сельского хозяйства РФ. Стратегия развития сельского хозяйства до 2030 года [Электронный ресурс]. — URL: <https://mcx.gov.ru/upload/iblock/fda/p8s312xvzbzgbnme51z16c4mmn5rnilp.pdf> – Дата обращения: 12.11.2024.

48. Митяков С. Н. Новые цели устойчивого развития России // Развитие и безопасность. – 2023. – Т. 1. – №. 17. – С. 21-35.

49. Мониторинг обеспеченности государств-членов и Союза сельскохозяйственной продукцией и продовольствием [Электронный ресурс]. – Режим доступа: <https://agro.eaeunion.org/MonitoringFS/Pages/default.aspx>, свободный – Дата обращения: 01.06.2025.

50. Невзоров, А. С. Роль и место больших данных в официальной статистике сельского хозяйства / А. С. Невзоров, О. Н. Бекетова, А. М. Иванов // Бухучет в сельском хозяйстве. – 2024. – № 7. – С. 484-494.

51. Невзоров, А. С. Экосистема для анализа больших данных в сельском хозяйстве / А. С. Невзоров, В. В. Демичев // Московский экономический журнал. – 2023. – Т. 8, № 5.

52. Неволина Т. С., Алешко Р. А. Парсинг больших объемов данных //ББК 3 П27. – 2019. – С. 56.

53. Нортон Д., Каплан Р. Сбалансированная система показателей. От стратегии к действию. – Litres, 2022.

54. Повестка дня в области устойчивого развития // Организация объединенных наций URL: <https://www.un.org/sustainabledevelopment/ru/about/development-agenda/> – Дата обращения: 15.11.2024.

55. Пыжикова Н. И., Колесняк А. А., Полянская Н. М. Дифференциация регионов по уровню самообеспечения продовольствием на основе оценки природных условий //Социально-экономический и гуманитарный журнал Красноярского ГАУ. – 2021. – №. 2 (20). – С. 23-35.

56. Рущицкая О. А. и др. АНАЛИЗ УРОВНЯ САМООБЕСПЕЧЕНИЯ ОСНОВНЫМИ ПРОДУКТАМИ ПИТАНИЯ В КОНТЕКСТЕ ДИНАМИКИ ПРОИЗВОДСТВА РАСТЕНИЕВОДЧЕСКОЙ ПРОДУКЦИИ И ДЕЯТЕЛЬНОСТИ СЕЛЬСКОХОЗЯЙСТВЕННЫХ ОРГАНИЗАЦИЙ //Вестник Академии знаний. – 2024. – №. 5 (64). – С. 355-358.

57. Сахаров А. Г., Колмар О. И. Перспективы реализации Целей устойчивого развития ООН в России //Вестник международных организаций: образование, наука, новая экономика. – 2019. – Т. 14. – №. 1. – С. 189-206.

58. Сборник трудов приуроченных к 74-й Всероссийской студенческой научно-практической конференции, посвященной 200-летию со дня рождения П.А.Ильенкова / Коллектив авторов; Российский государственный аграрный университет – МСХА имени К. А. Тимирязева. – Москва: РГАУ – МСХА имени К. А. Тимирязева, 2021. – 213 с.

59. Скрипкин К. Экономическая эффективность информационных систем. – Litres, 2022.

60. Современные проблемы статистики сельского хозяйства и окружающей природной среды / А. П. Зинченко, В. М. Баутин, А. Д. Думнов [и др.]. – Москва :

Российский государственный аграрный университет - МСХА им. К.А. Тимирязева, 2016. – 198 с.

61. Соколов О. В., Зайцева Н. А. ПОВЫШЕНИЕ УРОВНЯ САМООБЕСПЕЧЕНИЯ РЕГИОНА ПРОДУКЦИЕЙ САДОВОДСТВА //Наука и Образование. – 2023. – Т. 6. – №. 4.

62. Старикова Е. А. Современные подходы к трактовке концепции устойчивого развития //Вестник Российского университета дружбы народов. Серия: Экономика. – 2017. – Т. 25. – №. 1. – С. 7-17.

63. Ступкин, В. В. Методология оценки качества интегрированных библиотечно-информационных систем / В.В. Ступкин. - М.: Литера, 2019. - 273 с.

64. Тесля А. Б., Зайченко И. М., Хашева З. М. РАЗРАБОТКА КОНЦЕПЦИИ СТРАТЕГИЧЕСКОГО РАЗВИТИЯ РАЙОНОВ КРАЙНЕГО СЕВЕРА НА ОСНОВЕ ПОСТРОЕНИЯ СИСТЕМЫ СБАЛАНСИРОВАННЫХ ПОКАЗАТЕЛЕЙ В УСЛОВИЯХ ЦИФРОВОЙ ТРАНСФОРМАЦИИ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ //ГЛАВНЫЙ РЕДАКТОР. – 2022. – С. 58.

65. Технологии обработки больших данных средствами СУБД ADABAS : учеб. пособие / В. П. Часовских, В. Г. Лабунец, Е. Н. Стариков [и др.]. - Екатеринбург : Изд-во УрГЭУ, 2022. - 170 с.

66. Титов А. Д. МЕТОДЫ И АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА БОЛЬШИХ ДАННЫХ В СЕЛЬСКОМ ХОЗЯЙСТВЕ //Международной научно-практической конференции «Тренды развития сельского хозяйства и агрообразования в парадигме Зеленой экономики». – 2023. – С. 29.

67. Фаулер М. Архитектура корпоративных программных приложений / М. Фаулер; пер. с англ. П. В. Семенова. — Москва: Вильямс, 2020. — 542 с.

68. Филиппов, В. А. Многомерные СУБД при создании корпоративных информационных систем / В.А. Филиппов. - М.: Едиториал УРСС, 2019. - 194 с.

69. Формализация информации и BIG DATA : учеб. пособие / В. П. Часовских, М. П. Воронов, В. Г. Лабунец [и др.]. - Екатеринбург : Изд-во УрГЭУ, 2021. - 218 с.

70. Харитонов А. Е., Титов А. Д. Разработка модуля информационной системы распознавания образов //ЦИФРОВЫЕ ТЕХНОЛОГИИ АНАЛИЗА ДАННЫХ В СЕЛЬСКОМ ХОЗЯЙСТВЕ. – 2022. – С. 211-241.
71. Цебренько К. Н. Моделирование информационной системы учета результатов исследований в нотации UML //Электронный научный журнал. – 2016. – №. 7. – С. 113-116.
72. Чупина И. П., Симачкова Н. Н. РОЛЬ РЕГИОНАЛЬНЫХ ПРОЕКТОВ В ПОВЫШЕНИИ УРОВНЯ САМООБЕСПЕЧЕНИЯ СЕЛЬСКОХОЗЯЙСТВЕННОЙ ПРОДУКЦИЕЙ //Правовые и социально-экономические проблемы современной. – 2021. – С. 261.
73. Шелухин, О. И. Моделирование информационных систем / О.И. Шелухин. - Москва: Огни, 2023. - 536 с.
74. Ярочкин, В. Безопасность информационных систем / В. Ярочкин. - М.: Осъ-89, 2017. - 320 с.
75. Abney S. P. Principle-based parsing //12th Annual Conference. CSS Pod. – Psychology Press, 2022. – С. 1021-1021.
76. Atukunda P. et al. Unlocking the potential for achievement of the UN Sustainable Development Goal 2–‘Zero Hunger’–in Africa: targets, strategies, synergies and challenges //Food & nutrition research. – 2021. – Т. 65. – С. 10.29219/fnr. v65. 7686.
77. Bagnall A., Lines J., Bostrom A., Large J., Keogh E.. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 2017, V.31
78. Benfort B., T.Ojeda, R.Bilbro. Applied Text Analysis with Python. Enabling Language-Aware Data Products with Machine Learning. O’Reilly Media, 2018
79. Berrone P. et al. How can research contribute to the implementation of sustainable development goals? An interpretive review of SDG literature in management //International Journal of Management Reviews. – 2023. – Т. 25. – №. 2. – С. 318-339.
80. Dashieva, B. Analysis of the influence of agricultural climatic conditions on the allocation of labor resources in agriculture / B. Dashieva, A. Ukolova // VIII International Scientific and Practical Conference 'Current problems of social and labour



relations' (ISPC-CPSLR 2020) : Proceedings of the VIII International Scientific and Practical Conference, Makhachkala, 17–18 декабря 2020 года. Vol. 527. – Amsterdam: Atlantis Press, 2021.

81. DataStart. Библиотеки Python [Электронный ресурс] // datastart.ru: информ.-справочный портал. URL: <https://datastart.ru/blog/read/top-10-bibliotek-python-dlya-data-science> – Дата обращения 20.05.2025

82. Davletshina L. A., Telyuk M. S., Sadovnikova N. A. Inequality degree within the country in the context of realising the Sustainable Development Goals //Вестник университета. – 2022. – С. 6.

83. Demichev, V. V. Influence of investments and subsidies on the efficiency of agriculture in Russia during the implementation of state programs / V. V. Demichev, V. V. Maslakova // IOP Conference Series: Earth and Environmental Science, Yekaterinburg, 15–16 октября 2020 года. – IOP Publishing Ltd: IOP Publishing Ltd, 2021.

84. FAO. The State of Food Security and Nutrition in the World 2024: Ending Hunger, Food Security and Nutrition [Electronic Resource] / Food and Agriculture Organization of the United Nations (FAO). — URL: <https://openknowledge.fao.org/server/api/core/bitstreams/39dbc6d1-58eb-4aac-bd8a-47a8a2c07c67/content/state-food-security-and-nutrition-2024/ending-hunger-food-security.html#gsc.tab=0> – Дата обращения: 15.11.2024.

85. Few S. Information Dashboard Design: The Effective Visual Communication of Data / S. Few. — Oakland: Analytics Press, 2016. — 223 p.

86. Gebara C. H., Laurent A. National SDG-7 performance assessment to support achieving sustainable energy for all within planetary limits //Renewable and Sustainable Energy Reviews. – 2023. – Т. 173. – С. 112934.

87. Gee D. G. Early adversity and development: parsing heterogeneity and identifying pathways of risk and resilience //American Journal of Psychiatry. – 2021. – Т. 178. – №. 11. – С. 998-1013.

88. Github language stats [Электронный ресурс]. URL: [https://madnight.github.io/github/#/pull\\_requests/2023/1](https://madnight.github.io/github/#/pull_requests/2023/1) – Дата обращения: 21.12.2024.

89. Gliessman S. R. Agroecology: The Ecology of Sustainable Food Systems / S. R. Gliessman. — 3rd ed. — Boca Raton: CRC Press, 2015. — 405 p.
90. Nechaev V. I. Некоторые аргументы к методике расчета уровня самообеспечения России основными сельскохозяйственными культурами: методологический аспект //Economy, Work, Management in Agriculture. — 2022. — №. 4.
91. Pleskachev Y., Ponomarev Y. МОНИТОРИНГ ДОСТИЖЕНИЯ НАЦИОНАЛЬНЫХ ЦЕЛЕЙ УКАЗА№ 204 С ИСПОЛЬЗОВАНИЕМ МЕЖДУНАРОДНОГО ОПЫТА И СИСТЕМЫ ПОКАЗАТЕЛЕЙ УСТОЙЧИВОГО РАЗВИТИЯ (Monitoring the Achievement of National Goals of Decree No. 204 Using International Experience and the System of Sustainable Development Indicators) //Available at SSRN 3867499. — 2021.
92. Pretty J. Sustainable Intensification of Agriculture: Greening the World's Food Economy / J. Pretty, Z. P. Bharucha. — London: Routledge, 2018. — 292 p.
93. Provost F. Data Science for Business / F. Provost, T. Fawcett. — Sebastopol: O'Reilly, 2013. — 414 p.
94. Qin B. et al. A survey on text-to-sql parsing: Concepts, methods, and future directions //arXiv preprint arXiv:2208.13629. — 2022.
95. Rachmad Y. E. Danantara and the Future of Impact Investing: Strengthening ESG and SDG Commitments //The United Nations Global Compact. — 2025.
96. Shvedova N. A. The UN and the sustainable development goals: on the way to implementation //В РОССИЙСКОМ ОБЩЕСТВЕ. — 2022. — С. 4.
97. Smith J. Data-driven approaches to SDG monitoring in agri-food systems // Journal of Sustainable Development. — 2021. — Vol. 14, No. 3. — P. 112–125.
98. Streamlit documentation [Электронный ресурс]. — Режим доступа: <https://docs.streamlit.io/> — Дата обращения: 03.06.2025.
99. Van Latesteijn H. C. Sustainable Agricultural Development: The Role of International Cooperation / H. C. Van Latesteijn, J. Andeweg. — Dordrecht: Springer, 2019. — 210 p.

100. World Bank. Monitoring Agricultural Sustainability: A Review of Indicators and Data Systems. — Washington, DC: World Bank, 2021. — 134 p.

## **Глава 3. Программный сервис для определения заболеваний растений с применением технологий Big Data и компьютерного зрения**

### **3.1. Обзор технологий: Big Data, компьютерное зрение и веб-разработка в агротехнологиях**

Различные отрасли промышленности сталкиваются с проблемами, связанными с хранением и анализом больших объемов данных.

В настоящее время системы больших данных становятся очень важным фактором инноваций и роста благодаря аналитическому анализу информации, получаемой в результате массивной обработки полученных данных. Требования бизнеса и приложений варьируются в зависимости от области применения.

При работе с большими данными важно понимать, что делает данные «большими», ограничения работы с ними и что можно сделать для борьбы с этими ограничениями. Большие данные относятся не только к тому, сколько данных могут занимать места в хранилище, но и к целому ряду качеств.

Размер данных настолько велик, что они не могут поместиться на одном компьютере (например,  $> 1$  ТБ). Скорость создания данных высока и быстро накапливается (например, 100 ГБ в час). Форматы данных сложны и разнообразны (например, текст, изображения, видео). Неопределенность в отношении надежности и качества данных.

Новые данные, вычислительные возможности и методы могут создать некоторые возможности и проблемы, такие как интеграция статистики/машинного обучения для оценки множества моделей проведет калибровку по всем необходимым данным.

Традиционные исследовательские подходы основаны на первоначальном формировании некоторой модели, или накоплении и обработке данных с последующей оценкой параметров такой заранее сформированной модели.

Такое комплексное исследование решает задачу поиска модели или набора моделей, наиболее соответствующих некоторым экспериментальным данным. Более того, эта методология направлена на поиск наиболее интересных и надежных

версий. В основе технологии обработки больших данных лежат относительно простые принципы.

