## РЕФЕРАТ

Расчетно-пояснительная записка 20 с., 4 рис., 13 табл., 9 ист., 0 прил.

БАЗА ДАННЫХ, АВТОПАРКОВКА, POSTGRESQL, ИНДЕКС БАЗЫ ДАННЫХ, СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ.

Цель работы – проектирование и разработка базы данных и приложения для сети автопарковок.

В рамках курсовой работы был проведен анализ предметной области автопарковок, формализованы бизнес-правила, спроектирована и разработана база данных автопарковок и приложение для доступа к ней. Кроме того, было проведено исследование зависимости времени выполнения запроса от наличия индекс, его типа и количества записей в базе данных. Также, исследовалась зависимость объема требуемой памяти для хранения триггера от его типа и количества записей.

## СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Актуальность курсовой работы определяется постоянным ростом числа транспортных средств и острой нехваткой парковочных мест [1]. Поиск свободных парковок не только доставляет неудобство автовладельцам, но и приводит к загруженности дорог и аварийным ситуациям. Возможность получить информацию о наличии свободных мест, а также возможность забронировать заранее парковочное место значительно облегчает планирование передвижений жителей городов и мегаполисов.

Целью курсовой работы является разработка базы данных и приложения для сети автопарковок.

Для достижения поставленной цели, необходимо решить следующие задачи:

1) провести анализ предметной области;
2) формализовать требования к создаваемой системе;
3) спроектировать базу данных и приложение для доступа к ней;
4) реализовать спроектированное программное обеспечение;
5) исследовать зависимость времени выполнения запроса и занимаемой триггером памяти от типа триггера и объема записей базы данных.

# 1 Аналитический раздел

В данном разделе приводится анализ предметной области и формализуются данные и задачи программного обеспечения. Также приводится анализ моделей баз данных.

## 1.1 Анализ предметной области

Создание парковочных мест является неотъемлемой частью организации дорожно-транспортной системы. Один из самых удобных способов решения этой задачи – автоматические автопарковки [2]. Использование такого типа парковок позволяет не только снизить затраты на организацию инфраструктуры парковок, но и обеспечить круглосуточный сервис для автовладельцев.

Въезд и выезд на такие парковки осуществляется при помощи талонов или RFID-карт. Оплачивается же время парковки на специальных устройствах – паркоматах, поддерживающих оплату как наличными, так и банковской картой.

Как правило, на въезде на парковку стоит табло с указанием количества свободных мест. На самой парковки ведется видеонаблюдение, что повышает безопасность припаркованного транспорта.

## 1.2 Анализ существующих решений

Разработка приложения для сети автопарковок позволяет сделать процесс парковки более удобным. В частности, с помощью приложения возможно организовать бронь парковочного места, получить информацию о количестве свободных мест, оформить абонемент, оплатить время парковки с помощью QR-кода. Используя перечисленные возможности как критерии, был проведен анализ существующих решений, таких как:

— Came Vector [3];

— Квазар [4];

— Московский паркинг [5].

Результаты анализа приведены в таблице 1.1.

Таблица 1.1 – Сравнение известных решений

| | Московский паркинг | Came Vector | Квазар |
|---|---|---|---|
| Возможность брони | нет | нет | нет |
| Информация о количестве свободных мест | да | да | нет |
| Оформление абонемента | да | нет | да |
| Оплата с помощью QR-кода | нет | нет | нет |

Как видно из результатов анализа, ни одно из существующих решений не удовлетворяет всем решениям.

### 1.3 Формализация задача

В рамках курсовой работы необходимо разработать базу данных для хранения информации о парковках и приложение для получения и обработки информации, хранящейся в ней. В приложении необходимо реализовать следующих акторов: пользователь, гость (незарегистрированный пользователь), администратор парковки, паркомат (устройство для фиксации начала и окончания парковки). Диаграмма использования приложения приведена на рисунке 1.1.

