Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ



ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ УЧЕТА ОБОРУДОВАНИЯ И РАБОТЫ ЛАБОРАНТОВ «Мой ПТК»

Курсовой проект

ПМ.02. ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОФЕССИОНАЛЬНЫХ МОДУЛЕЙ

МДК.02.01 технология разработки программного обеспечения

Пояснительная записка курсового проекта по специальности  
09.02.07 Информационные системы и программирование

Руководитель

Л.Н. Цымбалюк  
« » 2025 года

Студент группы 3991

И.И.Барышников

« » 2025 года



Аннотация

Настоящий документ представляет собой пояснительную записку к курсовому проекту на тему «Разработка информационной системы контроля оборудования и учета работы лаборантов “Мой ПТК”». Пояснительная записка содержит подробное описание разработки программного продукта, предназначенного для оптимизации процессов инвентаризации, учета, перемещения и обслуживания техники в Политехническом колледже ПТИ НовГУ.

В разделе “Введение” сформулирована задача, определены цели и задачи проекта, обоснован выбор технологий разработки и инструментальных средств. Также представлены требования к функциональным и нефункциональным характеристикам разрабатываемой системы.

В разделе “Техническое описание” представлено технико-математическое описание задачи, включая математические методы, используемые для отображения интерактивного плана, и технологии, применяемые для разработки клиентской и серверной частей приложения. Подробно описана архитектура системы.

Раздел “Характеристика бизнес-процессов” содержит описание бизнес-процессов, которые автоматизируются с помощью разрабатываемого программного продукта, таких как инвентаризация, учет перемещения, техническое обслуживание и списание техники.

Пояснительная записка также включает описание функциональности системы, включая входные, выходные и промежуточные данные для каждой функции, требования к безопасности и быстродействию, а также требования к составу и параметрам технических средств.

Содержание

[Введение 4](#_Toc197639378)

[1 Общая (теоретическая) часть 6](#_Toc197639379)

[1.1 Постановка задачи 6](#_Toc197639380)

[1.2 Обоснование проектных решений 10](#_Toc197639381)

[1.3 Обзор и анализ существующих программных систем 12](#_Toc197639382)

[2 Практическая часть 14](#_Toc197639383)

[2.1 Анализ задачи 14](#_Toc197639384)

[2.2 Реализация 18](#_Toc197639385)

[2.3 Руководство оператора 26](#_Toc197639386)

[Заключение 28](#_Toc197639387)

[Список литературы 29](#_Toc197639388)

[Глоссарий 31](#_Toc197639389)

[Приложение А 32](#_Toc197639390)

[Приложение Б 37](#_Toc197639391)

[Приложение В 38](#_Toc197639392)

[Приложение Г 40](#_Toc197639393)

[Приложение Д 43](#_Toc197639394)

[Приложение Е 48](#_Toc197639395)

[Приложение Ж 55](#_Toc197639396)

[Приложение З 59](#_Toc197639397)

# Введение

В современных образовательных учреждениях, особенно в колледжах и техникумах, важной задачей является эффективный учет оборудования и контроль работы персонала, в частности лаборантов. Автоматизация этих процессов позволяет оптимизировать использование ресурсов, повысить ответственность сотрудников и обеспечить прозрачность учета материальных ценностей.

Целью данного курсового проекта является разработка информационной системы учета оборудования и работы лаборантов "Мой ПТК" для образовательного учреждения. Система должна предоставлять удобный инструментарий для:

* учета и классификации оборудования по типам и местоположению;
* контроля состояния и технических характеристик оборудования;
* учета рабочего времени и обязанностей лаборантов;
* генерации отчетов и аналитики.

Объектом исследования является процесс учета оборудования и организации работы лаборантов в профессионально-техническом колледже.

Основные задачи проекта:

* исследовать предметную область и требования к системе учета;
* разработать прототип и структуру базы данных;
* реализовать веб-приложение с использованием Django, DRF и JavaScript;
* протестировать готовый продукт на соответствие требованиям;
* разработать документацию к системе.

Ожидаемым результатом является полнофункциональная информационная система, которая позволит:

* вести централизованный учет всего оборудования учреждения;
* контролировать его состояние и перемещение между кабинетами;
* учитывать рабочее время лаборантов и их зоны ответственности;
* формировать различные отчеты для администрации.

Методы исследования, использованные в работе:

* анализ предметной области - изучение существующих систем учета и требований образовательных учреждений;
* моделирование - проектирование структуры базы данных и бизнес-процессов;
* теоретический анализ - исследование технологий для реализации (Django, DRF, JavaScript);
* экспериментальный метод - тестирование и доработка системы.

Практическая значимость проекта заключается в создании специализированного программного обеспечения, которое:

* повысит эффективность учета материальных ценностей;
* упростит контроль за состоянием оборудования;
* автоматизирует отчетность;
* может быть адаптировано для различных образовательных учреждений.

Разработанная система "Мой ПТК" представляет собой современное решение для автоматизации важных процессов в профессиональных учебных заведениях и может быть внедрена в реальную практику работы колледжей и техникумов.

# 1 Общая (теоретическая) часть

## 1.1 Постановка задачи

### 1.1.1 Обоснование необходимости разработки

В соответствии с заданием, требуется разработать информационную систему контроля и учета оборудования и работы лаборантов “Мой ПТК”. Разработка информационной системы контроля и учета оборудования и работы лаборантов “Мой ПТК” обусловлена необходимостью оптимизировать процессы, связанные с управлением техническими ресурсами в Политехническом колледже НовГУ Существующие методы учета оборудования, часто используемые в образовательных учреждениях, как правило, основаны на ручном ведении журналов, использовании электронных таблиц и других устаревших способах. Это приводит к ряду проблем, таких как неточность данных, сложность поиска необходимого оборудования, значительные временные затраты на учет и инвентаризацию, а также затрудненный контроль за состоянием и сроками эксплуатации оборудования. Система “Мой ПТК” призвана решить эти проблемы, предоставляя централизованную базу данных, в которой будет храниться вся информация об оборудовании: его наименование, инвентарный номер, серийный номер, технические характеристики, местоположение, состояние, дата ввода в эксплуатацию, история обслуживания и ремонтов. Это позволит значительно упростить процесс инвентаризации, обеспечив быстрый доступ к информации, минимизировать ошибки, связанные с ручным вводом данных, и повысить достоверность информации об оборудовании. Кроме того, система автоматизирует процессы, связанные с работой лаборантов. Ручной учет и контроль оборудования отнимают у лаборантов значительное количество времени, которое можно было бы использовать для выполнения более важных задач, таких как подготовка оборудования к занятиям, консультирование студентов и поддержание порядка в лабораториях. “Мой ПТК” автоматизирует рутинные операции, такие как ведение учета, формирование отчетов, отслеживание перемещений оборудования, что освободит лаборантов от рутинной работы, повысит эффективность их труда и позволит им сосредоточиться на обеспечении бесперебойной работы лабораторий и учебных кабинетов.

### 1.1.2 Технико-математическое описание задачи

Технико-математическое описание задачи заключается в следующих аспектах:

* для отображения кабинетов на плане этажей и оборудования на планах кабинетов необходимо использовать геометрические преобразования, а именно при масштабировании плана, а также при перемещении объектов по нему;
* при разработке клиентской части необходимо использовать язык гипертекстовой разметки HTML, каскадные таблицы стилей CSS, нативный JavaScript;
* при разработке серверной части будет использован фреймворк Python Django;
* обмен данными между клиентом и сервером обеспечить с помощью HTTP протокола;
* для структурированного представления данных при обмене между клиентом и сервером необходимо использовать формат JSON;
* для хранения информации о кабинетах, оборудовании и пользователях, а также любой другой информации использовать СУБД MySQL;
* при написании, отладке и тестирования программного кода использовать среду разработки Visual Studio Code;
* для отслеживания изменений в коде необходимо использовать систему контроля версий Git;
* для управления зависимостями проекта и установки библиотек и фреймворков допускается возможным использовать менеджер pip.

### 1.1.3 Характеристика бизнес-процессов

В информационной системе предусмотрены следующие бизнес-процессы:

* инвентаризация техники (Процесс периодической проверки наличия и состояния техники в учреждении). Участниками являются лаборанты и комендант;
* учет перемещения техники (Процесс отслеживания перемещения техники между кабинетами и подразделениями учреждения). Осуществляют этот процесс лаборанты, с подтверждением администрации;
* учет технического обслуживания и ремонта производится лаборантами под контролем администрации и начальства;
* списание техники производится администрацией и комендантом;
* планирование закупок техники осуществляется администрацией и комендантом.

### 1.1.4 Требования к программе

Требования к инструментальному программному обеспечению:

* требования к инструментальному ПО можно разбить на следующие составляющие:
* среда разработки (IDE). В качестве среды разработки необходимо использовать Visual Studio Code;
* в качестве системы контроля версий рекомендуется использовать Git;
* пакетный менеджер pip;
* для работы с базой данных предоставляется возможным использовать MySQL;
* При прототипировании необходимо использовать Figma.

Требования к функциональным характеристикам:

* отображение интерактивного плана;
* управление информацией о кабинетах;
* управление информацией об оборудовании;
* поиск оборудования;
* учет техники;
* управление пользователями.

Общие требования к функционалу программы:

* Удобство использования;
* Производительность;
* Безопасность;
* Надежность;
* Соблюдение стандартов кодирование;
* Соблюдение стандартов веб-разработки;
* Ведение журнала действий пользователей.

Требования к дизайну. Интерфейс должен быть интуитивно понятным и удобным в использовании для пользователей с различным уровнем компьютерной подготовки. Использовать современный и минималистичный дизайн. Цветовая палитра должна быть приятной для глаз и соответствовать корпоративному стилю НовГУ. Важно обеспечить четкость и читаемость информации. Использование единообразных элементов интерфейса (кнопки, поля ввода, заголовки). Навигация должна быть простой и понятной. Максимально использовать визуальные элементы для представления информации (например, интерактивный план учреждения). Интерфейс должен быть отзывчивым и адаптивным, обеспечивая корректное отображение на различных устройствах (компьютеры, планшеты, смартфоны) с разными разрешениями экрана. Предусмотреть возможность настройки интерфейса (например, выбор цветовой схемы, расположения элементов) для пользователей. Использование стандартов веб-дизайна (HTML, CSS, JavaScript) для обеспечения кроссбраузерности и соответствия современным требованиям. Дизайн информационной системы должен соответствовать тематике и дизайну, а также цветовой гамме системы НовГУ.

Требования к быстродействию. Система должна обеспечивать минимальное время отклика на действия пользователя (например, клик по кнопке, ввод данных) - не более 1-2 секунд. Не должно быть длительных задержек при выполнении операций. Также, программный продукт должен обеспечивать стабильную производительность даже при увеличении количества одновременных пользователей.

Требования к составу и параметрам технических средств. Любой современный компьютер или мобильное устройство, способное запускать современный веб-браузер (Chrome, Firefox, Safari, Edge). Также браузер должен поддерживать следующие технологии: HTML5, CSS3 и JavaScript.

Безопасная эксплуатация информационной системы «Мой ПТК» подразумевает соблюдение следующих требований:

1. Защита данных
   * доступ к данным только для авторизованных пользователей (лаборанты, администраторы, преподаватели;
   * шифрование паролей;
   * проверка вводимых данных (валидация форм).
2. Защита от угроз
   * Использование ORM Django для защиты от SQL инъекций;
   * Экранирование пользовательского ввода (шаблонизатор Django автоматически экранирует HTML/JS);
   * Токены в формах для защиты от CSRF атак.

## 1.2 Обоснование проектных решений

### 1.2.1 Обоснование выбора языков программирования

Выбор Django в качестве основного фреймворка для backend-разработки обусловлен его рядом преимуществ. Во-первых, Django имеет высокоуровневую архитектуру, что позволяет ускорить процесс разработки, упростить написание кода и сделать его более читаемым. Во-вторых, встроенный ORM (Object-Relational Mapper) Django значительно упрощает взаимодействие с базой данных, избавляя разработчика от необходимости писать сложные SQL-запросы. Это делает разработку более быстрой и менее подверженной ошибкам. В-третьих, Django предоставляет встроенные механизмы обеспечения безопасности, такие как защита от CSRF-атак, SQL-инъекций и других уязвимостей, что обеспечивает надежность разрабатываемого приложения.

Выбор HTML, CSS и JavaScript для frontend-разработки также обоснован. HTML обеспечивает структуру веб-страниц, CSS отвечает за их оформление, а JavaScript добавляет интерактивность и динамическое поведение. Вместе эти технологии позволяют создать современный и удобный пользовательский интерфейс. JavaScript, в частности, позволяет реализовать интерактивный план учреждения, обеспечивая удобную навигацию и отображение информации о размещении техники.

### 1.2.2 Инструментальные средства

Для разработки информационной системы “Мой ПТК” будет использоваться следующий инструментарий:

* Git: система контроля версий, которая позволит отслеживать изменения в коде, управлять разными версиями проекта, работать над проектом командой разработчиков и предотвращать потерю данных;
* Django Framework: фреймворк на языке Python, который будет использоваться для разработки backend-части приложения. Django обеспечивает структуру для создания веб-приложений, упрощает работу с базами данных и предоставляет инструменты для реализации различных функций, таких как обработка запросов, аутентификация пользователей и т.д.;
* HTML, CSS и JavaScript: эти технологии будут использоваться для создания frontend-части приложения, то есть пользовательского интерфейса. HTML отвечает за структуру веб-страниц, CSS - за их оформление, а JavaScript - за добавление интерактивности и динамического поведения;
* MySQL: система управления базами данных (СУБД), которая будет использоваться для хранения данных о технике, лаборантах и другой информации, необходимой для работы системы. MySQL предоставляет надежное хранилище данных и инструменты для эффективного управления ими;
* pip: менеджер пакетов Python, который будет использоваться для установки и управления зависимостями проекта, то есть сторонними библиотеками и модулями, необходимыми для работы Django и других компонентов системы.

### 1.2.3 Обоснование выбора среды программирования

Visual Studio Code выбран в качестве среды разработки благодаря своей легковесности, гибкости, широкой поддержке языков программирования, наличию мощных инструментов отладки и интеграции с Git, что обеспечивает эффективную разработку

### 1.2.4 Информационное обеспечение

Для реализации проекта “Мой ПТК” потребуется следующее информационное обеспечение. Git будет использоваться в качестве системы управления версиями, позволяющей отслеживать изменения в коде, обеспечивать совместную работу над проектом нескольких разработчиков и предотвращать потерю данных. В качестве системы управления базой данных (СУБД) будет использоваться Sqlite3. Sqlite3 является легкой, встроенной базой данных, которая хорошо подходит для небольших проектов и не требует установки отдельного сервера баз данных. Данные о технике, кабинетах, лаборантах и прочая информация, необходимая для работы системы, будут храниться в Sqlite3. Для создания документации, описания функциональности, требований и подготовки отчетов будут использоваться текстовые редакторы, такие как Word. Для подготовки презентационных материалов, демонстрации работы системы и визуализации данных планируется использовать приложения, например, PowerPoint.