Распределенная архитектура системы хранения данных и полная параллельная обработка на самом низком уровне, данные не перемещаются во время обработки, задачи обработки данных доставляются и выполняются на вычислительных ресурсах распределенных систем хранения, они не удаляются и не изменяются во время обработки, результаты просто сохраняются на месте.

Широкое внедрение параллелизма даже в массовые компьютеры открывает новые возможности для разработчиков, позволяющих реализовать многовариантные методы обработки.

Простота развертывания распределенных систем позволяет быстро наращивать возможности целевых систем. Рассматривая задачи компьютерного зрения, наиболее интересной является обработка больших данных в реальном времени.

Можно разделить этот процесс на пять фаз. Такие как дистилляция данных, разработка модели, проверка и развертывание, оценка в реальном времени и обновление модели.

Дистилляция данных включает в себя извлечение характеристик, объединение различных источников, фильтрацию по интересующим областям, выбор соответствующих характеристик и результатов для моделирования и экспорт данных в локальное хранилище.

Процессы на следующем этапе включают выбор характеристик, преобразование переменных, оценку модели, уточнение модели. Целью на этом этапе является создание прогностической модели, которая является мощной, надежной, понятной и реализуемой.

Далее идет этап проверка и развертывание. Целью этого этапа является тестирование модели для обеспечения ее работоспособности. Процесс проверки включает повторное извлечение новых данных, запуск и сравнение результатов с полученными данными, которые были удержаны в качестве выбора проверки. Если модель работает, ее можно развернуть в производственной среде.

По мере реализации модели можно оценить, как производительность реагирует на эти изменения. Возможно ли визуализировать, насколько улучшается модель, когда добавляется больше изображений. Эти визуализации называются кривыми обучения, графиками.

Архитектуры программного обеспечения для систем с большими данными ранее изучались эпизодически или интенсивно.

Однако предложить подходящую архитектуру программного обеспечения для систем с большими данными не просто, учитывая как прикладные требования, так и интересы заинтересованных сторон

Основные характеристики Big Data это объем, достоверность, разнообразие, значение и скорость обработки [1].

Само название Big Data означает в прямом смысле «большие данные». Big Data подразумевают под собой огромные «объемы» данных, которые ежедневно генерируются из множества источников, таких как бизнес-процессы, машины, платформы социальных сетей, сети, человеческие взаимодействия и многое другое.

Facebook может генерировать около миллиарда сообщений, 4,5 миллиарда раз, когда нажимается кнопка «Нравится», и более 350 миллионов новых постов загружаются каждый день. Технологии Big Data помогают человеку обрабатывать большие объемы данных.

Они могут быть структурированными, неструктурированными и полуструктурированными и собираются из разных источников. В прошлом информация собиралась только из баз данных и таблиц, но в наши дни данные поступают в виде массивов, то есть PDF-файлов, электронных писем, аудиозаписей, сообщений SM, фотографий, видео и т. д.

Если говорить о классификации данных, то можно их структурировать как структурированные данные в структурированной схеме вместе со всеми необходимыми столбцами. Они находятся в табличной форме, хранятся в системе управления реляционными базами данных.

Полуструктурированные в полуструктурированной схеме, эта схема не определена надлежащим образом, как JSON, XML, CSV, TSV и электронная почта.

Системы OLTP (обработка онлайн-транзакций) созданы для работы с полуструктурированными данными. Они хранятся в таблицах.

Все неструктурированные данные, журналы, аудиофайлы и изображения включены в неструктурированные данные. Некоторые организации имеют много доступных данных, но они не знают, как извлечь их ценность, поскольку информация недостаточно необработана.

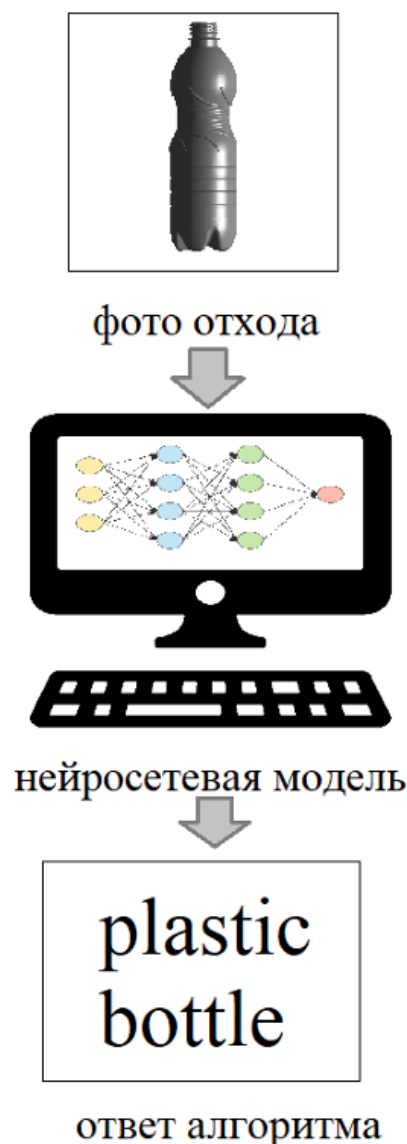
Квазиструктурированные данные представляет собой формат данных, который содержит текстовые данные с непоследовательными форматами.

Компьютерное зрение — это область искусственного интеллекта, которая использует машинное обучение и нейронные сети для обучения компьютеров и систем, позволяет извлекать ключевую информацию из цифровых изображений, видео и других визуальных данных, а также давать рекомендации или предпринимать действия при обнаружении проблем.

Компьютерное зрение работает почти так же, как и человеческое. Человеческое зрение имеет преимущество в виде контекста, который позволяет тренироваться различать объекты, насколько они далеки, движутся ли они или что-то не так с изображением.

Компьютерное зрение обучает машины выполнять эти функции, но оно должно делать это за гораздо меньшее время с помощью камер, данных и алгоритмов, а не сетчатки, зрительных нервов и зрительной коры. Поскольку система, обученная проверять продукты или следить за производственным активом, может анализировать тысячи продуктов или процессов в минуту, замечая незаметные дефекты или проблемы, она может быстро превзойти человеческие возможности.

На рисунке 1.1 представлена схема работы компьютерного зрения для классификации отходов. Отход фотографируется, снимок отправляется для анализа в нейросетевую модель, и, в результате, получается ответ алгоритма, например, plastic bottle (пластиковая бутылка).



**Рисунок 1.43 – Пример работы компьютерного зрения [3]**

Компьютерному зрению нужно много данных. Оно, не прекращая анализирует данные, пока не увидит различия и в конечном итоге не распознает изображения. Чтобы научить компьютер распознавать автомобильные шины, ему нужно предоставить огромное количество изображений шин и связанных с ними предметов, чтобы изучить различия и распознать шину, особенно без дефектов.

Для этого используются две основные технологии: тип машинного обучения, называемый глубоким обучением, и сверточная нейронная сеть (CNN).

Сверточная нейронная сеть была представлена Йеном Лекуном в 1980-х годах 21. В 2012 году, когда AlexNet 9 выиграла соревнование ImageNet, CNN стали более популярными и использовались в различных проектах для классификации и



распознавания изображений, обработки языков, медицинских изображений и т. д. В структуре сверточных нейронных сетей (CNN) признаки изображения извлекаются с сохранением 2D-структуры, при этом эта структура состоит из ряда фильтров. При перемещении фильтров по входному изображению происходит расчет для извлечения признаков.

Машинное обучение использует алгоритмические модели, которые позволяют компьютеру обучаться контексту визуальных данных. Если через модель подается достаточно данных, компьютер «посмотрит» на данные и научится отличать одно изображение от другого. Алгоритмы позволяют машине обучаться самостоятельно, а не программировать ее для распознавания изображения.

CNN помогает модели машинного обучения или глубокого обучения «смотреть», разбивая изображения на пиксели, которым присваиваются теги или метки. Она использует метки для выполнения сверток (математическая операция над двумя функциями для получения третьей функции) и делает прогнозы о том, что она «видит». Нейронная сеть выполняет свертки и проверяет точность своих предсказаний в серии итераций, пока предсказания не начнут сбываться. Затем она распознает или видит изображения способом, похожим на человеческий.

Подобно тому, как человек различает изображение на расстоянии, CNN сначала различает резкие края и простые формы, а затем заполняет информацию, выполняя итерации своих предсказаний. CNN используется для понимания отдельных изображений. Рекуррентная нейронная сеть (RNN) используется аналогичным образом для видеоприложений, чтобы помочь компьютерам понять, как изображения в серии кадров связаны друг с другом.

Веб-разработка, также известная как разработка веб-сайтов, относится к задачам, связанным с созданием, построением и обслуживанием веб-сайтов и веб-приложений, которые работают в Интернете в браузере. Тем не менее, он может также включать в себя веб-дизайн, веб-программирование и управление базами данных.

Она тесно связана с проектированием функций и функциональности приложений. Термин «разработка» обычно используется для обозначения разработки сайтов.

Основными инструментами, используемыми в веб-разработке, являются языки программирования HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) и JavaScript. Существует, однако, ряд других программ, используемых для «управления» или облегчения строительства сайтов, которые в противном случае пришлось бы делать «с нуля» путем написания кода. В эту категорию попадает ряд систем управления контентом (CMS), в том числе WordPress, Joomla!, Drupal, TYPO3 и Adobe Experience Manager.

Веб-разработка также включает работу с фреймворками и инструментами, которые упрощают процесс разработки. Фреймворки предоставляют набор готовых компонентов и инструментов для быстрой разработки приложений. Некоторые из популярных фреймворков включают React, Angular, Vue.js для клиентской разработки и Django, Ruby on Rails, Laravel для серверной разработки.

После завершения разработки веб-сайта или приложения, они размещаются на веб-сервере, чтобы пользователи могли получить к ним доступ через интернет.

### **3.2. Процессы диагностики заболеваний сельскохозяйственных культур: от традиционных к автоматизированным**

Сегодня при традиционной диагностике заболеваний сельскохозяйственных культур ключевую роль играет человек (агроном) [4].

Для диагностики заболевания ему требуется непосредственное присутствие в местах посадки культуры. Когда работник прибыл на место посадки, он начинает внешний осмотр первого растения на предмет заболеваний. Если агроном замечает и идентифицирует заболевание, то он заносит данные (дата и время, номер растения, заболевание) в дневник учёта.

Если у агронома есть подозрение, что растение заболело, но распознать заболевание на месте невозможно, тогда агроном производит фотосъёмку

поврежденного участка растения для дальнейшего анализа, используя литературу и другие источники.

После осмотра первого растения агроном переходит к следующему, пока не осмотрит всё. После осмотра всех растений агроном начинает идентификацию нераспознанных заболеваний и внесение результатов в дневник. Внешняя диагностика заболеваний завершена. Диаграмма, описывающая процесс, приведена в Приложении А на рис. А.1.

Современная диагностика заболеваний, основанная на наблюдении за человеком, требует много времени и денег. Кроме того, большие различия в симптомах из-за возраста инфицированной ткани, генетических вариаций и условий освещенности деревьев снижают точность обнаружения. Хотя модели компьютерного зрения могут повысить эффективность выявления заболеваний. Так, осмотр растений предлагается автоматизировать за счет установки камер, фотографирующих растение под определенным углом и в определенное время, а также путем обучения нейросетевого классификатора болезней.

Следует отметить, что капитальные затраты на процесс увеличиваются, но они компенсируются снижением постоянных затрат: дорогому агроному приходится реже посещать места посадки.

Во время  $T$  камера  $C$  будет фотографировать растение  $N$  с ракурса  $M$  и отправлять фото в обученный классификатор. Если классификатор не обрабатывает ранее загруженное фото, то начинается классификация: есть ли заболевание, при наличии запускается вторая классификация - тип заболевания. Если классификатор занят, то фото добавляется в очередь.

После распознавания результаты сохраняются на сервере в базе данных. Если классификатор не смог распознать тип заболевания, то входные данные отправляются агроному, который идентифицирует заболевание и проставляет в базе результат.

Если тип заболевания определён, то агроном уведомляется о заболевании для принятия мер. Диаграмма, описывающая процесс, приведена в Приложении А на рис. А.2.

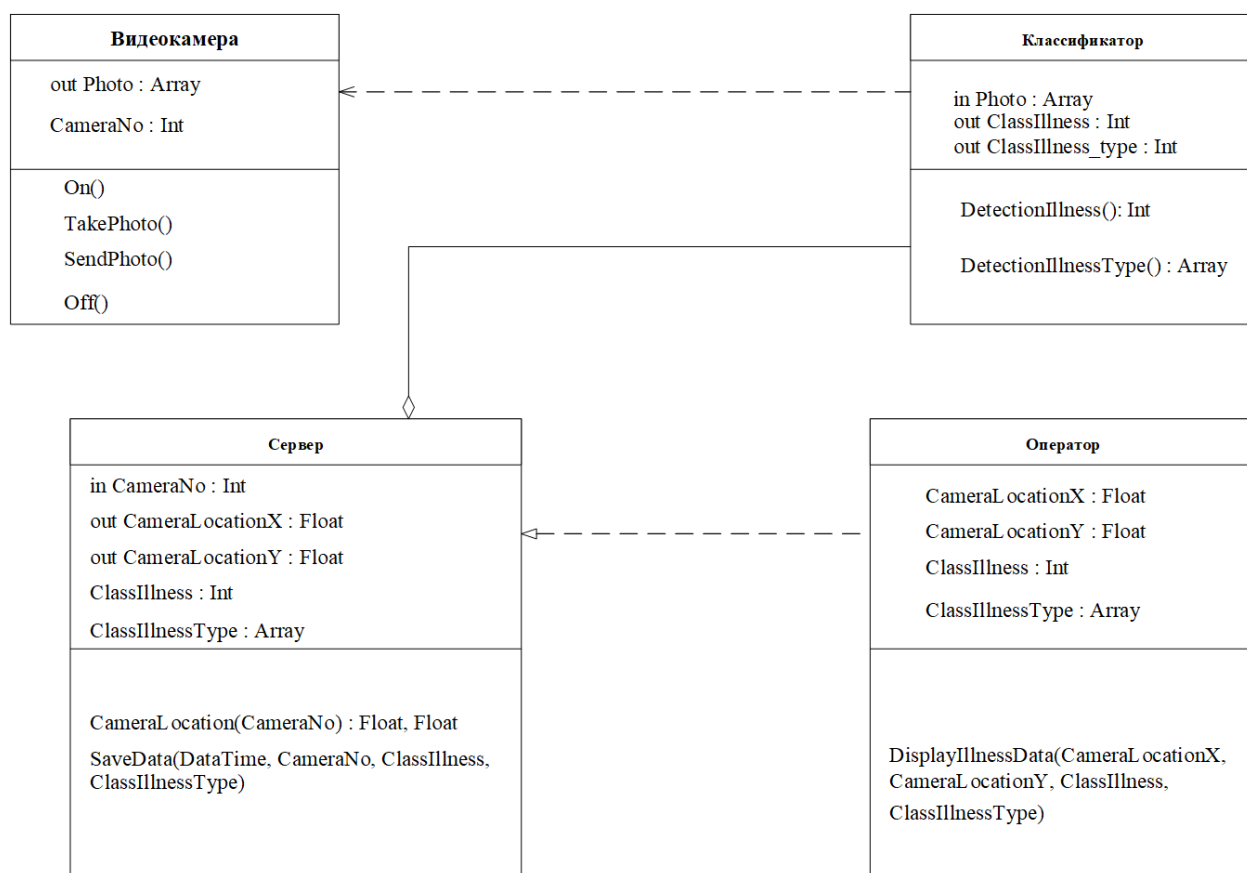
Система предусматривает загрузку фото для анализа в систему, определение наличия/отсутствия заболевания культуры, типа заболевания. Эти функции пользователь сможет использовать в системе.

