СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc160097838)

[1. ОБЩАЯ ЧАСТЬ 4](#_Toc160097839)

[1.1. Цель разработки 4](#_Toc160097840)

[1.2. Средства разработки 4](#_Toc160097841)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 4](#_Toc160097842)

[2.1. Постановка задачи 4](#_Toc160097843)

[2.1.1. Входные данные предметной области 4](#_Toc160097844)

[2.1.2. Выходные данные предметной области 4](#_Toc160097845)

[2.1.3. Подробные требования к проекту 4](#_Toc160097846)

[2.2. Внешняя спецификация 4](#_Toc160097847)

[2.2.1. Описание задачи 4](#_Toc160097848)

[2.2.2. Входные и выходные данные 4](#_Toc160097849)

[2.2.3. Методы 4](#_Toc160097850)

[2.2.4. Тесты 4](#_Toc160097851)

[2.2.5. Контроль целостности данных 4](#_Toc160097852)

[2.3. Проектирование 4](#_Toc160097853)

[2.3.1. Схема архитектуры приложения 4](#_Toc160097854)

[2.3.2. Логическая схема данных 4](#_Toc160097855)

[2.3.3. Физическая схема данных 4](#_Toc160097856)

[2.3.4. Структурная схема 4](#_Toc160097857)

[2.3.5. Функциональная схема 4](#_Toc160097858)

[2.3.6. Диаграмма классов 4](#_Toc160097859)

[2.3.7. Схема тестирования 4](#_Toc160097860)

[2.3.8. Схема пользовательского интерфейса 4](#_Toc160097861)

[2.4. Результат работы программы 4](#_Toc160097862)

[ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 4](#_Toc160097863)

[3.1. Инструментальные средства 4](#_Toc160097864)

[3.2. Отладка программы 4](#_Toc160097865)

[3.3. Защитное программирование 5](#_Toc160097866)

[3.4. Характеристики программы 5](#_Toc160097867)

[ЗАКЛЮЧЕНИЕ 5](#_Toc160097868)

[СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 5](#_Toc160097869)

[ПРИЛОЖЕНИЕ А. Текст программы 5](#_Toc160097870)

[ПРИЛОЖЕНИЕ Б. Сценарий и результаты тестовых испытаний 5](#_Toc160097871)

[ПРИЛОЖЕНИЕ В. Руководство пользователя 5](#_Toc160097872)

[ПРИЛОЖЕНИЕ Г. Скрипт базы данных 5](#_Toc160097873)

# ВВЕДЕНИЕ

Проект нацелен на разработку веб-приложения для учета и управления заявками на IT-поддержку. Заявки на IT-поддержку часто требуют систематизации, отслеживания статуса и сроков решения. Основная цель проекта – предоставить удобный веб-интерфейс, позволяющий пользователям создавать и отслеживать свои обращения, а сотрудникам IT-отдела – эффективно обрабатывать и анализировать обращения, назначать ответственных исполнителей, менять статусы и приоритеты.

Приложение будет обладать следующими ключевыми функциями: аутентификация и авторизация пользователей с различными ролями (клиент, сотрудник IT, администратор), регистрация заявок (тикетов), возможность редактирования, добавления комментариев, экспорт и импорт данных, формирование статистики и просмотр истории изменений.

Для реализации проекта будут использованы современные веб-технологии, такие как Python и фреймворк Django для серверной части, Microsoft SQL Server для хранения данных, Bootstrap для создания удобного интерфейса, а также дополнительные библиотеки для визуализации статистики (Chart.js) и резервного копирования.

Ожидаемым результатом проекта является создание веб-приложения с интуитивно понятным интерфейсом, высокой производительностью и надежностью, удовлетворящее потребности пользователей в удобном управлении заявками на IT-поддержку..

# ОБЩАЯ ЧАСТЬ

## Цель разработки

Целью разработки является обеспечение пользователей возможностью регистрировать заявки на IT-поддержку, отслеживать их статус, приоритет, назначенного исполнителя, а также управлять профилями пользователей и получать статистику по заявкам

## Средства разработки

Для проектирования, разработки и тестирования веб-приложения были использованы следующие программные и аппаратные средства, представленные в Таблице 1.

Таблица 1. Программные средства разработки ПО

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип средства | Название средства | Назначение |
| 1 | 2 | 3 | 4 |
| 1. | Операционная система | Microsoft Windows 11 | Организация взаимодействия программ и пользователя |
| 2. | Инструментальные средства разработки программного решения | Python 3.9+, Django, Visual Studio Code | Разработка серверной части веб-приложения |
| 3. | СУБД | Microsoft SQL Server | Хранение данных о пользователях, заявках, логах |
| 4. | Фреймвворк для фронтенда и UI | Bootstrap, HTML, CSS, JavaScript | Создание удобного и адаптивного интерфейса |
| 5. | Браузер | Google Chrome | Просмотр и тестирование веб-приложения |
| 6. | Дополнительные библиотеки | Chart.js, CSV парсер | Визуализация статистики и импорт/экспорт данных |
| 7. | Текстовый редактор | Microsoft Office 2019 | Написание документации |

В качестве средств вычислительной техники при разработке ПО

использовался ноутбук MSI Vector GP66 12UH-276XRU. Характеристики представлены в таблице 2.

Таблица 2. Технические средства

|  |  |  |
| --- | --- | --- |
| № | Тип средства | Название средства |
| 1 | 2 | 3 |
| Для разработки | | |
| Ноутбук MSI Vector GP66 12UH-276XRU | | |
| 1 | Размер экрана: | 15.6 |
| 2 | Разрешение экрана: | 1920x1080 |
| 3 | Линейка процессора: | Intel Core i7 |
| 4 | Количество ядер процессора: | 6 |
| 5 | Оперативная память: | 16 ГБ |
| 6 | Тип видеокарты: | дискретная |
| 7 | Видеокарта: | GeForce RTX 3070TI для ноутбуков |
| 8 | Конфигурация накопителей: | SSD |
| 9 | Общий объем всех накопителей: | 1TB |
| 10 | Операционная система | Windows 11 |

В Таблице 3 для использования веб-приложения конечным пользователям требуются следующие технические средств

Таблица 3. Технические средства для эксплуатации

|  |  |  |
| --- | --- | --- |
| № | Тип средства | Название средства |
| 1 | 2 | 3 |
| 1 | Веб-браузер: | Современный браузер (Chrome, Firefox, Edge) |
| 2 | Доступ к интернету: | Стабильное подключение |
| 3 | Операционная система клиента | Windows, macOS, Linux |
| 4 | Дисплей | Разрешение от 1366x768 и выше |

1. СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Необходимо разработать веб-приложение на языке Python с использованием фреймворка Django, предназначенное для учета заявок на IT-поддержку. Приложение должно обеспечивать аутентификацию пользователей, разграничение прав по ролям (администратор, сотрудник IT, клиент), а также возможность создания, редактирования, удаления заявок, добавления комментариев, изменения статусов и приоритетов. Данные хранятся в базе данных Microsoft SQL Server. Также необходимо реализовать экспорт и импорт данных (CSV, SQL), представления статистики и логирование действий.

## Входные данные предметной области

* Информация о пользователях (Уникальный идентификатор пользователя, имя пользователя, адрес электронной почты, хэш пароля, роль пользователя в системе)
* Данные о заявках (тикетах): (Уникальный идентификатор заявки, заголовок заявки, подробное описание проблемы, статус заявки, приоритет заявки, дата и время создания заявки, дата и время последнего обновления заявки, идентификатор пользователя, идентификатор пользователя , создавшего заявку)
* Комментарии к заявках: (Уникальный идентификатор комментария, идентификатор заявки, идентификатор пользователя, текст комментария, дата и время добавления комментария)
* Логи действий: (Уникальный идентификатор записи лога, идентификатор пользователя, описание совершённого действия, дата и время совершения действия)
* Справочники статусов и приоритетов: (Список статусов с идентификаторами и названиями, список приоритетов с идентификаторами и названиями)

## Выходные данные предметной области

* Список заявок с фильтрацией по статусу, приоритету, исполнителю (Идентификатор заявки, заголовок, текущий статус, приоритет, имя назначенного сотрудника, дата создания )
* Детальная информация о выбранное заявке с комментариями (Подробный заголовок и описание заявки, статус, приоритет, дата создания и последнего обновления, назначенный исполнитель, история изменений статуса или приоритета, список комментариев к заявке с указанием автора, даты и текста каждого комментария )
* Список пользователей, их роли и активность (Идентификатор пользователя, имя пользователя, email, роль в системе, информацию о том, какие действия пользователь совершал )
* Статистические данные в виде диаграмм (Распределение заявок по статусам, распределение заявок по приоритетам, Среднее время решения заявок)
* Экспортированные файлы (ID заявки, описание, дата создания, назначенный исполнитель)

## Подробные требования к проекту

Подробные требования к разработке мобильного приложения «Разработать систему для IT-поддержки» представлены в приложении А. Техническое задание

## Внешняя спецификация

## Описание задачи

Необходимо разработать веб-приложение на языке Python с использованием фреймворка Django, предназначенное для учета заявок на IT-поддержку. Приложение должно обеспечивать аутентификацию пользователей, разграничение прав по ролям (администратор, сотрудник IT, клиент), а также возможность создания, редактирования, удаления заявок, добавления комментариев, изменения статусов и приоритетов. Данные хранятся в базе данных Microsoft SQL Server. Также необходимо реализовать экспорт и импорт данных (CSV, SQL), представления статистики и логирование действий.