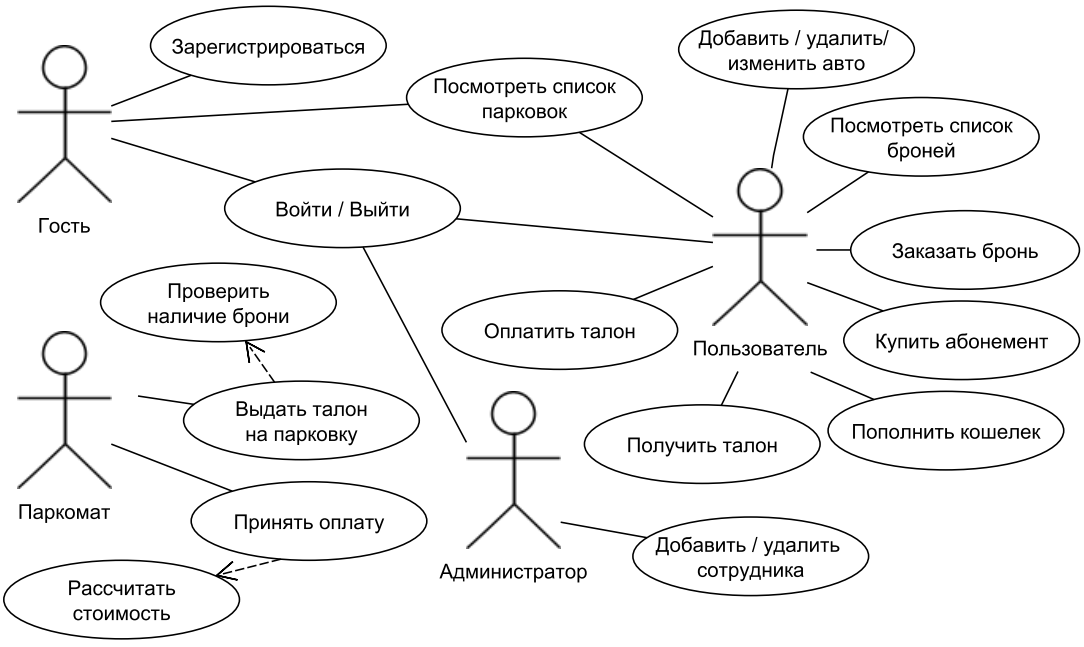


Рисунок 1.1 — Диаграмма использования приложения

## 1.4 Формализация данных

В результате анализа предметной области и формализации задачи был сформирован набор сущностей. Информация о них представлена в таблице 1.2.

Таблица 1.2 – Сущности базы данных

| Сущность | Атрибуты |
|---|---|
| Парковка | Адрес, количество мест (всего и свободных), действующий тариф |
| Автовладелец | ФИО, текущий абонемент, состояние счета |
| Бронь | Статус, даты начала и окончания, автовладелец, парковка |
| Парковочный талон | Статус, даты начала и окончания, автовладелец, парковка |
| Транспорт | Номер, марка |
| Абонемент | Название, стоимость, срок действия, бонусы |
| Тариф | Название, цена |
| Паркомат | Парковка, номер въезда / выезда |
| Сотрудник | ФИО, должность |
| Должность | Наименование, оклад |

На рисунке 1.2 приведена ER-диаграмма сущностей в нотации Чена.