Для представления компонентной архитектуры системы воспользуемся диаграммой классов (Рисунок 3.2. Компонент и его характеристики будут описаны как класс, его атрибуты и операции) [6, 10]. Автоматизация описана техническими, видеокамерой, классификатором, сервером, человеческий фактор, оператор, классами

На диаграмме классов основным классом является Сервер, в котором хранятся данные. Для получения информации о наличии/отсутствии заболевания, типе заболевания используется класс классификации (например, искусственный интеллект, обнаруживающий заболевания). Класс имеет отношения агрегации с основным классом Сервер: сначала мы создаем объект класса Классификатор, затем передаем его в качестве параметра конструктора классу Сервер, чтобы его можно было использовать как часть класса.

Класс классификации зависит от класса видеокамеры: изменение одного элемента (независимого) может привести к изменению семантики другого элемента (зависимого).

Класс Оператора – это интерфейс к классу Сервера и своего рода поставщик реализации.



**Рисунок 3.2 – Диаграмма классов**

В таблице 3.1 приводится список классов отдельных элементов проектируемой информационной системы, данные и возможные операции.

Стоит отметить, что автоматизация описанного выше процесса не только упрощает работу агронома (оператора), поскольку отпадает необходимость часто посещать место посадки для визуального осмотра растений, но и становится основой системы. предотвратить развитие заболевания, выявив его на ранней стадии. Также систему можно дополнить следующими функциями: химическое обнаружение заболеваний – необходимо установить датчики химического состава почвы, воздуха и воды; Оценка влажности – необходимо установить датчик влажности.

**Таблица 3.4 – Описание классов системы и данных**

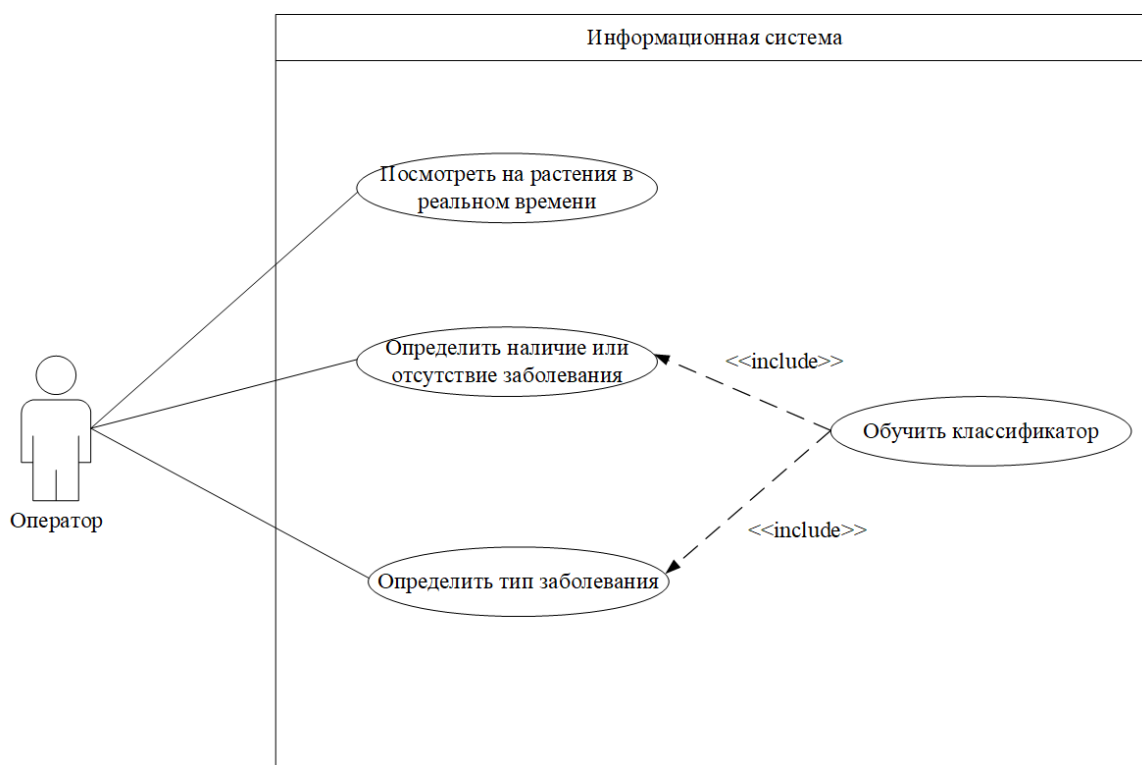
<b>Класс</b>	<b>Атрибуты</b>	<b>Операции</b>
Видеокамера	1. сделанное фото: out Photo : Array 2. номер камеры: CameraNo : Int	1. включить камеру: On() 2. сделать фото: TakePhoto() 3. отправить фото: SendPhoto() 4. выключить камеру: Off()
Классификатор	1. входное фото для классификации: in Photo : Array 2. наличие заболевания (есть/нет): out ClassIllness : Int 3. класс заболевания: out ClassIllnessType : Array	1. распознать наличие заболевания: DetectionIllness(): Int 2. распознать тип заболевания: DetectionIllnessType() : Array
Сервер	1. номер камеры, переданный от Видеокамера: in CameraNo : Int 2. расположение камеры по оси X (определяется по номеру камеры и таблице, хранимой на Сервере): out CameraLocationX : Float 3. расположение камеры по оси Y: out CameraLocationY : Float 4. переданное от Классификатора наличие заболевания: ClassIllness : Int 5. переданное от Классификатора класс заболевания: ClassIllnessType : Array	1. определить положение камеры: CameraLocation(CameraNo) : Float, Float 2. сохранить данные: SaveData(DataTime, CameraNo, ClassIllness, ClassIllnessType)
Оператор	расположение камеры по оси X: CameraLocationX : Float расположение камеры по оси Y: CameraLocationY : Float наличие заболевания: ClassIllness : Int класс заболевания: ClassIllnessType : Array	1. отобразить данные оператору: DisplayIllnessData(CameraLocationX, CameraLocationY, ClassIllness, ClassIllnessType)

Диаграмма вариантов использования, представленная на рисунок 3.3, описывает систему для определения заболеваний растений.

Оператор использует систему, чтобы посмотреть на растения в реальном времени, определить наличие или отсутствие заболевания и определить тип заболевания.

Системный элемент «Обучить классификатор» – это подсистема, которая используется для определения наличия или отсутствия заболевания и определения типа заболевания.

Обучить классификатор – это включенная подсистема, которая используется в обоих вариантах использования: определить наличие или отсутствие заболевания и определить тип заболевания.



**Рисунок 3.3 – Диаграмма вариантов использования**

Диаграмма кооперации описывает взаимодействие между объектами системы. В данной диаграмме, представленной на рис. 13, показана работа системы мониторинга состояния растений, включающей в себя следующие объекты:

- «Растение» – это объект, который представляет собой растение, нуждающееся в мониторинге;

- «Видеокамера» отвечает за съемку растения;
- «Классификатор» использует данные, полученные с видеокамеры, для анализа состояния растения;
- «Сервер» хранит данные о состоянии растения, полученные от классификатора, и предоставляет их оператору;
- «Оператор» – это человек, который использует систему мониторинга для отслеживания состояния растений.

Описание методов:

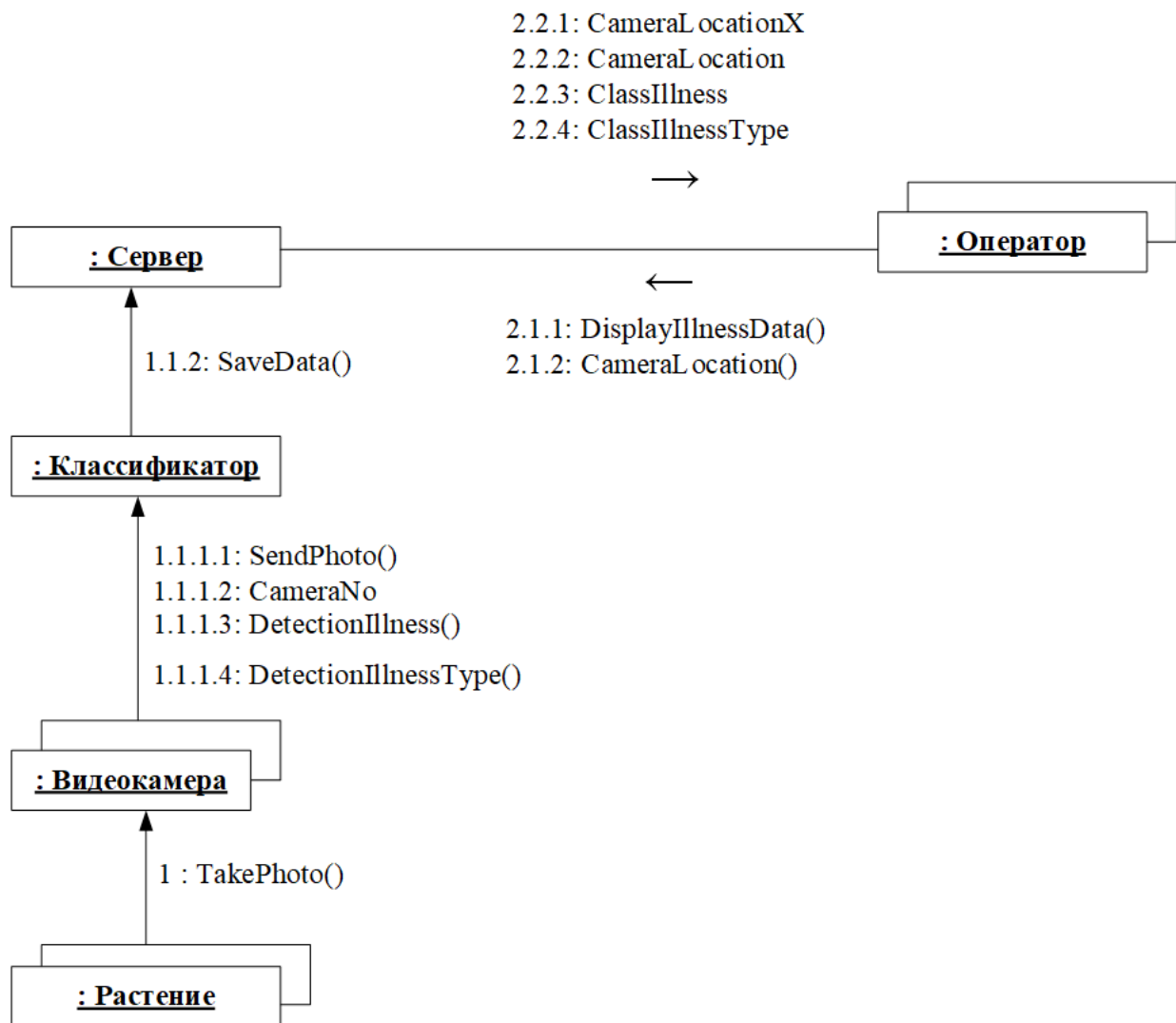
- TakePhoto() - метод, который запускается видеокамерой для получения изображения растения;
- SendPhoto() - метод, который отправляет изображение, полученное видеокамерой, классификатору;
- CameraNo - метод, который возвращает номер видеокамеры;
- DetectionIllness() - метод, который определяет наличие заболевания растения;
- DetectionIllnessType() - метод, который определяет тип заболевания растения;
- SaveData() - метод, который сохраняет информацию о состоянии растения на сервере;
- DisplayIllnessData() - метод, который отображает информацию о состоянии растения оператору;
- CameraLocation() - метод, который возвращает информацию о местоположении видеокамеры.

Взаимодействие между объектами:

- «Растение» не имеет активных действий, но является источником данных для системы;
- «Видеокамера» снимает изображение растения и отправляет его классификатору;



- «Классификатор» анализирует полученное изображение, используя метод `DetectionIllness()` и `DetectionIllnessType()` , и отправляет данные о состоянии растения на сервер;
- «Сервер» хранит полученные данные от классификатора, используя метод `SaveData()`, и отображает их оператору через метод `DisplayIllnessData()`;
- «Оператор» просматривает полученные данные о состоянии растения.



**Рисунок 3.4 – Диаграмма кооперации**

Таким образом, «Растение» является исходным объектом, который фотографируется «Видеокамерой». Фотография отправляется на «Классификатор», который определяет наличие и тип болезни, используя алгоритмы машинного обучения. «Классификатор» обрабатывает полученную фотографию, определяет тип заболевания, если оно обнаружено, и отправляет

результаты на «Сервер». «Сервер» сохраняет полученные данные о состоянии растения, в том числе, тип заболевания, фотографию и информацию о местоположении. «Оператор» запрашивает данные о состоянии растений с «Сервера», которые могут включать в себя информацию о типе заболевания и местоположении.

Важно отметить, что «Сервер» также может получать информацию о местоположении «Видеокамеры», что позволяет отслеживать состояние растений в разных местах.

В отличие от диаграммы последовательностей, диаграмма совместной работы показывает отношения между объектами. Диаграммы последовательностей и диаграммы совместной работы выражают схожую информацию, но отображают ее по-разному.