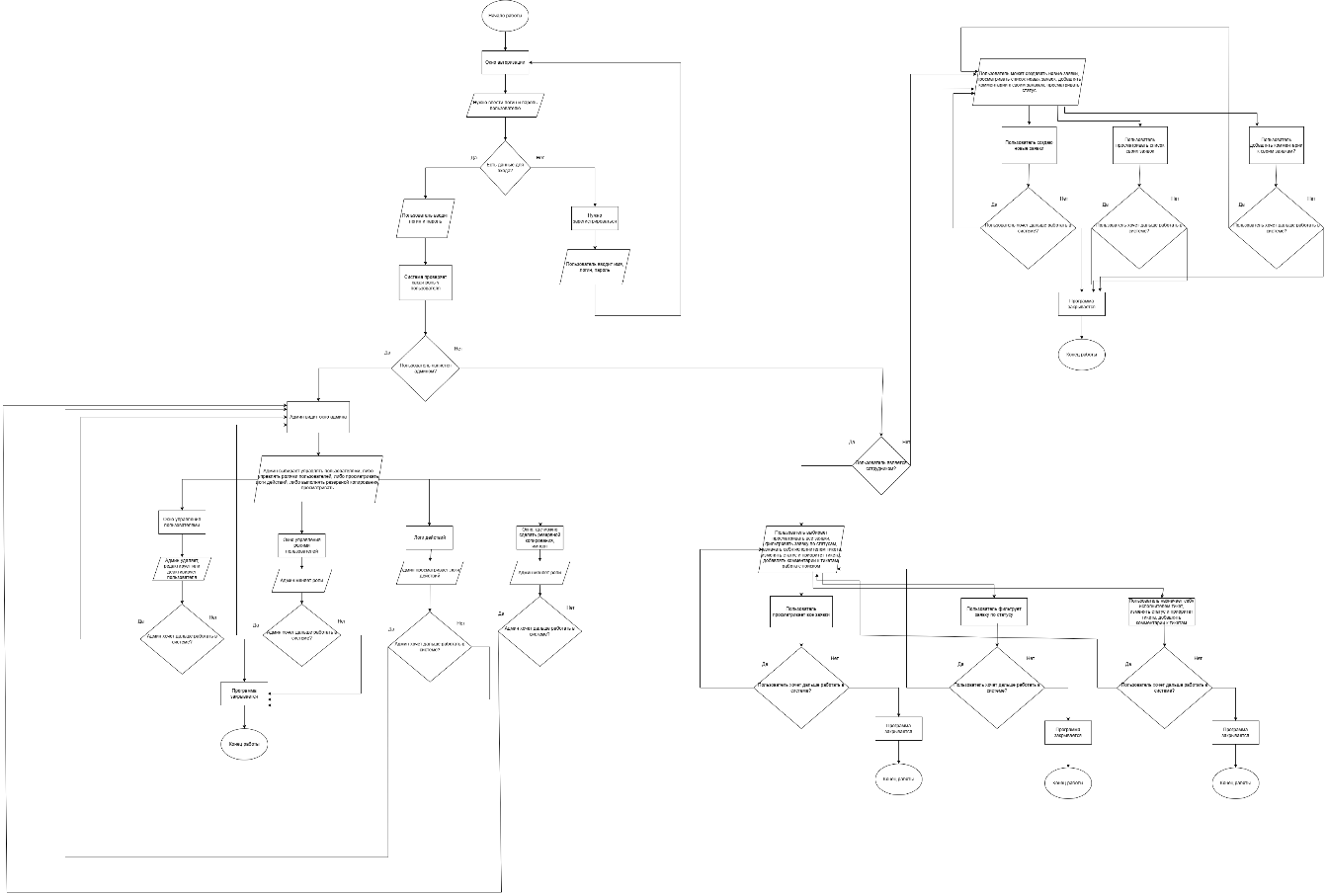


Рисунок 1 – Схема функционирования

Создание системы idef0

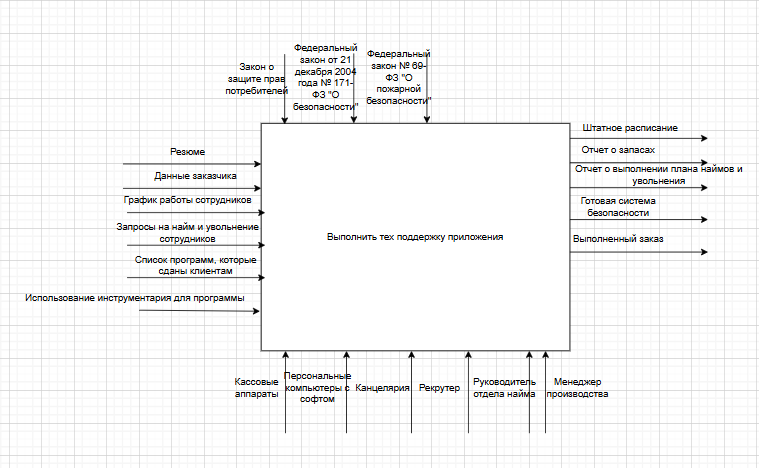


Рисунок 2 – Idef0

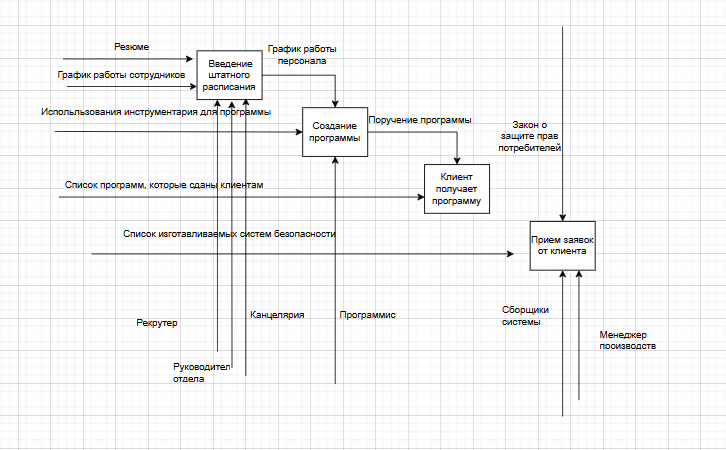


Рисунок 3 – idef0

## Входные и выходные данные

В таблице 4 представлены входные данные, которые предоставляет мобильное приложение.

Таблица 4. Входные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя | Тип | Ограничения | Описание |
| 1 | 2 | 3 | 4 | 5 |
| Таблица ролей | | | | |
| 1 | role\_id | INT IDENTITY | В виде цифр название | Первичный ключ |
| 2 | role\_name | NVARCHAR(50) | Минимум 6 символов | Название роли (уникальное) |
| Данные пользователей | | | | |
| 1 | user\_id | INT IDENTITY | Не может быть пустым, макс. 100 символов | Первичный ключ |
| 2 | username | NVARCHAR(50) | Максимум 10 символов | Имя пользователя (уникальное) |
| 3 | password\_hash | VARBINARY(512) | Макс. 1000 символов | Хэшированный пароль |
| 4 | email | NVARCHAR(100) | Почта с @ | Электронная почта (уникальная) |
| 5 | full\_name | NVARCHAR(100) | Должна быть валидное имя | Полное имя |
| 6 | role\_id | INT | >= 0 | Внешний ключ на Roles.role\_id |
| 7 | is\_active | BITr | >= 0 | Активен ли пользователь (по умолчанию 1) |
| Статусы таблица | | | | |
| 1 | status\_id | INT IDENTITY | Минимум 6 символов | Первичный ключ |
| 2 | status\_name | NVARCHAR(50) | Максимум 10 символов | Название статуса (уникальное) |
| Приоритеты таблица | | | | |
| 1 | priority\_id | INT IDENTITY | Максимум 10 символов | Первичный ключ |
| 2 | priority\_name | NVARCHAR(50) | Минимум 6 символов | Название приоритета (уникальное) |
| Таблица тикеты | | | | |
| 1 | ticket\_id | INT IDENTITY | Минимум 6 символов | Первичный ключ |
| 2 | title | NVARCHAR(200) | Должна быть валидное имя | Название тикета |
| 3 | description | TEXT | Должна быть валидное имя | Описание тикета |
| 4 | created\_at | DATETIME | Максимум 10 символов | Дата создания (по умолчанию GETDATE()) |
| 5 | updated\_at | DATETIME | Не может быть пустым, макс. 100 символов | Дата обновления |
| 6 | status\_id | INT | Тут должно быть нормально написанный ключ | Внешний ключ на Statuses.status\_id |
| 7 | priority\_id | INT | Тут должно быть нормально написанный ключ | Внешний ключ на Priorities.priority\_id |
| 8 | user\_id | INT | Не может быть пустым, макс. 100 символов | Внешний ключ на Users.user\_id (кто создал) |
| Таблица билеты ассигнат | | | | |
| 1 | assignment\_id | INT IDENTITY | Тут должно быть нормально написанный ключ | Первичный ключ |
| 2 | ticket\_id | INT | Тут должно быть нормально написанный ключ | Внешний ключ на Tickets.ticket\_id |
| 3 | assignee\_id | INT | Минимум 6 символов | Внешний ключ на Users.user\_id (ассигнат) |
| 4 | assigned\_at | DATETIME | Минимум 6 символов | Дата назначения (по умолчанию GETDATE()) |
| Таблица комментарией | | | | |
| 1 | comment\_id | INT IDENTITY | Тут должно быть нормально написанный ключ | Первичный ключ |
| 2 | ticket\_id | INT | Минимум 6 символов | Внешний ключ на Tickets.ticket\_id |
| 3 | user\_id | INT | Максимум 10 символов | Внешний ключ на Users.user\_id |
| 4 | comment\_text | TEXT | Должна быть валидное имя | Текст комментария |
| 5 | commented\_at | DATETIME | Должна быть валидное имя | Дата комментария (по умолчанию GETDATE()) |

В таблице 5 предоставлены входные данные, которые предоставляет мобильное приложение.

