МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема «Программное средство «Железнодорожный вокзал»»

**Исполнитель**

студент 3 курса 7 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тимошенко Д. В.

подпись, дата

**Руководитель**

Ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Якунович А. В.

должность, учен. степень, ученое звание подпись, дата

Допущен к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ Якунович А. В.

подпись дата инициалы и фамилия

Минск 2023

**Содержание**

[**Введение** 6](#_Toc149643741)

[1. Аналитический обзор литературы по теме проекта 7](#_Toc149643742)

[1.1 Сравнительный анализ теоретических и эвристических методов решения поставленных задач и существующих по данной тематике технических научных решений 7](#_Toc149643743)

[1.2 Аналитический обзор аналогов 8](#_Toc149643744)

[1.2.1 Аналог «Белорусская железная дорога» 8](#_Toc149643745)

[1.2.2 Аналог «Intercity.pl» 9](#_Toc149643746)

[1.3 Изучение требований, определение вариантов использования 9](#_Toc149643747)

[1.3.1 Определение основных функциональных требований к программному средству 10](#_Toc149643748)

[1.3.3 Определение вариантов использования 10](#_Toc149643749)

[1.4 Вывод 10](#_Toc149643750)

[2. Анализ и проектирование архитектуры приложения 11](#_Toc149643751)

[2.1 Описание информационых объектов и ограничений целостности в базе данных 11](#_Toc149643752)

[2.1.1 Таблица Stations 11](#_Toc149643753)

[2.1.2 Таблица Routes 12](#_Toc149643754)

[2.1.3 Таблица Schedule 12](#_Toc149643755)

[2.1.4 Таблица Trains 12](#_Toc149643756)

[2.1.5 Таблица Vans 13](#_Toc149643757)

[2.1.6 Таблица Passengers 13](#_Toc149643758)

[2.1.7 Таблица Payments 13](#_Toc149643759)

[2.3.8 Таблица Tickets 14](#_Toc149643760)

[2.1.9 Таблица Stations\_routes 14](#_Toc149643761)

[2.2 Диаграммы UML, взаимосвязь всех компонентов 14](#_Toc149643762)

[2.3 Обобщенная структура управления приложением 15](#_Toc149643763)

[2.4 Вывод 15](#_Toc149643764)

[3. Разработка модели базы данных 16](#_Toc149643765)

[**3.4 Вывод** 16](#_Toc149643766)

[4. Установка, настройка и использование Oracle 19c 17](#_Toc149643767)

[4.4 Вывод 17](#_Toc149643768)

[5. Тестирование, проверка работоспособности и анализ полученных результатов. 18](#_Toc149643769)

[5.3. Вывод. 18](#_Toc149643770)

[6. Руководство по использованию программного продукта 19](#_Toc149643771)

[6.1 Установка приложения 19](#_Toc149643772)

[6.3. Вывод 19](#_Toc149643773)

[Заключение 20](#_Toc149643774)

[Список использованных литературных источников 21](#_Toc149643775)

[Приложение 22](#_Toc149643776)

[Приложение А 22](#_Toc149643777)

**ВВЕДЕНИЕ**

Объектно-ориентированное программирование (ООП) является одним из основных подходов в современной разработке программного обеспечения. ООП позволяет организовать код в виде объектов, которые взаимодействуют друг с другом, образуя сложные системы. Этот подход позволяет создавать гибкие, модульные и легко поддерживаемые программы.

Также в нашем современном мире имеется огромный спрос пользователей на приложения, для взаимодействия с данными в базе данных, которые могли бы постоянно работать и позволять удобно пользоваться хранимой информацией. Особенно важным становится использование баз данных в железнодорожной отрасли для обработки и хранения данных о состоянии железнодорожных вокзалов и поездов.

В рамках данного проекта рассматривается разработка программного средства "Железнодорожный вокзал". Железнодорожные вокзалы играют важную роль в организации пассажирских перевозок, обеспечивая комфорт и безопасность пассажиров. Целью программного средства является автоматизация процессов, связанных с управлением и организацией работы железнодорожного вокзала.

Программное средство "Железнодорожный вокзал" будет разработано с использованием принципов и методов объектно-ориентированного программирования. ООП позволит представить различные сущности вокзала, такие как поезда, расписание, билеты и пассажиры, в виде объектов с определенными свойствами и поведением. Благодаря этому подходу, разработчики смогут создать модульную систему, которая будет легко расширяемой и поддерживаемой.

1. Аналитический обзор литературы по теме проекта

1.1 Сравнительный анализ теоретических и эвристических методов решения поставленных задач и существующих по данной тематике технических научных решений

Существует множество методов решения этой задачи, которые могут быть классифицированы по следующим параметрам:

1. Производительность: скорость выполнения запросов и эффективность использования ресурсов.
2. Надежность: способность системы работать без сбоев и сбоев, а также ее способность восстановиться после сбоя.
3. Гибкость: способность системы адаптироваться к изменениям требований и удовлетворять различным потребностям пользователей.
4. Расширяемость: возможность системы расширяться и добавлять новые функциональные возможности.

