Министерство образования Ярославской области
Государственное профессиональное образовательное
автономное учреждение Ярославской области
«ЯРОСЛАВСКИЙ ПРОМЫШЛЕННО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ
им. Н.П. ПАСТУХОВА»

УТВЕРЖДАЮ
Зам. директора по УПР
_____ [redacted]
«____»_____20____г.

ДИПЛОМНЫЙ ПРОЕКТ

Разработка веб-системы для учёта, инвентаризации и управления жизненным циклом оборудования предприятия

(тема дипломного проекта) Пояснительная записка

ДП 09.02.07.22ИПВ.02.00.00 ПЗ

Исполнитель
_____ Р.Н. Бальчонок
«____»_____20____г.

Руководитель проекта
_____ [redacted]
«____»_____20____г.

Заведующий кафедрой
_____ [redacted]
«____»_____20____г.

2026

ГПОАУ ЯО «Ярославский промышленно-экономический колледж им. Н.П. Пастухова»

УТВЕРЖДАЮ
Зам. директора по УМР
_____ [redacted]
"____" _____ 2026 г.

ЗАДАНИЕ

на дипломный проект

ДП. 09.02.07.22ИПВ.02

Студенту Бальчонку Роману Николаевичу
(фамилия, имя, отчество)

1. Тема дипломного проекта Разработка веб-системы для учёта, инвентаризации и управления жизненным циклом оборудования предприятия
2. Исходные данные
   – ГОСТ 34.320-96 Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.
   – ГОСТ 34.602-2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
   – ГОСТ Р 57193-2025 Системная и программная инженерия. Процессы жизненного цикла систем.
   – ГОСТ Р 59792-2021 Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем.
   – ГОСТ Р 59793-2021 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
   – ГОСТ Р 59795-2021 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
   – ГОСТ Р 59853-2021 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
   – ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств.
   – СТО ЯПЭК СМК 02-4.2.2-4-06-26.
   – Проектирование и разработка информационных систем: учебник О.Н. Перлова, О.П. Ляпина, А.В. Гусева. — 4-е изд., испр. и доп.— Москва: Издательский центр «Академия», 2023. — 256с.
   – Федорова Г.Н. Осуществление интеграции программных модулей: учебник. Изд. Academia. Среднее профессиональное образование. 2023. – 272 с.
   – Федорова Г.Н. Сопровождение информационных систем: учебник для студ. учреждений сред.проф. образования – М.: Издательский центр «Академия», 2022. – 318 с.

3. Содержание пояснительной записки (перечень вопросов, подлежащих разработке)

Введение

1. Теоретическая часть

Глава 1. Анализ предметной области.

Глава 2. Анализ аналогов и прототипов сайта (веб-приложения) данной предметной области.

Глава 3. Технико-экономическое обоснование создания сайта (веб-приложения).

3.1. Резюме и идея проекта;

3.2. Общее описание задачи;

3.3. Составление плана работ в программе MS Project.

Глава 4. Разработка первичной документации.

Глава 5. Техника безопасности.

2. Практическая часть.

Глава 1. Проектирование сайта (веб-приложения).

Глава 2. Реализация физической модели.

Глава 3. Организация безопасности сайта (веб-приложения).

Глава 4. Тестирование и проверка работоспособности.

Глава 5. Аппаратные и программные средства для функционирования

Глава 6. Инструкции пользователя и администратора сайта (веб-приложения).

Заключение.

Список использованных источников.

Приложения.

4. Содержание дополнительной части (с указанием обязательных материалов)

–файлы с компонентами сайта (веб-приложения) на CD, DVD, Flash-накопителях.

Дата выдачи задания «08» апреля 2026г.

Срок сдачи дипломного проекта «13» июня 2026г.

Заведующий кафедрой _____ [redacted]
подпись (Ф.И.О.)

Руководитель дипломного проекта _____ [redacted]
подпись (Ф.И.О.)

Задание принял к исполнению студент _____ Бальчонок Р.Н.
подпись (Ф.И.О.)

Примечание:

1.В конце пояснительной записки должен быть приведен список использованных источников (книг, стандартов и других нормативных документов, интернет-сайтов и т.д.).

2.Объем пояснительной записки не должен выходить за пределы 50–70 страниц, не считая приложений.

3.Форма задания может быть изменена в соответствии с содержанием дипломного проекта.

Содержание

Введение....4
1. Теоретическая часть....6
Глава 1. Анализ предметной области....6
Глава 2. Анализ аналогов и прототипов сайта (веб-приложения) данной предметной области....11
Глава 3. Технико-экономическое обоснование создания сайта....18
3.1 Резюме и идея проекта....18
3.2 Общее описание задачи....20
3.3 Составление плана работ....23
Глава 4. Разработка первичной документации....25
Глава 5. Техника безопасности....28
2. Практическая часть....30
Глава 1. Проектирование сайта (веб-приложения)....30
Глава 2. Реализация физической модели....36
Глава 3. Организация безопасности сайта (веб-приложения)....50
Глава 4. Тестирование и проверка работоспособности....60
Глава 5. Аппаратные и программные средства для функционирования....68
Глава 6. Инструкции пользователя и администратора системы....70
Заключение....71
Список использованных источников....72
Приложение А. Руководство пользователя....75
Приложение Б. Руководство по стилю....99
Приложение В. Бриф на разработку сайта....101
Приложение Г. Техническое задание на разработку сайта....105
Приложение Д. План работ и диаграмма Ганта....112
Приложение Е. Диаграмма IDEF0....114
Приложение Ж. Диаграмма вариантов использования....117
Приложение З. Диаграмма IDEF1X....118
Приложение И. Диаграмма диаграмма последовательностей....119
Приложение К. Общая концепция....122
Приложение Л. Спецификация для диаграммы вариантов использования....129
Приложение М. Глоссарий к диаграмме IDEF0....135

Введение

Тема дипломного проекта — разработка веб-системы для учёта, инвентаризации и управления жизненным циклом оборудования предприятия.

Актуальность разработки обусловлена необходимостью повышения эффективности управления материально-техническими ресурсами организации. В условиях современного предприятия оборудование является одним из ключевых активов, от состояния, доступности и правильного распределения которого напрямую зависит стабильность рабочих процессов. При отсутствии централизованной системы учёта возникают проблемы, связанные с потерей информации о местонахождении оборудования, дублированием данных, ошибками в статусах объектов и затруднениями при проведении инвентаризации и формировании отчётности. Использование информационных технологий позволяет устранить указанные недостатки и обеспечить прозрачность операций с оборудованием.

Одним из приоритетных направлений развития информационных технологий является разработка веб-ориентированных автоматизированных систем, обеспечивающих централизованный доступ к данным и поддержку бизнес-процессов организации. Внедрение веб-системы учёта оборудования позволяет автоматизировать процессы регистрации объектов, контроля их состояния, фиксации перемещений и формирования отчётной документации. Это снижает нагрузку на персонал, уменьшает влияние человеческого фактора и повышает достоверность учётных данных. Дополнительно использование механизмов авторизации и разграничения прав доступа обеспечивает защиту информации от несанкционированного доступа.

Объектом исследования является процесс учёта и управления оборудованием предприятия.

Предметом исследования является автоматизация процессов регистрации, инвентаризации и управления жизненным циклом оборудования с использованием веб-технологий.

Целью дипломного проекта является разработка веб-системы, обеспечивающей централизованный учёт оборудования, контроль его состояния, фиксацию перемещений и поддержку полного жизненного цикла объектов.

Для достижения поставленной цели необходимо решить следующие задачи:

— изучить предметную область и определить основные процессы учёта оборудования;

— провести анализ аналогов и прототипов;

— разработать первичную проектную документацию;

— спроектировать структуру системы и базы данных;

— реализовать программный продукт;

— провести тестирование разработанной системы;

— подготовить пользовательскую документацию.

В процессе выполнения дипломного проекта используются следующие программные средства:

1. Visual Studio Code — среда разработки программного обеспечения;

2. MariaDB — система управления базами данных для хранения информации об оборудовании;

3. Python — язык программирования для реализации серверной части веб-приложения;

4. Apache HTTP Server — веб-сервер для обработки клиентских запросов;

5. Git — система контроля версий;

6. Visio — инструмент для построения диаграмм;

7. PlantUML — инструмент для построения диаграмм;

8. Microsoft Office Word — средство подготовки текстовой документации;

9. Microsoft PowerPoint — средство подготовки презентации для защиты дипломного проекта.

1. Теоретическая часть

Глава 1. Анализ предметной области

Предметной областью разрабатываемой информационной системы является учёт, регистрация, перемещение и контроль состояния оборудования организации. В рассматриваемом проекте объект автоматизации связан с материально-технической базой предприятия: вычислительной техникой, периферийными устройствами, сетевым оборудованием, оргтехникой и иными материальными ресурсами, используемыми в текущей деятельности. Согласно техническому заданию, система ориентирована на ведение централизованного учёта оборудования, фиксацию его статусов, хранение истории перемещений, привязку объектов к помещениям и ответственным лицам, а также формирование отчётных документов.

С практической точки зрения данная предметная область находится на пересечении трёх направлений: внутреннего оперативного учёта имущества, инвентаризационного контроля и информационной поддержки управленческих решений. В международной практике близкий по смыслу класс решений относится к Enterprise Asset Management: это подход, при котором программные средства используются для контроля активов и оборудования на всём жизненном цикле, с целью повышения эффективности использования имущества, снижения простоев и сокращения затрат. Для дипломного проекта это важно не ради красивого англоязычного ярлыка, а потому что сама логика системы строится именно вокруг жизненного цикла объекта: поступление, регистрация, эксплуатация, перемещение, ремонт, списание. [8]

Юридическая и нормативная основа предметной области в России достаточно чёткая. Федеральный закон № 402-ФЗ «О бухгалтерском учёте» закрепляет обязанность ведения бухгалтерского учёта экономическими субъектами, а также устанавливает необходимость проведения инвентаризации активов и обязательств. Это означает, что имущество организации не может существовать в режиме «где-то лежит, потом найдём», как это обожают делать

некоторые хозяйствующие субъекты с интеллектом табуретки. Учёт активов должен быть подтверждаемым, системным и документированным. [1]

Дополнительно правовой контур предметной области определяется ФСБУ 6/2020 «Основные средства» и ФСБУ 26/2020 «Капитальные вложения», который регулирует подходы к признанию, оценке, амортизации и списанию объектов основных средств. Для разрабатываемой системы это означает, что сведения об оборудовании должны быть не просто перечнем наименований, а набором структурированных данных, пригодных для идентификации объекта, контроля его состояния и последующего использования в учётных и управленческих процедурах. Иначе говоря, система должна хранить не только факт существования оборудования, но и его учётно значимые характеристики. [2]

Отдельное значение для предметной области имеют положения, регламентирующие проведение инвентаризации имущества и финансовых обязательств, утверждённые приказом Минфина России № 4н. Данный документ устанавливает порядок проведения инвентаризации и оформления её результатов. Из этого следует важный практический вывод: автоматизированная система учёта оборудования должна поддерживать процессы сверки фактического наличия имущества с учётными данными, фиксировать расхождения, обеспечивать формирование документов по перемещению, списанию и уточнению сведений об объекте. То есть система не заменяет инвентаризацию как процедуру, а делает её выполнимой без ритуального страдания в Excel на три тысячи строк. [3]

Содержательно предметная область включает несколько базовых сущностей. К ним относятся: единица оборудования, категория оборудования, место размещения, материально ответственное или ответственное лицо, статус объекта, операция перемещения, операция списания, пользователь системы и отчётный документ. Между этими сущностями существуют устойчивые связи. Каждый объект оборудования относится к определённой категории, находится в конкретной локации, может быть закреплён за ответственным лицом, обладает текущим статусом и историей изменений. Именно эти связи формируют основу логической модели данных будущей системы. Такой подход полностью

согласуется и с техническим заданием, где прямо предусмотрены категории оборудования, статусы, история перемещений, акты и отчёты.

Основными участниками процессов в данной предметной области выступают сотрудники администрации, технические специалисты, кладовщик, сотрудник по работе с имуществом, руководитель подразделения и иные уполномоченные пользователи. Их интересы в системе различаются. Одним требуется регистрация новых единиц оборудования, другим — изменение статуса и оформление перемещений, третьим — получение сводной отчётности, четвёртым — контроль сохранности имущества. Следовательно, предметная область изначально предполагает разграничение ролей и прав доступа. Это особенно важно, поскольку система работает не просто как электронный список вещей, а как среда, в которой действия пользователей влияют на достоверность учётных данных и на управленческие решения.

Если рассматривать предметную область с позиции бизнес-процессов, то в ней можно выделить следующие ключевые процессы:

1. постановка оборудования на учёт;
2. регистрация характеристик объекта;
3. закрепление оборудования за местом эксплуатации и ответственным лицом;
4. изменение статуса объекта;
5. перемещение между помещениями, подразделениями или пользователями;
6. фиксация неисправностей, ремонта и утраты;
7. списание оборудования;
8. формирование отчётности и проведение инвентаризационной сверки.

Каждый из перечисленных процессов должен быть отражён в структуре информационной системы, иначе получится не модуль учёта, а декоративный интерфейс ради галочки. Причём особую важность имеют именно процессы изменения состояния объекта во времени, поскольку они обеспечивают актуальность данных и позволяют отслеживать весь жизненный цикл оборудования. [8]

Характерной проблемой предметной области является высокая зависимость качества учёта от человеческого фактора. При бумажном или слабо формализованном учёте типичными становятся дублирование записей, потеря данных о местонахождении оборудования, отсутствие единой истории перемещений, ошибки в статусах объектов и затруднения при подготовке отчётности. Кроме того, при разрозненном хранении информации в таблицах, мессенджерах и локальных файлах резко возрастает риск расхождения между фактическим наличием имущества и данными учёта. Именно поэтому автоматизация здесь оправдана не модой на «цифровую трансформацию», а прямой хозяйственной необходимостью.

Для рассматриваемой предметной области особенно значимы требования к достоверности, сохранности и защищённости от фальсификации данных. Поскольку система оперирует сведениями об имуществе организации и, согласно техническому заданию, может учитывать пользователей системы, необходимо обеспечить разграничение доступа, журналирование действий, резервное копирование и защиту от несанкционированного вмешательства. Такой подход согласуется как с логикой автоматизации учёта, так и с требованиями, закреплёнными в техническом задании и нормативной базе, включая положения о защите информации и персональных данных.

Следует также отметить, что предметная область учёта оборудования не сводится исключительно к бухгалтерскому учёту в узком смысле. Для разрабатываемой системы более корректно говорить об оперативном контуре управления имуществом, который поддерживает хозяйственные процессы организации и при необходимости, может служить источником данных для бухгалтерских процедур, инвентаризации и внутренней отчётности. Это важное различие: система должна быть достаточно строгой, чтобы опираться на нормативные требования, но при этом достаточно удобной для повседневной работы сотрудников. Иначе ей обеспечена судьба любой неудобной учётной системы: её будут ненавидеть, игнорировать и заполнять задним числом.

По итогам анализа можно сделать вывод, что предметная область характеризуется большим числом однотипных операций, высокой значимостью точности учётных данных, необходимостью формализации статусов и перемещений имущества, а также наличием нормативных требований к инвентаризации и документированию результатов. Следовательно, разработка автоматизированной веб-системы учёта оборудования является обоснованной и целесообразной. Такая система должна обеспечить централизованное хранение сведений об объектах, поддержку их жизненного цикла, разграничение прав пользователей, формирование отчётности и повышение общей прозрачности процессов управления материально-техническими ресурсами.

Глава 2. Анализ аналогов и прототипов сайта (веб-приложения) данной предметной области

Конкурентный анализ проведён с целью определения функциональных и юзабилити-пробелов на рынке систем учёта активов, которые необходимо учесть при проектировании интерфейса. Среди релевантных конкурентов выделяются три класса продуктов: открытые ITSM/ITAM-решения (GLPI), лёгкие веб-asset-менеджеры (Snipe-IT) и комплексные корпоративные платформы (ServiceNow), а также отечественные интеграционные продукты на базе 1С для инвентаризации. GLPI представляет собой зрелое решение с широкой функциональностью по учёту активов и интегрированным helpdesk-модулем; его преимущества — полнота справочной модели и развитые модули управления лицензиями и контрактами. При этом GLPI склонен к избыточной сложности для небольших кампусов, требует значительной начальной конфигурации и сопровождается ненулевым порогом вхождения для персонала, что делает его неудобным как «быстрое» решение для учебной инфраструктуры. [4]

Рисунок 1. Интерфейс системы GLPI

Snipe-IT ориентирован на простоту внедрения и прозрачность работы с инвентарём: модель-ориентированный подход, поддержка штрих- и QR-кодирования, уведомления о завершающейся гарантии и удобный REST API для интеграций. Эти качества делают Snipe-IT привлекательным для учреждений,

желающих получить базовую функциональность без больших затрат. В то же время у Snipe-IT выявляются ограничения при работе с расширенными процессами (например, сложные сценарии списания, многозвенные закупочные цепочки, глубокая интеграция с бухгалтерией), что создаёт необходимость доработок при попытке «взрослого» внедрения. [5]

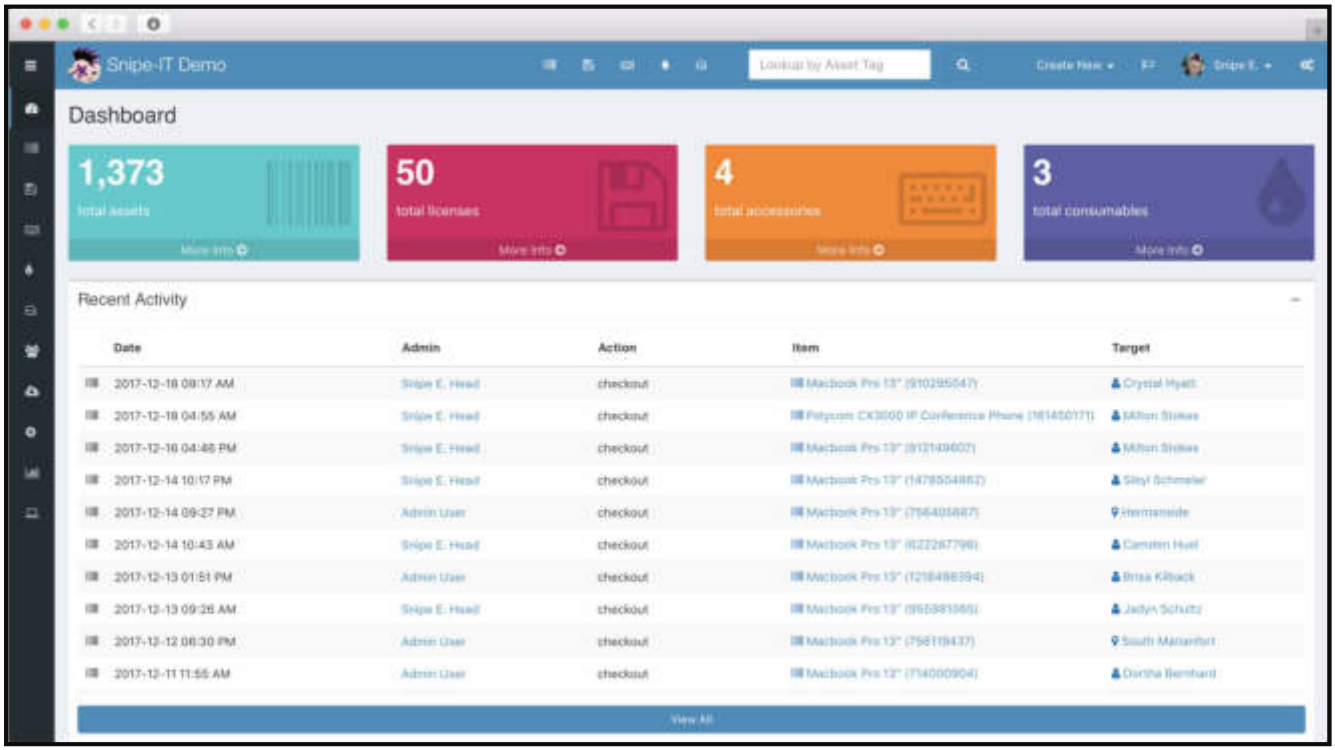


Рисунок 2. Интерфейс системы Snipe-IT

ServiceNow представляет собой продукт корпоративного уровня с широкими возможностями автоматизации жизненного цикла активов, интегрированными аналитическими панелями и поддержкой управленческих процессов. Для крупных организаций это безальтернативный вариант в силу мощи платформы и зрелости экосистемы, однако для учебного кампуса его главные недостатки — высокая стоимость владения, сложность внедрения и необходимость выделения специализированного персонала под сопровождение, что делает ServiceNow непропорциональным решением для целей ГПОАУ ЯО «ЯПЭК». [6]

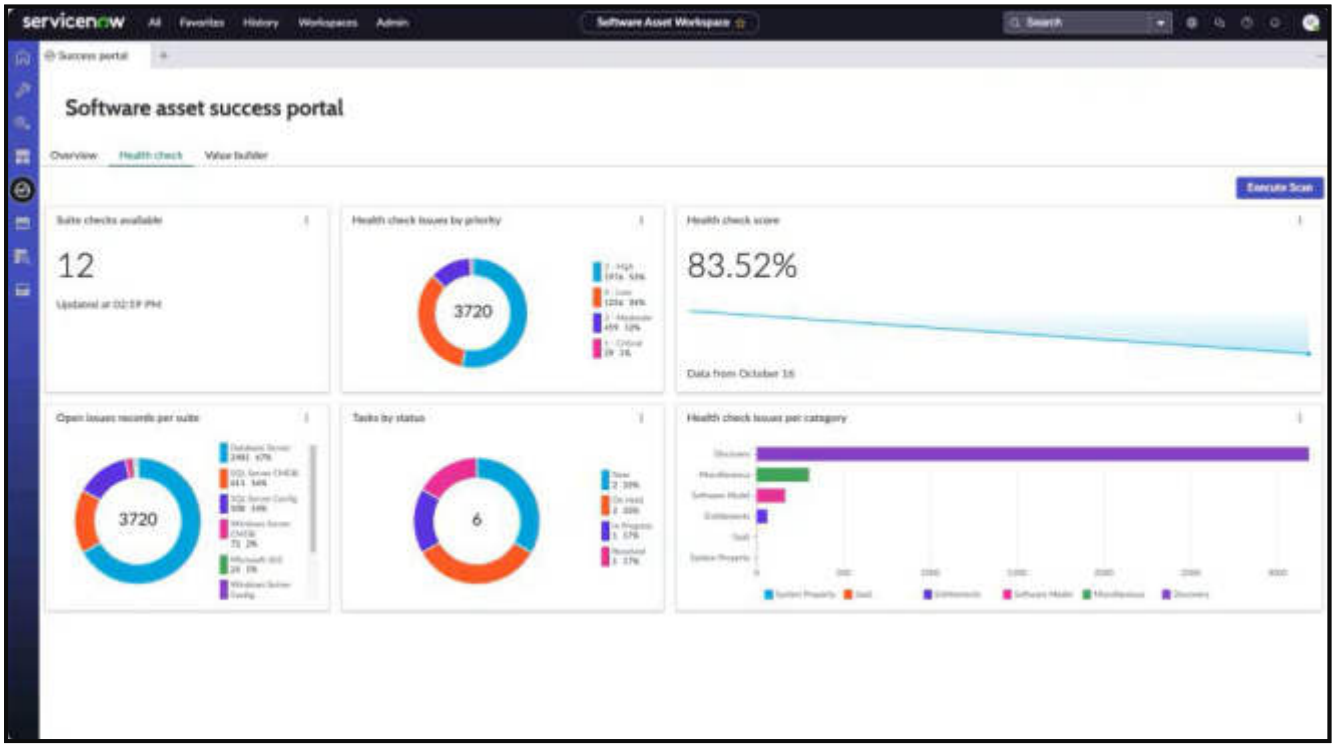


Рисунок 3. Интерфейс системы ServiceNow

Отечественные решения на базе 1С обладают очевидным преимуществом в части соответствия локальной бухгалтерской и регламентной практике, возможностей интеграции с расчётом амортизации и наличием партнёрской сети для внедрения. Тем не менее, ряд предложений представляет собой более «корпоративный» или «монолитный» подход с ограниченной гибкостью интерфейса и необходимостью участия партнёра при модификации под специфические процессы кампуса; мобильность и простота пользовательского интерфейса у таких решений часто уступают западным аналогам. [7]

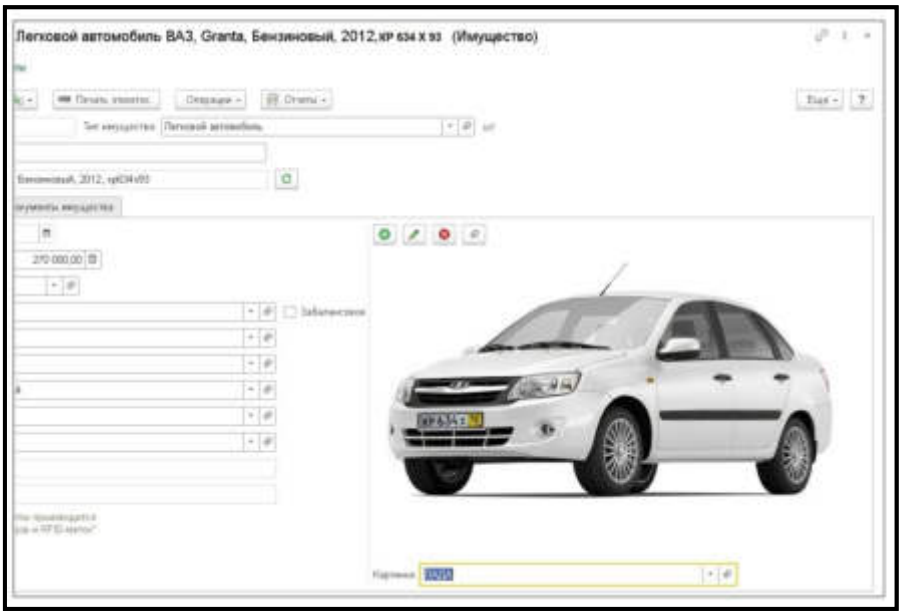


Рисунок 4. Интерфейс системы 1С: Инвентаризация

Для обобщения результатов анализа аналогов была сформирована сравнительная оценочная таблица. В качестве критериев использованы конкурентные преимущества, значимые для разрабатываемой автоматизированной веб-системы учёта оборудования: простота интерфейса, полнота учётной модели, поддержка жизненного цикла оборудования, удобство внедрения, возможность интеграции, соответствие регламентному учёту, масштабируемость, гибкость доработки и пригодность для использования в условиях учебной или небольшой организационной инфраструктуры. Оценивание критериев производится по трёхбалльной системе, где:

1 балл — критерий реализован на низком уровне либо соответствует требованиям разрабатываемой системы минимально;

2 балла — критерий реализован на среднем уровне и частично соответствует требованиям разрабатываемой системы;

3 балла — критерий реализован на высоком уровне и в наибольшей степени соответствует требованиям разрабатываемой системы.

Основанием для выставления оценок послужили выявленные особенности GLPI, Snipe-IT, ServiceNow и решений на базе 1С, включая их функциональные преимущества и ограничения.

Таблица 1. Анализ аналогов и прототипов.

№	Критерий оценки	GLPI	Snipe-IT	ServiceNow	1С: Инвентаризация
1	Простота и интуитивная понятность интерфейса	2	3	3	2
2	Полнота модели учёта оборудования и активов	3	2	3	3
3	Поддержка жизненного цикла оборудования	3	2	3	2
4	Удобство быстрого внедрения без сложной настройки	2	3	1	2
5	Возможность интеграции с внешними системами и API	2	3	3	2
6	Соответствие локальной бухгалтерской и регламентной практике	2	1	2	3
7	Масштабируемость и пригодность для крупных структур	2	2	3	3
8	Гибкость адаптации под специфические процессы организации	2	2	3	2
9	Пригодность для учебного кампуса или небольшой организации	2	3	1	2
Итог		**20**	**21**	**22**	**21**

Описание критериев:

— простота и интуитивная понятность интерфейса — характеризует удобство визуального восприятия системы, логичность расположения элементов управления и возможность выполнения основных операций без длительного обучения пользователя;

— полнота модели учёта оборудования и активов — отражает наличие средств для фиксации карточек оборудования, инвентарных данных, статусов, мест размещения и связанных с объектами учёта сведений;

— поддержка жизненного цикла оборудования — предполагает возможность сопровождения оборудования на основных этапах эксплуатации, включая поступление, регистрацию, перемещение, обслуживание, списание или вывод из использования;

— удобство быстрого внедрения без сложной настройки — характеризует возможность запуска системы в эксплуатацию без длительной предварительной

настройки, привлечения узких специалистов и существенного изменения существующих процессов организации;

— возможность интеграции с внешними системами и API — отражает наличие механизмов обмена данными с иными информационными системами, сервисами, учётными решениями или программными интерфейсами;

— соответствие локальной бухгалтерской и регламентной практике — характеризует пригодность решения для использования с учётом требований внутреннего учёта, инвентаризации, документального сопровождения и отчётности организации;

— масштабируемость и пригодность для крупных структур — отражает способность системы сохранять работоспособность и управляемость при увеличении количества оборудования, пользователей, подразделений и учётных операций;

— гибкость адаптации под специфические процессы организации — предполагает возможность настройки или доработки решения под особенности конкретной организации, включая изменение форм учёта, статусов, справочников и бизнес-процессов;

— пригодность для учебного кампуса или небольшой организации — характеризует соответствие решения условиям ограниченной инфраструктуры, умеренному объёму учётных данных и необходимости простого сопровождения без избыточной корпоративной сложности.

По результатам оценки наибольшую суммарную оценку получил ServiceNow, что объясняется его зрелостью, масштабируемостью и широтой функциональных возможностей. Вместе с тем данный результат не означает его практическое превосходство для рассматриваемого проекта, поскольку высокая оценка достигается за счёт корпоративной сложности, стоимости владения и необходимости специализированного сопровождения. Решения на базе 1С демонстрируют сильную сторону в части регламентного и бухгалтерского учёта, однако уступают по гибкости интерфейса и удобству быстрой адаптации. Snipe-IT показывает лучший баланс простоты и практической применимости, но ограничен при реализации сложных процессов списания, закупок и глубокой интеграции с

бухгалтерией. GLPI обладает развитой функциональностью, но для небольшой инфраструктуры может быть избыточно сложным и требовать существенной предварительной настройки. Таким образом, практическая применимость рассмотренных аналогов ограничивается не только их функциональным составом, но и сложностью внедрения, сопровождения и адаптации под реальные организационные процессы.

Исходя из проведённого анализа, в проекте сделан акцент на комбинированной стратегии: использовать лёгкую и интуитивно понятную клиентскую часть, характерную для простых веб-asset-менеджеров, и одновременно предусмотреть расширяемую серверную модель, совместимую с функциональными возможностями зрелых ITAM-платформ. Такой подход позволяет реализовать минимальный пользовательский сценарий оперативного учёта оборудования «из коробки», включая регистрацию объектов, изменение статусов, фиксацию перемещений и формирование отчётности, но при этом сохранить возможность последующего развития системы за счёт интеграции с бухгалтерскими механизмами, процессами закупок, API внешних систем и расширенными средствами аналитики. Выбранная стратегия направлена на исключение типичных недостатков рассмотренных аналогов: перегруженности интерфейса у GLPI, ограниченности предметной области у Snipe-IT, избыточной инфраструктурной сложности у ServiceNow и недостаточной гибкости отдельных решений на базе 1С.

Глава 3. Технико-экономическое обоснование создания сайта

3.1 Резюме и идея проекта

Технико-экономическое обоснование создания программного продукта представляет собой этап предварительной оценки целесообразности выполнения работ по разработке автоматизированной информационной системы. В рамках данного фрагмента технико-экономического обоснования рассматривается не расчёт материальных ресурсов, оборудования или иных затратных показателей, а только распределение времени, необходимого для последовательного выполнения основных этапов разработки АИС «Инвентарис».

Разрабатываемая система «Инвентарис» ориентирована на автоматизацию процессов учета, инвентаризации и контроля жизненного цикла оборудования предприятия. Основная идея проекта заключается в создании единого централизованного инструмента, позволяющего хранить сведения об оборудовании, отслеживать изменения его состояния, фиксировать перемещения и обеспечивать доступ к актуальной информации в удобной форме.

В отличие от разрозненных таблиц и локальных решений, предлагаемый программный продукт реализует целостную модель работы с данными, в которой каждая единица оборудования рассматривается как объект с историей изменений. Такой подход позволяет обеспечить прослеживаемость операций, повысить достоверность данных и упростить контроль за состоянием материальных ресурсов организации.

Ключевой особенностью проекта является использование унифицированной модели прав доступа, позволяющей разграничивать действия пользователей в зависимости от их роли. Это обеспечивает базовый уровень защищенности данных и предотвращает несанкционированные изменения информации.

Реализация системы предполагает поэтапную разработку, включающую анализ предметной области, проектирование архитектуры, разработку пользовательского интерфейса и бизнес-логики, а также интеграцию с системой хранения данных. При этом основное внимание уделяется удобству эксплуатации, расширяемости и поддерживаемости программного продукта.

Для реализации проекта требуется минимальный набор технических и программных средств, обеспечивающих разработку и последующее функционирование веб-приложения:

— персональный компьютер или ноутбук с установленной операционной системой (Linux/MacOS/Windows);

— среда разработки (например, Visual Studio Code);

— доступ к сети Интернет;

— система управления базами данных (MariaDB или MySQL);

— интерпретатор языка программирования Python и необходимые библиотеки;

— веб-браузер для тестирования и эксплуатации системы.

Для функционирования системы требуется доступ к вычислительным ресурсам. Опционально: наличие электроснабжения или автономного источника питания.

Таким образом, разработка АИС «Инвентарис» не требует специализированного дорогостоящего оборудования и может быть реализована в стандартной программной среде, что подтверждает практическую целесообразность и доступность проекта.

3.2 Общее описание задачи

Работу над созданием автоматизированной информационной системы «Инвентарис» необходимо начинать с общего описания задачи, поскольку именно на данном этапе определяется логика дальнейшего выполнения проекта. Общее описание задачи включает формирование представления о жизненном цикле программного продукта, определение последовательности работ, выделение основных этапов разработки и установление контрольных событий. Без указанного этапа разработка превращается в набор разрозненных действий, что для отчётной документации, к счастью или к сожалению, считается не лучшим достижением человеческой мысли.