Из-за формата диаграммы совместной работы они, как правило, лучше подходят для аналитических операций (см. раздел Действие: анализ прецедентов). В частности, они, как правило, лучше подходят для изображения более простых взаимодействий меньшего числа объектов. Однако по мере того, как количество объектов и сообщений растет, диаграмму становится все труднее для чтения. Кроме того, трудно отобразить дополнительную описательную информацию, такую как время, точки принятия решений или другую неструктурированную информацию, которую можно легко добавить к примечаниям на диаграмме последовательности.

Итак, вот несколько вариантов использования, для которых мы хотим создать схему совместной работы:

Моделирование совместной работы между объектами или ролями, обеспечивающими функциональные возможности сценариев использования и операций

Моделирование механизмов в рамках архитектурного проекта системы

Захват взаимодействий, которые показывают передачу сообщений между объектами и ролями в рамках совместной работы

Моделирование альтернативных сценариев в рамках сценариев использования или операций, связанных с совместной работой различных объектов и взаимодействий

Поддержка идентификации объектов (следовательно, классов), которые участвуют в сценариях использования.

На основе диаграммы кооперации построены диаграммы последовательности, которая визуализирует особенность реализации конкретного варианта использования.

Диаграмма последовательности на рисунок 3.5 описывает взаимодействие между четырьмя объектами: «Растение», «Видеокамера», «Классификатор» и «Сервер».

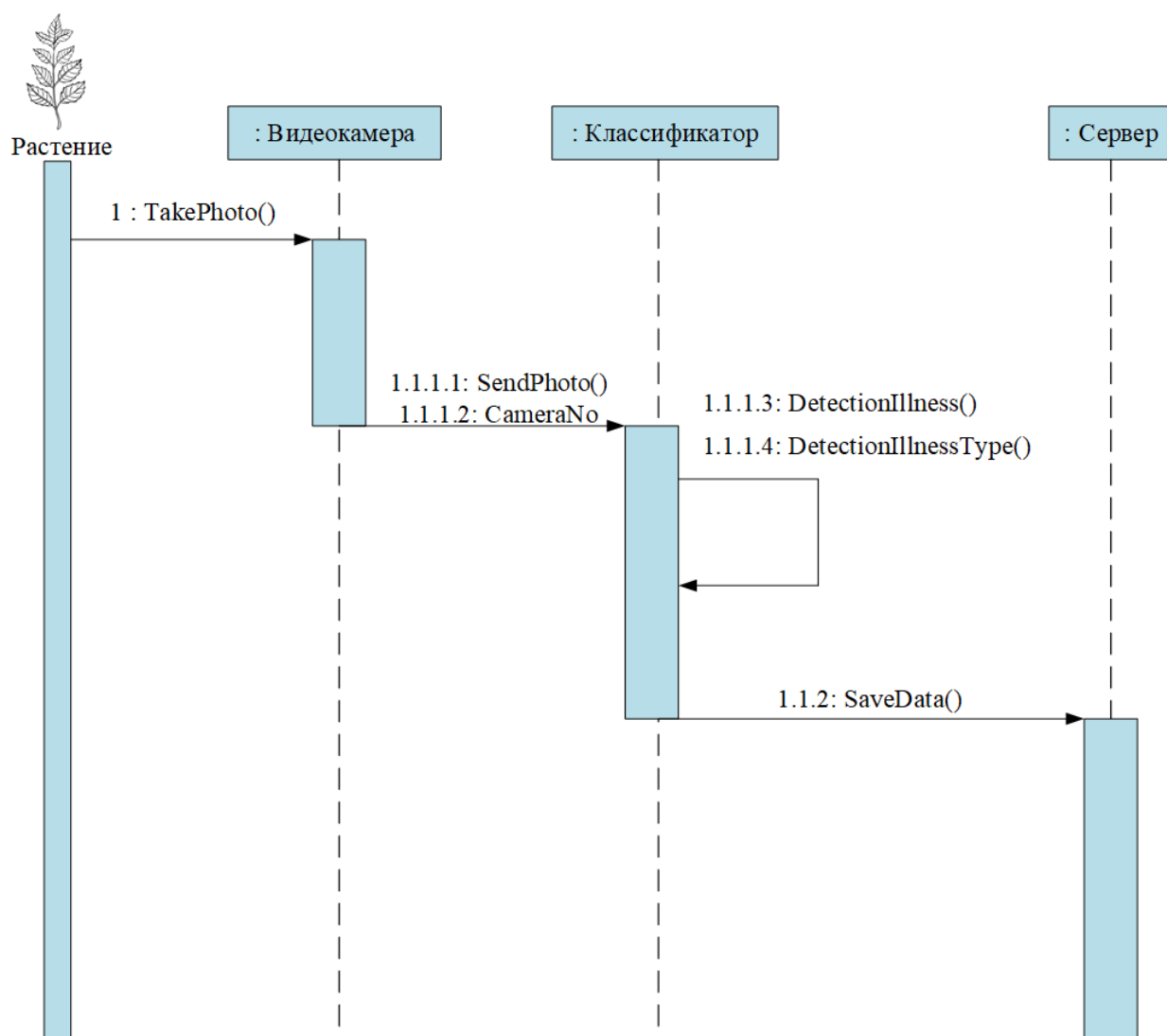
В рамках процесса автоматической регистрации данных о растениях с помощью камеры, классификации и сохранения информации на сервере выполняется инициализация. «Растение» активирует «Видеокамеру», отправляя ей сообщение «TakePhoto()».

Далее происходит инициализация. «Растение» активирует «Видеокамеру», отправляя ей сообщение «TakePhoto()».

После идет сбор данных. «Видеокамера» выполняет запрос, делая снимок (фотографирует растение). Или «Видеокамера» отправляет полученную информацию, включающую фотографию растения, «Классификатору» с помощью сообщений «SendPhoto()» и «CameraNo» (номер камеры).

Далее следует обработка данных. «Классификатор» анализирует полученное фото растения, проверяя на наличие заболеваний, используя методы «DetectionIllness()» и «DetectionIllnessType()», либо «Классификатор» отправляет полученные данные "Серверу" с помощью сообщения «SaveData()».

И завершающий этап это сохранение данных. «Сервер» получает данные от «Классификатора» и сохраняет их в своей базе данных.



**Рисунок 3.6 – Диаграмма последовательности для записи данных по растениям с камеры N на сервер**

Также построены дополнительно две диаграмма последовательности, визуализирующие два других варианта использования, при этом одна диаграмма реализована с помощью средств PlantUML (<https://plantuml.com/ru/sequence-diagram>) и представлена с кодом.

Диаграмма последовательности для получения данных с сервера (Рисунок 3.6) показывает, как пользователь получает данные с сервера через форму. Для данной диаграммы представлен код в листинге 10. Получения данных с сервера выполняется определенная последовательность. «Оператор» выбирает действие «Показать информацию о болезни» в форме, «Форма» отправляет запрос на подключение к серверу, после «Форма» отправляет запрос на получение данных с сервера, «Сервер» выполняет запрос, чтобы получить данные из своей базы

данных, «Сервер» возвращает полученные данные форме, «Форма» выводит полученные данные. И после пользователь получает информацию о болезни.

### Листинг 3.1 – Код диаграммы последовательности из PlantUML

```
@startuml
title "Диаграмма последовательности # 2\ получение данных с сервера"
'Отключение нижних дубликатов
hide footbox
'skinparam sequenceParticipant underline
'autonumber 1.
actor Оператор as A1
participant ": Рабочая форма" as Form
participant ": Сервер" as Server

A1 -> Form++ : 1: DisplayIllnessData()
Form -> Server++ : 2: ConnectToServer()
Form ->(30) Server : 3: GetData()
Form--
Server -> Server++ : 4: SELECT
Server--
Server --> Form-- : 5: exported data
Form++
Form -> Form++ : 7: showData()
Form--
Form --> A1 : 8: data
@enduml
```

Диаграмма последовательности # 2\ получение данных с сервера

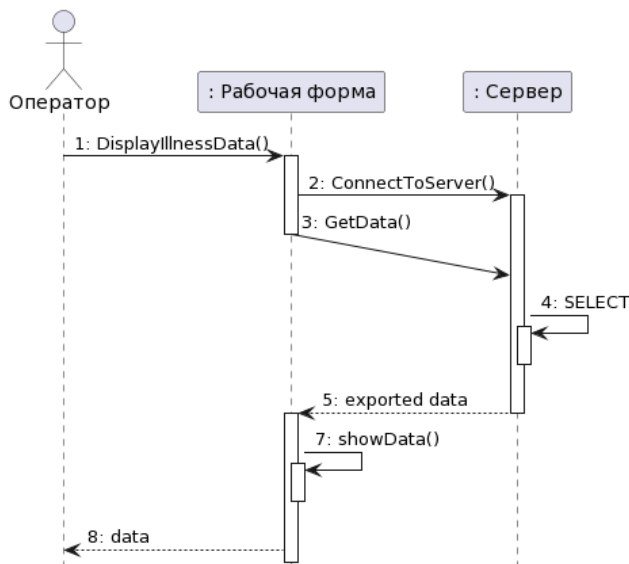


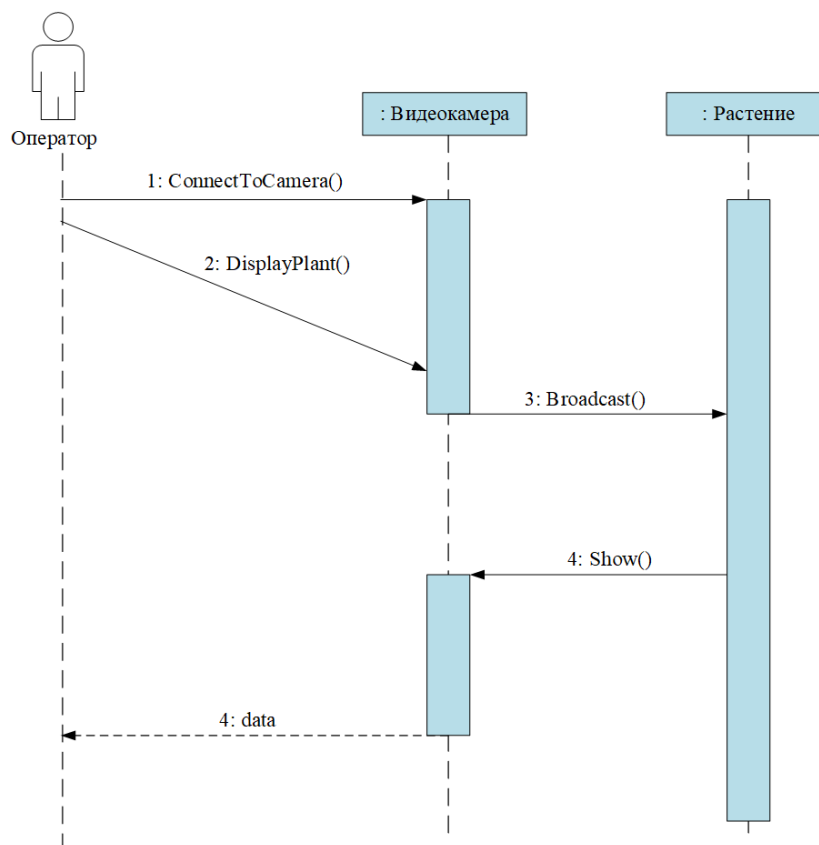
Рисунок 3.744 – Диаграмма последовательности для получения данных с сервера

На рисунок 3.8 представлена диаграмма последовательности для просмотра на растения через камеру N в режиме онлайн. оператор может наблюдать за

растением онлайн с помощью видеокамеры по определенному алгоритму алгоритму.

Сначала «Оператор» устанавливает соединение с видеокамерой, вызывая метод `ConnectToCamera()`, после «Оператор» просит видеокамеру показать изображение растения, вызывая метод `DisplayPlant()`, затем «Видеокамера» транслирует изображение растения методом `Broadcast()`, в свою очередь «Растение», получив команду `Show()`, отправляет видеокамере данные, «Видеокамера» отправляет полученные данные оператору, «Оператор» получает данные и видит изображение растения на своем устройстве.

Этот процесс повторяется постоянно, позволяя оператору наблюдать за растением в режиме онлайн.



**Рисунок 3.8 – Диаграмма последовательности для наблюдением на растение через камеру N в режиме онлайн**

Разработанная диаграмма состояний (Рисунок 3.8) иллюстрирует пошаговую обработку данных, начиная с получения изображения (фото) растения и заканчивая

отправкой результатов на сервер, описывает процесс детекции заболеваний сельскохозяйственных культур с использованием машинного обучения.

Если производить описание диаграммы, то стоит начать с начального состояния, которое включает в себя начало процесса.

Далее система ожидает получение данных в формате изображения. После система ждет получения фото с изображениями растения для анализа.

Когда фото пришло, происходит переход к состоянию «Подготовка данных» при успешном получении изображения.

Ошибка возникает при отсутствии изображения. Переход к состоянию «Отправлено сообщение об ошибке сис. администратору». Затем происходит процесс подготовки данных. На этом этапе проводится предварительная обработка изображения, например, изменение размера, нормализация, перед передачей в модель машинного обучения.

После происходит переход к состоянию «Определение наличия заболевания, его типа» после успешной обработки изображения.

Следующим идет процесс перехода к состоянию «Определение наличия заболевания, его типа» после успешной обработки изображения. На этом этапе может возникнуть ошибка при проблемах с обработкой изображения.

Происходит переход к состоянию «Отправлено сообщение об ошибке сис. администратору»;

Когда нужно определить наличие заболевания или его типа, то первым делом модель машинного обучения анализирует обработанное изображение и определяет, присутствует ли заболевание, и если да, то какой именно тип.

После приходит ответ классификатора. Переход к состоянию «Передача данных на сервер» после получения результатов от модели.

Ошибка возникает при ошибках в работе модели машинного обучения. Переход к состоянию «Отправлено сообщение об ошибке сис. администратору».

Рассмотрим передачу данных на сервер. В первую очередь необходимо описать результаты анализа, включая наличие/отсутствие заболевания, тип и

дополнительную информацию, передаются на сервер для последующего хранения и обработки.

