

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра программирования и информационных технологий

Техническое задание  
на разработку веб-сайта  
«Инвентаризация и управления складом»

Исполнители

\_\_\_\_\_ Н. В. Вернигоров  
\_\_\_\_\_ Д. А. Неупокоев  
\_\_\_\_\_ Н. М. Шестопалов  
\_\_\_\_\_ Н. П. Михайлов  
\_\_\_\_\_ Д. А. Толпыгин  
\_\_\_\_\_ Д. О. Голубева

Заказчик

\_\_\_\_\_ В. С. Тарасов

Воронеж 2025

## Содержание

1 Общие сведения.....	6
1.1 Наименование приложения.....	6
1.2 Разработчики и заказчик .....	6
1.3 Перечень документов, на основании которых создается система.....	6
1.4 Состав и содержание работ по созданию системы.....	7
1.5 Порядок контроля и приемки автоматизированной системы .....	7
2 Цели и назначение создания автоматизированной системы .....	9
2.1 Цели создания АС .....	9
2.2 Назначения создания АС.....	9
3 Функциональные требования.....	10
3.1 Функциональные возможности для неавторизованных пользователей. 10	
3.2 Функциональные возможности для рабочего .....	10
3.3 Функциональные возможности для администратора .....	10
3.4 Функциональные возможности для системного администратора.....	11
4 Нефункциональные требования .....	12
5 Требования к системе .....	13
5.1 Требования к системе в целом.....	13
5.1.1 Требования к структуре и функционированию системы .....	13
5.1.2 Требования к защите информации от несанкционированного доступа .....	14
5.1.3 Требования к аутентификации .....	14
5.1.4 Требования к численности и квалификации персонала .....	14
5.1.5 Требования к патентной чистоте.....	15
5.1.6 Требования к масштабируемости и открытости .....	15
5.1.7 Обработка ошибок .....	15
5.2 Требования к видам обеспечения.....	15
5.2.1 Требования к программному обеспечению приложения .....	15
5.2.2 Требования к лингвистическому обеспечению .....	16

6 Структура приложения .....	17
7 Дизайн приложения .....	18
8 Навигация по приложению .....	19
8.1 Основное навигационное меню .....	19
9 Описание страниц приложения .....	20
9.1 Главная страница .....	20
9.2 Страница входа.....	20
9.3 Страница приобретения лицензии .....	20
9.4 Страница загрузки видео.....	20
9.5 Страница историй проверок.....	20
9.6 Страница сотрудников.....	20
9.7 Страница складов .....	21
9.8 Страница компаний .....	21
Приложение .....	22

## Термины, используемые в техническом задании

- **Backend** - программно-аппаратная часть сервиса.
- **Django** - высокоуровневый Python фреймворк, который позволяет быстро создавать безопасные и поддерживаемые сайты.
- **Frontend** - сторона пользовательского интерфейса к программно-аппаратной части сервиса.
- **GitHub** - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.
- **Python** - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.
- **REST API** - стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.
- **Гость** - неавторизованный на веб-сервисе человек, пользующийся ограниченным функционалом веб-сервиса.
- **Администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса в пределах компании.
- **Системный администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса за пределами компании.
- **Клиентская сторона** - программно-аппаратная часть веб-приложения. Отвечает за функционирование внутренней части веб-приложения.
- **Личный кабинет** - раздел сервиса, в котором Пользователь может получить доступ к своим данным.

— **Лицензия** - коммерческая лицензия на использование веб-сайта.

## **1 Общие сведения**

### **1.1 Наименование приложения**

Полное наименование сайта: «Storix» - веб приложение инвентаризации и управлением складом.

Условное обозначение сайта: «Storix»

### **1.2 Разработчики и заказчик**

Заказчик: Старший Преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий

Разработчик: 1 команда группы 10.

Состав команды разработчика:

- Вернигоров Никита Владимирович;
- Неупокоев Дмитрий Александрович;
- Шестопалов Никита Максимович;
- Михайлов Никита Павлович;
- Толпыгин Данил Алексеевич;
- Голубева Дарья Олеговна.