Жизненный цикл автоматизированной информационной системы представляет собой совокупность стадий, через которые проходит программный продукт от момента принятия решения о его создании до передачи заказчику и последующего использования. Для рассматриваемого проекта жизненный цикл включает подготовку требований, проектирование, реализацию, тестирование и ввод в действие. Такой подход позволяет последовательно перейти от анализа предметной области к созданию проектных моделей, затем к разработке программной части, проверке работоспособности и подготовке системы к эксплуатации.

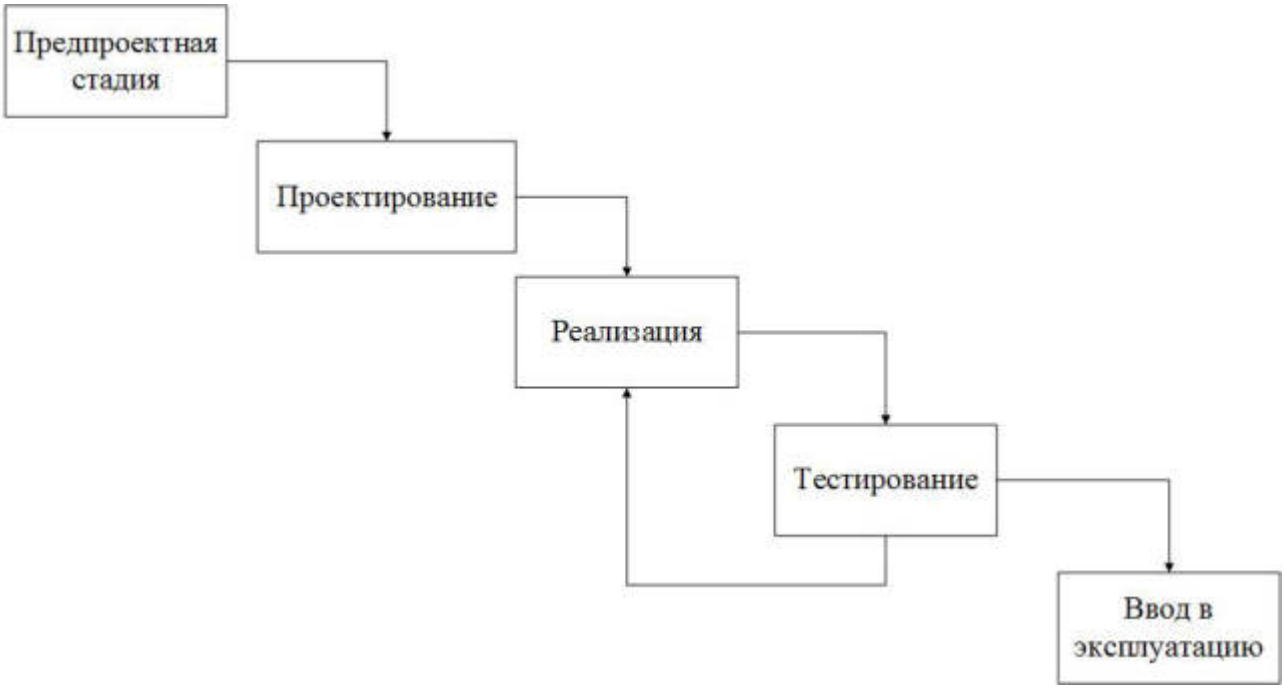


Рисунок 5. Жизненный цикл АИС «Инвентарис»

На этапе планирования выполняются работы, направленные на подготовку исходной базы проекта. В указанный этап входят анализ предметной области, анализ аналогов и прототипов разрабатываемого программного продукта, разработка фрагмента технико-экономического обоснования, общее описание задачи, формирование первичной документации, составление календарного плана работ и рассмотрение требований техники безопасности. Результатом этапа является подготовленный комплект первичной документации, позволяющий перейти к проектированию системы.

Этап проектирования предназначен для формирования проектного представления о будущей автоматизированной информационной системе. В рамках данного этапа определяются структура АИС, состав основных модулей, логика взаимодействия пользователей с системой, а также общие принципы построения информационной модели. Для АИС «Инвентарис» на данном этапе особое значение имеет проектирование учёта оборудования, статусов, перемещений, ответственных лиц и отчётных данных, поскольку именно эти элементы образуют основу будущей системы.

Этап реализации физической модели связан с практическим созданием программного продукта. На данном этапе выполняется разработка базы данных, реализация пользовательского интерфейса и программных механизмов, обеспечивающих регистрацию оборудования, изменение его состояния, фиксацию перемещений, поиск информации и формирование отчётности. Также предусматривается организация безопасности АИС, включая разграничение прав доступа и обеспечение корректной работы с учётными данными.

После завершения реализации система передаётся на тестирование. На данном этапе выполняется проверка работоспособности программного продукта, выявление ошибок, контроль корректности основных пользовательских сценариев и подтверждение готовности системы к дальнейшему использованию. Тестирование необходимо для того, чтобы убедиться, что созданная АИС выполняет заявленные функции, а не просто красиво существует в отчёте, как некоторые особенно декоративные программные продукты.

Заключительным этапом жизненного цикла в рамках данного проекта является ввод в действие. На этом этапе выполняется настройка аппаратных и программных средств, необходимых для функционирования АИС, подготавливается руководство пользователя, проводится обучение персонала и осуществляется передача программного продукта вместе с технической документацией заказчику. Данный этап фиксирует завершение разработки и переход системы из состояния проектного решения в состояние готового программного продукта.

Для контроля выполнения проекта были выделены основные контрольные события: готовность первичной документации, готовность программного продукта к тестированию, полная готовность программного продукта, а также передача программного продукта и технической документации заказчику. Наличие контрольных событий позволяет формально зафиксировать завершение ключевых стадий и обеспечить управляемость процесса разработки.

Таким образом, общее описание задачи позволило определить состав и последовательность работ по созданию АИС «Инвентарис», установить основные этапы жизненного цикла и выделить контрольные события проекта. Сформированная структура работ обеспечивает логичный переход от планирования к проектированию, реализации, тестированию и вводу системы в действие.

3.3 Составление плана работ

При подготовке календарного плана была определена общая продолжительность работ, а также выполнено распределение задач по стадиям жизненного цикла программного продукта. Общий срок выполнения проекта установлен с 20.04.2026 по 30.05.2026 и составляет 36 дней. Данный период включает планирование, проектирование и реализацию, тестирование, а также ввод программного продукта в действие. Такой порядок позволяет не только зафиксировать состав работ, но и определить логическую последовательность их выполнения, чтобы последующие этапы опирались на результаты предыдущих.

На этапе планирования предусмотрено выполнение работ по анализу предметной области, анализу аналогов и прототипов, разработке фрагмента технико-экономического обоснования, общему описанию задачи, подготовке первичной документации, составлению плана работ и рассмотрению требований техники безопасности. Продолжительность данного этапа составляет 10 дней, с 20.04.2026 по 30.04.2026. Завершением этапа является контрольное событие «Готовность первичной документации», которое фиксирует окончание подготовительной части проекта.

Следующий этап объединяет проектирование и реализацию программного продукта. Его продолжительность составляет 15 дней, с 02.05.2026 по 19.05.2026. В рамках данного периода выполняется проектирование АИС, реализация физической модели, а также организация безопасности автоматизированной информационной системы. Завершением этапа является контрольное событие «Готовность программного продукта к тестированию», после которого система может быть передана на проверку работоспособности.

На тестирование программного продукта отведено 3 дня, с 19.05.2026 по 21.05.2026. На данном этапе выполняется проверка работоспособности системы, выявление возможных ошибок и подтверждение соответствия реализованного функционала ранее сформированным требованиям. Завершением тестирования выступает контрольное событие «Полная готовность программного продукта», фиксирующее возможность перехода к подготовке системы к использованию.

Заключительным этапом является ввод программного продукта в действие. Его продолжительность составляет 8 дней, с 22.05.2026 по 30.05.2026. На данном этапе выполняется настройка аппаратных и программных средств, необходимых для функционирования АИС, разрабатывается руководство пользователя, проводится обучение персонала, а также осуществляется передача программного продукта и технической документации заказчику. Детализированный календарный план работ и диаграмма Ганта представлены в Приложении Д.

Таким образом, разработка фрагмента технико-экономического обоснования позволила определить временные границы проекта, распределить работы по стадиям жизненного цикла и зафиксировать контрольные события. Принятый план выполнения работ является достаточным для организации последовательной разработки АИС «Инвентарис» и обеспечивает возможность контроля хода проекта на каждом из основных этапов.

Глава 4. Разработка первичной документации

Разработка первичной документации является одним из обязательных этапов подготовки программного продукта к последующему проектированию и реализации. На данном этапе выполняется формализация исходных требований к автоматизированной информационной системе, уточняется назначение программного продукта, определяются основные функции, категории пользователей, ограничения и порядок дальнейшей разработки. Первичная документация необходима для того, чтобы зафиксировать исходные договорённости между заказчиком и исполнителем, а также обеспечить единое понимание состава и назначения создаваемой АИС «Инвентарис».

В рамках разработки первичной документации были подготовлены документы, определяющие исходную постановку задачи, общее представление о системе и формализованные требования к её созданию. К таким документам относятся бриф на разработку автоматизированной информационной системы, общая концепция программного продукта и техническое задание. Указанные документы связаны между собой и образуют последовательную основу для дальнейшего проектирования: бриф фиксирует первичные сведения и ожидания заказчика, общая концепция раскрывает замысел и функциональные границы системы, а техническое задание закрепляет требования к составу, функциям, структуре и условиям эксплуатации программного продукта.

Бриф на разработку АИС «Инвентарис» был подготовлен как исходный документ, предназначенный для первичного сбора сведений о проекте. В нём были отражены общая информация о системе, цель её создания, основные проблемы предметной области, предполагаемые категории пользователей, ключевые функциональные требования, требования к интерфейсу, технические условия эксплуатации, вопросы безопасности и ожидаемый результат разработки. Бриф позволяет зафиксировать начальное представление о будущем программном продукте и определить направления, которые должны быть дополнительно раскрыты в последующей проектной документации. Подготовленный бриф представлен в Приложении В.

На основании сведений, полученных при анализе предметной области и подготовке брифа, была разработана общая концепция автоматизированной информационной системы. Общая концепция определяет назначение системы, её основные цели, состав функциональных модулей, категории пользователей, принципы взаимодействия с интерфейсом, требования к надёжности, безопасности и сопровождению. Данный документ является промежуточным проектным основанием, позволяющим перейти от общей идеи автоматизации к конкретному описанию структуры будущего программного продукта. В рамках концепции АИС «Инвентарис» рассматривается как веб-система для централизованного учёта оборудования, контроля статусов, фиксации перемещений и формирования отчётности. Общая концепция программного продукта представлена в Приложение К.

Особое значение в составе первичной документации имеет техническое задание. Техническое задание представляет собой документ, в котором закрепляются цели создания системы, характеристика объекта автоматизации, требования к системе в целом, требования к выполняемым функциям, видам обеспечения, порядку контроля, приёмки и документирования. Для АИС «Инвентарис» техническое задание определяет назначение системы, состав пользователей, основные функции учёта оборудования, требования к сохранности информации, разграничению доступа, эксплуатации и сопровождению. Наличие технического задания позволяет использовать его в качестве основного документа при дальнейшем проектировании, реализации и проверке соответствия программного продукта установленным требованиям. Разработанное техническое задание представлено в Приложении Г.

При разработке первичной документации были определены основные функциональные возможности будущей системы. К ним относятся регистрация оборудования, ведение карточек объектов, контроль текущего статуса, фиксация перемещений, привязка оборудования к помещениям и ответственным лицам, формирование отчётности, управление пользователями и ведение журнала действий. Отдельное внимание было уделено требованиям к безопасности, поскольку система работает с учётными данными об имуществе организации и

действиях пользователей. В связи с этим в документации предусмотрены разграничение прав доступа, резервное копирование, журналирование операций и требования к сохранности информации.

Также в процессе разработки первичной документации были уточнены требования к интерфейсу и пользовательскому взаимодействию. Система должна иметь деловой, простой и предсказуемый интерфейс, обеспечивающий быстрый доступ к основным разделам: учёту оборудования, перемещениям, отчётности и административным функциям. Для снижения вероятности ошибок при вводе данных должны применяться структурированные формы, валидация значений и подтверждение критически важных операций. Такой подход обеспечивает удобство эксплуатации системы и снижает зависимость качества учёта от случайных действий пользователей, потому что, как показывает практика, случайные действия пользователей способны разрушить больше, чем некоторые вредоносные программы.

Разработка первичной документации позволила сформировать документальную основу проекта и определить дальнейший порядок выполнения работ. На основании брифа, общей концепции и технического задания становится возможным перейти к проектированию структуры базы данных, пользовательского интерфейса, функциональных модулей и механизмов обработки данных. Таким образом, первичная документация выполняет роль связующего звена между анализом предметной области и непосредственным проектированием автоматизированной информационной системы.

В результате выполнения данного этапа были подготовлены исходные документы, необходимые для дальнейшей разработки АИС «Инвентарис»: бриф на разработку системы, общая концепция и техническое задание. Указанные документы обеспечивают формализацию требований, фиксируют назначение и функциональные границы программного продукта, а также определяют основные условия его создания и последующей эксплуатации.

Глава 5. Техника безопасности

Общие правила безопасности

Техника безопасности при работе с компьютером на предприятии предусматривает наличие общедоступной инструкции, в которой указаны обязательные требования к обустройству рабочего места и процессу использования техники. Эти правила едины для всех организаций, их выполнение контролируется руководящими органами.

Техника безопасности при работе с компьютером

В рамках проведения инструктажа до работника доводятся сведения о необходимых действиях, выполняемых на каждом этапе рабочего процесса. Техника безопасности в начале работы на компьютере требует выполнения следующих операций:

— проверить исправность элементов электропитания, обеспечивающей работу компьютера, включая электропроводку, выключатели, вилки и розетки, при помощи которых аппаратура подключается к сети;

— проверить его работоспособность.

При выполнении работ нужно соблюдать следующие правила:

— запрещается класть на корпус и дисплей компьютера посторонние предметы, прикасаться к элементам аппаратуры мокрыми руками, производить чистку корпуса оборудования, находящегося под напряжением;

— в случае обнаружения неисправности или задымления компьютера немедленно прекратить работу и сообщить об этом непосредственному руководителю;

— эксплуатировать компьютер только с соблюдением инструкции, установленной производителем;

— избегать частого и необоснованного включения и выключения компьютера во время работы.

После завершения работы сотруднику нужно выполнить следующие действия:

— выключить компьютер с использованием алгоритма, установленного производителем;

— обесточить периферийное оборудование;

— убедиться в отключении техники. Техника безопасности при работе с принтером

— проверить исправность элементов электропитания, обеспечивающей работу компьютера, включая электропроводку, выключатели, вилки и розетки, при помощи которых аппаратура подключается к сети;

— проверить его работоспособность;

— запрещается класть на корпус посторонние предметы, прикасаться к элементам аппаратуры мокрыми руками;

— по окончании работ обесточить оборудование;

— убедиться в отключении техники.

Аварийные ситуации

Своевременная бдительность поможет избежать опасных ситуаций для жизни и сохранить целостность техники.

Действия в аварийных ситуациях:

1. при неполадках любого рода в электроснабжении устройства необходимо сразу отключить компьютер от сети;

2. если обнаружен оголённый провод, то необходимо оперативно оповестить всех работников офиса, не допуская чьего-либо контакта с ним;

3. в каждом учреждении должны находиться огнетушители ОУБ-3 или ОУ-2, а также ведра и полотна в необходимом количестве. Персонал обязан знать о том, где находятся средства для гашения пламени и куда нужно звонить в случае пожара;

4. при поражении человека электрическим током, прежде всего, оказывается первая помощь: искусственное дыхание и внешний интенсивный массаж сердца. В первые же мгновения после удара током, вызывается скорая помощь.

2. Практическая часть

Глава 1. Проектирование сайта (веб-приложения)

Проектирование сайта, рассматриваемого в рамках данного проекта как веб-приложение автоматизированной информационной системы «Инвентарис», выполнялось после завершения анализа предметной области, подготовки брифа, общей концепции и технического задания. На данном этапе были определены функциональная структура системы, состав основных пользовательских сценариев, логическая модель данных и порядок взаимодействия пользователя с серверной частью приложения. Проектирование выполнялось не как самостоятельная абстрактная работа, а как продолжение ранее сформированных требований, зафиксированных в брифе на разработку системы, техническом задании и общей концепции программного продукта.

Основным исходным документом для проектирования выступало техническое задание, представленное в Приложении Г. В нём были определены назначение АИС «Инвентарис», состав пользователей, требования к функциям системы, требования к хранению данных, разграничению прав доступа, сохранности информации и последующей эксплуатации. В соответствии с требованиями ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» техническое задание является основным документом, определяющим требования и порядок создания автоматизированной системы, поэтому именно оно было использовано в качестве формального основания для построения проектных моделей [10]. Дополнительно при проектировании учитывались сведения, полученные на этапе брифинга с заказчиком, представленного в Приложении В, а также положения общей концепции системы, представленной в Приложении К.

Проектирование выполнялось с учётом стадийности создания автоматизированных систем. В соответствии с ГОСТ Р 59793-2021 «Национальный стандарт Российской Федерации. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы.

Стадии создания» процесс создания автоматизированной системы рассматривается как совокупность взаимосвязанных стадий и этапов работ, направленных на создание системы, соответствующей установленным требованиям [9]. Поэтому на этапе проектирования были подготовлены модели, позволяющие перейти от текстового описания требований к более формализованному представлению будущего программного продукта.

В рамках проектирования была создана функциональная модель процесса регистрации оборудования в нотации IDEF0. Данная диаграмма представлена в Приложении Е и включает контекстный блок A0, декомпозицию процесса на блоки A1–A4, а также дополнительную декомпозицию блока A1 до уровня A11–A13. Контекстный блок A0 отражает общий процесс регистрации оборудования в системе. В качестве входных данных используются сведения авторизации пользователя, запрос на регистрацию и данные нового оборудования. Управляющими воздействиями выступают правила АИС, требования законодательства в области защиты и обработки информации, а также правила учёта и инвентаризации оборудования. Механизмами выполнения процесса являются пользователь, веб-интерфейс, сервер приложения, база данных и журнал событий. На выходе формируются результат регистрации, карточка оборудования и запись в реестре.

Декомпозиция функциональной модели позволяет уточнить внутреннюю последовательность обработки операции регистрации оборудования. В составе модели выделены четыре основных процесса: проверка доступа пользователя, принятие данных об оборудовании, проверка корректности введённых сведений и формирование результата. Такой подход позволяет показать, что регистрация оборудования в системе не является простой записью строки в таблицу, а включает проверку полномочий, приём исходных данных, контроль их корректности, обращение к базе данных и фиксацию результата в журнале действий. Использование IDEF0 в данном случае целесообразно, поскольку данная методика предназначена для функционального моделирования процессов, их входов, выходов, управляющих воздействий и механизмов выполнения [12].

Для описания взаимодействия пользователей с системой была разработана диаграмма вариантов использования, представленная в Приложении Ж, а также спецификация, представленная в Приложении Л. Данная диаграмма отражает основные роли пользователей и функции, доступные им в рамках АИС «Инвентарис». В качестве акторов выделены сотрудник, руководитель и бухгалтер. Сотрудник взаимодействует с функциями управления оборудованием, включая просмотр, регистрацию и редактирование оборудования. Руководитель отвечает за управление пользователями, управление ролями, создание новых пользователей, создание ролей, удаление учётных записей, а так же за формирование справочников системы. Бухгалтер использует систему преимущественно для формирования отчётов. У каждого актёра дополнительно указаны права в формате ESR, подробнее о системе прав можно узнать в разделе 2, главе 3 «Организация безопасности сайта (веб-приложения)», а так же в руководстве пользователя предоставленном в Приложении А (Раздел 2 пункт 2.1 «Модель прав ESR»).

Диаграмма вариантов использования была сформирована на основании функциональных требований технического задания и общей концепции системы. Она позволяет определить границы ответственности пользователей, а также зафиксировать состав основных пользовательских сценариев. Применение UML-диаграмм при проектировании программных систем соответствует общепринятому подходу к моделированию программного обеспечения; стандарт ISO/IEC 19505-2:2012 «Информационные технологии. Унифицированный язык моделирования группы по управлению объектами (OMG UML). Часть 2. Сверхструктура» определяет UML как язык моделирования, используемый для анализа, проектирования и реализации программных систем [14]. В рамках данного проекта диаграмма вариантов использования применялась для уточнения функционального состава веб-приложения и распределения возможностей между категориями пользователей.

Отдельным результатом проектирования стала информационная модель базы данных в нотации IDEF1X, представленная в Приложении З. Данная модель описывает структуру хранения данных, состав сущностей и связи между ними. Центральной сущностью является таблица equipment_items, предназначенная для

хранения сведений об объектах оборудования. С ней связаны таблицы моделей оборудования, категорий, статусов, помещений, закупок, поставщиков, пользователей, ролей, перемещений, записей обслуживания и инвентарных меток. Такая структура позволяет хранить не только карточку оборудования, но и связанные с ней сведения о происхождении, местоположении, статусе, ответственном лице, истории перемещений и обслуживании.

Модель IDEF1X была сформирована на основании требований к ведению учёта оборудования, фиксации статусов, регистрации перемещений и формированию отчётности. Её назначение заключается в том, чтобы заранее определить логическую структуру базы данных до перехода к физической реализации. Использование IDEF1X позволяет представить данные в виде сущностей, атрибутов, первичных и внешних ключей, что особенно важно для систем, где требуется целостность связей и исключение дублирования информации. Стандарт FIPS PUB 184 «Integration Definition for Information Modeling (IDEF1X)» (Методология информационного моделирования IDEF1X) описывает IDEF1X как язык информационного моделирования, применяемый для разработки логической модели данных [13].

Также в ходе проектирования была подготовлена диаграмма последовательностей, представленная в Приложении И. Она описывает порядок взаимодействия пользователя, веб-интерфейса, сервера приложения, базы данных и сессионного хранилища при выполнении ключевых сценариев. В диаграмме рассмотрены авторизация пользователя, загрузка списка оборудования и запрос отчёта. Для каждого сценария отражены запросы, проверки, обращения к базе данных, успешные ответы и альтернативные ветви обработки ошибок.

Особое значение в диаграмме последовательностей имеет отображение альтернативных сценариев. При авторизации предусмотрены варианты успешного входа и ошибки авторизации. При загрузке списка оборудования учитывается проверка токена сессии, наличие прав на чтение списка и возможный отказ при недействительной сессии или недостатке прав. Аналогичная проверка происходит и при запросе отчёта, проверяется действительность сессии, наличие прав на чтение в соответствующем контуре и при прошествии всех проверок формируется

результат. Такой подход позволяет заранее описать не только основной сценарий работы системы, но и обработку типовых ошибок, которые неизбежно появляются там, где пользователь получает форму ввода и право нажимать кнопки.

Сформированные диаграммы в совокупности обеспечивают несколько уровней проектного описания веб-приложения. Диаграмма IDEF0 отражает функциональную логику процесса регистрации оборудования. Диаграмма вариантов использования показывает взаимодействие пользователей с системой и распределение функций между ролями. Диаграмма IDEF1X раскрывает структуру данных и связи между основными сущностями. Диаграмма последовательностей описывает динамику выполнения пользовательских сценариев и взаимодействие компонентов системы. Такой набор проектных материалов позволяет перейти от общего описания требований к конкретной структуре программного продукта.

Проектные модели также обеспечивают связь между аналитической и реализационной частями работы. На основании диаграммы вариантов использования возможно определить состав экранных форм и разделов веб-приложения. На основании IDEF0-модели уточняется порядок обработки операции регистрации оборудования. На основании IDEF1X-диаграммы формируется структура базы данных. На основании диаграммы последовательностей определяется порядок обработки запросов между клиентской частью, сервером приложения, базой данных и журналом действий. Тем самым проектирование выполняет роль промежуточного этапа между техническим заданием и непосредственной реализацией физической модели.

В результате выполнения этапа проектирования были подготовлены основные проектные материалы, необходимые для дальнейшей разработки АИС «Инвентарис». К ним относятся функциональная модель IDEF0 с блоком A0 и декомпозицией A1-A4 и декомпозицей A1.1-A1.3, диаграмма вариантов использования, информационная модель данных IDEF1X и диаграмма последовательностей. Все указанные материалы представлены в приложениях к отчёту: функциональная модель приведена в Приложении Е, глоссарий к диаграмме IDEF0 в Приложении М, диаграмма вариантов использования — в Приложении Ж, диаграмма IDEF1X — в Приложении З, диаграмма

последовательностей — в Приложении И. Разработанные диаграммы позволяют подтвердить, что проектирование веб-приложения выполнено на основании ранее сформированных требований и обеспечивает достаточную основу для последующей реализации программного продукта.

Глава 2. Реализация физической модели

Реализация физической модели определяет фактический состав таблиц базы данных, типы данных, первичные и внешние ключи, обязательность заполнения полей, а также способ обращения программных модулей к сохраненным сведениям. В рамках разработки веб-системы «Инвентарис» физическая модель была реализована как реляционная структура MariaDB, связанная с серверной частью приложения через слой SQL-запросов и программные функции обработки пользовательских операций.

Для реализации физической модели веб-системы «Инвентарис» использовались язык программирования Python, микрофреймворк Flask, шаблонизатор Jinja2, библиотека PyMySQL, СУБД MariaDB и интерфейсные компоненты Bootstrap. Серверная часть приложения разделена на функциональные модули, каждый из которых отвечает за отдельный участок предметной области: оборудование, справочники, перемещения, отчеты, учетные записи и общие операции преобразования данных.

При разработке программного продукта была принята схема, при которой структура базы данных хранит не только актуальное состояние оборудования, но и сведения, необходимые для восстановления цепочки изменений. В карточке оборудования предусмотрено поле ссылки на предыдущую версию, за счет чего редактирование значимых реквизитов не приводит к потере исторических данных. Такой подход позволяет физической модели отражать жизненный цикл инвентарной единицы без создания отдельного архива в виде внешних файлов или неструктурированных журналов.

На основе ERD-модели были разработаны следующие таблицы базы данных:

— roles — таблица, хранящая перечень ролей и конфигурацию прав доступа;

— statuses — таблица, хранящая статусы оборудования;

— categories — таблица, хранящая категории оборудования;

— suppliers — таблица, хранящая сведения о поставщиках;

— equipment_models — таблица, хранящая модели оборудования и их технические характеристики;

— locations — таблица, хранящая сведения о помещениях и местах размещения оборудования;

— users — таблица, хранящая учетные записи сотрудников, связанных с операциями системы;

— purchases — таблица, хранящая сведения о закупках;

— equipment_items — таблица, хранящая карточки инвентарных единиц и их версии;

— item_movements — таблица, хранящая журнал перемещений оборудования;

— maintenance_records — таблица, хранящая сведения об обслуживании оборудования;

— asset_tags — таблица, хранящая сведения об инвентарных метках.

Физическая структура базы данных построена по принципу разделения справочной, учетной и операционной информации. Справочные таблицы используются для нормализации повторяющихся значений, учетные таблицы фиксируют основные сущности предметной области, а операционные таблицы отражают действия, выполняемые с оборудованием в процессе эксплуатации системы. За счет этого в базе данных уменьшается дублирование сведений, а изменение справочного значения не требует изменения структуры связанных карточек.

Основной таблицей физической модели является equipment_items. В ней хранится инвентарная единица, ее модель, текущее помещение, статус, сведения о закупке, стоимость, даты покупки и гарантии, а также служебные поля создания, обновления и связи с предыдущей версией. Таблицы categories, statuses, locations, equipment_models и suppliers формируют набор атрибутов, используемых при создании карточек оборудования и построении отчетных представлений. Таблица item_movements дополняет версионность карточек отдельным журналом фактов перемещения.

Словарь данных физической модели приведен в таблицах 2-13. В столбце «Ключ» указывается назначение поля в структуре таблицы: PK обозначает первичный ключ, FK обозначает внешний ключ. В столбце «Обязательный к

заполнению» значение Y означает, что поле не допускает NULL, а значение N означает, что поле может быть оставлено пустым.

Таблица 2. Поля таблицы «roles»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	name	VARCHAR(150)	Y
	permissions	CHAR(3)	Y
	description	TEXT	N

Таблица 3. Поля таблицы «statuses»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	name	VARCHAR(100)	Y
	description	TEXT	N

Таблица 4. Поля таблицы «categories»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	name	VARCHAR(150)	Y
	description	TEXT	N

Таблица 5. Поля таблицы «suppliers»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	name	VARCHAR(255)	Y
	contact	VARCHAR(255)	N
	address	VARCHAR(255)	N
	phone	VARCHAR(50)	N
	email	VARCHAR(150)	N

Таблица 6. Поля таблицы «equipment_models»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	manufacturer	VARCHAR(150)	N
	model_name	VARCHAR(200)	Y
FK	category_id	INT UNSIGNED	Y
	spec	JSON	N
	warranty_months	SMALLINT UNSIGNED	N

Таблица 7. Поля таблицы «locations»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	building	VARCHAR(100)	N
	floor	VARCHAR(50)	N
	room	VARCHAR(100)	Y
	description	TEXT	N

Таблица 8. Поля таблицы «users»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	full_name	VARCHAR(200)	Y
	email	VARCHAR(150)	N
	phone	VARCHAR(50)	N
FK	role_id	INT UNSIGNED	N
	employee_number	VARCHAR(100)	N
	login	VARCHAR(32)	Y
	password_hash	VARCHAR(256)	Y
	hired_date	DATE	N
	active	TINYINT(1)	Y

Таблица 9. Поля таблицы «purchases»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
FK	supplier_id	INT UNSIGNED	N
	order_number	VARCHAR(150)	N
	invoice_number	VARCHAR(150)	N
	purchase_date	DATE	Y
	total_cost	DECIMAL(12,2)	N
	notes	TEXT	N

Таблица 10. Поля таблицы «equipment_items»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
	inventory_number	VARCHAR(100)	Y
	serial_number	VARCHAR(200)	N
FK	model_id	INT UNSIGNED	Y
FK	location_id	INT UNSIGNED	N
FK	purchase_id	INT UNSIGNED	N
FK	status_id	INT UNSIGNED	Y
	purchase_date	DATE	N
	warranty_until	DATE	N
	price	DECIMAL(12,2)	N

Ключ	Имя поля	Тип данных	Обязательный к заполнению
	created_at	DATETIME	Y
FK	created_by	INT(10)	Y
	updated_at	DATETIME	Y
FK	updated_by	INT(10)	N
FK	id_before_update	INT(10)	N

Таблица 11. Поля таблицы «item_movements»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	BIGINT UNSIGNED AUTO_INCREMENT	Y
FK	item_id	INT UNSIGNED	Y
	movement_type	VARCHAR(50)	Y
FK	from_location_id	INT UNSIGNED	N
FK	to_location_id	INT UNSIGNED	N
FK	performed_by	INT UNSIGNED	N
	performed_at	DATETIME	Y
	note	TEXT	N

Таблица 12. Поля таблицы «maintenance_records»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	BIGINT UNSIGNED AUTO_INCREMENT	Y
FK	item_id	INT UNSIGNED	Y
	maintenance_date	DATE	Y
	performed_by_user_id	INT UNSIGNED	Y
	description	TEXT	N
	cost	DECIMAL(12,2)	N
	next_due	DATE	N

Таблица 13. Поля таблицы «asset_tags»

Ключ	Имя поля	Тип данных	Обязательный к заполнению
PK	id	INT UNSIGNED AUTO_INCREMENT	Y
FK	item_id	INT UNSIGNED	Y
	tag_code	VARCHAR(200)	Y
	issued_at	DATE	N
FK	issued_by	INT UNSIGNED	N

Для обращения к физической модели используется отдельный низкоуровневый слой работы с MySQL/MariaDB. Он инкапсулирует открытие соединения, выбор SQL-запроса по имени, выполнение операций чтения и записи, а также получение страниц данных для таблиц интерфейса. Такое разделение снижает связность маршрутов Flask с деталями подключения к базе данных:

обработчик страницы получает подготовленный набор аргументов, передает их в слой доступа к данным и возвращает результат в шаблон.

SQL-запросы вынесены из обработчиков в отдельную конфигурацию. Благодаря этому программный код не перегружается длинными запросами, а физическая модель остается читаемой: таблицы и связи описываются в схеме базы данных, запросы отвечают за выборку нужного представления, а обработчики страниц управляют последовательностью действий пользователя.

```
def fetch_page(self, query_name, params=None, page=1, per_page=25):
    params = dict(params or {})
    offset = max(page - 1, 0) * per_page
    sql = self._get_sql(query_name)
    sql = f"{sql}\nLIMIT %(limit)s OFFSET %(offset)s"
    params.update({"limit": per_page, "offset": offset})
    return self._run_with_retry(lambda: self.fetch_all(sql, params))
```

Рисунок 6. Фрагмент кода функции fetch_page()

Функциональная часть веб-системы разделена на модули по назначению. Модуль app.py выполняет сборку Flask-приложения, подключение маршрутов и формирование главной страницы. Модуль equipment.py содержит операции с карточками оборудования и версиями. Модуль directories.py реализует работу со справочниками атрибутов. Модуль movements.py отвечает за журнал перемещений, а модуль reports.py формирует отчетные представления и цепочки изменений. Вспомогательные преобразования вынесены в helpers.py, чтобы единообразно обрабатывать пустые значения, даты, числовые поля и человекочитаемые названия моделей или помещений.

Главная страница реализована как сводный экран физической модели. Она не создает новые данные, а выбирает агрегированные показатели по актуальным карточкам оборудования, статусам, категориям и перемещениям. Такой экран нужен для быстрой проверки наполненности базы и общего состояния инвентарного фонда. Скриншот страницы dashboard.html представлен на рисунке 7, фрагмент кода функции dashboard() представлен на рисунке 8.

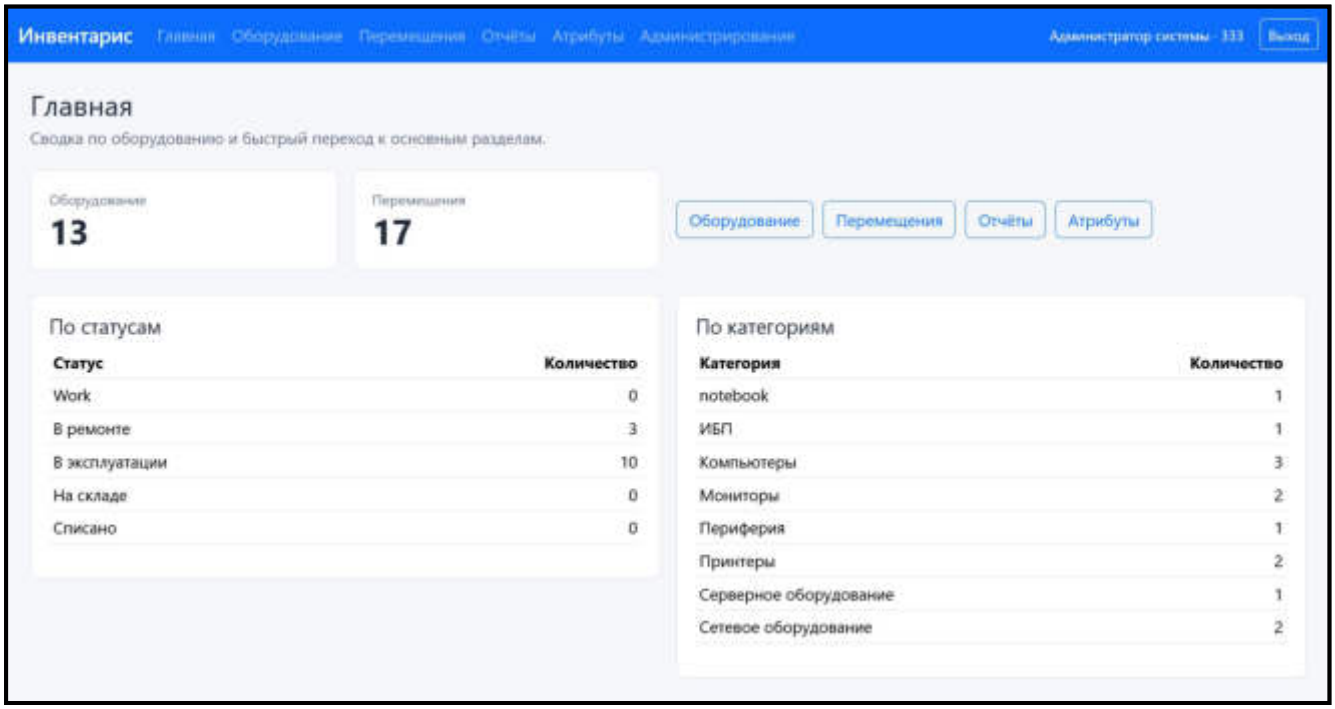


Рисунок 7. Скриншот функции dashboard.html

```
def dashboard():
    db = get_database()
    totals = db.fetch_one("dashboard_totals") or {}
    status_rows = db.fetch_all("dashboard_by_status")
    category_rows = db.fetch_all("dashboard_by_category")
    return render_template(
        "dashboard.html",
        totals=totals,
        status_rows=status_rows,
        category_rows=category_rows,
    )
```

Рисунок 8. Фрагмент кода функции рендера dashboard()

Основной рабочий раздел физической модели связан с таблицей equipment_items и связанными с ней справочниками. Функция list_equipment() выводит только актуальные карточки оборудования, применяет условия поиска по инвентарному номеру, модели, категории, статусу и помещению, а также использует постраничную выборку. Фрагмент кода шаблонизатора страницы equipment_list.html представлен на рисунке 9, фрагмент кода функции list_equipment() представлен на рисунке 10.

```
{% extends "base.html" %}
{% block title %}Оборудование · Инвентарис{% endblock %}
{% block content %}
<div class="page-header">
    <div>
        <h1>Оборудование</h1>
        <p>В списке отображаются только актуальные версии карточек оборудования.</p>
    </div>
    {% if has_permission("311") %}
        <a class="btn btn-primary" href="{{ url_for('equipment.create_equipment') }}">Добавить</a>
    {% endif %}
</div>
<div class="section-panel mb-3">
    <form method="get" class="row g-3 align-items-end">
        <div class="col-md-2">
            <label class="form-label">Инв. номер</label>
            <input class="form-control" name="inventory_number" value="{{ filters.inventory_number or '' }}">
        </div>
        <div class="col-md-2">
            <label class="form-label">Модель</label>
            <input class="form-control" name="model" value="{{ filters.model or '' }}">
        </div>
```

Рисунок 9. Фрагмент кода шаблонизатора страницы equipment_list.html

```
def list_equipment():
    args = read_equipment_filter_args(request.args)
    page = to_int(request.args.get("page"), default=1)
    rows = db.fetch_page("equipment_actual_filtered", args, page, PER_PAGE)
    total = db.fetch_one("equipment_actual_count", args)["cnt"]
    refs = load_equipment_references()
    return render_template("equipment/list.html", rows=rows,
                           refs=refs, args=args, page=page, total=total)
```

Рисунок 10. Фрагмент кода функции рендера list_equipment()

Создание новой инвентарной единицы реализовано через функцию create_equipment(). Перед записью значения из формы приводятся к типам физической модели: идентификаторы справочников преобразуются в целые числа, цена — в десятичное значение, даты — к формату даты. После успешной проверки запись сохраняется в таблицу equipment_items как первая актуальная версия карточки. Скриншот страницы equipment_form.html представлен на рисунке 11, фрагмент кода функции create_equipment() представлен на рисунке 12.

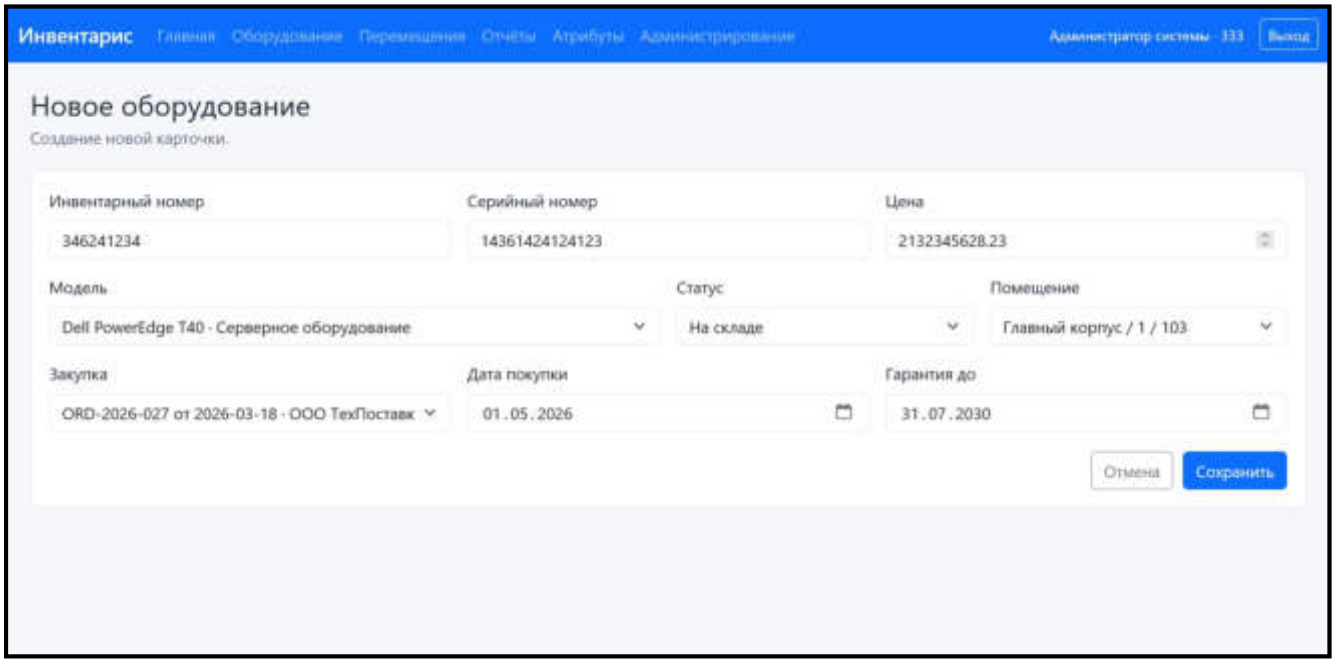


Рисунок 11. Скриншот страницы equipment_form.html

```
def create_equipment():
    refs = load_equipment_references()
    if request.method == "POST":
        data, errors = read_equipment_form(request.form, refs)
        if not errors:
            data["created_by"] = session["user_id"]
            db.execute_write("equipment_insert", data)
            flash("Оборудование добавлено", "success")
            return redirect(url_for("equipment.list_equipment"))
    return render_template("equipment/form.html", refs=refs, item=None)
```

Рисунок 12. Фрагмент кода функции create_equipment()

Справочники физической модели реализованы единым механизмом. Для каждого вида справочника задается конфигурация полей, обязательных значений и SQL-запросов. За счет этого категории, статусы, помещения, модели оборудования и поставщики обрабатываются не набором разрозненных экранов, а общей функцией, которая подставляет нужную конфигурацию выбранного справочника. Скриншот страницы directory_list.html представлен на рисунке 13, фрагмент кода функции list_directory() представлен на рисунке 14.

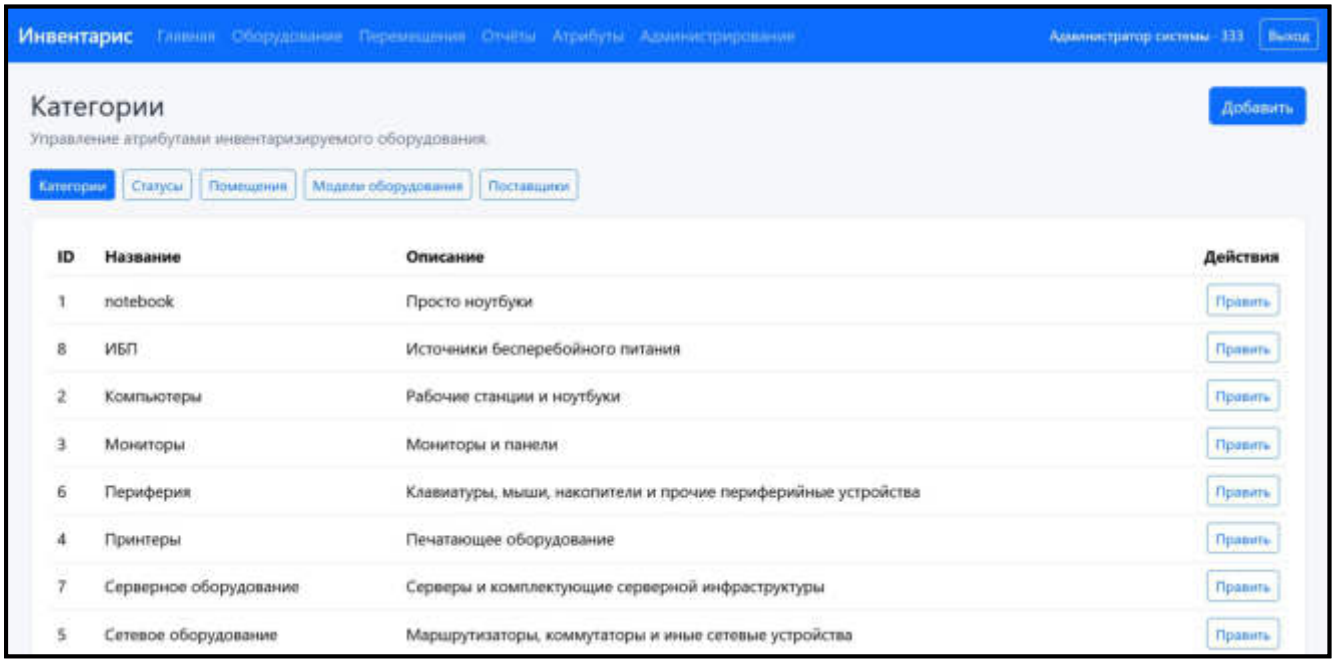


Рисунок 13. Скриншот страницы directory_list.html

```
def list_directory(kind):
    cfg = get_directory(kind)
    rows = db.fetch_all(cfg["select_query"])
    return render_template("directories/list.html", kind=kind, cfg=cfg, rows=rows)

def read_directory_form(kind, form):
    cfg = get_directory(kind)
    data = {field: empty_to_none(form.get(field)) for field in cfg["fields"]}
    validate_required(data, cfg["required"])
    return data
```

Рисунок 14. Фрагмент кода функции list_directory()

Перемещение оборудования реализовано как связанная операция над двумя элементами физической модели. Сначала определяется актуальная карточка оборудования и новое помещение, затем создается новая версия карточки с обновленной ссылкой на locations, после чего в item_movements записывается факт операции. Такой порядок сохраняет текущее состояние оборудования и одновременно фиксирует самостоятельную запись журнала. Скриншот фрагмент кода шаблонизатора страницы movement_form.html представлен на рисунке 15, фрагмент кода функции create_movement() представлен на рисунке 16.

```
{% extends "base.html" %}
{% block title %}Новое перемещение · Инвентарис{% endblock %}
{% block content %}
<div class="page-header">
    <div>
        <h1>Новое перемещение</h1>
        <p>Операция доступна только для актуальных версий оборудования. При перемещении создаётся новая версия карточки.</p>
    </div>
</div>
<div class="section-panel">
    <form method="post" class="row g-3">
        <div class="col-md-6">
            <label class="form-label">Оборудование</label>
            <select class="form-select" name="item_id" required>
                <option value="">Выберите оборудование</option>
                {% for item in references["items"] %}
                    <option value="{{ item.id }}" {% if form_data.item_id|string == item.id|string %}selected{% endif %}>{{ item.item_name }}</option>
                {% endfor %}
            </select>
        </div>
        <div class="col-md-3">
            <label class="form-label">Тип операции</label>
            <select class="form-select" name="movement_type" required>
                {% for value, title in references.movement_types %}
                    <option value="{{ value }}" {% if form_data.movement_type == value %}selected{% endif %}>{{ title }}</option>
                {% endfor %}
            </select>
        </div>
```

Рисунок 15. Фрагмент кода шаблонизатора страницы movement_form.html

```
def create_movement():
    refs = load_movement_references()
    if request.method == "POST":
        data = read_movement_form(request.form)
        item = db.fetch_one("equipment_actual_by_id", {"id": data["item_id"]})
        version_id = create_equipment_version(item, {"location_id":
            data["to_location_id"]})
        data.update({"item_id": version_id, "performed_by": session["user_id"]})
        db.execute_write("movement_insert", data)
        flash("Перемещение создано", "success")
        return redirect(url_for("movements.list_movements"))
    return render_template("movements/form.html", refs=refs)
```

Рисунок 16. Фрагмент кода функции create_movement()

Для просмотра цепочки изменений используется связь equipment_items.id_before_update. Функция построения схемы поднимает все версии одной инвентарной единицы, сравнивает соседние состояния и формирует последовательность переходов между ними. Благодаря этому физическая модель используется не только для хранения текущей карточки, но и для восстановления истории изменения атрибутов. Кода шаблонизатора страницы reports_change_diagram.html представлен на рисунке 17, фрагмент кода функции build_version_mermaid() представлен на рисунке 18. На рисунке 19 предоставлен фрагмент JS кода, который позволяет приближать и масштабировать схему изменения версий оборудования.

```
{% extends "base.html" %}
{% block title %}Схема изменений · Инвентарис{% endblock %}
{% block content %}
<div class="page-header">
    <div>
        <h1>Схема изменений</h1>
        <p>{{ item.inventory_number }} · {{ item.manufacturer or '' }} {{ item.model_name }}</p>
    </div>
    <div class="actions">
        <a class="btn btn-outline-secondary" href="{{ url_for('reports.changes_report') }}">К изменениям</a>
        <a class="btn btn-primary" href="{{ url_for('reports.download_change_diagram', item_id=item.id) }}">Скачать .mmd</a>
    </div>
</div>
<div class="section-panel mb-3">
    <h2 class="h5">Предпросмотр</h2>
    <p class="text-muted small mb-2">
        Колесо мыши приближает и отдаляет диаграмму. Зажмите левую кнопку мыши и двигайте диаграмму, чтобы рассмотреть нужный блок.
    </p>
    <div class="diagram-preview" id="diagramPreview">
        <div class="diagram-canvas" id="diagramCanvas">
            <div class="mermaid mb-0">
{{ diagram_code }}
            </div>
        </div>
    </div>
</div>
```

Рисунок 17. Скриншот функции change_diagram()

```
def build_version_mermaid(item_id):
    chain = load_version_chain(item_id)
    lines = ["flowchart LR"]
    for current, next_item in pairwise(chain):
        label = build_mermaid_edge_label(current, next_item)
        lines.append(f"V{current['id']} -->|{label}| V{next_item['id']}")
    return "\n".join(lines)
```

Рисунок 18. Фрагмент кода функции build_version_mermaid()

```
import mermaid from "https://cdn.jsdelivr.net/npm/mermaid@10/dist/mermaid.esm.min.mjs";
mermaid.initialize({ startOnLoad: false });
await mermaid.run({ querySelector: ".mermaid" });
const preview = document.getElementById("diagramPreview");
const canvas = document.getElementById("diagramCanvas");
const svg = canvas.querySelector("svg");
const svgBox = svg.viewBox.baseVal;
const baseWidth = svgBox?.width || svg.getBBox().width || svg.getBoundingClientRect().width;
const baseHeight = svgBox?.height || svg.getBBox().height || svg.getBoundingClientRect().height;
const minScale = 0.05;
let maxScale = 12;
const state = {
    scale: 1,
    x: 0,
    y: 0,
    dragging: false,
    startX: 0,
    startY: 0,
    pointerX: 0,
    pointerY: 0,
};
function applyTransform() {
    svg.style.width = `${Math.round(baseWidth * state.scale)}px`;
    svg.style.height = `${Math.round(baseHeight * state.scale)}px`;
    canvas.style.left = `${Math.round(state.x)}px`;
    canvas.style.top = `${Math.round(state.y)}px`;
}
function clampScale(value) {
    return Math.min(maxScale, Math.max(minScale, value));
}
function fitDiagram() {
    const rect = preview.getBoundingClientRect();
    const fitScale = Math.min(rect.width / baseWidth, rect.height / baseHeight, 1);
    state.scale = clampScale(fitScale);
    maxScale = Math.max(16, state.scale * 80);
    state.x = (rect.width - baseWidth * state.scale) / 2;
    state.y = (rect.height - baseHeight * state.scale) / 2;
    applyTransform();
}
preview.addEventListener("wheel", (event) => {
    event.preventDefault();
    const rect = preview.getBoundingClientRect();
    const pointerX = event.clientX - rect.left;
    const pointerY = event.clientY - rect.top;
    const previousScale = state.scale;
    const nextScale = clampScale(previousScale * (event.deltaY < 0 ? 1.12 : 0.88));
    const diagramX = (pointerX - state.x) / previousScale;
    const diagramY = (pointerY - state.y) / previousScale;
    state.x = pointerX - diagramX * nextScale;
    state.y = pointerY - diagramY * nextScale;
    state.scale = nextScale;
    applyTransform();
}, { passive: false });
```

Рисунок 19. Фрагмент кода JS функци масштабирования схемы

В результате реализации физической модели была создана структура базы данных, обеспечивающая хранение справочной информации, учет карточек оборудования, связь оборудования с закупками, помещениями, поставщиками, пользователями и журналом перемещений. Таблицы базы данных приведены к реляционному виду, имеют первичные ключи, внешние связи и поля, необходимые для контроля актуальности записей.

Программная реализация связана с физической моделью через выделенный слой доступа к данным и набор маршрутов Flask. Основные операции системы выполняются над таблицами базы данных без ручного обращения пользователя к СУБД: данные создаются, просматриваются и изменяются через веб-интерфейс, а обработчики приложения преобразуют пользовательские действия в SQL-операции. Версионный подход к карточкам оборудования позволяет сохранять историю изменений и использовать ее при построении отчетных и аналитических представлений.

Глава 3. Организация безопасности сайта (веб-приложения)

Организация безопасности веб-системы «Инвентарис» направлена на предотвращение несанкционированного доступа к данным, сохранение целостности учетной информации и исключение выполнения пользователем операций, не соответствующих его роли. Так как система предназначена для учета оборудования предприятия, основными объектами защиты являются карточки инвентарных единиц, справочники, сведения о пользователях, журнал перемещений и отчетные данные. Нарушение доступа к этим сведениям может привести не только к раскрытию внутренней информации, но и к искажению фактического состояния фонда оборудования.

Первоначальным этапом организации безопасности является выделение угроз, актуальных для автоматизированной информационной системы. Для системы «Инвентарис» наиболее существенными являются угрозы доступности, целостности и конфиденциальности данных. Данные угрозы приведены в таблице 14.

Таблица 14. Основные угрозы безопасности веб-системы

Вид угрозы	Содержание угрозы	Возможные последствия
Угроза доступности	Затруднение или невозможность обращения пользователя к разрешенным разделам системы.	Пользователь не может своевременно получить сведения об оборудовании, перемещениях и отчетных данных.
Угроза целостности	Несанкционированное изменение, удаление или искажение учетных сведений.	Карточки оборудования, статусы, помещения и журнальные записи перестают отражать фактическое состояние объекта учета.
Угроза конфиденциальности	Получение доступа к сведениям пользователем, которому данные не предназначены по роли.	Внутренние сведения о составе оборудования, закупках, пользователях и перемещениях становятся доступны посторонним лицам.

Наиболее опасной для разрабатываемого приложения является ситуация, при которой пользователь получает доступ к функционалу не через штатные элементы интерфейса, а путем прямого обращения к адресу маршрута. Поэтому ограничения реализованы не только на уровне отображения пунктов меню, но и на уровне обработчиков маршрутов. Даже если ссылка будет введена вручную в адресной строке браузера, серверная часть приложения выполняет проверку полномочий и при недостатке прав возвращает отказ в доступе.

В системе «Инвентарис» применяется специально разработанная ролевая модель разграничения доступа ESR. Права пользователя хранятся в роли и задаются строкой из трех цифр. Каждая позиция строки отвечает за отдельный функциональный контур приложения: E — оборудование и связанные операции, S — системный контур, R — отчеты. Такая модель позволяет компактно описывать права пользователя без создания большого количества разрозненных флагов.

Каждая цифра в строке ESR принимает значение от 1 до 3. Значение 1 означает отсутствие доступа к соответствующему контуру, значение 2 означает право чтения, значение 3 означает право записи. При этом право записи фактически включает возможность чтения, так как пользователь, выполняющий изменение, должен видеть изменяемые данные. Уровни прав приведены в таблице 15.

Таблица 15. Уровни прав ESR

Цифра	Значение	Практический смысл
1	Нет прав	Пользователь не имеет доступа к разделу или действию соответствующего контура. При прямом обращении к закрытому маршруту возвращается отказ в доступе.
2	Чтение	Пользователь может просматривать списки, карточки, журналы и отчеты, если соответствующая операция не изменяет данные.
3	Запись	Пользователь может создавать и изменять сведения в пределах соответствующего контура, если операция предусмотрена приложением.

Строка прав читается слева направо. Например, значение 211 означает, что пользователь имеет право чтения в контуре оборудования, но не имеет доступа к системному и отчетному контуру. Значение 311 разрешает создание и изменение карточек оборудования и перемещений, но не дает прав на системное администрирование и отчеты. Значение 333 соответствует полному доступу ко всем основным контурам приложения.

Таблица 16. Назначение позиций в строке ESR

Позиция	Обозначение	Функциональный контур	Примеры защищаемых действий
1	E	Оборудование	Просмотр списка оборудования, открытие карточки, создание карточки, редактирование, смена статуса, оформление перемещения.
2	S	Система	Работа со справочниками, пользователями, ролями и служебными сущностями, необходимыми для ведения учета.
3	R	Отчеты	Просмотр отчетов, фильтрация отчетных данных, просмотр изменений, выгрузка отчетных сведений.

Выбранная структура прав удобна тем, что проверка выполняется посимвольно. Для каждого endpoint-а задается минимальная требуемая строка. Если хотя бы одна цифра прав пользователя меньше соответствующей цифры требования, действие блокируется. Благодаря этому одна и та же функция проверки подходит для всех разделов приложения: оборудования, справочников, пользователей, ролей и отчетов.

Работа с защищенными разделами начинается после авторизации пользователя. Пользователь вводит логин и пароль, после чего приложение определяет учетную запись, ее активность, назначенную роль и строку прав ESR. Пароль в базе данных не хранится в открытом виде: в таблице пользователей сохраняется хэш пароля. Это снижает риск раскрытия паролей при получении доступа к данным таблицы users.

Нормализованная проверка выполняется функцией has_permissions. Функция получает текущие права пользователя и требуемую строку прав, приводит значения к единому виду, после чего сравнивает каждую позицию. Если текущая цифра меньше требуемой, возвращается запрет. Фрагмент кода проверки приведен на рисунке 20.

```
def has_permissions(user_permissions: Any, required_permissions: str) -> bool:
    current = permissions_value(user_permissions)
    required = permissions_value(required_permissions)
    for current_digit, required_digit in zip(current, required):
        if int(current_digit) < int(required_digit):
            return False
    return True
```

Рисунок 20. Фрагмент кода функции has_permissions

Проверка построена на сравнении каждой цифры ESR, поэтому она одинаково корректно обрабатывает как чтение, так и запись. Например, пользователь с правами 311 пройдет проверку для требования 211, так как запись по первому контуру выше чтения. При этом тот же пользователь не пройдет проверку для требования 131, поскольку во втором контуре значение 1 ниже требуемого значения 3.

Для повышения читаемости кода обработчиков используется промежуточная функция can (см. рис. 22). Она не содержит отдельной логики безопасности, а служит компактной прослойкой для вызова общей проверки прав. Это позволяет в декораторе и шаблонах использовать короткое условие и не дублировать алгоритм сравнения ESR в разных частях приложения. Централизация проверки снижает риск случайного расхождения правил доступа при сопровождении кода.

Для защиты маршрутов Flask используется декоратор require_permissions. Он указывается перед функцией endpoint-а (см. рис. 21) и задает минимально допустимые права для выполнения операции. Декоратор объединяет две проверки: наличие авторизованного пользователя и соответствие его прав требуемой строке ESR. Если пользователь не авторизован, маршрут закрывается механизмом login_required. Если пользователь авторизован, но его права ниже требуемых, выполнение обработчика прерывается ответом 403.

Пример объявления защищенного маршрута приведен ниже. В данном случае создание пользователя относится к системному контуру и требует прав 131, то есть отсутствия требований к оборудованию, права записи в системном контуре и отсутствия требований к отчетам.

```
@admin_bp.route("/users/create", methods=["GET", "POST"])
@require_permissions("131")
def user_create() -> Any:
```

Рисунок 21. Пример подключения декоратора к endpoint-у создания пользователя

Сам декоратор получает требуемую строку прав и формирует обертку над защищаемой функцией. Перед вызовом исходного обработчика выполняется проверка can. Если проверка возвращает отрицательный результат, приложение вызывает abort(403), и основная логика endpoint-а не выполняется. Это важно, потому что пользователь без прав не просто не видит форму, а вообще не доходит до кода создания или изменения данных.

```
def require_permissions(required_permissions: str) -> Callable[[Callable[..., Any]], Callable[..., Any]]:
    def decorator(view_func: Callable[..., Any]) -> Callable[..., Any]:
        @wraps(view_func)
        @login_required
        def wrapper(*args: Any, **kwargs: Any) -> Any:
            if not can(required_permissions):
                abort(403)
            return view_func(*args, **kwargs)
        return wrapper
    return decorator
```

Рисунок 22. Фрагмент кода декоратора require_permissions

Использование декораторов удобно тем, что правило доступа находится рядом с маршрутом. При просмотре исходного кода сразу видно, какие права требуются для открытия endpoint-а. Это снижает вероятность ошибки при сопровождении приложения: разработчику не нужно искать проверку внутри тела функции, а проверяющий может сразу определить, каким механизмом ограничен доступ к маршруту.

Доступ к endpoint-ам системы распределен по функциональным контурам. Операции просмотра требуют уровня чтения, а операции создания и изменения требуют уровня записи. Разделы, относящиеся к пользователям, ролям и справочникам, вынесены в системный контур, поскольку их изменение влияет на структуру и правила работы всей системы. Отчеты вынесены в отдельный контур, так как пользователь может нуждаться в отчетных данных без права изменять карточки оборудования или справочники.

В таблице 17 приведено распределение прав для основных групп endpoint-ов. Таблица отражает прикладную логику разграничения доступа и используется как краткая схема проверки маршрутов.

Таблица 17. Требуемые права для основных endpoint-ов системы

Раздел	Основные endpoint-ы	Права	Разрешаемое действие
Главная страница	/	Авторизованный пользователь	Просмотр краткой сводки по оборудованию и переход к доступным разделам.
Авторизация	/login, /logout	Без ESR для входа; авторизация для выхода	Вход в систему и завершение сеанса пользователя.
Оборудование: просмотр	/equipment, /equipment/<id>	211	Просмотр списка актуального оборудования и детальной карточки.
Оборудование: изменение	/equipment/create, /equipment/<id>/edit, /equipment/<id>/status	311	Создание карточки, редактирование актуальной версии и смена статуса.
Перемещения: просмотр	/movements	211	Просмотр журнала перемещений и связанных записей оборудования.
Перемещения: создание	/movements/create	311	Создание записи перемещения и обновление актуального местонахождения оборудования.
Справочники: просмотр	/directories/<name>	121	Просмотр категорий, статусов, помещений, моделей и поставщиков.
Справочники: изменение	/directories/<name>/create, /directories/<name>/<id>/edit	131	Создание и редактирование справочных записей.
Пользователи и роли	/admin/users, /admin/roles	131	Просмотр и изменение учетных записей, ролей и назначенных прав.
Отчеты: просмотр	/reports, /reports/movements, /reports/changes	112	Просмотр отчетов, применение фильтров и анализ данных без изменения записей.
Отчеты: выгрузка	/reports/equipment.csv	113	Формирование CSV-файла с отчетными сведениями по выбранным фильтрам.

Наиболее показательной является разница между требованиями 112 и 113. Пользователь с правом чтения отчетов может просматривать отчетные страницы, но не обязательно должен получать право на выгрузку данных. Для выгрузки

используется уровень записи в отчетном контуре, поскольку формирование файла рассматривается как отдельное действие по выдаче данных из системы.

Пользователь с правами 311 может вести карточки оборудования и перемещения, но не получает автоматически доступа к системным справочникам, пользователям и отчетам. Это позволяет создать роль оператора учета, который выполняет практическую работу с оборудованием, но не управляет ролями и не получает расширенные отчетные срезы. Пользователь с правами 222, напротив, может читать сведения в основных контурах, но не может создавать или изменять записи.

В системе выделяются роли, которым назначается конкретная строка ESR. Название роли используется для удобства администратора и пользователя, а фактическое решение о доступе принимает строка прав. Поэтому две роли с разными названиями, но одинаковым кодом ESR будут иметь одинаковые технические возможности. Такой подход упрощает сопровождение: можно переименовывать роли и уточнять их описание без изменения алгоритма проверки.

Типовое распределение ролей строится от минимально необходимого доступа. Полные права назначаются только администратору, так как значение 333 открывает все основные контуры. Оператору учета достаточно прав 311, поскольку его задачи связаны с карточками оборудования и перемещениями. Пользователю, которому необходимы только сведения и отчетные данные, может быть назначена роль с правами чтения, например 222 или иная близкая комбинация, определенная администратором. Ознакомиться с подробным разбором примеров ролей и их прав можно в таблице 18.

Таблица 18. Примеры пользовательских ролей и доступных действий

Условная роль	Пример прав	Доступные действия	Ограничения
Администратор	333	Полный доступ к оборудованию, системному контуру и отчетам.	Роль должна назначаться ограниченному числу пользователей.
Оператор учета	311	Создание и редактирование оборудования, смена статуса, оформление перемещений.	Нет доступа к системному администрированию и отчетному контуру.
Пользователь просмотра	222	Просмотр карточек, справочников и отчетов без изменения данных.	Нельзя создавать, редактировать или выгружать данные, если для действия требуется запись.
Пользователь отчетов	112	Просмотр отчетных страниц и фильтрация отчетных данных.	Нет доступа к изменению оборудования и системных сущностей.

Дополнительно в системе реализованы ограничения, связанные не только с ролью пользователя, но и с состоянием данных. Например, учетная запись может быть деактивирована. Деактивация блокирует вход пользователя, но не удаляет запись из базы данных, благодаря чему сохраняются связи с ранее выполненными действиями. Это важно для целостности истории: исполнитель операции остается известным даже после прекращения доступа.

Для административного контура предусмотрена защита последнего активного администратора. Система не должна допустить ситуацию, при которой единственный пользователь с полными правами будет отключен или переведен на роль без доступа 333. Такое ограничение относится к организационной безопасности приложения: оно предотвращает потерю управления системой из-за ошибочного изменения роли.

Для карточек оборудования действует отдельное ограничение: изменять допускается только актуальную версию записи. Архивная карточка сохраняется для истории и не должна редактироваться как текущая. Это решение защищает целостность цепочки версий, поскольку старые состояния оборудования не затираются последующими исправлениями.

Помимо проверки прав, в приложении используются дополнительные прикладные ограничения, направленные на сохранение корректности данных. Они не заменяют модель ESR, но дополняют ее. Пользователь может иметь право выполнить действие, однако конкретная операция все равно должна пройти

проверку формы, ссылочных значений и состояния объекта. Это позволяет снизить вероятность внесения некорректных или противоречивых сведений в базу данных.

К числу таких мер относится проверка обязательных полей. При создании пользователя, роли, оборудования или справочной записи система не должна принимать неполные данные, если они требуются для дальнейшей работы. Для дат и числовых значений выполняется приведение формата, чтобы в базу данных не попадали произвольные строки. Для справочных ссылок проверяется существование выбранной записи, поскольку пользователь не должен иметь возможность передать несуществующий идентификатор через измененную форму.

Для учетных записей контролируется уникальность логина. Это исключает конфликт при авторизации и предотвращает неоднозначность при связывании действий с пользователем. Для карточек оборудования контролируется уникальность актуального инвентарного номера. Благодаря этому в системе не появляются две текущие карточки с одним инвентарным идентификатором, что особенно важно для инвентаризации.

Отдельной мерой защиты целостности является версионность карточек оборудования. При изменении значимых сведений создается новая версия записи, а предыдущая остается в базе как архивная. В результате изменение не уничтожает прежнее состояние объекта. Такой подход повышает устойчивость системы к ошибочным действиям пользователя и позволяет восстановить цепочку изменений при анализе спорной ситуации.

Таблица 19. Прикладные меры защиты данных

Мера	Назначение	Результат применения
Авторизация пользователя	Ограничение доступа к разделам системы только для пользователей с учетной записью.	Неавторизованный пользователь не получает доступ к защищенным функциям.
Проверка ESR	Сравнение прав пользователя с минимальными правами endpoint-а.	Пользователь выполняет только те действия, которые разрешены его ролью.
Хранение хэша пароля	Исключение хранения пароля в открытом виде.	Снижается риск раскрытия паролей при доступе к данным учетных записей.
Деактивация без удаления	Блокировка входа пользователя с сохранением исторических связей.	Сохраняется целостность записей, где пользователь указан исполнителем или автором.
Проверка актуальности карточки	Запрет изменения архивных версий оборудования.	История изменений не искажается последующими правками.

