

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

Институт дополнительного образования

Высшая инженерная школа

Выпускная квалификационная работа

по программе профессиональной переподготовки

«Разработчик прикладного программного обеспечения (язык JAVA)»

«Клиент-серверное приложение, осуществляющее продажу билетов на железнодорожный транспорт»

Выполнил: Исаков А. А.
Руководитель: Колосов К. Д.

Санкт-Петербург
2021

Обзор предметной области

Покупка билетов через интернет имеет множество преимуществ по сравнению с покупкой в кассе:

1. Удобство выбора направления и даты отправки
2. Экономия времени на покупку
3. Сокращение количества социальных контактов
4. Возможность покупки билетов с отправлением из другого города

Цель работы

- Цель ВКР

Создать приложение моделирующее работу информационной системы по продаже билетов на железнодорожный транспорт.

- Задачи

1. Разработать структуру базы данных для хранения данных
2. Разработать структуру классов приложения
3. Разработать пользовательский интерфейс
4. Выполнить программную реализацию
5. Провести тестирование

Требования

Приложение должно обеспечивать следующую функциональность:

1. Для клиентов компании

1. поиск поезда, проходящего от станции А до станции В в заданный промежуток времени
2. расписание поездов по станции
3. покупка билета, если
 - a) есть свободные места
 - b) пассажир с такими же именем, фамилией и датой рождений еще не зарегистрирован в выбранном поезде
 - c) до отправления поезда не менее 10 минут

2. Для сотрудников компании

1. добавление новых станций, поездов
2. просмотр всех зарегистрированных на поезд пассажиров
3. просмотр всех поездов

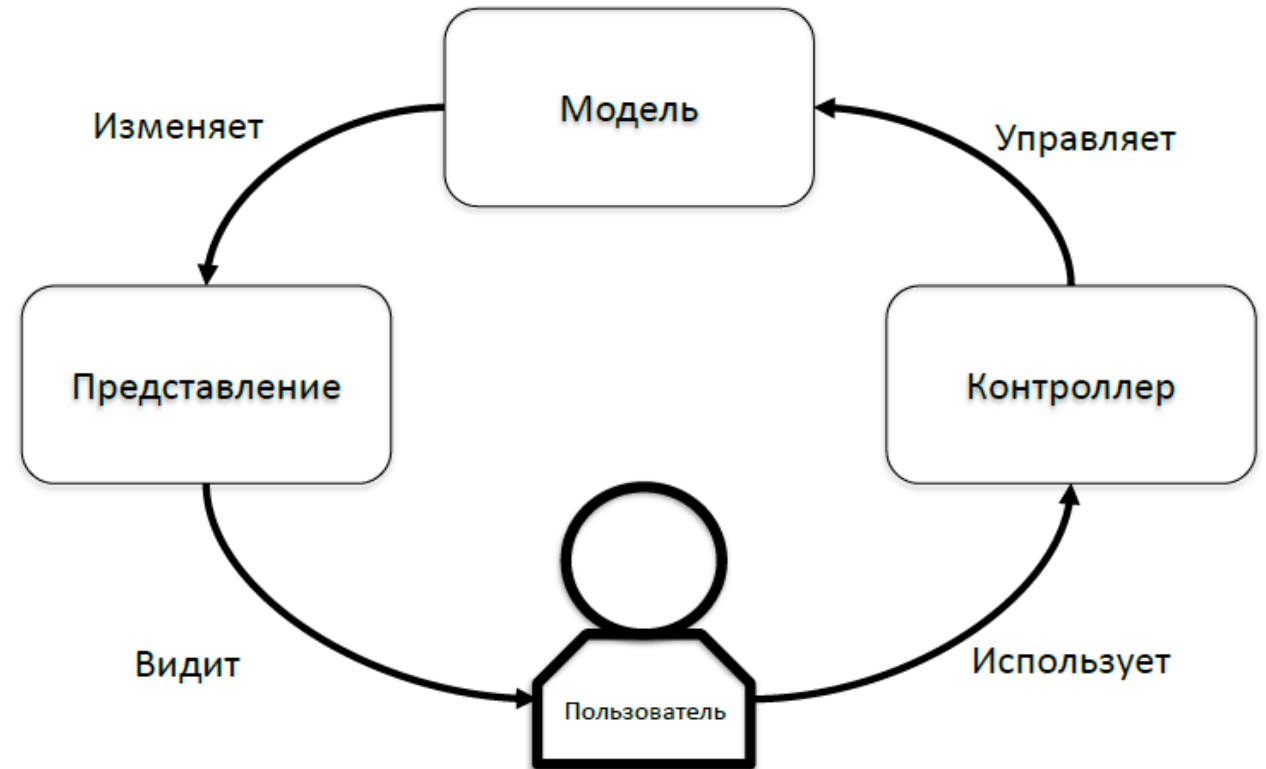
Требования

Нефункциональные требования:

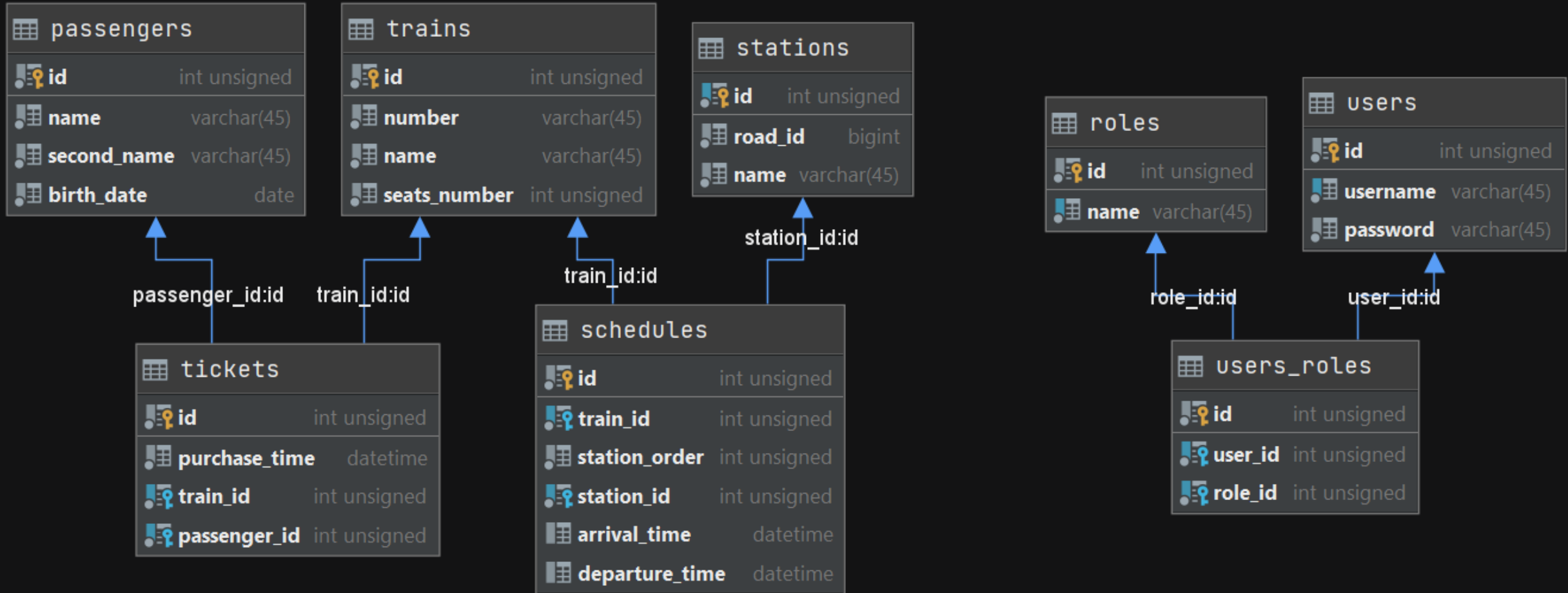
1. Приложение должно быть реализовано на языке Java
2. Язык интерфейса: английский
3. Приложение является многопользовательским, типа клиент-сервер с соединением по сети
4. Приложение должно обрабатывать аппаратные и программные ошибки

Выбор архитектуры

Для построения архитектуры приложения был выбран паттерн проектирования Model-View-Controller (MVC)



Структура базы банных



Структура приложения

Приложение состоит из 7 пакетов:

- config – хранит классы конфигурации
- controller – хранит контроллеры Spring MVC
- dao - хранит классы доступа к БД
- dto – хранит возвращаемые классы
- mappers – хранит мапперы
- model – хранит классы сущности
- service – хранит бизнес логику

Пакет model

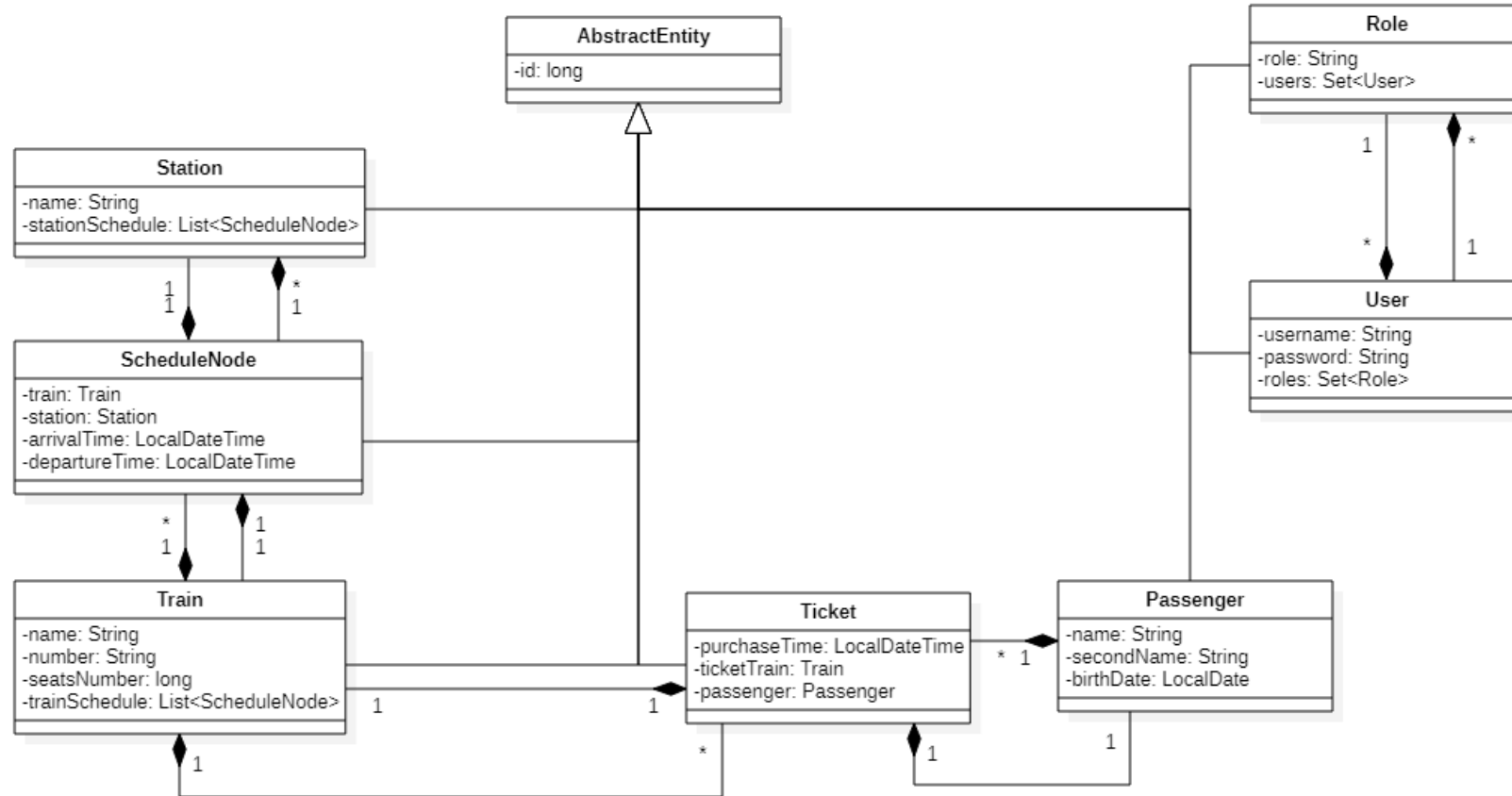


Рис. 1 Структура пакета model

Структура пакетов config и controller

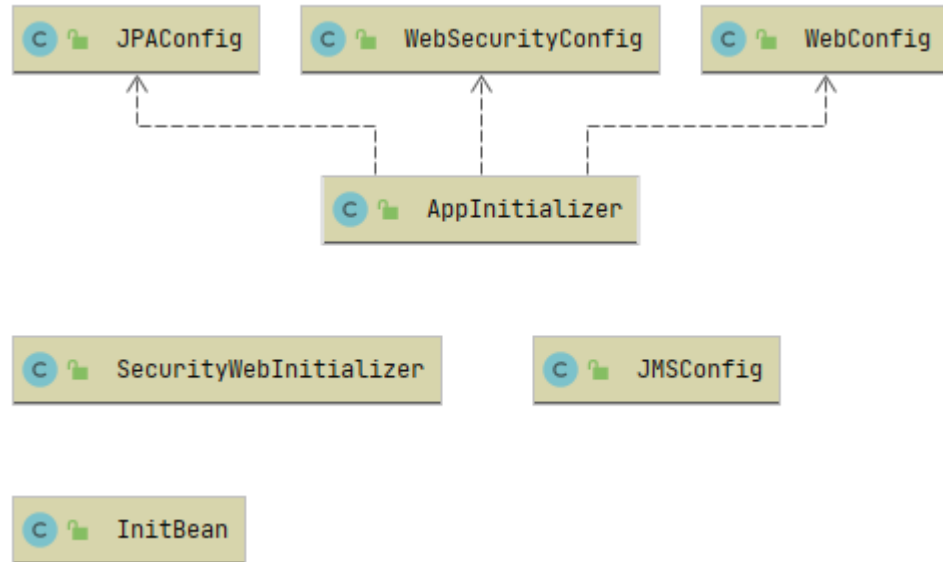


Рис. 2 Структура пакета config