## 1.3 Обзор и анализ существующих программных систем

### 1.3.1 Прямые аналоги

Системы, решающие схожие задачи учета оборудования и управления активами в образовательных учреждениях.

1. 1С:Колледж

* Преимущества: интеграция с бухгалтерией, поддержка законодательства РФ;
* Недостатки: сложность настройки, высокая стоимость лицензии.

1. АВЕРС:Колледж

* Преимущества: готовые отчетные формы, работа с инвентаризацией;
* Недостатки: устаревший интерфейс, ограниченная кастомизация.

### 1.3.2 Косвенные аналоги

1. GLPI

* возможности для адаптации: гибкий учет оборудования, поддержка кастомизации атрибутов;
* ограничения: нет специализации под образовательные учреждения.

1. Snipe-IT

* возможности для адаптации: простой интерфейс, API для интеграции, мобильное приложение;
* ограничения: нет учета работы лаборантов.

# 2 Практическая часть

## 2.1 Анализ задачи

На данном этапе будет произведен всесторонний анализ задачи разработки информационной системы “Мой ПТК”. Будут определены основные функциональные требования к системе, включая учет оборудования, контроль его местоположения, управление перемещениями, планирование технического обслуживания, ведение истории ремонтов и учет лаборантов. Будут проанализированы существующие процессы учета техники в Политехническом колледже ПТИ НовГУ, выявлены их сильные и слабые стороны, а также определены основные проблемы, которые система должна решить. Будут сформулированы функциональные и нефункциональные требования к системе, такие как производительность, масштабируемость, безопасность и удобство использования. Будет проведено детальное описание предметной области, включающее в себя структуру данных, взаимодействие пользователей с системой и разработку сценариев использования. В результате анализа будет сформировано четкое представление о задачах, которые должна решать система, и разработана спецификация, определяющая все аспекты разработки.

Проведено исследование текущего учета техники в Политехническом колледже ПТИ НовГУ (или определены требования к будущему учету, если учет отсутствует). Выявлены недостатки: ручной учет, отсутствие централизованной базы данных, сложности отслеживания перемещений и планирования обслуживания.

Определены функциональные требования: учет оборудования (название, инвентарный номер, характеристики, дата приобретения, стоимость, ответственный, состояние, фотографии), учет кабинетов/аудиторий, интерактивный план, управление перемещениями, техническое обслуживание (планирование, учет), ремонт, учет лаборантов, отчетность, поиск, безопасность.

Разработаны сценарии использования: добавление оборудования, перемещение оборудования, отметка ТО, формирование отчета, поиск оборудования.

Определены нефункциональные требования: производительность, удобство использования, масштабируемость, безопасность.

### 2.1.1 Информационное моделирование предметной области

Модель предметной области указана на рисунке.

### 2.1.2 Проектирование пользовательского интерфейса

Авторизация в системе происходит через страницу входа (рисунок А.7).

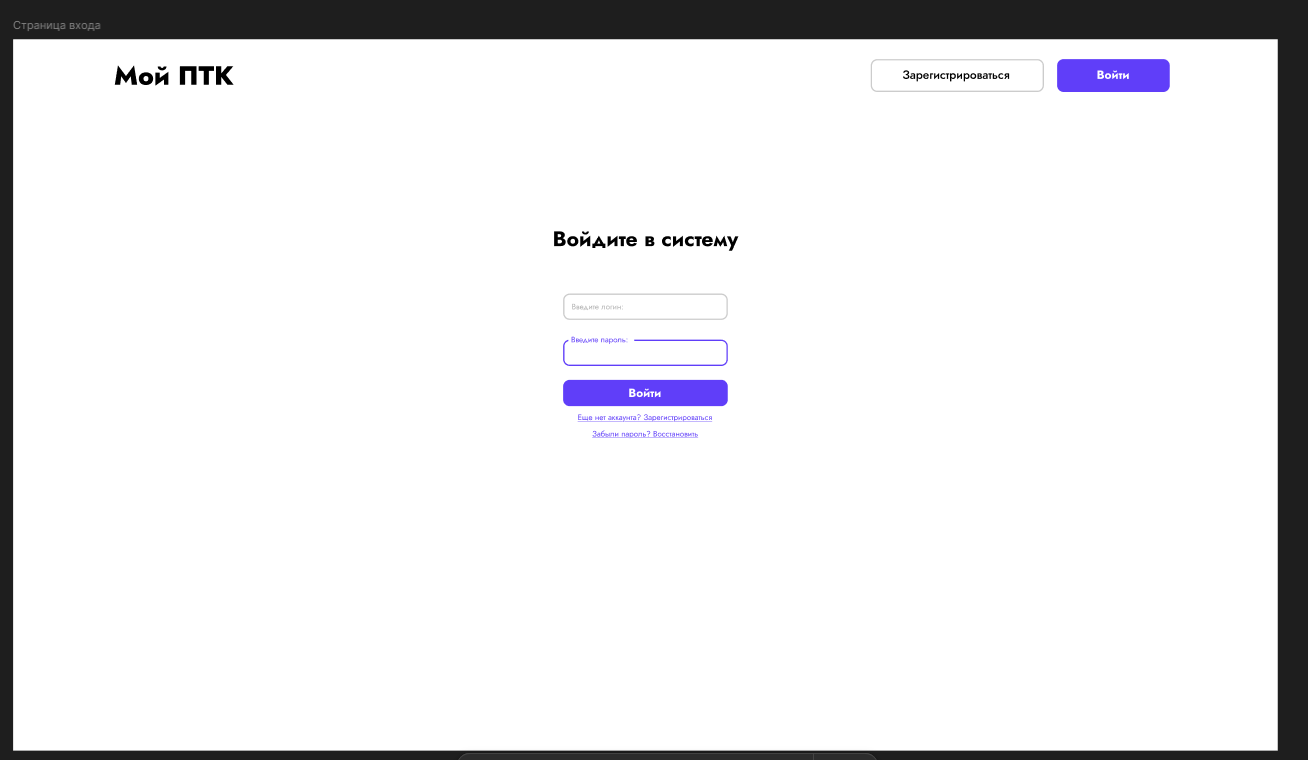


Рисунок 0‑1 - Макет входа в систему

Регистрация пользователя реализована через систему заявок, которые формируются при отправке формы и подтверждаются администратором системы (рисунок 2).

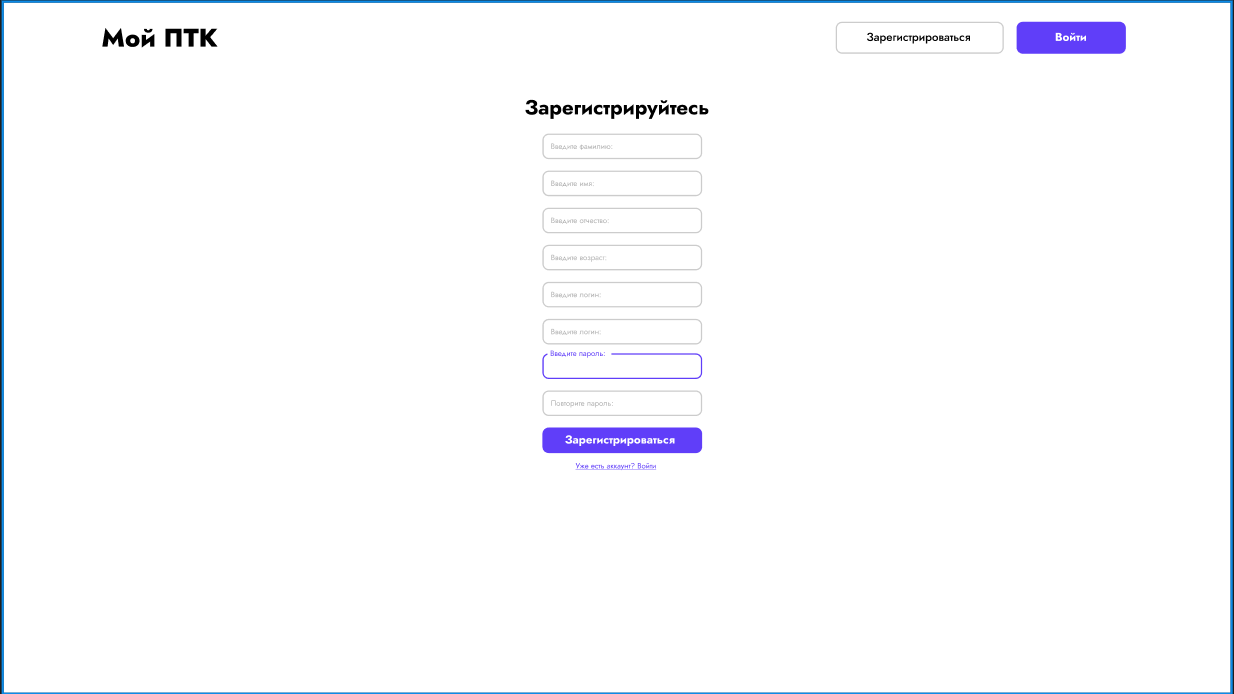


Рисунок 0‑2 - Макет страницы регистрации

После успешной авторизации пользователь перенаправляется на страницу с планами этажей колледжа (рисунок 3).



Рисунок 0‑3 - Макет главной страницы

При клике на кабинет с интерактивной карты происходит переадресация на страницу кабинета (рисунок 4).

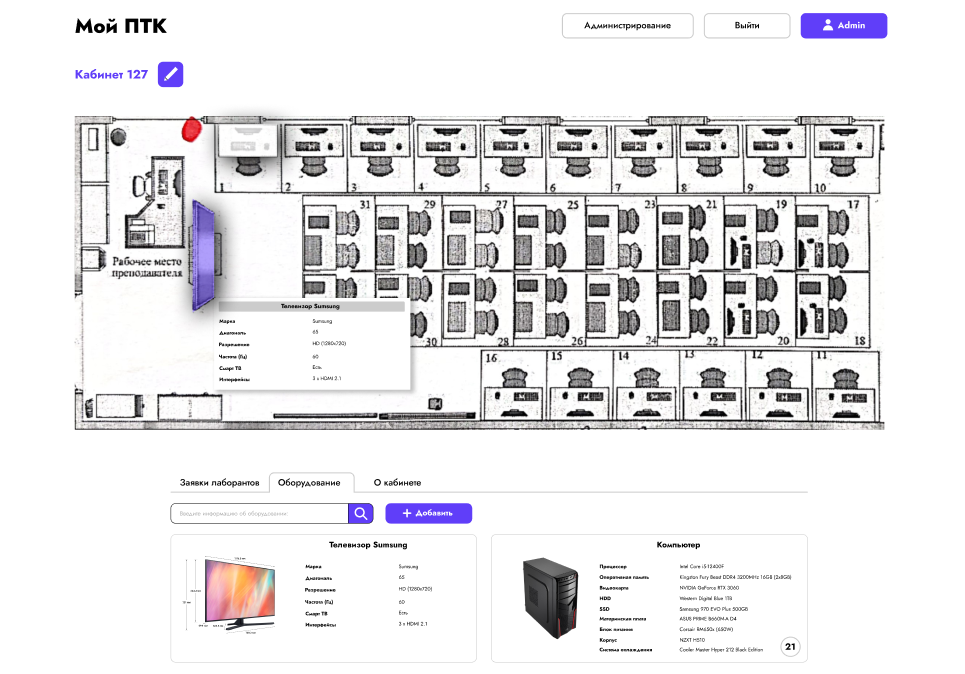


Рисунок 0‑4 - Макет страницы кабинета

Во вкладке оборудования можно добавить новое оборудование в кабинете при нажатии на соответствующую кнопку (рисунок 5).

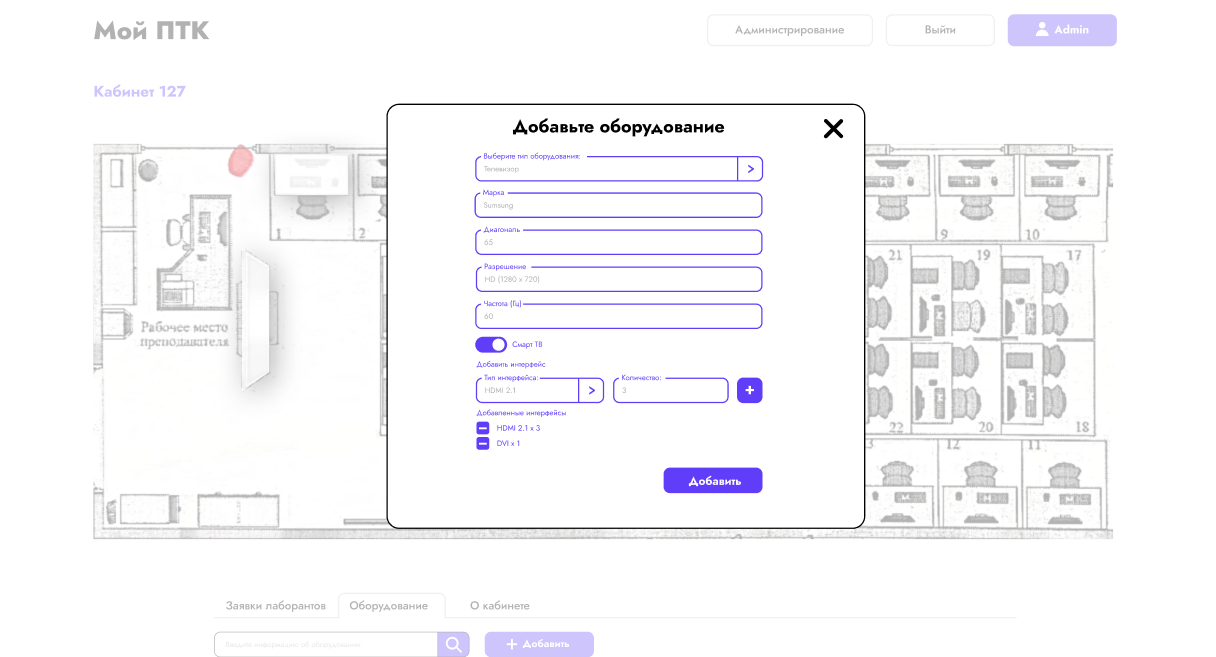


Рисунок 0‑5 - Макет формы добавления оборудования

Со страницы кабинета можно перейти на страницу оборудования при нажатии на его карточку (рисунок 6).

