



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

*НА ТЕМУ:*

База данных для транспортной системы завода

Студент

\_\_\_\_\_  
*подпись, дата*

Иванов В.А.

*фамилия, и.о.*

Руководитель курсовой работы

\_\_\_\_\_  
*подпись, дата*

Исаев А.Л.

*фамилия, и.о.*

2021 г.



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ  
Заведующий кафедрой ИУ7  
(Индекс)  
И.В. Рудаков  
(И.О.Фамилия)  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

## З А Д А Н И Е на выполнение курсового проекта

по дисциплине Базы данных

Студент группы ИУ7-62Б

Иванов Всеволод Алексеевич  
(Фамилия, имя, отчество)

Тема курсового проекта База данных для транспортной системы завода

Направленность КП (учебный, исследовательский, практический, производственный, др.)  
учебный

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

**Задание** Спроектировать и реализовать базу данных для транспортной системы завода. С помощью Web-приложения реализовать интерфейс для работы с информацией о заказах, записях о поездках, сотрудниках, дежурствах путём взаимодействия с базой данных. Также реализовать функционал для разных категорий пользователей

### Оформление курсового проекта:

Расчетно-пояснительная записка на 20-30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку задачи, введение, аналитическую, конструкторскую, технологическую части, заключение, список литературы.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.): на защиту проекта должна быть предоставлена презентация, состоящая из 15-20 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, диаграммы, интерфейс.

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 2021 г.

Руководитель курсового проекта

А.Л. Исаев  
(Подпись, дата) (И.О.Фамилия)

Студент

В.А. Иванов  
(Подпись, дата) (И.О.Фамилия)

## Реферат

Суть курсового проекта заключается в реализации базы данных для транспортной системы завода и web-приложения для предоставления интерфейса доступа к ней. В результате выполнения работы было получено программное обеспечение предоставляющее возможность планировать, отслеживать и регулировать работу службы доставки.

Приложение реализовано с использованием фреймворка Django на языке программирования Python 3. В качестве СУБД был использован PostgreSQL.

**Ключевые слова:** доставка, транспортная система, завод, web-приложение, PostgreSQL, Django.

РПЗ содержит 38 страниц, 12 таблиц, 23 иллюстрации, 9 ссылок на используемые источники.

# Оглавление

<b>Введение</b>	<b>6</b>
<b>1 Аналитическая часть</b>	<b>7</b>
1.1 Постановка задачи . . . . .	7
1.2 Формализация данных . . . . .	8
1.3 Формализация ролей . . . . .	9
1.4 Базы данных . . . . .	9
1.4.1 Модели данных . . . . .	10
1.4.2 СУБД . . . . .	11
<b>2 Конструкторская часть</b>	<b>12</b>
2.1 Диаграмма вариантов использования . . . . .	12
2.2 Проектирование базы данных . . . . .	13
2.2.1 Сущности базы данных . . . . .	13
2.2.2 Проектирование таблиц . . . . .	14
2.2.3 Нормализация схем отношений . . . . .	17
2.2.4 Схемы триггеров . . . . .	18
2.3 Архитектура приложения . . . . .	20
<b>3 Технологическая часть</b>	<b>22</b>
3.1 Выбор средств программной реализации . . . . .	22
3.1.1 Язык программирования . . . . .	22
3.1.2 СУБД и ORM . . . . .	22
3.1.3 Web-фреймворк . . . . .	23
3.2 UML-диаграммы компонентов приложения . . . . .	24
3.2.1 Компонент доступа к данным . . . . .	24
3.2.2 Компонент бизнес-логики . . . . .	25
3.2.3 Компонент представления . . . . .	26
3.2.4 Диаграмма приложения . . . . .	27

3.3	Реализация базы данных . . . . .	28
3.3.1	Создание таблиц . . . . .	28
3.3.2	Реализация триггеров . . . . .	28
3.3.3	. . . . .	28
3.4	Интерфейс приложения . . . . .	28
<b>Заключение</b>		<b>36</b>
<b>Список литературы</b>		<b>37</b>

## Введение

Одним из ключевых аспектов эффективного ведения любого крупного или среднего бизнеса в настоящее время является хорошая связь между его сотрудниками. Быстрый и удобный доступ к данным позволяет оперативно координировать всех работников и минимизировать временные издержки по передаче текущих задач. Поэтому, актуальным направлением в современном мире является разработка сервисов, реализующих данные возможности. Объектом разработки данного курсового проекта выбрана транспортная система завода, которая должна выполнять доставку заказов.

**Целью** данного курсовой работы является разработка базы данных для транспортной системы завода и web-приложения для доступа к ней.

Выделены следующие задачи курсового проекта:

- формализовать задание, определить необходимый функционал;
- для структурированного хранения информации спроектировать базу данных;
- проанализировать существующие СУБД и обосновать выбор одной из них;
- реализовать базу данных и интерфейс доступа к ней с использованием выбранной СУБД;
- обосновать выбор web-фреймворка;
- реализовать web-приложение для выделенного функционала.

## **1. Аналитическая часть**

### **1.1. Постановка задачи**

Необходимо разработать базу данных, которая будет хранить информацию о сотрудниках, заказах, записях, дежурствах и о проездах машин через контрольно-пропускные пункты, а также web-приложение для предоставления этих данных.

Также должны быть реализованы различные категории сотрудников, для которых будет предоставляться определённый функционал по работе с данными транспортной системы. Для этого должна существовать возможность регистрации аккаунтов и дальнейшего взаимодействия с ними.

## 1.2. Формализация данных

В соответствии с предметной областью, соответствующей описаным набором требований, база данных должна хранить сущности, описанные ER-диаграммой на рисунке 1.1