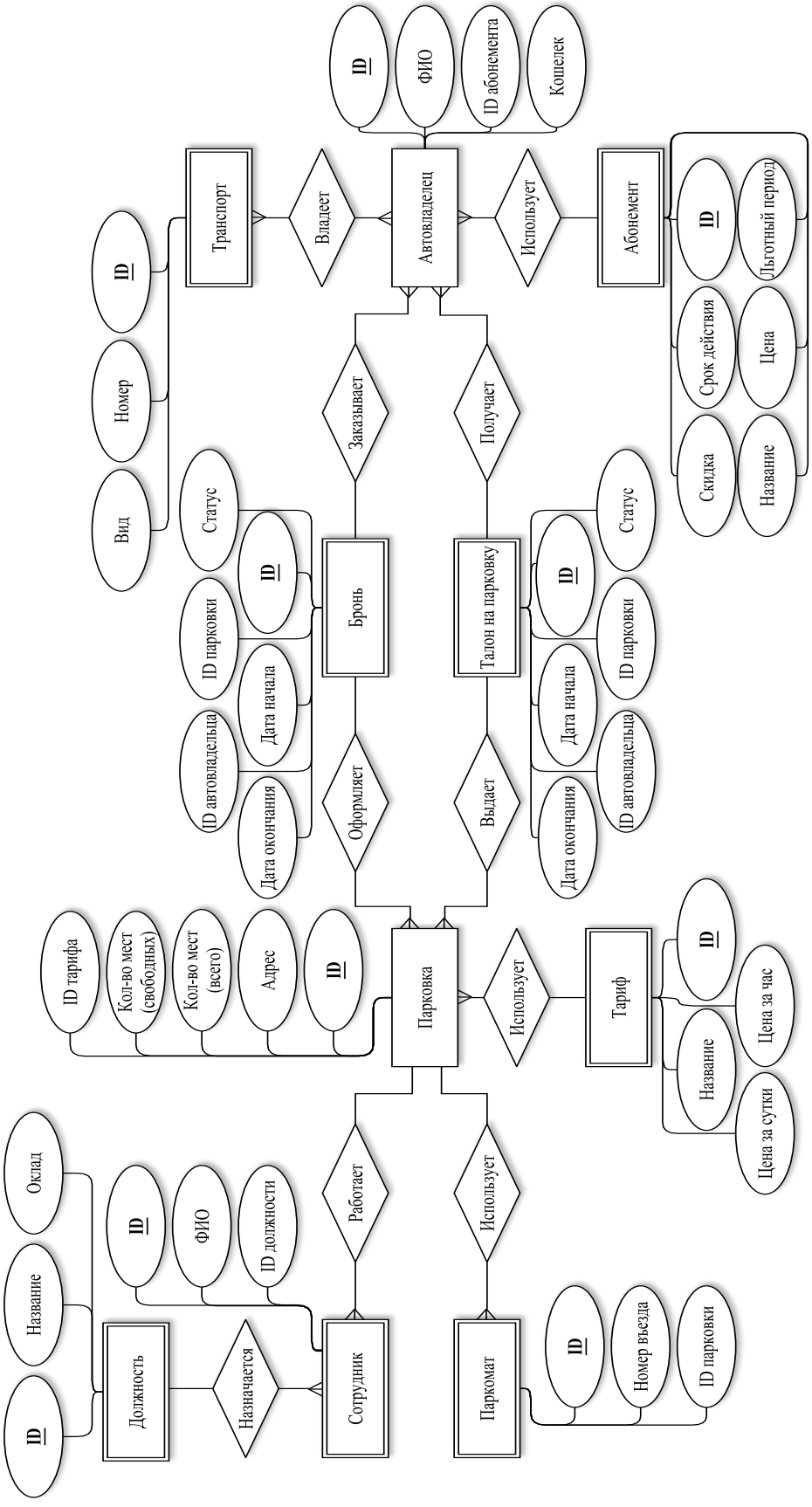


Рисунок 1.2 — ER-диаграмма сущностей в нотации Чена

## 1.5 Анализ баз данных

База данных – это некоторый набор перманентных данных, используемых прикладными программными системами какого-либо предприятия [6].

По модели хранения базы данных делятся на три группы:

— дореляционные

— реляционные

— постреляционные.

### 1.5.1 Дореляционные базы данных

Дореляционные модели близки к управлению данными во внешней памяти на низком уровне, что приводит к наилучшему использованию памяти, но такие модели данных имеют сложность использования. Так как логика процедуры выбора данных зависит от физической организации этих данных, то данная модель не является полностью независимой от приложения и как следствие приводит к усложнению действий изменений в базе данных [7].

### 1.5.2 Реляционные базы данных

Данные реляционных баз хранятся в виде таблиц, которые могут иметь связи с другими таблицами через внешние ключи, таким образом образуя отношения [7].

Структура реляционных баз данных позволяет связывать информацию из разных таблиц с помощью внешних ключей, которые используются для уникальной идентификации любого атомарного фрагмента данных в этой таблице. Другие таблицы могут ссылаться на этот внешний ключ.

### 1.5.3 Постреляционные базы данных

Классическая реляционная модель предполагает неделимость данных, хранящихся в полях записей таблиц. Это означает, что информация в таблице представляется в первой нормальной форме. Существует ряд случаев, когда это ограничение мешает эффективной реализации приложений. Постреляционная база данных использует трехмерные структуры, позволяя хранить в полях таблицы другие таблицы, расширяя таким образом возможности по описанию сложных объектов реального мира [8].

## 1.6 Вывод

В данном разделе был проведен анализ предметной области и существующих, формализованы задача и используемые данные. Также был проведен анализ баз данных, в результате которого было решено использовать реляционную базу данных. Такой выбор обусловлен необходимостью использования разнообразных запросов различной сложности.

# 2 Конструкторский раздел

В данном разделе приведена диаграмма проектируемой базы данных, описана ролевая модель и хранимая процедура базы данных.

## 2.1 Описание сущностей базы данных

Исходя из формализации данных, приведенной в подразделе 1.4, база данных должна состоять из следующих таблиц:

— parkings – таблица парковок;

— auto_owners – таблица автовладельцев (пользователей);

— bookings – таблица броней;

— tickets – таблица парковочных талонов;

— cars – таблица машин;

— subscriptions – таблица абонементов;

— tariffs – таблица парковочных тарифов;

— parking_meters – таблица паркоматов;

— employees – таблица сотрудников;

— jobs – таблица должностей;

На рисунке 2.1 представлена ER-диаграмма базы данных.