### **1.3 Перечень документов, на основании которых создается система**

- Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149–ФЗ;
- Федеральный закон "О персональных данных" от 27.07.2006 N 152–ФЗ;

- Федеральный закон "О коммерческой тайне" от 29.07.2004 N 98–ФЗ;
- Закон РФ от 07.02.1992 N 2300-1 (ред. от 11.06.2021) "О защите прав потребителей".

#### **1.4 Состав и содержание работ по созданию системы**

Состав и содержание работ по созданию сайта включают в себя следующие этапы:

- Сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы 16.02.25 – 01.03.25;
- Анализ предметной области, анализ конкурентов и построение структуры требований, ведущих к решению поставленных задач и целей 01.03.25 – 16.03.25;
- Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД 16.03.25 – 01.04.25;
- Разработка рабочего проекта, состоящего из написания программного кода, отладки и корректировки кода программы 01.04.25 – 16.05.25;
- Проведение тестирования программного обеспечения 16.05.25 – 01.06.25.

#### **1.5 Порядок контроля и приемки автоматизированной системы**

Предварительные отчёты по работе будет проводиться во время рубежных аттестаций:

- 1 аттестация (конец марта 2025) - создан репозиторий проекта на GitHub, распределены задачи проекта в таскменеджере Jira, создан

- проект с общей логикой системы, предоставлены промежуточные результаты по курсовому проекту и готовое техническое задание;
- 2 аттестация (конец апреля 2025) - написана основополагающая часть кода приложения, реализована БД и ее взаимодействие с сервером, проведена отладка и доработка кода, проведено тестирование по работе системы;
- 3 аттестация (конец мая 2025) - разработан курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.



## **2 Цели и назначение создания автоматизированной системы**

### **2.1 Цели создания АС**

Целями создания сайта являются:

- Упрощение процесса проведения инвентаризаций за счёт использования ИИ;
- Минимизация вероятности ошибок при инвентаризации;
- Получение прибыли с подписки.

### **2.2 Назначения создания АС**

Приложение позволяет решать следующие задачи:

- Загружать видеозапись со склада для определения при помощи ИИ товаров и их местоположения, а также сравнения полученной информации с данными компаний;
- Создание складов с назначением на них рабочих.

### **3 Функциональные требования**

Для взаимодействия с данной системой выделяют следующие виды пользователей:

- Неавторизованный пользователь;
- Рабочий;
- Администратор;
- Системный администратор.

#### **3.1 Функциональные возможности для неавторизованных пользователей**

- При первом использовании сайта пользователь может зарегистрироваться и продолжить сеанс в авторизованной зоне;
- При повторном использовании сайта пользователь может авторизоваться в ранее созданный аккаунт.

#### **3.2 Функциональные возможности для рабочего**

- Рабочий может загрузить видео;
- Рабочий может просмотреть отчёт;
- Рабочий может просмотреть историю инвентаризаций.

#### **3.3 Функциональные возможности для администратора**

- Администратор может сменить временный пароль;
- Администратор может создать/удалить склад;
- Администратор может создать/удалить аккаунт рабочего;
- Администратор может просмотреть список складов и рабочих;

- Администратор может добавить рабочего на склад;
- Администратор может убрать рабочего со склада;
- Администратор может просмотреть историю инвентаризаций.

### **3.4 Функциональные возможности для системного администратора**

Системный администратор может создать/удалить аккаунт компании

## 4 Нефункциональные требования

- Производительность: Приложение должно обеспечивать быструю загрузку и отзывчивость интерфейса, даже при большой активности пользователей, а также масштабироваться для работы с большим количеством пользователей;
- Безопасность: Данные пользователей должны быть защищены от несанкционированного доступа, данные для входа будут выдаваться администраторами;
- Надежность: Приложение должно быть стабильным и надежным, минимизируя возможность сбоев и потери данных;
- Совместимость с различными платформами: Сайт должен быть доступен для использования на большинстве популярных браузеров (Яндекс, Google, Firefox);
- Документация: Приложение должно предоставлять понятную и структурированную документацию, охватывающую все аспекты приложения, включая функциональность, API, компоненты системы и бизнес-процессы. Документация должна быть полной, ясной и актуальной, чтобы обеспечить легкость понимания как пользователями, так и разработчиками.