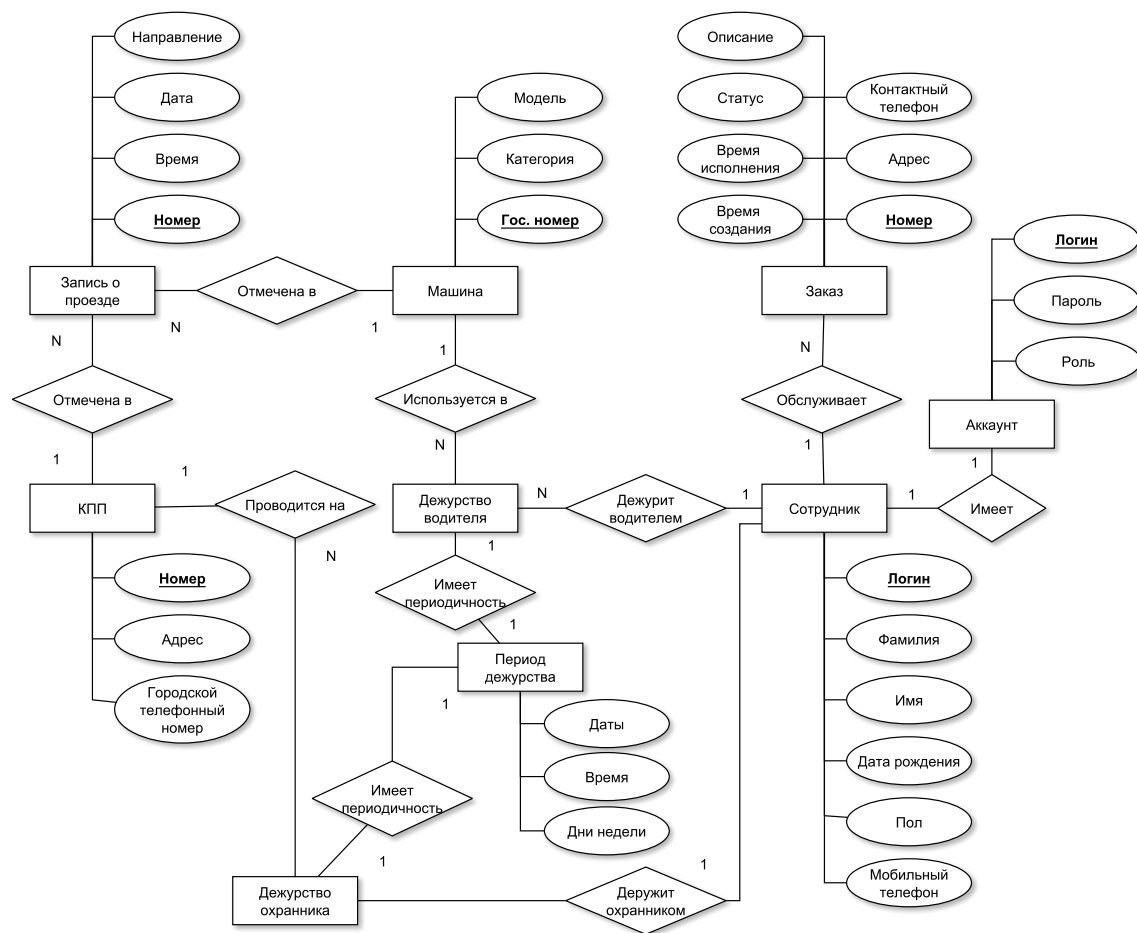


Рис. 1.1 — ER-диаграмма сущностей



### 1.3. Формализация ролей

Для функционирования транспортной системы были выделено несколько должностей сотрудников, названия и функционал которых приведены в таблице 1.1.

Таблица 1.1 — Роли сотрудников и их функционал

Должность	Функционал
Администратор	Назначение дежурств и заказов, рассмотрение заявок на регистрацию, создание новых сущностей: машин, КПП, заказов
Водитель	Выбор заказа, выполнение заказа, просмотр дежурств и личной информации
Охранник	Создание записей о проезде, просмотр дежурств и личной информации
Неподтверждённый сотрудник	Просмотр личной информации
Неавторизованный пользователь	Регистрация, авторизация

Для упрощения регистрации новых сотрудников в системе принято решение предоставить им возможность заполнения заявок, которые в дальнейшем могут быть подтверждены администратором.

### 1.4. Базы данных

База данных — это совокупность взаимосвязанных структурированных данных, относящихся к определенной предметной области и организованных так, чтобы обеспечить независимость данных от программ обработки. [1]

### 1.4.1. Модели данных

Модель данных - это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Эти объекты позволяют моделировать структуру данных, а операторы — поведение данных. [2]

Существует три основные модели данных.

- Иерархическая. База данных представляется в виде древовидной структуры, состоящей из объектов различных уровней. Объекты находятся в отношении предка к потомку, причём объект может иметь только одного предка и любое количество потомков.
- Сетевая. Отличием от иерархической модели данных является отсутствие ограничения на количество предков. База данных в такой модели состоит из набора экземпляров записи и экземпляров связей между записями.
- Реляционная. Основной идеей является то, что все наборы данных представляются в виде множеств. База данных состоит из множества двумерных таблиц. Таблицы состоят из записей и полей, каждая запись имеет собственные значения для каждого поля. Иерархия элементов отсутствует.

Реляционная модель в настоящее время является наиболее гибкой и удобной в использовании, а также обладает наибольшим выбором в области систем управления базой данных, поэтому принято решение о её использовании в данной работе.

### 1.4.2. СУБД

Система управления базами данных (сокращённо СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных [1].

В процессе разработки было принято решение использовать СУБД для взаимодействия с базой данных, так как оно реализует множество функций, которые потребуются в работе с базой данных.

### Вывод

Результатом аналитического раздела стала постановка задачи, формализация данных программы и ролей пользователей, описание способов хранения данных и управления ими.



## 2.2. Проектирование базы данных

### 2.2.1. Сущности базы данных

На рисунке 2.2 представлена ER-диаграмма сущностей базы данных.