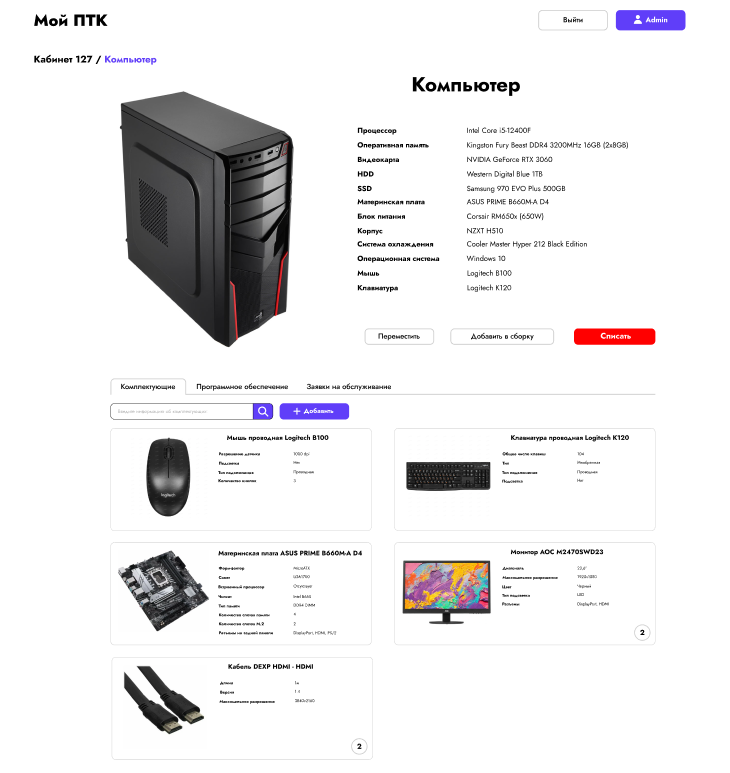


Рисунок 0‑6 - Макет страницы оборудования

## 2.2 Реализация

### 2.2.1 Практическая реализация алгоритмов

2.2.1.1 Описание ключевых алгоритмов

1. Алгоритм добавления нового оборудования (блок-схема описана на рисунке Д.1, листинги кода З.1.1 – 3.1.3). При добавлении нового оборудования система сначала получает данные от пользователя, включая тип оборудования, инвентарный номер и характеристики. Затем проводится проверка данных - проверяется уникальность инвентарного номера и соответствие характеристик схеме выбранного типа оборудования. После успешной проверки создается новая запись в базе данных с указанием местоположения и ответственного лица.
2. Алгоритм перемещения оборудования (блок-схема описана на рисунке Д.2, листинг кода З.2). При перемещении оборудования между кабинетами система запрашивает идентификатор оборудования, новое местоположение и причину перемещения. Проверяются права пользователя на выполнение этой операции. Затем обновляется информация о местоположении оборудования, создается запись в истории перемещений и отправляются уведомления заинтересованным лицам. Система поддерживает возможность отката операций и ведет полный журнал изменений;
3. Алгоритм авторизации (блок-схема описана на рисунке Д.3, листинг кода З.3). При входе в систему проверяются учетные данные пользователя и активность его учетной записи. Для каждого действия проверяются права доступа, включая права на конкретные объекты. Все действия пользователей логируются, ведется учет активных сессий. Система использует современные методы аутентификации, в том числе аутентификацию по сессии;
4. Алгоритм поиска оборудования (блок-схема описана на рисунке Д.4, листинг кода З.4). При выполнении поиска оборудования система принимает поисковый запрос от пользователя, который может содержать частичные совпадения по различным атрибутам оборудования, таким как наименование, модель, серийный номер или местоположение. Система применяет полнотекстовый поиск по текстовым полям и точное соответствие для других полей. Пользователь имеет возможность фильтровать результаты по дополнительным критериям, таким как состояние оборудования;
5. Алгоритм регистрации в системе (блок-схема описана на рисунке Д.5, листинг кода З.5). Алгоритм регистрации новых пользователей в системе начинается с получения регистрационных данных, включающих логин, пароль, адрес электронной почты и личные данные. Система выполняет валидацию данных на соответствие заданным требованиям, включая проверку сложности пароля и уникальность логина и электронной почты. После успешной валидации создается новая учетная запись пользователя, данные которой хешируются и сохраняются в базе данных. Пользователю отправляется письмо для подтверждения регистрации и активации учетной записи. После подтверждения учетная запись активируется, и пользователю предоставляется возможность войти в систему.

### 2.2.2 Используемые методы

2.2.2.1 Встроенные методы

Ниже приведены методы используемые в JavaScript.

1. Методы массивов
   * Array.map() - создает новый массив, преобразуя каждый элемент;
   * Array.filter() - возвращает новый массив с элементами, прошедшими проверку;
   * Array.find() - возвращает первый элемент, удовлетворяющий условию;
   * Array.forEach() - выполняет функцию для каждого элемента;
   * Array.includes() - проверяет наличие элемента в массиве.
2. Методы объектов
   * Object.keys() - возвращает массив ключей объекта;
   * Object.values() - возвращает массив значений объекта;
   * Object.entries() - преобразует объект в массив пар [ключ, значение].
3. Методы строк
   * String.trim() - удаляет пробелы с обоих концов строки;
   * String.includes() - проверяет наличие подстроки;
   * String.replace() - заменяет подстроку на новое значение;
   * String.split() - разбивает строку на массив по разделителю.
4. Методы работы с DOM
   * document.querySelector() - возвращает первый элемент, соответствующий CSS-селектору;
   * document.addEventListener() - добавляет обработчик события;
   * element.classList - управление CSS-классами элемента (add/remove/toggle);
   * element.setAttribute() - устанавливает значение атрибута элемента.
5. Методы асинхронной работы
   * fetch() - выполняет HTTP-запрос;
   * Promise.then() - обрабатывает успешное выполнение Promise;
   * Promise.catch() - обрабатывает ошибки Promise;
   * async/await - синтаксис для работы с асинхронным кодом.
6. Другие важные методы
   * JSON.parse() - преобразует JSON-строку в объект;
   * JSON.stringify() - преобразует объект в JSON-строку;
   * localStorage - работа с локальным хранилищем браузера.

Ниже приведены методы используемые в Python (Django)

1. Методы моделей (Models)
   * objects.all() - возвращает все объекты модели;
   * objects.filter() - фильтрация объектов по условиям;
   * objects.get() - получение одного объекта (вызывает исключение, если не найден);
   * save() - сохранение объекта в БД;
   * delete() - удаление объекта из БД;
   * \_\_str\_\_() - определяет строковое представление объекта.
2. Методы представлений (Views)
   * @login\_required - ограничивает доступ для авторизованных пользователей;
   * redirect() - перенаправление на другой URL;
   * render() – обрабатывает и возвращает обработанный шаблон.
3. Методы форм (Forms)
   * is\_valid() - проверяет валидность данных формы;
   * cleaned\_data - возвращает очищенные данные формы;
   * save() - сохраняет данные формы в модель;
   * ModelForm - автоматически создает форму на основе модели.
4. Методы шаблонов (Templates)
   * {% for %} - цикл по коллекции объектов;
   * {% if %} - условное отображение;
   * {{ variable }} - вывод переменной;
   * {% url %} - генерация URL по имени маршрута;
   * {% csrf\_token %} - защита от CSRF-атак.
5. Сессии и аутентификация
   * authenticate() - проверка учетных данных;
   * login()/logout() - управление сеансом пользователя.

2.2.2.2 Сторонние методы

1. JavaScript
   * collectFormData() – функция формирования тела запроса к серверу с формы;
   * createListField() – функция создания элемента поля списка;
   * getCookie() – функция возвращающая значение переменной, хранящейся в cookie;
   * updateMap() - процедура обновления положения интерактивной карты;
   * transformMap() – процедура перемещения интерактивной карты;
   * initSizes() – функция инициализации размеров карты.
   * toSlide() – процедура изменения текущего слайда изображений кабинета.
2. Python (Django)
   * index() – функция обработчик главной страницы;
   * register\_view() – функция обработчик страницы регистрации;
   * login\_view() – функция обработчик страницы авторизации;
   * logout\_view() – функция обработчик выхода из системы;
   * cabinet\_view() – функция обработчик страницы кабинета;
   * confirm\_user\_ajax() – функция для подтверждения заявки на регистрацию вызываемая с админки;
   * addEquipment() – функция обработчик страницы добавления нового оборудования.

### 2.2.3 Составные части программы и связи между ними

В проекте "Мой ПТК" реализована модульная архитектура (представлена на рисунке А.5), где каждый компонент выполняет строго определенные функции. Диаграмма взаимодействия модулей представлена на рисунке Г.2). Основные составные части системы и их взаимодействие можно описать следующим образом:

Центральным элементом системы является ядро на базе Django-фреймворка, которое координирует работу всех модулей. Ядро обрабатывает HTTP-запросы, управляет маршрутизацией и обеспечивает взаимодействие между компонентами.

Система включает несколько ключевых функциональных модулей. Модуль моделей данных определяет структуру хранения информации в базе данных через Django ORM. Основные модели включают Equipment (оборудование), EquipmentType (типы оборудования), User (пользователи) и Room (аудитории). Этот модуль отвечает за валидацию данных и реализацию бизнес-логики.

API-слой, построенный на Django REST Framework, предоставляет RESTful интерфейс для взаимодействия с фронтендом. Он включает сериализаторы для преобразования данных между форматами моделей и JSON, а также ViewSets для обработки CRUD-операций.

Административный интерфейс на базе Django Admin предоставляет полный CRUD-функционал для администраторов системы.

Взаимодействие между модулями происходит по следующей схеме. При поступлении запроса от пользователя через веб-интерфейс, фронтенд формирует HTTP-запрос к API. DRF-вьюсеты обрабатывают запрос, используя модели данных, которые через ORM преобразуют операции в SQL-запросы к MySQL. Обратный поток данных идет от БД через ORM и сериализаторы DRF к фронтенду, который обновляет пользовательский интерфейс.

Управление состоянием системы требует соблюдения определенных зависимостей. Фронтенд зависит от доступности API, а бэкенд - от стабильного подключения к MySQL. Изменение структуры моделей требует последовательного выполнения миграций, обновления сериализаторов DRF и корректировки документации API.

Архитектура проекта Djnago представлена на рисунке Г.1.

Архитектура системы построена на трех ключевых принципах. Принцип разделения ответственности обеспечивает, что каждый модуль решает строго определенный круг задач. Принцип слабой связанности позволяет модулям взаимодействовать через четкие интерфейсы, что упрощает замену компонентов. Принцип масштабируемости обеспечивает возможность расширения системы, например, через добавление новых типов оборудования с помощью JSON-схем.

Взаимодействие компонентов можно представить в виде текстовой схемы: пользовательский запрос через фронтенд поступает в Django-вьюсеты, которые через ORM обращаются к MySQL.

Такая архитектура обеспечивает гибкость при доработках, удобство тестирования отдельных компонентов, возможность поэтапного внедрения и стабильность работы благодаря изолированности модулей. Каждый компонент имеет четко определенный API для взаимодействия, что соответствует современным стандартам разработки корпоративных решений.

### 2.2.4 Реализация интерфейса

Для информационной системы "Мой ПТК" была разработана интуитивно понятная и современная пользовательская среда с соблюдением принципов UX/UI-дизайна (рисунок Ж.1 – Ж.7).

Цветовая схема интерфейса:

* + основной акцентный цвет: #603EF9 (ярко-синий);
  + фон: #FFFFFF (белый);
  + текст: #000000 (черный);
  + второстепенные элементы: #E9E9E9 (светло-серый);
  + ошибки/предупреждения: #FF0000 (красный);
  + успешные действия: #00FF26 (зеленый).

Типографика:

* + основной шрифт: Roboto (16px);
  + заголовки: Roboto Medium (24px);
  + мелкий текст: Roboto (14px).

Элементы интерфейса:

1. Кнопки
   * основные действия: #603ef9b6 с белым текстом, скругление 10px;
   * второстепенные: белый фон с обводкой #E9E9E9;
   * при наведении: основной цвет меняется на #603EF9.
2. Формы ввода
   * поля ввода: белый фон, серая обводка (#E9E9E9), скругление 10px;
   * активное поле: обводка #603EF9;
   * выпадающие списки: стиль, согласованный с полями ввода.

## 2.3 Руководство оператора

### 2.3.1 Назначение программы

Программа «Мой ПТК» предназначена для автоматизации учета оборудования и контроля работы лаборантов в профессионально-технических колледжах.

Основные функции:

* + ведение базы данных оборудования (добавление, редактирование, списание);
  + фиксация перемещения между кабинетами;
  + назначение ответственных лиц (лаборантов);
  + контроль рабочего времени лаборантов.

### 2.3.2 Условия выполнения

Технические требования:

* + ОС: Windows 10/11, Linux (Ubuntu 20.04+), macOS 12+;
  + браузер: Chrome, Edge, Firefox (последние версии);
  + доступ в интернет.

Программные требования:

* + для серверной части: Python 3.10+, Django 4.2+.

Права доступа:

* + администратор: полный доступ;
  + лаборант: редактирование закрепленного оборудования;
  + преподаватель: редактирование закрепленного оборудования.

UML диаграмма бизнес прецедентов отображена на рисунке Б.1.

### 2.3.3 Выполнение программы

При локальном запуске необходимо сначала запустить сервер с проектом:

* + зайти в директорию проекта;
  + открыть командную строку;
  + запустить команду python manage.py runserver;
  + перейти по указанному адресу и войти в свою учетную запись или подать заявку на регистрацию;
  + после успешной авторизации в системе пользователю будет доступен определенный его ролью функционал.

### 2.3.4 Работа пользователя

Для администратора:

* + регистрация новых пользователей (лаборантов);
  + настройка типов оборудования и их атрибутов.

Для лаборанта:

* + отметка о проведенном ТО оборудования;
  + выполнение заявок на ремонт;
  + редактирование закрепленного оборудования;
  + просмотр закрепленного оборудования.

Для преподавателя:

* + подтверждение выполнения заявки на ремонт или наладку оборудования;
  + просмотр и редактирование закрепленного оборудования;
  + формирование заявок на ремонт оборудования.

### 2.4 Тестирование

Тест кейсы работы программного продукта отображены в таблицах Е.1 – Е.2.

# Заключение

В результате проведенной работы была разработана информационная система “Мой ПТК”, предназначенная для автоматизации учета техники в Политехническом колледже ПТИ НовГУ. В ходе реализации были выполнены все поставленные задачи: проведен анализ предметной области, определены функциональные и нефункциональные требования, разработана архитектура системы, выбран оптимальный набор программных и технических средств, реализованы основные алгоритмы и интерфейсы.

Система обеспечивает эффективное управление техническими ресурсами, позволяя вести централизованный учет оборудования, отслеживать его местоположение, планировать техническое обслуживание и ремонт, а также формировать различные отчеты. Интерактивный план обеспечивает наглядное представление данных о размещении оборудования, упрощая его поиск и инвентаризацию. Разработанный интерфейс, учитывающий принципы удобства использования, делает работу с системой интуитивно понятной и комфортной для пользователей.

Таким образом, информационная система “Мой ПТК” является эффективным и практичным решением для оптимизации управления техническими ресурсами в Политехническом колледже ПТИ НовГУ, соответствующим всем поставленным требованиям и готовым к внедрению и использованию.