Существует также множество технических научных решений в контексте ООП, включая разработку программных инструментов, фреймворков и библиотек. Например, фреймворки Spring и Hibernate предоставляют инструменты для разработки и управления объектно-ориентированными приложениями, обеспечивая реализацию инверсии управления и персистентности данных.

Теоретические методы решения задач в ООП представляют собой формальные подходы и модели, разработанные для анализа и проектирования систем на основе ООП. В литературе были исследованы различные теоретические методы, такие как моделирование классов, анализ наследования и принципы SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion).

Например, в работе "Object-Oriented Analysis and Design with Applications" Гради Буча[1] описывает принципы и методы анализа и проектирования в ООП. Автор предлагает использовать моделирование классов для выявления основных сущностей системы и определения их взаимодействия. Анализ наследования позволяет строить иерархии классов, обеспечивая повторное использование кода и гибкость системы.

Эвристические методы решения задач в ООП основаны на эмпирическом опыте и эвристических правилах, позволяющих разработчикам применять стандартные решения для типовых задач. В литературе были изучены различные эвристические методы, такие как шаблоны проектирования, архитектурные принципы и методологии разработки, такие как Agile и Scrum.

Например, в книге "Design Patterns: Elements of Reusable Object-Oriented Software" Гамма, Хелм, Джонсон и Влиссидес[2] представляют каталог шаблонов проектирования, которые предлагают решения для типичных проблем проектирования в ООП. Эти шаблоны, такие как "Фабричный метод", "Стратегия" и "Наблюдатель", позволяют разработчикам создавать гибкие и расширяемые системы.

1.2 Аналитический обзор аналогов

В современном мире, когда скорость и точность играют важную роль, транспортные компании становятся все более популярными. Одной из главных задач таких компаний является надежная доставка клиентов в места назначения.

1.2.1 Аналог «Белорусская железная дорога»

В качестве аналога рассмотрим приложение белорусской железной дороги [3], которое представлено на рисунке 1.1.

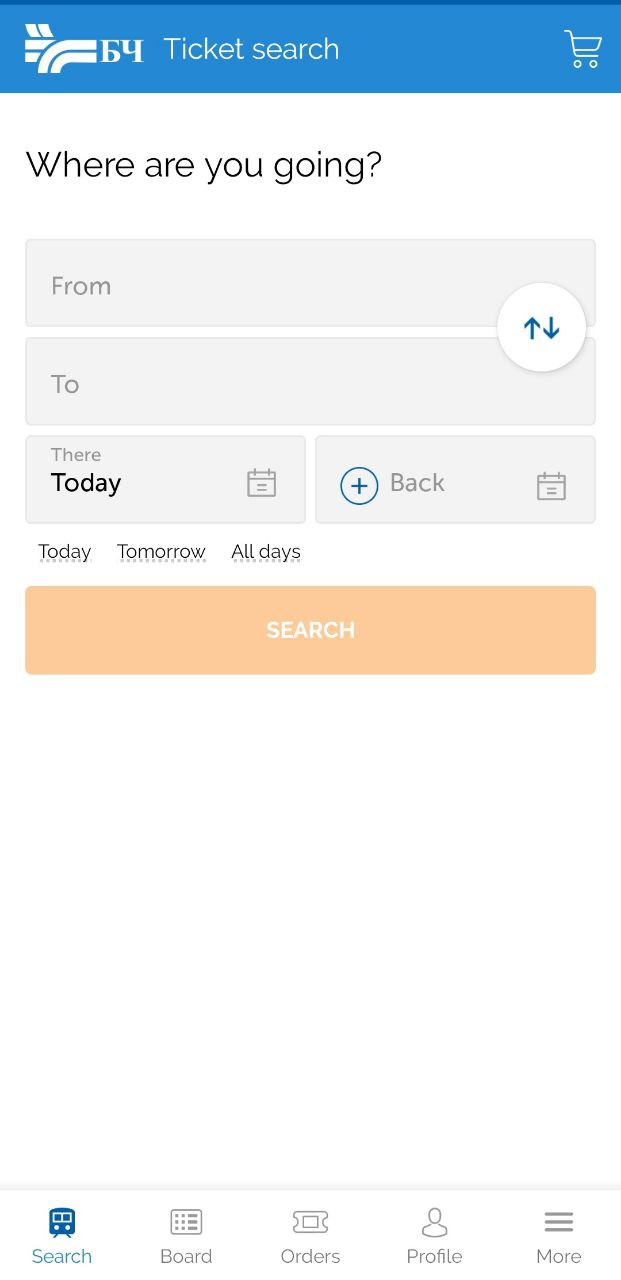


Рисунок 1.1 – Приложение «БЧ. Мой поезд»

Данное приложение позволяет из любой точки страны, где имеется доступ в интернет, забронировать билет на поезд. Пример представлен на рисунке 1.2.