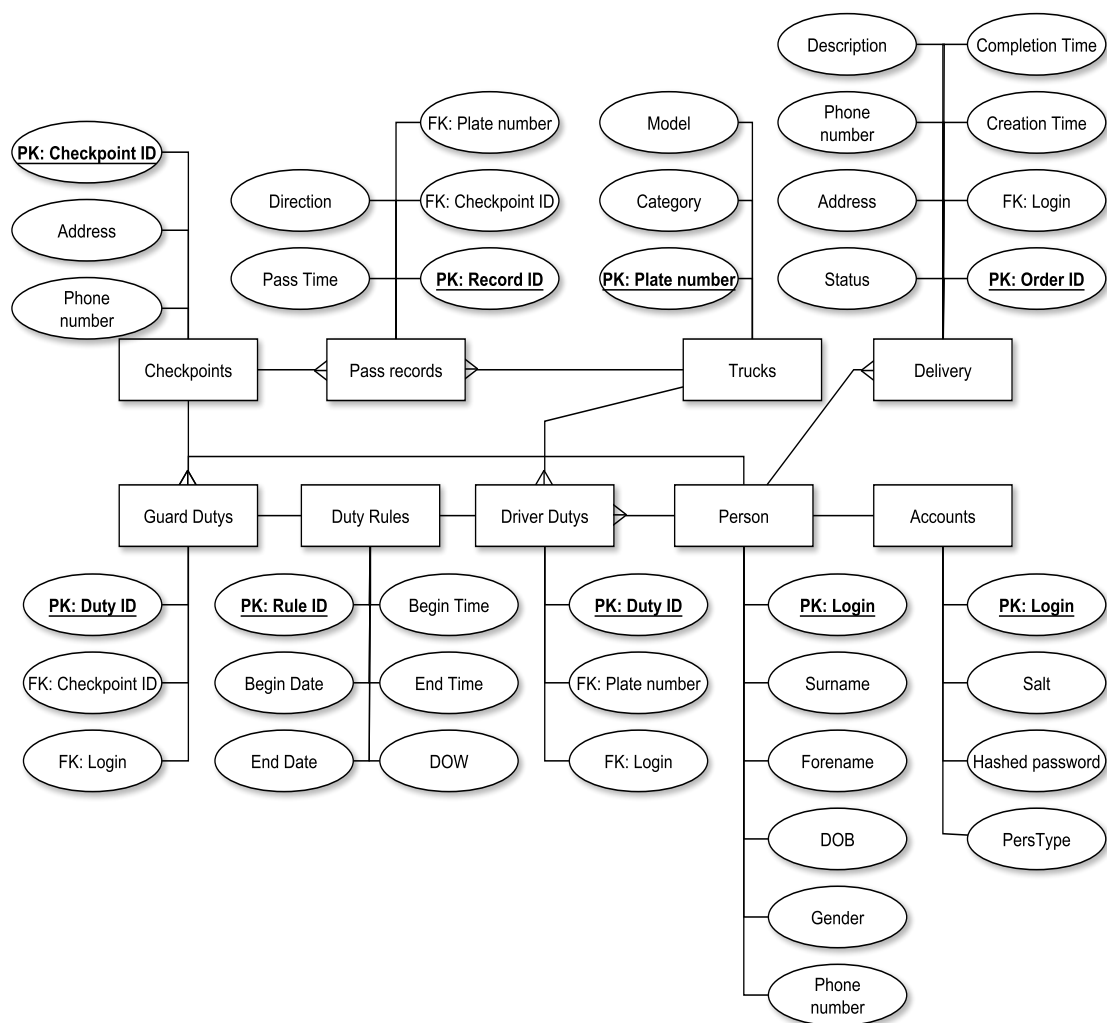


Рис. 2.2 — ER-диаграмма сущностей базы данных

### 2.2.2. Проектирование таблиц

На основании выделенных сущностей база данных должна содержать таблицы, описание которых приведено в таблицах 2.1-2.8

Для обеспечения приватности пароля к аккаунту было принято решение использовать хэширование. В качестве ролей используются значения admin, driver, guard (соотв. администратор, водитель, охранник).

Таблица 2.1 — Accounts (таблица аккаунтов)

Атрибут	Тип	Комментарий
Login	Строка	Логин аккаунта, РК
PersType	Строка	Роль аккаунта
Salt	Строка	"Соль" для хэширования пароля
HashedPassword	Строка	Хэшированный пароль

Таблица 2.2 — Person (таблица личной информации)

Атрибут	Тип	Комментарий
Login	Строка	Логин аккаунта, РК, FK
Surname	Строка	Фамилия сотрудника
Forename	Строка	Имя сотрудника
DOB	Дата	Дата рождения сотрудника
Gender	Строка	Пол сотрудника
PhoneNumber	Строка	Номер мобильного телефона сотрудника

В качестве статуса заказа используются значения not\_assigned, in\_transit, delivered (соотв. заказ не назначен водителю, заказ в процессе доставки и заказ доставлен).

Таблица 2.3 — Trucks (таблица машин)

Атрибут	Тип	Комментарий
PlateNumber	Строка	Гос. номер машины, РК
Category	Строка	Марка машины
Model	Строка	Модель машины

Таблица 2.4 — Delivery (таблица заказов)

Атрибут	Тип	Комментарий
OrderID	Целое число	Уникальный номер заказа, РК
Login	Строка	Логин водителя, доставляющего или доставившего заказ, FK
Address	Строка	Адрес заказчика
PhoneNumber	Строка	Контакты заказчика
Status	Строка	Статус заказа
Description	Строка	Содержание заказа
CreationTime	Дата, время	Дата и время создания заказа
CompletionTime	Дата, время	Дата и время завершения заказа

Таблица 2.5 — Checkpoints (таблица КПП)

Атрибут	Тип	Комментарий
CheckpointID	Целое число	Номер КПП, РК
Address	Строка	Адрес КПП
PhoneNumber	Строка	Номер телефона КПП

Дежурства как охранников, так и водителей имеют циклический характер по неделе. Поэтому рациональнее хранить не каждое дежурство отдельно, а правило, содержащее диапазон дат и дни недели, в которые сотрудник дежурит. Атрибут дней недели в таблице хранит строку, определяющую дни по номеру (например стро-

ка 013 соответствует дежурству по понедельникам, вторникам и четвергам). Период (или расписание) дежурства выделено в отдельную таблицу.

Таблица 2.6 — DutyRules (таблица расписаний дежурств)

Атрибут	Тип	Комментарий
RuleID	Целое число	Номер периода дежурства, РК
BeginDate	Дата	Дата начала периода дежурства
EndDate	Дата	Дата окончания периода дежурства
BeginTime	Время	Время начала дежурства
EndTime	Время	Время окончания дежурства
DOW	Строка	Дни недели дежурства

Таблица 2.7 — GuardDutys (таблица дежурств охранников)

Атрибут	Тип	Комментарий
DutyID	Целое число	Номер дежурства, РК
Login	Строка	Логин дежурящего охранника, FK
CheckpointID	Целое число	Номер КПП, FK
RuleID	Целое число	Номер периода дежурства, FK

Таблица 2.8 — DriverDutys (таблица дежурств водителей)

Атрибут	Тип	Комментарий
DutyID	Целое число	Номер дежурства, РК
Login	Строка	Логин дежурящего водителя, FK
PlateNumber	Строка	Гос. номер машины, FK
RuleID	Целое число	Номер периода дежурства, FK

В качестве направления проезда используются значения in, out (соотв. въезд и выезд с территории предприятия).



Таблица 2.9 — PassRecords (таблица записей проездов)

Атрибут	Тип	Комментарий
RecordID	Целое число	Номер записи, РК
PlateNumber	Строка	Гос. номер машины, FK
CheckpointID	Целое число	Номер КПП, FK
PassTime	Дата, время	Дата и время проезда
Direction	Строка	Направление проезда

Также было принято создать таблицу для хранения информации о событиях в базе данных.

Таблица 2.10 — LogActions (таблица активности пользователей)

Атрибут	Тип	Комментарий
Actor	Строка	Роль, осуществившая действие
ActTime	Дата, время	Время действия
Description	Строка	Описание действия

### 2.2.3. Нормализация схем отношений

Распространённым подходом в проектировании баз данных является метод нормализации, служащий для устранения избыточности и упрощения контроля целостности. При проектировании данной базы данных была поставлена цель соответствия третьей нормальной форме, что является достаточным для выполнения описанных целей нормализации[3].

В разработанной модели данных переменная отношения находится в первой нормальной форме, так как выполняется условие атомарности. В каждом кортеже, соответствующем некоторой опи-

санной таблице, каждый компонент является атомарным, то есть содержит только одно значение для каждого из атрибутов[4]. Данное утверждение основанно на том, что каждый атрибут имеет простой тип данных.

Также созданная переменная отношения находится во второй нормальной форме, так как в каждой таблице любой не ключевой атрибут полностью зависит от первичного ключа. Для перехода в данную форму из таблиц дежурств охранников и водителей была отдельно выделена таблица расписаний дежурств.

Можно утверждать, что спроектированная переменная отношения находится в третьей нормальной форме. Во всех таблицах не ключевые атрибуты не связаны отношениями между собой, поэтому можно утверждать, что они зависят от первичного ключа нетранзитивно, что и является условием нахождения в третьей нормальной форме.

#### **2.2.4. Схемы триггеров**

В соответствии с описанной ранее схемой взаимодействия, существуют некоторые действия, которые могут выполнять сразу несколько ролей. Для сбора дополнительной информации о данных действиях требуется реализовать следующие триггеры.

- Триггер сохранения информации добавления записи о проезде. Данное действие может осуществлять как охранник, так и администратор, но таблица PassRecords не хранит информации об авторе, поэтому было решено создать данный триггер. Его схема представлена на рисунке 2.3

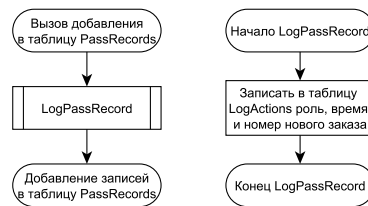


Рис. 2.3 — Схема алгоритма триггера на добавление в таблицу PassRecords

- Триггер сохранения информации обновления заказа. В данном случае действие могут совершать водитель и администратор путём изменения статуса заказа и установлением времени его доставки. Его схема представлена на рисунке 2.4

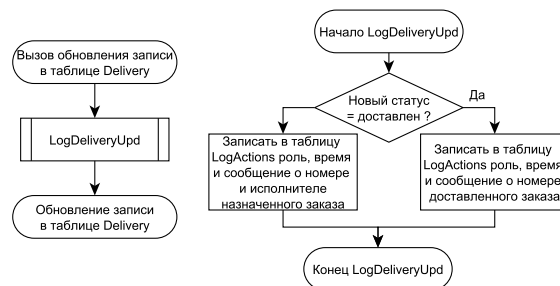


Рис. 2.4 — Схема алгоритма триггера на обновление в таблице Delivery

Также в системе нельзя допускать удаление аккаунтов действующих сотрудников, так как с ними связана некоторая информация. Например, для доставленных заказов обязательно требуется хранить информацию о его курьере. Поэтому также был создан триггер на удаление записей из таблицы Accounts, запрещающий удалять сотрудников с активным статусом. Схема данного триггера представ-

лена на рисунке 2.5

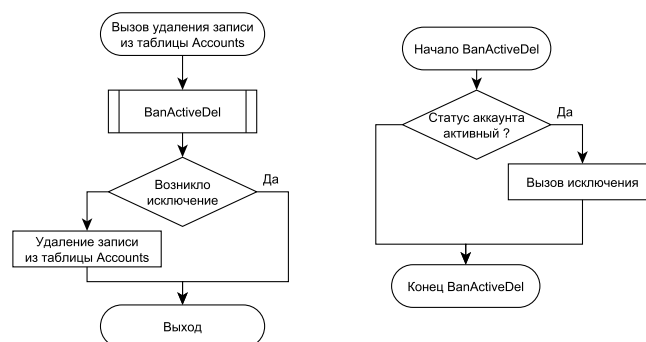


Рис. 2.5 — Схема алгоритма триггера на удаление из таблицы Accounts

## 2.3. Архитектура приложения

Наиболее известным подходом к проектированию архитектуры web-приложения является шаблон MVC. Данный паттерн предполагает разделение приложения на три описанные части.

- Model (модель) - компонент бизнес-логики, отвечает за взаимодействие с базой данных и основную обработку информации.
- View (представление) - компонент, отвечающий за отображения данных, полученных в результате работы модели.
- Controller (контроллер) - компонент, отвечающий за получение запроса от пользователя, передачу их модели для дальнейшего обновления представления.

В данной курсовой работе будет использован описанный подход, так как он чётко разграничивает зоны каждого компонента, что позволяет сделать их независимыми.

## **Вывод**

Результатом конструкторской части стала разработка сценариев использования приложения, его архитектуры и проектирование таблиц и триггеров базы данных.

## 3. Технологическая часть

### 3.1. Выбор средств программной реализации

#### 3.1.1. Язык программирования

В качестве языка программирования был выбран Python 3[7], по следующим причинам.

- Поддержка ООП, требуемого для архитектурного шаблона MVC.
- Наличие опыта работы с данным языком.
- Данный язык является популярным в разработке web-приложений, поэтому он обладает большим количеством web-фреймворков и библиотек для доступа к СУБД.

#### 3.1.2. СУБД и ORM

Наиболее популярными реляционными СУБД являются Oracle, MySQL и PostgreSQL[5]. Их сравнение[6] можно свести в таблицу 3.1

Таблица 3.1 — Сравнение реляционных СУБД

СУБД	Преимущества	Недостатки
Oracle	+ надёжность системы + поддержка современности функционала	- конечная стоимость СУБД - высокие системные требования
MySQL	+ прост в установке и использовании + производительность + широкий базовый функционал	- проблемы с надёжностью - не полная поддержка SQL
Продолжение на следующей странице		

Таблица 3.1 – продолжение

СУБД	Преимущества	Недостатки
PostgreSQL	+ лёгкая масштабируемость + открытость ПО + поддержка текстовых форматов (в т.ч. json)	- низкая скорость выполнения пакетных операций - сложности поиска хостинга

Наиболее подходящим для небольших организаций, которые являются целевыми для разрабатываемой программы, является PostgreSQL[6]. Учитывая наличие опыта работы с данным средством, оно был выбран в качестве СУБД для данного курсового проекта.

В качестве ORM (технология объектно-реляционного преобразования) был выбран реeewe[8], так как она является простой в освоении и использовании, а также позволяет легко подменять СУБД.

### 3.1.3. Web-фреймворк

В качестве web-фреймворка был выбран Django[9], по следующим причинам.

- Django использует шаблон проектирования MVC.
- Лёгкая масштабируемость.
- Готовые решения для наиболее востребованных задач.
- Поддержка шаблонизации html страниц.

## 3.2. UML-диаграммы компонентов приложения

### 3.2.1. Компонент доступа к данным

В данной работе компонент доступа к данным реализован с использованием паттерна проектирования Repository. UML диаграмма компонента изображена представлена на рисунке 3.1