# Список литературы

1. ГОСТ 2.114–95 Единая система конструкторской документации. Технические условия – Госстандарт России – 17 с.;
2. ГОСТ 19.201–78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению – Государственный комитет по стандартизации Совета Министров СССР – 12 с.
3. ГОСТ 34.602–89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы – Министерство связи СССР – 13 с.
4. ГОСТ 19.102–77 «Единая система программной документации: общие требования к программным документам" – Государственный комитет по стандартизации СССР – 6 с.
5. ГОСТ 34.601–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Языки программирования" – Государственный комитет по стандартизации Совета Министров СССР – 14 с.
6. ГОСТ 2.105 – 79 Единая система конструкторской документации. Общие требования к текстовым документам. – Государственный комитет СССР по стандартам, 1980. – 25 с., 18.04.2024.
7. ГОСТ 2.701 – 2008 Единая система конструкторской документации. Схемы. Виды и типы. Общие требования к выполнению. – М.: Стандартинформ 2009. – 16 с.,18.04.2024.
8. ГОСТ 2.743 – 91 Единая система конструкторской документации. Обозначения условные графические в схемах. Элементы цифровой техники. –ИПК Издательство стандартов, 1993. – 45 с., 18.04.2024.
9. Официальная документация Django: (<https://www.djangoproject.com/>): Основной источник информации по фреймворку Django, его функциям, возможностям, API и лучшим практикам разработки.
10. Официальная документация MySQL: (<https://dev.mysql.com/doc/>) Руководства и справочники по системам управления базами данных, используемым для хранения данных.
11. Основы HTML, CSS и JavaScript: (Различные учебные материалы, онлайн-курсы, справочники): Общие ресурсы, посвященные основам веб-разработки.
12. Официальная документация Django REST Framework: (<https://www.django-rest-framework.org/>): Подробное руководство по использованию Django REST Framework для создания RESTful API.

# Глоссарий

Оборудование – материальный объект (компьютер, проектор, прибор), состоящий на учете в системе, имеющий инвентарный номер и технические характеристики.

Инвентарный номер – уникальный цифровой или буквенно-цифровой идентификатор, присваиваемый каждому объекту оборудования для однозначной идентификации.

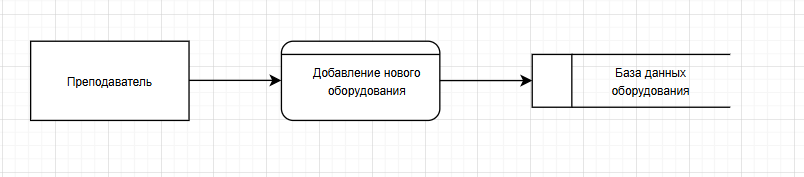
Лаборант – сотрудник образовательного учреждения, закрепленный за оборудованием и отвечающий за его сохранность и работоспособность.

Атрибуты оборудования – набор технических характеристик (например, для компьютера: процессор, объем ОЗУ, тип накопителя), хранящихся в структурированном виде (JSON).

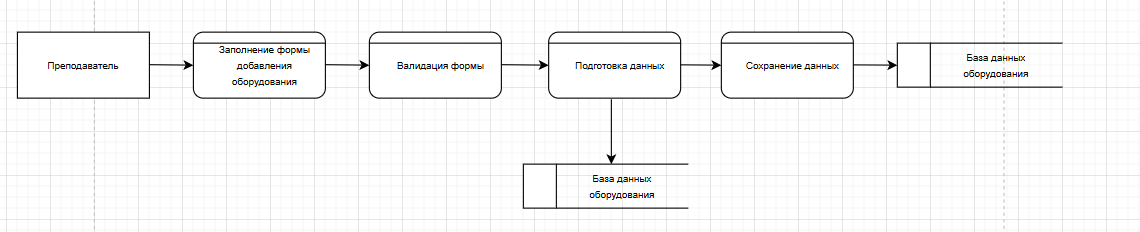
Аудит изменений – журнал всех операций с оборудованием (перемещение, ремонт, списание) с указанием пользователя, даты и измененных параметров.

# Приложение А

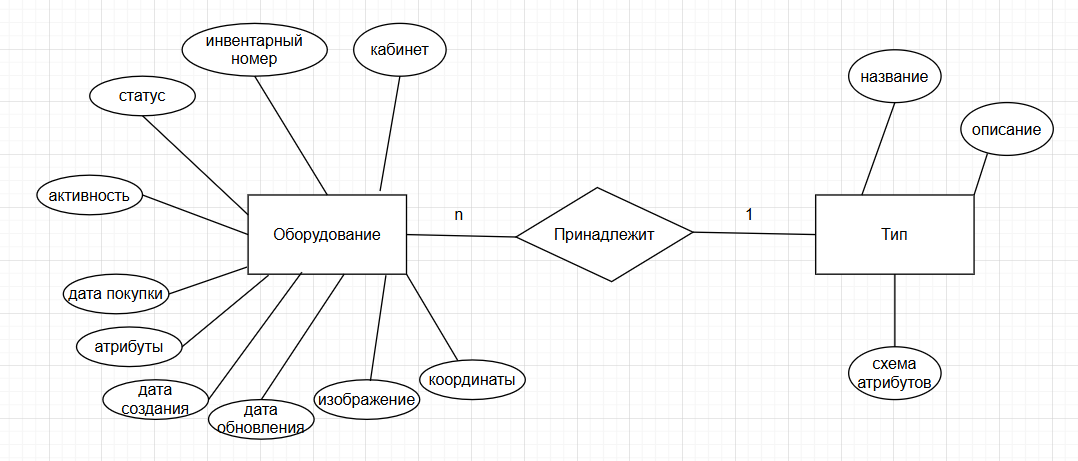
(обязательное)



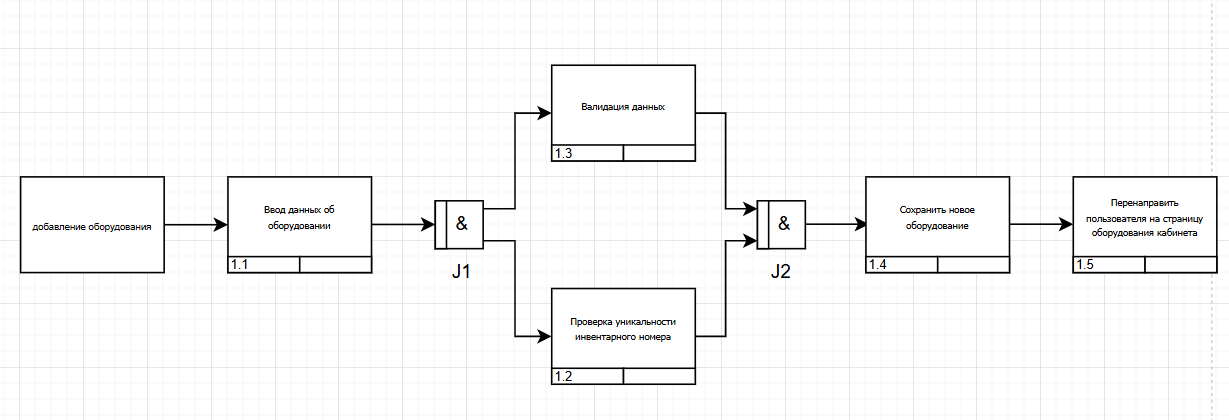
##### Рисунок А.1 – DFD модель создания нового оборудования



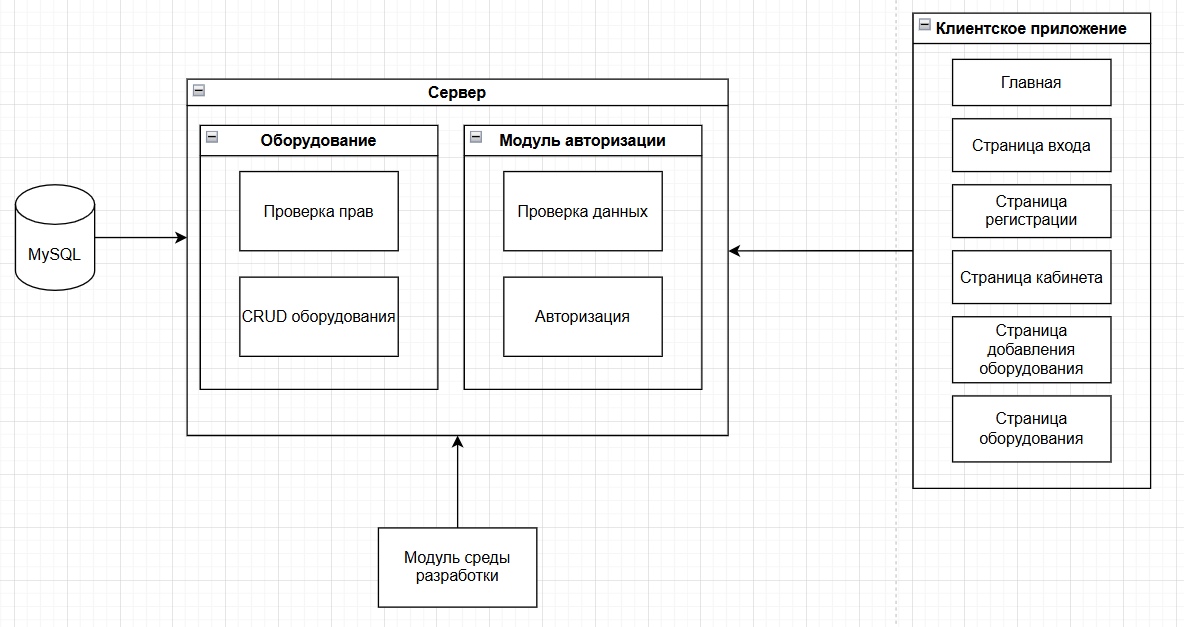
##### Рисунок А.2 – Декомпозиция DFD модели создания нового оборудования



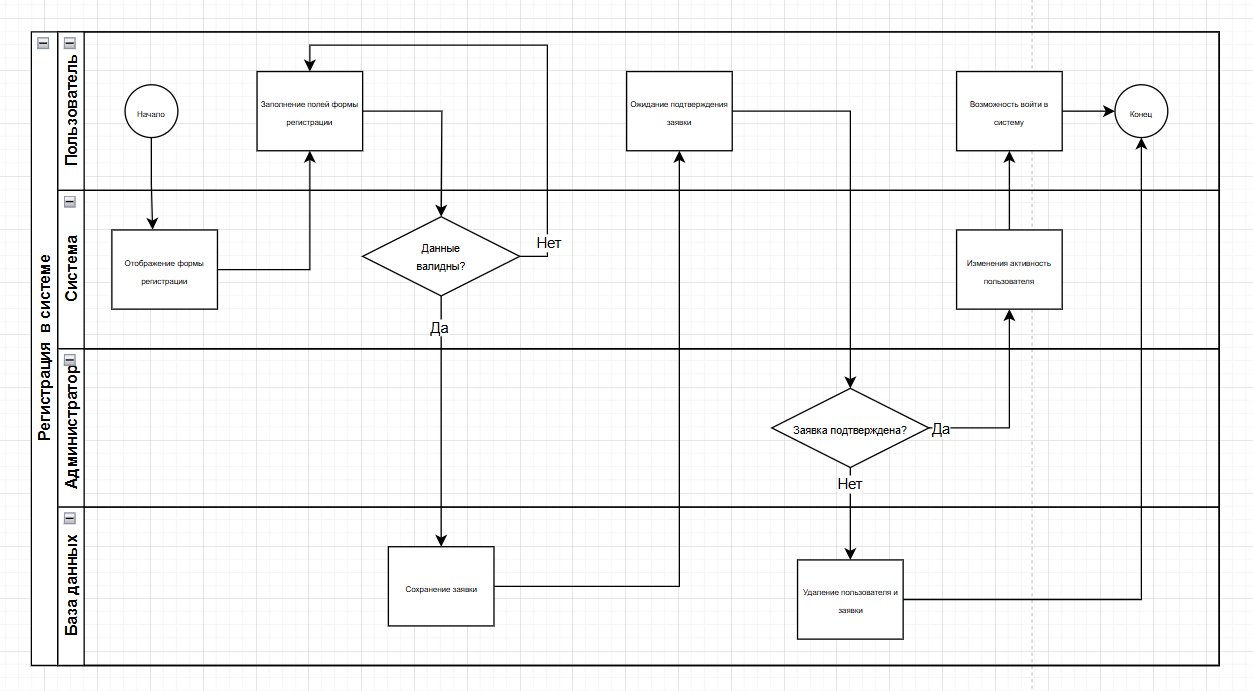
##### Рисунок А.3 – ER модель оборудование - тип



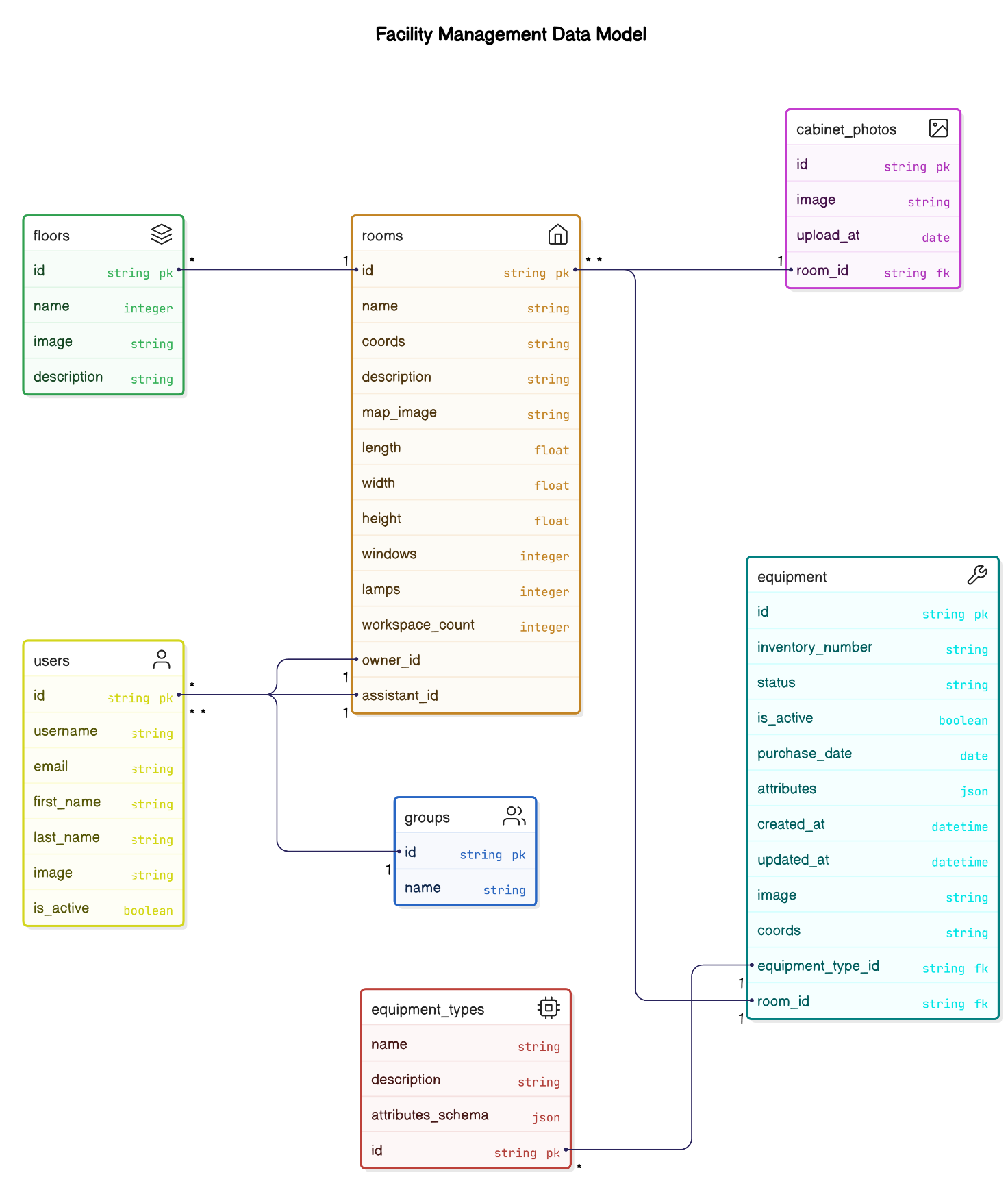
##### Рисунок А.4 – IDEF3 модель создания оборудования