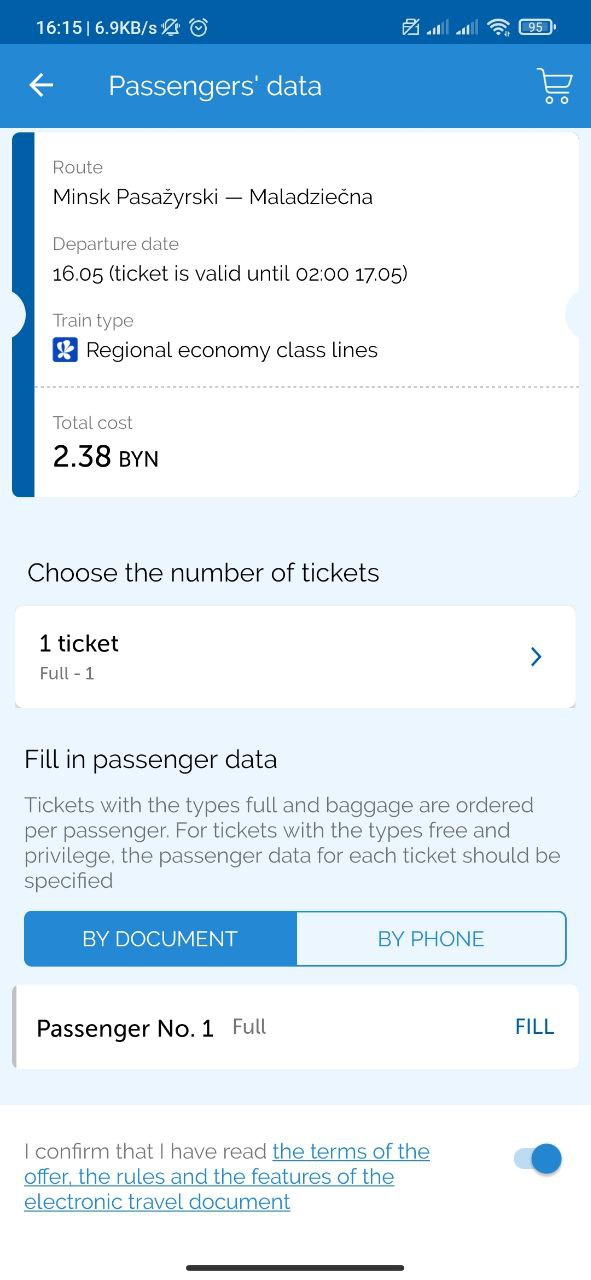


Рисунок 1.2 – Бронирование билета в приложении «БЧ. Мой поезд»

Также приложение позволяет просмотреть расписание рейсов онлайн. Пример на рисунке 1.3.

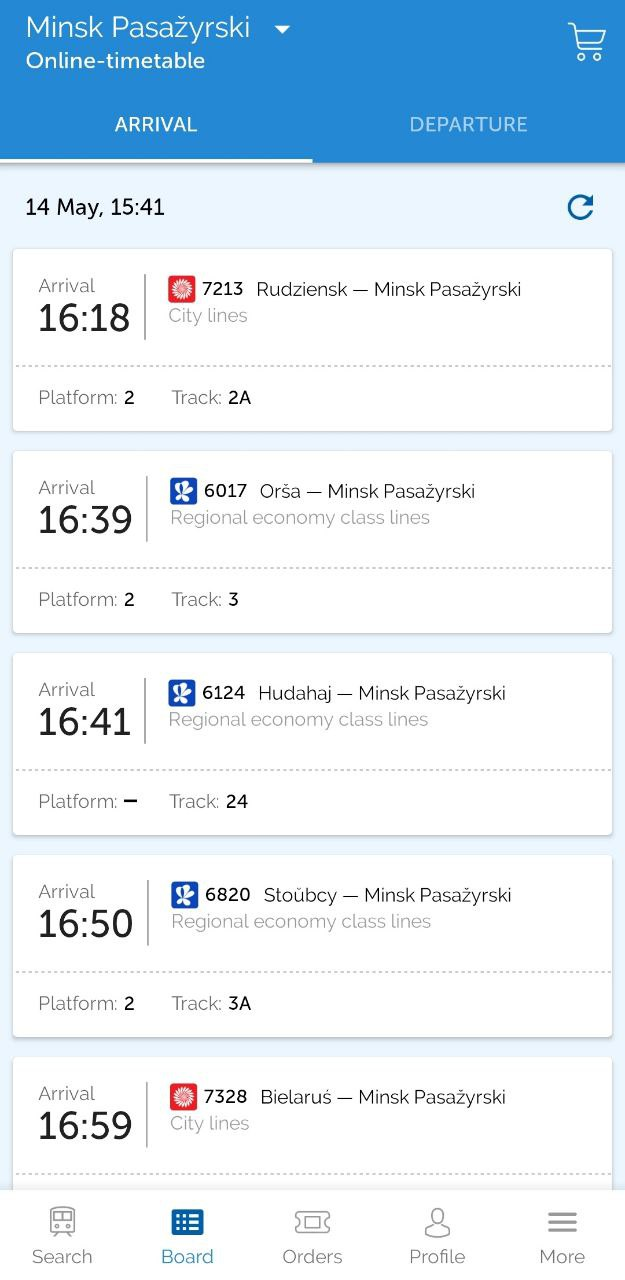


Рисунок 1.3 – Расписание рейсов в приложении «БЧ. Мой поезд»

Однако, как и любое подобное приложение, у него есть минусы. Одним из таких минусов можно назвать ошибки при бронировании билетов. У некоторых пользователей возникают проблемы с бронированием билетов через приложение. Например, система может выдавать ошибки при оплате или не отображать доступные для бронирования билеты.

1.2.2 Аналог «Intercity.pl»

Еще одним аналогом является сайт «Intercity.pl» [4], предоставляющий услуги по бронированию билетов на территории Польши. Сайт продемонстрирован на рисунке 1.4.

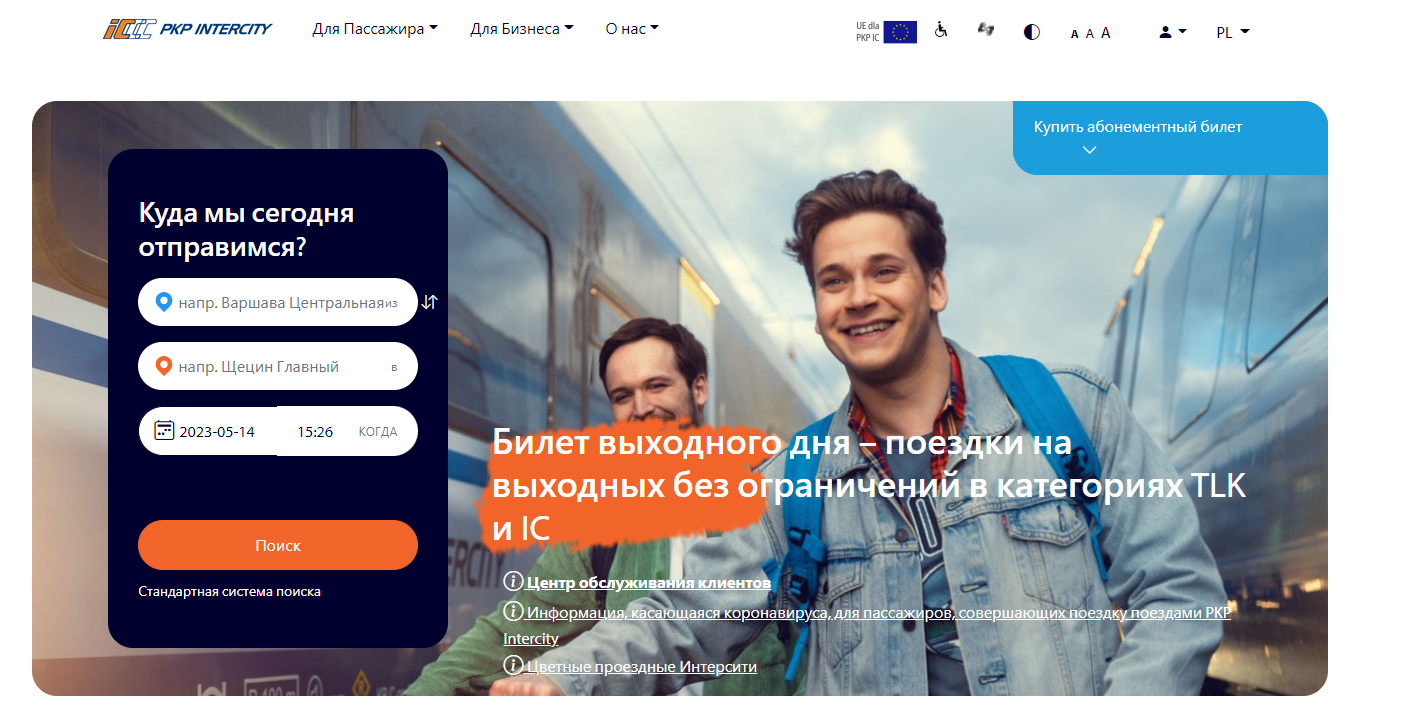


Рисунок 1.4 – Сайт «Intercity.pl»

Из минусов, можно отметить, наличие только возможности поиска билетов и бронирования билетов.

1.3 Изучение требований, определение вариантов использования

Важным этапом разработки проекта является изучение требований и определение возможных вариантов использования.

1.3.1 Определение основных функциональных требований к программному средству

В ходе изучения требований было выявлено, что программное средство железнодорожного вокзала должна включать информацию о поездах, вагонах, расписании движения, билетах и о пассажирах. Были определены следующие варианты использования:

1. Поддерживать работу с базой данных.
2. Предоставление информации о расписании движения поездов пассажирам.
3. Создавать, изменять, удалять данных из базы данных.
4. Хранение информации о билетах на поезда, включая стоимость, маршрут, дату и время отправления, место в вагоне.
5. Выполнение поисковых запросов и фильтрации.
6. Ведение учета проданных билетов и свободных мест на каждый поезд.
7. Предоставлять интерфейс для бронирования билетов.
8. Обеспечение защиты данных и доступа только авторизованных пользователей к базе данных.

1.3.3 Определение вариантов использования

Вариантами использования программного средства являются:

1. Авторизация пользователей или менеджеров.
2. Просмотр расписания поездов.
3. Просмотр информации о наличии мест на поездах.
4. Бронирование места на поезд.
5. Создание расписания со стороны менеджера.
6. Просмотр оформленных билетов.
7. Администрирование со стороны менеджера.

1.4 Вывод

В данном разделе проведен аналитический обзор литературы по теме проекта, а также осуществлен сравнительный анализ теоретических и эвристических методов решения поставленных задач и существующих технических научных решений.

Также был проведен аналитический обзор аналогов, включающий два конкретных примера аналогов: «Белорусская железная дорога» и «Intercity.pl». Это позволило определить преимущества и недостатки существующих решений и использовать эту информацию при разработке нового проекта.

Важным этапом является изучение требований, определение основных функциональных требований к приложению и вариантов использования. Это позволило определить ключевые аспекты, которые необходимо учитывать при проектировании и разработке проекта.

2. Анализ и проектирование архитектуры приложения

2.1 Описание информационых объектов и ограничений целостности в базе данных

Для разработки веб-приложения в рамках курсового проекта понадобилась база данных с необходимой конфигурацией сущностей для хранения всей необходимой информации.

Структуру базу данных, ограничения целостности, связи и поля можно увидеть в приложении A и на рисунке 2.1.

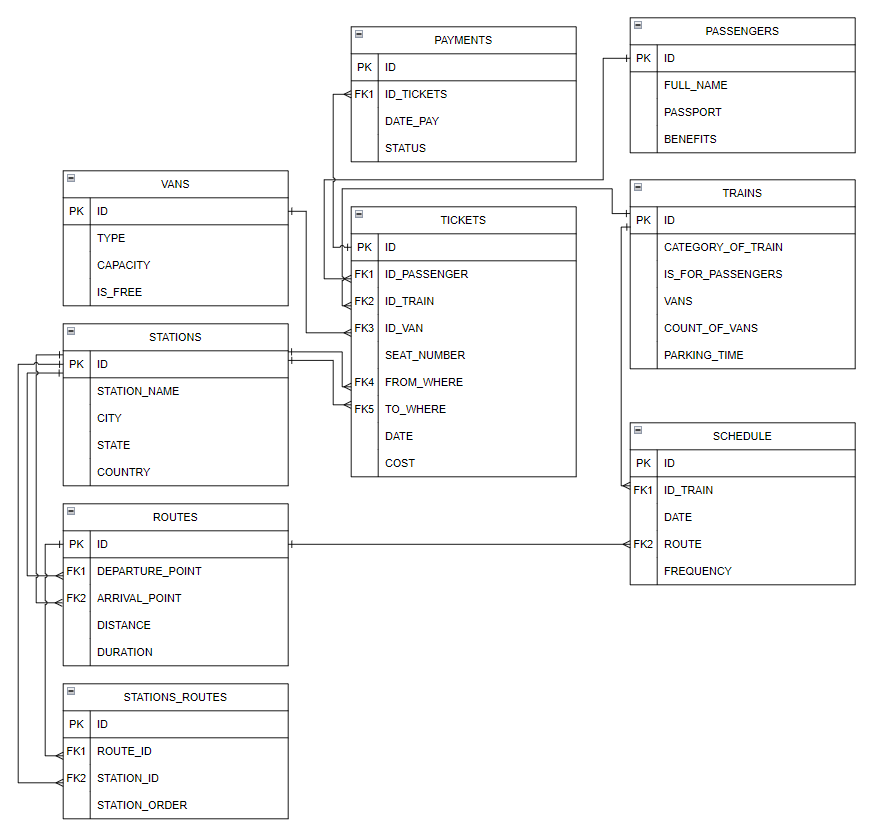


Рисунок 2.1 – Структура базы данных

В базе данных существует 9 таблиц, каждая из которых выполняет свою функцию.