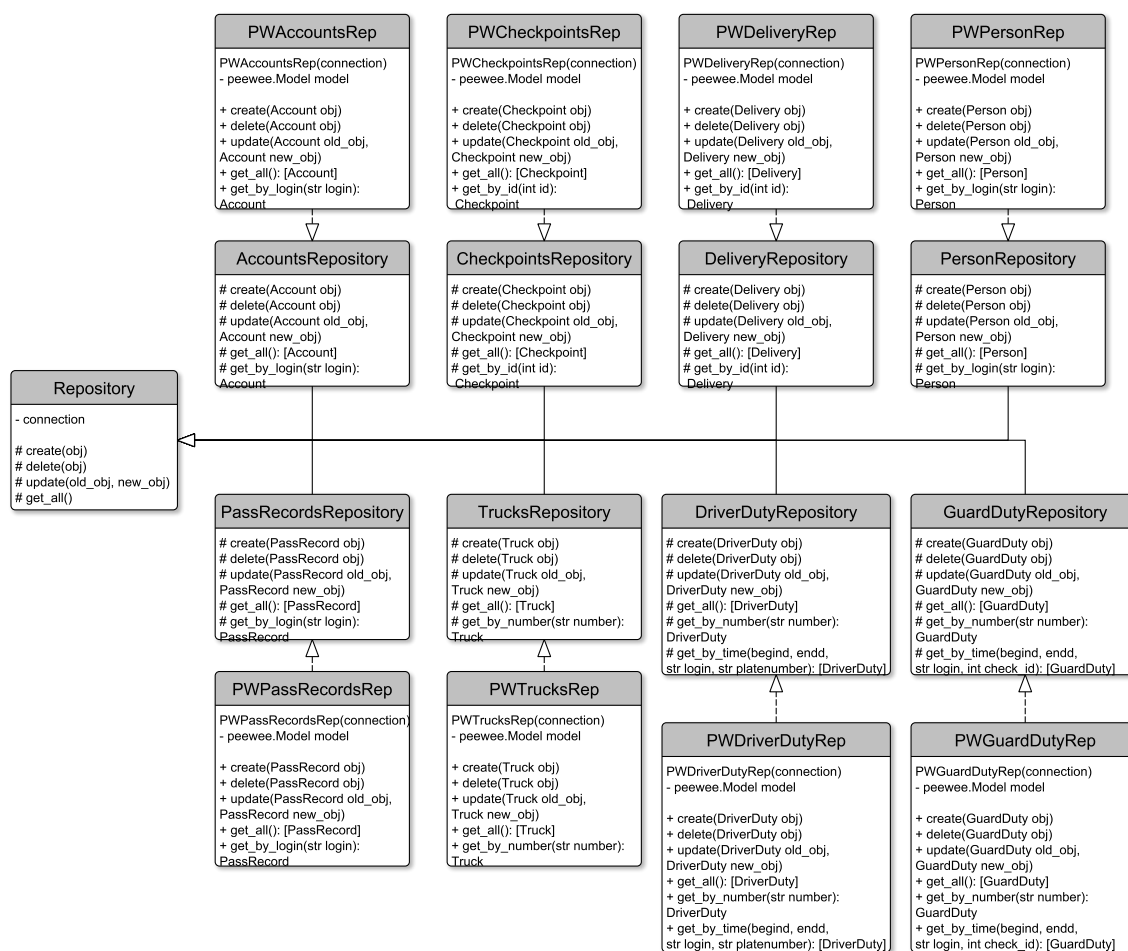


Рис. 3.1 — UML-диаграмма компонента доступа к данным



### 3.2.2. Компонент бизнес-логики

В соответствии с подходом MVC был создан компонент бизнес-логики, выполняющий основную обработку данных, UML диаграмма которого представлена на рисунке 3.2

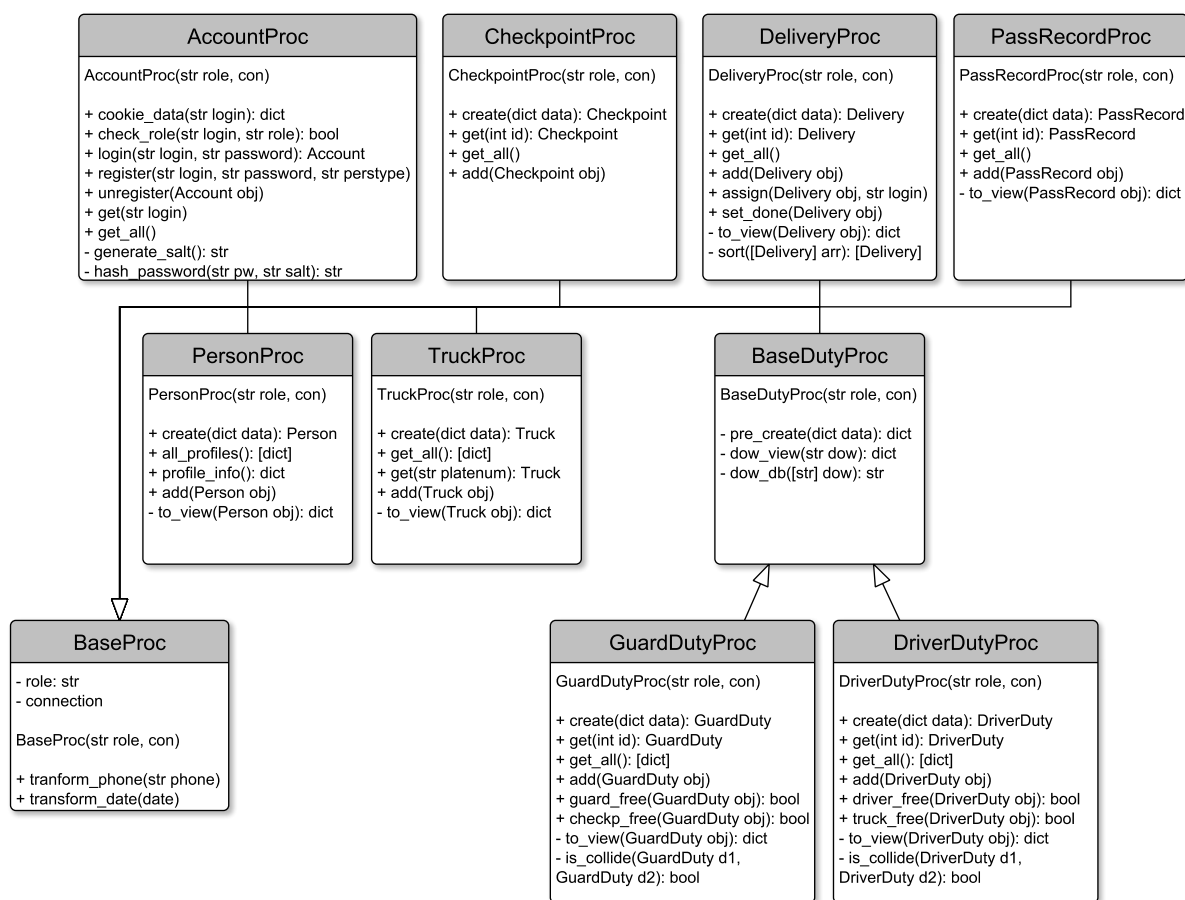


Рис. 3.2 — UML-диаграмма компонента бизнес-логики

### 3.2.3. Компонент представления

Также создан компонент представления, выполняющий отображение web-страниц в ответ на запросы пользователя, UML диаграмма которого представлена на рисунке 3.3. Помимо этого был реализован технический компонент представления, отображающий информацию в символьном виде, для возможности тестирования компонента бизнес-логики.

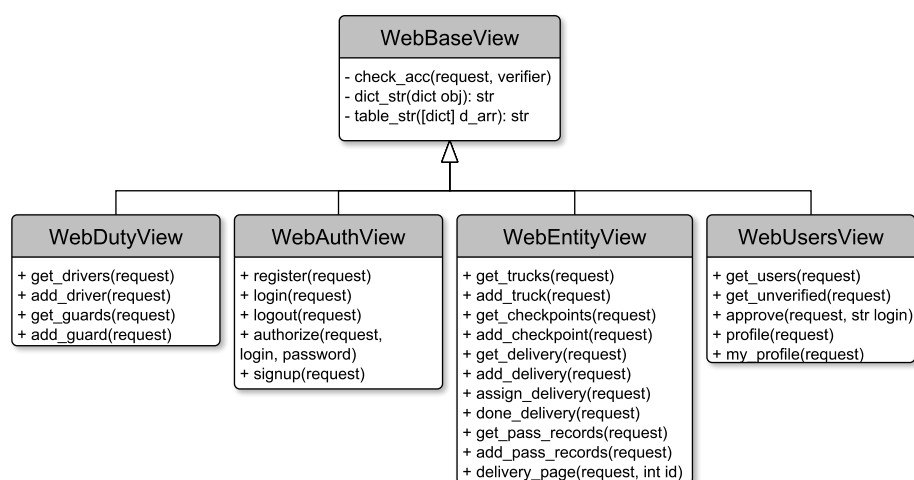


Рис. 3.3 — UML-диаграмма компонента web-представления

### 3.2.4. Диаграмма приложения

Все перечисленные компоненты можно объединить в одну UML-диаграмму всего приложения на рисунке 3.4

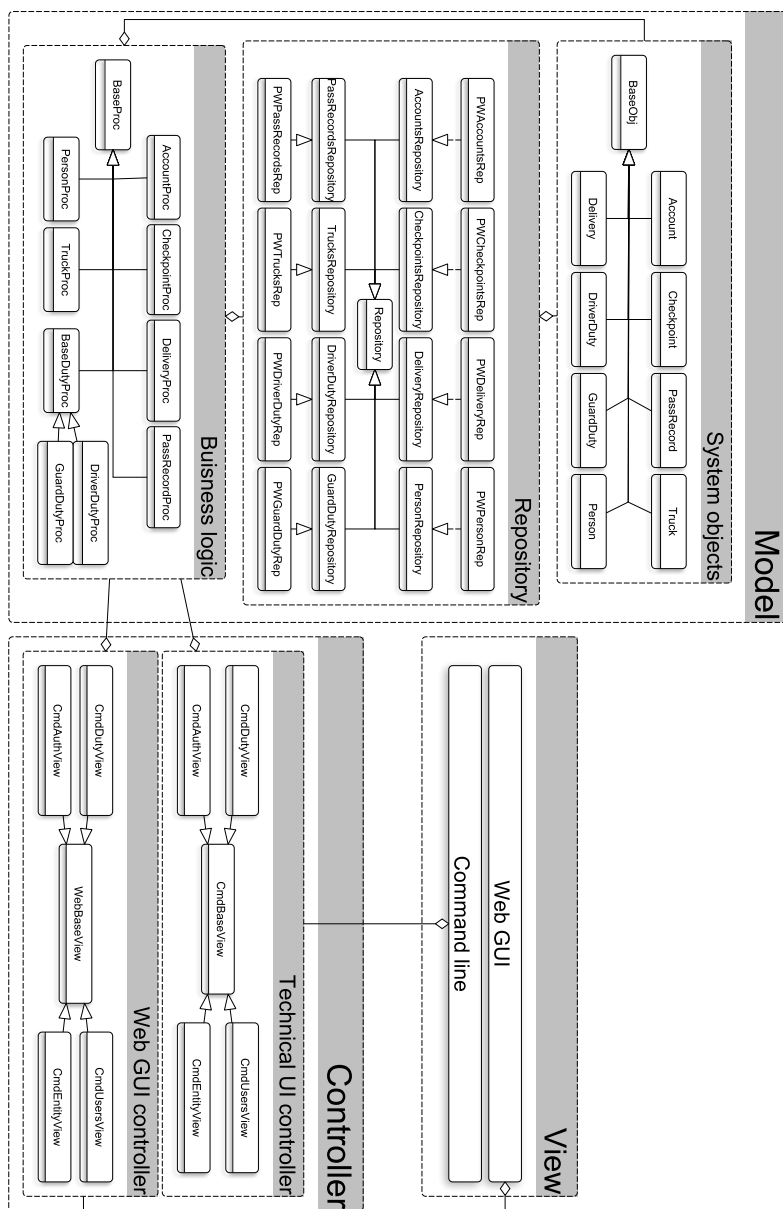


Рис. 3.4 — UML-диаграмма компонента web-представления

### **3.3. Реализация базы данных**

#### **3.3.1. Создание таблиц**

#### **3.3.2. Реализация триггеров**

#### **3.3.3.**

### **3.4. Интерфейс приложения**

Для авторизованного пользователя в заголовке каждой страницы содержится панель навигации, предоставляющая возможность перейти на все страницы, функционально соответствующие его роли. Данная панель изображена на примере администратора на рисунке 3.5. Некоторые пункты меню являются выпадающими, их можно открыть наведением мыши. Также в левом углу панели присутствует элемент информационного или ошибочного сообщения.

На рисунках 3.5-3.11 приведена демонстрация ключевых страниц для роли администратора. На рисунках 3.12, 3.13 и 3.14, 3.15 приведены страницы, уникальные для ролей водителя и охранника соответственно.

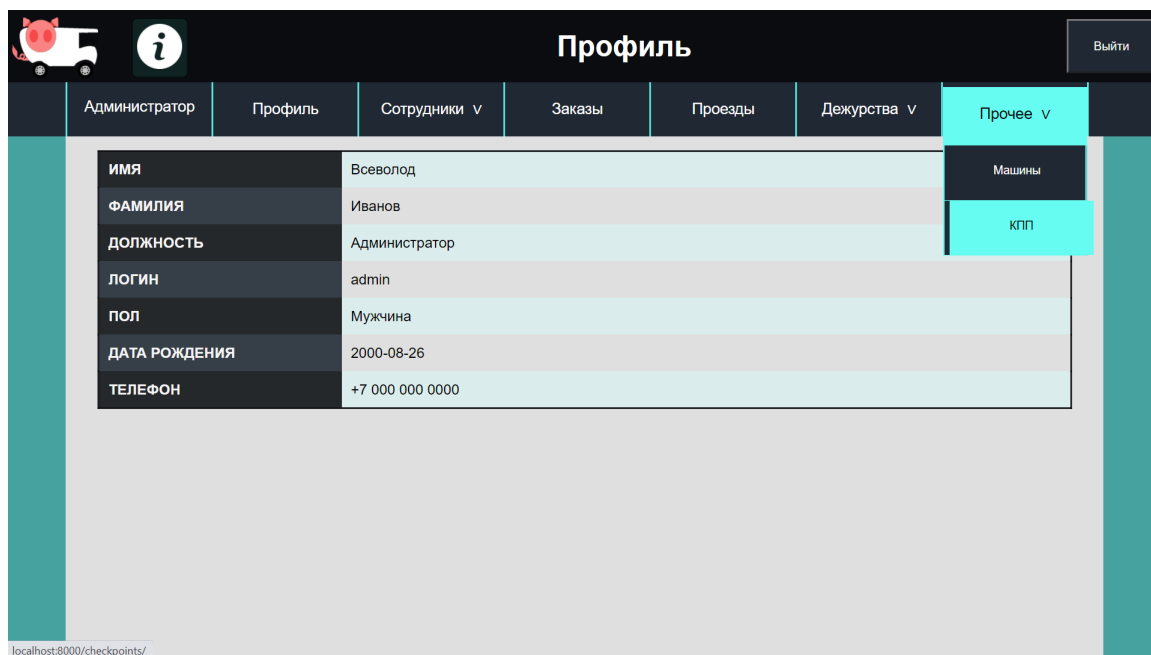



Рис. 3.5 — Страница профиля администратора

Для неавторизованного пользователя доступны страницы входа в аккаунт и регистрации (рисунок 3.16 и 3.17)

## Вывод

Результатом технологической части стал выбор средств программной реализации, реализация и описание структуры базы данных и приложения, визуальная демонстрация интерфейса приложения.