## **5 Требования к системе**

### **5.1 Требования к системе в целом**

#### **5.1.1 Требования к структуре и функционированию системы**

Приложение должно быть построено на трехуровневой архитектуре: клиент (веб-приложение) - сервер - база данных.

Пользователь взаимодействует с серверной частью приложения при помощи клиентского приложения. Сервер посредством REST API возвращает пользователю необходимые данные. Администратор взаимодействует с серверной частью приложения при помощи клиентского приложения.

Серверная часть приложения включает в себя:

- Сервис регистрации и авторизации;
- API сервис;
- Сервис для работы с базой данных;
- Сервис обработки данных.

Backend:

- Язык программирования Python;
- Фреймворк Django;
- СУБД PostgreSQL.

Frontend:

- Язык гипертекстовой разметки HTML;
- Формальный язык описания внешнего вида документа CSS;
- Язык программирования JavaScript;
- Фреймворк React.

### **5.1.2 Требования к защите информации от несанкционированного доступа**

Фреймворк Django включает в себя множество встроенных механизмов защиты:

- CSRF-защита (Cross-Site Request Forgery): Django автоматически генерирует уникальный токен для каждой сессии пользователя и вставляет его в формы. При отправке формы этот токен проверяется для предотвращения атак CSRF.
- X-Content-Type-Options: nosniff: Django автоматически устанавливает заголовок X-Content-Type-Options: nosniff, чтобы предотвратить браузер от интерпретации файлов с неправильным Content-Type.

### **5.1.3 Требования к аутентификации**

В системе должна быть реализована идентификация и проверка доступа при входе в систему по логину и паролю длиной не менее 6 символов.

Система защиты должна подвергаться проверке подлинности идентификации — осуществлять аутентификацию.

Система не должна предоставлять доступ к защищенным данным неавторизованным пользователям.

### **5.1.4 Требования к численности и квалификации персонала**

В системе предполагается наличие ролей пользователей — администратор, рабочий, компания и неавторизованный пользователь.

Пользователь с ролью администратор должен обладать знаниями и навыками необходимыми для поддержания работоспособности системы.

Использование системы подразумевает, что как авторизованные, так и неавторизованные пользователи имеют определённый уровень знаний и умений в обращении с компьютером или ноутбука. Это включает в себя понимание основ навигации по интерфейсу, способности к осуществлению управление мышкой или тачпадом для выполнения команд, а также использование клавиатуры для ввода текста. Пользователям также необходимо уметь загружать видеозаписи в приложение.

### **5.1.5 Требования по патентной части**

Приложение должно быть разработано в соответствии с законодательством об авторских правах и патентах, без нарушения соответствующих лицензий. В случае нарушения данного требования, полную ответственность за нарушение несет сторона, ответственная за разработку приложения.

### **5.1.6 Требования к масштабируемости и открытости**

Проект должен предоставлять возможность добавлять новую функциональность с минимальным изменением существующего кода.

### **5.1.7 Обработка ошибок**

В случае возникновения ошибок пользователь должен получать соответствующие сообщение об ошибке. Приложение должно поддерживать обработку следующих основных ошибок:

- Некорректный ввод данных;
- Системный сбой.

## **5.2 Требования к видам обеспечения**

### **5.2.1 Требования к программному обеспечению приложения**

Требования к программному обеспечению клиентской части:

Сайт должен быть доступен для использования на большинстве популярных браузеров:

- Microsoft Edge 110.0.1587.57;
- Google Chrome 110.0.5481.177;
- Mozilla Firefox 111.0;
- Yandex Browser 23.1.5.

Требования к программному обеспечению серверной части:

- Серверная часть приложения должна быть реализована на языке программирования Python с использованием фреймворка Django;
- В качестве СУБД для разработки необходимо использовать PostgreSQL.

В результате разработки данные требования могут расширяться.

### **5.2.2 Требования к лингвистическому обеспечению**

Приложение должно поддерживать русский язык. Язык ввода-вывода данных русский.