Таблица 5. Выходные данные

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Регистрация пользователя | | | | | |
| № | Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | username | String | Не может быть пустым, макс. 50 символов | Текстовое поле | Имя пользователя |
| 2 | password | String | Не может быть пустым, макс. 128 символов | Поле пароля | Пароль пользователя |
| 3 | confirm\_password | String | Должен совпадать с password | Поле пароля | Подтверждение пароля |
| 4 | email | String (Email) | Не может быть пустым, уникальное | Текстовое поле | Электронная почта |
| 5 | full\_name | String | Макс. 100 символов, может быть пустым | Текстовое поле | Полное имя пользователя |
| 6 | role | String | Должно существовать в таблице Roles | Выпадающий список | Роль пользователя |
| Вход пользователя | | | | | |
| 1 | username | String | Не может быть пустым | Текстовое поле | Имя пользователя |
| 2 | password | String | Не может быть пустым | Поле пароля | Пароль пользователя |
| Список Пользователей | | | | | |
| 1 | user\_id | Integer | Первичный ключ | - | Уникальный идентификатор пользователя |
| 2 | username | String | Уникальное | - | Имя пользователя |
| 3 | full\_name | String | Макс. 100 символов | - | Полное имя пользователя |
| 4 | email | String (Email) | Уникальное | - | Электронная почта |
| 5 | role | String | Ссылка на Roles | - | Роль пользователя |
| Создание Тикета | | | | | |
| № | Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | title | String | Не может быть пустым, макс. 200 символов | Текстовое поле | Название тикета |
| 2 | description | String | Макс. 1000 символов, может быть пустым | Текстовое поле | Описание проблемы |
| 3 | status | String | Должно существовать в таблице Statuses | Выпадающий список | Статус тикета |
| Просмотр Тикета | | | | | |
| 1 | ticket\_id | Integer | Первичный ключ | - | Уникальный идентификатор тикета |
| 2 | title | String | - | - | Название тикета |
| 3 | description | String | - | - | Описание проблемы |
| 4 | status | String | - | - | Статус тикета |
| 5 | priority | String | - | - | Приоритет тикета |
| 6 | created\_at | DateTime | - | - | Дата создания тикета |
| 7 | updated\_at | DateTime | Может быть пустым | - | Дата обновления тикета |
| Редактирование Тикета | | | | | |
| № | Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | title | String | Не может быть пустым, макс. 200 символов | Текстовое поле | Название тикета |
| 2 | description | String | Макс. 1000 символов, может быть пустым | Текстовое поле | Описание проблемы |
| 3 | status | String | Должно существовать в таблице Statuses | Выпадающий список | Статус тикета |
| Удаление Тикета | | | | | |
| 1 | - | ticket\_id | Integer | Первичный ключ | - |
| Импорт Тикетов из CSV | | | | | |
| 1 | 1 | file | File | Должен быть в формате CSV | Файл (CSV) |

## Методы

При разработке веб-приложения был применен язык разметки HTML, а также использованы принципы наследования, инкапсуляции, паттерн «Builder» и методология объектно-ориентированного программирования (ООП). Для создания интерфейса приложения использовались CSS (включая фреймворк Bootstrap) и язык программирования JavaScript, включая компоненты такие как кнопки, текстовые поля, списки, вкладки, окна диалогов и изображения. Фреймворк Django обеспечил структуру проекта и управление серверной частью, позволяя эффективно реализовать модели, формы и представления. Пример интерфейса представлен на рисунке 2.

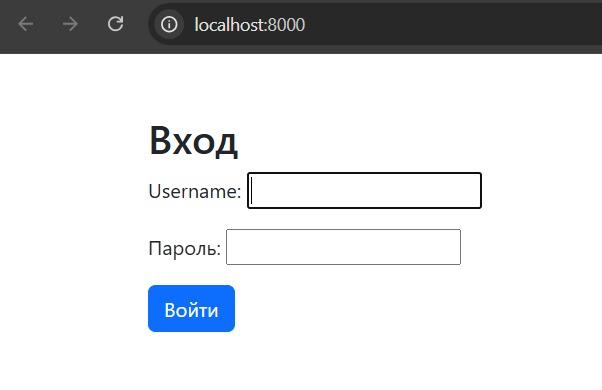


Рисунок 2 – Пример интерфейса

Для того, чтобы на данные выводились были сделаны методы в admin.py, чтобы обеспечить передачу данных в нужный формат.

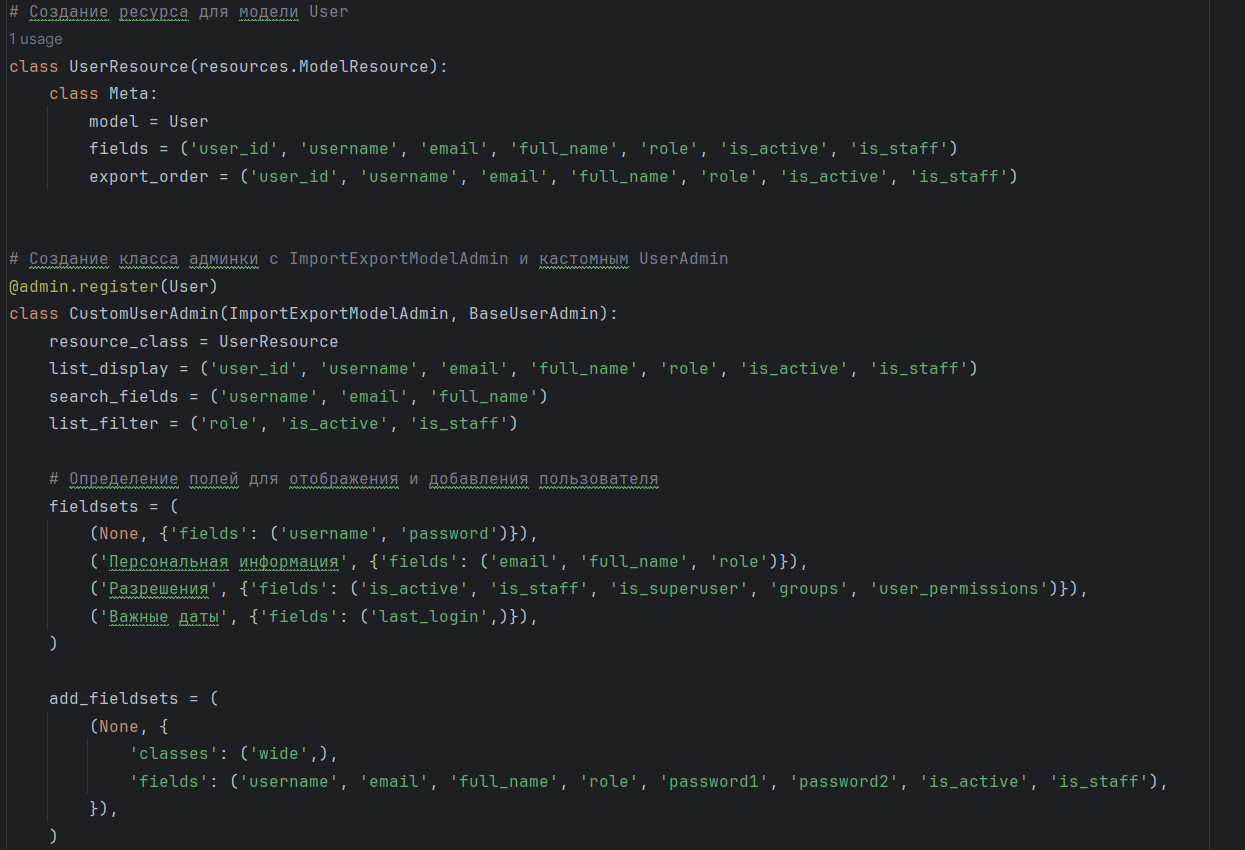


Рисунок 3 – Передача данных

В рамках веб-приложения используются формы для создания и обработки данных, связанных с тикетами и комментариями к ним. Для этого применяются классы, унаследованные от forms.ModelForm, что обеспечивает автоматическое связывание форм с моделями базы данных.

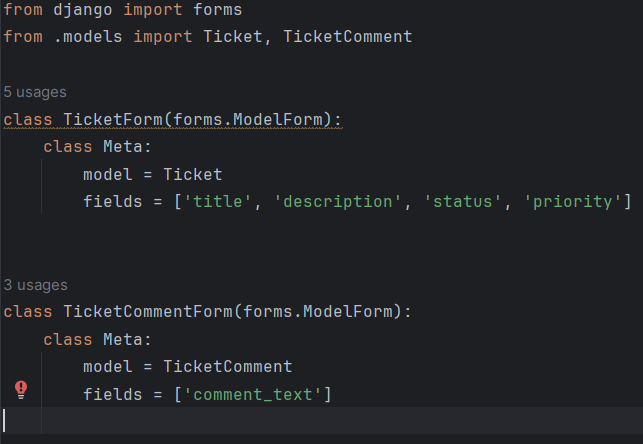


Рисунок 4 – Добавление данных

* + 1. Тесты

В приложении В «Сценарий и результаты тестовых испытаний» представлен сценарий тестирования и результаты проведенных тестов. При тестировании программного комплекса были использованы различные методы:

1) Функциональное тестирование, оценивающее реализованные функции ПО и их правильность;

2) Ручное тестирование без использования дополнительных программных средств;

3) Позитивное и негативное тестирование для проверки соответствия поведения ПО ожиданиям пользователя;

4) Тестирование «белого ящика» с доступом к коду программного продукта и «черного ящика» без доступа к коду;

5) Тестирование на основе требований, где требования были определены до начала тестирования.

## Контроль целостности данных

## В ходе разработки информационной системы были внедрены методы проверки вводимых данных, обеспечивающие их надежность. Таблица отражает контроль за целостностью данных, описывающий сценарии и реакции приложения при выполнении функций.

Таблица 6. Проверка целостности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ситуация | Аномалия | Реакция | Примечание |
| 1 | 2 | 3 | 4 | 5 |
| Авторизация | | | | |
| 1 | Пользователь ввел некорректный email | Некорректный формат email | Показать сообщение об ошибке "Неверный формат email." | Проверка с помощью Валидация поля email в форме UserLoginForm |
| 2 | Пользователь не ввел пароль | Пустое поле пароль | Показать сообщение об ошибке "Введите пароль." | Проверка с помощью TextUtils.isEmpty(password) |
| 3 | Введены корректные email и пароль | Нет аномалии | Вызвать метод user\_login() и перенаправить на список тикетов | Успешная авторизация пользователя |
| Регистрация | | | | |
| 1 | Пользователь не ввел имя пользователя | Пустое поле имя пользователя | Показать сообщение об ошибке "Введите имя пользователя." | Валидация поля username в UserRegistrationForm |
| 2 | Пользователь авторизован | Пользователь не ввел email | Считать тип пользователя из базы данных и перейти на соответствующую страницу | Валидация поля email в UserRegistrationForm |
| 3 | Пользователь ввел некорректный email | Некорректный формат email | Показать сообщение об ошибке "Неверный формат email." | Валидация поля email |
| 4 | Пользователь не ввел пароль | Пустое поле пароль | Показать сообщение об ошибке "Введите пароль." | Валидация поля password в UserRegistrationForm |
| Создание тикета | | | | |
| 1 | Пользователь не ввел заголовок тикета | Пустое поле заголовок | Показать сообщение об ошибке "Введите заголовок тикета." | Валидация поля title в TicketForm |
| 2 | Пользователь не выбрал статус тикета | Не выбран статус | Показать сообщение об ошибке "Выберите статус тикета." | Валидация поля status в TicketForm |
| 3 | Пользователь не выбрал приоритет тикета | Не выбран приоритет | Показать сообщение об ошибке "Выберите приоритет тикета."" | Валидация поля priority в TicketForm |
| 4 | Все данные введены корректно | Нет аномалии | Вызвать метод ticket\_create() и создать тикет | Успешное создание тикета |
| Добавление комментария к тикету | | | | |
| 1 | Пользователь не ввел текст комментария | Пустое поле категория | Показать сообщение об ошибке "Введите текст комментария." | Валидация поля comment\_text в TicketCommentForm |
| 2 | Все данные введены корректно | Нет аномалии | Вызвать метод добавления комментария и сохранить его | Успешное добавление комментария к тикету |
| Экспорт тикетов в CSV | | | | |
| 1 | Пользователь не авторизован | Пользователь не вошел в систему | Перенаправить на страницу входа | Проверка наличия user\_id в сессии |
| 2 | Пользователь авторизован | Нет аномалии | Вызвать метод ticket\_export\_csv() и скачать файл | Успешный экспорт тикетов в CSV |

## Проектирование

## Схема архитектуры приложения

В проекте используется многоуровневая архитектура, которая обеспечивает разделение ответственности и упрощает поддержку и масштабирование приложения. Основные слои архитектуры включают:

Слой бизнес-логики - этот слой отвечает за обработку данных, реализацию бизнес-правил и управление состоянием приложения. Бизнес-логика реализована в представлениях (views) и формах (forms). Представления взаимодействуют с моделями для выполнения операций создания, чтения, обновления и удаления данных.UI, и выполнение запросов к репозиторию.

Слой доступа к данным этот слой управляет взаимодействием с базой данных, доступ к данным осуществляется через ORM (Object-Relational Mapping) Django, который позволяет работать с базой данных с помощью Python-классов и методов

Хранилизе данных отвечает за физическое хранение данных. Используется SQL Server в качестве базы данных, с которой взаимодействует Django через настройки подключения.

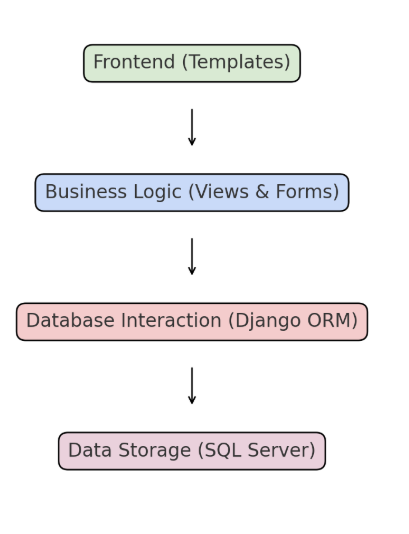


Рисунок 7 – Структурная схема

## Логическая схема данных

Логическая схема, которая используется для работы в проекте с базами данных

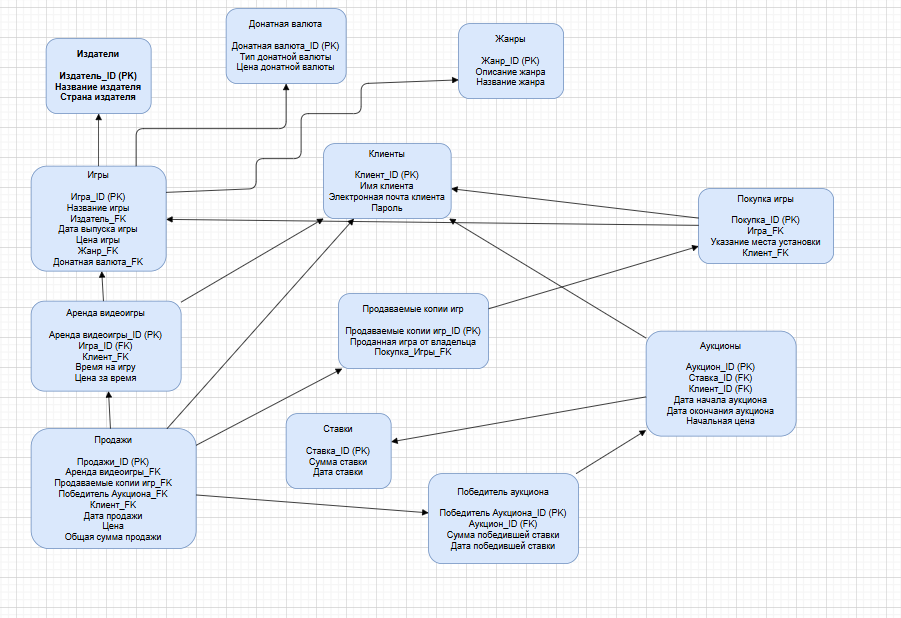


Рисунок 8 – Логическая схема

* + 1. Физическая схема данных

Физическая модель послужила основой для реализации хранения данных и работы БД

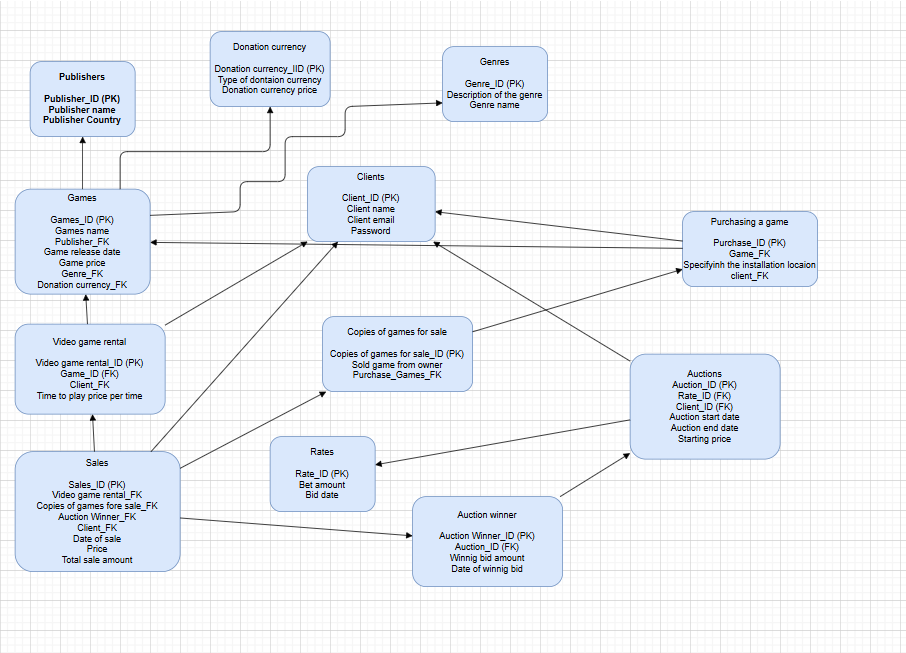


Рисунок 9 – Физическая модель

В базе данных было создано 3 сущности (таблицы), каждая содержит поля для записи информации, у каждой присутствует уникальное поле идентификатор.

Таблица 6. Словарь данных базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Поле | Тип данных | Обязательность | Описание |
| 1 | 2 | 3 | 4 | 5 |
| Таблица «Books» | | | | |
| PK | Books\_id | Integer | Not null | Идентификатор книги |
| FK | Categories\_Id | Integer | Not null | Идентификатор категории |
|  | description | String | Нет | Описание книги |
|  | downloadsCount | Integer | Да | Количество загрузок |
|  | timestamp | Long | Да | Время создания |
|  | title | String | Да | Название |
|  | uid | String | Да | Уникальный идентификатор пользователя |
|  | url | String | Да | URL для загрузки |
|  | viewsCount | Integer | Да | Количество просмотров |
| Таблица «Roles» | | | | |
| PK | Roles\_Id | Integer | Да | Уникальный идентификатор категории |
|  | roles | String | Да | Название категории |
|  | timestamp | Long | Да | Время создания категории |
| Таблица «Users» | | | | |
| PK | Users\_id | Integer | Да | Уникальный идентификатор пользователя |
|  | email | String | Да | Email пользователя |
|  | name | String | Да | Имя пользователя |
|  | profileImage | String | Нет | URL изображения профиля пользователя |
|  | timestamp | Long | Да | Время создания профиля пользователя |
|  | uid | String | Да | Уникальный идентификатор пользователя |
|  | userType | String | Да | Тип пользователя (например, "user" или "admin") |

## Структурная схема

На рисунке 13 представлена структурная схема мобильного приложения, в котором показано взаимодействие классов