Сотрудники

Выйти

Администратор

Профиль

Сотрудники v

Заказы


Проезды

Дежурства v

Прочее v

Должность	Имя	Фамилия	Логин
Администратор	Иванов	Всеволод	admin
Водитель	Гослинг	Райан	driver
Водитель	Техников	Павел	driver2
Охранник	Золотова	Роза	guard
Охранник	Боросов	Филипп	guard2002

Рис. 3.6 — Страница просмотра всех сотрудников



Неподтверждённые сотрудники

Выйти

Администратор

Профиль

Сотрудники v

Заказы


Проезды

Дежурства v

Прочее v

предполагаемая должность	фамилия	имя	логин	действие
Водитель	Рокатански	Макс	mad_me	<div>✓</div> <div>✗</div>

Рис. 3.7 — Страница просмотра заявок регистрации



Заказы

Выйти

Администратор

Профиль

Сотрудники v

Заказы

Проезды

Дежурства v

Прочее v

НОМЕР	СТАТУС	КУРЬЕР	АДРЕС	КОНТАКТЫ
7	Не назначен	Не назначен	проспект Мира, дом 21, корпус 5	+7 918 241 2441
5	В доставке	driver	проезд Кукушкина, дом 21	+7 412 224 1241
4	Доставлен	driver2	проезд Кукушкина, дом 21	+7 412 224 1241
3	Доставлен	driver	ул. Хихишина, дом 123, корпус 4	+7 498 312 1412
2	Доставлен	driver	проезд Колотушкина, дом 21	
1	Доставлен	driver2	улица Сталеваров, дом 102	

Адрес

ул. Пушкина, 9

Телефон

+7 000 123 4567


Описание

Опишите заказ

Добавить

Закрыть

Рис. 3.8 — Страница просмотра и создания заказов



Записи о проездах

Выйти

Администратор

Профиль

Сотрудники v

Заказы

Проезды

Дежурства v

Прочее v

МАШИНА	ВОДИТЕЛЬ	КПП	ВРЕМЯ	НАПРАВЛЕНИЕ
M010EP077	driver2	3	03.06.2021 23:03:05	Въезд <input checked="" type="checkbox"/>
M010EP077	driver2	3	03.06.2021 22:53:33	<input type="checkbox"/> Выезд
M010EP077	-	3	03.06.2021 16:42:15	Въезд <input checked="" type="checkbox"/>
M000MM003	-	4	03.06.2021 16:10:28	<input type="checkbox"/> Выезд
M010EP077	-	5	31.05.2021 11:47:34	<input type="checkbox"/> Выезд
M010EP077	-	3	31.05.2021 11:22:17	Въезд <input checked="" type="checkbox"/>

Создать запись

Рис. 3.9 — Страница просмотра и регистрации записей о проездах

## Дежурства водителей

Выйти

Администратор
Профиль
Сотрудники v
Заказы
Проезды
Дежурства v
Прочее v

МАШИНА	ВОДИТЕЛЬ	ПЕРИОД	ВРЕМЯ	ДНИ РАБОТЫ
Текущие дежурства				
M010EP077	driver	с 30.05.2021 по 30.06.2021	09:00 - 18:00	
Все дежурства				
M010EP077	driver2	с 09.07.2021 по 21.07.2021	09:00 - 18:00	
M000MM003	driver	с 23.08.2021	09:00 - 18:00	
M010EP077	driver2	с 03.06.2021 по 11.06.2021	17:30 - 23:55	
M010EP077	driver	с 30.05.2021 по 30.06.2021	09:00 - 18:00	

Номер машины

Водитель

Период дежурства

 - 

☐ Бессрочно

Время дежурства

 - 

Рабочие дни

☐ ПН
 ☐ ВТ
 ☐ СР
 ☐ ЧТ
 ☐ ПТ
 ☐ СБ
 ☐ ВС

Добавить

Закрыть

Рис. 3.10 — Страница просмотра и назначения дежурств водителей

## Машины

Выйти

Администратор
Профиль
Сотрудники v
Заказы
Проезды
Дежурства v
Прочее v

ГОСУДАРСТВЕННЫЙ НОМЕР	МАРКА	МОДЕЛЬ
M010EP077	Лада	Калина
M000PH777	Жигуль	2101
M000MM003	Газель	Next
M000MM555	Ласточка	Б52

Государственный номер

  RUS

Марка

Модель

Добавить

Закрыть

Рис. 3.11 — Страница просмотра и регистрации машин





Рис. 3.12 — Страница просмотра и выбора заказа (для роли водителя)

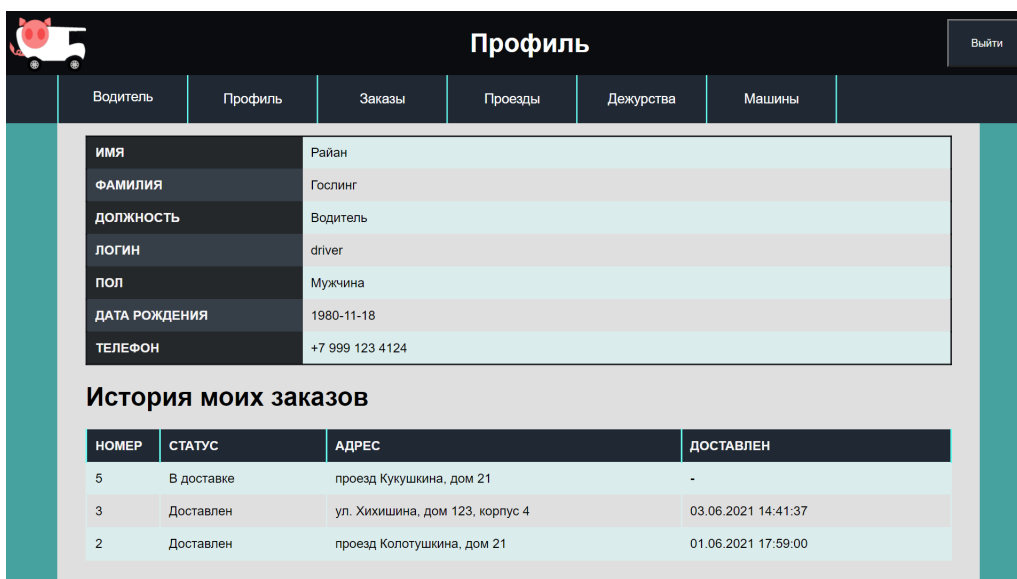


Рис. 3.13 — Личная страница (для роли водителя)