## **6 Структура приложения**

Приложение должно содержать следующие страницы/виды страниц:

- Главная страница;
- Страница входа;
- Страница приобретения лицензии;
- Страница загрузки виде;
- Страница истории проверок;
- Страница сотрудников;
- Страница складов.

## 7 Дизайн приложения

Приложение должно быть выполнено в минималистичном стиле.

Основные цвета приложения:

- Светло-серый цвет (#EDF2F6) – используется как цвет фона приложения;
- Синий цвет (#0C60A5) – используется как цвет бокового меню, цвет кнопок;
- Белый цвет (#FFFFFF) - используется как цвет фона карточки с информацией, цвет текста на синем фоне. Также используется в логотипе;
- Черный цвет (#000000) – используется как цвет текста на белом фоне.

При реализации приложения допускаются расхождения с данной цветовой палитрой, не нарушающие общий стиль приложения.

## **8 Навигация по приложению**

### **8.1 Основное навигационное меню**

Основным навигационным меню у авторизованного пользователя должно выступать боковое меню слева.

Боковое меню должно включать в себя следующие кнопки (для рабочего):

- Загрузка видео;
- Истории проверок.

Боковое меню должно включать в себя следующие кнопки (для аккаунта компании):

- Склады;
- Сотрудники;
- История проверок.

## **9 Описание страниц приложения**

### **9.1 Главная страница**

Данная страница доступна для всех пользователей. Содержит:

- Кнопка входа в аккаунт;
- Кнопку «Узнать тариф».

### **9.2 Страница входа**

Данная страница доступна для неавторизованных пользователей. Содержит:

- Поле ввода электронной почты пользователя;
- Поле ввода пароля;
- Кнопка войти.

### **9.3 Страница приобретения лицензии**

На страницу можно попасть по кнопке «Узнать тариф». По центру страницы описаны плюсы нашего приложения, сверху показана стоимость тарифа, снизу кнопка «Получить доступ».

### **9.4 Страница загрузки видео**

На страницу можно попасть по кнопке «Загрузить видео» в боковом меню. На странице можно загрузить видео и получить отчёт.

### **9.5 Страница историй проверок**

На страницу можно попасть по кнопки «История проверок» в боковом меню. На странице можно просмотреть старые загруженные видео и отчёты.

### **9.6 Страница сотрудников**

Эта страница доступна только аккаунту компании. На страницу можно попасть по кнопки «Сотрудники» в боковом меню. Эта страница

предоставляет пользователю возможность просмотра всех сотрудников компании.

Страница содержит следующий функционал:

- Создать аккаунт рабочего;
- Удалить аккаунт рабочего.

## **9.7 Страница складов**

Эта страница доступна только аккаунту компании. На страницу можно попасть по кнопке «Склады» в боковом меню. Эта страница предоставляет пользователю возможность просмотра всех складов компании.

Страница содержит следующий функционал:

- Создать новый склад;
- Добавить на склад рабочего;
- Удалить со склада рабочего.

## **9.8 Страница компаний**

Данная страница доступна только для администратора. На страницу можно попасть по кнопке «Компании» в боковом меню.

Страница содержит следующий функционал:

- Создать аккаунт компании;
- Удалить аккаунт компании.

## Приложение

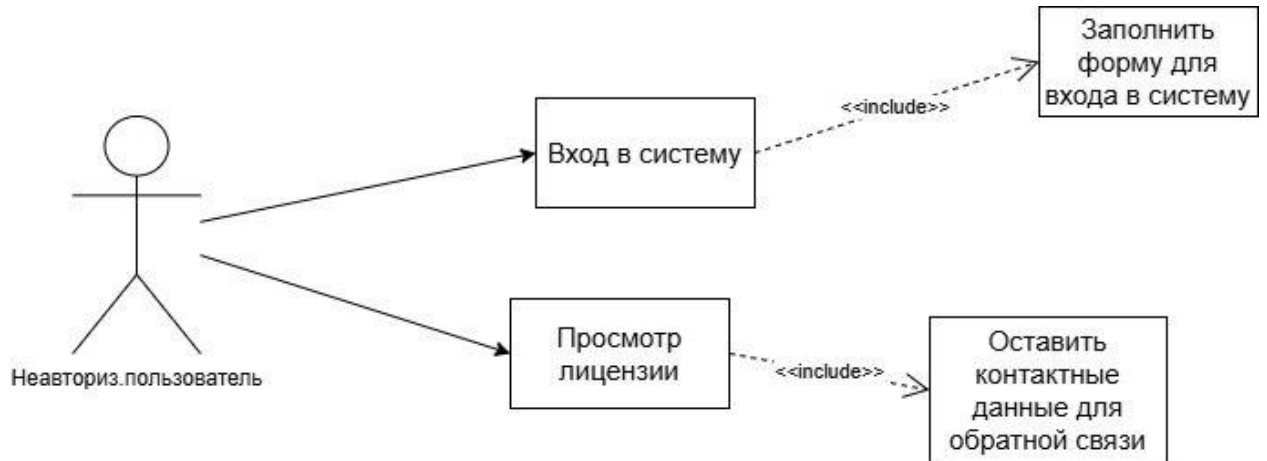


Рисунок 1 - Диаграмма прецедентов. Неавторизованный пользователь.



Рисунок 2 - Диаграмма прецедентов. Рабочий.