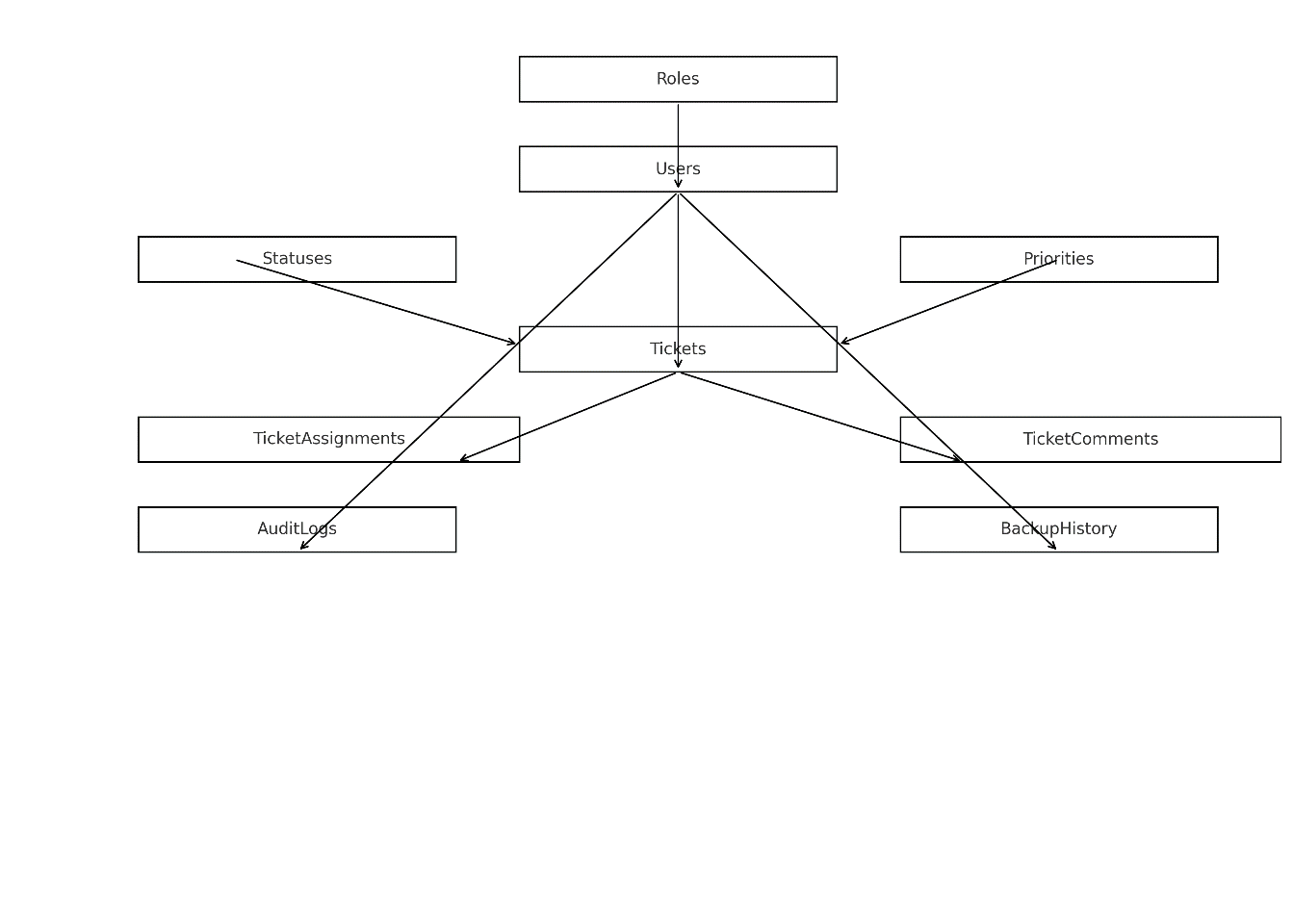


Рисунок 10 – Структура схема

Таблица 7. Описание моделей структурной схемы

|  |  |  |
| --- | --- | --- |
| № | Название модуля | Описание |
| Таблицы модулей | | |
| 1 | CategoryAdd | Добавление новой категории. |
| 2 | DashboardAdmin | Панель управления для администратора, отображающая данные и предоставляющая доступ к функциям администрирования. |
| 3 | DashboardUser | Панель управления для пользователя, предоставляющая доступ к основным функциям приложения. |
| 4 | ForgotPassword | Восстановление пароля пользователя. |
| 5 | Login | Вход в систему, включает методы валидации данных пользователя. |
| 6 | Main | Запускаемая при старте приложения. |
| 7 | PdfAdd | Добавления нового PDF-файла. |
| 8 | PdfDetail | Просмотра деталей выбранного PDF-файла. |
| 9 | PdfEdit | Редактирования PDF-файлов. |
| 10 | PdfListAdmin | Отображения списка PDF-файлов для администратора. |
| 11 | PdfView | Просмотра PDF-файлов. |
| 12 | Profile | Просмотра профиля пользователя. |
| 13 | ProfileEdit | Редактирования профиля пользователя. |
| 14 | Registe | Регистрации нового пользователя. |
| 15 | Splash | Сплэш-экран, отображаемый при запуске приложения. |
| Таблица моделей | | |
| 1 | ModelUser | Модель данных для категории, содержит поля и методы для работы с категориями. |
| 2 | ModelTicket | Модель данных для комментариев, содержит поля и методы для работы с комментариями. |
| 3 | Modelt\_support | Модель данных для PDF-файлов, содержит поля и методы для работы с PDF. |

## Функциональная схема

На рисунке 11 представлена функциональная схема мобильного приложения, на нем изображена эксплуатация

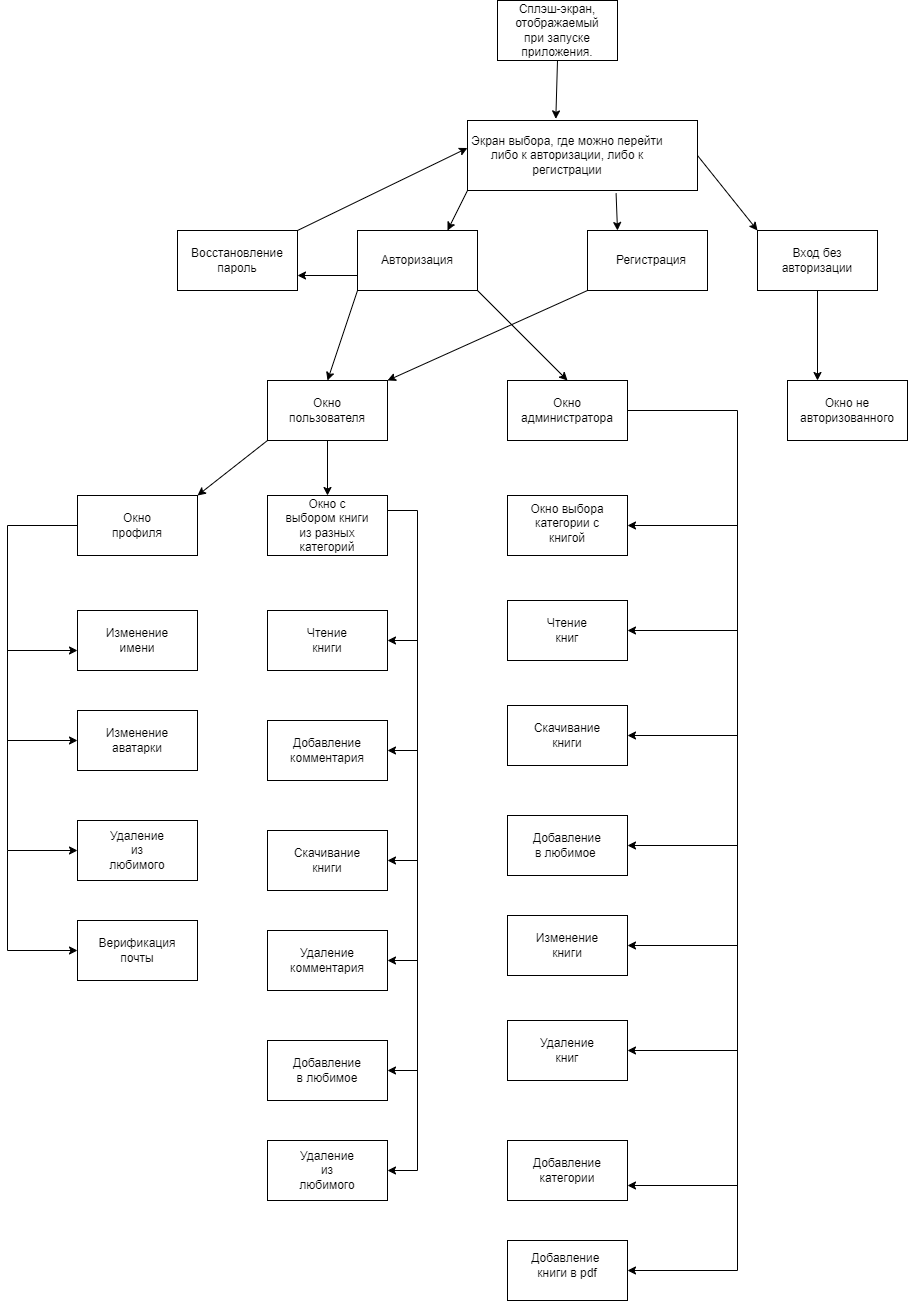


Рисунок 11 – Функциональная схема

## Диаграмма классов

На рисунке 12 показана диаграмма классов, на которой показано расположение и отношение классов друг к другу

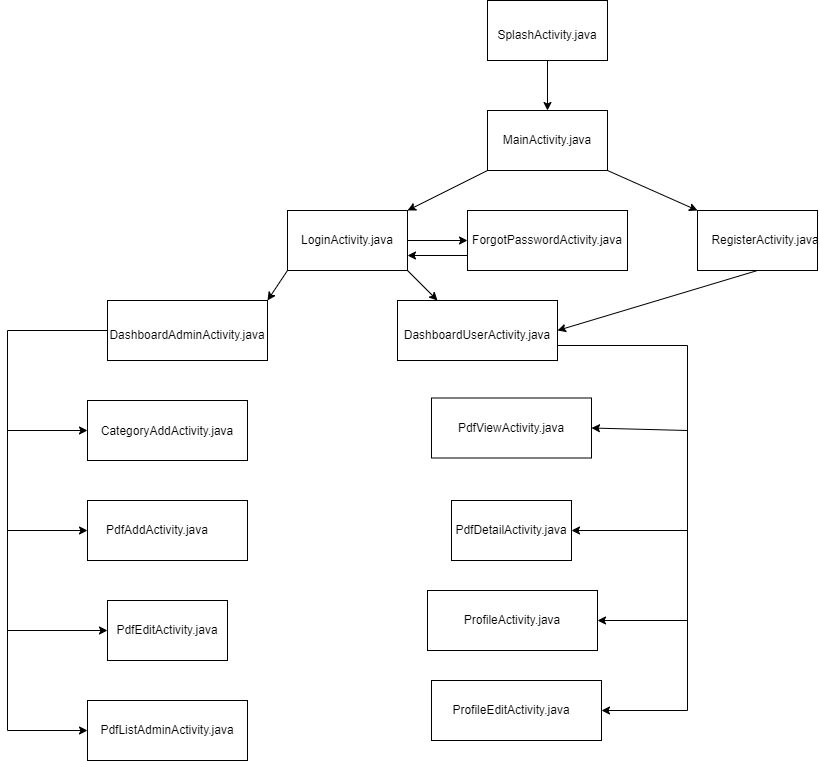


Рисунок 12 – Диаграмма классов

## Схема тестирования

На рисунке 13 представлена схема тестирования, в которой приведены тесты, которые будут проводиться