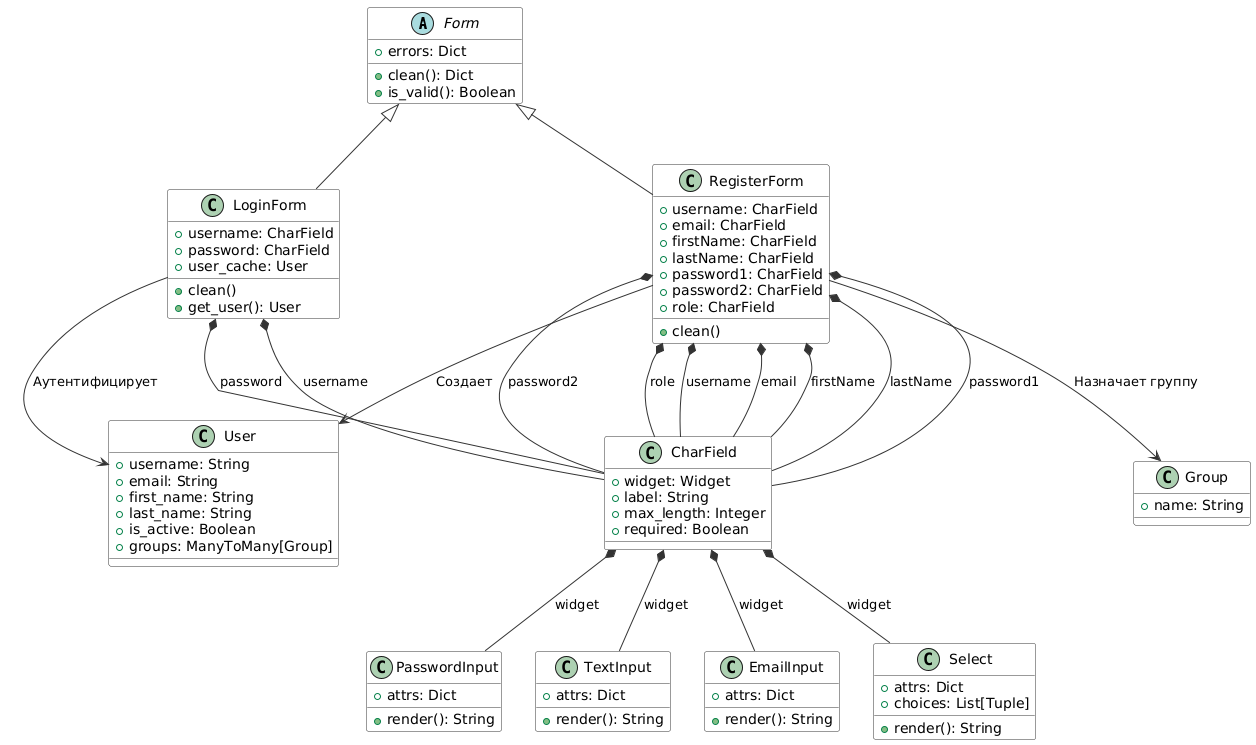
##### Рисунок А.5 – Диаграмма архитектуры программного комплекса



##### Рисунок А.6 – Диаграмма плавательные дорожки для регистрации пользователя



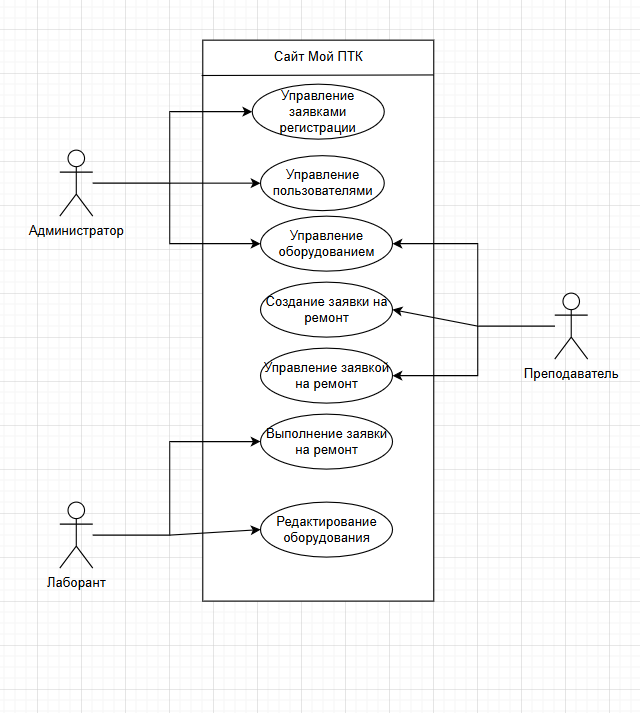
##### Рисунок А.7 – IDEF1x диаграмма функциональности сайта



##### Рисунок А.8 – UML модель классов

# Приложение Б

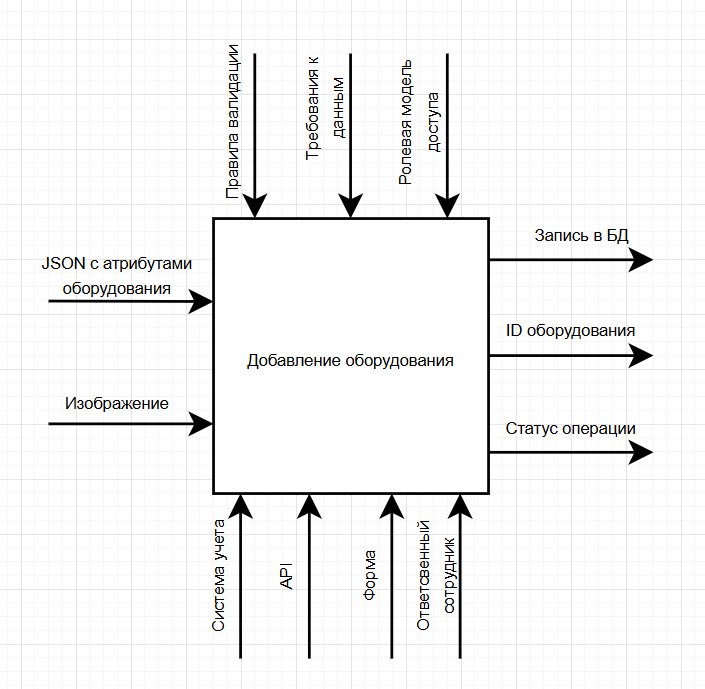
(обязательное)



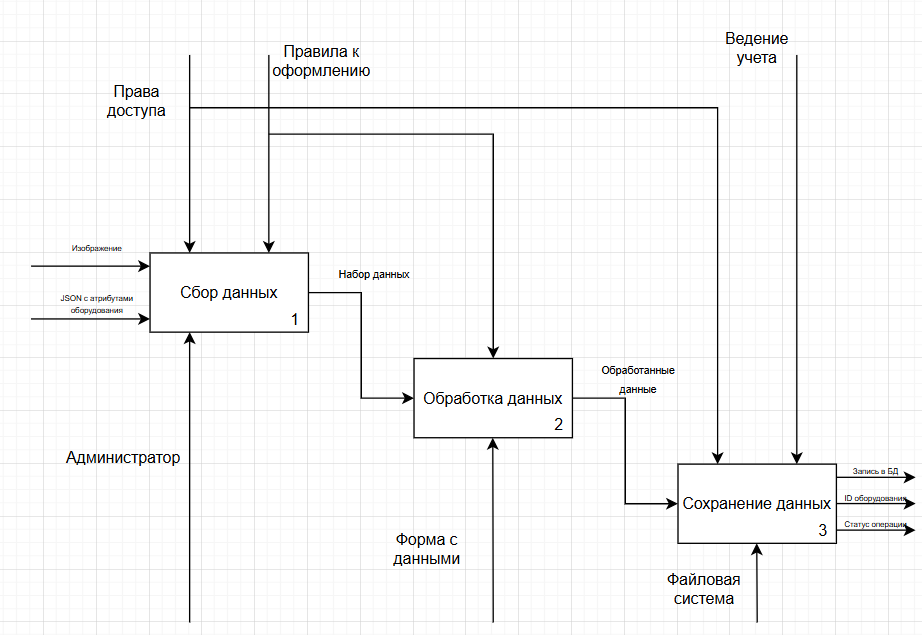
##### Рисунок Б.1 – UML модель бизнес-прецедентов

# Приложение В

(обязательное)



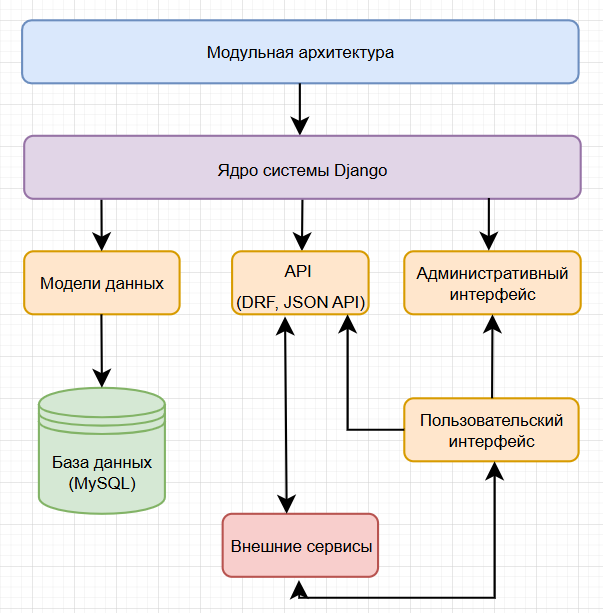
##### Рисунок В.1 - IDEF0 модель функционала сайта



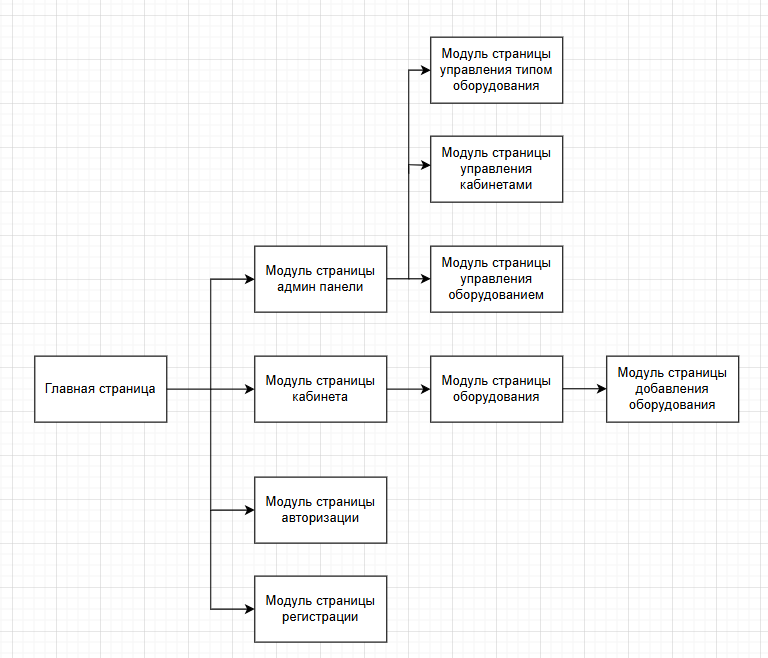
##### Рисунок В.2 - Декомпозиция IDEF0 модели

# Приложение Г

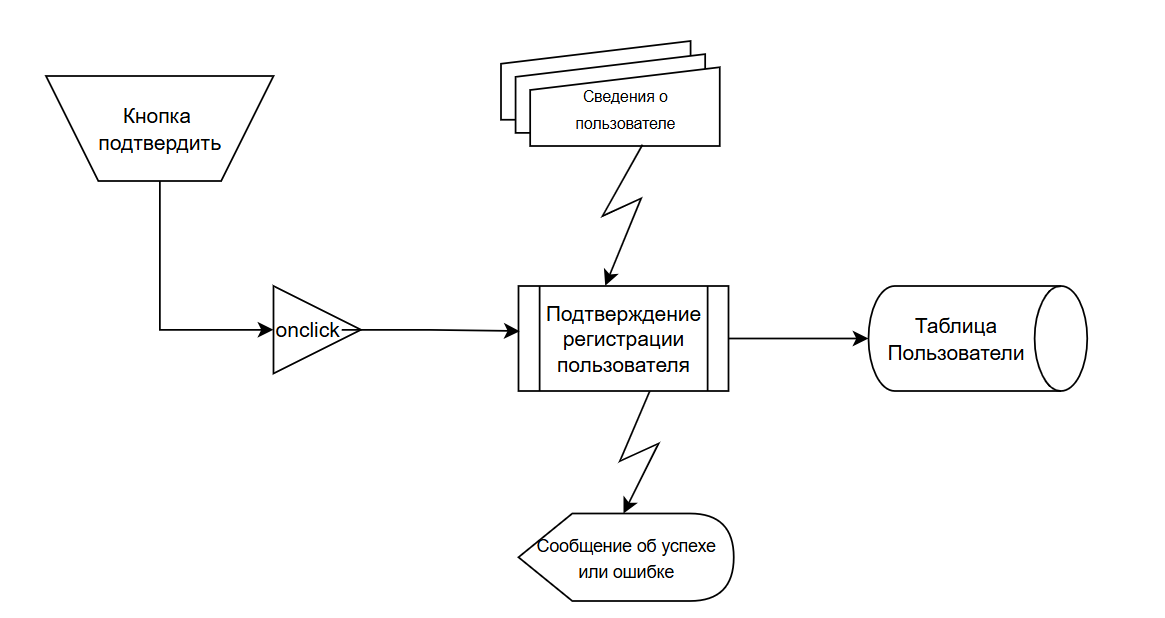
(обязательное)



##### Рисунок Г.1 – Архитектура проекта Django



##### Рисунок Г.2 – Диаграмма взаимодействия программных модулей



##### Рисунок Г.3 – Диаграмма внутри модуля подтверждения регистрации

##### Рисунок Г.4 – Диаграмма развертывания системы

# Приложение Д

(обязательное)