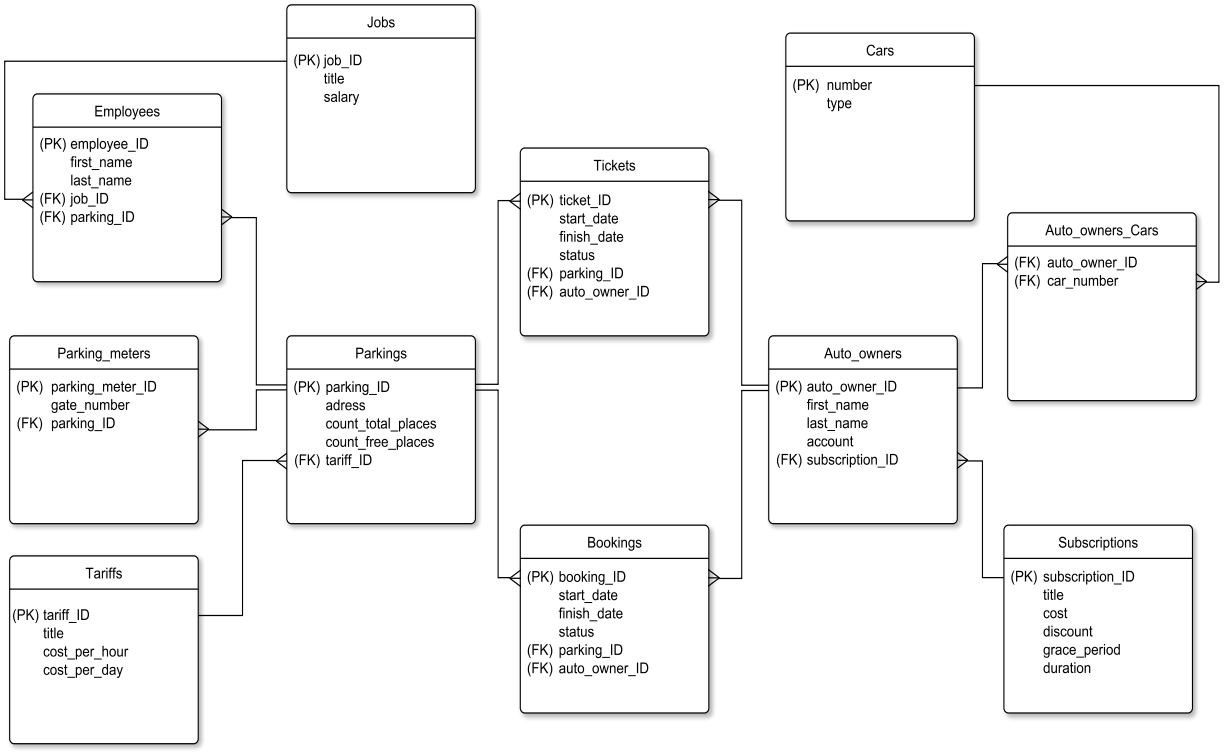


Рисунок 2.1 — ER-диаграмма базы данных

В связи с тем, что один и тот же автомобиль может принадлежать нескольким пользователям, и при этом у одного пользователя может быть несколько автомобилей, таблицы cars и auto_owners находятся в отношении многие-ко-многим. Для реализации такой связи была введена таблица-связка auto_owners_cars.

Информация о структуре и ограничениях каждой из перечисленных таблиц указана в таблицах 2.1 – 2.11.

Таблица 2.1 – Инормация о таблице auto_owners

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| auto_owner_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор пользователя |
| first_name | VARCHAR(32) | NOT NULL | Имя |
| last_name | VARCHAR(32) | NOT NULL | Фамилия |
| account | INT | NOT NULL, $\geq 0$ | Счет |
| subscription_ID | INT | FOREIGN KEY | Идентификатор абонемента |

Таблица 2.2 – Инормация о таблице auto_owner_cars

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| auto_owner_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор пользователя |
| car_number | VARCHAR(32) | NOT NULL, FOREIGN KEY | Номер авто |

Таблица 2.3 – Инормация о таблице cars

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| number | VARCHAR(32) | NOT NULL, PRIMARY KEY | Номер авто |
| model | VARCHAR(32) | NOT NULL | Модель авто |

Таблица 2.4 – Инормация о таблице subscriptions

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| subscription_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор абонемента |
| title | VARCHAR(32) | NOT NULL | Название |
| discount | FLOAT | NOT NULL,≥ 0, < 1 | Скидка |
| cost | INT | NOT NULL, > 0 | Стоимость |
| grace_period | INT | NOT NULL,≥ 0 | Льготные часы |
| duration | INTERVAL | NOT NULL | Срок действия абонемента |

Таблица 2.5 – Инормация о таблице parkings

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| parking_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор парковки |
| adress | VARCHAR(32) | NOT NULL | Адрес |
| cnt_total_places | INT | NOT NULL, > 0 | Количество мест всего |
| cnt_free_places | INT | NOT NULL, ≥ 0 | Количество свободных мест |
| tariff_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор тарифа |

Таблица 2.6 – Инормация о таблице tariffs

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| tariff_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор тарифа |
| title | VARCHAR(32) | NOT NULL | Название |
| cost_per_hour | INT | NOT NULL, ≥ 0 | Стоимость часа |

Таблица 2.7 – Инормация о таблице parking_meters

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| parking_meter_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор паркомата |
| gate_number | INT | NOT NULL, $\geq 0$ | Номер въезда |
| parking_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор парковки |

Таблица 2.8 – Инормация о таблице employees

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| employee_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор сотрудника |
| first_name | VARCHAR(32) | NOT NULL | Имя |
| last_name | VARCHAR(32) | NOT NULL | Фамилия |
| job_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор должности |
| parking_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор парковки |