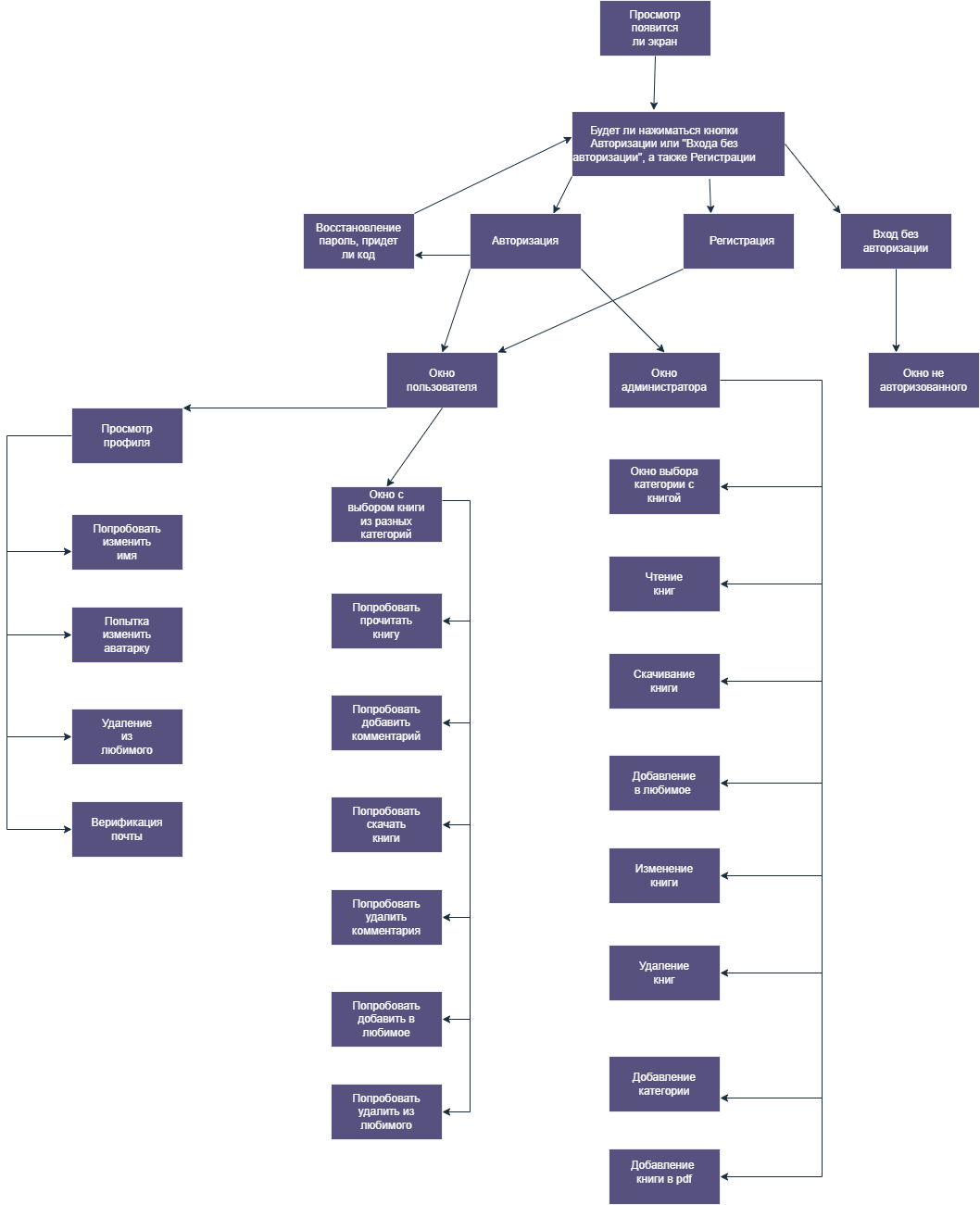


Рисунок 13 – Схема тестирования

## Схема пользовательского интерфейса

На рисунке 14 показана схема пользовательского интерфейса веб-приложения. Здесь можно увидеть интерфейс для пользователя

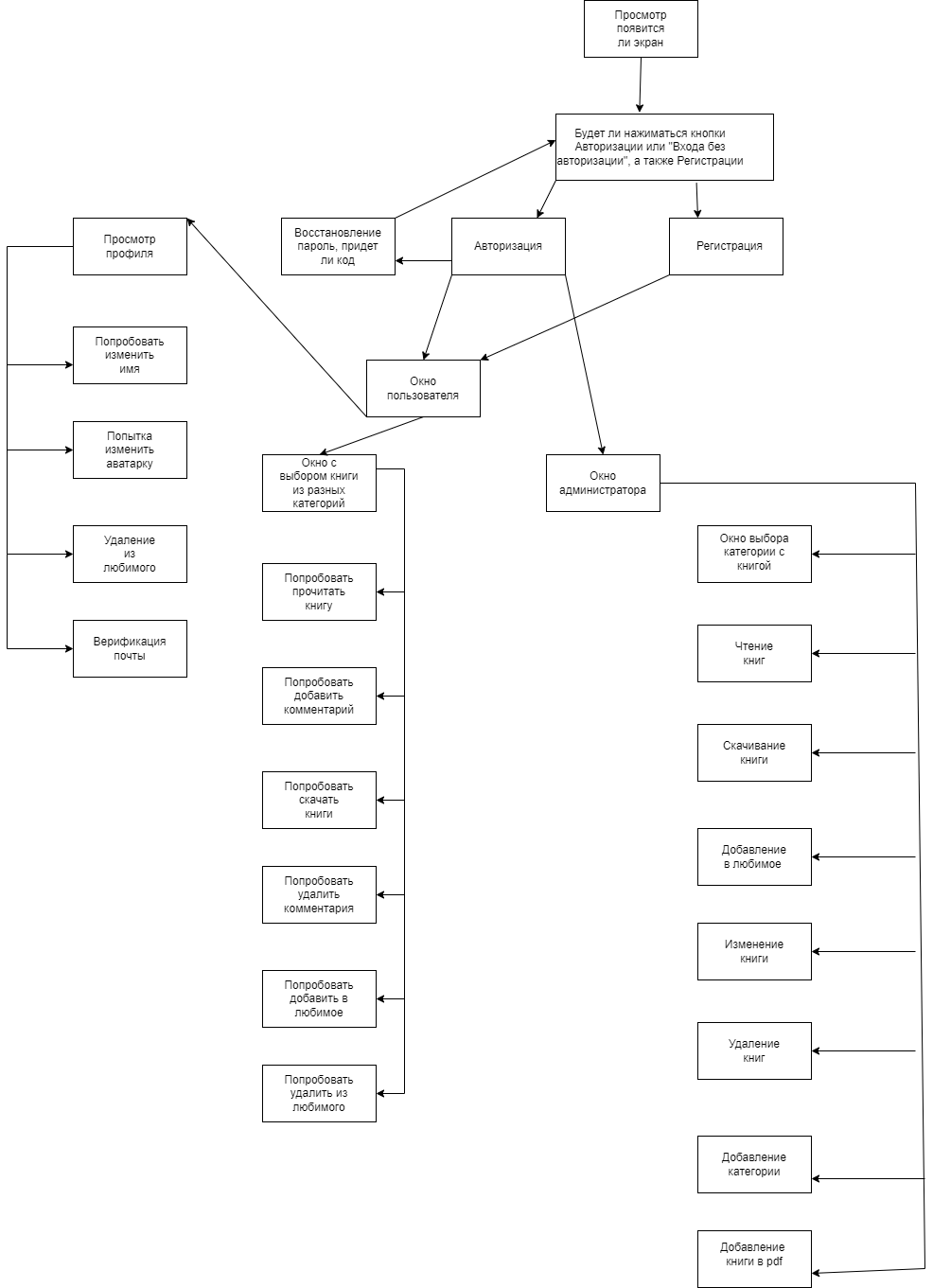


Рисунок 14 – Схема интерфейса

## Результат работы программы

В ходе разработки программного решения по поставленной задаче, было разработано мобильное приложение.

Рассмотреть практическое применение веб-приложения можно в Приложении Г. Руководство пользователя