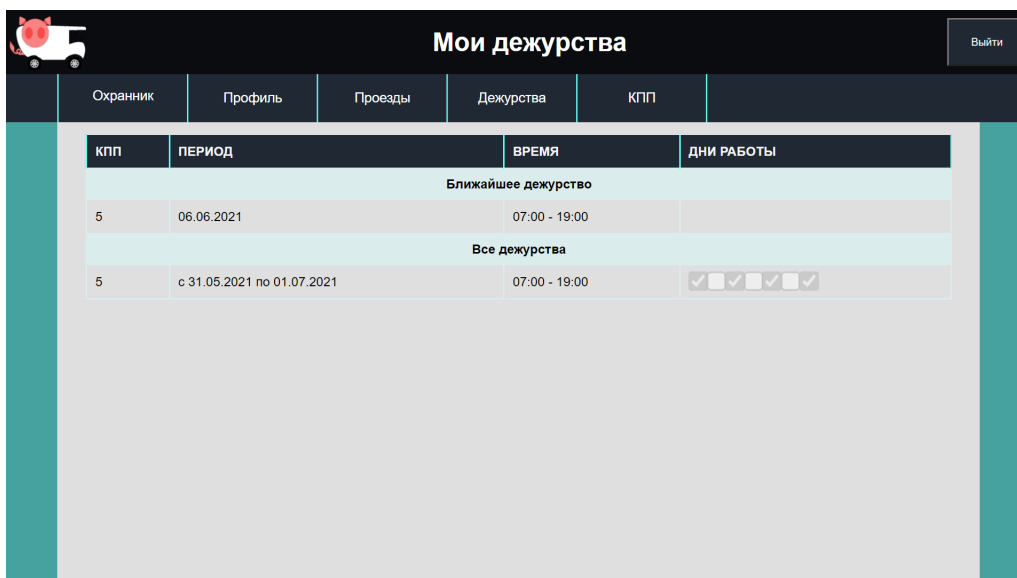


Рис. 3.14 — Страница просмотра дежурств (для роли охранника)

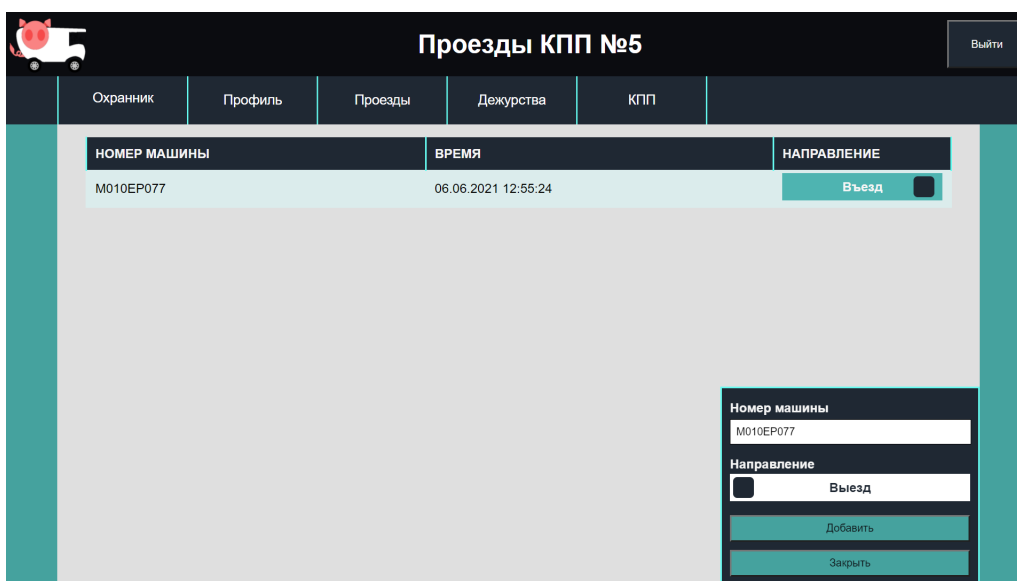


Рис. 3.15 — Страница просмотра и добавления записей проездов (для роли охранника)

Авторизация

Войти

Логин

admin

Пароль

\*\*\*\*\*

Войти

Зарегистрироваться

Рис. 3.16 — Страница авторизации

Регистрация

Войти

Логин

Пароль

Выберите должность

Имя

Фамилия

Мужчина ☒ Женцина ☐

Дата рождения

01.01.2000

Телефон

+7 000 123 4567

Создать аккаунт

Рис. 3.17 — Страница регистрации

## Заключение

В ходе курсовой работы была достигнута поставленная цель: было разработано web-приложение и база данных для транспортной системы завода. Выполнены все задачи работы.

В ходе работы была формализована задача, определён и реализован необходимый функционал по взаимодействию с данными. Проведён анализ и выбор реляционной СУБД, с помощью которой была спроектирована и реализована база данных. Также были выбраны и использованы средства программной реализации web-приложения для предоставления графического интерфейса для работы с данными.

Реализованное приложение позволяет потенциальным сотрудникам завода удобно просматривать и изменять компоненты транспортной системы, в соответствии со своими обязанностями.

## Список литературы

1. Коннолли Т., Бегг К. Базы данных: проектирование, реализация, сопровождение. Теория и практика, 3-е изд. : Пер. с англ. : Уч. пос. –М.: Изд. дом "Вильямс 2003. –1440 с.
2. Дейт К. Дж. Введение в системы баз данных. — 8-е изд. — М.: «Вильямс», 2006.
3. Чухраев И.В., Жукова И.В. Оптимизация работы с информацией в базах данных // Инновационная наука. 2016. №4-3 (16).
4. Информационные системы и базы данных: организация и проектирование: учеб. пособие. — СПб.: БХВ-Петербург, 2009. — 528 с.: ил. — (Учебная литература для вузов)
5. Тортика Алексей Сергеевич, Ершов Алексей Сергеевич ОБЗОР И СРАВНИТЕЛЬНЫЙ АНАЛИЗ СОВРЕМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ // Вестник СГТУ. 2020. №4 (87)
6. Драч В.Е., Родионов А.В., Чухраева А.И. Выбор системы управления базами данных для информационной системы промышленного предприятия // Электромагнитные волны и электронные системы. 2018. Т. 23. № 3. С. 71-80.
7. Документация Python 3.9.5 [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/>, свободный (дата обращения: 01.05.2021).
8. Документация ORM peewee [Электронный ресурс]. Режим доступа: <http://docs.peewee-orm.com/en/latest>, свободный (дата обращения: 01.05.2021).

9. Документация Web-фреймворка Django [Электронный ресурс].  
Режим доступа: <https://docs.djangoproject.com/en/3.2/>, свободный (дата обращения: 01.05.2021).