Таблица 2.9 – Инормация о таблице jobs

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| job_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор должности |
| title | VARCHAR(32) | NOT NULL | Название |
| salary | INT | NOT NULL, > 0 | Оклад |

Таблица 2.10 – Инормация о таблице bookings

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| booking_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор брони |
| start_date | TIMESTAMP | NOT NULL | Дата начала |
| finish_date | TIMESTAMP | > start_date | Дата окончания |
| status | BOOLEAN | NOT NULL | Статус |
| parking_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор парковки |
| auto_owner_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор пользователя |

Таблица 2.11 – Инормация о таблице tickets

| Столбец | Тип данных | Ограничение | Описание |
|---|---|---|---|
| ticket_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор талона |
| start_date | TIMESTAMP | NOT NULL | Дата начала |
| finish_date | TIMESTAMP | > start_date | Дата окончания |
| status | BOOLEAN | NOT NULL | Статус |
| parking_ID | INT | NOT NULL, FOREIGN KEY | Идентификатор парковки |
| auto_owner_ID | INT | NOT NULL, PRIMARY KEY | Идентификатор пользователя |

## 2.2 Ролевая модель

Согласно диаграмме использования приложения, представленной на рисунке 1.1, необходимо выделить следующие роли:

— гость – обладает правами только на просмотр таблиц parkings, tariffs, subscriptions;

— автовладелец – обладает правами на просмотр и изменение записей, связанных с ним, во всех таблицах, кроме employees, jobs и parking_meters;

— паркомат – обладает правами на просмотр всех таблиц, создание и обновление записей в таблицах bookings и tickets;

— Администратор – обладает всеми правами доступа ко всем таблицам.

## 2.3 Хранимая процедура

В проектируемой базе данных таблицы cars и auto_owners связанны отношением многие-ко-многим. При создании нового авто необходимо обновить сразу две таблицы: таблицу cars и таблицу-связку auto_owners_cars. Для повешения надежности и вместе с тем упрощения процесса добавления нового авто, было решено написать хранимую процедуру insert_car, выполняющую эту задачу.

## 2.4 Диаграмма классов приложения

На рисунке 2.2 приведена UML-диаграмма классов.

