Technical description

SBB

Author:

Isakov Artem

Saint-Petersburg, 2021

Content

1. Task

To develop web-application that simulates the operation of the information system of a company that carries out passenger railroad transportation. The application have to perform the required user`s cases.

User cases:

1. for a clients:
   - 1.1. search for a train passing from station A to station B at a chosen time interval
   - 1.2. train schedule for the station
   - 1.3. buying a ticket if:
     - 1.3.1. train have are free seats;
     - 1.3.2. passenger with the same name, surname and date of birth has not yet been registered on the selected train;
     - 1.3.3. at least 10 minutes before the train departure;
2. for company employees:
   - 2.1. add new stations and trains;
   - 2.2. view all passengers registered for the train;
   - 2.3. view all;

Additionally develop a separate client application for an electronic billboard. The application should display a list of all trains departing or arriving at the station on the current day.

2. Project goals.

1. The robust, useful and reliable system.
2. Cohesive data model.
3. User-friendly interface.
4. Cohesive data model.
5. Separate access to different system`s part.

3. Application description.

Web-application has two type of user: clients and employees.

Client can view train schedule, station schedule, for a train passing from station A to station B at a chosen time interval, buy tickets.

Employees can add, edit, delete stations, trains, train schedules and new employees. Employees can also view all registered passengers for train, list all passengers.

There is an authentication mechanism in system that control access to portal. Each user in application has access level that display what information he could get and what couldn`t.

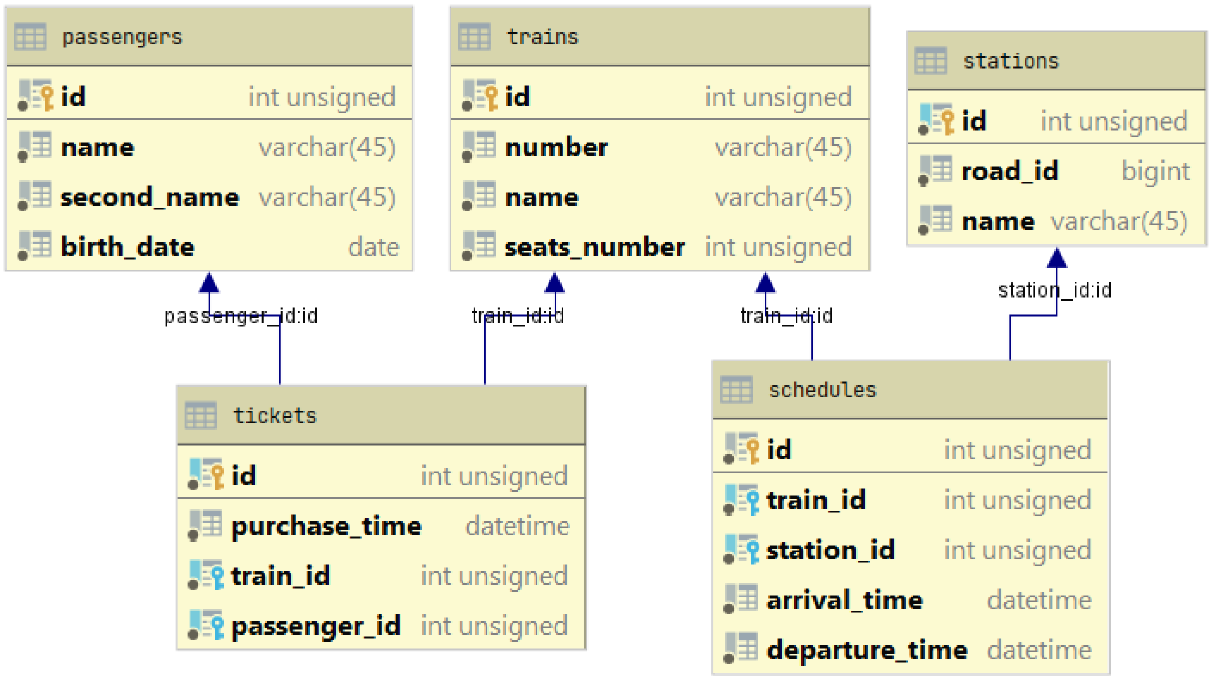
All data store in reliable database.

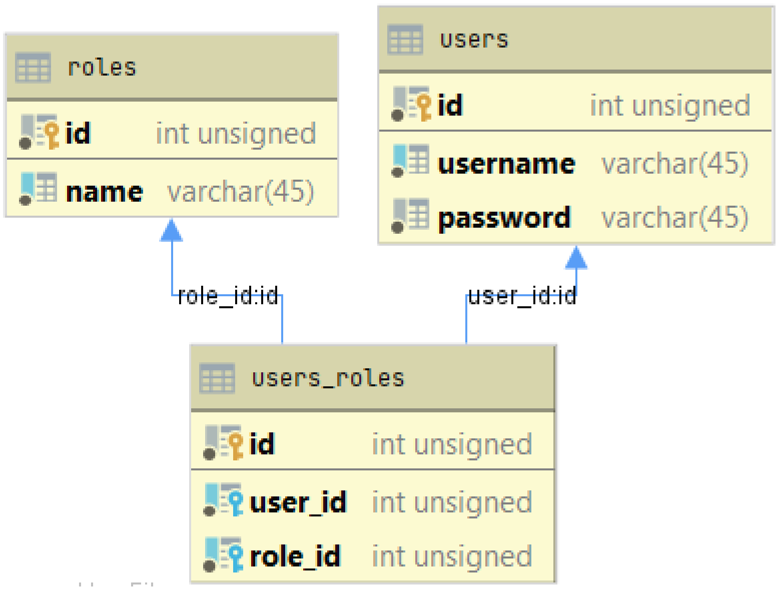
4. Used Technologies.

1. Instruments:
   1.1. IDE - IntelliJ IDEA
   1.2. Maven 3.6.3

2. Technologies:
   2.1. Mapstruct
   2.2. ArtemisMQ
   2.3. Ajax
   2.4. Bootstrap 4.3
   2.5. DB - MySQL 8.0
   2.6. EJB 3
   2.7. Java 8
   2.8. Javascript
   2.9. Jquery
   2.10. JPA 2.0
   2.11. JSF 2.2.13
   2.12. JSP 2.1
   2.13. Junit 4.12
   2.14. Log4j 1.2.17
   2.15. Mockito 1.10.19
   2.16. REST
   2.17. Spring 4.3.9
   2.18. Spring Security 4.2.3
   2.19. Tomcat 8.5.61
   2.20. WildFly 22

5. Database model.



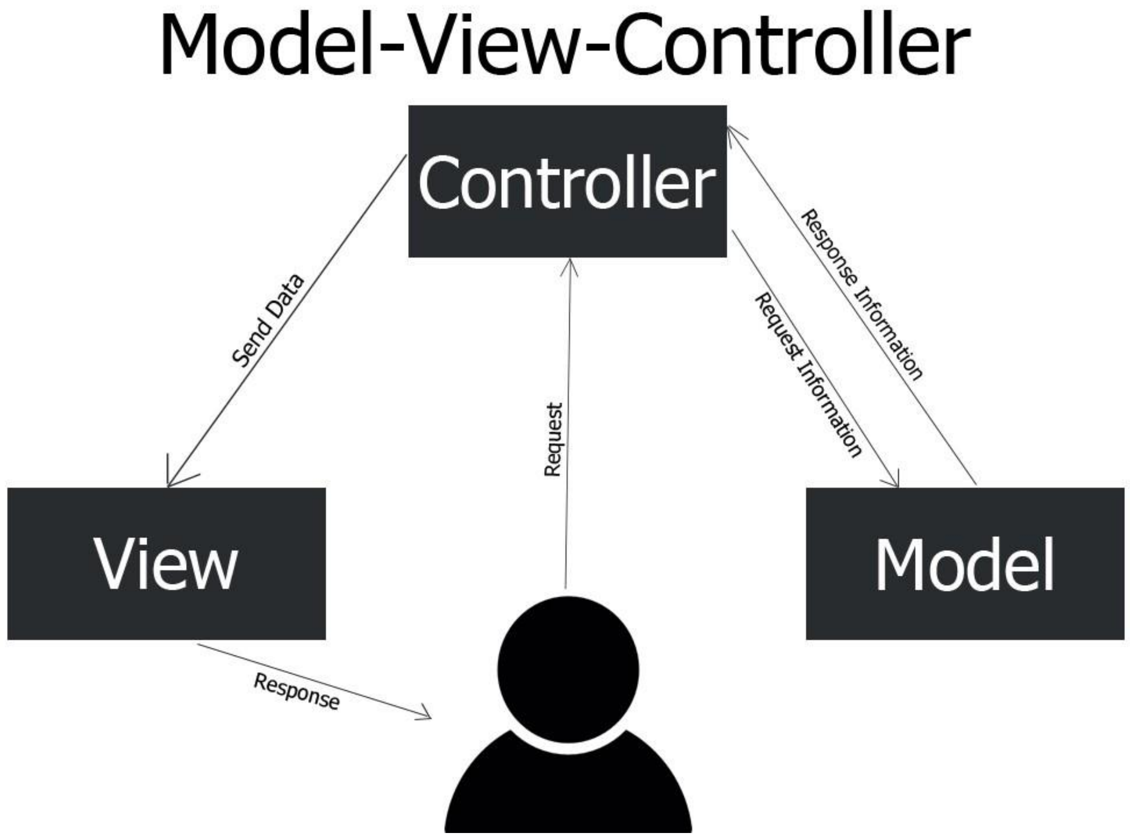


6. System infrastructure.

1. Front-end (browser presentation level):
   1.1. Web-pages - JSP
   1.2. