Обзор аналогов

**Приложение «БЧ. Мой поезд» — основная задача бронирование и покупка билетов.** Достоинства: Приятный интерфейс, хорошая система бронирования мест. Интерфейс данного приложения показан на рисунке 1

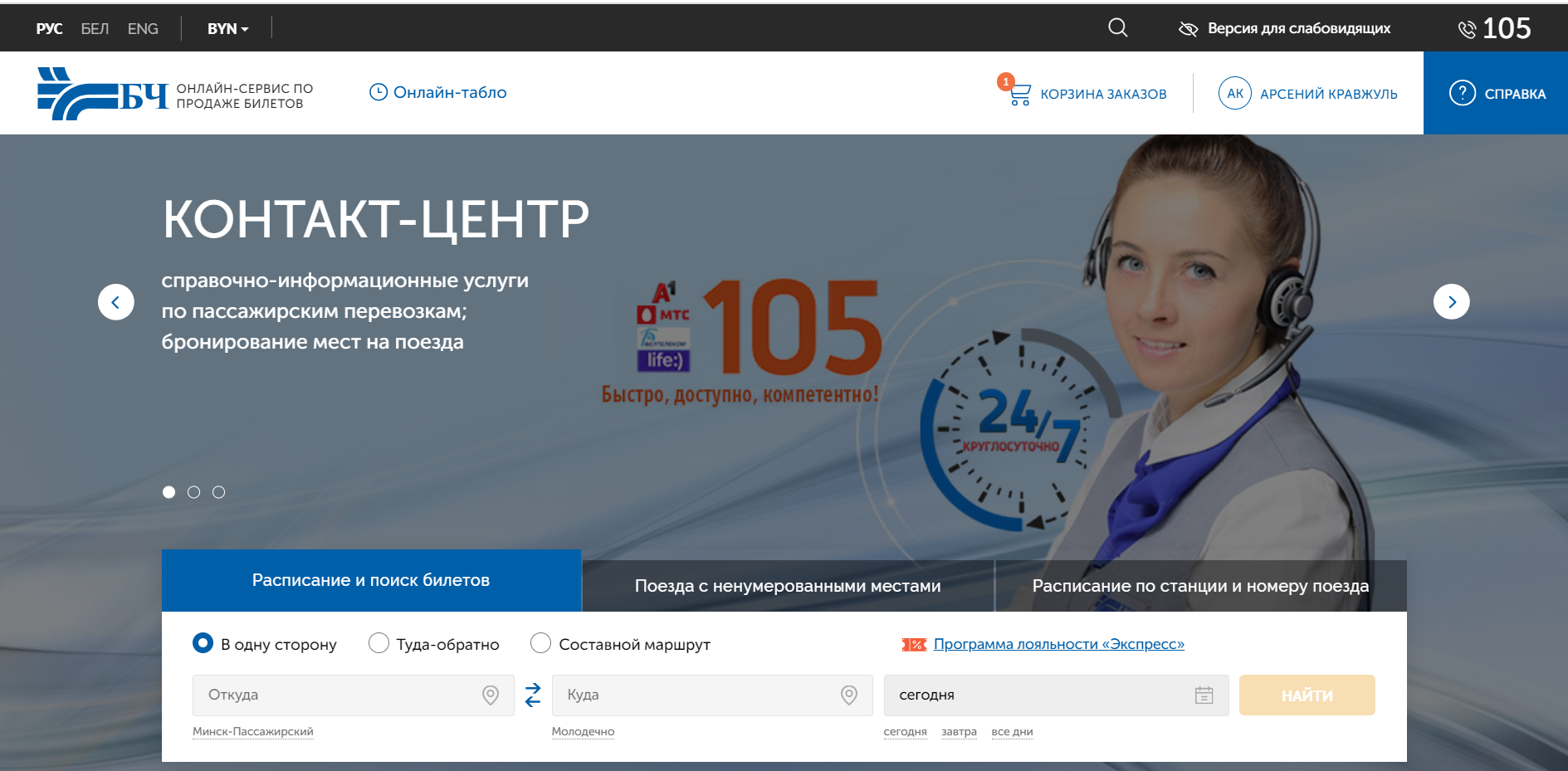


Рисунок 1 – «**БЧ. Мой поезд**»

«**РЖД**» — приложение для бронирования и оплаты билетов. Полностью аналогичен предыдущему прототипу **«БЧ. Мой поезд», имеет такие же достоинства.** Интерфейс данного приложения показан на рисунке 2

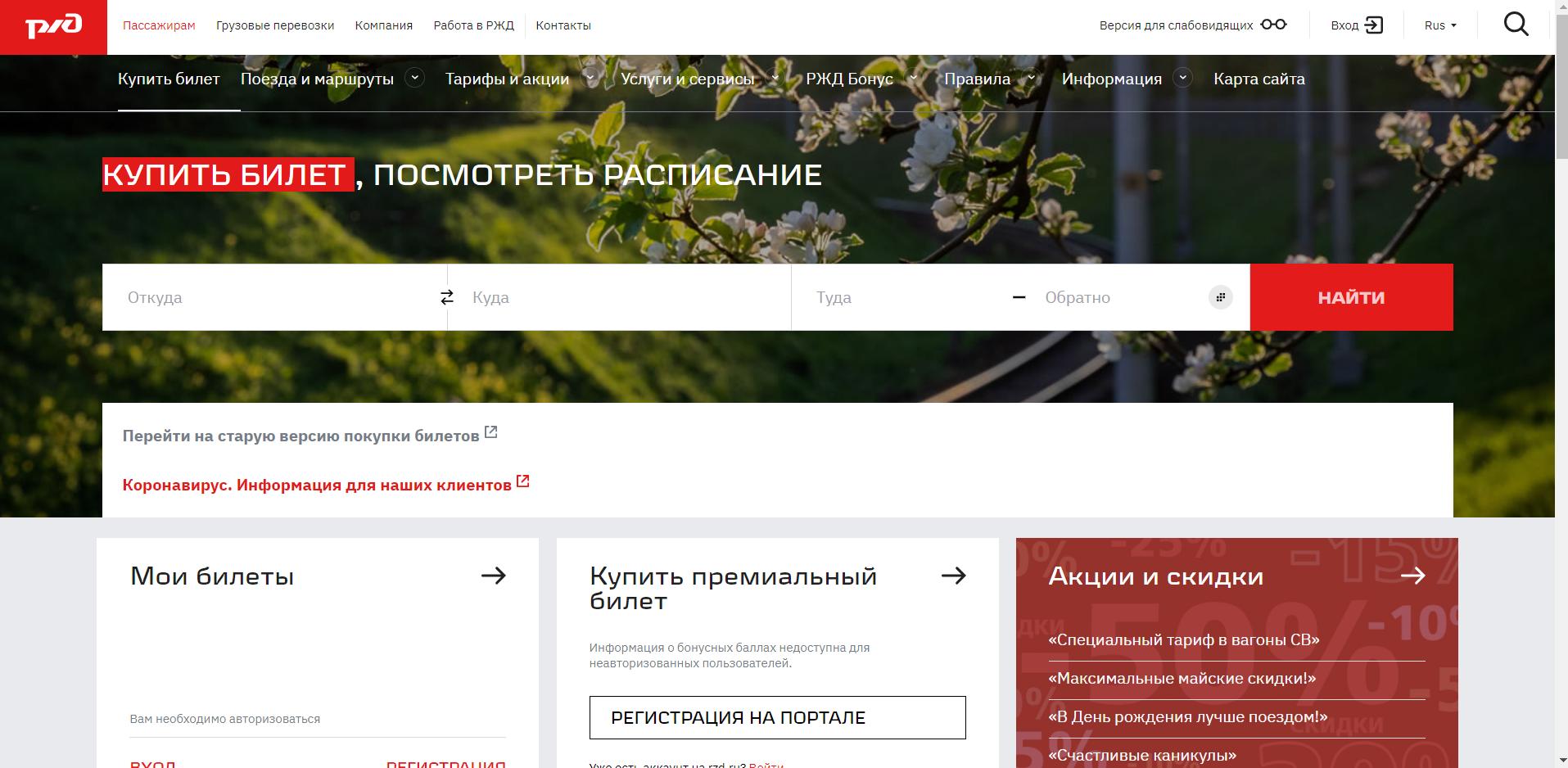


Рисунок 2 – «РЖД»

## Схема базы данных

Реляционная модель

Реляционная модель данных — это способ рассмотрения данных, то есть предписание для способа представления данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов). Она связана с тремя аспектами данных: структурой (объекты), целостностью и обработкой данных (операторы).

В реляционной модели, в отличие от иерархической или сетевой, не существует физических отношений. Вся информация хранится в виде таблиц (отношений), состоящих из рядов и столбцов. А данные двух таблиц связаны общими столбцами, а не физическими ссылками или указателями. Для манипуляций с рядами данных существуют специальные операторы.

В отличие от двух других типов СУБД, в реляционных моделях данных нет необходимости просматривать все указатели, что облегчает выполнение запросов на выборку информации по сравнению с сетевыми и иерархическими СУБД. Это одна из основных причин, почему реляционная модель оказалась более удобна. В реляционной модели, как объекты, так и их отношения представлены только таблицами, и ничем более.

Распространённые реляционные СУБД: Oracle, Sybase, DB2, Ingres, Informix и MS-SQL Server.

Для создания базы данных был выбран Microsoft SQL Server 2019.

Модель базы данных — тип модели данных, которая определяет логическую структуру базы данных и принципиально определяет, каким образом данные могут быть сохранены, организованы и обработаны.

Текущая база данных состоит из восьми таблиц, ее структура представлена на рисунке 3.

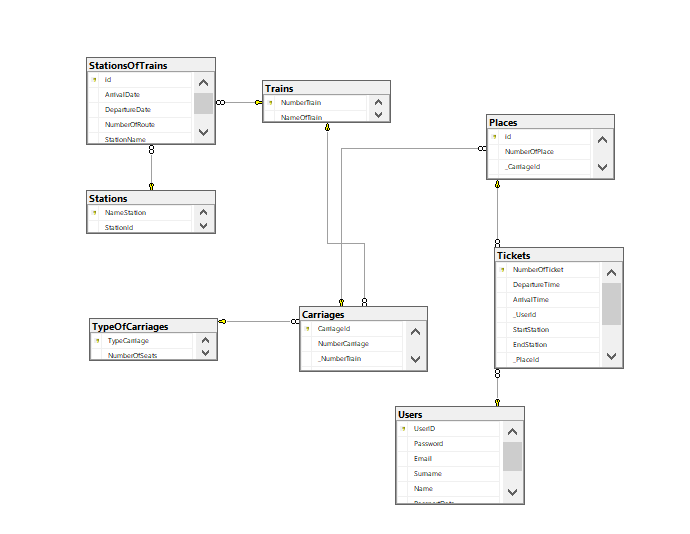


Рисунок 3 – Структура базы данных

На рисунке 4 представлена структура таблицы Users, которая содержит информацию о добавленных пользователях.

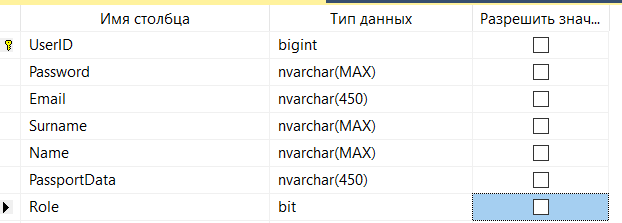


Рисунок 4 – Структура таблицы «Users»

Таблица содержит поля:

* *UserID*– идентификатор пользователя— ключевое поле;
* *Password* – имя пользователя.
* *Email* – Электронная почта
* *Surname* – Фамилия
* *Name* – Имя
* *PassportData* – Паспортные данные
* *Role* – Роль (если true – администратор false – обычный пользователь)

На рисунке 5 представлена структура таблицы TypesOfCarriage, которая содержит информацию о типах вагона.

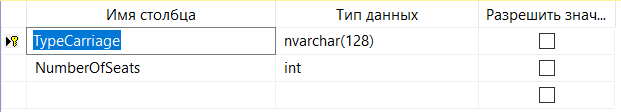


Рисунок 5 – Структура таблицы «TypesOfCarriage»

Таблица содержит поля:

* *TypeCarriage* –  тип вагона – ключевое поле;
* *NumberOfSeats* – количество мест;

На рисунке 6 представлена структура таблицы Carriage, которая содержит информацию о вагонах.

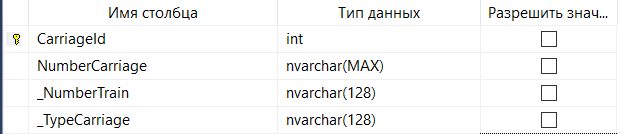


Рисунок 6 – Структура таблицы «Carriage»

Таблица содержит поля:

* *CarriageId* – айди вагона — ключевое поле;
* *NumberCarriage* – номер вагона;
* *\_TypeCarriage* – тип вагона из таблицы TypesOfCarriage;
* *\_NumberTrain* – номер поезда;

На рисунке 7 представлена структура таблицы Train, которая содержит информацию о рейсах поездов.

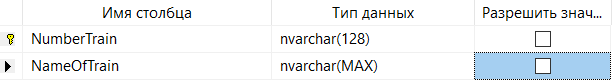


Рисунок 7 – Структура таблицы «Train»

Таблица содержит поля:

* *NumberTrain* – номер поезда – ключевое поле;
* *NameOfTrain* – название поезда;

На рисунке 8 представлена структура таблицы Station, которая содержит информацию о станциях.

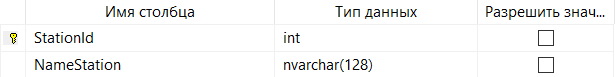


Рисунок 8 – Структура таблицы «Station»

Таблица содержит поля:

* *StationId* – айди станции – ключевое поле;
* *NameStation* – название станции;

На рисунке 9 представлена структура таблицы StationsOfTrain, которая содержит информацию о станциях в конкретном рейсе поезда.

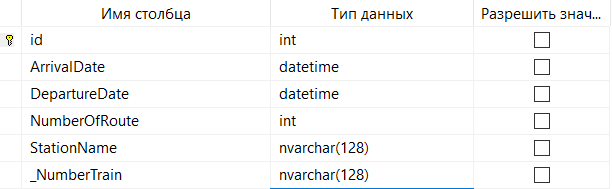


Рисунок 9 – Структура таблицы «StationsOfTrain»

Таблица содержит поля:

* *id* – айди – ключевое поле
* *ArrivalDate* – дата прибытия;
* *DepartureDate* – дата отправления;
* *NumberOfRoute* – номер в маршруте;
* *StationName* – название станции изи таблицы Station;
* *\_NumberTrain* – номер поезда.

На рисунке 10 представлена структура таблицы Place, которая содержит информацию о месте в поезде.

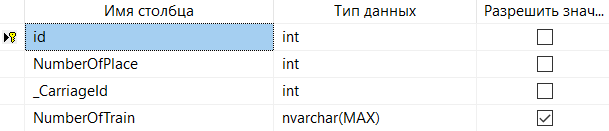


Рисунок 10 – Структура таблицы «Place»

Таблица содержит поля:

* *id* – айди – ключевое поле
* *NumberOfPlace* – номер места;
* *\_CarriageId* – айди вагона из таблицы Carriage;
* *NumberOfTrain* – номер поезда;

На рисунке 11 представлена структура таблицы Tickets, которая содержит информацию о заказе билета.

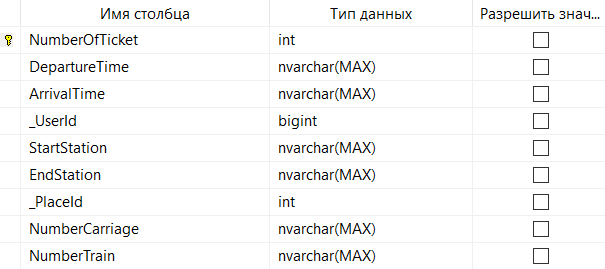


Рисунок 11 – Структура таблицы «Tickets»

Таблица содержит поля:

* *NumberOfTicket* – номер билета – ключевое поле
* *DepartureTime* – время отправления;
* *ArrivalTime* – время прибытия;
* *\_UserId* – айди пользователя из таблицы Users;
* *StartStation* – станция отправления;
* *EndStation* – станция прибытия;
* *\_PlaceId* – айди места из таблицы Places;
* *NumberCarriage* – номер поезда;
* *NumberTrain* – номер поезда;

**Схемы взаимодействия**

Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

В данной программе существует две роли. На диаграмме вариантов использования, которая представлена на рисунке 12, представлена роль администратора.

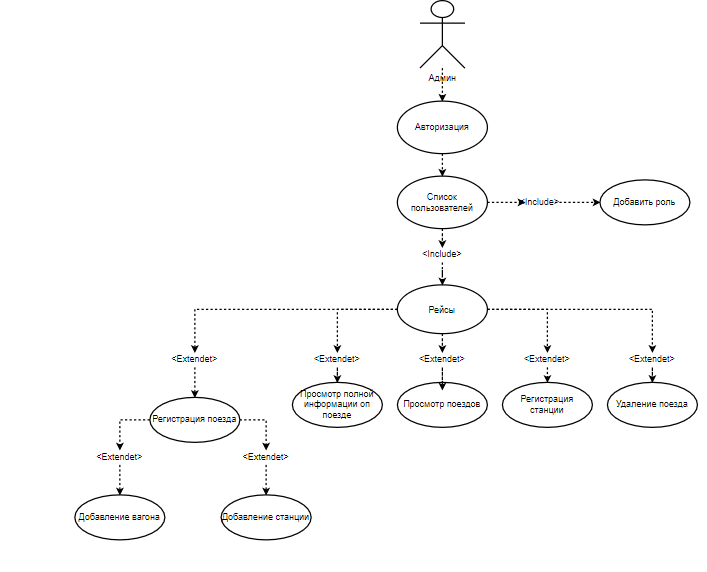


Рисунок 12 – Схема взаимодействия для администратора

На диаграмме вариантов использования, которая представлена на рисунке 13, представлена роль пользователя.

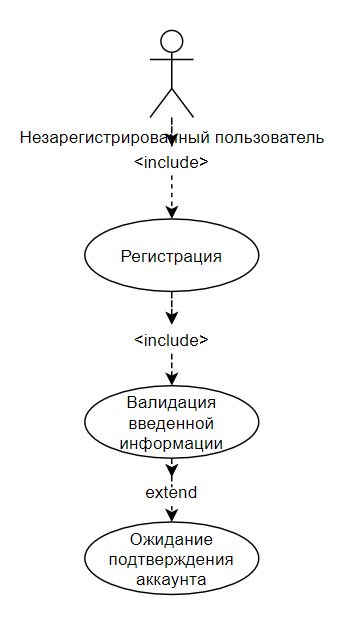
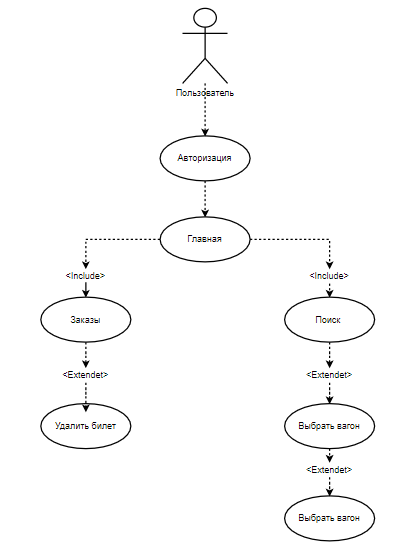


Рисунок 13 – Схема взаимодействия для клиента

**Блок-схема**

Блок-схема – тип схем, описывающих [алгоритмы](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

Блок-схема алгоритма регистрации станции представлена на рисунке 14.

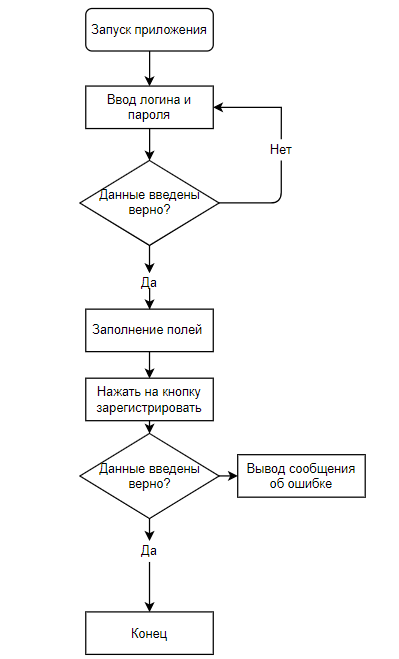


Рисунок 2.2.3 – Блок-схема добавления станции