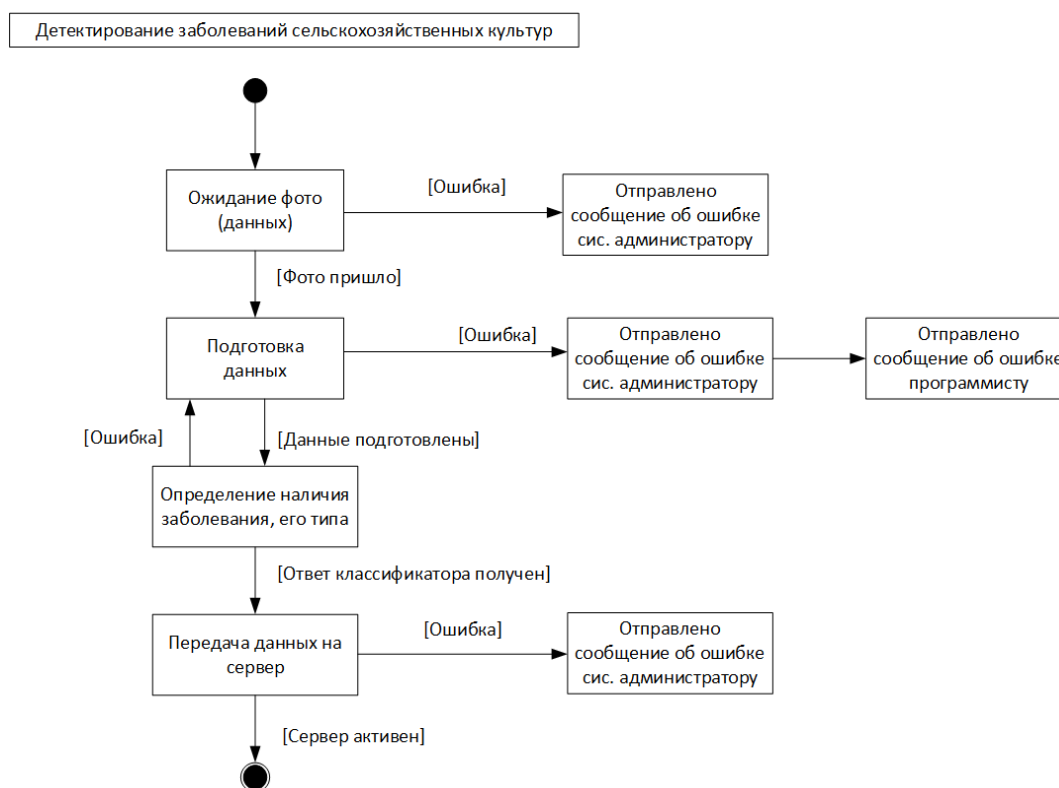
Если сервер подтвердил получение данных, то происходит переход к состоянию «Конец процесса» после успешной передачи данных на сервер.

Ошибка возникает при ошибках в передаче данных на сервер. Переход к состоянию «Отправлено сообщение об ошибке сис. администратору»;

Отправление сообщение об ошибке сис. администратору. В случае ошибки в любом из шагов процесса, отправляется сообщение об ошибке системному администратору для последующей обработки.

В конце процесса происходит переход к состоянию «Конец процесса» после отправки сообщения об ошибке.

Алгоритм завершает работу. Происходит переход к состоянию "Начальное состояние" для начала нового процесса. В случае ошибки отправляется сообщение об ошибке сис. администратору, происходит переход к состоянию «Отправлено сообщение об ошибке сис. администратору» для отправки сообщения об ошибке.



**Рисунок 3.9 – Диаграмма состояний**



### 3.3. Архитектура и реализация сервиса на основе сверточных нейронных сетей (на примере яблони)

Чтобы сделать пользовательский интерфейс (UI), через который пользователи будут загружать изображения растений и получать результаты анализа будут использованы для разработки бэкенда – NestJS.

Nest (NestJS) — это фреймворк для создания эффективных, масштабируемых серверных приложений Node.js. Он использует прогрессивный JavaScript, создан с использованием TypeScript и полностью поддерживает его (но при этом позволяет разработчикам писать код на чистом JavaScript) и сочетает в себе элементы ООП (объектно-ориентированное программирование), ФП (функциональное программирование) и ФРП (функциональное реактивное программирование).

Под капотом Nest использует надежные фреймворки HTTP-серверов, такие как Express (по умолчанию), и при желании может быть настроен на использование Fastify !

Nest обеспечивает уровень абстракции выше общих фреймворков Node.js, но также предоставляет их API напрямую разработчику. Это дает разработчикам свободу использовать множество сторонних модулей, которые доступны для базовой платформы.

Для разработки фронтенда будет использован React - популярная библиотека JavaScript для создания пользовательских интерфейсов. При его помощи можно создать компоненты, которые будут повторно использованы, что поможет сделать код более читаемым [2].

Для упрощения работы со стилями буду использовать Tailwind CSS, он представляет собой фреймворк CSS с предопределенными классами, которые можно использовать для создания и проектирования веб-страниц непосредственно в разметке. Он позволяет писать CSS в HTML в виде предопределенных классов.

Фронтендная часть приложения будет организована в виде отдельных компонентов React, таких как форма загрузки изображений, отображение результатов анализа.



**Рисунок 3.10 – Разработанный прототип приложения в Figma [9]**

Большие объемы изображений и метаданные будут храниться в распределенной файловой системе Amazon S3.

Amazon S3 (Simple Storage Service) представляет собой облачную службу объектного хранения, которая предлагает возможность хранить и извлекать любые объемы данных.

В контексте этого приложения каждое изображение растения может храниться как отдельный объект, а его метаданные (такие как дата, местоположение, размер файла и т. д.) — как соответствующие свойства объекта.

Кроме того, PostgreSQL, реляционная база данных с открытым исходным кодом, которую можно использовать для хранения других данных приложения, например пользовательских данных. PostgreSQL предоставляет широкий спектр функций для хранения, организации и управления структурированными данными. В таблице пользователей можно будет указать различные столбцы, такие как имя пользователя, имя, адрес электронной почты, хэш пароля и другую соответствующую информацию.

Структура таблицы пользователей в PostgreSQL имеет следующий вид:

### Листинг 3.2 – Код для создания таблицы users

```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  username VARCHAR(50) NOT NULL,  
  email VARCHAR(100) NOT NULL,  
  password_hash VARCHAR(100) NOT NULL,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

В данном примере, столбцы таблицы включают id, который является уникальным идентификатором пользователя (создается автоматически с помощью функциональности SERIAL); username, по-русски как имя пользователя; email или электронная почта пользователя; password\_hash или хэш пароля пользователя, который обычно хранится в зашифрованном виде. На стороне бэкенда приложения предлагается использовать argon2 (для него есть библиотека в NestJS); created\_at или просто дата и время создания записи о пользователе (устанавливается по умолчанию на текущее время).

Пользовательские данные могут храниться в user таблице базы данных PostgreSQL, а изображения растений и их метаданные — в распределенной файловой системе, такой как Amazon S3.

Обработка данных изображений растений является важным шагом перед анализом. Она помогает подготовить их к более эффективному изучению.

Изображения растений могут быть разного размера, что может затруднить их обработку и анализ. Поэтому размеры необходимо изменить, чтобы привести их к одному стандартному виду или уменьшить разрешение для более эффективной обработки.

Они могут содержать различные типы шума, ошибки изображения, помехи или искажения. Фильтрация шума удалит эти нежелательные элементы и улучшит качество изображений.

Данные могут быть чувствительны к различным проблемам, таким как слабое освещение, размытость или отсутствие фокуса. Улучшение качества изображения может исправить эти проблемы и сделать изображения более четкими и информативными для анализа.

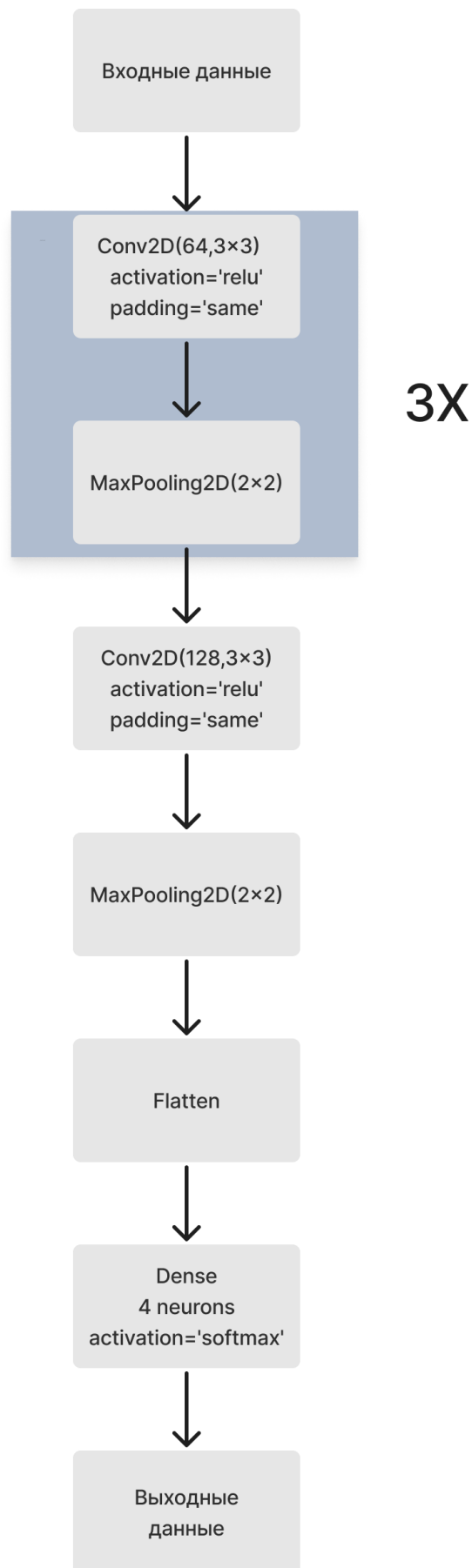
Сегментация фотографий в свою очередь позволяет выделить области или объекты, представляющие интерес на изображении. Например, на фотографии растения можно выделить отдельные листья или цветы.

Применение алгоритмов компьютерного зрения для анализа изображений растений, глубокое обучение нейронных сетей можно использовать для классификации и обнаружения заболеваний на основе визуальных характеристик. Такие модели, как сверточные нейронные сети (CNN), можно обучать на больших наборах данных о заболеваниях растений.

На рисунке 3.11 показана блок-схема каждого уровня предлагаемой модели. На входе находится изображение растения размером 224x224 пикселей и 3 цветовых канала (RGB).

Каждый сверточный слой (Conv2D) использует фильтры для извлечения функций из изображения, а слой понижающей дискретизации (MaxPooling2D) уменьшает размерность данных. После нескольких повторяющихся слоев свертки и понижающей дискретизации данные проходят через слой сглаживания и преобразуются в одномерный вектор.

Затем данные передаются в полностью связанный (плотный) слой с 4 нейронами, который использует функцию активации softmax для прогнозирования вероятности каждого класса. Ниже представлена блок-схема каждого слоя модели. [7].



### **Рисунок 3.11 – Отображение последовательности слоев модели на блок-схеме**

Хранение и анализ результатов анализа изображений растений является важным шагом в выявлении тенденций заболеваний растений, который добудет ценную информацию, ее можно будет использовать для прогнозирования распространения болезни и разработки эффективных стратегий борьбы с заболеваниями.

Также оно играет важную роль в понимании болезней растений и разработке эффективных стратегий борьбы с ними, чтобы выявлять закономерности и тенденции, прогнозировать распространение болезней и разрабатывать стратегии борьбы.

После анализа изображений растений результаты нужно будет сохранить для дальнейшего изучения. Сюда может входить информация о типе заболевания, степени тяжести, площади повреждения, времени визуализации и других характеристиках. Результаты сохраняются в базе данных или файловой системе для последующего доступа.

Для выявления тенденций и тенденций развития болезней растений необходим анализ зафиксированных результатов.

Глубокое изучение продукта анализа поможет предсказать распространение болезней растений. После получится спрогнозировать вероятность возникновения заболевания в конкретных условиях и на конкретной территории.

Масштабируемость и отказоустойчивость безусловно важные факторы при проектировании системы обработки данных, особенно в условиях увеличения потоков данных и необходимости обеспечения непрерывной работы системы.

Горизонтальное масштабирование используется для обработки растущего потока данных, система должна быть способна масштабироваться горизонтально, то есть можно будет добавить дополнительные вычислительные ресурсы, серверы или узлы, для распределения нагрузки и увеличения производительности системы.

Если говорить об управлении ресурсами, то при масштабировании системы необходимо учитывать возможности памяти, хранилища и сеть. Вероятно

потребуется оптимизация кода и алгоритмов для лучшего использования. Также важно учесть возможность автоматического масштабирования системы на основе изменяющейся нагрузки, чтобы обеспечить оптимальную производительность.

Отказоустойчивость – это способность системы продолжать работать без сбоев при выходе из строя одного или нескольких ее компонентов.

Целью создания отказоустойчивой системы является предотвращение сбоев, возникающих из-за единой точки отказа, обеспечение высокой доступности и непрерывности работы критически важных приложений или систем.

Отказоустойчивые системы используют резервные компоненты, которые автоматически заменяют отказавшие компоненты, гарантируя отсутствие потери обслуживания. Такие как аппаратные системы, которые резервируются идентичными или эквивалентными системами.

Системы программного обеспечения, которые резервируются другими экземплярами программного обеспечения. База данных с информацией о клиентах может непрерывно реплицироваться на другую машину. Если основная база данных выходит из строя, операции могут быть автоматически перенаправлены на вторую базу данных.

Источники питания, которые стали отказоустойчивыми с использованием альтернативных источников. Ведь во многих организациях есть генераторы, которые могут взять на себя работу в случае отключения основного электроснабжения.

Аналогичным образом любую систему или компонент, являющийся единой точкой отказа, можно сделать отказоустойчивым с помощью резервирования.

Резервное копирование данных сможет обеспечить отказоустойчивость системы механизмами резервного копирования данных.

Важно иметь механизмы мониторинга работы системы, чтобы быстро обнаруживать проблемы и сбои. При возникновении сбоя или отказа системы необходимо предусмотреть способы автоматического восстановления. Распределенная архитектура системы может быть использована для обеспечения отказоустойчивости.

Интеграция с внешними системами безусловно важный аспект при разработке приложений для обработки данных об активах. С помощью нее можно получить дополнительную информацию, которая поможет вам в анализе и принятии решений.

Интеграция с внешними системами расширит функционал приложения и даст возможность получить дополнительную информацию, которая будет полезна для анализа.

Интеграция с базами данных растений обеспечивает доступ к различным типам растений, таким как признаки, характеристики, классификация и т. д..

Интеграция с службами метеорологической информации предоставляет информацию о погодных условиях, климате и других погодных параметрах, которые могут повлиять на рост и развитие растений.

И если производить интеграцию интерфейсов приложений и веб-сервисов позволяет получать доступ к данным из различных источников, таких как географические данные, данные о почве, климатические данные и многое другое поможет расширить функциональность приложения.

Приложения должны иметь возможность отправлять запросы и получать ответы для взаимодействия с внешними системами и обмена данными. Можно осуществлять интеграцию с использованием различных протоколов связи, таких как RESTful API, SOAP, MQTT и т. д.

Визуализация данных стала важной частью использования бизнес-результатов. С помощью нее пользователь может представить информацию и результаты анализа в удобном и понятном формате.

Визуализация результатов анализа и данных о заболеваниях растений важна для обеспечения простого и понятного пользовательского интерфейса. Красные контуры, цветные шкалы, графики и таблицы, а также интерактивные элементы предоставят возможность пользователю визуализировать результаты анализа и легко интерпретировать данные о состоянии здоровья растений.

Дополнительная информация о некоторых вариантах отображения результатов представляется как один из способов показать результаты — выделить



на изображении больные части растения красным цветом. Таким образом, пользователь может четко видеть, где на растении находится болезнь или повреждение. Эта визуализация быстро выявит проблемные места и привлечет к ним внимание для дальнейшего анализа;

Другой способ визуализировать результаты — использовать цветовую шкалу или тепловую карту. Когда оттенки красного используются для обозначения тяжести заболевания или повреждения растения. Чем сильнее красный цвет, тем серьезнее проблема. Такая визуализация укажет пользователю на отклонения, и он сможет быстро оценить масштабы ущерба и принять необходимые меры.

Также можно представить информацию в виде графиков и таблиц, показывающих различные функции и информацию об установке. Можно создать график, показывающий, как болезни растений меняются с течением времени.

Визуализация результатов бывает интерактивной, когда пользователь взаимодействует с данными и получает более подробную информацию. Будет предложена возможность увеличивать и панорамировать изображение растения для более детального анализа.

В рамках поиска набора данных для определения заболеваний растений с применением технологий Big Data и компьютерного зрения был выбран <https://www.kaggle.com/competitions/plant-pathology-2020-fgvc7> [5, 8].

Набор данных состоит из фотографий и соответствующим им описанию (всего 1821 штук) таких как имя фотографии (`image_id`).

Целевых значения же включают наличие заболевания (да/нет, `healthy`), наличие более одного заболевания (да/нет, `multiple_diseases`), наличие ржавых пятен (да/нет, `rust`), наличие парши (да/нет, `scab`).

### **Листинг 3.3 – Код для отображение части данных**

```
train_df.head()
```

	image_id	healthy	multiple_diseases	rust	scab	label
0	Train_0	0	0	0	1	3
1	Train_1	0	1	0	0	1
2	Train_2	1	0	0	0	0
3	Train_3	0	0	1	0	2
4	Train_4	1	0	0	0	0

**Рисунок 3.12 – Результат выполнения Листинга 2**

Выберем случайные изображения и отобразим их.



**Рисунок 3.13 – Изображения**

Далее требуется обработать данные и провести аугментацию. Аугментация изображений преобразует исходные фото с целью создания новых вариантов информации для обучения модели. Зачастую в наборе данных для изучения может быть ограниченное количество изображений, приводящее к переобучению модели или недостаточной репрезентативности выборки. Что поможет расширить и разнообразить набор данных путем применения различных трансформаций к изображениям.

Аугментация необходима, потому что она увеличивает размер обучающего набора. В наборе менее 2 тысяч экземпляров, это мало для выборки подобных задач. Путем применения случайных преобразований, таких как повороты, сдвиги, масштабирование, отражение и изменение яркости, можно генерировать новые

варианты изображений, увеличивая общее количество доступных образцов для обучения модели. Также она снизит риск переобучения модели путем предоставления ей разнообразных вариантов изображений. Улучшит устойчивость к вариациям, таким как изменения масштаба, освещения, повороты и т.д.

В следующем коде используется объект `ImageDataGenerator` из библиотеки Keras для генерации данных изображений и их аугментации. Чтобы система работала корректно необходимо создать объект `datagen` типа `ImageDataGenerator`, который будет использоваться для обработки и аугментации изображений

После параметр `rescale=1.0/255` укажет на необходимость масштабирования значений пикселей в диапазоне `[0, 1]`.

И параметры `shear_range`, `zoom_range`, `horizontal_flip` и `vertical_flip` определяют типы аугментаций, которые будут применены к изображениям во время обучения модели.

Если рассматривать параметр `validation_split=0.2`, то он указывает на то, что 20% данных будут использоваться для валидации модели.

После создается переменная `Train_DIR`, содержащая полный путь к директории с обучающими изображениями.

Далее создается генератор данных `train_datagen`, который будет поставлять обучающие изображения модели порциями (batch) размером `BATCH_SIZE`. Он использует директорию `Train_DIR` как источник данных и устанавливает размер изображений `target_size=(IMG_SIZE, IMG_SIZE)`. Классы изображений представлены в формате категорий (`class_mode='categorical'`). Генератор использует только подмножество данных для обучения, указанное параметром `subset='training'`.

В свою очередь генератор данных `val_datagen` для валидационных изображений. Он имеет аналогичные параметры, но использует другое подмножество данных для валидации (`subset='validation'`).

Хочется отметить, что объекты `train_datagen` и `val_datagen` предоставляют генераторы данных для обучения и валидации модели, соответственно, с применением аугментации изображений.

### Листинг 3.4 – Код генерации данных изображений и их аугментации

```
datagen=ImageDataGenerator(rescale=1.0/255,
                           shear_range=0.2,
                           zoom_range=0.2,
                           horizontal_flip=True,
                           vertical_flip=True,
                           validation_split=0.2)

Train_DIR = os.path.join(MAIN_PATH, 'train')

train_datagen=datagen.flow_from_directory(Train_DIR,
                                          target_size=(IMG_SIZE, IMG_SIZE),
                                          batch_size=BATCH_SIZE,
                                          class_mode='categorical',
                                          subset='training')
val_datagen=datagen.flow_from_directory(Train_DIR,
                                       target_size=(IMG_SIZE, IMG_SIZE),
                                       batch_size=BATCH_SIZE,
                                       class_mode='categorical',
                                       subset='validation')
```

Определение модели нейронной сети для классификации изображений растений можно представить как изображено в коде.

### Листинг 3.5 – Код сверточной нейронной модели

```
model=Sequential()
model.add(Conv2D(64, (3, 3), activation='relu', padding='same', input_shape=(IMG_SIZE, IMG_SIZE, 3)))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(2, 2))
model.add(Flatten())
model.add(Dense(4, activation='softmax'))
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
```

Если разбирать каждый слой по отдельности, то можно сказать, что в `model=Sequential()` создается объект модели `Sequential`, который позволяет добавлять слои последовательно друг за другом.

`model.add(Conv2D(64,(3,3),activation='relu',padding='same',input_shape=(IMG_SIZE,IMG_SIZE,3)))`. Слой является сверточным слоем (Conv2D). Он использует 64 фильтра размером 3x3 и функцию активации ReLU.

Параметр `padding='same'` означает, что входные данные будут заполнены нулями по краям, чтобы сохранить размерность. `input_shape=(IMG_SIZE,IMG_SIZE,3)` определяет размер входного изображения (IMG\_SIZE x IMG\_SIZE) и количество цветовых каналов (3 - RGB).

`model.add(MaxPooling2D(2,2))`: этот слой является слоем подвыборки (MaxPooling2D). Он уменьшает размерность данных, выбирая максимальное значение из окна размером 2x2.

Последующие слои Conv2D и MaxPooling2D повторяются несколько раз для извлечения более высокоуровневых признаков из изображения. Каждый сверточный слой (Conv2D) использует 64 фильтра размером 3x3 и функцию активации ReLU, а каждый слой подвыборки (MaxPooling2D) уменьшает размерность данных.

`model.add(Flatten())`: этот слой выполняет операцию "сплющивания" данных, преобразуя многомерные данные в одномерный вектор. Он не имеет параметров и используется для подготовки данных перед подачей их в полносвязный слой.

`model.add(Dense(4,activation='softmax'))` полносвязный слой (Dense). У него есть 4 выходных нейрона, соответствующих количеству классов, которые мы хотим предсказать. Функция активации softmax используется для получения вероятностного распределения по классам.

Полученная модель нейронной сети состоит из нескольких сверточных слоев для извлечения признаков из изображений, слоев подвыборки для уменьшения размерности данных, слоя Flatten для преобразования данных в одномерный вектор и полносвязного слоя с функцией активации softmax для классификации на 4 класса.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_1 (Conv2D)	(None, 112, 112, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2d_2 (Conv2D)	(None, 56, 56, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 4)	100356

=====  
Total params: 249,860  
Trainable params: 249,860  
Non-trainable params: 0  
=====

**Рисунок 3.14 – Слои модели**

Обучим модель, у которой будут размер партии 16, количество итераций модели 50. В качестве оптимизатора был выбран Adam, который является одним из самых эффективных алгоритмов оптимизации в обучении нейронных сетей.

**Листинг 3.6 – Обучение модели**

```
IMG_SIZE = 224
BATCH_SIZE = 16
EPOCHS = 50
history = model.fit(train_datagen,
validation_data=val_datagen,
epochs=EPOCHS,
steps_per_epoch=train_datagen.samples//BATCH_SIZE,
validation_steps=val_datagen.samples//BATCH_SIZE,
callbacks=callbacks)
```

Описание аргументов метода fit():

- `train_datagen` представляет собой генератор данных для обучения модели, предоставляет для модели пакеты данных каждой эпохи обучения.
- `validation_data` является генератором данных для валидации модели, предоставляет модели пакеты данных для оценки ее производительности на отложенной выборке.

- `epochs` или количество эпох обучения модели. Эпоха - это один проход по всем обучающим данным.
- `steps_per_epoch` или количество шагов, выполняемых в каждой эпохе обучения. Обычно это количество образцов обучающих данных, деленное на размер пакета (`BATCH_SIZE`).
- `validation_steps` представляется в виде количество шагов (пакетов данных), выполняемых в каждой эпохе валидации. Обычно это количество образцов валидационных данных, деленное на размер пакета (`BATCH_SIZE`).
- `callbacks` является списком обратных вызовов, которые будут применяться во время обучения модели. Они могут выполнять различные действия, такие как сохранение модели, ранняя остановка обучения или изменение параметров обучения.

```
Epoch 1/50
91/91 [=====] - ETA: 0s - loss: 1.2757 - accuracy: 0.3370
Epoch 1: val_loss improved from inf to 1.25146, saving model to ./best_weights.h5
91/91 [=====] - 384s 4s/step - loss: 1.2757 - accuracy: 0.3370 - val_loss: 1.2515 - val_accuracy: 0.3324
Epoch 2/50
91/91 [=====] - ETA: 0s - loss: 1.2486 - accuracy: 0.3141
Epoch 2: val_loss did not improve from 1.25146
91/91 [=====] - 373s 4s/step - loss: 1.2486 - accuracy: 0.3141 - val_loss: 1.2676 - val_accuracy: 0.2784
Epoch 3/50
91/91 [=====] - ETA: 0s - loss: 1.2554 - accuracy: 0.3287
Epoch 41/50
91/91 [=====] - ETA: 0s - loss: 0.1300 - accuracy: 0.9570
Epoch 41: val_loss did not improve from 0.21871
Restoring model weights from the end of the best epoch: 31.
91/91 [=====] - 355s 4s/step - loss: 0.1300 - accuracy: 0.9570 - val_loss: 0.2542 - val_accuracy: 0.9233
Epoch 41: early stopping
```

### Рисунок 3.15 – Результат работы кода по обучению модели

В результате обучения модели на 41-й эпохе были получены следующие метрики:

- Loss (потеря): 0.1300
- Accuracy (точность): 0.9570

### Листинг 3.7 – Отображение результатов обучения

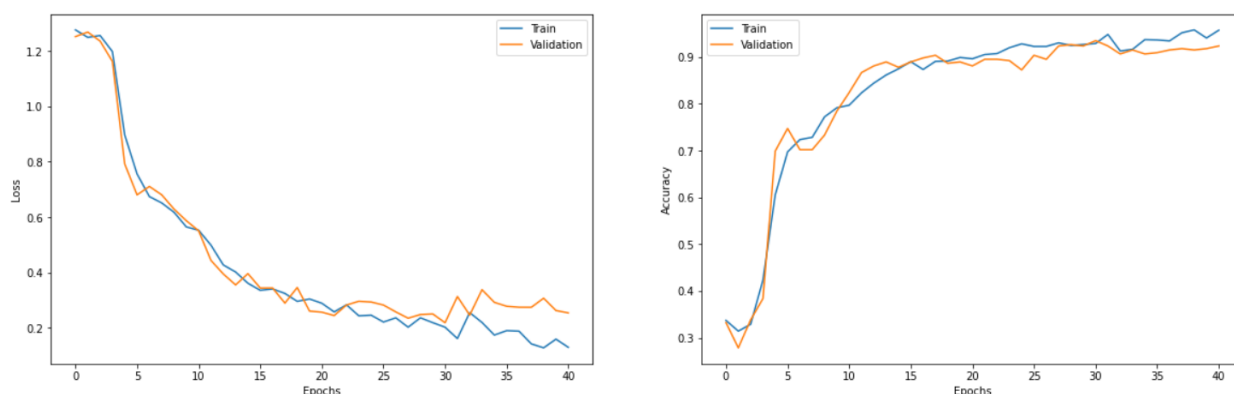
```
from sklearn.metrics import roc_auc_score,
precision_recall_fscore_support

plt.figure(1, figsize = (20, 6))

plt.subplot(121)
plt.plot(history.history['loss'], label = 'Train')
plt.plot(history.history['val_loss'], label = 'Validation')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
```