2.1.1 Таблица Stations

Данная таблица используется для хранения информации о станциях остановки составов.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждой точки остановки.
* Cтолбец STATION\_NAME. Хранит название станции.
* Cтолбец CITY. Хранит название города, в котором расположена станция.
* Cтолбец STATE. Хранит название региона, в котором расположена станция.
* Cтолбец COUNTRY. Хранит название страны, в котором расположена станция.

Данная таблица содержит ограничение первичного ключа для столбца ID, а также все столбцы должны содержать какое-либо значение.

2.1.2 Таблица Routes

Данная таблица используется для хранения информации о маршрутах передвижения составов.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого маршрута.
* Cтолбец DEPARTURE\_POINT. Хранит ID начальной точки.
* Cтолбец ARRIVAL\_POINT. Хранит ID конечной точки.
* Cтолбец DISTANCE. Хранит расстояние, проходящее составом.
* Cтолбец DURATION. Хранит время, проводимое в пути.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также для столбцов DEPARTURE\_POINT и ARRIVAL\_POINT существуют ограничения целостности по внешнему ключу к таблице STATIONS.

2.1.3 Таблица Schedule

Данная таблица используется для хранения информации о расписании передвижения составов.

В её состав входят следующие столбцы:

* Cтолбец ID. Хранит уникальный идентификатор каждого маршрута.
* Cтолбец ID\_TRAIN. Хранит ID поезда, зарезервированного для маршрута.
* Cтолбец DATE. Хранит дату начала выдвижения состава по маршруту.
* Cтолбец ROUTE. Хранит ID маршрута передвижения состава.
* Cтолбец FREQUENCY. Хранит частоту использования данной ячейки расписания.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также для столбцов ID\_TRAIN и ROUTE существуют ограничения целостности по внешнему ключу к таблице TRAINS и ROUTES соответсвенно.

Также таблица включает ограничение check на столбец FREQUENCY.

2.1.4 Таблица Trains

Данная таблица используется для хранения информации о поездах(составах).

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого состава.
* Cтолбец CATEGORY\_OF\_TRAIN. Хранит категорию данного состава.
* Cтолбец IS\_FOR\_PASSENGERS. Хранит число, обозначающее возможность использования данного состава для перевозки пассажиров.
* Cтолбец VANS. Хранит массив ID вагонов, из которых состоит поезд.
* Cтолбец COUNTS\_OF\_VANS. Хранит количество вагонов в составе.
* Cтолбец PARKING\_TIME. Хранит время стоянки состава.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также таблица включает ограничение check на столбец IS\_FOR\_PASSENGERS.

2.1.5 Таблица Vans

Данная таблица используется для хранения информации о вагонах.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого вагона.
* Cтолбец TYPE. Хранит тип вагона.
* Cтолбец CAPACITY. Хранит вместимость вагона.
* Столбец IS\_FREE. Хранит число, указывающее, прикреплен ли вагон к составу.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также таблица включает ограничение check на столбец IS\_FREE.

2.1.6 Таблица Passengers

Данная таблица используется для хранения информации о пассажирах.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого пассажира.
* Cтолбец FULLNAME. Хранит ФИО пассажира.
* Cтолбец PASSPORT. Хранит паспортные данные пассажира.
* Cтолбец BENEFITS. Хранит льготы, применяемые к пассажиру.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также таблица включает ограничение check на столбец BENEFITS.

2.1.7 Таблица Payments

Данная таблица используется для хранения информации о платежах.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого платежа.
* Cтолбец ID\_TICKET. Хранит ID билета, по которому произошла оплата.
* Cтолбец DATE\_PAY. Хранит дату оплаты.
* Cтолбец STATUS. Хранит статус оплаты.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также для столбцов ID\_TICKET существует ограничение целостности по внешнему ключу к таблице TICKETS.

Также таблица включает ограничение check на столбец STATUS.

2.3.8 Таблица Tickets

Данная таблица используется для хранения информации о билетах.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого билета.
* Cтолбец ID\_PASSENGER. Хранит ID пассажира, которому был выдан билет.
* Cтолбец ID\_TRAIN. Хранит ID поезда, на который забронирован билет.
* Cтолбец ID\_VAN. Хранит ID вагона, в котором находится арендованное место.
* Cтолбец SEAT\_NUMBER. Хранит номер арендованного места.
* Cтолбец FROM\_WHERE. Хранит место посадки на поезд.
* Cтолбец TO\_WHERE. Хранит место выхода из поезда.
* Cтолбец DATE. Хранит дату поездки.
* Cтолбец COST. Хранит стоимость билета.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также для столбцов ID\_PASSENGER, ID\_TRAIN и ID\_VAN существуют ограничения целостности по внешнему ключу к таблице PASSENGERS, TRAINS и VANS соответсвенно.