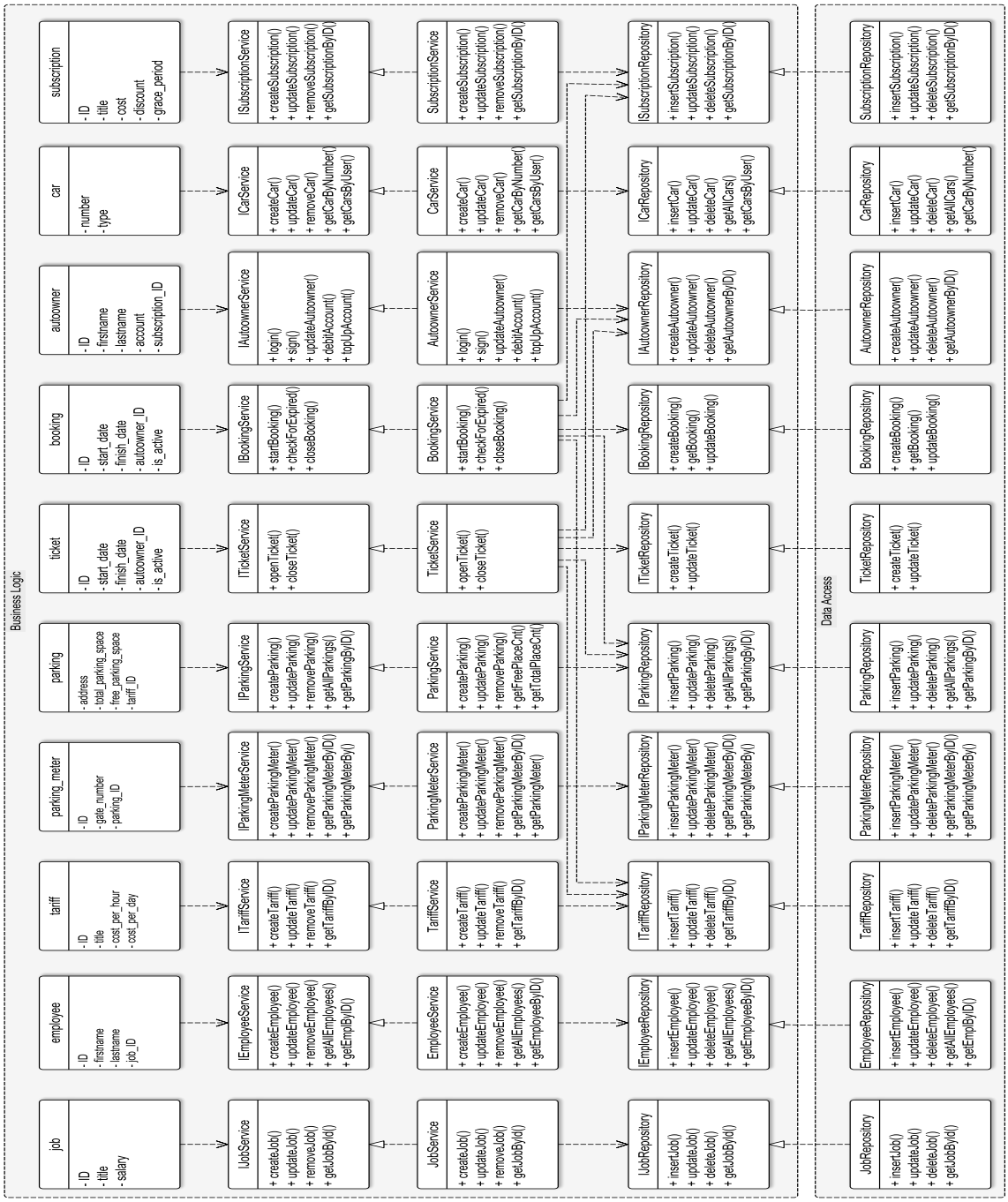


Рисунок 2.2 — UML-диаграмма классов

Проектирование велось в соответствии с принципами «чистой архитектуры» [9], в частности был спроектирован компонент бизнес-логики, полностью независимый от компонента доступа к данным и пользовательского интерфейса.

## 2.5 Вывод

В данном разделе были описаны сущности проектируемой базы данных, ролевая модель и хранимая процедура. Также приведена ER-диаграмма базы данных и UML-диаграмма классов.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. *Вавринчук П. А.*, *Рябкова Е. Б.* Паркинг – основное решение дефицита парковочных мест // Новые идеи нового века. — 2014.

2. *Анисимова Н. А.*, *Потлова Л. А.* Обоснование инновационных преимуществ автоматических парковок для автомобилей // Научный вестник воронежского государственного архитектурно-строительного университета. — 2017.

3. Came Vector [Электронный ресурс]. — Режим доступа: `https://www.vector-ap.ru/` (дата обращения: 23.04.2024).

4. Квазар [Электронный ресурс]. — Режим доступа: `https://kvazar.ru/` (дата обращения: 23.04.2024).

5. Московский паркинг [Электронный ресурс]. — Режим доступа: `https://parking.mos.ru/` (дата обращения: 23.04.2024).

6. *Дейт К. Д.* Введение в системы баз данных. — «Вильямс», 2006.

7. *Кузнецов С. Д.* Основы современных баз данных. — Центр Информационных Технологий, 1998.

8. *Парфенов Ю. П.* Постреляционные хранилища данных: учебное пособие. — Центр Информационных Технологий, 2016.

9. *Мартин Р.* Чистая архитектура. — «Питер», 2018.