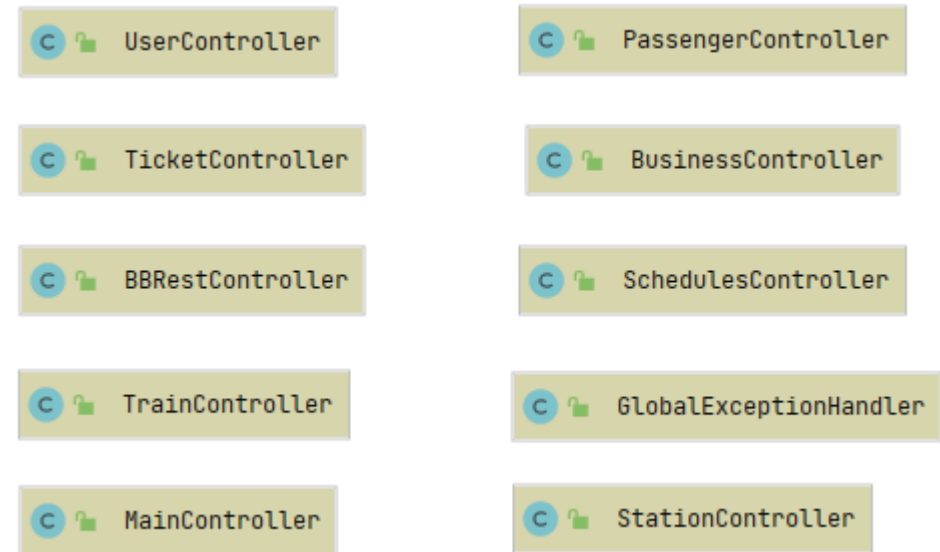


Рис. 3 Структура пакета controller

Пакет service

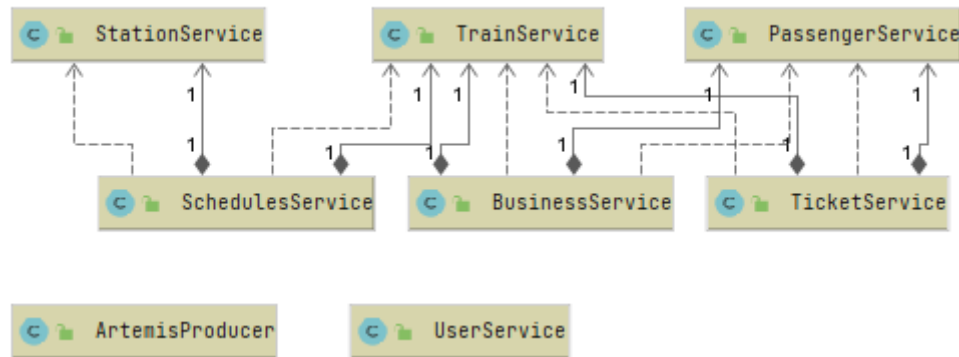


Рис. 4 Структура пакета service

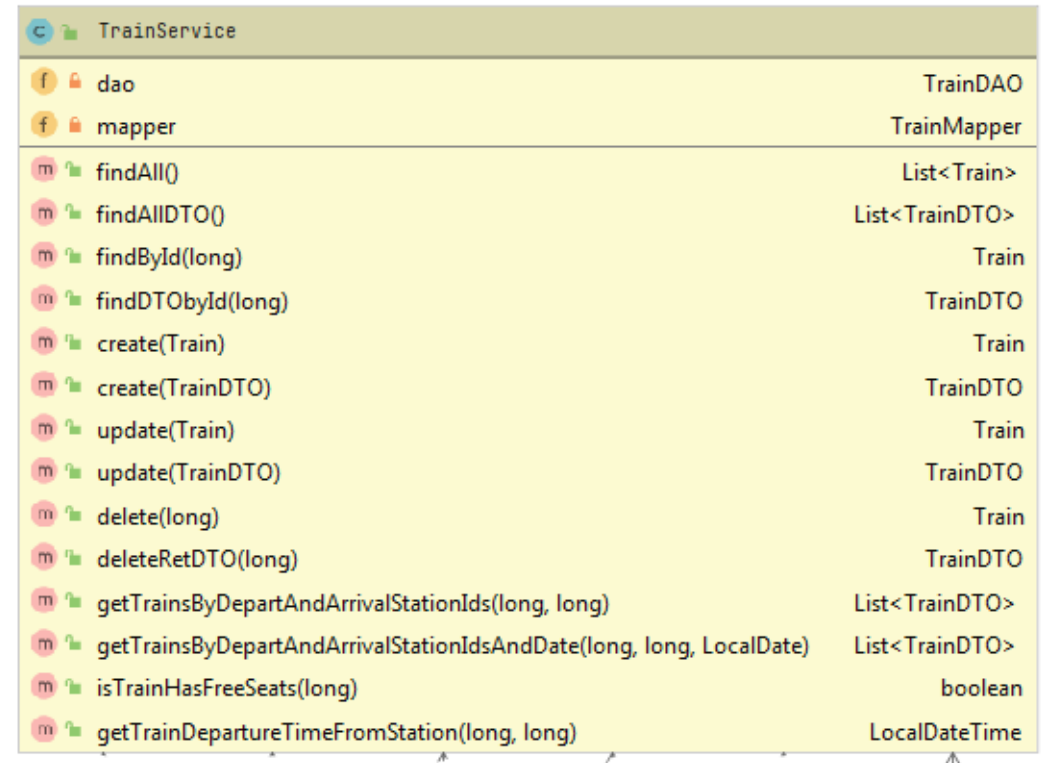


Рис. 5 Класс TrainService

Пример обращения к БД. Класс TrainDAO

```
public List<Train> getTrainsByDepartAndArrivalStationIdsAndDate(long departId, long arrivalId, LocalDate departDate) {
    String departDateS = departDate.toString();
    String query = "SELECT t.id, t.name, t.number, t.seats_number from trains t\n" +
        "INNER JOIN schedules s ON (s.train_id = t.id AND s.station_id = :departId)\n" +
        "INNER JOIN schedules s1 ON (s1.train_id = t.id AND s1.station_id = :arrivalId)\n" +
        "WHERE \n" +
        "s.departure_time < s1.arrival_time \n" +
        "AND \n" +
        "Date(s.departure_time) = :departDate";
    List<Train> trains = null;
    try {
        trains = entityManager.createNativeQuery(query, Train.class)
            .setParameter( name: "departId", departId)
            .setParameter( name: "arrivalId", arrivalId)
            .setParameter( name: "departDate", departDateS)
            .getResultList();
    } catch (Exception e) {
        logger.error(e.toString());
    }
    return trains;
}
```