Вход в аккаунт

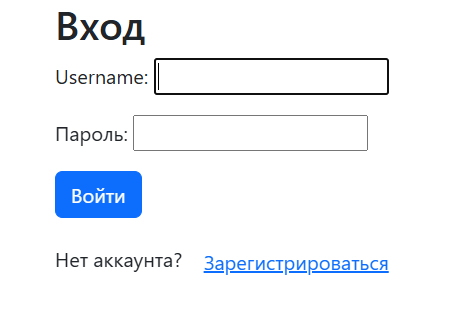


Рисунок № 15 – Вход

Регистрация в аккаунте

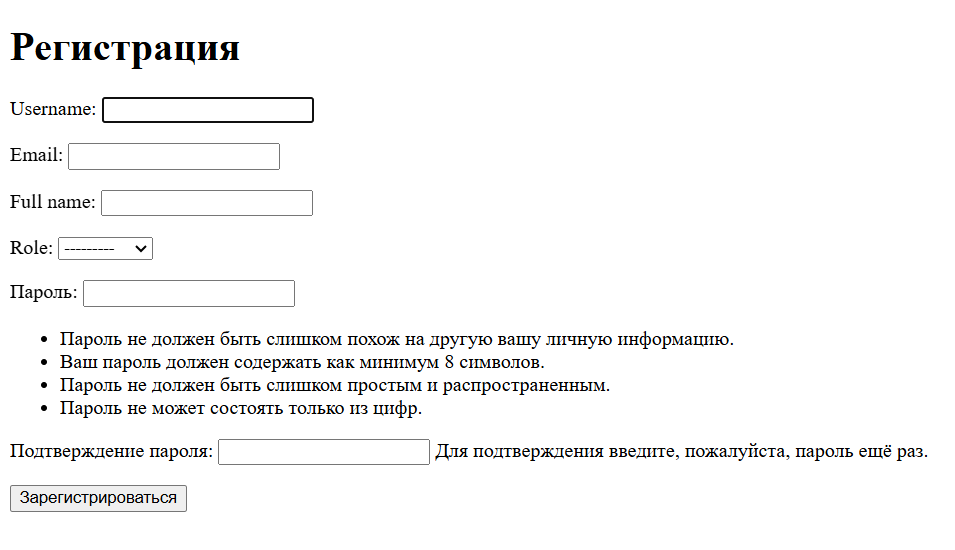


Рисунок № 16 – Регистрация

Добавление пользователя, таблиц

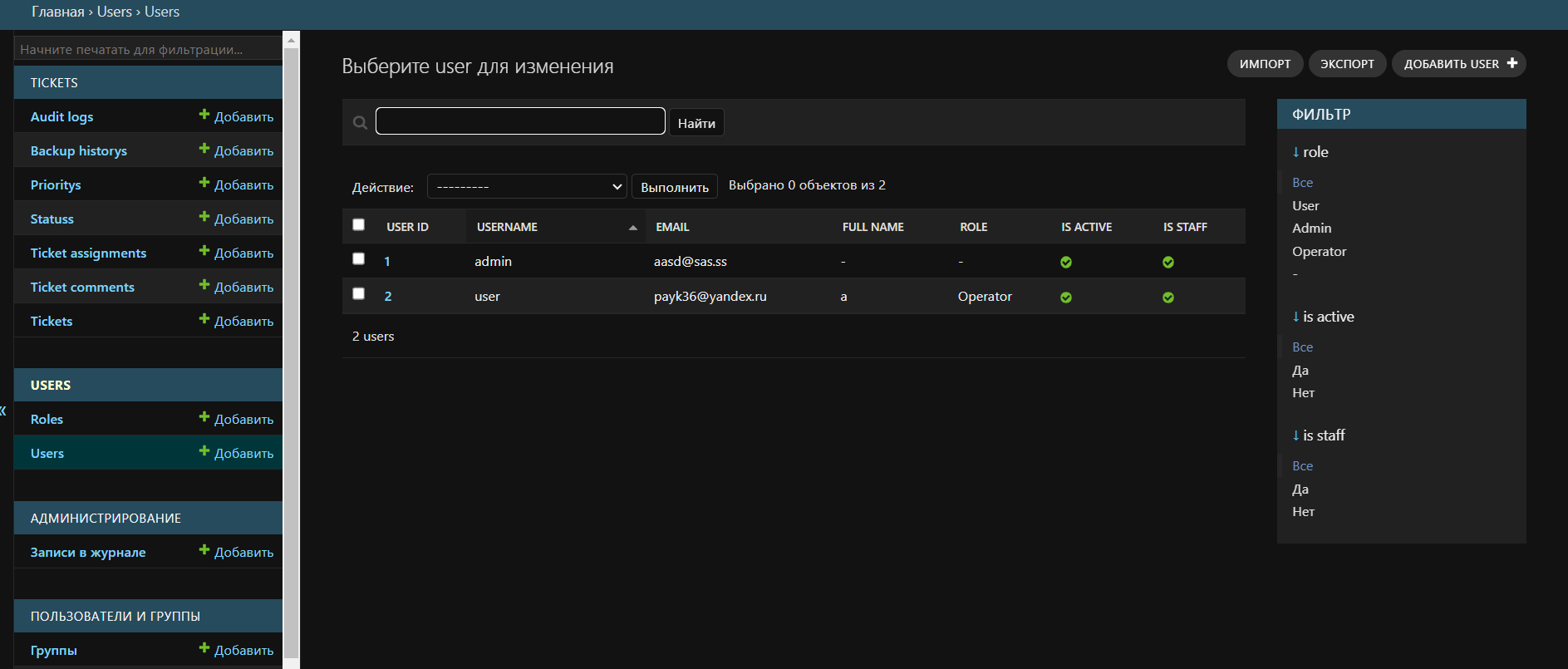


Рисунок № 17 – Добавление пользователя, таблиц

Изменение пользователя с кучей правок

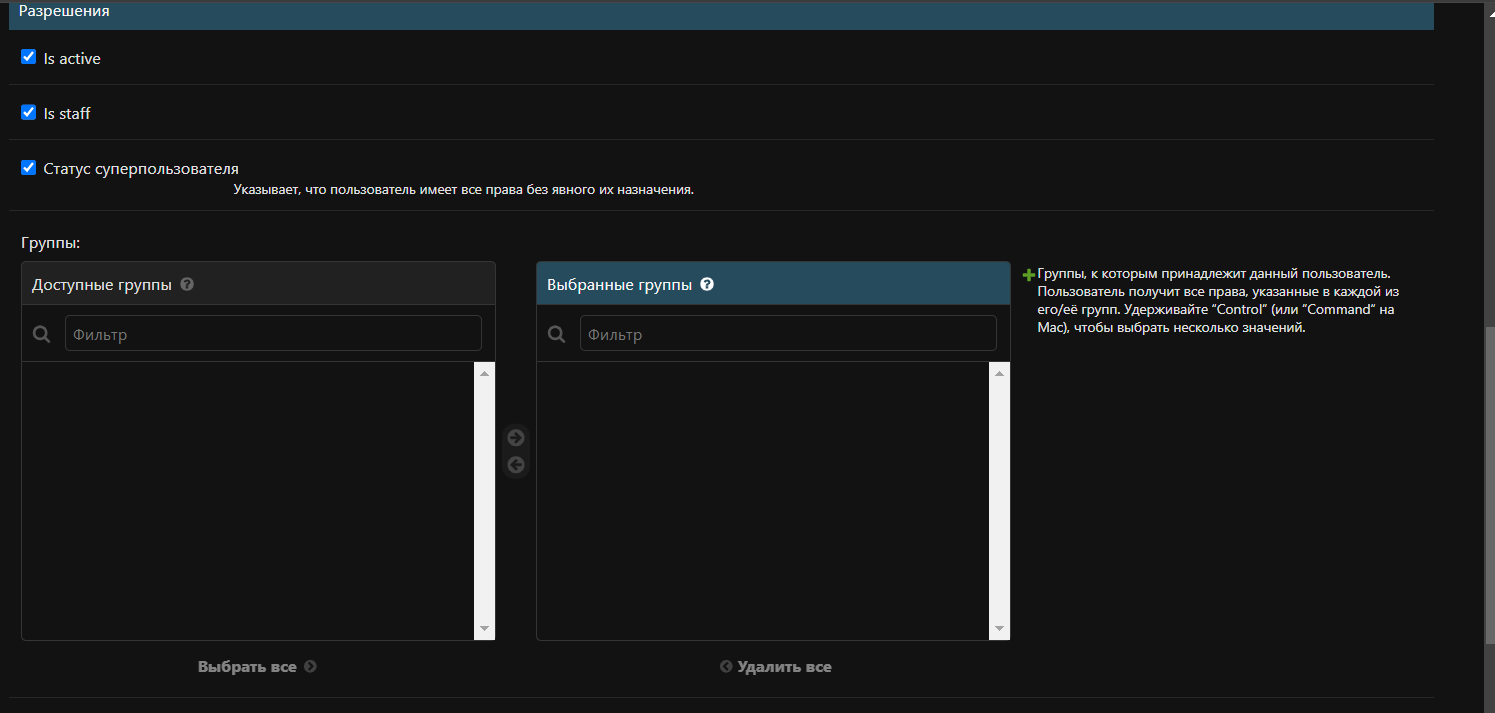


Рисунок № 18 – Изменение пользователя

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## 3.1. Инструментальные средства

Для разработки мобильного приложения использовалась среда разработки Android Studio. Среда разработки была выбрана благодаря простому использованию, поддержки языка Java.

За основной текстовый редактор был взят Microsoft Word 2019 года. Пакет Microsoft Office был изначально встроен в компьютер и является лучшим программным обеспечением для работы с документами среди приложений. Word позволяет создавать документы распространенного формата, которые каждый пользователь ПК сможет доступно прочитать и распечатать, что также повлияло на выбор.

## 3.2. Отладка программы

Проект представляет собой систему управления заявками технической поддержки (IT Support System) с использованием базы данных ItSupportDB и веб-приложения на основе фреймворка Django. Система предназначена для управления пользователями, тикетами, их статусами и приоритетами, а также обеспечивает возможность резервного копирования и ведения логов.

Пример ошибки показан на рисунке 35

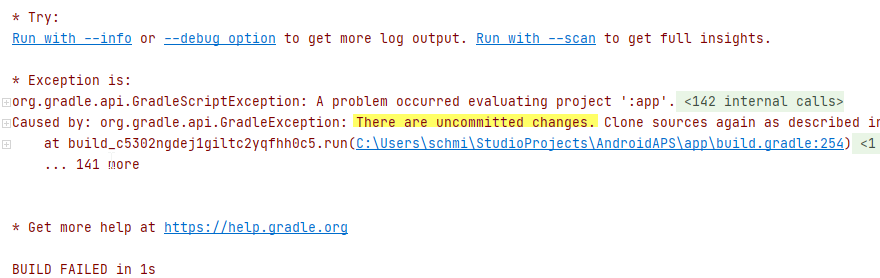


Рисунок 35 - Ошибка

## 3.3. Защитное программирование

В программном коде предусмотрены защиты от ошибок и неожиданных вылетов приложения в различных частях кода, у меня использована проверка на «null»