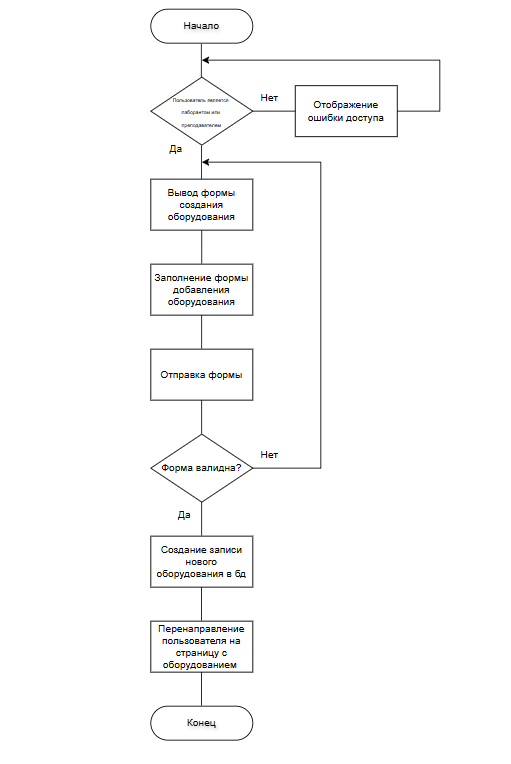


Рисунок Д.1 - Алгоритм добавления нового оборудования

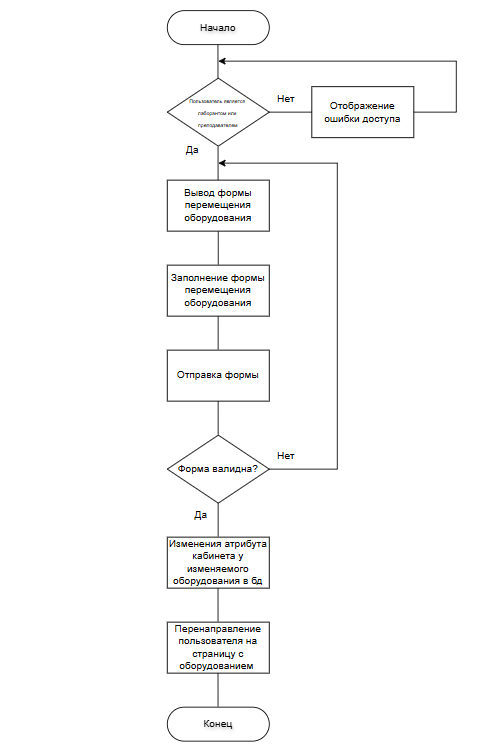


Рисунок Д.2 - Алгоритм перемещения оборудования

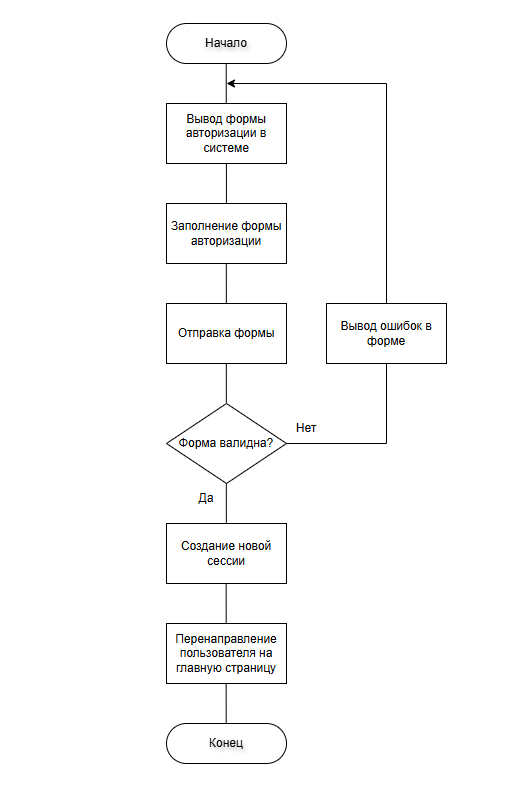


Рисунок Д.3 – Авторизация в системе

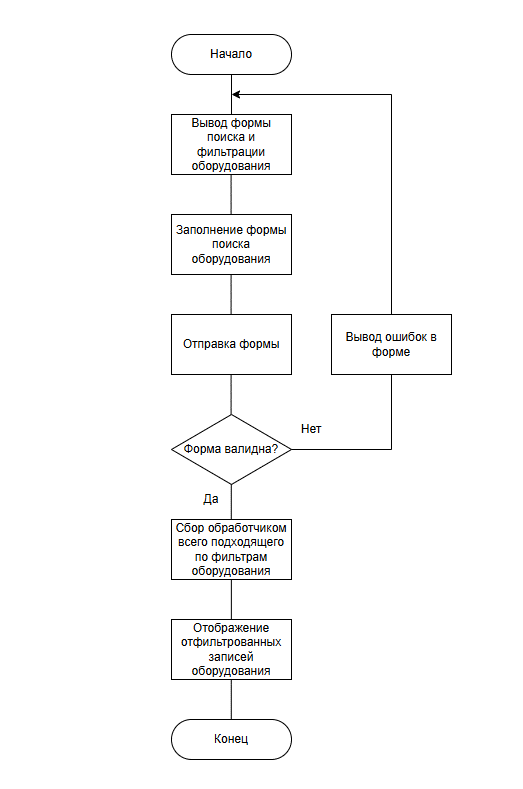


Рисунок Д.4 – Фильтрация и поиск оборудования

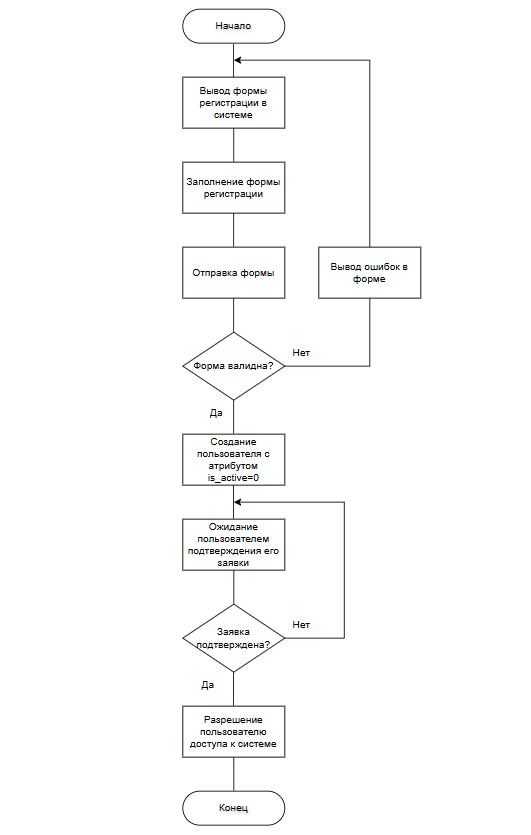


Рисунок Д.5 –Регистрация в системе

# Приложение Е

(обязательное)

Таблица Е.1 – Тест-кейс TC001

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC001 |
| Заголовок | | | Проверка добавления оборудования с валидными данными |
| Предусловие | | | Пользователь авторизован и имеет права на добавление оборудования. |
| Тестовые данные | | | Тип: “Компьютер”, Инв. номер: “INV-2024-001”, Наименование: “Компьютер Lenovo ThinkCentre”, Модель: “M75q”, Серийный номер: “1234567890”, Кабинет: “201” |
| Ожидаемый результат | | | Успешное добавление оборудования в систему |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти во вкладку кабинета или этажа “Оборудование” | Программа отображает содержимое вкладки “ Оборудование” | pass |
| 2 | Нажать кнопку “Добавить оборудование” | Программа перенаправляет пользователя на страницу добавления оборудования | pass |
| 3 | Заполнить все обязательные поля формы валидными данными | Отсутствие ошибок валидации на клиенте | pass |
| 4 | Нажать кнопку “Сохранить” | Отправка запроса на сервер | pass |
| Постусловие | | |  |
| Результат | | | Успешное создание новой записи оборудования |

Таблица Е.2 – Тест-кейс TC002

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC002 |
| Заголовок | | | Попытка добавления оборудования с дублирующимся инвентарным номером |
| Предусловие | | | Пользователь аутентифицирован и имеет права на добавление оборудования. В системе уже существует оборудование с инвентарным номером “INV-2024-001”. |
| Тестовые данные | | | Тип: “Монитор”, Инв. номер: “INV-2024-001”, Наименование: “Монитор Samsung”, Модель: “LS24R350”, Серийный номер: “ABCDEFGHIJ”, Кабинет: “202” |
| Ожидаемый результат | | | Система не позволяет добавить оборудование с дублирующимся инвентарным номером. |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти во вкладку кабинета или этажа “Оборудование” | Программа отображает содержимое вкладки “ Оборудование” | pass |
| 2 | Нажать кнопку “Добавить оборудование” | Программа перенаправляет пользователя на страницу добавления оборудования | pass |
| 3 | Заполнить все обязательные поля формы, указав существующий инвентарный номер | Отсутствие ошибок валидации на клиенте | pass |
| 4 | Нажать кнопку “Сохранить” | Отправка запроса на сервер | pass |
| Постусловие | | |  |
| Результат | | | Вывод ошибки о существовании такого оборудования |

##### Таблица Е.3 – Тест кейс TC003

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC003 |
| Заголовок | | | Попытка добавления оборудования с пустыми обязательными полями |
| Предусловие | | | Пользователь авторизован и имеет права на добавление оборудования. |
| Тестовые данные | | | Тип: “”, Инв. номер: “”, Наименование: “”, Модель: “”, Серийный номер: “”, Кабинет: “”, Ответственный: “” |
| Ожидаемый результат | | | Система не позволяет сохранить оборудование с пустыми обязательными полями. |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти во вкладку кабинета или этажа “Оборудование” | Программа отображает содержимое вкладки “ Оборудование” | pass |
| 2 | Нажать кнопку “Добавить оборудование” | Программа перенаправляет пользователя на страницу добавления оборудования | pass |
| 3 | Заполнить все обязательные поля формы валидными данными | Отсутствие ошибок валидации на клиенте | pass |
| 4 | Нажать кнопку “Сохранить” | Отправка запроса на сервер | pass |
| Постусловие | | |  |
| Результат | | | Отклонение создания новой записи оборудования |

##### Таблица Е.4 – Тест кейс ТС004

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC004 |
| Заголовок | | | Успешная подача заявки на регистрацию |
| Предусловие | | | Пользователь не зарегистрирован в системе. Страница регистрации доступна. |
| Тестовые данные | | | Логин: “newuser”, Пароль: “StrongPassword123”, Email: “newuser@example.com”, ФИО: “Новый Пользователь” |
| Ожидаемый результат | | | Пользователь успешно отправил заявку на регистрацию, учетная запись создана, заявка ожидает подтверждения. |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти на страницу регистрации | Программа отображает содержимое страницы регистрации | pass |
| 2 | Заполнить все поля формы регистрации валидными данными. | Отсутствие ошибок валидации на клиенте | pass |
| 3 | Нажать кнопку “Зарегистрироваться”. | Отправка запроса на сервер | pass |
| Постусловие | | |  |
| Результат | | | Успешная отправка заявки на регистрацию в системе |

##### Таблица Е.5 – Тест кейс ТС005

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC005 |
| Заголовок | | | Попытка регистрации с существующим email |
| Предусловие | | | Пользователь с email “user@example.com” уже зарегистрирован в системе. Страница регистрации доступна. |
| Тестовые данные | | | Логин: “anotheruser”, Пароль: “AnotherStrongPassword”, Email: “user@example.com”, ФИО: “Другой Пользователь” |
| Ожидаемый результат | | | Система не позволяет зарегистрироваться с email, который уже используется. Отображается сообщение об ошибке. |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти на страницу регистрации | Программа отображает содержимое страницы регистрации | pass |
| 2 | Заполнить все поля формы регистрации валидными данными. | Отсутствие ошибок валидации на клиенте | pass |
| 3 | Нажать кнопку “Зарегистрироваться”. | Отправка запроса на сервер | pass |
| Постусловие | | | Email “user@example.com” по-прежнему связан с существующей учетной записью. |
| Результат | | | Отклонение регистрации в системе |

##### Таблица Е.6 – Тест кейс ТС006

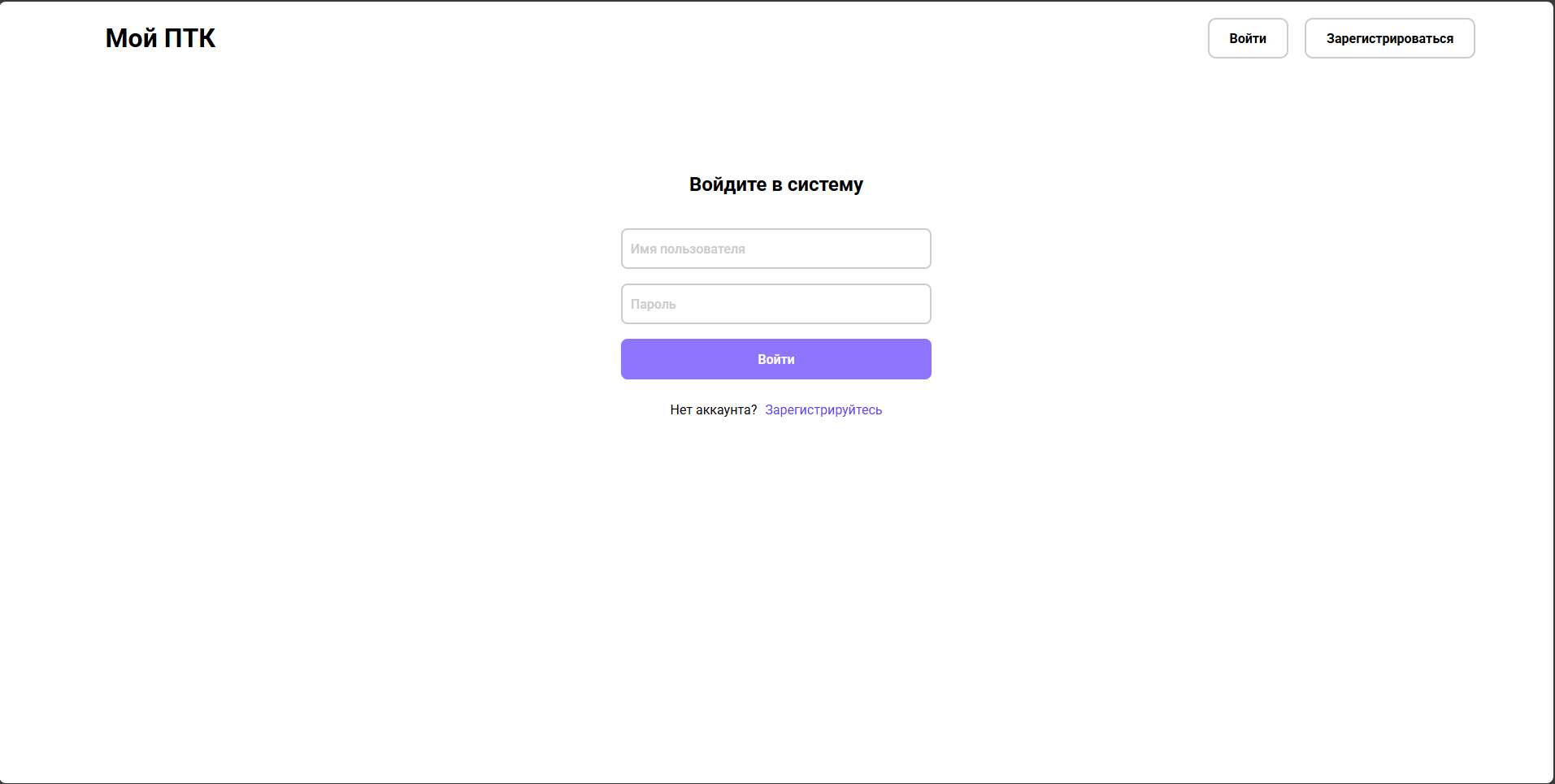
|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC006 |
| Заголовок | | | Успешная авторизация с правильными учетными данными |
| Предусловие | | | Пользователь с логином “user” и паролем “CorrectPassword” зарегистрирован и активирован в системе. Страница авторизации доступна. |
| Тестовые данные | | | Логин: “user”, Пароль: “CorrectPassword” |
| Ожидаемый результат | | | Пользователь успешно авторизован и перенаправлен на главную страницу системы. |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти на страницу авторизации | Программа отображает содержимое страницы авторизации | pass |
| 2 | Ввести правильный логин и пароль. | Отсутствие ошибок валидации на клиенте | pass |
| 3 | Нажать кнопку “Войти”. | Отправка запроса на сервер | pass |
| Постусловие | | | Пользователь находится в авторизованном состоянии. |
| Результат | | | Успешная авторизация в системе |