Мера	Назначение	Результат применения
Валидация форм	Контроль обязательных полей, дат, чисел и справочных ссылок.	В базу данных попадают только данные, пригодные для дальнейшей обработки.

При недостатке прав приложение возвращает пользователю страницу отказа в доступе с кодом 403. Использование данного кода важно с точки зрения логики веб-приложения: запрос пользователя был понятен серверу, однако выполнение действия запрещено. Это отличается от ситуации несуществующего адреса, когда возвращается ошибка 404.

Обработка отказа выполняется до вызова основной функции endpoint-а. Поэтому при попытке открыть закрытый адрес не происходит частичного выполнения операции. Например, пользователь без прав 131 не может создать пользователя через прямой POST-запрос к endpoint-у создания учетной записи, потому что декоратор прервет выполнение до чтения и сохранения формы.

Такой механизм также упрощает сопровождение приложения. Вся логика отказа сосредоточена в одном месте: маршрут сообщает требуемые права, декоратор проверяет доступ, а обработчик 403 показывает пользователю результат. За счет этого код endpoint-ов остается сосредоточенным на бизнес-операции, а не на повторяющихся проверках прав в каждой функции.

В результате организации безопасности веб-системы «Инвентарис» реализована прикладная модель разграничения доступа, основанная на ролях и трехпозиционном коде ESR. Каждая роль определяет уровень доступа к контурам оборудования, системных функций и отчетов. Проверка прав выполняется серверной частью приложения при обращении к endpoint-ам, что исключает зависимость безопасности от элементов интерфейса и блокирует прямой доступ к закрытым маршрутам.

Основными средствами защиты являются авторизация пользователя, хэширование паролей, декораторы проверки доступа, возврат ошибки 403, деактивация учетных записей без удаления истории, защита последнего администратора, проверка актуальности карточек оборудования и валидация пользовательских данных. Перечисленные меры снижают вероятность несанкционированного изменения данных, случайной потери управляемости и нарушения целостности учетной информации.

Глава 4. Тестирование и проверка работоспособности

Для подтверждения правильности работы веб-системы «Инвентарис» было проведено функциональное тестирование основных пользовательских сценариев. Тестирование необходимо для проверки соответствия фактического поведения приложения ожидаемому результату, выявления ошибок при вводе данных, проверки корректности переходов между разделами и подтверждения того, что реализованные функции выполняют поставленные задачи.

Тестирование программного обеспечения представляет собой процесс выполнения программы в заранее определенных условиях с последующим сравнением фактического результата с ожидаемым. В рамках данной главы проверялись не внутренние технические характеристики серверной среды, а прикладная работоспособность системы: авторизация пользователя, отображение доступных разделов, работа с карточками оборудования, справочниками, перемещениями и отчетами.

Основным методом проверки выбран функциональный подход, при котором каждая операция рассматривается как отдельный пользовательский сценарий. Такой подход позволяет проверить работу системы с точки зрения конечного пользователя и убедиться, что ввод данных, обработка формы, сохранение сведений и вывод результата выполняются согласованно.

Для фиксации результатов применялся тестовый случай. Тестовый случай представляет собой описание конкретной проверки, включающее исходные условия, последовательность действий пользователя, ожидаемый результат, фактический результат и итоговый статус выполнения. Использование единого шаблона делает проверки сопоставимыми между собой и позволяет быстро определить, какая функция была проверена и чем завершилось выполнение сценария.

При составлении тестовых случаев использовались контрольные учетные записи с различными правами доступа. Для сценариев изменения данных применялась учетная запись с правами записи в соответствующем контуре. Для проверки ограничений доступа использовалась учетная запись с недостаточными правами. Тестовые записи оборудования, помещений и моделей рассматривались

как контрольные данные и не предназначались для постоянного использования в рабочем реестре.

Структура тестового случая, использованная при проверке работоспособности системы, представлена в таблице 20.

Таблица 20. Шаблон тестового случая

Id теста	**Название теста**	**Краткое описание теста**	**Условия выполнения**
ТС-XX	Наименование проверки	Описание проверяемого действия	Роль, исходные данные и предварительные условия
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
Пошаговое выполнение проверки	Результат, который должен быть получен	Результат, полученный при выполнении	Успешно / завершилось ошибкой

Расшифровка параметров шаблона приведена в таблице 21. В ходе тестирования каждый параметр заполнялся до начала выполнения проверки или непосредственно после получения фактического результата.

Таблица 21. Расшифровка шаблона тестового случая

Параметр	**Описание**
Id теста	Уникальный идентификатор тестового случая, позволяющий ссылаться на конкретную проверку в тексте и таблицах результатов.
Название теста	Краткое наименование проверяемой функции или пользовательского действия.
Краткое описание теста	Описание цели проверки и ожидаемого смысла выполняемого сценария.
Условия выполнения	Исходные данные, используемая роль пользователя, предварительно созданные справочники или карточки, необходимые для выполнения проверки.
Этапы теста	Последовательность действий пользователя в веб-интерфейсе системы.
Ожидаемый результат	Состояние интерфейса или данных, которое должно быть получено при корректной работе приложения.
Фактический результат	Результат, зафиксированный при выполнении тестового случая.
Статус	Итоговая оценка выполнения проверки: успешно, если фактический результат совпал с ожидаемым, или завершилось ошибкой, если обнаружено несоответствие.

Ниже приведены тестовые случаи, сформированные для проверки основных функций веб-системы «Инвентарис». В качестве результата указывается состояние системы, полученное после выполнения описанных действий.

Таблица 22. Тестовый случай ТС-01

Id теста	**Название теста**	**Краткое описание теста**	**Условия выполнения**
ТС-01	Авторизация с неверными учетными данными	Проверка отказа во входе при вводе неверного логина или пароля.	Открыта страница входа. Используются несуществующие или некорректные учетные данные.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть страницу входа в систему. 2. Ввести неверный логин или неверный пароль. 3. Нажать кнопку «Войти».	Авторизация не выполняется. Пользователь остается на странице входа. Отображается сообщение об ошибке авторизации.	Авторизация не выполнена. Страница входа осталась открытой. Сообщение об ошибке отображено.	Успешно.

Таблица 23. Тестовый случай ТС-02

Id теста	**Название теста**	**Краткое описание теста**	**Условия выполнения**
ТС-02	Авторизация с правильными учетными данными	Проверка входа пользователя при вводе корректного логина и пароля.	Открыта страница входа. Создана активная учетная запись пользователя с назначенной ролью.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть страницу входа в систему. 2. Ввести корректный логин. 3. Ввести корректный пароль. 4. Нажать кнопку «Войти».	Пользователь перенаправляется на главную страницу. В верхней панели отображаются имя пользователя и код прав ESR.	Главная страница открыта. Имя пользователя и код ESR отображаются в верхней панели.	Успешно.

Таблица 24. Тестовый случай ТС-03

Id теста	**Название теста**	**Краткое описание теста**	**Условия выполнения**
ТС-03	Проверка ограничения доступа к разделу администрирования	Проверка блокировки перехода к системному разделу при недостаточных правах пользователя.	Пользователь авторизован под ролью без права записи в системном контуре, например с правами 311 или 222.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Авторизоваться под учетной записью с недостаточными системными правами. 2. Выполнить прямой переход к странице списка пользователей или создания пользователя. 3. Дождаться ответа приложения.	Система не открывает защищенный раздел. Пользователю отображается страница отказа в доступе с кодом 403.	Переход заблокирован. Открыта страница 403.	Успешно.

Таблица 25. Тестовый случай ТС-04

Id теста	**Название теста**	**Краткое описание теста**	**Условия выполнения**
ТС-04	Фильтрация списка оборудования	Проверка поиска актуальных карточек оборудования по значениям фильтров.	Пользователь авторизован. В системе существует не менее одной актуальной карточки оборудования.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть раздел «Оборудование». 2. Ввести часть инвентарного номера или выбрать статус. 3. Нажать кнопку «Найти».	В таблице отображаются только актуальные карточки оборудования, соответствующие заданным условиям фильтрации.	Список оборудования перестроен по заданному условию. Посторонние записи в таблице отсутствуют.	Успешно.

Таблица 26. Тестовый случай ТС-05

Id теста	Название теста	Краткое описание теста	Условия выполнения
ТС-05	Создание карточки оборудования с корректными данными	Проверка добавления новой инвентарной единицы через форму создания оборудования.	Пользователь имеет право записи по контуру оборудования. В справочниках существуют модель и статус.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть раздел «Оборудование». 2. Нажать кнопку «Добавить». 3. Заполнить инвентарный номер, модель и статус. 4. При необходимости заполнить серийный номер, помещение, цену и даты. 5. Нажать кнопку сохранения.	Карточка оборудования создается. Пользователь возвращается к списку или карточке записи. Новая инвентарная единица отображается в списке оборудования.	Карточка оборудования создана и отображается в списке актуальных записей.	Успешно.

Таблица 27. Тестовый случай ТС-06

Id теста	Название теста	Краткое описание теста	Условия выполнения
ТС-06	Создание карточки оборудования без обязательных полей	Проверка валидации формы создания оборудования при отсутствии обязательных сведений.	Пользователь имеет право записи по контуру оборудования. Открыта форма создания оборудования.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть форму создания оборудования. 2. Не заполнять инвентарный номер, модель или статус. 3. Нажать кнопку сохранения.	Система не создает запись. Пользователь остается на форме и получает предупреждение о необходимости заполнить обязательные поля.	Запись не создана. Форма осталась открытой, предупреждение о некорректном заполнении отображено.	Успешно.

Таблица 28. Тестовый случай ТС-07

Id теста	Название теста	Краткое описание теста	Условия выполнения
ТС-07	Редактирование оборудования с созданием новой версии	Проверка сохранения истории изменения карточки оборудования при редактировании актуальной записи.	Пользователь имеет право записи по контуру оборудования. В системе существует актуальная карточка оборудования.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть актуальную карточку оборудования. 2. Нажать кнопку «Редактировать». 3. Изменить одно или несколько полей, например статус, помещение или цену. 4. Сохранить форму. 5. Открыть карточку после сохранения.	Система создает новую версию карточки. Предыдущая запись становится архивной, а новая версия отображается как актуальная.	Новая версия карточки создана. Ссылка на предыдущую версию отображается, актуальные сведения обновлены.	Успешно.

Таблица 29. Тестовый случай ТС-08

Id теста	Название теста	Краткое описание теста	Условия выполнения
ТС-08	Создание перемещения оборудования	Проверка фиксации перемещения оборудования между помещениями.	Пользователь имеет право записи по контуру оборудования. В системе существует актуальное оборудование и не менее двух помещений.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть раздел «Перемещения». 2. Нажать кнопку создания перемещения. 3. Выбрать оборудование, тип операции и новое помещение. 4. Заполнить примечание при необходимости. 5. Сохранить перемещение.	В журнале перемещений появляется новая операция. В актуальной карточке оборудования меняется помещение, при этом создается новая версия карточки.	Операция появилась в журнале. Помещение в актуальной карточке изменено, новая версия карточки создана.	Успешно.

Таблица 30. Тестовый случай ТС-09

Id теста	Название теста	Краткое описание теста	Условия выполнения
ТС-09	Проверка валидации JSON в справочнике моделей	Проверка отказа в сохранении модели оборудования при некорректном заполнении поля характеристик.	Пользователь имеет право записи по системному контуру. Открыта форма создания или редактирования модели оборудования.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть раздел «Атрибуты». 2. Перейти к справочнику моделей оборудования. 3. Открыть форму создания или редактирования модели. 4. В поле характеристик ввести некорректный JSON. 5. Нажать кнопку сохранения.	Запись не сохраняется. Система показывает предупреждение о некорректном формате JSON.	Сохранение заблокировано. Предупреждение о некорректном формате характеристик отображено.	Успешно.

Таблица 31. Тестовый случай ТС-10

Id теста	Название теста	Краткое описание теста	Условия выполнения
ТС-10	Формирование отчета по оборудованию и выгрузка CSV	Проверка построения основного отчета по оборудованию с применением фильтров и выгрузкой результата.	Пользователь имеет право просмотра отчетов. Для выгрузки CSV пользователь имеет право записи по отчетному контуру.
Этапы теста	**Ожидаемый результат**	**Фактический результат**	**Статус**
1. Открыть раздел «Отчеты». 2. Выбрать категорию, статус или помещение. 3. Нажать кнопку «Сформировать». 4. Проверить таблицу отчета и сводные блоки. 5. При наличии права выгрузки нажать кнопку CSV.	Отчет строится по выбранным фильтрам. Таблица и сводки соответствуют текущему набору данных. При выгрузке формируется CSV-файл с данными отчета.	Отчет сформирован по выбранным фильтрам. CSV-файл выгружен с корректным набором полей.	Успешно.

По результатам выполнения тестовых случаев установлено, что основные функции веб-системы «Инвентарис» выполняются в соответствии с ожидаемым поведением. Авторизация корректно различает успешный и неуспешный вход, ограничение доступа блокирует действия пользователя без необходимых прав, формы оборудования и справочников выполняют проверку обязательных и структурированных полей, а операции редактирования и перемещения сохраняют логику версионного учета.

Также подтверждена работоспособность отчетного контура: данные отображаются с учетом выбранных фильтров, а CSV-выгрузка формируется для пользователя с достаточными правами. Обнаруженные в ходе проверки несоответствия в представленных тестовых случаях не зафиксированы, фактические результаты совпали с ожидаемыми.

Разработанные тестовые случаи соответствуют ГОСТ Р 59792-2021 «Национальный стандарт Российской Федерации. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем» [16].

Глава 5. Аппаратные и программные средства для функционирования

Для обеспечения функционирования веб-ориентированной автоматизированной информационной системы «Инвентарис» используется комплекс аппаратных и программных средств, реализующих клиент-серверную архитектуру приложения.

Серверная часть системы отвечает за обработку HTTP-запросов, выполнение бизнес-логики, управление сессиями пользователей и взаимодействие с базой данных. Клиентская часть обеспечивает доступ к функционалу системы через веб-интерфейс.

К аппаратным средствам относятся:

— сервер или персональный компьютер, выполняющий функции размещения приложения;

— клиентские устройства пользователей (ПК, ноутбуки);

— сетевое оборудование, обеспечивающее доступ к системе.

Минимальные системные требования для серверной части:

— оперативная память: 4 ГБ и более;

— процессор: не ниже 2-ядерного (Intel Core i3 или аналог);

— свободное дисковое пространство: не менее 10 ГБ;

— стабильное сетевое подключение.

Минимальные системные требования для клиентской части:

— оперативная память: 2 ГБ и более;

— современный веб-браузер (Google Chrome, Mozilla Firefox, Microsoft Edge);

— доступ к сети Интернет или локальной сети.

Программные средства серверной части включают:

— операционную систему (предпочтительно Linux, допускается macOS или Windows);

— интерпретатор языка программирования Python версии 3.10 и выше;

— веб-фреймворк Flask для реализации серверной логики;

— WSGI-сервер (например, Gunicorn или uWSGI), обеспечивающий взаимодействие веб-приложения с веб-сервером;

— веб-сервер Apache HTTP Server с модулем mod_wsgi для развертывания WSGI-приложения;

— модуль mod_ssl для обеспечения защищённого соединения по протоколу HTTPS;

— библиотеку Flask-Session для управления пользовательскими сессиями;

— шаблонизатор Jinja2 для генерации HTML-страниц;

— библиотеку PyMySQL для взаимодействия с базой данных;

— систему управления базами данных MariaDB или MySQL;

— конфигурационные файлы приложения (config.json, sql_queries.json).

Программные средства клиентской части включают:

— веб-браузер с поддержкой HTML5, CSS3 и JavaScript;

Для разработки и сопровождения системы используются:

— среда разработки Visual Studio Code;

— система контроля версий Git;

— инструменты администрирования СУБД (например, phpMyAdmin).

Таким образом, для функционирования АИС «Инвентарис» используются современные программные средства, соответствующие архитектуре веб-приложений и обеспечивающие надёжную и безопасную работу системы.

Глава 6. Инструкции пользователя и администратора системы

В соответствии с требованиями технического задания (Приложение Г) в рамках разработки автоматизированной информационной системы «Инвентарис» было подготовлено руководство пользователя и администратора системы, включающее описание функциональных возможностей, порядка работы с интерфейсом, а также особенностей управления учетными записями и справочными данными. Руководство пользователя разработано в соответствии с ГОСТ Р 59795–2021 «Национальный стандарт Российской Федерации. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов» в части требований к содержанию документа. [15]

Руководство охватывает основные сценарии взаимодействия с системой, включая авторизацию, работу с карточками оборудования, управление справочниками, регистрацию перемещений и формирование отчетов. Также в документе подробно описана модель разграничения прав доступа ESR, определяющая допустимые действия пользователей в зависимости от их роли.

С технической точки зрения руководство отражает особенности реализации системы, построенной на языке программирования Python с использованием микрофреймворка Flask. Для обработки HTTP-запросов и маршрутизации используются встроенные механизмы Flask, а контроль доступа реализован через пользовательские декораторы и серверную проверку прав. Работа с базой данных осуществляется через библиотеку PyMySQL, обеспечивающую взаимодействие с СУБД MariaDB (или MySQL). Пользовательский интерфейс реализован с применением шаблонизатора Jinja2 и CSS-фреймворка Bootstrap.

Дополнительно в рамках проектной документации было разработано руководство по стилю интерфейса, определяющее единые требования к оформлению элементов пользовательского интерфейса, структуре страниц и визуальному представлению данных.

Ознакомиться с полным руководством пользователя и администратора системы можно в Приложении А. Руководство по стилю представлено в Приложении Б.

Заключение

В ходе выполнения дипломного проекта была проведена значительная работа, направленная на разработку веб-ориентированной автоматизированной информационной системы для учёта, инвентаризации и управления жизненным циклом оборудования предприятия. В рамках проекта был подготовлен полный комплекс проектной и эксплуатационной документации, включающий техническое задание, общую концепцию, модели проектирования, описание реализации, а также руководство пользователя.

На этапе проектирования была выполнена проработка структуры системы, определены основные сущности, их взаимосвязи, а также логика функционирования программного продукта. Были разработаны функциональные модели, отражающие ключевые бизнес-процессы, и сформирована архитектура приложения с учётом требований к расширяемости и удобству сопровождения.

В процессе разработки реализованы основные модули системы, обеспечивающие работу с оборудованием, справочниками, пользователями, перемещениями и отчётами. Особое внимание уделено механизмам версионности данных, позволяющим отслеживать изменения состояния объектов во времени. Проведено тестирование программного продукта с использованием разработанных тестовых сценариев, что позволило подтвердить корректность работы основных функций системы.

Разработанный программный продукт был представлен заказчику в полном объёме. Система готова к развертыванию и эксплуатации в рамках целевой инфраструктуры. Работы по непосредственной интеграции в существующую информационную среду организации, а также настройке окружения и переносу данных остаются в зоне ответственности заказчика.

Таким образом, поставленные цели дипломного проекта достигнуты, а разработанная система соответствует предъявляемым требованиям и может быть использована для автоматизации процессов учёта и контроля оборудования предприятия.

Список использованных источников

Нормативно-правовые акты:

1. Российская Федерация. Законы. О бухгалтерском учете: Федеральный закон от 06.12.2011 № 402-ФЗ: [ред. от 15.12.2025]. — Текст: электронный // КонсультантПлюс: справочная правовая система. — URL: https://www.consultant.ru/document/cons_doc_LAW_122855/ (дата обращения: 01.05.2026).

2. Об утверждении Федеральных стандартов бухгалтерского учета ФСБУ 6/2020 ‘Основные средства’ и ФСБУ 26/2020 ‘Капитальные вложения’: приказ Минфина России от 17.09.2020 № 204н: [ред. от 30.05.2022]. — Текст: электронный // КонсультантПлюс: справочная правовая система. — URL: https://www.consultant.ru/document/cons_doc_LAW_365338/ (дата обращения: 01.05.2026).

3. Об утверждении Федерального стандарта бухгалтерского учета ФСБУ 28/2023 ‘Инвентаризация’: приказ Минфина России от 13.01.2023 № 4н. — Текст: электронный // КонсультантПлюс: справочная правовая система.

Электронные издания:

4. GLPI // Система инвентаризации и учёта активов внутри компании URL: https://www.glpi-project.org/ (дата обращения: 01.05.2026).

5. Snipe-IT // Система управления складом URL: https://snipeitapp.com/product (дата обращения: 01.05.2026).

6. ServiceNow // Система управления IT продуктом URL: https://www.servicenow.com/products/it-asset-management.html (дата обращения: 01.05.2026).

7. 1С Решения // 1С:Инвентаризация и управление имуществом URL: https://solutions.1c.ru/catalog/asm/features (дата обращения: 01.05.2026).

8. Enterprise Asset Management (EAM): overview of asset lifecycle management // IBM: официальный сайт. — URL: https://www.ibm.com/think/topics/enterprise-asset-management (дата обращения: 01.05.2026). — Текст: электронный.

Нормативно-технические и нормативные документы:

9. ГОСТ Р 59793-2021. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. — Москва: Госстандарт СССР, 1990. — Текст: непосредственный. (ИФАП)

10. ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. — Москва: Стандартинформ, 2020. — Текст: непосредственный. (ИФАП)

11. ГОСТ Р 57100-2025/ISO/IEC/IEEE 42010:2011. Системная и программная инженерия. Описание архитектуры. — Москва: Стандартинформ, 2016. — Текст: непосредственный. (ИФАП)

12. Integration Definition for Function Modeling (IDEF0): FIPS PUB 183. — Gaithersburg: National Institute of Standards and Technology, 1993. — Текст: электронный // NIST: официальный сайт. — URL: NIST FIPS PUB 183 (дата обращения: 07.05.2026). (rsmu.ru)

13. Integration Definition for Information Modeling (IDEF1X): FIPS PUB 184. — Gaithersburg: National Institute of Standards and Technology, 1993. — Текст: электронный // GovInfo : официальный сайт. — URL: FIPS PUB 184 (дата обращения: 07.05.2026). (nntu.ru)

14. ISO/IEC 19505-2:2012. Information technology. Object Management Group Unified Modeling Language (OMG UML). Part 2: Superstructure. — Geneva: International Organization for Standardization, 2012. — Текст: электронный // OMG: официальный сайт. — URL: OMG UML Specification (дата обращения: 07.05.2026). (school.hse.ru)

15. ГОСТ Р 59795–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов. — Москва: Стандартинформ, 2021. — Текст: непосредственный.

16. ГОСТ Р 59792–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к тестированию. — Москва: Стандартинформ, 2021. — Текст: непосредственный.

Приложение А. Руководство пользователя

1. Введение

1.1 Назначение системы

«Инвентарис» представляет собой веб-ориентированную автоматизированную информационную систему для учета оборудования организации. Система поддерживает ведение карточек инвентарных единиц, хранение справочников, фиксацию перемещений, просмотр текущего состояния фонда оборудования и формирование отчетов по актуальным данным и истории изменений.

Основная практическая ценность системы состоит в том, что пользователь работает не с разрозненными таблицами и документами, а с единым реестром. Каждая единица оборудования имеет инвентарный номер, модель, категорию, статус, помещение, сведения о закупке, стоимости, гарантии и истории обновлений. При изменении значимых сведений система не затирает старую карточку, а создает новую версию, благодаря чему сохраняется прослеживаемость жизненного цикла оборудования.

Руководство предназначено для администраторов, операторов учета, сотрудников, отвечающих за перемещение и контроль оборудования, а также для пользователей, которым требуется просматривать отчеты. Документ описывает доступные разделы, правила работы с формами, ограничения прав доступа и порядок чтения сведений на основных страницах.

1.2 Техническая основа

Приложение реализовано на языке Python с использованием микрофреймворка Flask. Сеансы пользователей хранятся через механизм Flask-Session, HTML-интерфейс построен на шаблонах Jinja2 и компонентах Bootstrap, а данные размещаются в СУБД MariaDB или MySQL с кодировкой utf8mb4. Для доступа к базе данных используется библиотека PyMySQL, запросы вынесены в файл sql_queries.json, а начальная структура БД описана в db_init.sql.

Конфигурация приложения задается в config.json или, при его отсутствии, в config.example.json. В конфигурации определяются секретный ключ Flask, параметры подключения к MySQL/MariaDB, расположение SQL-файлов, каталог

журналов, режим отладки, параметры серверной сессии и возможность автоматической установки схемы базы данных. При стандартном запуске веб-сервер слушает адрес 127.0.0.1 и порт 5000.

Интерфейс системы является браузерным. После входа пользователь видит верхнюю навигационную панель, где расположены переходы к главной странице, оборудованию, перемещениям, отчетам, атрибутам и администрированию. Справа отображается имя текущего пользователя и его код прав ESR, что помогает быстро понять, с каким уровнем доступа выполняется работа.

1.3 Источники сведений для руководства

Настоящее руководство подготовлено как эксплуатационный документ по фактической реализации и проектной документации. При составлении использованы маршруты Flask, декораторы проверки прав, SQL-запросы, структура таблиц базы данных, шаблоны страниц, проектные диаграммы, Бриф, техника безопасности, техническое задание. Поэтому названия кнопок, состав полей, логика версий и ограничения доступа соответствуют текущему состоянию проекта.

Скриншоты включены в руководство как контрольные изображения интерфейса.

1.4 Общие принципы работы

Работа в системе начинается с авторизации. После успешного входа Flask сохраняет в сессии идентификатор пользователя, логин, ФИО, роль и права. При каждом обращении к защищенным страницам система повторно проверяет пользователя в базе данных. Если учетная запись отключена, не имеет роли или более не существует, сессия очищается и пользователь возвращается на страницу входа.

Большинство операций разделены на чтение и запись. Просмотр списков, карточек и отчетов доступен пользователям с правом чтения соответствующего контура, а создание и изменение данных требует права записи. При недостатке прав приложение возвращает страницу 403 с сообщением о невозможности выполнить действие.

1.5 Область применения

Веб-система «Инвентарис» применяется для автоматизации учета, инвентаризации и контроля жизненного цикла оборудования предприятия. Средство автоматизации предназначено для подразделений, которые ведут реестр оборудования, фиксируют изменения состояния инвентарных единиц, контролируют перемещения между помещениями и используют отчетные сведения по актуальным данным и истории изменений.

Система может использоваться сотрудниками, ответственными за ведение карточек оборудования, администраторами справочников и учетных записей, а также пользователями, которым требуется получать сведения из отчетов. Использование системы предполагает работу с данными, уже принятыми в организации как учетные: инвентарными номерами, сведениями о моделях, помещениях, статусах, закупках, поставщиках и исполнителях операций.

1.6 Назначение и условия применения

Система предназначена для автоматизации операций создания и сопровождения карточек оборудования, ведения справочников, фиксации перемещений, просмотра истории изменений, формирования отчетов и выгрузки данных в CSV-формате. Доступ к операциям определяется назначенной пользователю ролью и трехпозиционным кодом прав ESR.

Для применения системы требуется рабочее место пользователя с современным веб-браузером и сетевым доступом к серверу приложения. На стороне сервера должны быть развернуты приложение «Инвентарис», база данных MariaDB или MySQL, конфигурационные файлы приложения и каталог журналов. Конкретные параметры серверной среды определяются администратором при установке и сопровождении системы.

Перед началом эксплуатации должны быть созданы учетные записи пользователей, назначены роли, заполнены основные справочники и проверена доступность базы данных. Пользователь должен владеть базовыми навыками работы с веб-интерфейсами: переходом по разделам, заполнением форм, применением фильтров, чтением таблиц и сохранением файлов, формируемых браузером.

1.7 Подготовка к работе

Для начала работы пользователь получает у администратора адрес веб-системы, логин, пароль и сведения о назначенной роли. После открытия адреса системы в браузере пользователь должен перейти на страницу входа, ввести учетные данные и убедиться, что после авторизации отображается главная страница с верхней навигационной панелью.

Первичная проверка работоспособности выполняется по доступности основных элементов интерфейса: имени текущего пользователя, кода прав ESR, пунктов верхнего меню, сводных таблиц на главной странице и переходов в разрешенные разделы. Если после входа отдельные разделы недоступны, необходимо сверить назначенную роль с требуемыми правами, указанными в настоящем руководстве.

Администратору перед началом регулярной работы рекомендуется проверить подключение к базе данных, наличие начальных справочников, корректность ролей, возможность создания тестовой карточки оборудования и формирование основного отчета. После проверки тестовые данные могут быть удалены или оставлены только в том случае, если они используются как контрольный пример для обучения пользователей.

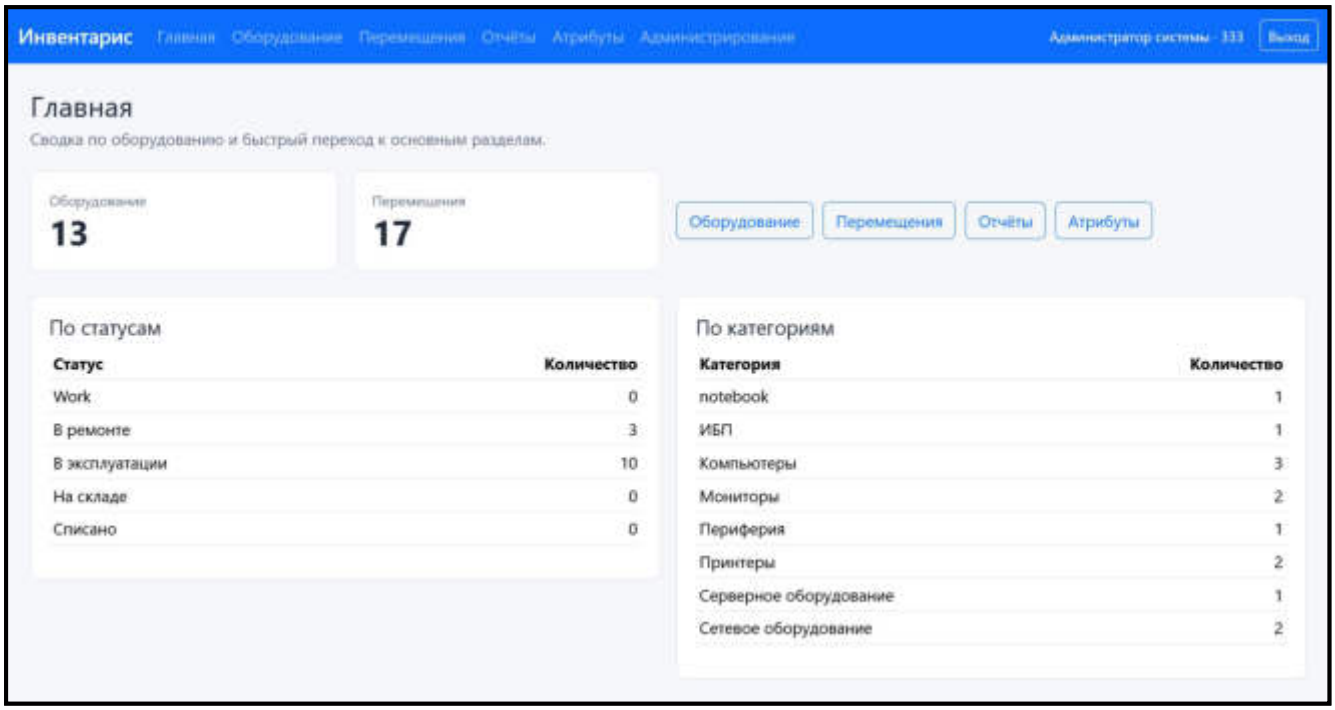


Рисунок 23. Главная страница системы «Инвентарис»

2 Администрирование

2.1 Модель прав ESR

Права пользователя задаются ролью. В роли хранится трехзначная строка ESR, где каждая позиция отвечает за отдельный контур системы: E означает оборудование и связанные операции, S означает системный контур, включая справочники и администрирование, R означает отчеты. Каждая цифра принимает значение от 1 до 3.

Таблица 32. Уровни прав ESR

Цифра	Значение	Практический смысл
1	Нет прав	Раздел или действие недоступны, при прямом переходе открывается страница отказа в доступе.
2	Чтение	Пользователь может просматривать сведения, списки, карточки и отчеты соответствующего контура.
3	Запись	Пользователь может создавать или изменять данные в соответствующем контуре, если такая операция предусмотрена интерфейсом.

Проверка прав выполняется посимвольно. Требование маршрута также задается трехзначной строкой. Например, требование 211 означает, что для перехода достаточно иметь чтение по оборудованию и не требуется доступ к системному контуру и отчетам. Если хотя бы одна цифра прав пользователя меньше требуемой цифры, действие блокируется.

На практике строка 333 означает полные права администратора. Строка 311 дает право создавать и редактировать оборудование и перемещения, но не открывает системные справочники и отчеты. Строка 222 дает пользователю доступ на чтение к основным разделам без возможности вносить изменения.

2.2 Доступность разделов по правам

Таблица 33. Требуемые права для разделов системы

Раздел или действие	Требуемые права	Что доступно пользователю
Главная страница	Авторизованный пользователь	Сводка по оборудованию, быстрые переходы, таблицы по статусам и категориям.
Список оборудования и карточка	211	Просмотр актуального оборудования, фильтры, открытие карточки и архивных версий по ссылкам.
Создание, редактирование, смена статуса оборудования	311	Добавление карточки, создание новой версии существующей карточки, изменение статуса.
Список перемещений	211	Просмотр журнала перемещений и переход к карточке оборудования.

Раздел или действие	Требуемые права	Что доступно пользователю
Создание перемещения	311	Фиксация операции перемещения, передачи, ремонта или списания с созданием новой версии карточки.
Справочники атрибутов	121	Просмотр категорий, статусов, помещений, моделей и поставщиков.
Создание и редактирование справочников	131	Добавление и изменение записей справочников.
Список пользователей и ролей	131	Администрирование учетных записей и ролей.
Основные отчеты, отчеты по перемещениям и изменениям	112	Просмотр отчетов и фильтрация данных.
Выгрузка CSV отчета по оборудованию	113	Формирование CSV-файла по текущим фильтрам отчета.

Пункты верхнего меню отображаются всем авторизованным пользователям, однако фактический доступ определяется серверной проверкой. Поэтому пользователь может видеть ссылку на раздел, но при недостатке прав получить отказ. Это поведение защищает данные даже при попытке открыть адрес напрямую.

2.3 Список пользователей

Раздел «Пользователи» предназначен для просмотра учетных записей и управления их активностью. В таблице отображаются идентификатор, ФИО, логин, email, телефон, табельный номер, назначенная роль, код прав и признак активности. Для активных учетных записей доступно действие деактивации, а для каждой записи — переход к редактированию.