Рис. 6 Метод DAO из класса TrainDAO

Карта сайта приложения

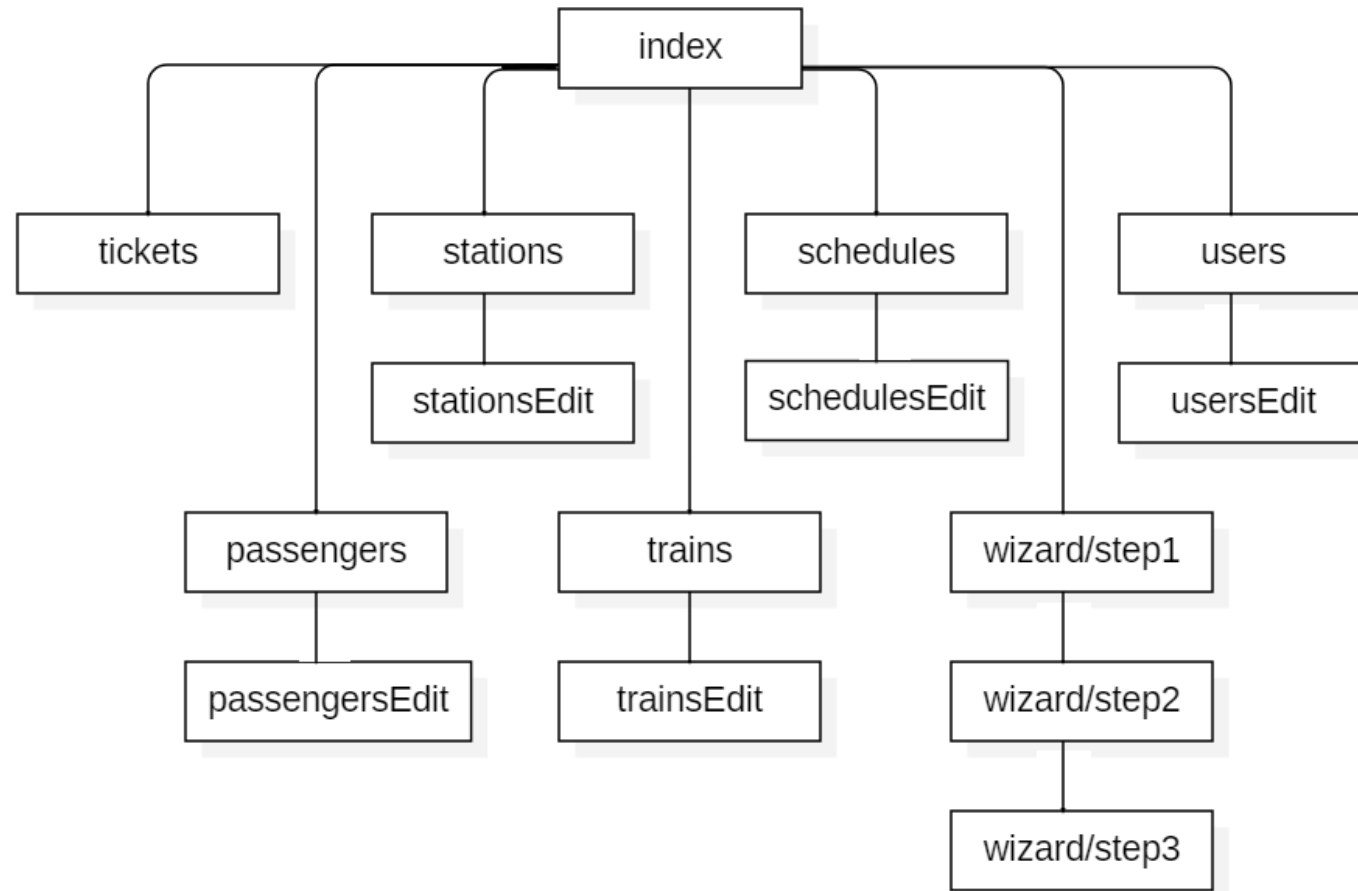


Рис. 7 Карта сайта приложения

Select departure and destination stations, and travel date

Departure Station

Arrival Station

Depart date

Find tickets

Рис. 8 Окно выбора направления и даты

localhost:8080/wizard/trainfinder?departStationId=1&arrivalStationId=7&DepartDate=

SBB Home Buy ticket Stations list Trains list Log in

Number	Name	Seats number				Action
			Station	Arrival date	Departure date	
255A	Long Range train	500	Downlothlake	--:--	2021-03-28 15:32	Select train
			Andganhills	2021-03-28 15:38	2021-03-28 15:40	
			Bluffspar	2021-03-28 15:56	2021-03-28 15:58	
			Central	2021-03-28 16:08	--:--	

Рис. 9 Окно доступных поездов по выбранному направлению

Train №: 255A "Long Range train"

Route: Downlothlake — Central

First name

Aaron

Second name

Henly

Birth date

25.02.1980



Save

Ticket № 30 successfully registered

Рис. 10 Окно заказа билета

Использованные технологии

Область использования	Технологии
Фронтэнд	JSP, Bootstrap, Ajax, JS, jQuery
Бекэнд	Java SE, Spring, Spring MVC, Spring Security, JPA, Hibernate, JMS, ArtemisMQ
База данных	MySQL
Инструменты	Tomcat, Maven, log4j, MapStruct

Заключение

По итогу выполнены все задачи и цель работы:

- Задачи

1. Разработана структура базы данных для хранения данных
2. Разработана структура классов приложения
3. Разработан пользовательский интерфейс
4. Выполнена программная реализация
5. Проведено тестирование

- Цель ВКР

Создано приложение моделирующее работу информационной системы по продаже билетов на железнодорожный транспорт.

Спасибо за внимание!