##### Таблица Е.7 – Тест кейс ТС007

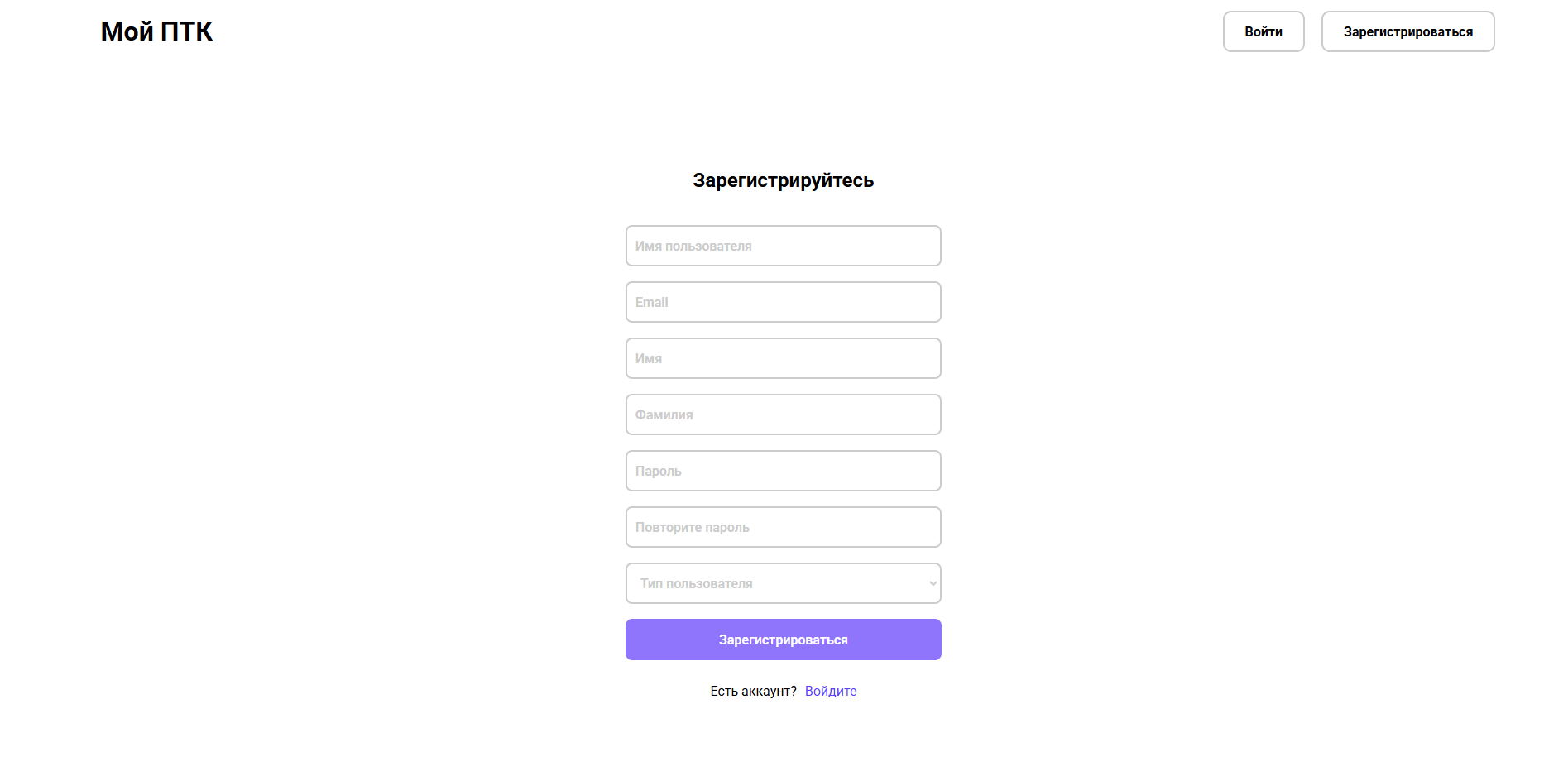
|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC007 |
| Заголовок | | | Попытка авторизации с неверным паролем |
| Предусловие | | | Пользователь с логином “user” зарегистрирован и активирован в системе. Страница авторизации доступна. |
| Тестовые данные | | | Логин: “user”, Пароль: “IncorrectPassword” |
| Ожидаемый результат | | | Система не позволяет авторизоваться с неверным паролем. Отображается сообщение об ошибке. |
| Шаг | Действие | Предполагаемый результат | Фактический результат (pass / fail) |
| 1 | Перейти на страницу авторизации | Программа отображает содержимое страницы авторизации | pass |
| 2 | Ввести правильный логин и неверный пароль. | Отсутствие ошибок валидации на клиенте | pass |
| 3 | Нажать кнопку “Войти”. | Отправка запроса на сервер | pass |
| Постусловие | | | Пользователь остается неавторизованным. |
| Результат | | | Отклонение авторизации в системе. |

# Приложение Ж

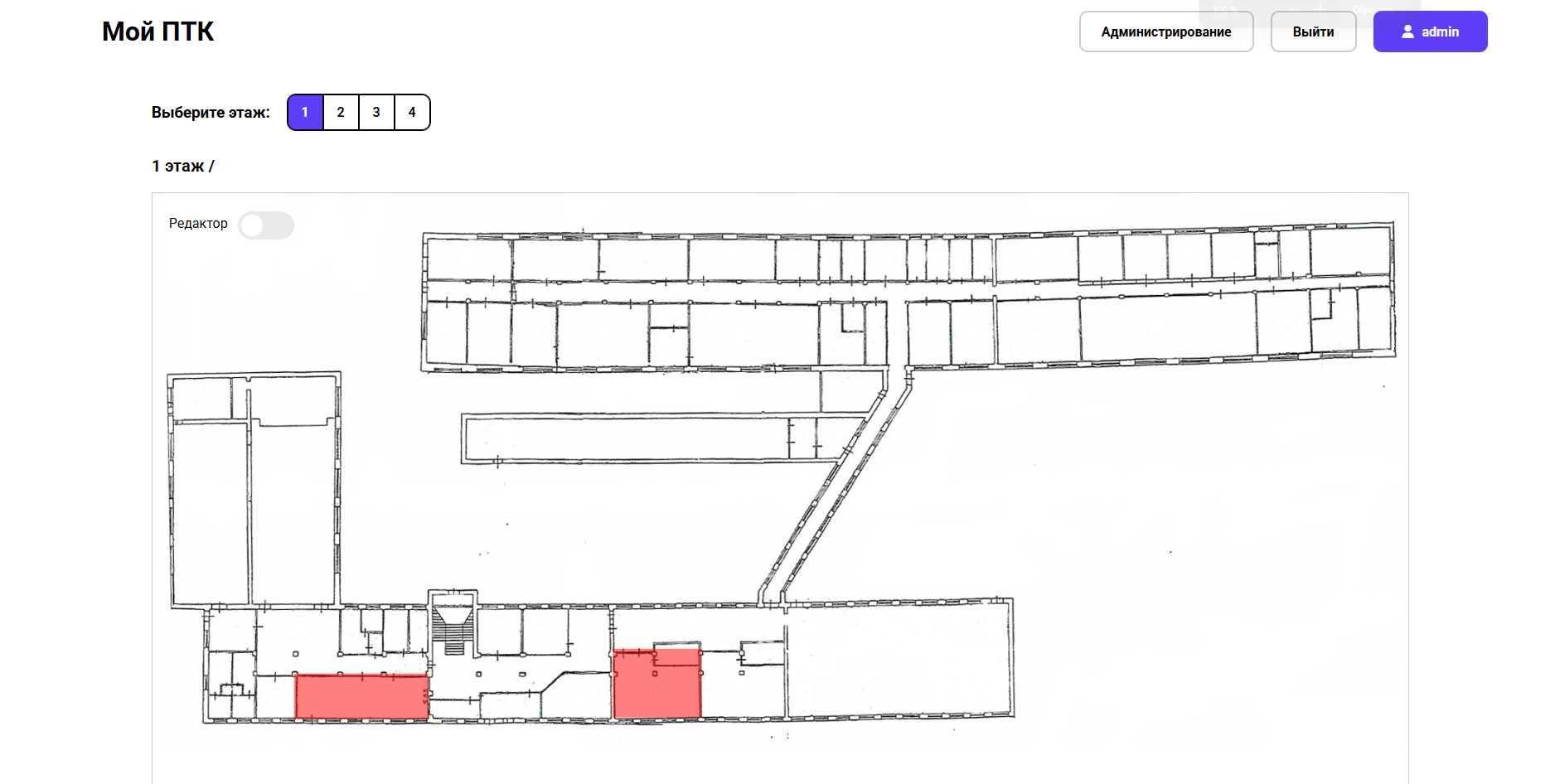
(обязательное)



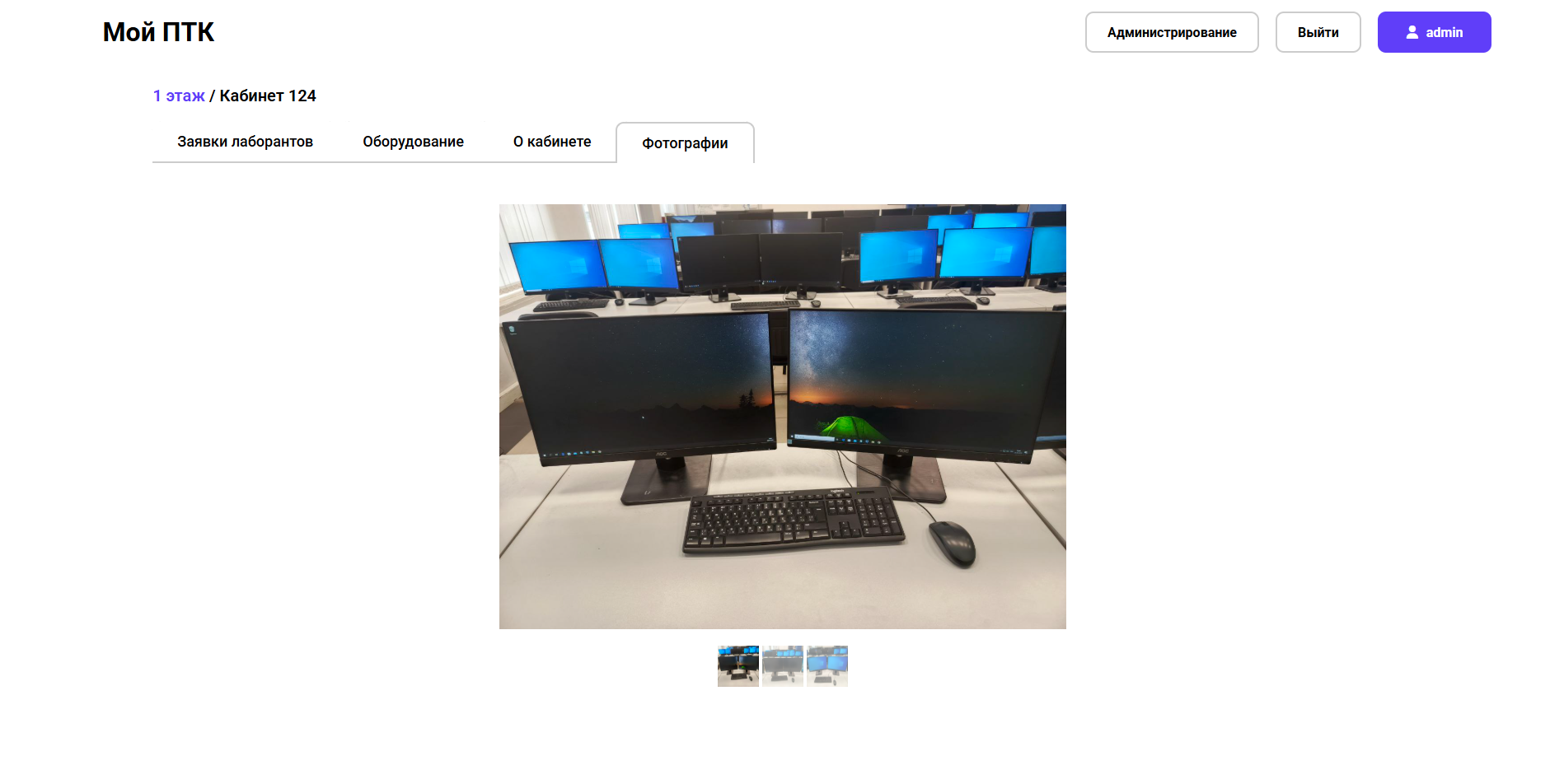
##### Рисунок Ж.1 – Страница входа в систему