```
plt.subplot(122)
plt.plot(history.history['accuracy'], label = 'Train')
plt.plot(history.history['val_accuracy'], label = 'Validation')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.show()
```



**Рисунок 3.16 – Отображение результатов обучения**

Для распознавания заболеваний на листьях яблони была написана функция `check_apple_leaf()`, включающая следующие шаги:

Для начала, формируется полный путь к изображению, используя переменную `image_name` и константу `IMAGE_DIR`. Затем происходит открытие изображения с помощью `Image.open()` и сохранение его в переменную `image_result`.

Далее, происходит загрузка и преобразование тестового изображения с использованием методов `image.load_img()`, `image.img_to_array()`, а также нормализации значений пикселей до диапазона `[0, 1]`.

Для соответствия ожидаемому формату модели происходит расширение размерности тестового изображения, после чего к нему применяется обученная модель с помощью метода `predict()` для получения предсказания категории.

Затем создается список категорий `Categories`, содержащий метки классов, и изображение отображается с помощью `plt.imshow()`.

Заголовок графика устанавливается с использованием метки категории, соответствующей наиболее вероятному классу (`Categories[np.argmax(result)]`), после чего график отображается с помощью `plt.show()`.



Таким образом, функция `check_apple_leaf()` осуществляет загрузку и анализ изображения листа яблони с применением предварительно обученной модели, предоставляя результат в виде графика с указанием предсказанной категории.

### Листинг 3.8 – Функцию для распознавания заболевания

```
def check_apple_leaf(image_name):
    DIR = os.path.join(IMAGE_DIR, image_name)
    image_result = Image.open(DIR)

    test_image = image_result
    image.load_img(DIR, target_size=(IMG_SIZE, IMG_SIZE))
    test_image = image.img_to_array(test_image)
    test_image = test_image/255
    test_image = np.expand_dims(test_image, axis=0)
    result = model.predict(test_image)

    Categories = ['healthy', 'multiple_disease', 'rust', 'scab']
    image_result = plt.imshow(image_result)
    plt.title(Categories[np.argmax(result)])
    plt.show()
```

### Листинг 3.9 – Распознавание заболевания

```
check_apple_leaf('Test_1002.jpg')
```

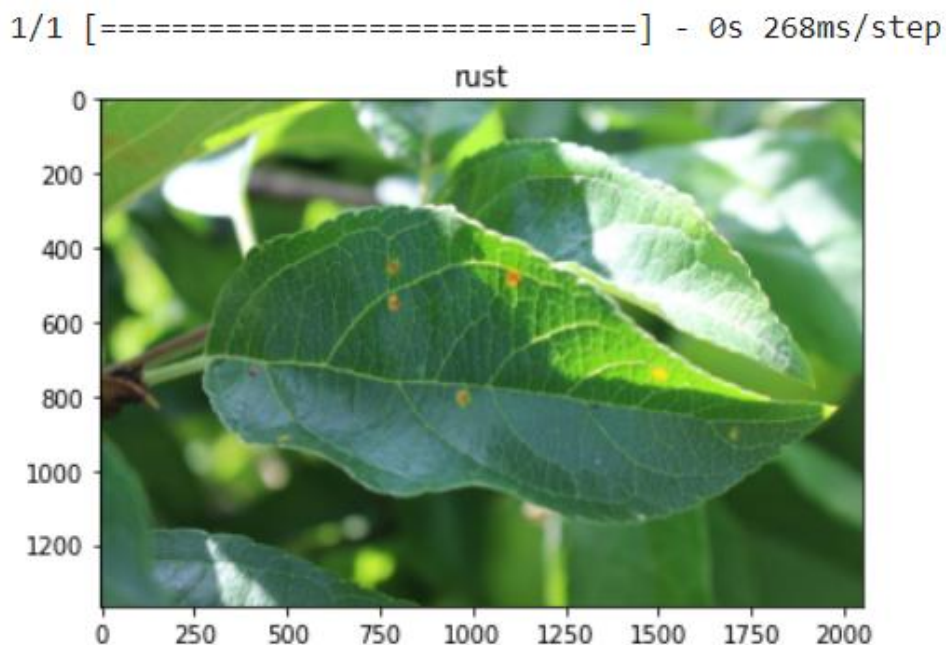
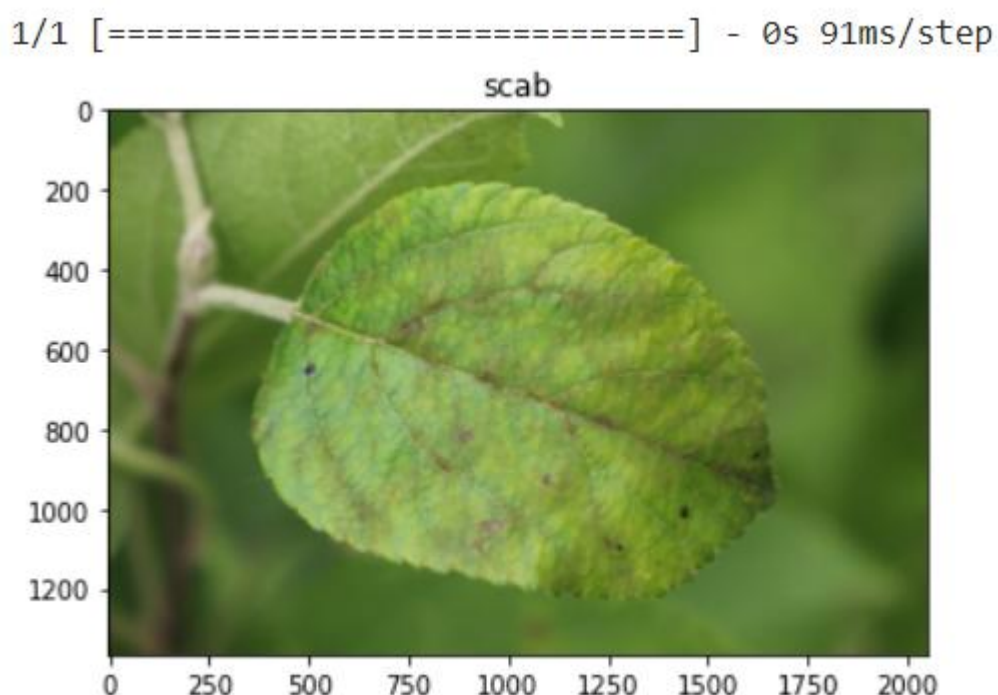


Рисунок 3.17 – Результат распознавания

### Листинг 3.10 - Распознавание заболевания

```
check_apple_leaf('Test_1018.jpg')
```



**Рисунок 3.18 – Результат распознавания**

По рисункам 3.17-3.18 видно, что модель верно распознала наличие ржавых пятен и парши соответственно. Также численные показатели эффективности модели говорят о ее качественности и пригодности к использованию.

Важно отметить, что разработанное приложение является лишь одним из примеров применения технологий Big Data и компьютерного зрения в сельском хозяйстве. Эти технологии имеют широкий потенциал и могут быть использованы для решения других задач, связанных с растениеводством, таких как определение уровня урожайности, контроль качества почвы и др.

### **Библиографический список к главе 3**

1. Бенджио Ио., Гудфеллоу Я., Курвилль А. Глубокое обучение. М.: ДМК Пресс, 2018. 652 с.
2. Бэнкс А., Порселло Е. React и Redux: функциональная веб-разработка — СПб.: Питер, 2018. — 336 с
3. Душечкина Е.А., Костромин Н.С., Паня А., Сивова А.Н. Сортировка отходов с помощью нейросетевой модели как метод предупреждения пожаров - Материалы Молодежной программы 25-ой Международной специализированной

выставки и Форума «Безопасность и охрана труда» БИОТ-2021. Москва, 2021. – 25-28сс.

4. Дьяков Ю. Т., Еланский С.Н. Общая фитопатология – М. : Юрайт, 2017.- 230 с.

5. Костромин Н.С., Сивова А.Н. Перспективы применения нейросетей для решения проблем ННН-рыболовства и пиратства в Арктической зоне России // Российская Арктика. 2020. № 11. С. 24–30.

6. Новиков Ф.А. Анализ и проектирование на UML. 2007.; Фаулер М. UML. Основы, 3е издание. – Пер. с англ. – СПб: СимволПлюс, 2004. – 192 с.

7. Рыбаков А. В., Выборнов Н. А., Рыбаков И. А. Анализ методов компьютерного зрения, перспективных для применения в агропромышленном комплексе // Прикаспийский журнал: управление и высокие технологии. 2022. №1 (57).

8. Сивова А. Н., Ровина В. О. Распознавание заболеваний сельскохозяйственных культур. Труды 65-й Всероссийской научной конференции МФТИ в честь 115-летия Л. Д. Ландау, 3–8 апреля 2023 г. Прикладная математика и информатика. — М: Физматкнига, 2023. — 286 с.

9. Сивова А. Н., Дашиева Б. Ш. Разработка приложения для определения заболеваний растений с применением технологий Big Data и компьютерного зрения. Материалы Международной научной конференции молодых учёных и специалистов, посвящённой 150-летию со дня рождения А.Я. Миловича, г. Москва, 3-5 июня 2024 г.: сборник статей. Том 1 – Москва: Издательство РГАУ - МСХА, 2024

10. Флегонтов А.В. Моделирование информационных систем. Unified Modeling Language [Электронный ресурс]: учеб. пособие / А.В. Флегонтов, И.Ю. Матюшичев. — Электрон. дан. — Санкт-Петербург: Лань, 2018. — 112 с. — Режим доступа: <https://e.lanbook.com/book/102244>.

## **ПРИЛОЖЕНИЯ**

## Приложение 1

Settings.py (основные настройки)

```
import os
from datetime import timedelta

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

SECRET_KEY = 'ваш-secret-key'

DEBUG = True

ALLOWED_HOSTS = ['*']

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'rest_framework_simplejwt',
    'corsheaders',
    'django_filters',
    'core',
    'products',
    'news',
    'users',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'config.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
```

```

        'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
            'django.contrib.messages.context_processors.messages',
        ],
    },
]

```

```
WSGI_APPLICATION = 'config.wsgi.application'
```

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'organic_db',
        'USER': 'organic_user',
        'PASSWORD': 'organic_pass',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}

```

```

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

```

REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    ),
    'DEFAULT_FILTER_BACKENDS': (
        'django_filters.rest_framework.DjangoFilterBackend',
    ),
    'DEFAULT_PAGINATION_CLASS': 'rest_framework.pagination.PageNumberPagination',
    'PAGE_SIZE': 20
}

```

```

SIMPLE_JWT = {
    'ACCESS_TOKEN_LIFETIME': timedelta(minutes=60),
    'REFRESH_TOKEN_LIFETIME': timedelta(days=1),
}

```

```

}

CORS_ALLOW_ALL_ORIGINS = True

LANGUAGE_CODE = 'ru-ru'
TIME_ZONE = 'Europe/Moscow'
USE_I18N = True
USE_L10N = True
USE_TZ = True

STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static')

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

AUTH_USER_MODEL = 'users.User'

```

users/models.py (пользователи)

```

from django.contrib.auth.models import AbstractUser
from django.db import models

class User(AbstractUser):
    USER_TYPE_CHOICES = (
        (1, 'Admin'),
        (2, 'Moderator'),
        (3, 'User'),
    )

    user_type = models.PositiveSmallIntegerField(choices=USER_TYPE_CHOICES, default=3)
    phone = models.CharField(max_length=20, blank=True)
    address = models.TextField(blank=True)

```

```

    def __str__(self):
        return self.username

```

products/models.py (товары и заказы)

```

from django.db import models
from users.models import User

class Category(models.Model):
    name = models.CharField(max_length=100)
    slug = models.SlugField(unique=True)

    def __str__(self):
        return self.name

class Product(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE)
    name = models.CharField(max_length=200)
    description = models.TextField()

```

```

price = models.DecimalField(max_digits=10, decimal_places=2)
organic_certificate = models.BooleanField(default=False)
image = models.ImageField(upload_to='products/')
created_at = models.DateTimeField(auto_now_add=True)
updated_at = models.DateTimeField(auto_now=True)

```

```

def __str__(self):
    return self.name

```

```

class Order(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    products = models.ManyToManyField(Product, through='OrderItem')
    total_price = models.DecimalField(max_digits=10, decimal_places=2)
    created_at = models.DateTimeField(auto_now_add=True)
    status = models.CharField(max_length=50, default='Processing')

    def __str__(self):
        return f"Order #{self.id}"

```

```

class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)
    price = models.DecimalField(max_digits=10, decimal_places=2)

    def __str__(self):
        return f"{self.product.name} x {self.quantity}"

```

news/models.py (парсер для soz.bio)

```

import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin
from django.utils import timezone
from .models import NewsItem

```

```

def parse_soz_bio():
    BASE_URL = "https://soz.bio"
    NEWS_URL = urljoin(BASE_URL, "/news/") # Основная страница новостей

    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36',
        'Accept-Language': 'ru-RU,ru;q=0.9',
    }

    try:
        # Загружаем страницу с новостями
        response = requests.get(NEWS_URL, headers=headers, timeout=15)
        response.raise_for_status() # Проверка на ошибки HTTP

        soup = BeautifulSoup(response.text, 'lxml') # Используем lxml для быстрого парсинга

```



```

# Ищем все новостные блоки
news_blocks = soup.find_all('div', class_='post')

if not news_blocks:
    print("На странице не найдено новостей")
    return False

for item in news_blocks:
    try:
        # Извлекаем заголовок
        title_tag = item.find('h2', class_='post-title')
        title = title_tag.get_text(strip=True) if title_tag else "Без заголовка"

        # Извлекаем ссылку на полную новость
        link = urljoin(BASE_URL, title_tag.find('a')['href']) if title_tag and title_tag.find('a') else
NEWS_URL

        # Извлекаем основной текст
        content_div = item.find('div', class_='post-content')
        if content_div:
            # Объединяем все параграфы в один текст
            content = '\n'.join(p.get_text(strip=True) for p in content_div.find_all('p'))
        else:
            content = "Содержание недоступно"

        # Извлекаем дату публикации (если есть)
        meta_div = item.find('div', class_='post-meta')
        if meta_div and meta_div.find('time'):
            published_at = meta_div.find('time')['datetime'] # ISO-формат
        else:
            published_at = timezone.now()

        # Сохраняем в базу данных
        NewsItem.objects.update_or_create(
            title=title,
            defaults={
                'content': content,
                'source_url': link,
                'published_at': published_at,
                'is_parsed': True # Флаг, что новость спарсена
            }
        )

    except Exception as e:
        print(f"Ошибка при обработке новости: {str(e)}")
        continue

return True

except requests.exceptions.RequestException as e:
    print(f"Ошибка при запросе к {NEWS_URL}: {str(e)}")
    return False

```

```
except Exception as e:
    print(f'Неожиданная ошибка: {str(e)}')
    return False
```

Запуск через Celery (tasks.py)