2.1.9 Таблица Stations\_routes

Данная таблица используется для хранения информации о порядке следования станций в маршрутах.

В её состав входят следующие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждой станции в маршруте.
* Cтолбец ROUTE\_ID. Хранит ID маршрута, в котором привязана станция.
* Cтолбец STATION\_ID. Хранит ID станции, которая используется в маршруте.
* Cтолбец STATION\_ORDER. Хранит номер станции в маршруте.

Данная таблица содержит ограничение первичного ключа для столбца ID.

Также для столбцов ROUTE\_ID, STATION\_ID существуют ограничения целостности по внешнему ключу к таблице ROUTES, STATIONS соответственно.

2.2 Диаграммы UML, взаимосвязь всех компонентов

Диаграммы UML могут помочь визуализировать взаимосвязь всех компонентов в приложении, а также описать функциональность системы и ее взаимодействие с внешними пользователями или другими системами.

Диаграмма вариантов использования была создана для описания функциональности программного средства железнодорожного вокзала и ее взаимодействия с внешними пользователями. На диаграмме показаны актеры, варианты использования и связи между ними.

Актерами на диаграмме являются пользователи и менеджеры. Варианты использования включают бронирование билетов, просмотр расписания поездов, изменение информации о поездах и т.д.

Сама диаграмма представлена на рисунке 2.1.

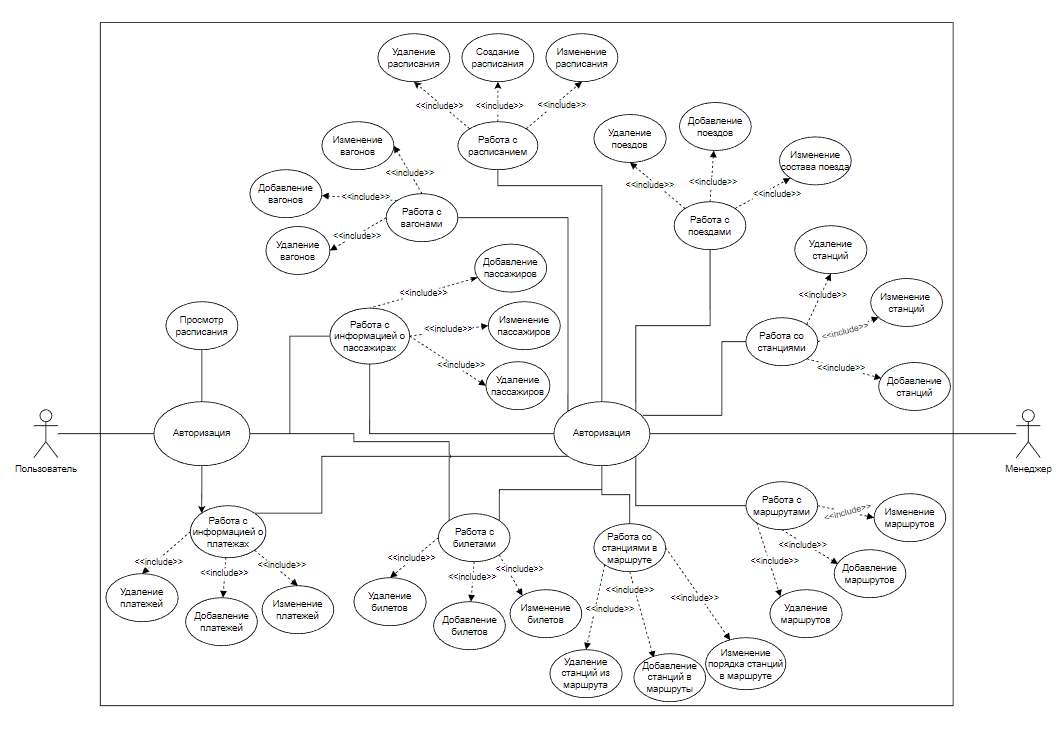


Рисунок 2.2 – Диаграмма вариантов использования

Диаграмма вариантов использования помогает лучше понять функциональность программного средства железнодорожного вокзала и ее взаимодействие с внешними пользователями, что может быть полезным при проектировании и разработке системы.

2.3 Обобщенная структура управления приложением

2.4 Вывод

В данном разделе была разработана архитектура проекта, описано взаимодействие приложения с базой данных, составлена диаграмма вариантов использования. Также была описана структура базы данных вместе с ограничениями целостности.

3. Проектирование программного средства

**3.4 Вывод**

В данном разделе была разработана модель данных, была составлена UML диаграмма базы данных, а также описана каждая таблица в диаграмме. Были описаны виды разработанных объектов базы данных. Также были описаны процедуры импорта и экспорта из JSON. Была рассмотрена используемая технология, которая значительно упрощает работу по администрированию базы данных.