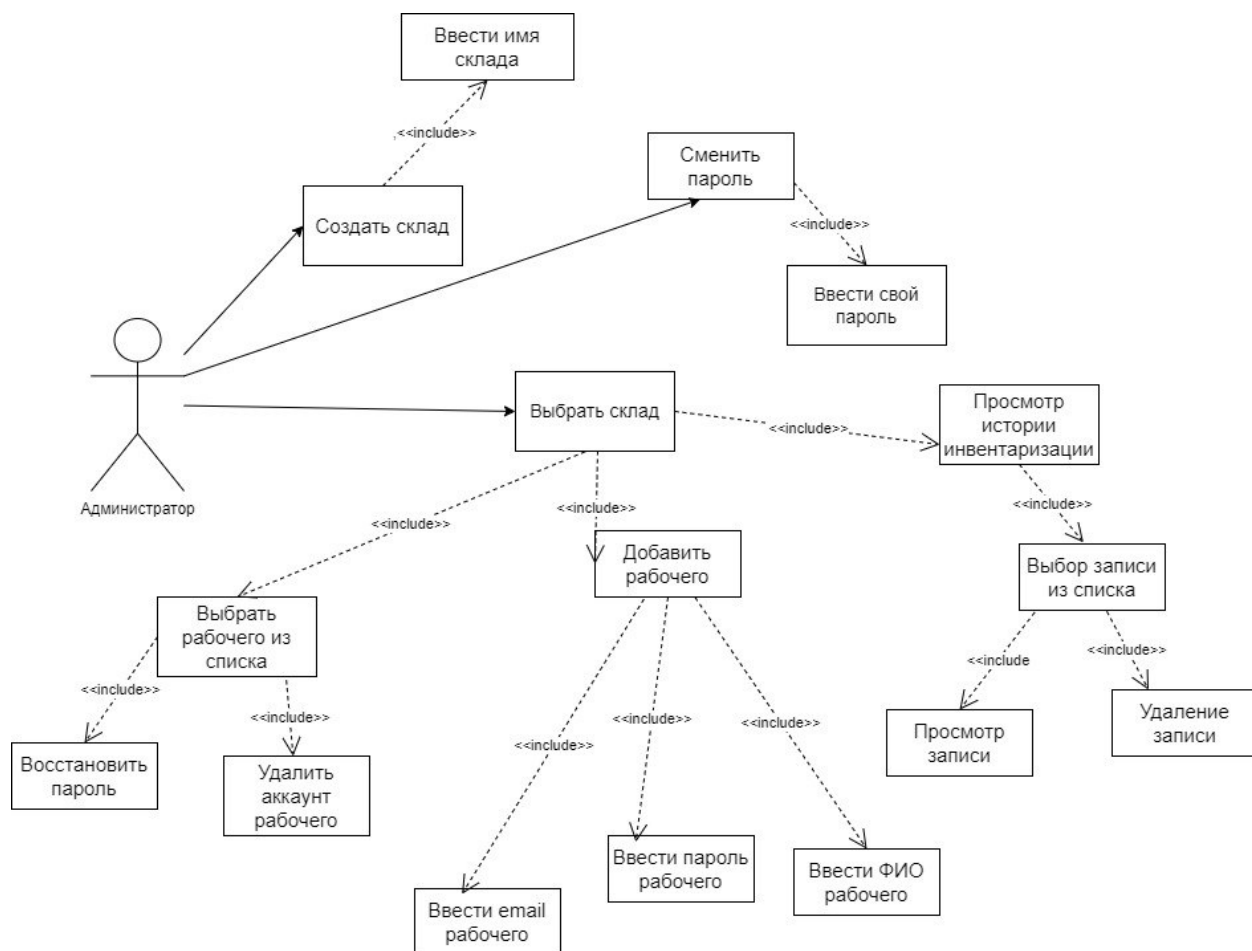


Рисунок 3 - Диаграмма прецедентов. Администратор.

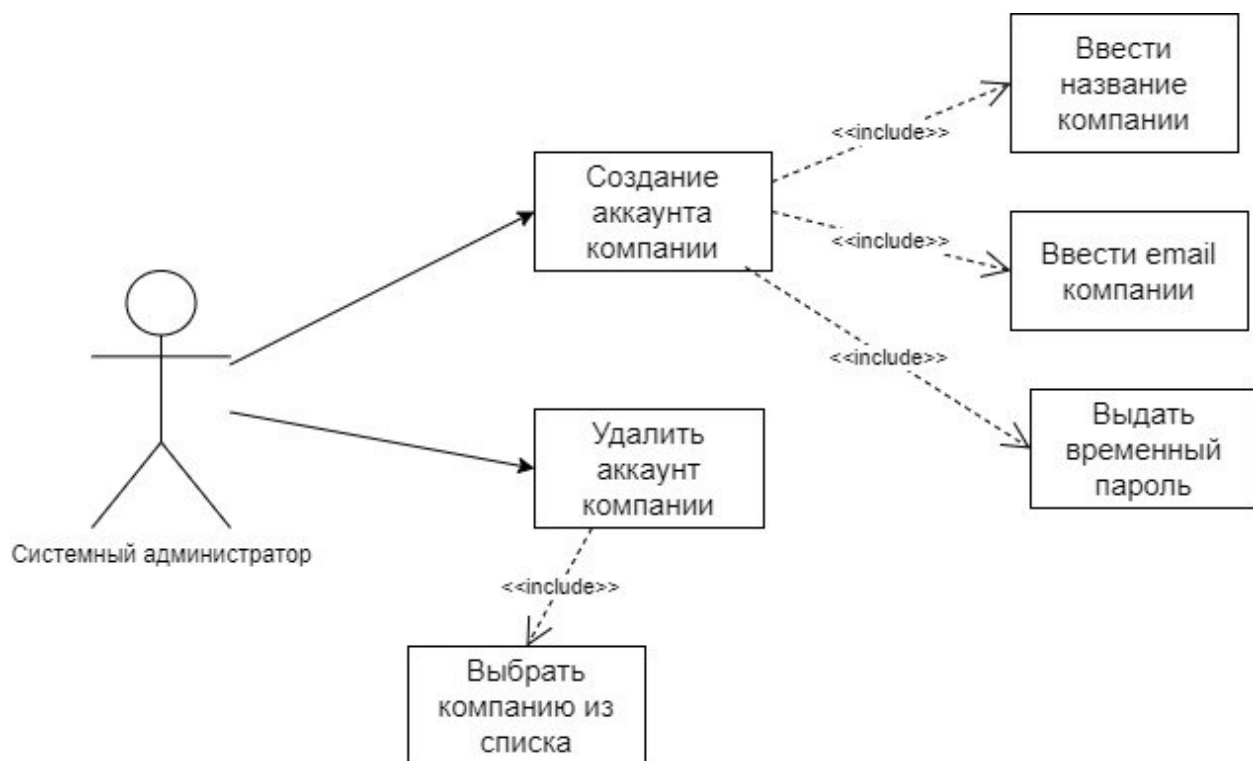


Рисунок 4 - Диаграмма прецедентов. Системный администратор.