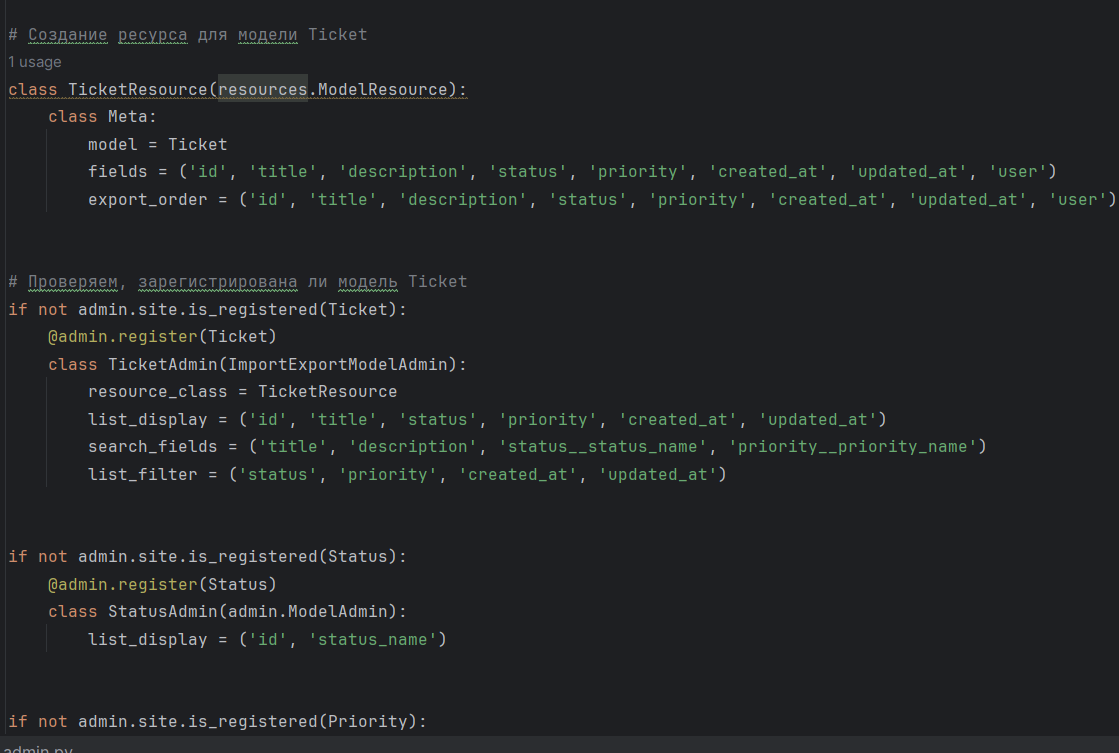


Рисунок 36 – Проверка

Обработка ошибок при отправке верификационного письма

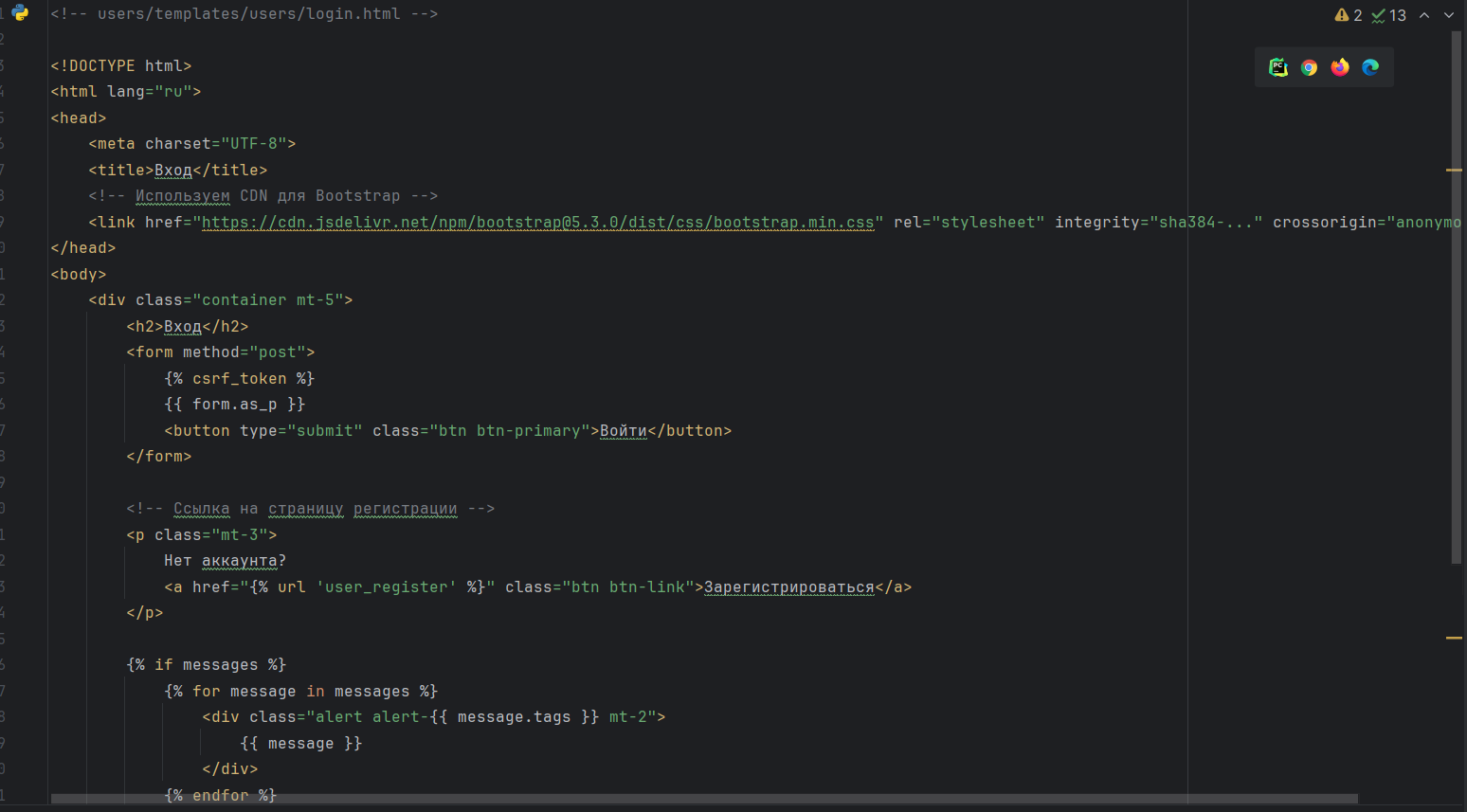


Рисунок 37 – Проверка ошибок при верификации

Обработка ошибок при загрузке пользователя

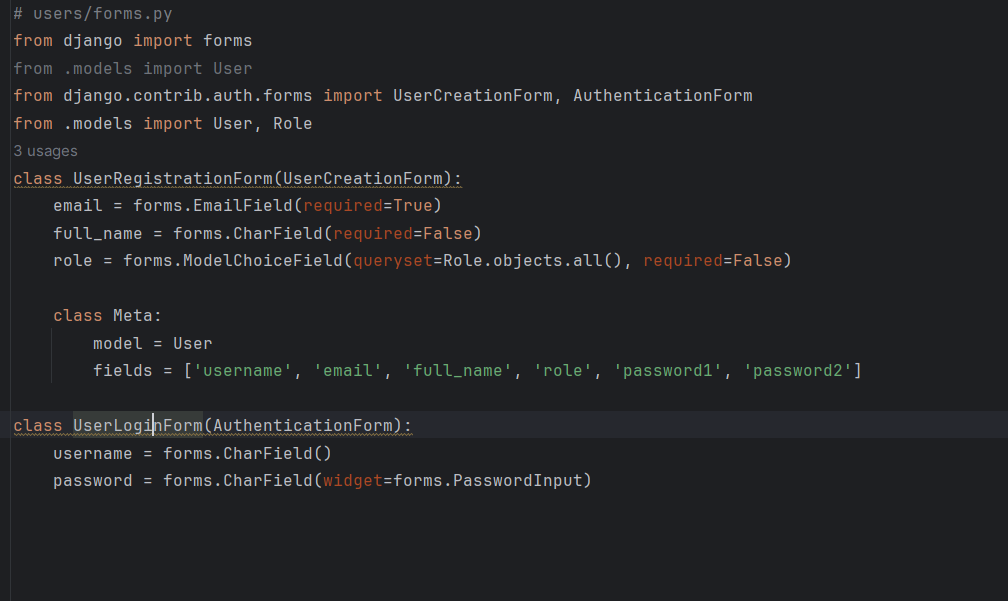


Рисунок 38 – Любимая книга

## 3.4. Характеристики программы

Разработанное веб-приложение нужно для системы по учете it-заявок

Характеристики программы представлены в приложении А «Текст программы» в таблице 1 «Модули».

# ЗАКЛЮЧЕНИЕ

В данном проекте была разработана система управления заявками технической поддержки с использованием базы данных ItSupportDB. Основной целью проекта было создание веб-приложения, позволяющего пользователям создавать, отслеживать и управлять тикетами, а администраторам эффективно управлять системой и контролировать процесс обработки заявок.

Проект включает функционал для различных ролей пользователей, таких как администратор и обычный пользователь, что позволяет гибко распределять доступ и возможности каждого типа пользователя. Система поддерживает авторизацию и регистрацию пользователей, обеспечивая безопасность и надежность доступа. Реализована валидация данных на этапе регистрации и входа, что исключает ошибки и обеспечивает корректность введенных данных.

# СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

1. JetBrains - IntelliJ IDEA: https://www.jetbrains.com/idea/ (Дата использования 23.05.2024)
2. Oracle - Java SE Documentation: https://docs.oracle.com/en/java/javase/index.html (Дата использования 24.05.2024)
3. Stack Overflow -Java: https://stackoverflow.com/questions/tagged/java (Дата использования 09.06.2024)
4. StackOverflow: https://stackoverflow.com (Дата использования 16.04.2023)

# GitHub - Android Firebase Authentication: https://github.com/firebase/quickstart-android/tree/master/auth (Дата 23 использования: 28.05.24)

1. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен. - СПб.: Питер, 2016. - 96 c. (Дата использования: 15.05.23)
2. Oracle - Java™ Platform, Enterprise Edition (Java EE) 8: https://docs.oracle.com/javaee/8/index.html (Дата использования: 25.05.24)
3. Java - https://learn.javascript.ru (Дата использования 28.05.2024)
4. Герман, О.В. Программирование на Java и C# / О.В. Герман. - СПб.: BHV, 2012. - 512 c. (Дата использования: 20.05.24)
5. Соломон, М. Oracle Программирование на языке Java / М. Соломон. - М.: Лори, 2010. - 484 c. (Дата использования: 01.06.24)