4. Установка, настройка и использование Oracle 19c

4.4 Вывод

В данном разделе были рассмотрены созданные роли, а также приведены их листинги. Также было проведено тестирование скорости выполнения запросов по некоторым таблицам и столбцам.

5. Тестирование, проверка работоспособности и анализ полученных результатов.

5.3. Вывод.

В данном разделе были протестированы возможности клиентов и менеджеров. По итогам тестирования можно сказать, что все работает корректно.

6. Руководство по использованию программного продукта

6.1 Установка приложения

6.3. Вывод

Было составлено четкое и понятное руководство пользователя для работы с приложением. Были рассмотрены базовые функции приложения, а также предоставлены скриншоты.

Заключение

В данной работе была реализована база данных железнодорожного вокзала с использованием СУБД Oracle и средств мониторинга состояния. Целью работы было создание эффективной и надежной базы данных, которая позволит эффективно управлять данными о поездах, рейсах, билетах и пассажирах.

В рамках работы были поставлены следующие задачи: разработка модели базы данных, создание необходимых объектов (таблиц, представлений, индексов, ограничений), реализация процедур импорта и экспорта данных, тестирование производительности и разработка руководства пользователя.

Была разработана модель базы данных, состоящая из таблиц, связанных между собой отношениями. В модели были учтены основные аспекты работы железнодорожного вокзала, такие как рейсы, поезда, билеты, пассажиры, станции и т.д.

На основе разработанной модели были созданы необходимые объекты базы данных, такие как таблицы, представления, индексы и ограничения. Были использованы средства мониторинга состояния СУБД Oracle для оптимизации работы базы данных.

Важным этапом работы было тестирование производительности базы данных. Были проведены тесты выполнения запросов. Результаты тестирования показали полную работоспособность базы данных.

Было разработано руководство пользователя, которое содержит инструкции по использованию приложения базы данных железнодорожного вокзала. В нем описаны основные функции и возможности базы данных.

Как итог, работа позволила создать удобную в использовании и гибкую в управлении базу данных, которая может быть использована для управления данными о железнодорожных рейсах и пассажирах, а также может быть адаптирована под конкретные потребности пользователей.

СПИСОК ЛИТЕРАТУРЫ

1. Object-Oriented Analysis and Design with Applications [Текст]: Гради Буч [и др.]. – 3-е изд., перераб. и доп. — М.: Издательский дом «Вильямс», 26.02.2020. — 720 с.
2. Design Patterns: Elements of Reusable Object-Oriented Software [Текст]: Э. Гамма [и др.]. — СПб.: Питер, 2001. — 368 с.
3. БЧ. Мой поезд [Электронный ресурс] / Режим доступа:

https://play.google.com/store/apps/details?id=by.rw.client/ – Дата доступа 08.03.2023.

1. Intercity.pl [Электронный ресурс] / Режим доступа:

https://www.intercity.pl/ – Дата доступа 09.03.2023.

1. Жуков А. А. Мониторинг баз данных СПб.: БХВ-Петербург, 2010. 320 с.
2. Development of Railway Station Management Information System Based on Internet of Things / J. Lin [и др.] // IEEE Access. 2019. №1. С. 25-29.
3. Форум StackOverflow [Электронный ресурс] / Справочник Режим доступа:

https://ru.stackoverflow.com/ – Дата доступа 02.04.2023.

1. Оффициальный сайт Oracle [Электронный ресурс] / Справочник Режим доступа:

https://www.oracle.com/ – Дата доступа 6.04.2023.

1. Оффициальная документация Oracle [Электронный ресурс] / Справочник Режим доступа:

https://docs.oracle.com/en/ – Дата доступа 12.04.2023.

1. Информационный портал Oracle-patches [Электронный ресурс] / Справочник Режим доступа:

https://oracle-patches.com/ – Дата доступа 18.04.2023.

1. Статья по Oracle Enterprice manager [Электронный ресурс] / Справочник Режим доступа:

https://habr.com/ru/companies/tinkoff/articles/525436/ – Дата доступа 20.04.2023.

1. Информационный портал Oracle-dba [Электронный ресурс] / Справочник Режим доступа:

https://oracle-dba.ru/docs – Дата доступа 24.04.2023.

1. Документация по Oracle PL/SQL [Электронный ресурс] / Справочник Режим доступа:

https://oracleplsql.ru/ – Дата доступа 26.04.2023.

1. Оффициальный блог Oracle [Электронный ресурс] / Справочник Режим доступа:

https://blogs.oracle.com/ – Дата доступа 01.05.2023.

1. Статья по коллекциям в Oracle [Электронный ресурс] / Справочник Режим доступа:

https://habr.com/ru/articles/254355/ – Дата доступа 04.05.2023.

Приложение

Приложение А