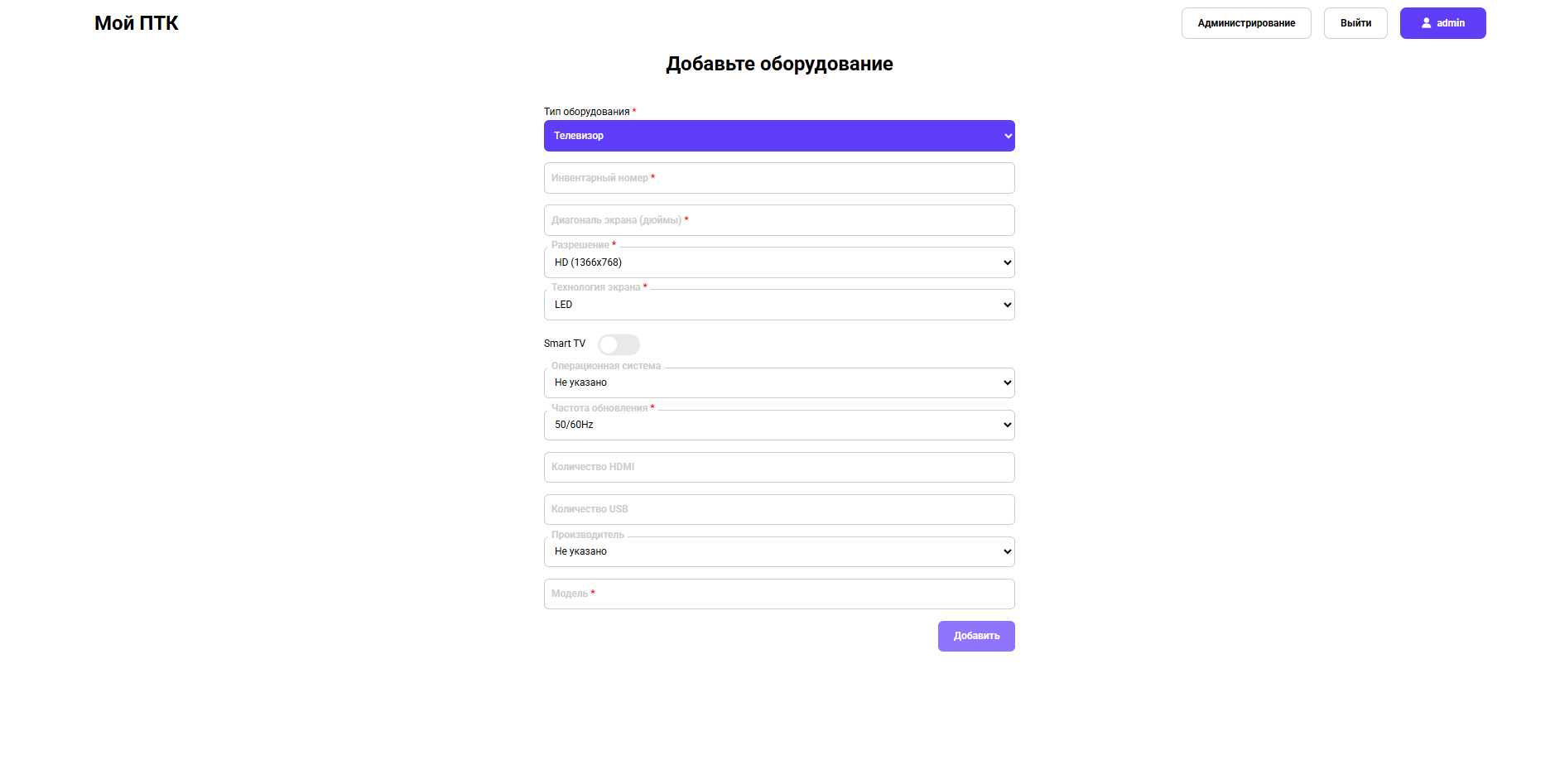
##### Рисунок Ж.2 – Страница регистрации в системе



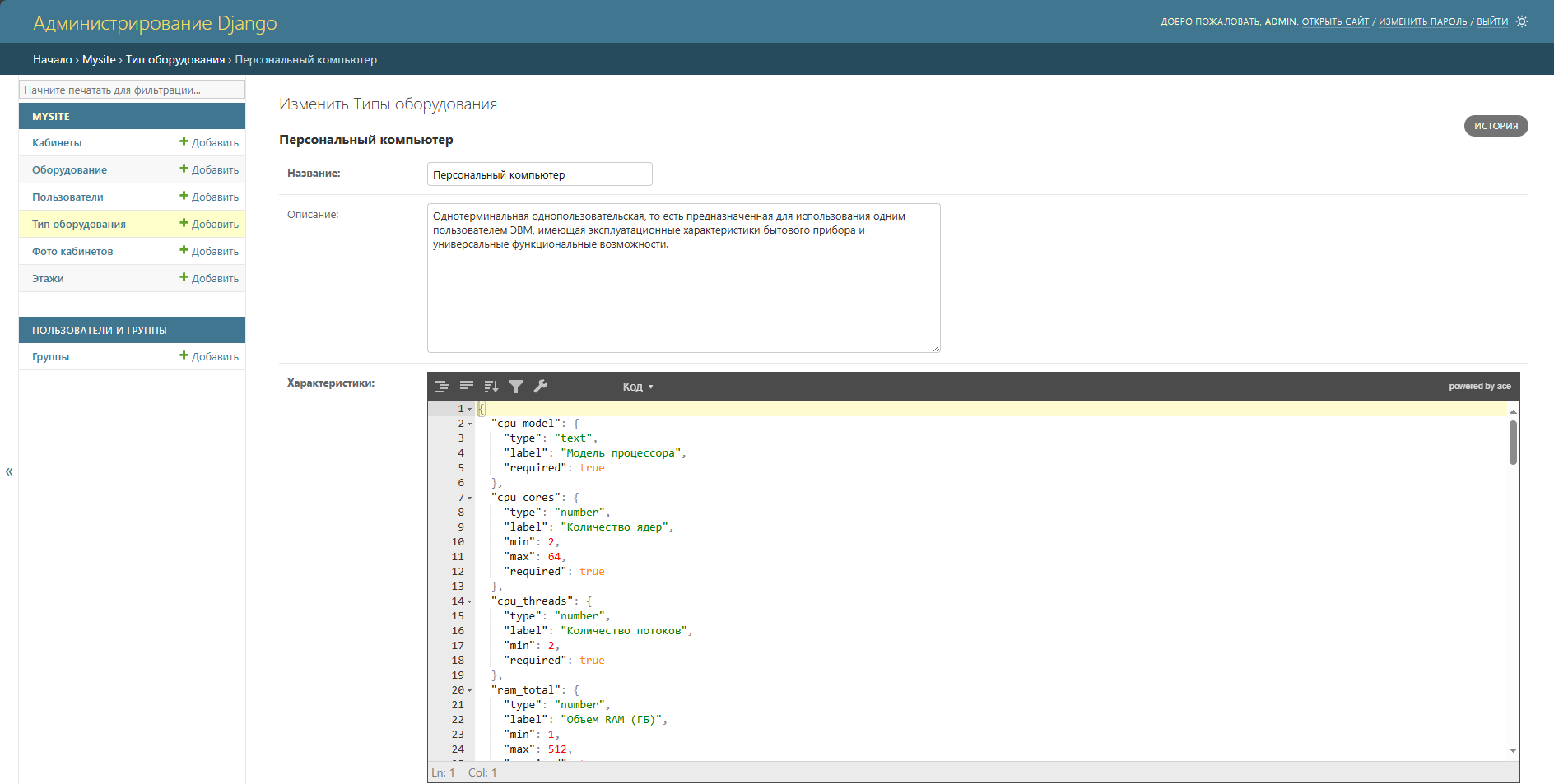
##### Рисунок Ж.3 – Страница 1 этажа



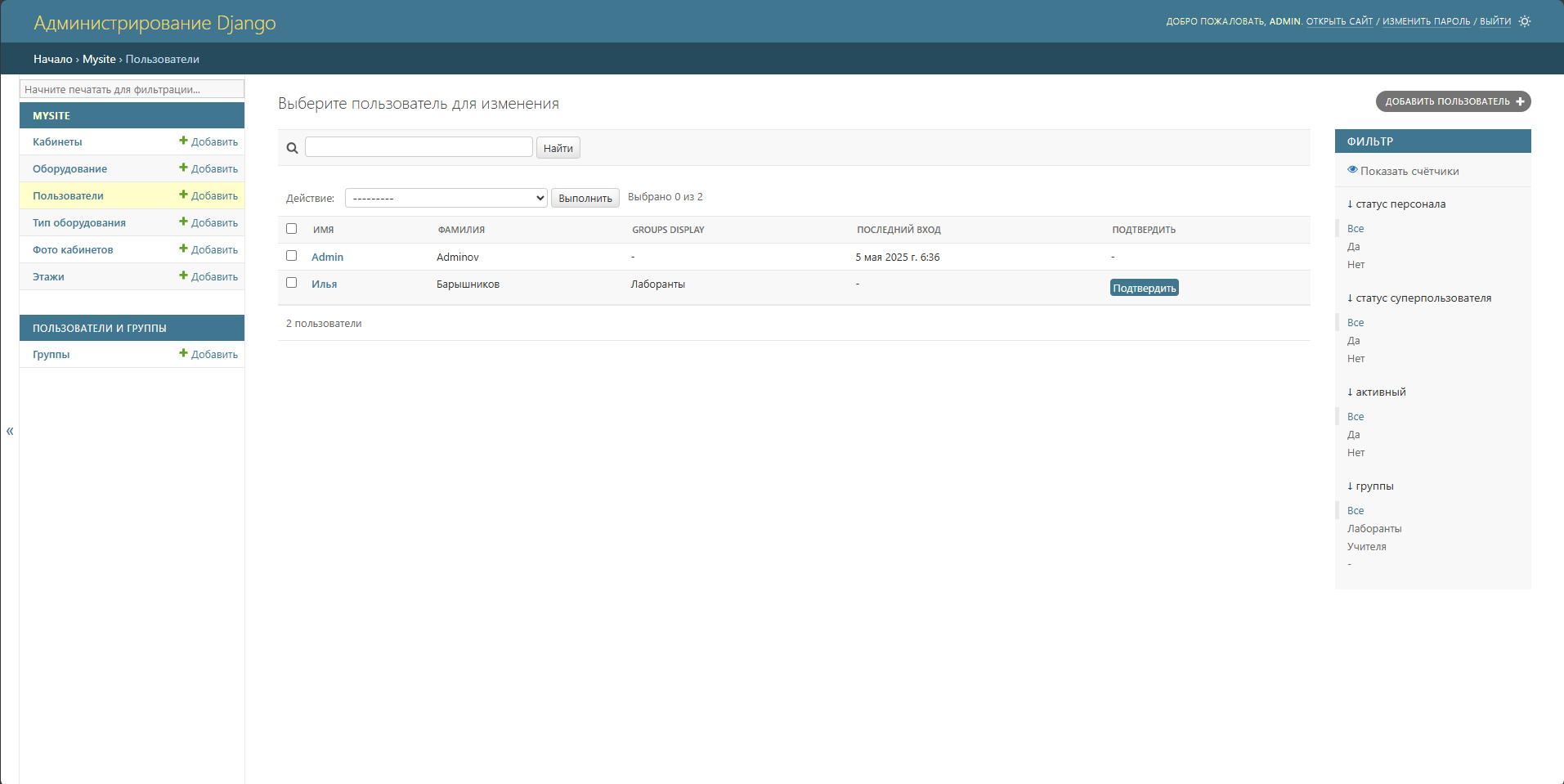
##### Рисунок Ж.4 – Страница фотографий кабинета



##### Рисунок Ж.5 – Страница добавления нового оборудования



##### Рисунок Ж.6 – Страница типа оборудования в админ панели



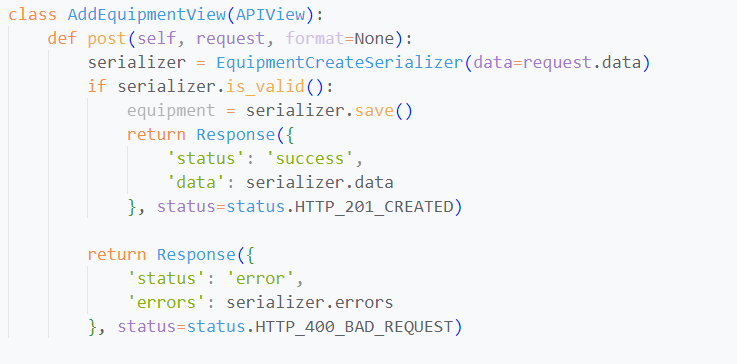
##### Рисунок Ж.7 – Страница пользователей и их заявок в админ панели

# Приложение З

(обязательное)



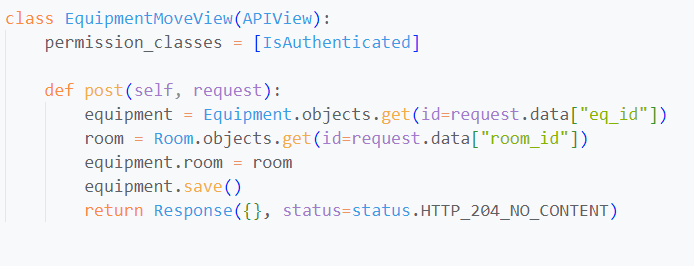
##### Листинг З.1.1 – Обработчик страницы добавления оборудования



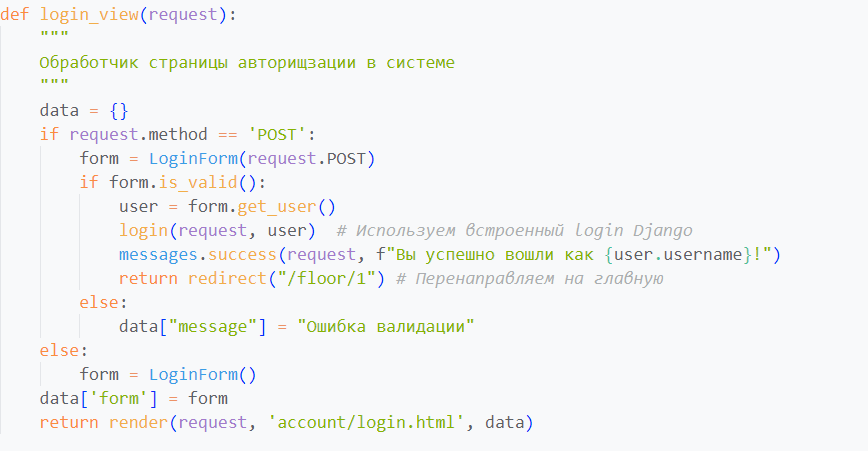
##### Листинг З.1.2 – Обработчик запроса к API на создание оборудования



##### Листинг З.1.3 – Сериализатор для оборудования



##### Листинг З.2– Обработчик api пути перемещения оборудования



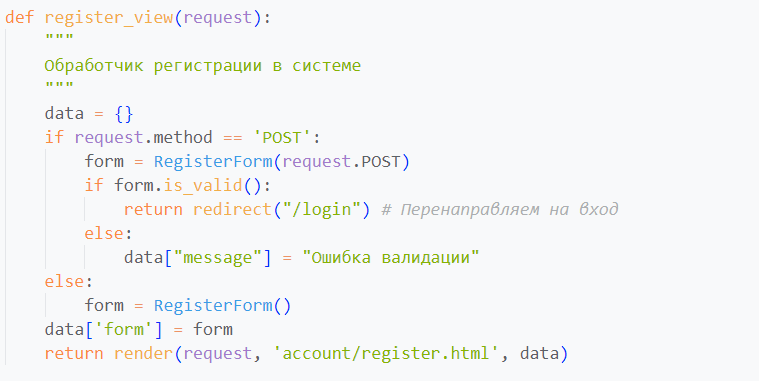
##### Листинг З.3 – Обработчик страницы авторизации



##### Листинг З.4.1 – Форма поиска оборудования



##### Листинг З.4.2 – Обработчик страницы кабинета и поиска оборудования



##### Листинг З.5 – Обработчик страницы регистрации