```
from celery import shared_task
from .parsers import parse_soz_bio
```

```
@shared_task
def parse_soz_bio_task():
    result = parse_soz_bio()
    return {
        'status': 'success' if result else 'failed',
        'timestamp': str(timezone.now())
```

users/serializers.py (API )

```
from rest_framework import serializers
from django.contrib.auth import get_user_model
from rest_framework_simplejwt.serializers import TokenObtainPairSerializer
```

```
User = get_user_model()
```

```
class UserSerializer(serializers.ModelSerializer):
    class Meta:
        model = User
        fields = ['id', 'username', 'email', 'user_type', 'phone', 'address']
        extra_kwargs = {'password': {'write_only': True}}

    def create(self, validated_data):
        user = User.objects.create_user(
            username=validated_data['username'],
            email=validated_data.get('email', ''),
            password=validated_data['password'],
            user_type=validated_data.get('user_type', 3)
        )
        return user
```

```
class CustomTokenObtainPairSerializer(TokenObtainPairSerializer):
    @classmethod
    def get_token(cls, user):
        token = super().get_token(user)
        token['user_type'] = user.user_type
        return token
```

recommendations/utils.py (рекомендательная система)

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```

from sklearn.preprocessing import MinMaxScaler
from django.db.models import Count, Q, F
from django.core.cache import cache
from products.models import Product, OrderItem, Review
from datetime import timedelta
import json
from scipy.sparse import hstack

class OrganicProductRecommender:
    def __init__(self):
        self.vectorizer = TfidfVectorizer(
            stop_words=['organic', 'bio', 'eco'],
            ngram_range=(1, 2),
            max_features=5000
        )
        self.scaler = MinMaxScaler()
        self.feature_weights = {
            'purchase_count': 0.35,
            'discount': 0.25,
            'rating': 0.2,
            'novelty': 0.1,
            'text_similarity': 0.1
        }

    def _get_product_data(self):
        """Сбор и подготовка данных о продуктах"""
        date_threshold = timezone.now() - timedelta(days=90)

        products = Product.objects.annotate(
            purchase_count=Count(
                'order_items',
                filter=Q(order_items__order__created_at__gte=date_threshold)
            ),
            review_count=Count('reviews'),
            avg_rating=Coalesce(F('average_rating'), 3.0),
            is_discounted=Q(discount_price__lt=F('price'))
        ).select_related('category').prefetch_related('reviews')

        return list(products)

    def _prepare_features(self, products):
        """Подготовка признаков для рекомендаций"""
        # Текстовые признаки
        text_data = [f"{p.name} {p.category.name} {p.organic_certificate_id}" for p in products]
        text_features = self.vectorizer.fit_transform(text_data)

        # Числовые признаки
        numeric_data = np.array([
            [p.purchase_count,
             1 if p.is_discounted else 0,
             p.avg_rating,
             (timezone.now() - p.created_at).days]

```

```

        for p in products
    ])

    # Нормализация
    numeric_features = self.scaler.fit_transform(numeric_data)

    # Комбинированные признаки
    combined_features = hstack([text_features, numeric_features])

    return combined_features

def _calculate_similarity(self, features):
    """Вычисление матрицы схожести"""
    # Косинусная схожесть для текстовых признаков
    text_sim = cosine_similarity(features[:, :self.vectorizer.max_features])

    # Евклидова метрика для числовых признаков
    numeric_sim = 1 - pairwise_distances(
        features[:, self.vectorizer.max_features:].toarray(),
        metric='euclidean'
    )

    # Комбинированная схожесть
    combined_sim = (
        text_sim * self.feature_weights['text_similarity'] +
        numeric_sim * (1 - self.feature_weights['text_similarity'])
    )

    return combined_sim

def get_popular_products(self, num=8):
    """Рекомендации популярных товаров (статистический подход)"""
    cache_key = f'popular_organic_products_{num}'
    cached = cache.get(cache_key)

    if cached:
        return json.loads(cached)

    products = self._get_product_data()
    features = self._prepare_features(products)

    # Вычисление популярности
    popularity_scores = []
    for i, p in enumerate(products):
        score = (
            p.purchase_count * self.feature_weights['purchase_count'] +
            (5 if p.is_discounted else 0) * self.feature_weights['discount'] +
            p.avg_rating * self.feature_weights['rating'] +
            (1 if (timezone.now() - p.created_at).days < 30 else 0) * self.feature_weights['novelty']
        )
        popularity_scores.append((i, score))

```

```

# Топ-N товаров
top_indices = sorted(popularity_scores, key=lambda x: x[1], reverse=True)[:num]
result = [products[i] for i, _ in top_indices]

cache.set(cache_key, json.dumps(result), 3600)
return result

def get_content_based_recommendations(self, product_id, num=5):
    """Контентные рекомендации на основе схожести товаров"""
    products = self._get_product_data()
    features = self._prepare_features(products)
    sim_matrix = self._calculate_similarity(features)

    product_indices = {p.id: i for i, p in enumerate(products)}
    idx = product_indices.get(product_id)

    if idx is None:
        return []

    sim_scores = list(enumerate(sim_matrix[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Исключаем исходный товар и выбираем топ-N
    top_indices = [i for i, _ in sim_scores[1:num+1]]
    return [products[i] for i in top_indices]

def get_hybrid_recommendations(self, user=None, num=10):
    """Гибридные рекомендации (статистика + контент)"""
    # Популярные товары
    popular = self.get_popular_products(num//2)

    # Персонализированные (если пользователь авторизован)
    personalized = []
    if user and hasattr(user, 'product_views'):
        viewed_products = user.product_views.all()[:3]
        for p in viewed_products:
            personalized.extend(
                self.get_content_based_recommendations(p.product_id, num//3)
            )

    # Смешивание результатов
    recommendations = list(set(popular + personalized))
    return sorted(
        recommendations,
        key=lambda x: (
            x.purchase_count * 0.4 +
            (1 if x.is_discounted else 0) * 0.3 +
            x.avg_rating * 0.2 +
            (1 if (timezone.now() - x.created_at).days < 30 else 0) * 0.1
        ),
        reverse=True
    )[:num]

```

## Приложение 2



Рисунок 1 – Диаграмма IDEF3

### Приложение 3

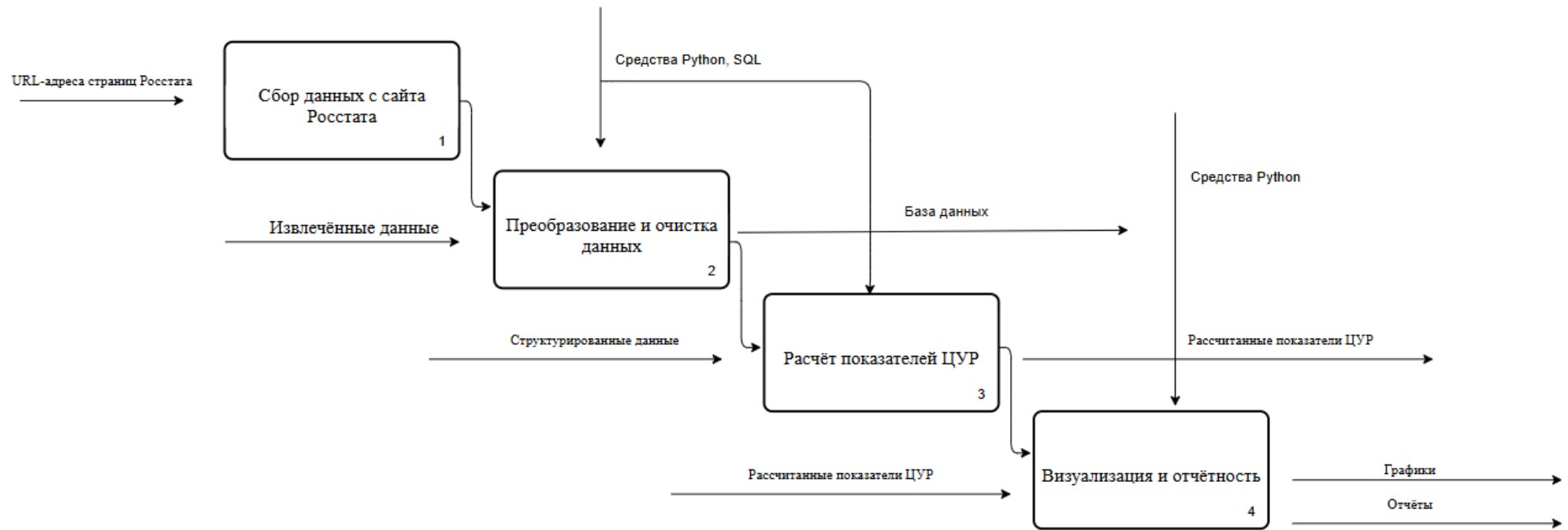


Рисунок 2 – Диаграмма IDEF0

## Приложение 4

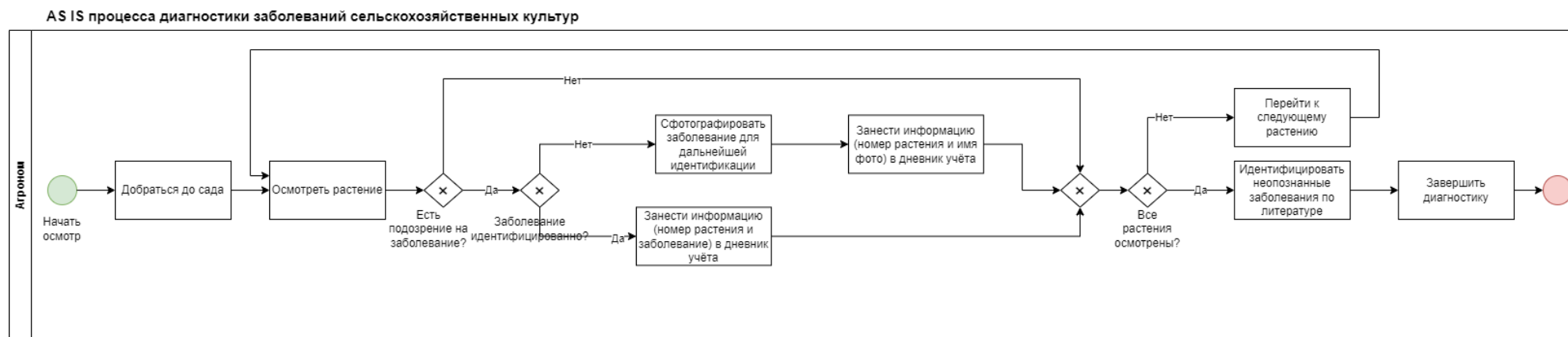


Рисунок 3 – AS IS процесса диагностики заболевания сельскохозяйственных культур



## Приложение 5

ТО ВЕ процесса диагностики заболеваний сельскохозяйственных культур

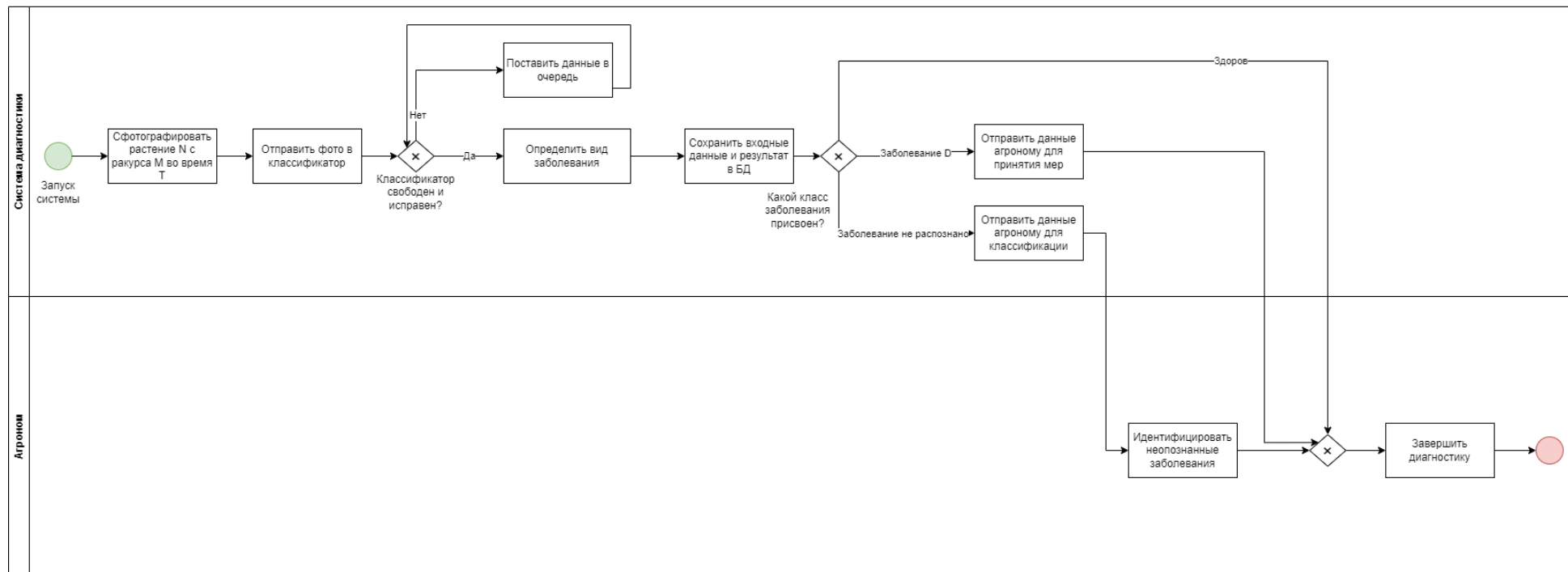


Рисунок 4 – ТО ВЕ процесса диагностики заболевания сельскохозяйственных культур

Научное издание

Хоружий Людмила Ивановна  
Уколова Анна Владимировна  
Джикия Мэри Константиновна  
Дашиева Баярма Шагдаровна  
Семенова Софья Олеговна  
Ветошкин Артем Юрьевич  
Титов Артем Денисович  
Титова Дарья Алексеевна  
Быков Денис Витальевич  
Сивова Александра Николаевна

## **ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ В АПК**

Коллективная монография

Издается в авторской редакции

Компьютерный набор А.Д. Титов

Подписано к изданию 22.12.2025.

Объем данных 5,65 Мб.

Тираж 10 экз.

ФГБОУ ВО РГАУ – МСХА имени К.А. Тимирязева  
127434 Москва, ул. Тимирязевская, 49