Деактивация не удаляет пользователя из базы данных. Учетная запись остается в истории как связанный исполнитель операций, но пользователь больше не сможет войти в систему. Это важно для сохранения корректных ссылок на авторов карточек, перемещений и изменений.

Система дополнительно защищает последнего активного администратора. Если пользователь с правами 333 является единственным активным администратором, его нельзя отключить или перевести на роль без полного доступа. Такое ограничение снижает риск случайной потери административного управления.

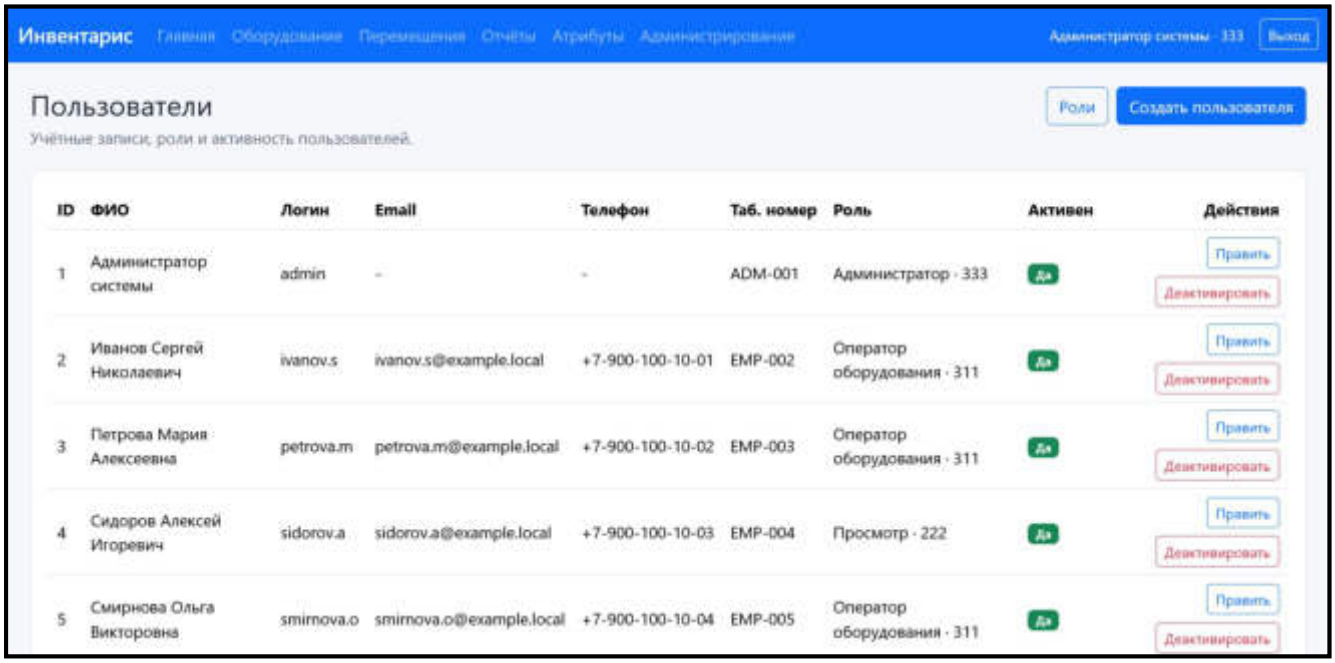


Рисунок 24. Список пользователей

2.4 Создание и редактирование пользователя

Форма пользователя содержит ФИО, логин, пароль, роль, email, телефон, табельный номер, дату приема и флаг активности. При создании пароль обязателен. При редактировании пароль можно оставить пустым, тогда он не изменяется. Роль выбирается из списка существующих ролей, рядом с названием отображается код ESR.

Перед сохранением приложение проверяет обязательные поля, корректность даты приема, существование выбранной роли и уникальность логина. Если значение не проходит проверку, пользователь остается на форме и видит предупреждение. После успешного сохранения система возвращает администратора к списку пользователей.

При создании учетной записи рекомендуется сразу назначать минимально достаточную роль. Операторам, которые только вносят оборудование и перемещения, обычно достаточно роли с правами 311. Пользователям, которым нужны отчеты без изменения данных, назначается роль с чтением нужных контуров.

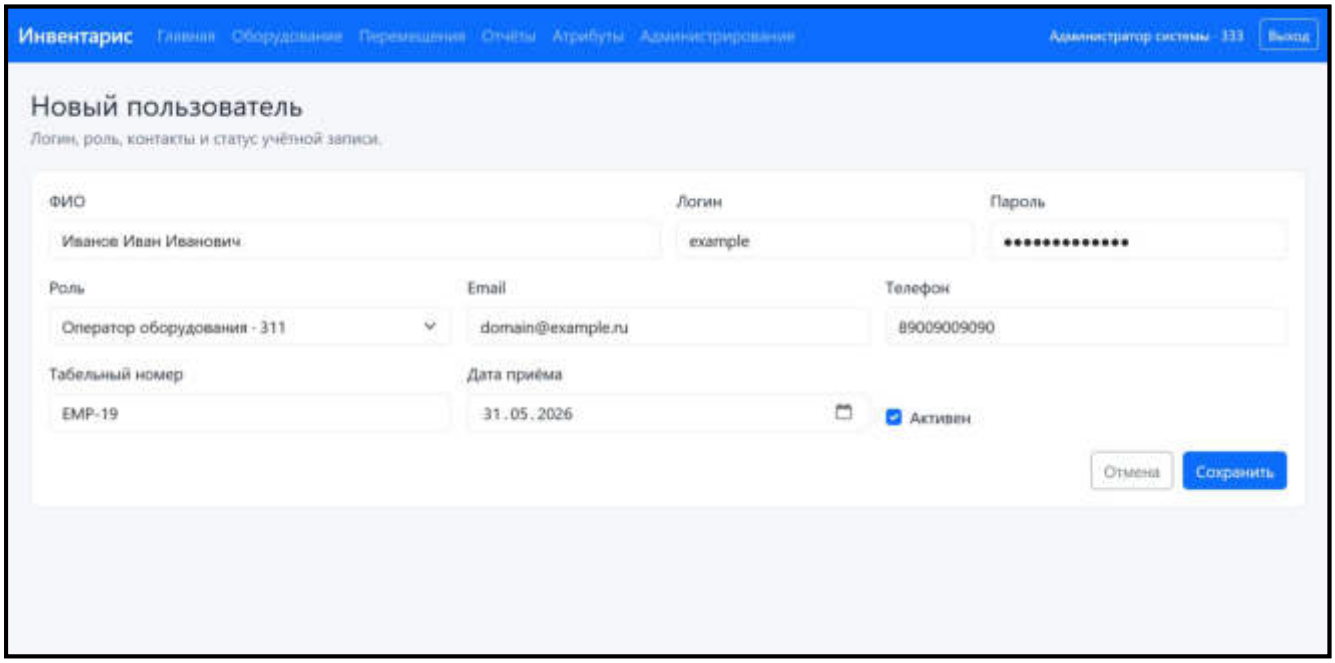


Рисунок 25. Форма создания пользователя

2.5 Список ролей

Раздел «Роли» показывает набор доступных ролей системы. Для каждой роли указаны идентификатор, название, права ESR и описание. Через этот экран администратор может перейти к созданию новой роли или к редактированию существующей.

Роли являются центральным механизмом разграничения доступа. Изменение прав роли влияет на всех активных пользователей, которым она назначена. Поэтому перед корректировкой роли следует понимать, какие сотрудники используют ее в ежедневной работе и какие разделы им необходимы.

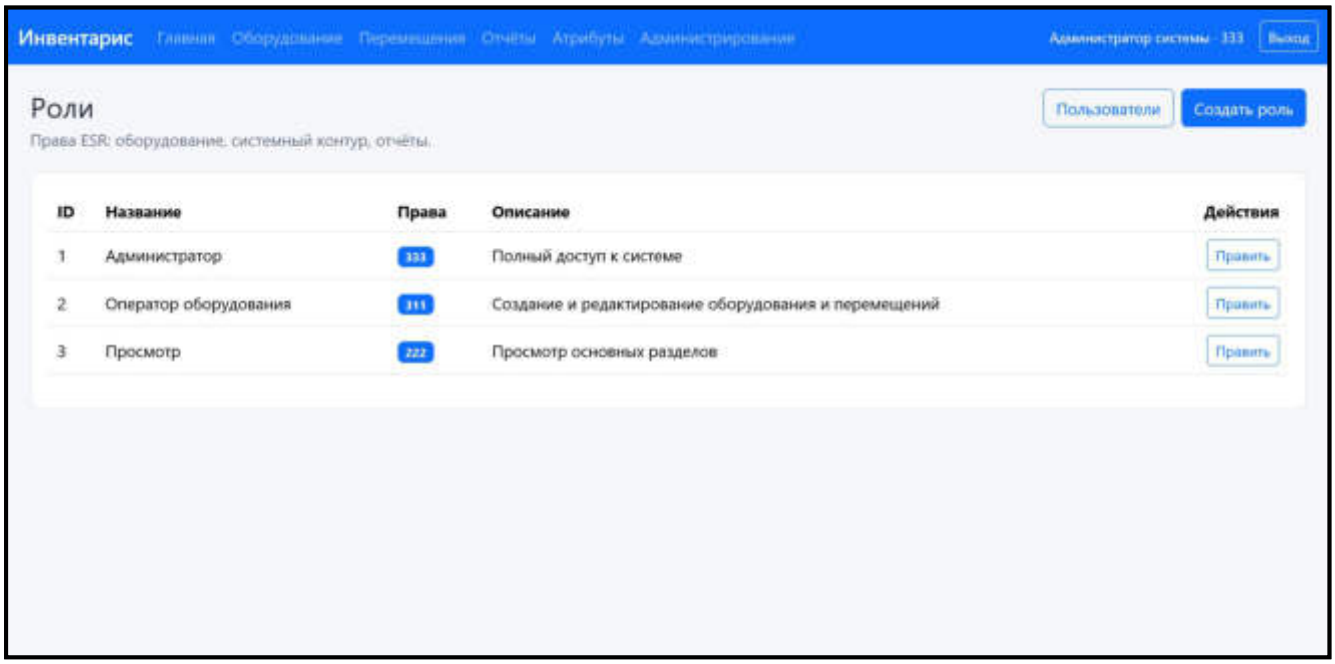


Рисунок 26. Список ролей

2.6 Создание и редактирование роли

Форма роли содержит название, строку прав ESR и описание. Строка прав должна состоять ровно из трех цифр, каждая из которых равна 1, 2 или 3. При вводе другого значения система не сохранит роль и покажет предупреждение.

Описание роли следует заполнять как управленческое пояснение: для кого предназначена роль и какие операции она разрешает. Например, описание «Создание и редактирование оборудования и перемещений» помогает администратору позже отличить оператора учета от пользователя только для просмотра.

При редактировании роли с полными правами действует та же защита последнего администратора. Если изменение прав приведет к тому, что в системе не останется активного пользователя с доступом 333, сохранение будет заблокировано.

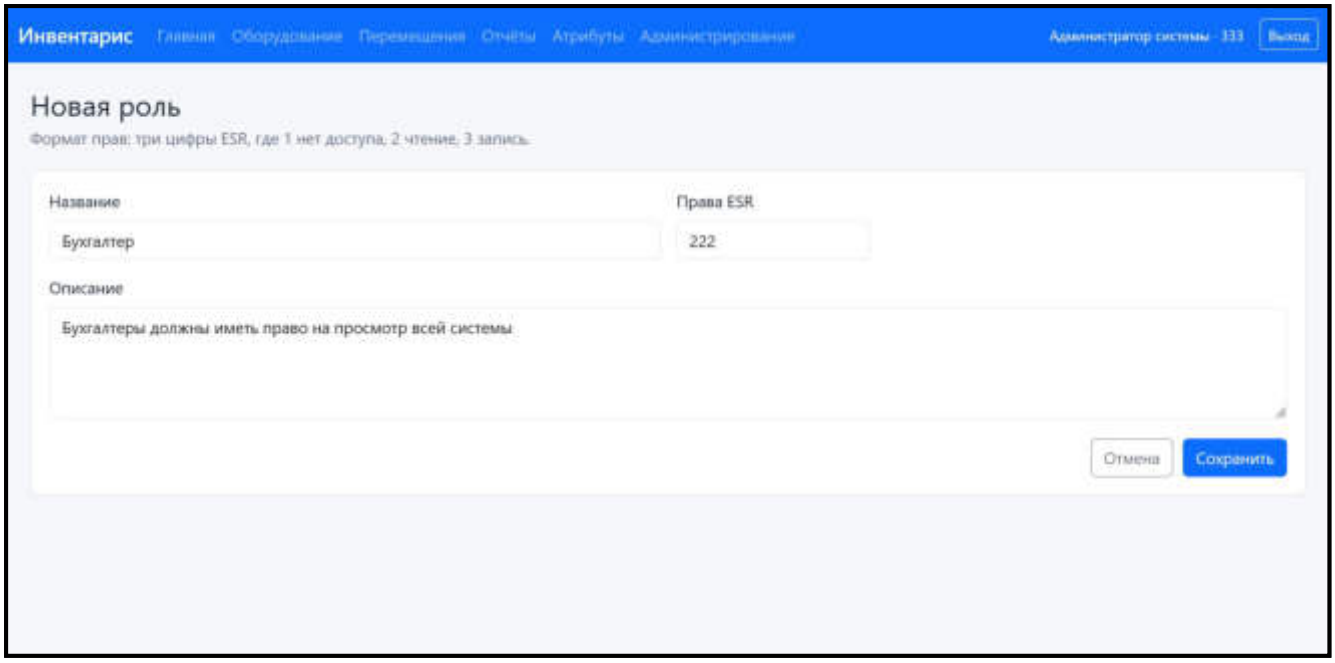


Рисунок 27. Форма создания роли

3 Использование системы

3.1 Верхняя навигация и главная страница

После входа пользователь попадает на главную страницу. Верхняя панель служит основным способом перемещения между разделами. Название системы ведет на главную страницу, а пункты «Оборудование», «Перемещения», «Отчеты», «Атрибуты» и «Администрирование» открывают соответствующие рабочие области.

Главная страница выполняет роль краткой сводки. На ней отображается общее количество актуального оборудования, количество перемещений, быстрые кнопки перехода и две таблицы распределения: по статусам и по категориям. Эти данные полезны для первичного контроля состояния инвентарного фонда без перехода в детальные отчеты.

Если часть данных не может быть загружена из-за ошибки подключения к базе данных, пользователь получает предупреждение. При штатной работе таблицы на главной странице строятся по актуальным версиям карточек оборудования, то есть архивные версии не увеличивают показатели.

3.2 Список оборудования

Раздел «Оборудование» является основным рабочим экраном оператора. В таблице отображаются только актуальные версии карточек, то есть последняя версия каждой инвентарной единицы. Таблица содержит ID, инвентарный и серийный номера, производителя, модель, категорию, помещение, статус, цену, дату покупки и доступные действия.

Над таблицей расположена форма фильтрации. Пользователь может искать по инвентарному номеру, модели, выбирать категорию и статус, а также вводить часть названия помещения. Кнопка «Найти» применяет условия и обновляет список. При большом количестве записей используется постраничная навигация «Назад» и «Дальше».

Инвентарный номер является ссылкой на карточку оборудования. Кнопка «Открыть» выполняет тот же переход. Если пользователь имеет право записи по контуру оборудования, рядом появляется кнопка «Править» и в заголовке страницы доступна кнопка «Добавить».

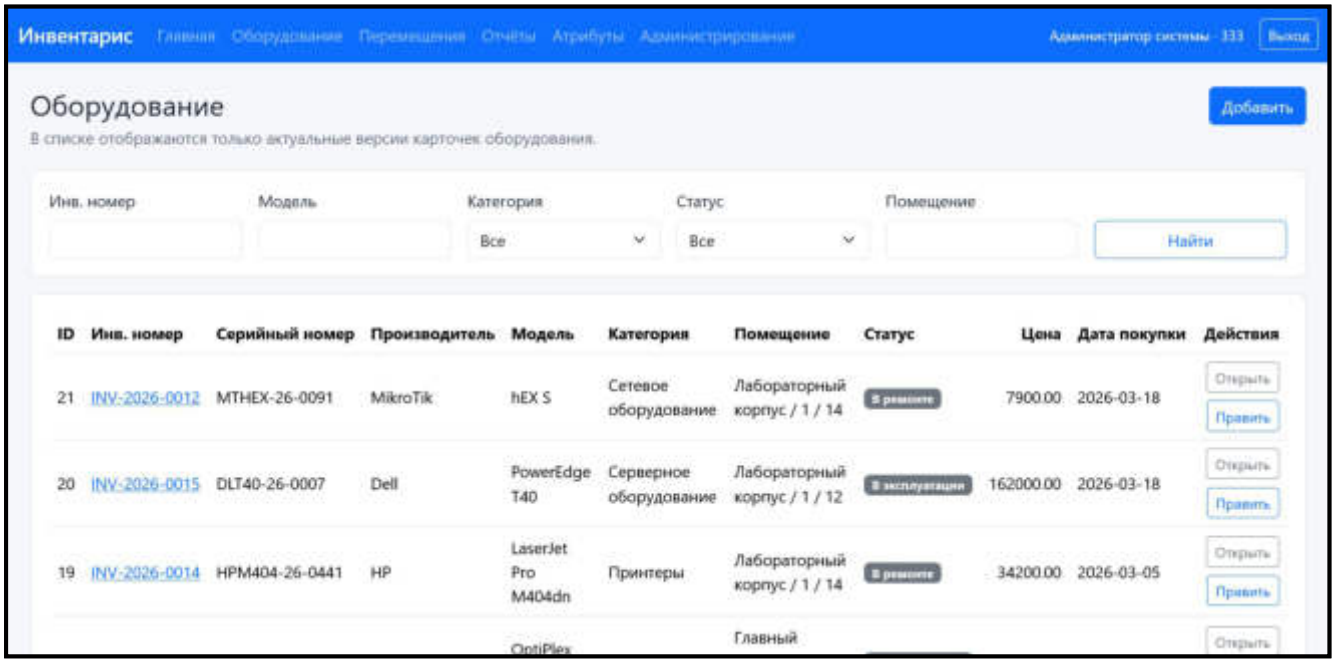


Рисунок 28. Список оборудования с фильтрами

3.3 Создание оборудования

Форма создания карточки открывается кнопкой «Добавить» в списке оборудования. Обязательными являются инвентарный номер, модель и статус. Дополнительно можно указать серийный номер, цену, помещение, закупку, дату покупки и дату окончания гарантии.

Модель выбирается из справочника моделей оборудования. В списке моделей отображаются производитель, название модели и категория, поэтому перед массовым вводом оборудования желательно заполнить справочники категорий и моделей. Помещение и закупка также выбираются из справочников, при отсутствии значения поле можно оставить пустым.

При сохранении система проверяет уникальность актуального инвентарного номера. Если уже существует актуальная карточка с таким номером, новая запись не создается. Это правило предотвращает появление двух текущих единиц с одинаковым инвентарным идентификатором.

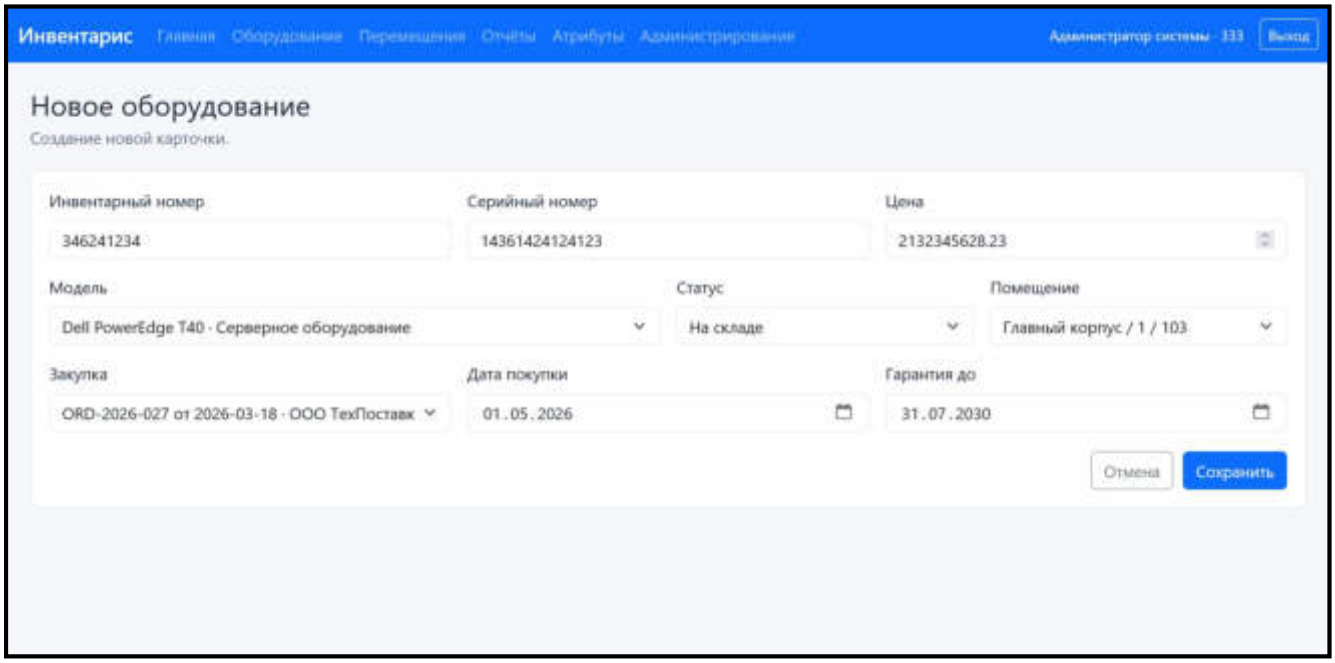


Рисунок 29. Форма создания оборудования

3.4 Карточка оборудования и версии

Карточка оборудования показывает детальные сведения об инвентарной единице. В ней собраны серийный номер, статус, помещение, даты покупки и гарантии, цена, закупка, поставщик, дата создания, автор создания, автор обновления версии, ссылка на предыдущую версию и характеристики модели в формате JSON.

Если карточка является архивной, система показывает предупреждение и запрещает редактирование. Архивной становится та версия, у которой уже появилась более новая версия. Если текущая карточка была создана как обновление, в верхней части отображается информационное сообщение с идентификатором предыдущей версии.

Для актуальной карточки и пользователя с правом записи доступны действия «Редактировать», «Сменить статус» и «Переместить». Редактирование и смена статуса не изменяют запись на месте, а создают новую версию. Это означает, что история значимых изменений остается доступной для отчетов и диаграммы версий.

3.5 Редактирование оборудования

Форма редактирования визуально близка к форме создания, но в заголовке прямо указано, что при сохранении будет создана новая версия оборудования. Старое состояние останется в базе как архивная запись, а новая запись станет актуальной.

Пользователь может изменить инвентарный номер, серийный номер, цену, модель, статус, помещение, закупку и даты. Система повторно проверяет обязательные поля, формат дат, числовое значение цены, существование выбранных справочных записей и уникальность инвентарного номера среди других актуальных карточек.

Такой способ редактирования особенно важен для аудита. Например, изменение помещения, статуса или стоимости можно позднее увидеть в отчете «Изменения за период», а полную цепочку изменений — на странице схемы изменений.

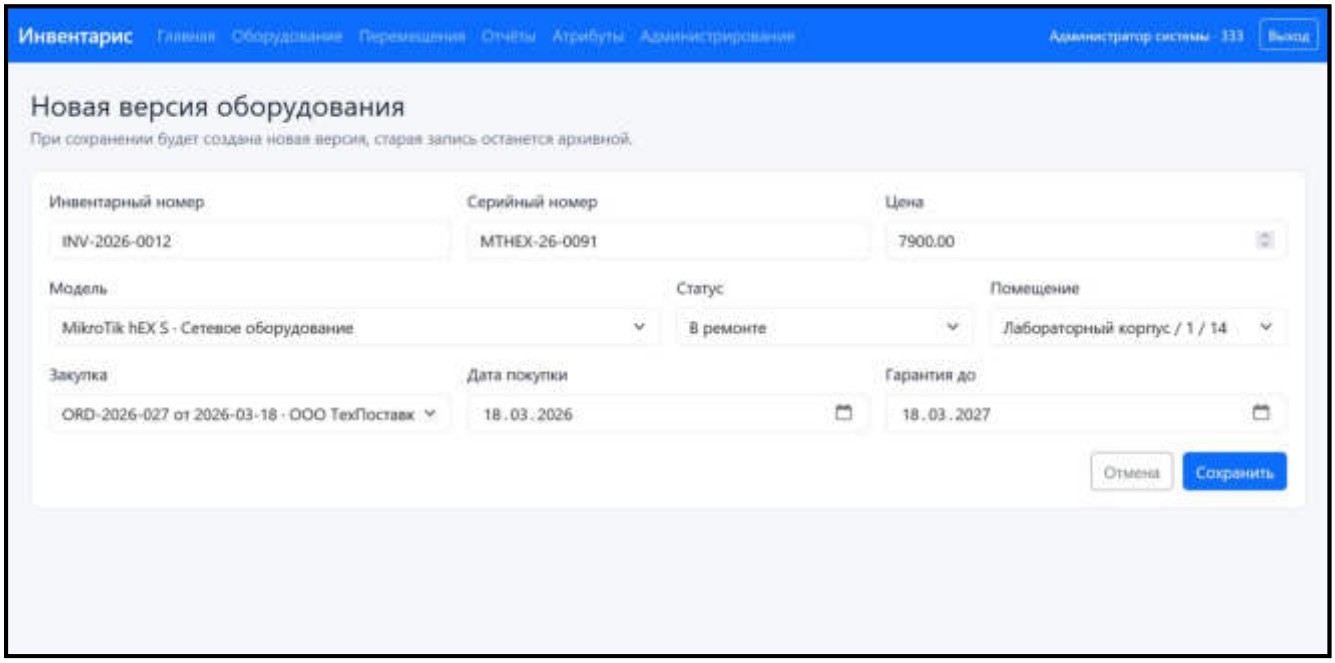


Рисунок 30. Форма редактирования оборудования

3.6 Смена статуса

Смена статуса является укороченным сценарием редактирования, когда требуется изменить только состояние оборудования. Пользователь открывает карточку актуальной записи, нажимает «Сменить статус», выбирает новый статус из справочника и сохраняет форму. Система создает новую версию карточки с теми же данными, но с обновленным статусом.

Статусы берутся из справочника «Статусы». Типовой набор начальных данных включает «В эксплуатации», «На складе», «В ремонте» и «Списано». Организация может изменить или расширить этот набор через справочники при наличии системных прав записи.

3.7 Справочники атрибутов

Раздел «Атрибуты» объединяет справочники, которые используются в карточках оборудования и отчетах. На странице представлены вкладки «Категории», «Статусы», «Помещения», «Модели оборудования» и «Поставщики». Пользователь с правом чтения системного контура может просматривать записи, а пользователь с правом записи видит кнопки добавления и редактирования.

Категории и статусы содержат название и описание. Помещения описываются зданием, этажом, номером или названием помещения и описанием. Модели оборудования содержат производителя, модель, категорию, характеристики JSON и срок гарантии в месяцах. Поставщики содержат название, контактное лицо, адрес, телефон и email.

Справочники следует вести аккуратно, потому что они влияют на удобство фильтрации и качество отчетов. Например, если одна и та же категория заведена в нескольких вариантах написания, отчет по категориям будет раздроблен. Перед созданием новой справочной записи рекомендуется проверить, нет ли уже подходящего значения.

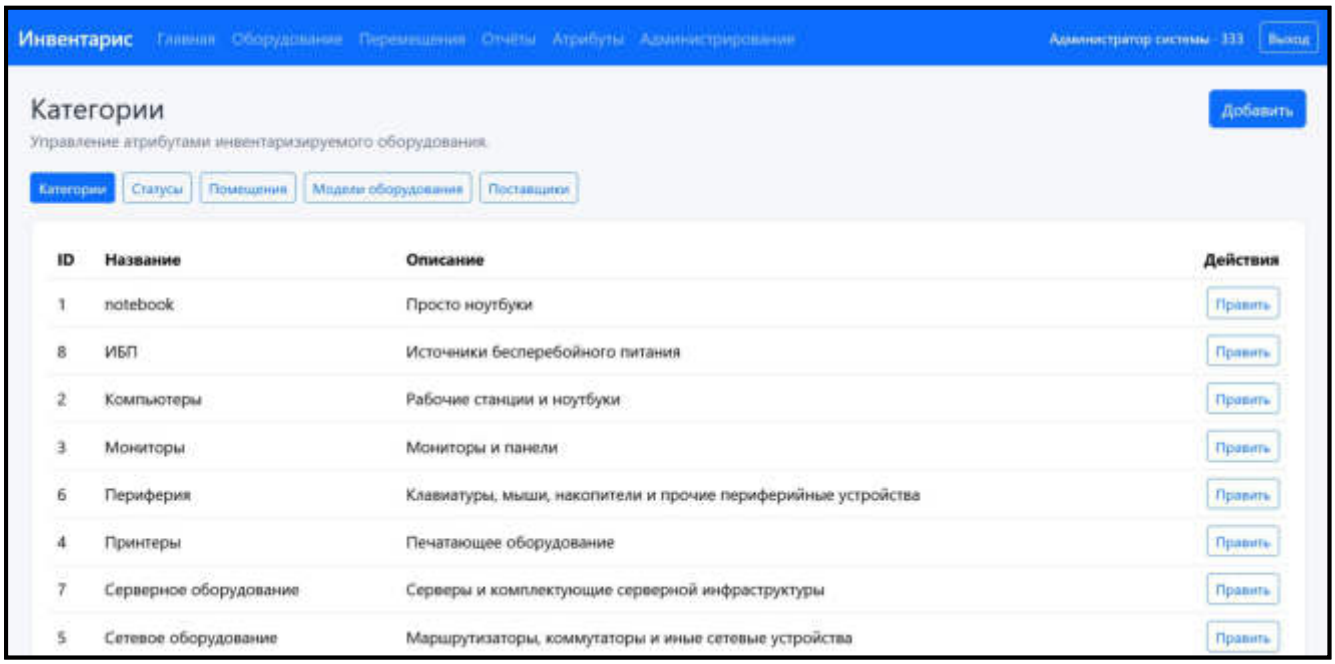


Рисунок 31. Справочник атрибутов.

3.8 Создание и редактирование справочной записи

Форма справочника строится динамически в зависимости от выбранного типа справочника. Для обязательных полей включена проверка заполнения.

Числовые поля, такие как гарантия в месяцах, должны содержать число. Поле характеристик модели должно быть корректным JSON, иначе запись не будет сохранена.

При создании модели оборудования сначала выбирается категория. Поэтому категории следует заполнять до моделей. После сохранения модель становится доступной в форме создания оборудования и в отчетных фильтрах через связанную категорию.

3.9 Список перемещений

Раздел «Перемещения» хранит журнал операций с оборудованием. В таблице отображаются дата операции, оборудование, тип, исходное помещение, новое помещение, исполнитель и примечание. Если связанная запись оборудования уже стала архивной, рядом с ней показывается отметка «архив».

Журнал перемещений не заменяет версионность карточек, а дополняет ее. При создании перемещения система создает новую актуальную версию карточки с новым помещением, а затем записывает операцию в журнал. Таким образом, текущий список оборудования показывает новое помещение, а журнал сохраняет факт операции и ее исполнителя.

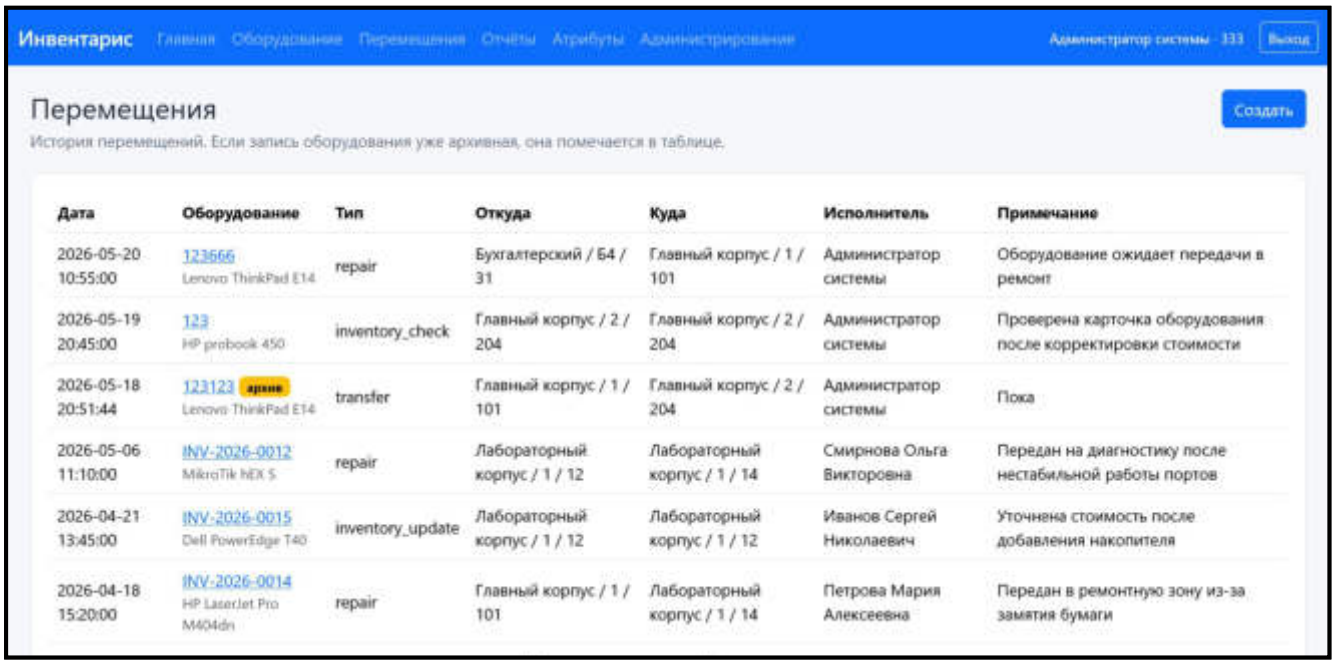


Рисунок 32. Журнал перемещений

3.10 Создание перемещения

Форма нового перемещения содержит выбор оборудования, тип операции, новое помещение и примечание. Доступны типы «Перемещение», «Передача», «Ремонт» и «Списание». Оборудование выбирается из списка актуальных карточек, а помещение — из справочника помещений.

Система не позволит создать операцию для архивной версии оборудования, для несуществующего помещения или для случая, когда новое помещение совпадает с текущим. После успешного сохранения пользователь возвращается к журналу перемещений и видит новую запись.

Поле примечания рекомендуется использовать для контекста операции: номер заявки, основание передачи, комментарий по ремонту или иное пояснение. Это повышает полезность журнала при последующем разборе движения оборудования.

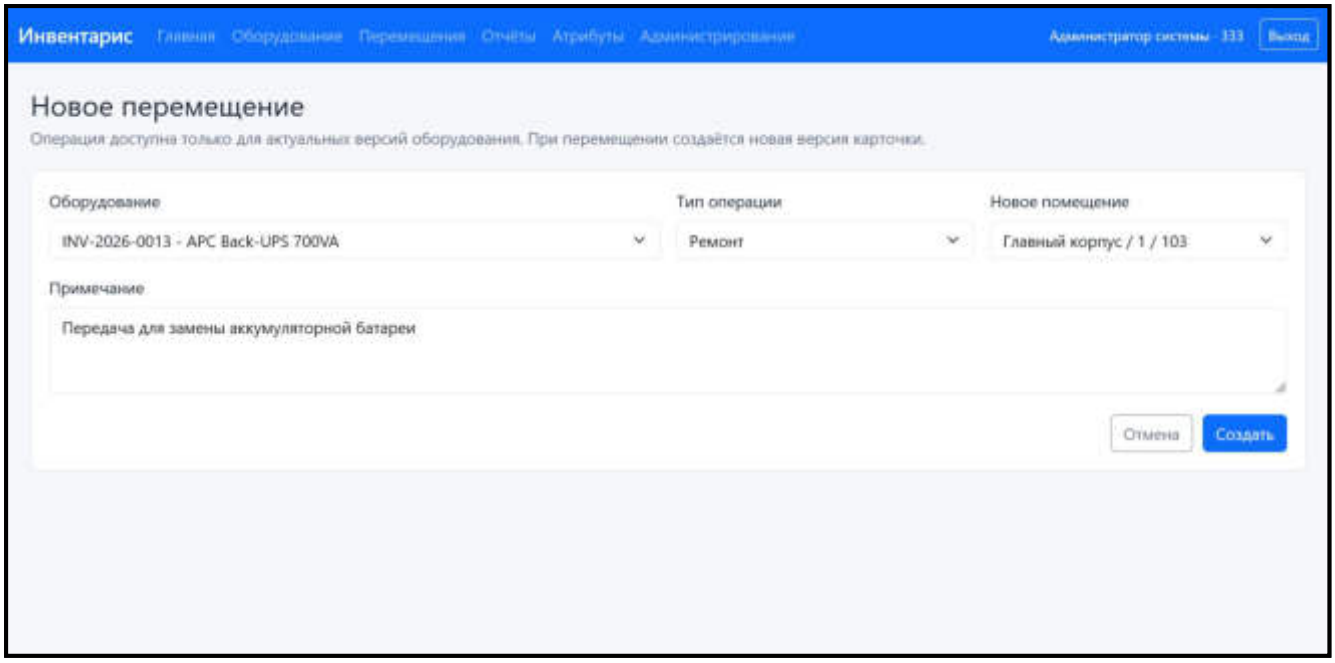


Рисунок 33. Форма создания перемещения

3.11 Основной отчет по оборудованию

Раздел «Отчеты» показывает актуальное оборудование и агрегированные сводки. В верхней части страницы расположены фильтры по категории, статусу и помещению. После нажатия «Сформировать» система перестраивает таблицу актуального оборудования и сводные блоки.

Основная таблица содержит ID, инвентарный номер, серийный номер, производителя, модель, категорию, статус, помещение и цену. Ниже расположены компактные сводки по статусам, категориям и помещениям. Они помогают быстро оценить распределение фонда оборудования.

Если пользователь имеет право записи в контуре отчетов, то есть код R равен 3, на странице появляется кнопка CSV. Выгрузка формирует файл equipment_report.csv с разделителем «точка с запятой» и применяет те же фильтры, которые выбраны на странице отчета.

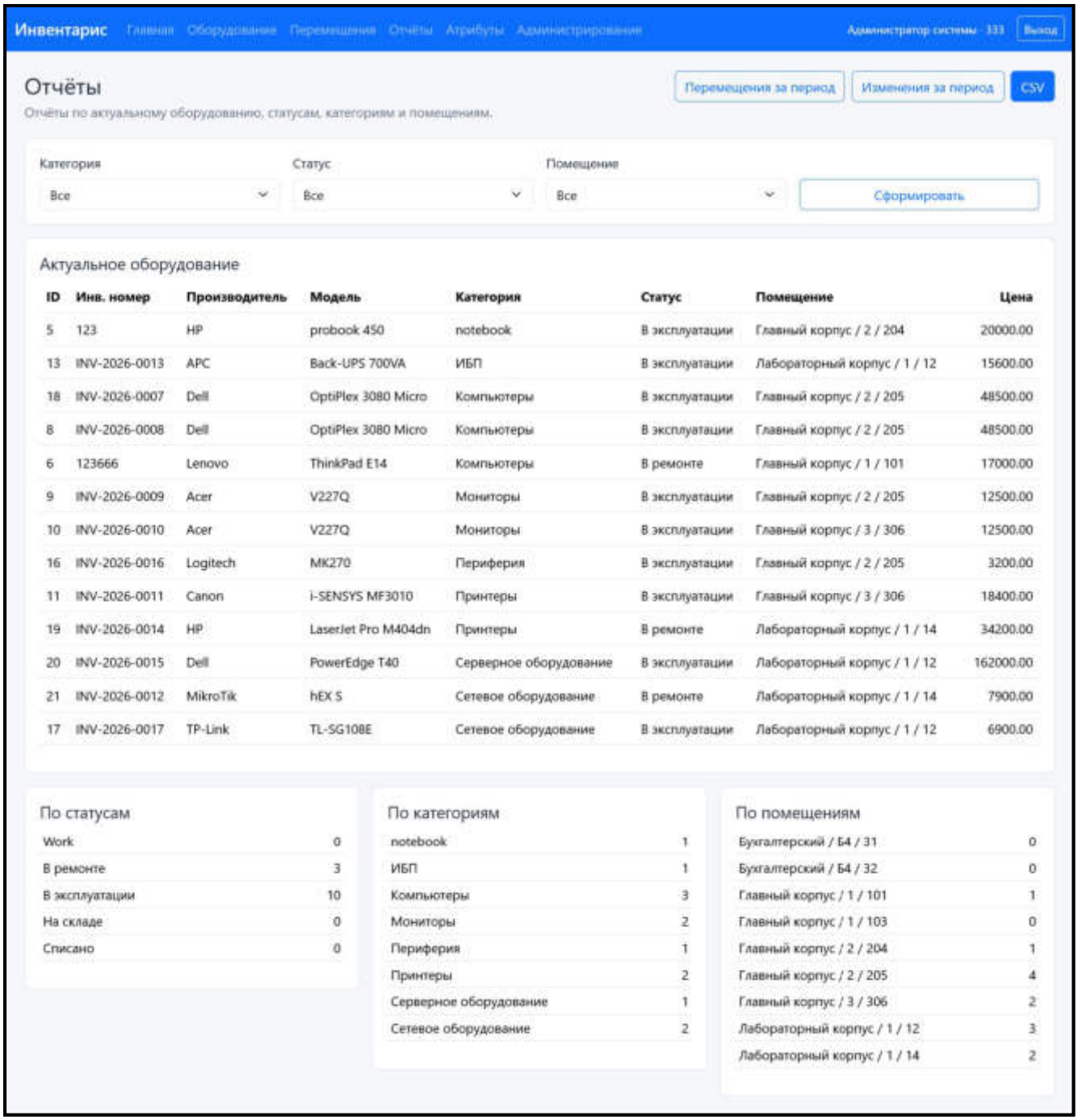


Рисунок 34. Основной отчет по оборудованию

3.12 Отчет по перемещениям за период

Отчет «Перемещения за период» предназначен для анализа журнала операций в заданном интервале дат. Пользователь задает дату начала и дату окончания в формате ГГГГ-ММ-ДД и нажимает «Сформировать». Если дата введена некорректно, система показывает предупреждение и формирует отчет по тем условиям, которые удалось распознать.

Таблица отчета содержит дату операции, оборудование, тип, исходное помещение, новое помещение, исполнителя и примечание. Данные сортируются от новых к старым. Отчет удобен для контроля работ за месяц, проверки перемещений между кабинетами или подготовки сведений по списанию и ремонту.

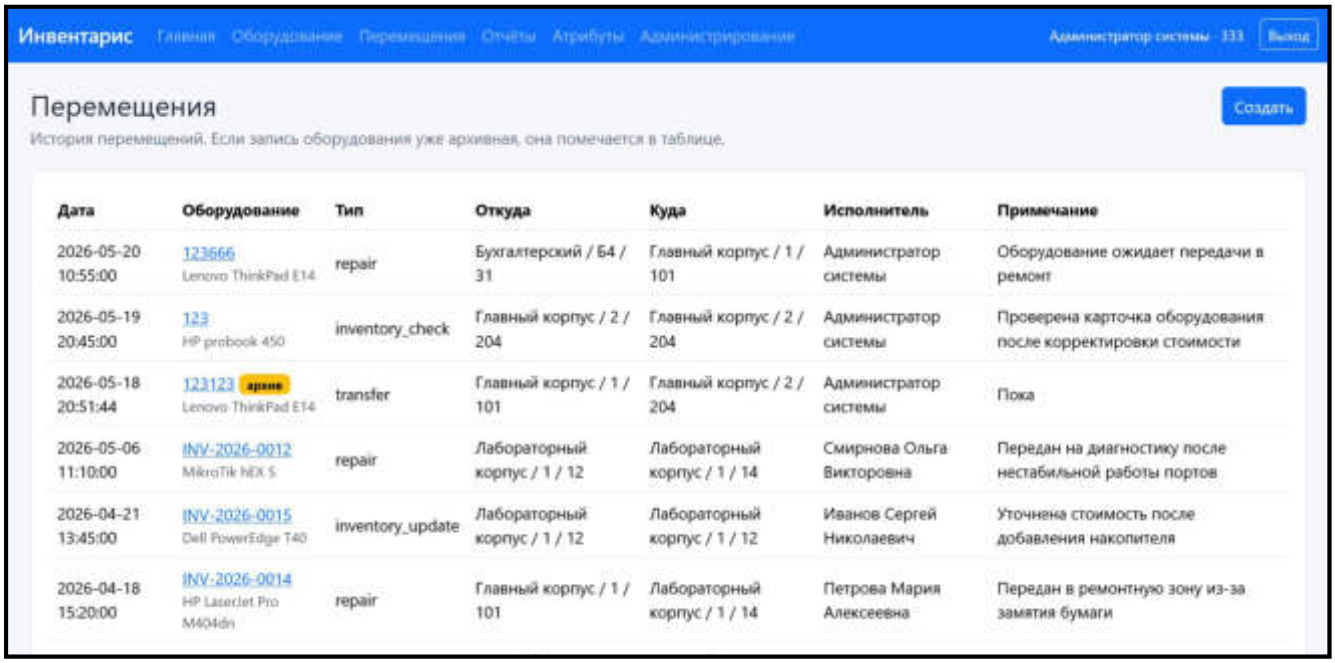


Рисунок 35. Отчет по перемещениям за период

3.13 Отчет по изменениям за период

Отчет «Изменения за период» показывает новые версии карточек оборудования, созданные в выбранном интервале дат. Для каждой версии система сравнивает ее с предыдущей и выводит список измененных полей: инвентарный номер, серийный номер, модель, категория, статус, локация, даты, гарантия и цена.

Если версия была создана технически и отображаемые поля не изменились, система показывает соответствующее сообщение. Для каждой строки доступна кнопка «Открыть схему», которая ведет к диаграмме цепочки версий выбранного оборудования.

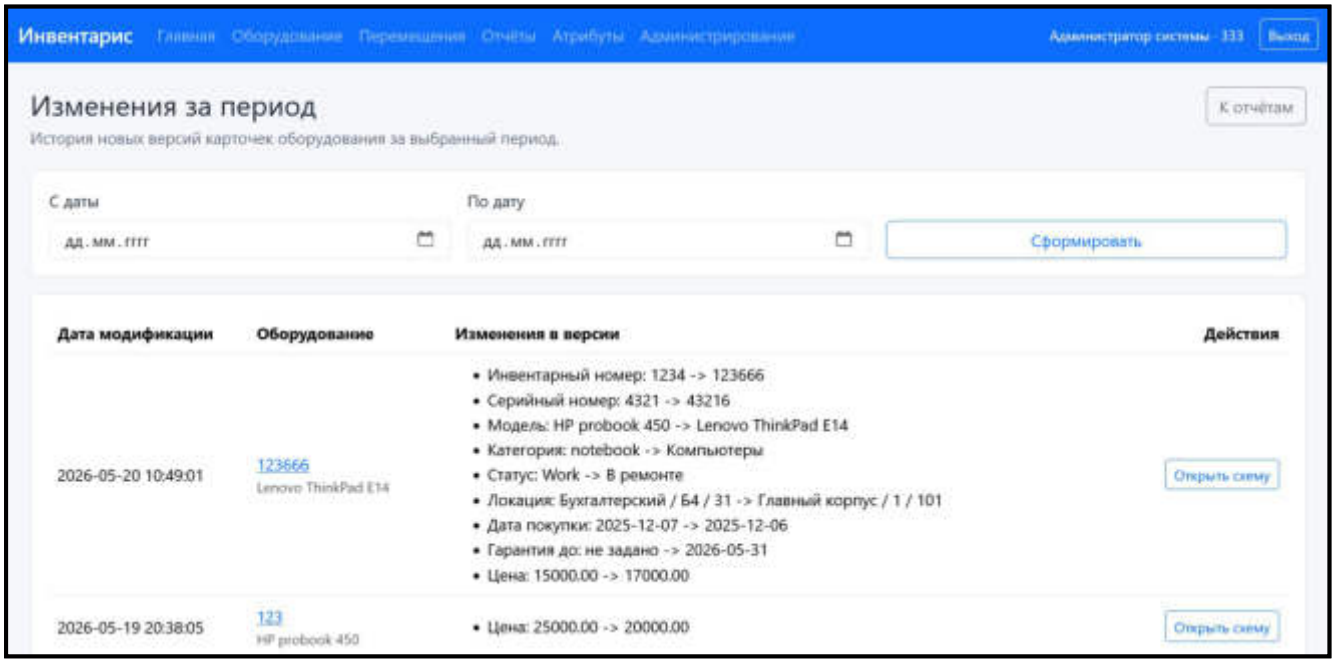


Рисунок 36. Отчет по изменениям за период

3.14 Схема изменений

Страница схемы изменений строит цепочку версий конкретной карточки оборудования. Верхний блок содержит интерактивный предпросмотр Mermaid-диаграммы. Колесо мыши изменяет масштаб, перетаскивание позволяет перемещать схему внутри области просмотра, двойной щелчок возвращает диаграмму к вписанному масштабу.

Ниже расположена таблица «Цепочка версий», где для каждой версии указаны порядковый номер, ID, дата, модель, статус, локация и автор версии. Еще ниже находится Mermaid-код диаграммы. Его можно скопировать или скачать в файл .mmd через кнопку в заголовке страницы.

Схема изменений особенно полезна, когда необходимо объяснить, как оборудование перешло из одного состояния в другое. Ребра диаграммы подписываются измененными полями, поэтому пользователь видит не только последовательность версий, но и смысл каждого перехода.

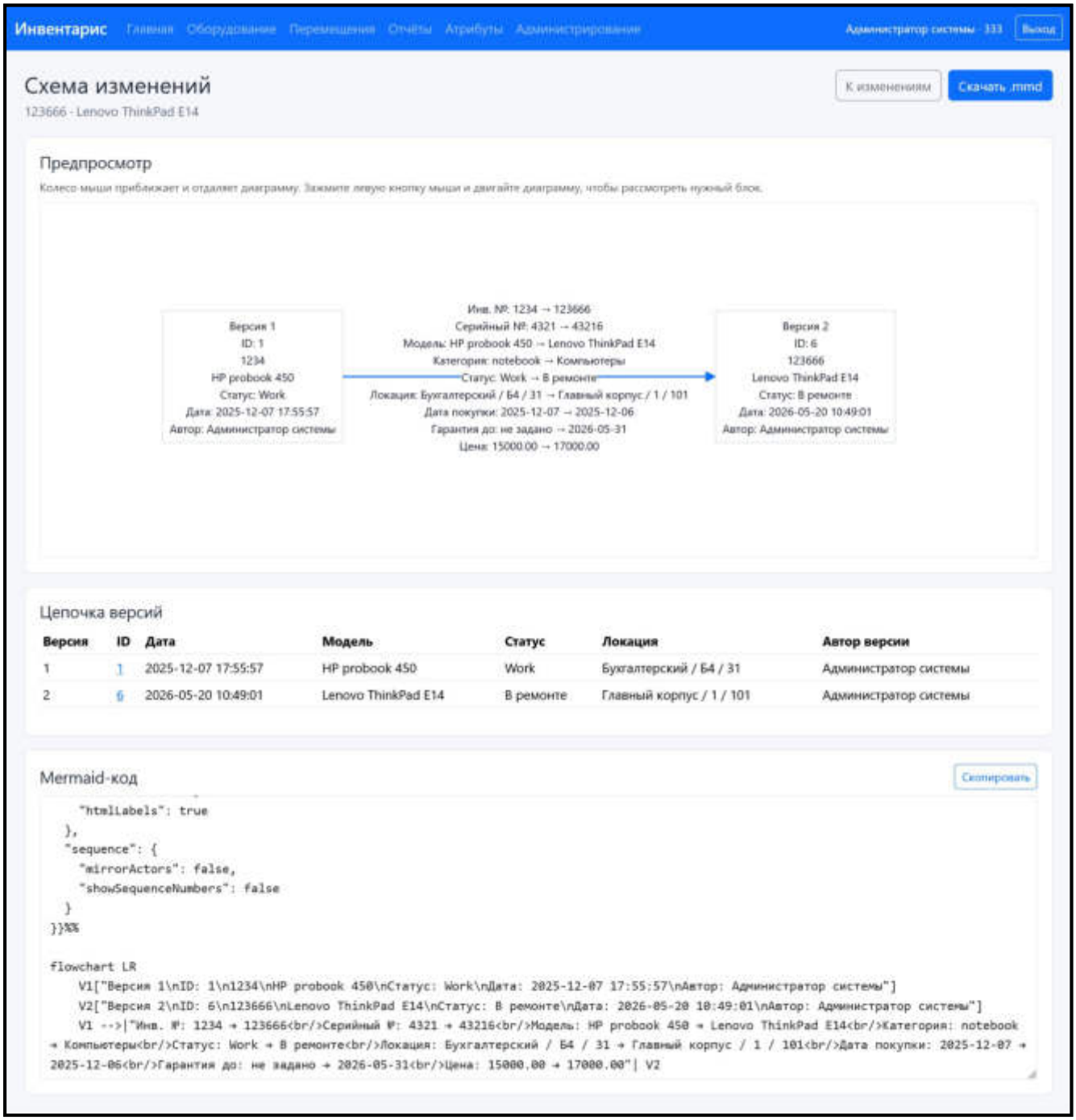


Рисунок 37. Схема изменений оборудования

3.15 Экспорт CSV

CSV-выгрузка доступна из основного отчета при наличии права 113 или выше по отчетному контуру. Файл содержит идентификатор, инвентарный номер, серийный номер, производителя, модель, категорию, помещение, статус и цену. Кодировка ответа устанавливается как UTF-8, а имя файла по умолчанию — equipment_report.csv.

Перед выгрузкой следует установить нужные фильтры на странице отчета. Система использует те же значения категории, статуса и помещения, которые переданы в адресе страницы. Это позволяет получить выгрузку не по всей базе, а по конкретному срезу.

3.16 Работа с ошибками и сообщениями

Система использует всплывающие сообщения Bootstrap для подтверждений, предупреждений и ошибок. Успешные операции сопровождаются сообщениями вроде «Пользователь создан», «Оборудование добавлено» или «Перемещение создано». Ошибки валидации показываются как предупреждения и обычно требуют исправить поля формы.

Если пользователь пытается выполнить действие без прав, открывается страница 403. Если страница не найдена, приложение выводит сообщение «Страница не найдена». При непредвиденной ошибке приложение записывает подробности в журнал и показывает пользователю нейтральное сообщение о необходимости повторить действие позже.

3.17 Аварийные ситуации

К аварийным ситуациям при эксплуатации системы относятся длительная недоступность сервера приложения, отказ соединения с базой данных, невозможность авторизации при корректных учетных данных, появление непредвиденных ошибок при сохранении данных, обнаружение некорректных сведений в карточках оборудования или подозрение на несанкционированный доступ.

При длительной недоступности системы пользователь должен прекратить повторные попытки сохранения данных, зафиксировать выполняемое действие и передать сведения администратору. До восстановления работоспособности не следует повторно отправлять одну и ту же форму несколько раз, поскольку после восстановления связи это может привести к созданию лишних записей или дублированию пользовательских действий.

При обнаружении ошибочных данных пользователь должен проверить, является ли открытая карточка актуальной версией. Если ошибка относится к текущим сведениям, исправление выполняется через предусмотренные операции

редактирования, смены статуса или перемещения. Если ошибка связана с исторической версией, журналом перемещений или отчетными данными, необходимо передать информацию администратору для проверки базы данных и журналов приложения.

При подозрении на несанкционированный доступ пользователь должен завершить работу с системой, не передавать свои учетные данные другим лицам и сообщить администратору о признаках инцидента. Администратор проверяет активность учетной записи, назначенные роли, журналы приложения и при необходимости деактивирует учетную запись до выяснения обстоятельств.

4 Рекомендации по эксплуатации

4.1 Подготовка справочников

Перед началом регулярного учета рекомендуется заполнить справочники в логическом порядке. Сначала создаются категории и статусы, затем помещения и поставщики, после этого модели оборудования. Такой порядок уменьшает количество остановок при заполнении карточек, потому что все выпадающие списки уже будут содержать нужные значения.

В справочнике моделей поле характеристик допускает JSON. Его удобно использовать для технических параметров, которые не вынесены в отдельные поля: процессор, объем памяти, диагональ, интерфейсы, комплектация и другие сведения. JSON должен быть синтаксически корректным, например {"cpu": "Intel Core", "ram": "16 GB"}.

4.2 Ведение карточек оборудования

Инвентарный номер следует вводить в едином формате, принятом в организации. Изменять его можно, но каждое изменение создает новую версию карточки и будет отражено в отчете изменений. Серийный номер, цена и даты не являются обязательными, однако их заполнение повышает ценность отчетов и карточки.

Редактировать следует только актуальную карточку. Если пользователь открыл архивную версию через историю или ссылку на предыдущую запись, система покажет предупреждение и не позволит сохранить изменения. Для

изменения текущих данных нужно вернуться к актуальной записи из списка оборудования.

4.3 Работа с перемещениями

Перемещение следует использовать всякий раз, когда меняется помещение оборудования и требуется сохранить отдельный факт операции. Если нужно изменить только статус без изменения помещения, удобнее использовать действие «Сменить статус» в карточке. Если одновременно меняются многие поля, целесообразно открыть полное редактирование карточки.

Тип операции помогает интерпретировать движение: «Перемещение» подходит для обычной смены места, «Передача» — для передачи между ответственными зонами, «Ремонт» — для отправки в ремонтное помещение или сервисную зону, «Списание» — для фиксации вывода из эксплуатации при одновременном изменении местонахождения или статуса.

4.4 Администрирование доступа

Не рекомендуется выдавать роль 333 большему числу пользователей, чем требуется для сопровождения системы. Для повседневного учета обычно достаточно роли 311, а для руководителей или контролеров — роли с чтением оборудования и отчетов. При создании новых ролей следует описывать их назначение в поле «Описание».

После изменения ролей желательно проверить вход под учетной записью соответствующего типа или попросить пользователя подтвердить доступность нужных разделов. Так можно быстро обнаружить ситуацию, когда роль имеет чтение там, где требуется запись, или наоборот дает избыточный доступ.

4.5 Рекомендации по освоению

Освоение системы рекомендуется выполнять последовательно: сначала изучить модель прав ESR и общую навигацию, затем перейти к справочникам, после этого отработать создание карточки оборудования, редактирование актуальной версии, смену статуса, оформление перемещения и формирование отчетов. Такой порядок позволяет пользователю понять связь между справочниками, карточками, журналом операций и отчетными формами.

В качестве контрольного примера можно использовать тестовую единицу оборудования. Администратор или оператор создает тестовую категорию, статус, помещение и модель, после чего добавляет карточку оборудования с условным инвентарным номером. Затем пользователь выполняет смену статуса, создает перемещение в другое помещение, открывает карточку оборудования, проверяет цепочку версий и формирует отчет по оборудованию с фильтром по выбранной категории или статусу.

Контрольный пример считается выполненным корректно, если новая карточка отображается в списке оборудования, после редактирования предыдущая версия становится архивной, перемещение появляется в журнале, отчет учитывает актуальную версию карточки, а схема изменений показывает последовательность созданных версий. После завершения обучения тестовые записи следует удалить штатными средствами, если такая операция предусмотрена правами и политикой эксплуатации, либо пометить их как учебные в соответствии с правилами организации.

5 Термины и сокращения

Таблица 34. Термины руководства

Термин	Описание
Актуальная карточка	Последняя версия записи оборудования, у которой нет более новой дочерней версии.
Архивная карточка	Предыдущая версия оборудования, сохраненная для истории и недоступная для редактирования.
ESR	Трехпозиционная модель прав: Equipment, System, Reports.
Контур E	Оборудование, карточки, статусы оборудования, перемещения и связанные операции.
Контур S	Системные функции: справочники, пользователи, роли и административные операции.
Контур R	Отчеты, отчетные фильтры, CSV-выгрузка и диаграммы истории.
Mermaid	Текстовый формат описания диаграмм, используемый на странице схемы изменений.

Руководство по стилю

Цветовая схема

— Основной фон интерфейса — светло-серый RGB (248, 249, 250); фон карточек, таблиц и форм — белый RGB (255, 255, 255).

— Основной акцент интерфейса — синий RGB (13, 110, 253). Для границ используется серый RGB (222, 226, 230), для статусов — зелёный, жёлтый, красный и тёмно-серый цвета.

Таблица 35. Цветовая схема

Основной фон	Дополнительный фон	Акцентирование внимания
RGB (248, 249, 250) HEX #F8F9FA	RGB (255, 255, 255) HEX #FFFFFF RGB (222, 226, 230) HEX #DEE2E6	RGB (13, 110, 253) HEX #0D6EFD RGB (25, 135, 84) HEX #198754 RGB (220, 53, 69) HEX #DC3545

Шрифтовое оформление

— – Используется системный шрифт браузера без засечек: Segoe UI, Arial.

— – Основной текст — обычный, заголовки и кнопки — полужирные.

Таблица 36. Шрифтовое оформление

Элемент интерфейса	Размер	Начертание
Название системы и верхняя навигация	14-16 px	полужирное/обычное
Заголовок страницы	20-24 px	полужирное
Основной текст, подписи и поля форм	14-16 px	обычное
Таблицы и кнопки	14 px	обычное/полужирное

Основные элементы интерфейса

— – Верхняя навигационная панель выполняется в синем цвете и содержит основные разделы системы.

— – Рабочая область имеет светло-серый фон, а данные размещаются в белых карточках, таблицах и формах.

— – Основные кнопки выполняются в синем цвете, служебные статусы выделяются зелёным, жёлтым, красным или серым.

Таблица 37. Основные элементы интерфейса

Элемент	Оформление	Назначение
Верхнее меню	синий фон, белый текст	переход между разделами системы
Карточка / форма	белый фон, серые границы	размещение основной информации и полей ввода
Таблица	белый фон, тонкие серые линии	отображение списков оборудования, пользователей и отчетов
Основная кнопка	синий фон, белый текст	создание, сохранение, формирование отчета
Служебные метки	зелёный, жёлтый, красный, серый	отображение статусов и предупреждений

Таким образом, визуальное оформление веб-системы «Инвентарис» построено на светлом фоне, синих акцентных элементах, белых информационных блоках и нейтральных серых границах. Выбранная схема обеспечивает простое восприятие таблиц, форм и навигационных элементов без избыточного декоративного оформления.

Приложение В. Бриф на разработку сайта

Бриф на разработку АИС «Инвентарис»

1. Общая информация о проекте

Название системы:

Автоматизированная информационная система учёта оборудования «Инвентарис»

Заказчик:

ГПОАУ ЯО «ЯПЭК»

Цель проекта:

Создание веб-системы для централизованного учёта оборудования, контроля его состояния, перемещений и формирования отчётности

Проблема, которую решает система:

Отсутствие единого источника данных об оборудовании, ошибки учёта, сложность поиска информации и подготовки отчётов

2. Цели и задачи системы

Основная цель:

Обеспечить прозрачный, структурированный и актуальный учёт оборудования предприятия.

Ключевые задачи:

- регистрация и хранение данных об оборудовании;
- контроль статусов (эксплуатация, ремонт, списание и др.);
- фиксация перемещений;
- формирование отчётности;
- управление пользователями и правами доступа

3. Пользователи системы

Основные категории:

- администратор системы;
- технический персонал;
- сотрудники, ответственные за имущество;
- пользователи с ограниченным доступом

Вопросы:

1. Какие действия разрешены каждой роли?
2. Есть ли сценарии, где один пользователь временно получает расширенные права?
3. Нужна ли детализация прав (например, запрет удаления, но разрешение редактирования)?

4. Функциональные требования

Основной функционал:

- учёт оборудования (карточки объектов);
- управление статусами;
- учёт перемещений;
- журнал действий;
- формирование отчётов

Вопросы:

1. Какие поля обязательны для карточки оборудования?
2. Нужны ли уникальные инвентарные номера или допускаются дубликаты?
3. Какие типы отчётов критичны (например, списание, наличие, перемещения)?
4. Требуется ли экспорт (PDF, Excel)?
5. Нужна ли история изменений с возможностью отката?

5. Бизнес-процессы

Основные процессы:

- добавление оборудования;
- изменение статуса;
- перемещение между объектами;
- списание;
- формирование отчётов

Вопросы:

1. Кто инициирует перемещение и кто его подтверждает?
2. Нужны ли акты (документы) при операциях?
3. Есть ли регламентированные сценарии (например, списание только через администратора)?

6. Интерфейс и UX

Требования:

- простой и понятный интерфейс;
- табличное отображение данных;
- фильтрация и поиск;
- минимализм и деловой стиль

Вопросы:

1. Какие разделы должны быть на главной странице?
2. Какие данные пользователь должен видеть сразу?
3. Нужен ли дашборд (статистика, сводка)?
4. Есть ли требования к адаптивности (мобильные устройства)?

7. Технические требования

Общие требования:

- веб-приложение;
- работа через браузер;
- сервер: Linux/Windows;
- СУБД: PostgreSQL/MySQL

Вопросы:

1. Будет ли система разворачиваться в локальной сети или с доступом извне?
2. Требуется ли интеграция с существующими сервисами (авторизация, LDAP)?
3. Какие требования к производительности (кол-во пользователей)?

8. Безопасность и надёжность

Требования:

- разграничение прав доступа;
- журналирование действий;

— резервное копирование;

— защита от несанкционированного доступа

Вопросы:

1. Как часто выполнять резервное копирование?
2. Требуется ли двухфакторная аутентификация?
3. Какие данные считаются критическими?

9. Эксплуатация и сопровождение

Требования:

— наличие документации;

— обучение пользователей;

— регулярное обслуживание системы

Вопросы:

1. Кто будет администрировать систему?
2. Нужна ли система логирования ошибок?
3. Требуется ли интерфейс для настройки системы без разработчика?

10. Ожидаемый результат

В результате разработки должна быть создана веб-система, обеспечивающая:

— централизованный учёт оборудования;

— контроль его состояния и перемещений;

— удобный доступ к актуальной информации;

— снижение ошибок и нагрузки на персонал

Приложение Г. Техническое задание на разработку сайта

Техническое задание

1. Общие сведения

1.1 Полное наименование системы и её условное обозначение

Веб-система для учёта, инвентаризации и управления жизненным циклом оборудования предприятия. Далее называемая условным обозначением: АИС «Инвентарис».

1.2 Наименование разработчика и реквизиты заказчика

Заказчик: ГПОАУ ЯО «ЯПЭК».

Разработчик: Студент группы 22ИПВ, Бальчонок Роман Николаевич.

1.3 Основания для разработки АИС

Основанием разработки является учебное задание, выдаваемое ГПОАУ ЯО «ЯПЭК» в рамках производственной практики и выполнения проектной работы обучающегося.

Дополнительно разработка проводится в соответствии с требованиями следующих нормативных документов:

— ГОСТ 34.602-89 «Информационная технология. Техническое задание на создание автоматизированной системы»;

— ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания»;

— ГОСТ 34.201-2020 «Виды, комплектность и обозначение документов»;

— РД 50-34.698-90 «Требования к содержанию документов автоматизированных систем»;

— ГОСТ Р 2.105-2019 «ЕСКД. Общие требования к текстовым документам»;

— ГОСТ Р 56939-2024 «Защита информации. Основные положения»;

— ФЗ-149 «Об информации, информационных технологиях и защите информации»;

— ФЗ-152 «О персональных данных» (в части учёта пользователей системы)

1.4 Сроки выполнения работ

— Начало работ: 20 апреля 2026 г.

— Окончание работ: 30 мая 2026 г.

1.5 Источник финансирования

Учебная деятельность не предполагает коммерческого финансирования. Работы выполняются **в рамках учебного плана**, без привлечения внешних средств.

1.6 Порядок оформления и предъявления результатов

Результатом разработки являются:

— программный модуль веб-приложения;

— структура базы данных;

— документация пользователя;

— техническая и проектная документация.

Заказчику (ГПОАУ ЯО «ЯПЭК») передаются:

— электронный архив проекта;

— инструкции и методические материалы;

— доступ к демонстрационной версии сайта.

Документация разрабатывается в соответствии с ГОСТ 34.201-2020 и ГОСТ ЕСПД.

Приёмка результатов осуществляется комиссией ГПОАУ ЯО «ЯПЭК» в рамках учебного контроля.

2. Назначение и цели создания системы

2.1 Назначение системы

АИС «Инвентарис» предназначена для автоматизации учёта материально-технического оборудования, закреплённого за кластерами, помещениями, зонами и сотрудниками ярославского кампуса ГПОАУ ЯО «ЯПЭК». Система обеспечивает структурированное хранение данных, оперативную фиксацию изменений и формирование отчётной информации по оборудованию любой категории.

2.2 Цели создания системы

Основной целью разработки является создание единой цифровой среды, позволяющей обеспечить прозрачный и дисциплинированный контроль над

имуществом кампуса. Система должна обеспечить точный учёт вычислительной техники, периферийных устройств и вспомогательного оборудования, а также регламентировать процесс перемещения этих ресурсов внутри кампуса.

Дополнительными целями являются:

— снижение нагрузки на административный персонал;

— своевременное выявление неисправностей и случаев потери оборудования;

— сокращение числа ошибок при ведении учёта;

— обеспечение доступности актуальных данных для сотрудников кампуса;

— формирование отчётности по установленным критериям;

— демонстрация студентам примера современной автоматизированной системы, применяемой в рамках цифровой инфраструктуры ГПОАУ ЯО «ЯПЭК».

3. Характеристика объекта автоматизации

3.1 Краткие сведения об объекте

Объектом автоматизации является материально-техническая база ГПОАУ ЯО «ЯПЭК» в Ярославле, включающая рабочие станции, периферию, сетевое оборудование, оргтехнику и вспомогательные устройства, используемые в учебном процессе.

Пользователи системы:

— сотрудники администрации;

— специалисты технического отдела;

— кладовщик;

— сотрудник по работе с имуществом;

— бухгалтеры.

4. Требования к системе

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию

АИС должна обеспечивать следующие основные функции:

— учёт оборудования по категориям;

— фиксирование статусов («в эксплуатации», «в ремонте», «списано», «утеряно»);

— ведение истории перемещений;

— привязку оборудования к помещениям;

— генерацию отчётов.

Система должна функционировать в круглосуточном режиме, обеспечивая доступ уполномоченным пользователям в любое время суток.

4.1.2 Требования к средствам связи

Для взаимодействия компонентов системы используется стандартная локальная сеть кампуса ГПОАУ ЯО «ЯПЭК». Доступ к веб-приложению осуществляется через HTTP(S) протоколы.

4.1.3 Требования к совместимости

Система должна корректно функционировать в инфраструктуре кампуса, взаимодействуя с:

— внутренней сетью;

— сервисами аутентификации;

— сетевым оборудованием стандартов IEEE 802.3/802.11.

4.1.4 Требования к диагностированию

Проверка целостности данных, целостности БД и доступности серверной части осуществляется по мере необходимости, но не реже одного раза в месяц.

4.1.5 Требования к надёжности

Система должна сохранять работоспособность при:

— отказе отдельных программных модулей;

— действиях пользователя с ошибками;

— кратковременном отсутствии питания;

— отказах сетевых соединений.

4.1.6 Требования к персоналу

К работе с системой допускаются лица:

— владеющие навыками работы с ПК на уровне уверенного пользователя;

— прошедшие инструктаж по работе с системой.

4.1.7 Требования по безопасности

Эксплуатация системы должна соответствовать:

— ГОСТ 27201-87;

— требованиям ФЗ-152 в части обработки персональных данных;

— локальным регламентам ГПОАУ ЯО «ЯПЭК» по информационной безопасности.

4.1.8 Требования к эксплуатации и обслуживанию

Плановое обслуживание проводится не реже одного раза в месяц. Для обслуживания требуется наличие специалиста по сетевым технологиям или системного администратора.

4.1.9 Требования к сохранности информации

В системе должны быть предусмотрены:

— резервные копии БД;

— транзакционный контроль;

— механизм отката операций;

— раздельное хранение резервных носителей.

4.1.10 Требования к защите от внешних воздействий

Оборудование и ПО системы должны быть защищены от:

— электромагнитных помех;

— скачков напряжения;

— сетевых атак;

— несанкционированного доступа.

4.2 Требования к задачам, выполняемым системой

Система должна обеспечивать автоматизацию следующих функций:

— регистрация оборудования;

— учёт всех его параметров;

— изменение статуса;

— регистрация перемещения;

— оформление актов;

— формирование отчётов;

— управление пользователями.

4.3 Требования к видам обеспечения

4.3.1 Информационное обеспечение

Система хранит структурированную информацию об оборудовании и пользователях; дополнительные требования отсутствуют.

4.3.3 Программное обеспечение

На серверной стороне допускается использование:

— Windows или Linux;

— СУБД PostgreSQL/MySQL;

— веб-фреймворков уровня Flask/Django/Laravel/Node.js (по выбору разработчика).

Клиентская часть — любой современный браузер.

4.3.4 Техническое обеспечение

Специальных требований нет; достаточно стандартного серверного оборудования кампуса.

4.3.5 Методическое обеспечение

Должны быть разработаны:

— руководство пользователя;

— руководство администратора;

— методические рекомендации по эксплуатации

— руководства и рекомендации могут быть объединены в один документ.

5. Порядок контроля и приемки системы

Система подлежит приёмке после завершения всех работ, представления документации, предоставления демонстрационной версии и подтверждения её работоспособности.

Приёмка оформляется актом.

6. Требования к подготовке объекта автоматизации

До ввода системы необходимо:

— установить ПО;

— провести обучение пользователей;

— обеспечить сетевую конфигурацию;

— заполнить первичную базу данных.

7. Требования к документированию

Документы разрабатываются в соответствии с:

— ГОСТ 34.201-2020;

— ГОСТ 34.602-2020;

— ГОСТ Р 2.105-2019.

Предоставляются:

1. описание функций системы;
2. схема функциональной структуры;
3. схема БД;
4. руководство пользователя объединённое с методическими указаниями;
5. протокол испытаний.

Подписи:

Дата подписания технического задания: Подпись заказчика_____

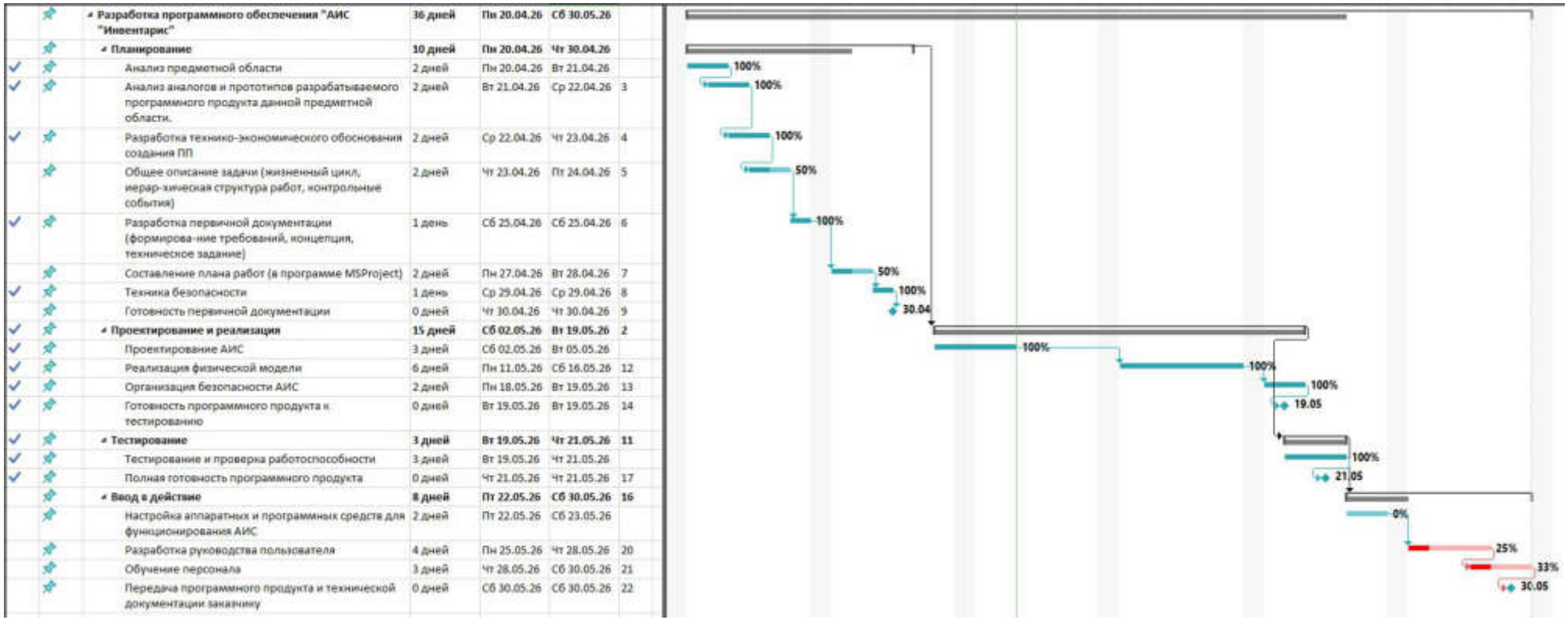
____/_____/2026 г.

Подпись исполнителя_____

Приложение Д. План работ и диаграмма Ганта

Таблица 38. План работ

№	Название задачи	Длительность	Начало	Окончание
1	**Разработка программного обеспечения «АИС «Инвентарис»»**	**36 дней**	**Пн 20.04.26**	**Сб 30.05.26**
1.1	**Планирование**	**10 дней**	**Пн 20.04.26**	**Чт 30.04.26**
1.2	Анализ предметной области	2 дней	Пн 20.04.26	Вт 21.04.26
1.3	Анализ аналогов и прототипов разрабатываемого программного продукта данной предметной области.	2 дней	Вт 21.04.26	Ср 22.04.26
1.4	Разработка технико-экономического обоснования создания ПП	2 дней	Ср 22.04.26	Чт 23.04.26
1.5	Общее описание задачи (жизненный цикл, иерархическая структура работ, контрольные события)	2 дней	Чт 23.04.26	Пт 24.04.26
1.6	Разработка первичной документации (формирование требований, концепция, техническое задание)	1 день	Сб 25.04.26	Сб 25.04.26
2	Составление плана работ (в программе MSProject)	2 дней	Пн 27.04.26	Вт 28.04.26
2.1	Техника безопасности	1 день	Ср 29.04.26	Ср 29.04.26
2.2	Готовность первичной документации	0 дней	Чт 30.04.26	Чт 30.04.26
2	**Проектирование и реализация**	**15 дней**	**Сб 02.05.26**	**Вт 19.05.26**
2.1	Проектирование АИС	3 дней	Сб 02.05.26	Вт 05.05.26
2.2	Реализация физической модели	6 дней	Пн 11.05.26	Сб 16.05.26
2.3	Организация безопасности АИС	2 дней	Пн 18.05.26	Вт 19.05.26
2.4	Готовность программного продукта к тестированию	0 дней	Вт 19.05.26	Вт 19.05.26
3	**Тестирование**	**3 дней**	**Вт 19.05.26**	**Чт 21.05.26**
3.1	Тестирование и проверка работоспособности	3 дней	Вт 19.05.26	Чт 21.05.26
3.2	Полная готовность программного продукта	0 дней	Чт 21.05.26	Чт 21.05.26
4	**Ввод в действие**	**8 дней**	**Пт 22.05.26**	**Сб 30.05.26**
4.1	Настройка аппаратных и программных средств для функционирования АИС	2 дней	Пт 22.05.26	Сб 23.05.26
4.2	Разработка руководства пользователя	4 дней	Пн 25.05.26	Чт 28.05.26
4.3	Обучение персонала	3 дней	Чт 28.05.26	Сб 30.05.26
4.4	Передача программного продукта и технической документации заказчику	0 дней	Сб 30.05.26	Сб 30.05.26



		Разработка программного обеспечения "АИС "Инвентарис"	36 дней	Пн 20.04.26	Сб 30.05.26	
		Планирование	10 дней	Пн 20.04.26	Чт 30.04.26	
✓		Анализ предметной области	2 дней	Пн 20.04.26	Вт 21.04.26	
✓		Анализ аналогов и прототипов разрабатываемого программного продукта данной предметной области.	2 дней	Вт 21.04.26	Ср 22.04.26	3
✓		Разработка технико-экономического обоснования создания ПП	2 дней	Ср 22.04.26	Чт 23.04.26	4
		Общее описание задачи (жизненный цикл, иерар-хическая структура работ, контрольные события)	2 дней	Чт 23.04.26	Пт 24.04.26	5
✓		Разработка первичной документации (формирова-ние требований, концепция, техническое задание)	1 день	Сб 25.04.26	Сб 25.04.26	6
		Составление плана работ (в программе MSProject)	2 дней	Пн 27.04.26	Вт 28.04.26	7
✓		Техника безопасности	1 день	Ср 29.04.26	Ср 29.04.26	8
		Готовность первичной документации	0 дней	Чт 30.04.26	Чт 30.04.26	9
✓		Проектирование и реализация	15 дней	Сб 02.05.26	Вт 19.05.26	2
✓		Проектирование АИС	3 дней	Сб 02.05.26	Вт 05.05.26	
✓		Реализация физической модели	6 дней	Пн 11.05.26	Сб 16.05.26	12
✓		Организация безопасности АИС	2 дней	Пн 18.05.26	Вт 19.05.26	13
✓		Готовность программного продукта к тестированию	0 дней	Вт 19.05.26	Вт 19.05.26	14
✓		Тестирование	3 дней	Вт 19.05.26	Чт 21.05.26	11
✓		Тестирование и проверка работоспособности	3 дней	Вт 19.05.26	Чт 21.05.26	
✓		Полная готовность программного продукта	0 дней	Чт 21.05.26	Чт 21.05.26	17
		Ввод в действие	8 дней	Пт 22.05.26	Сб 30.05.26	16
		Настройка аппаратных и программных средств для функционирования АИС	2 дней	Пт 22.05.26	Сб 23.05.26	
		Разработка руководства пользователя	4 дней	Пн 25.05.26	Чт 28.05.26	20
		Обучение персонала	3 дней	Чт 28.05.26	Сб 30.05.26	21
		Передача программного продукта и технической документации заказчику	0 дней	Сб 30.05.26	Сб 30.05.26	22

Рисунок 38. Диаграмма Ганта с отслеживанием

Приложение Е. Диаграмма IDEF0

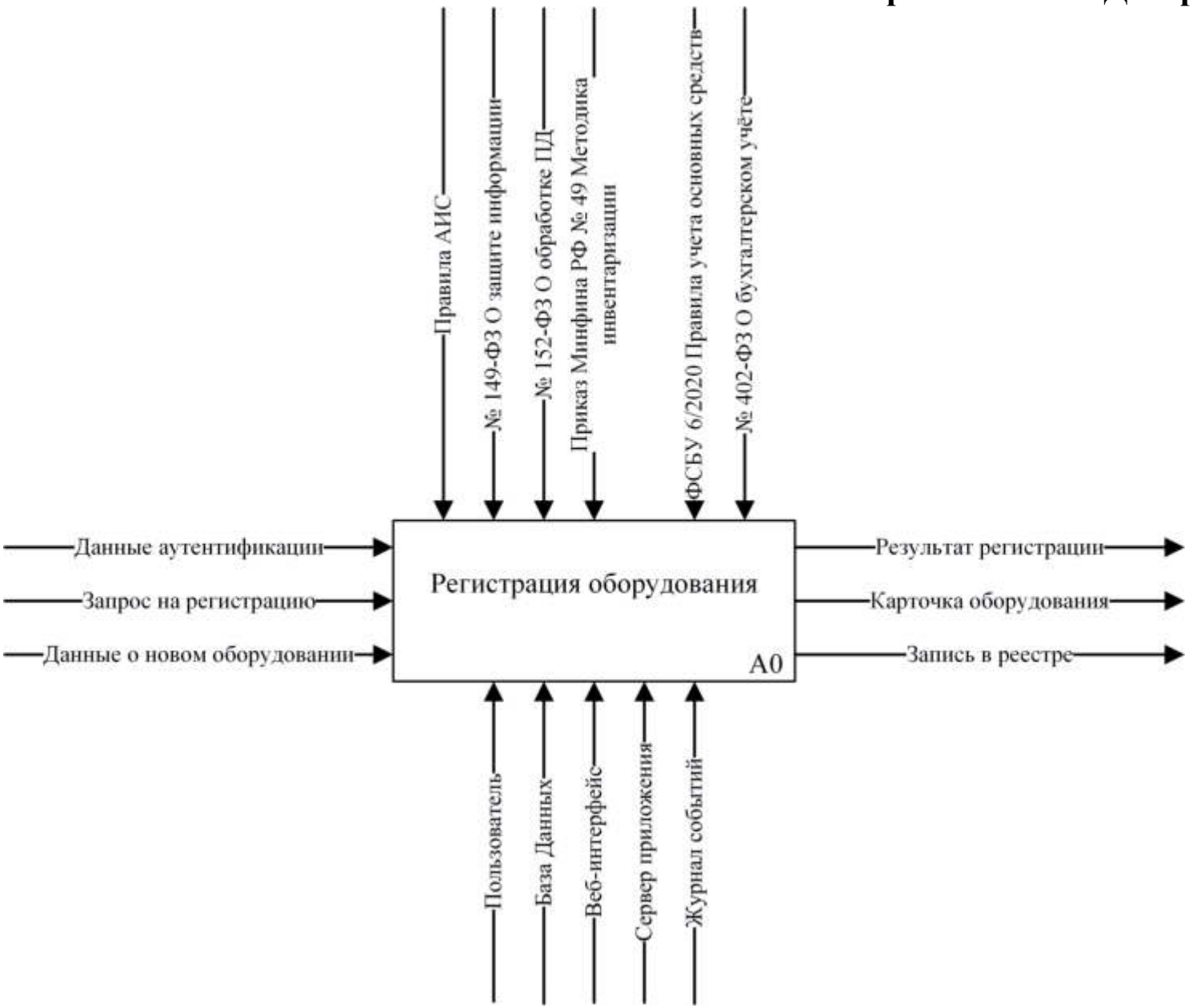


Рисунок 39. IDEF0 блок A0

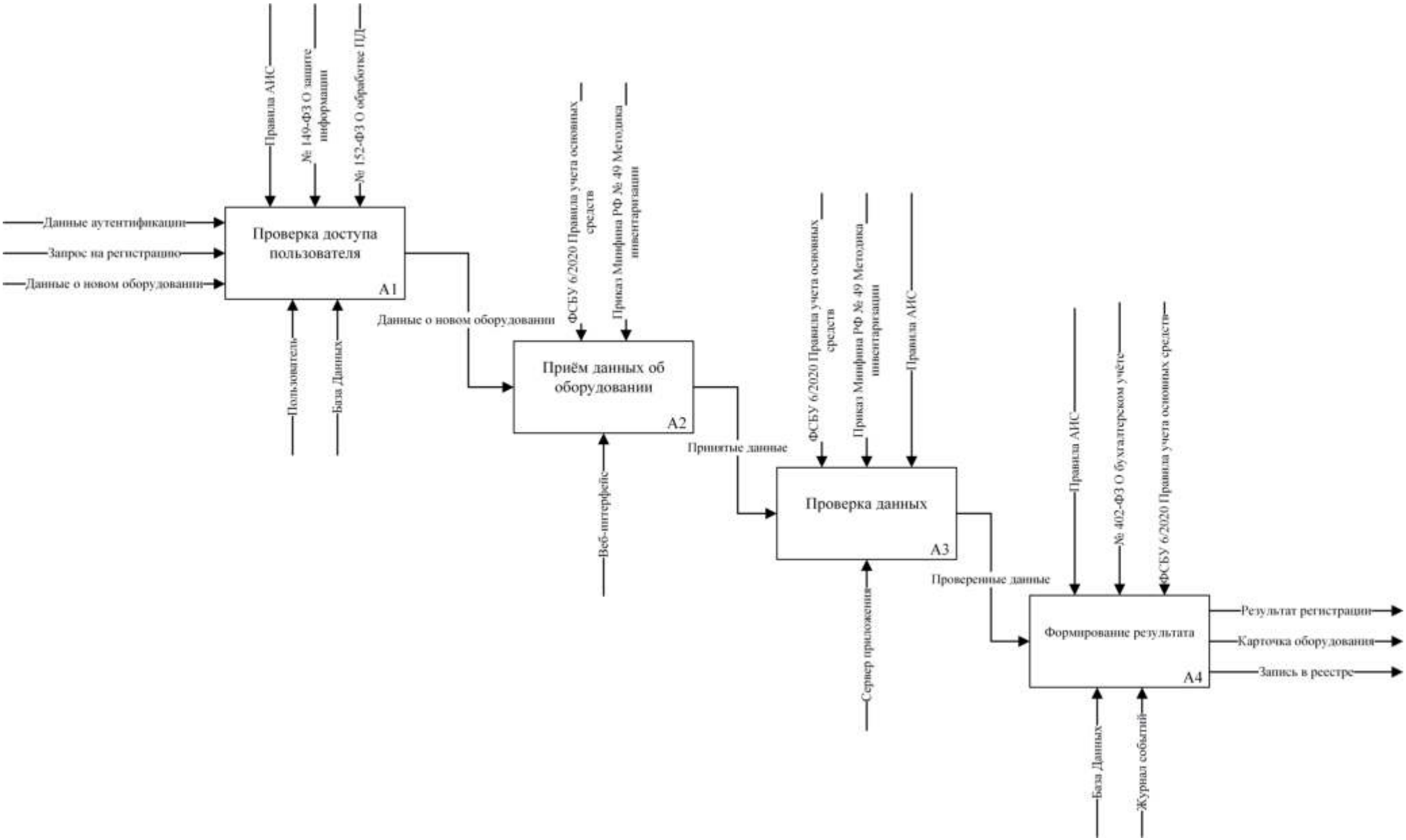


Рисунок 40. IDEF0 декомпозиция A1-A4

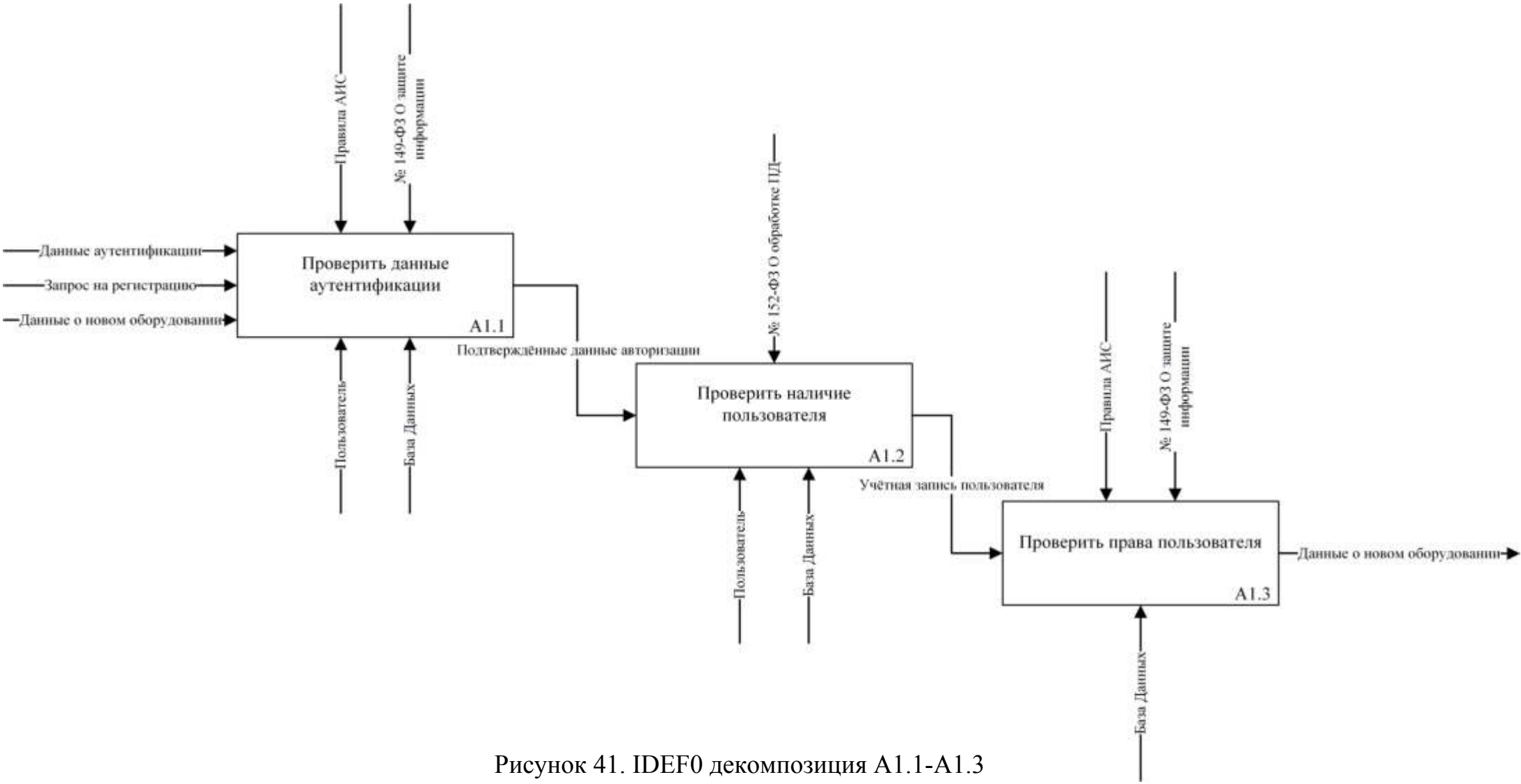


Рисунок 41. IDEF0 декомпозиция A1.1-A1.3

Приложение Ж. Диаграмма вариантов использования

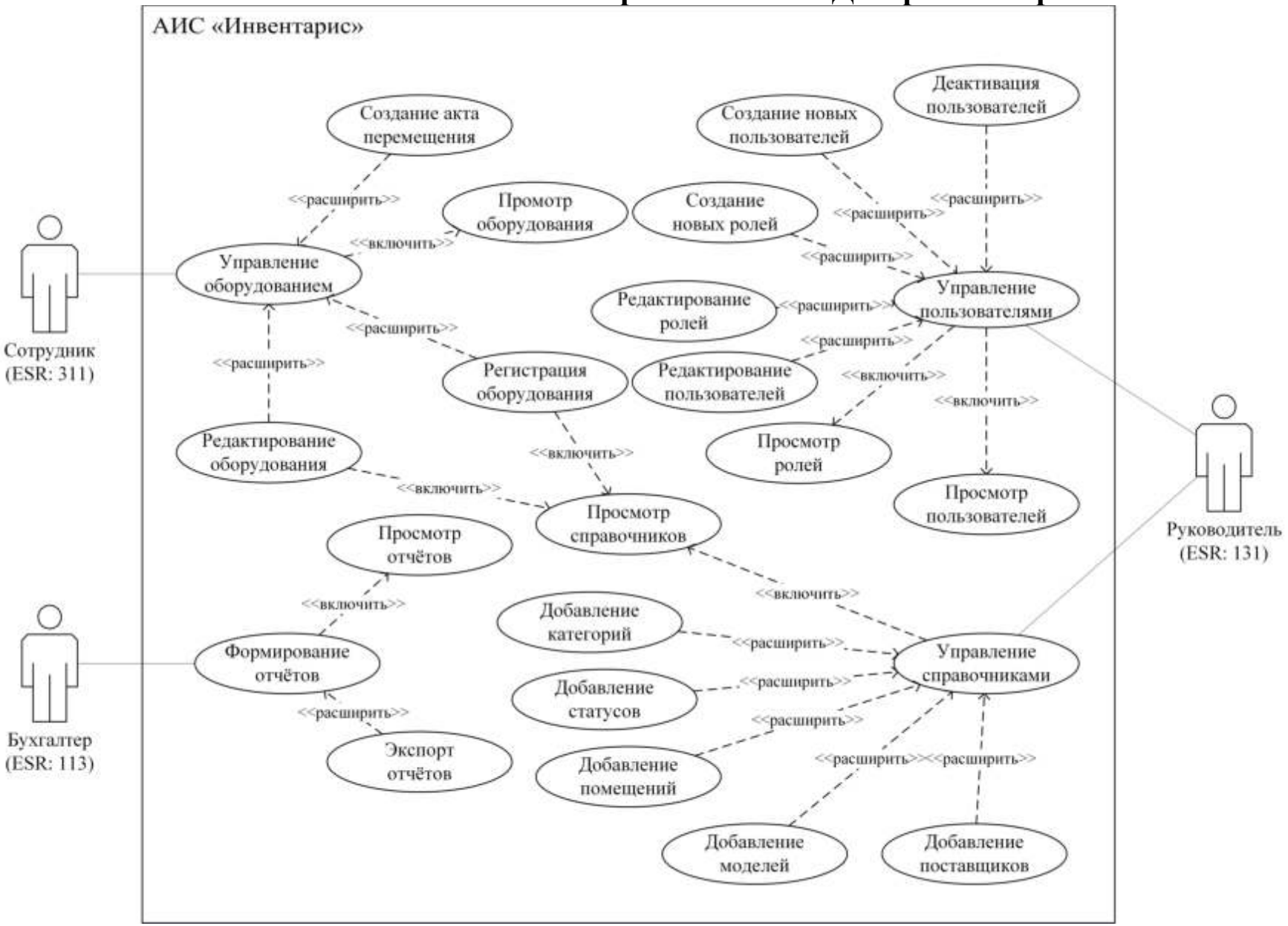


Рисунок 42. Диаграмма вариантов использования

Приложение 3. Диаграмма IDEF1X

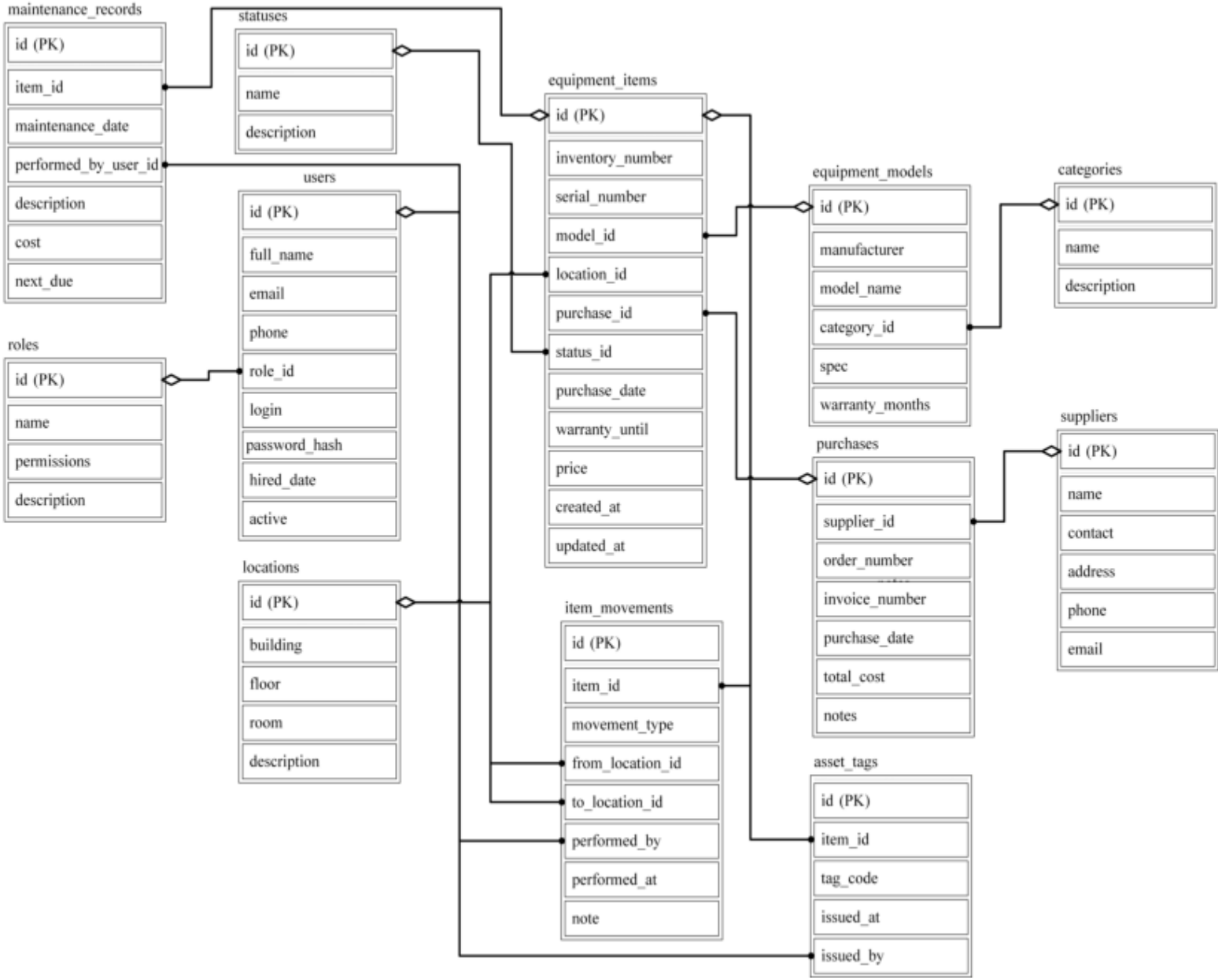


Рисунок 43. Схема базы данных

Приложение И. Диаграмма диаграмма последовательностей

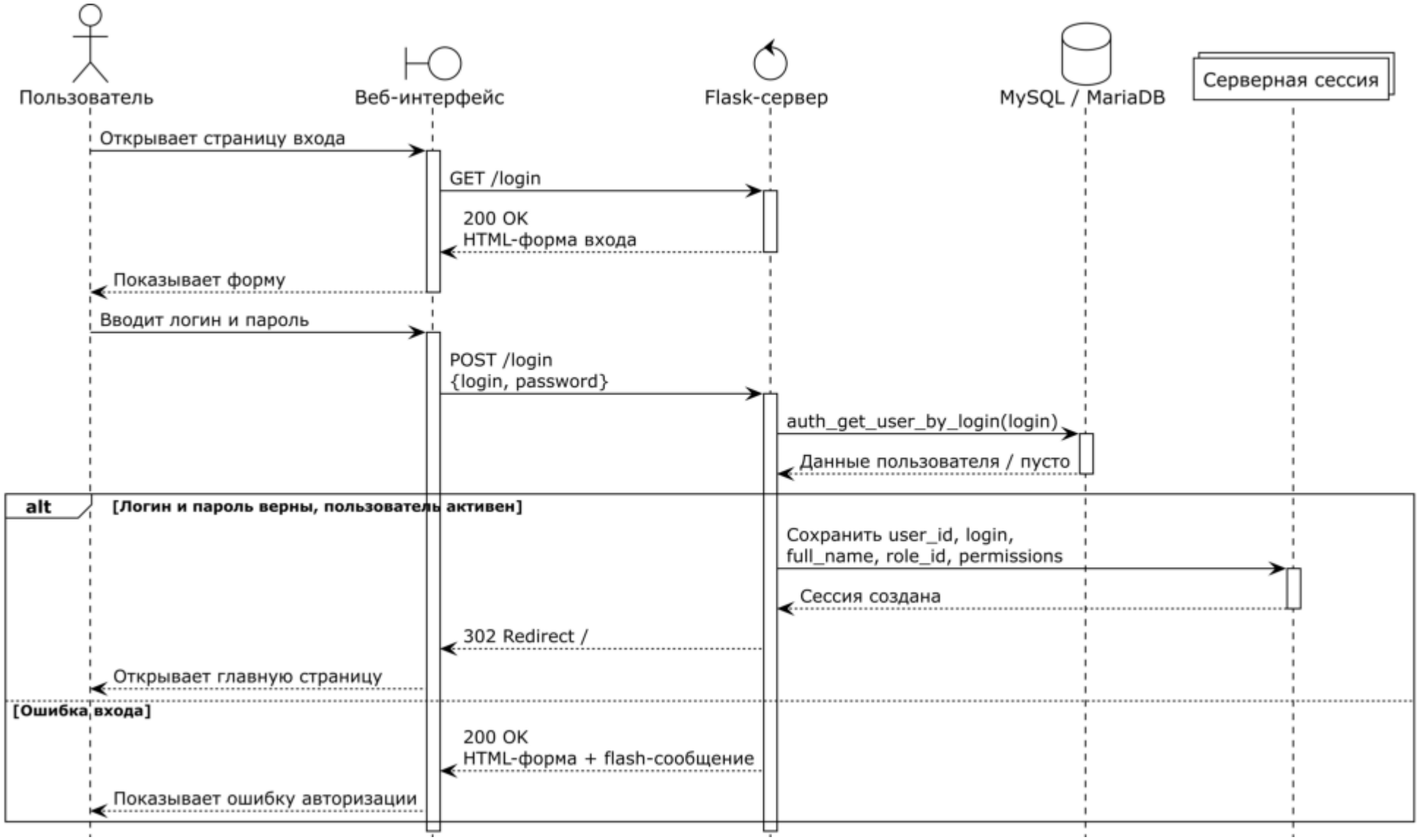


Рисунок 44. Диаграмма последовательностей: авторизация

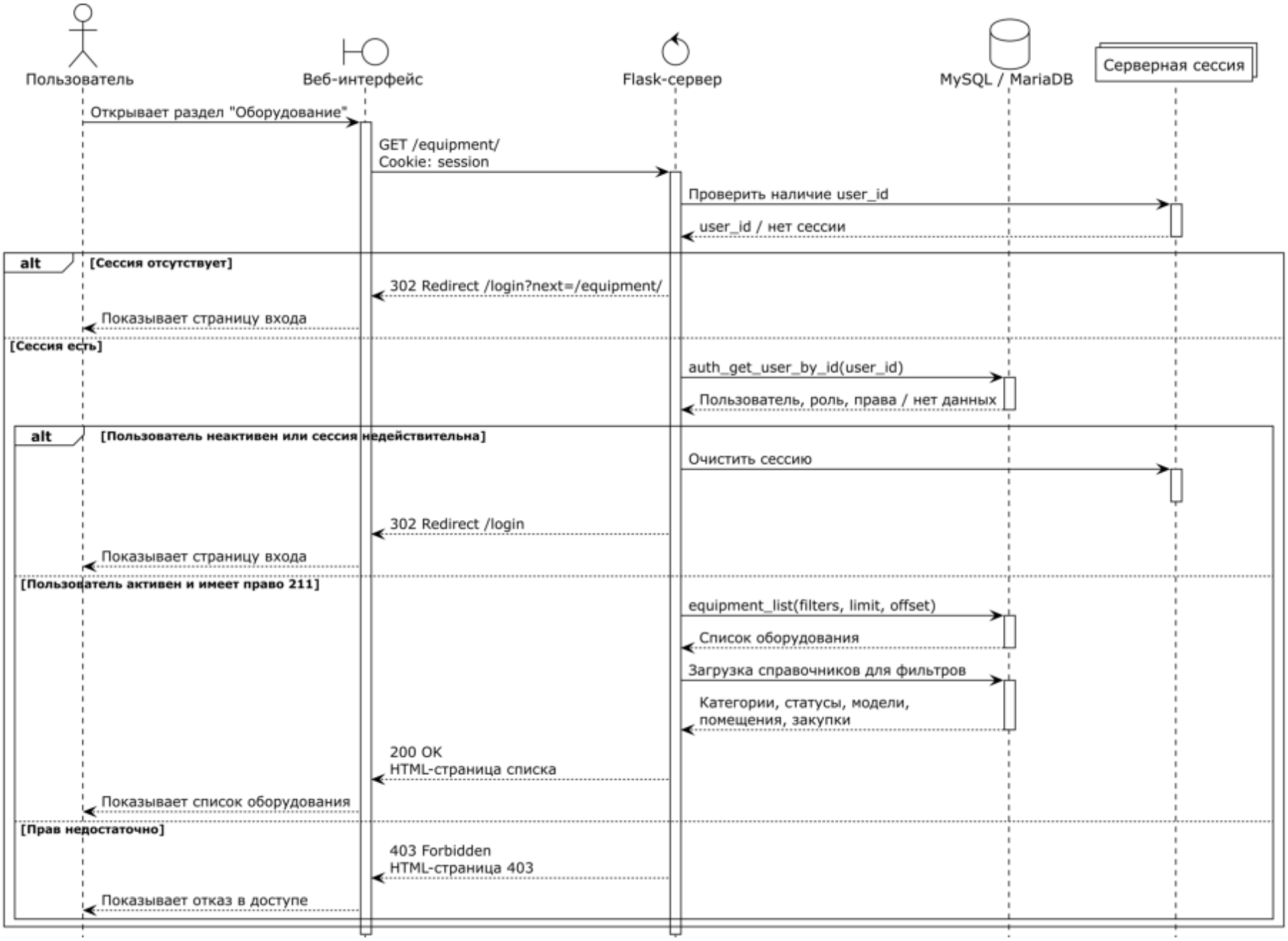


Рисунок 45. Диаграмма последовательностей: загрузка списка оборудования

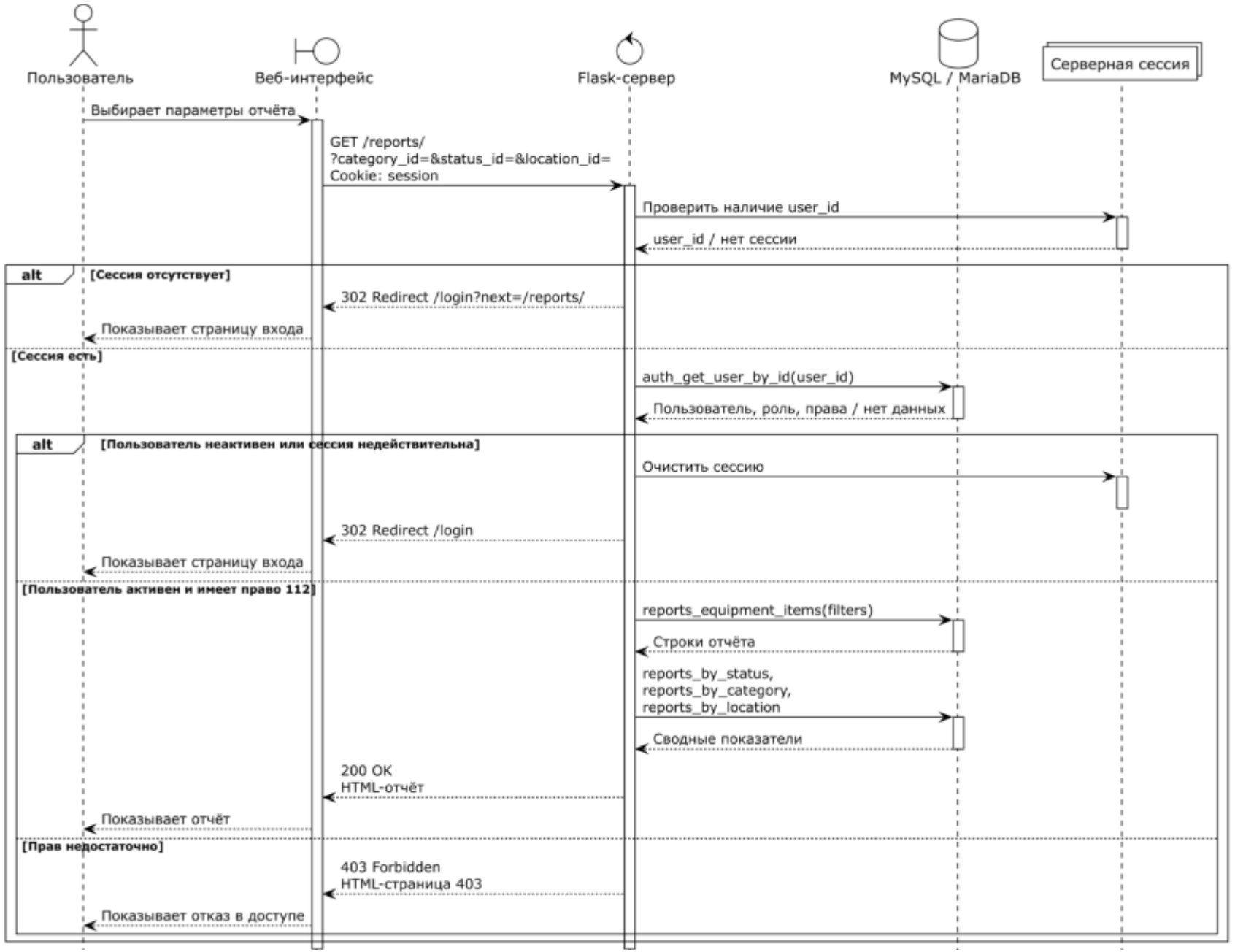


Рисунок 46. Диаграмма последовательностей: запрос отчёта

Приложение К. Общая концепция

Общая концепция
автоматизированной информационной системы «Инвентарис»

Общая концепция автоматизированной информационной системы «Инвентарис» разработана на основании результатов анализа предметной области, требований к создаваемому программному продукту и целей автоматизации процессов учёта оборудования предприятия. Документ определяет общее представление о назначении системы, принципах её функционирования, составе основных модулей, категориях пользователей, а также ключевых требованиях к интерфейсу, надёжности и защите данных. Концепция является промежуточным проектным документом, обеспечивающим переход от анализа предметной области к формализации требований в техническом задании и дальнейшему проектированию информационной системы.

Необходимость разработки данной системы обусловлена тем, что процессы регистрации, перемещения, контроля состояния и списания оборудования в организациях связаны с обработкой большого объёма однотипных данных и требуют постоянного поддержания их актуальности. При отсутствии централизованного цифрового решения сведения об оборудовании, его местоположении и текущем статусе, как правило, распределяются между таблицами, бумажными документами и локальными файлами, что приводит к ошибкам, задержкам при поиске информации и затруднениям при подготовке отчётности. В этих условиях внедрение специализированной веб-системы позволяет обеспечить единый контур хранения данных и повысить прозрачность операций с имуществом.

1 Общие положения и назначение системы

Разрабатываемая автоматизированная информационная система предназначена для учёта материально-технического оборудования предприятия и ориентирована на использование в рамках локальной корпоративной инфраструктуры. Система должна обеспечивать централизованное хранение

сведений об объектах учёта, регистрацию их характеристик, фиксацию статусов, ведение истории перемещений, а также поддержку формирования отчётной информации по заданным параметрам.

Концептуально АИС «Инвентарис» рассматривается как прикладное веб-приложение с разграничением прав доступа, единым пользовательским интерфейсом и серверной частью, обеспечивающей обработку, хранение и обновление данных. Такой подход позволяет использовать систему с различных рабочих мест без установки сложного клиентского программного обеспечения, что особенно важно для организации, в которой работа с учётными данными может выполняться несколькими категориями сотрудников.

В основе создаваемой системы лежит модель жизненного цикла оборудования. Каждый объект рассматривается не как статичная запись в реестре, а как сущность, состояние которой изменяется в процессе эксплуатации. Для каждого экземпляра оборудования должны храниться основные идентификационные сведения, сведения о категории, месте размещения, ответственном лице, текущем статусе и истории ранее выполненных операций. Это обеспечивает целостность учёта и возможность последующего анализа движения имущества во времени.

2 Цели разработки и ожидаемые результаты

Ключевой целью разработки является создание единой цифровой среды, обеспечивающей упорядоченный, достоверный и оперативный учёт оборудования предприятия. Внедрение системы должно способствовать снижению количества ошибок при ведении учётных записей, сокращению времени на поиск информации об объектах, повышению прозрачности движения имущества и улучшению качества подготовки внутренней отчётности.

Дополнительно система должна решать ряд прикладных задач, связанных с повседневной эксплуатацией имущества. К ним относятся регистрация новых единиц оборудования, изменение их статусов, оформление перемещений между помещениями и ответственными лицами, фиксация фактов ремонта, утраты или списания, а также получение выборок и отчётов по различным критериям. В

результате эксплуатации системы организация должна получить инструмент, позволяющий в любой момент времени определить фактическое и учётное состояние материально-технической базы.

Ожидаемым результатом реализации концепции является создание программного решения, пригодного для дальнейшего развития. Архитектура системы должна позволять расширение функциональности без радикального изменения базовой модели данных и логики интерфейса. Это означает, что на первоначальном этапе система реализует основной контур учёта оборудования, однако в дальнейшем может быть дополнена механизмами интеграции с внешними сервисами, расширенной аналитикой, дополнительными отчётными формами и иными прикладными возможностями.

3 Категории пользователей и принципы взаимодействия

Концепция системы предусматривает использование ролевой модели доступа. Это необходимо для разграничения функций пользователей, ограничения доступа к критически важным операциям и обеспечения ответственности за вносимые изменения. Категории пользователей определяются исходя из бизнес-процессов предметной области и включают администраторов системы, сотрудников технического подразделения, работников, ответственных за имущество, сотрудников администрации и пользователей с ограниченным правом просмотра.

Администратор системы должен обладать полным доступом к настройкам, управлению учётными записями, справочниками и журналом действий. Сотрудники, непосредственно работающие с оборудованием, должны иметь возможность регистрировать объекты, изменять их статус, оформлять перемещения и просматривать историю операций. Руководящий или административный персонал должен использовать систему преимущественно для поиска сведений, контроля состояния имущества и формирования отчётности. Ограниченные пользователи могут получать доступ только к просмотру части данных без возможности их редактирования.

Принципы взаимодействия пользователей с системой должны быть основаны на простоте и однозначности выполняемых действий. Основные операции должны быть доступны из логически понятных разделов интерфейса, а критически важные действия, связанные с удалением данных, изменением статуса или оформлением списания, должны сопровождаться подтверждением. Пользовательский сценарий должен быть построен таким образом, чтобы минимизировать вероятность ошибочного ввода и сократить количество лишних переходов между разделами.

4 Функциональная модель системы

Функциональная концепция АИС «Инвентарис» предполагает наличие набора взаимосвязанных подсистем, обеспечивающих выполнение основных задач учёта оборудования. Базовым элементом системы является модуль регистрации оборудования, в рамках которого создаются карточки объектов с указанием наименования, категории, инвентарного или условного номера, характеристик, местоположения и ответственного лица. Для обеспечения полноты учёта карточка объекта должна поддерживать последующее изменение данных с сохранением истории значимых операций.

Отдельное место в концепции занимает модуль управления состояниями оборудования. Он предназначен для фиксации текущего статуса объекта, например нахождения в эксплуатации, на ремонте, в резерве, в списании или в состоянии утраты. Изменение статуса должно производиться по установленным сценариям и сопровождаться записью в журнале действий. Такой механизм обеспечивает прослеживаемость жизненного цикла оборудования и позволяет исключить потерю информации о ранее совершённых операциях.

Следующая важная функциональная группа связана с учётом перемещений. Система должна поддерживать регистрацию фактов передачи оборудования между помещениями, подразделениями или ответственными лицами. Каждое перемещение должно фиксироваться как самостоятельное событие с указанием даты, участников операции и основания изменения размещения. Наличие этой функции принципиально важно для поддержания достоверности учётных данных и проведения последующих инвентаризационных сверок.

Функциональная концепция также включает средства поиска, фильтрации и формирования отчётов. Пользователь должен иметь возможность быстро находить нужные объекты по ключевым параметрам, формировать выборки по категориям, статусам, местам размещения и ответственным лицам, а также получать сводную информацию для целей контроля и анализа. Отчётный модуль должен быть ориентирован на подготовку внутренних документов и аналитических выборок, используемых при инвентаризации и управлении имуществом.

В состав системы должен входить и административный контур, обеспечивающий управление пользователями, ролями, справочниками и служебными параметрами. Наличие такого модуля необходимо для централизованного сопровождения системы, поддержания её актуальности и предотвращения несанкционированных действий со стороны пользователей.

5 Требования к структуре интерфейса и пользовательскому опыту

Концепция интерфейса ориентирована на практичность, визуальную простоту и предсказуемость поведения элементов управления. Основные разделы системы должны быть организованы по функциональному принципу и включать страницу обзора, раздел учёта оборудования, модуль перемещений, раздел формирования отчётов и административный блок. Навигация между разделами должна быть постоянной и интуитивно понятной, что позволяет сократить время адаптации пользователей к работе в системе.

При проектировании интерфейса особое внимание должно уделяться форме представления данных. Табличные представления должны обеспечивать удобный просмотр большого количества объектов, поддерживать фильтрацию и сортировку, а карточки оборудования должны содержать только значимую для пользователя информацию без перегрузки второстепенными деталями. Для операций ввода и редактирования данных следует использовать понятные формы с логичной группировкой полей и валидацией вводимых значений.

С точки зрения визуального оформления система должна сохранять деловой и нейтральный характер. Использование избыточных декоративных элементов нецелесообразно, поскольку основная задача интерфейса заключается в ускорении

повседневной работы пользователей. Вместо этого акцент должен быть сделан на читаемости текста, контрастности элементов управления, единообразии расположения кнопок и форм, а также на наличии явных сообщений об успешном выполнении операций или о возникающих ошибках.

Таблица 39. Основные модули общей концепции системы.

Модуль	Назначение	Основные функции
Учёт оборудования	Ведение карточек объектов	регистрация объектов, редактирование данных, просмотр характеристик
Статусы и история	Контроль состояния оборудования	изменение статусов, журналирование операций, просмотр истории
Перемещения	Учёт смены местоположения и ответственных лиц	оформление перемещений, фиксация даты и основания операции
Отчётность	Подготовка сводной информации	поиск, фильтрация, формирование отчётных выборок
Администрирование	Сопровождение системы	управление пользователями, ролями, справочниками и настройками

6 Требования к надёжности, безопасности и сопровождению

Так как система работает с данными об имуществе организации и содержит сведения о действиях пользователей, концепция предусматривает обязательное обеспечение сохранности, целостности и защищённости информации. На прикладном уровне это выражается в использовании механизмов аутентификации, разграничения прав доступа, регистрации ключевых действий пользователей и резервного копирования данных. Такой подход позволяет снизить риск несанкционированного доступа, случайного удаления информации и возникновения неустранимых расхождений в учётных данных.

С точки зрения надёжности система должна сохранять работоспособность при типовых пользовательских ошибках и кратковременных сбоях сетевого взаимодействия. Для этого логика обработки данных должна предусматривать проверки корректности вводимых значений, подтверждение критических операций, а также устойчивую серверную обработку запросов. Дополнительно

должна быть обеспечена возможность последующего сопровождения системы, включая обновление справочников, корректировку учётных данных уполномоченными лицами и техническое администрирование программного окружения.

Концепция также предполагает документирование основных принципов эксплуатации системы. Это необходимо для единообразного использования программного продукта различными категориями пользователей и для последующего перехода к полноценной эксплуатационной документации. Таким образом, вопросы безопасности и сопровождения закладываются не как дополнительные меры, а как обязательные свойства создаваемой информационной системы.

7 Вывод по общей концепции

В результате разработки общей концепции была сформирована целостная модель будущей автоматизированной информационной системы «Инвентарис», определяющая её назначение, функциональные границы, состав основных модулей, категории пользователей и общие требования к организации интерфейса и обработке данных. Концепция позволила зафиксировать, каким образом разрабатываемая система должна решать задачи учёта оборудования и какие принципы должны лежать в основе её дальнейшего проектирования.

Сформированная концепция выступает связующим звеном между анализом предметной области и техническим заданием. На её основе возможно дальнейшее уточнение требований, построение проектных моделей, разработка структуры базы данных и реализация программного продукта. Тем самым общая концепция не только описывает замысел создаваемой системы, но и определяет логическую основу всех последующих этапов дипломного проекта.

Приложение Л. Спецификация для диаграммы вариантов использования

Спецификация вариантов использования АИС «Инвентарис»

Настоящая спецификация составлена на основании актуальной диаграммы вариантов использования АИС «Инвентарис» и отражает основные действия акторов в пределах автоматизированной информационной системы. В документе приведено укрупнённое описание прецедентов без детализации альтернативных потоков, исключений и технических условий обработки данных, поскольку назначение спецификации состоит в кратком пояснении состава функциональных возможностей, показанных на диаграмме.

В качестве акторов на диаграмме выделены сотрудник, бухгалтер и руководитель. Сотрудник выполняет действия, связанные с учётом оборудования, регистрацией инвентарных единиц и использованием справочных данных при заполнении карточек. Бухгалтер использует систему для просмотра, формирования и экспорта отчётной информации. Руководитель осуществляет сопровождение пользователей, ролей и справочников, необходимых для администрирования и поддержания корректной структуры данных АИС.

Прецедент: «Управление оборудованием»

Описание: Сотрудник переходит к разделу учёта оборудования и выполняет операции с карточками объектов инвентаризации. В рамках данного прецедента система предоставляет доступ к просмотру сведений об оборудовании, регистрации новых единиц, редактированию ранее внесённых данных и созданию акта перемещения при изменении места эксплуатации, ответственного лица или иных учётных параметров объекта.

Прецедент: «Просмотр оборудования»

Описание: Сотрудник открывает перечень оборудования и получает сведения о зарегистрированных объектах, включая инвентарный номер, наименование, модель, категорию, статус, помещение, поставщика и иные данные, предусмотренные структурой карточки. Система отображает актуальную информацию в удобном для просмотра виде и позволяет использовать её как

основание для последующего редактирования, регистрации связанных событий или формирования документов.

Прецедент: «Регистрация оборудования»

Описание: Сотрудник создаёт новую запись об оборудовании, заполняя основные характеристики объекта и выбирая значения из справочников. После проверки заполненных данных система сохраняет карточку оборудования в базе данных, связывает её с указанными категорией, моделью, статусом, помещением и поставщиком, после чего объект становится доступен в общем перечне оборудования.

Прецедент: «Редактирование оборудования»

Описание: Сотрудник выбирает ранее зарегистрированную единицу оборудования и вносит изменения в её карточку, если сведения были уточнены или изменились в процессе эксплуатации. Система принимает обновлённые значения, проверяет корректность обязательных полей и сохраняет новую редакцию записи, обеспечивая актуальность информации в учётной части АИС.

Прецедент: «Создание акта перемещения»

Описание: Сотрудник оформляет акт перемещения оборудования в случае передачи объекта между помещениями, подразделениями или ответственными лицами. Система использует сведения из карточки оборудования и введённые пользователем данные о перемещении, после чего формирует документ, подтверждающий изменение текущего местонахождения или эксплуатационного состояния объекта.

Прецедент: «Формирование отчётов»

Описание: Бухгалтер обращается к разделу отчётности и задаёт параметры формирования отчёта по оборудованию, его размещению, состоянию или движению. Система обрабатывает имеющиеся учётные данные и формирует отчётную информацию в виде, пригодном для анализа, контроля и последующего использования при подготовке внутренних документов предприятия.

Прецедент: «Просмотр отчётов»

Описание: Бухгалтер открывает сформированные или доступные в системе отчёты и анализирует сведения об оборудовании, инвентарных единицах, помещениях, статусах и иных показателях. Система предоставляет данные в табличном или ином представлении, достаточном для проверки полноты учёта и выявления расхождений в сведениях об объектах.

Прецедент: «Экспорт отчётов»

Описание: Бухгалтер выполняет выгрузку отчётной информации из системы для дальнейшего хранения, передачи или использования во внешних документах. Система формирует экспортируемый файл на основании выбранных параметров отчёта и предоставляет пользователю результат в формате, пригодном для последующей обработки.

Прецедент: «Просмотр справочников»

Описание: Пользователь с соответствующими правами запрашивает перечень справочных значений, которые применяются при заполнении карточек оборудования и сопровождении учётных данных. Система возвращает актуальные списки категорий, статусов, помещений, моделей, поставщиков и иных значений, чтобы пользователь мог выбирать существующие записи без повторного ручного ввода и снижать риск расхождений в данных.

Прецедент: «Управление справочниками»

Описание: Руководитель переходит к разделу справочников и выполняет действия по сопровождению нормативно-справочной информации, используемой в карточках оборудования и отчётности. В рамках данного прецедента система предоставляет просмотр справочников и возможность добавления новых категорий, статусов, помещений, моделей и поставщиков, необходимых для корректного ведения инвентарного учёта.

Прецедент: «Добавление категорий»

Описание: Руководитель вносит в справочник новую категорию оборудования, если существующего значения недостаточно для классификации объектов. Система сохраняет запись в справочнике категорий и делает её

доступной при регистрации и редактировании оборудования, что обеспечивает единообразную группировку инвентарных единиц.

Прецедент: «Добавление статусов»

Описание: Руководитель добавляет новый статус оборудования, отражающий состояние объекта в процессе эксплуатации, ремонта, списания или иного этапа жизненного цикла. После сохранения система включает статус в перечень доступных значений и позволяет использовать его при заполнении карточек оборудования и формировании отчётности.

Прецедент: «Добавление помещений»

Описание: Руководитель регистрирует новое помещение в справочнике, если оборудование необходимо закрепить за местом эксплуатации, отсутствующим в текущем перечне. Система сохраняет указанное помещение и предоставляет возможность использовать его при регистрации оборудования, редактировании карточек и оформлении перемещения.

Прецедент: «Добавление моделей»

Описание: Руководитель добавляет новую модель оборудования в справочник, указывая сведения, необходимые для последующего выбора модели в карточке объекта. После сохранения система включает модель в перечень доступных значений, что позволяет корректно классифицировать оборудование и поддерживать единообразие справочных данных.

Прецедент: «Добавление поставщиков»

Описание: Руководитель вносит сведения о новом поставщике оборудования, если соответствующая организация отсутствует в справочнике. Система сохраняет запись о поставщике и делает её доступной для выбора в карточках оборудования, отчётах и иных связанных операциях, где требуется указание источника поступления объекта.

Прецедент: «Управление пользователями»

Описание: Руководитель переходит в раздел управления пользователями и выполняет операции по сопровождению учётных записей и их полномочий. В рамках данного прецедента система предоставляет функции просмотра

пользователей, создания новых пользователей, редактирования данных, деактивации учётных записей, просмотра ролей, создания новых ролей и редактирования существующих ролей.

Прецедент: «Просмотр пользователей»

Описание: Руководитель открывает список пользователей системы и получает сведения об учётных записях, их текущем состоянии и назначенных ролях. Система отображает данные, необходимые для контроля состава пользователей и принятия решений о создании, изменении или деактивации отдельных записей.

Прецедент: «Создание новых пользователей»

Описание: Руководитель создаёт новую учётную запись для лица, которому должен быть предоставлен доступ к АИС. Система принимает основные данные пользователя, связывает запись с выбранной ролью или набором прав и сохраняет её в базе данных, после чего пользователь может быть допущен к работе в пределах назначенных полномочий.

Прецедент: «Редактирование пользователей»

Описание: Руководитель изменяет сведения существующей учётной записи, если необходимо уточнить данные пользователя, изменить состояние записи или скорректировать назначенные права доступа. Система проверяет внесённые изменения и сохраняет их, обеспечивая соответствие пользовательских полномочий текущим организационным требованиям.

Прецедент: «Деактивация пользователей»

Описание: Руководитель выводит из активного использования учётную запись пользователя, которому больше не требуется доступ к системе. Система выполняет деактивацию только при наличии соответствующих прав и сохраняет сведения об учётной записи в базе данных, исключая её дальнейшее применение для входа и выполнения операций в АИС.

Прецедент: «Просмотр ролей»

Описание: Руководитель открывает перечень ролей, используемых в системе для разграничения доступа к функциональным возможностям. Система отображает состав доступных ролей и связанные с ними параметры прав, чтобы руководитель мог оценить действующую модель доступа перед созданием, назначением или редактированием ролей.

Прецедент: «Создание новых ролей»

Описание: Руководитель создаёт новую роль для разграничения доступа к функциям системы. Система сохраняет роль и связанные с ней параметры доступа, после чего она может применяться при создании или редактировании пользователей, что позволяет поддерживать управляемую модель полномочий внутри АИС.

Прецедент: «Редактирование ролей»

Описание: Руководитель изменяет параметры существующей роли, если требуется уточнить состав прав или адаптировать модель доступа под организационные изменения. Система проверяет корректность заданных параметров, сохраняет новую конфигурацию роли и применяет её при дальнейшем назначении полномочий пользователям.

Приложение М. Глоссарий к диаграмме IDEF0

Глоссарий к диаграмме IDEF0
«регистрация оборудования»

Настоящий глоссарий составлен к функциональной модели IDEF0 процесса регистрации оборудования. В глоссарии приведены краткие определения основных функциональных блоков, входных и выходных потоков, управляющих воздействий, механизмов и промежуточных данных, используемых на контекстной диаграмме A0, диаграмме декомпозиции A1–A4 и диаграмме уточнения блока A1 до уровня A11–A13. Описание носит справочный характер и предназначено для единообразного понимания элементов модели при включении диаграмм в проектную документацию.

1 Функциональные блоки модели

Таблица 40. Функциональные блоки модели IDEF0

Код	Наименование	Тип элемента	Краткое определение
A0	Регистрация оборудования	Контекстный блок	Обобщённый процесс обработки запроса на регистрацию нового оборудования с последующим формированием результата, карточки оборудования и записи в реестре.
A1	Проверка доступа пользователя	Блок декомпозиции A0	Проверка возможности пользователя выполнить операцию регистрации оборудования на основании данных аутентификации, запроса и внутренних правил АИС.
A2	Приём данных об оборудовании	Блок декомпозиции A0	Получение сведений о новом оборудовании через веб-интерфейс после прохождения проверки доступа пользователя.
A3	Проверка данных	Блок декомпозиции A0	Контроль полноты и корректности принятых сведений об оборудовании с учётом правил АИС и требований учёта.
A4	Формирование результата	Блок декомпозиции A0	Создание итогового результата регистрации, карточки оборудования и записи в реестре на основании проверенных данных.
A1.1	Проверить данные аутентификации	Блок декомпозиции A1	Первичная проверка переданных пользователем данных аутентификации и формирование подтверждённых данных авторизации.
A1.2	Проверить наличие пользователя	Блок декомпозиции A1	Проверка существования соответствующей учётной записи пользователя в базе данных.
A1.3	Проверить права пользователя	Блок декомпозиции A1	Проверка полномочий найденной учётной записи на выполнение регистрации нового оборудования.

2 Входные, выходные и промежуточные потоки данных

Таблица 41. Потоки данных модели

Код	Наименование	Тип элемента	Краткое определение
I1	Данные аутентификации	Вход	Сведения, используемые для идентификации пользователя и проверки возможности доступа к функции регистрации оборудования.
I2	Запрос на регистрацию	Вход	Инициирующее обращение пользователя к АИС для выполнения операции регистрации нового оборудования.
I3	Данные о новом оборудовании	Вход / поток данных	Сведения о регистрируемом оборудовании, передаваемые в систему и далее используемые при приёме, проверке и регистрации.
F1	Подтверждённые данные авторизации	Промежуточный поток	Результат успешной проверки данных аутентификации, передаваемый на этап проверки наличия пользователя.
F2	Учётная запись пользователя	Промежуточный поток	Найденная в базе данных запись о пользователе, используемая при проверке его прав.
F3	Принятые данные	Промежуточный поток	Данные об оборудовании, полученные через веб-интерфейс и переданные на этап проверки.
F4	Проверенные данные	Промежуточный поток	Сведения об оборудовании, прошедшие проверку полноты, корректности и соответствия установленным правилам.
O1	Результат регистрации	Выход	Итог обработки запроса, отражающий успешное или неуспешное завершение процедуры регистрации оборудования.
O2	Карточка оборудования	Выход	Структурированное представление сведений о зарегистрированном оборудовании для просмотра и дальнейшей работы в АИС.
O3	Запись в реестре	Выход	Сформированная запись в реестре оборудования, предназначенная для хранения и последующего учёта.

3 Управляющие воздействия

Таблица 42. Управляющие элементы модели

Код	Наименование	Тип элемента	Краткое определение
C1	Правила АИС	Управление	Внутренние правила работы системы, определяющие порядок проверки доступа, обработки данных и формирования результата регистрации.
C2	№ 149-ФЗ «О защите информации»	Управление / НПА	Нормативное ограничение, учитываемое при проверке аутентификации, доступа и защиты обрабатываемой информации.
C3	№ 152-ФЗ «О персональных данных»	Управление / НПА	Требования к обработке персональных данных пользователя при проверке его наличия, учётной записи и прав доступа.
C4	Приказ Минфина РФ № 49, методика инвентаризации	Управление / методика	Методические положения, используемые при приёме и проверке сведений об оборудовании в рамках инвентаризационного учёта.
C5	ФСБУ 6/2020, правила учёта основных средств	Управление / НПА	Правила учёта основных средств, применяемые при обработке и проверке сведений о регистрируемом оборудовании.
C6	№ 402-ФЗ «О бухгалтерском учёте»	Управление / НПА	Нормативное основание для формирования учётной записи в реестре и итоговых регистрационных данных.

4 Механизмы и ресурсы выполнения процесса

Таблица 43. Механизмы модели

Код	Наименование	Тип элемента	Краткое определение
M1	Пользователь	Механизм	Участник процесса, инициирующий запрос на регистрацию и передающий данные, необходимые для выполнения операции.
M2	База данных	Механизм	Средство хранения и предоставления сведений об учётных записях, оборудовании и результатах регистрации.
M3	Веб-интерфейс	Механизм	Пользовательский интерфейс АИС, через который вводятся и передаются сведения о новом оборудовании.
M4	Сервер приложения	Механизм	Программная часть АИС, выполняющая обработку запросов, проверку данных и формирование результата.
M5	Журнал событий	Механизм	Средство фиксации действий и событий, связанных с регистрацией оборудования и изменением данных в системе.

Таким образом, приведённый глоссарий фиксирует основные термины, используемые на представленных диаграммах IDEF0, и обеспечивает связность между контекстным уровнем модели, декомпозицией процесса регистрации оборудования и уточнением процедуры проверки доступа пользователя.