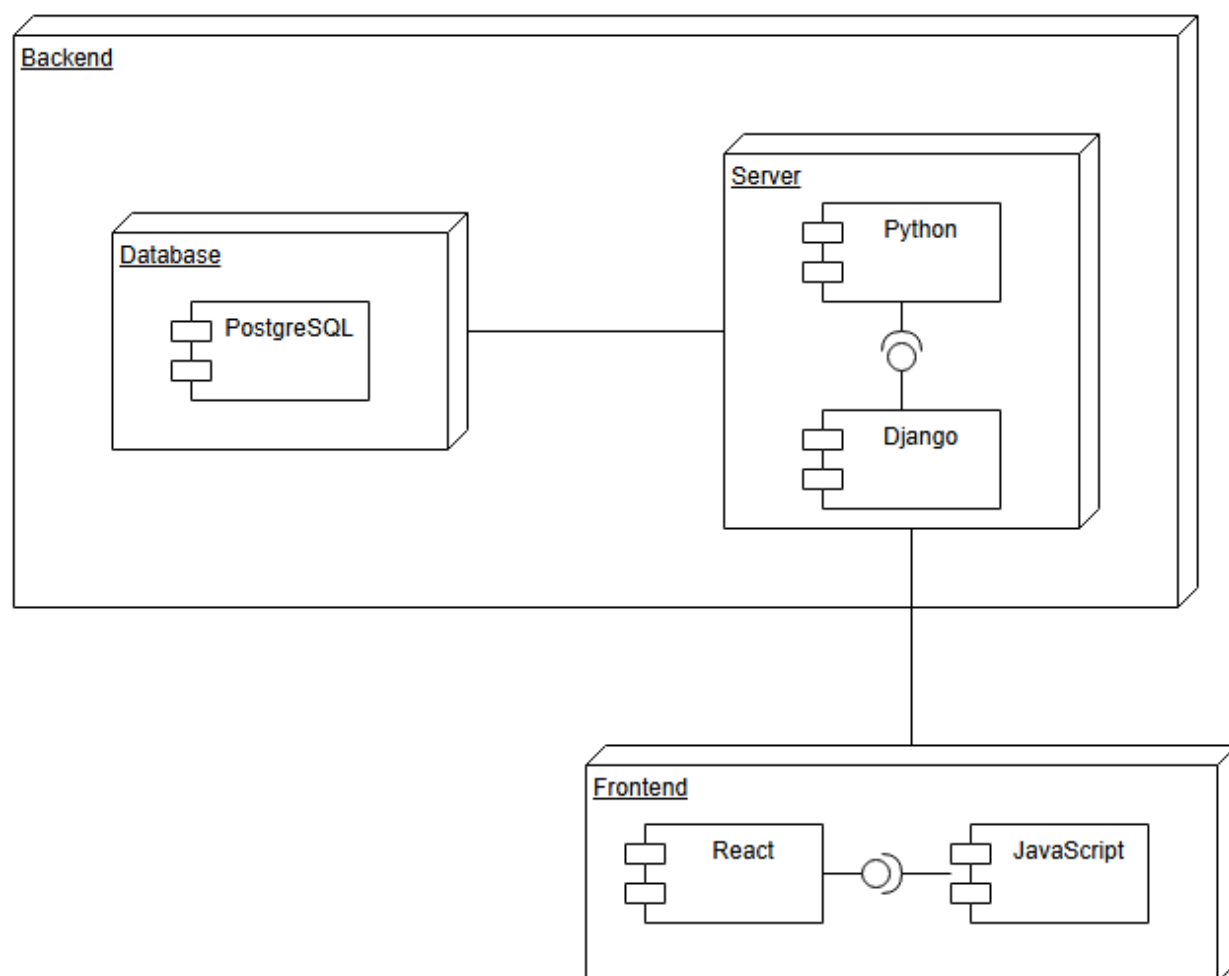


Рисунок 6 - Диаграмма развёртывания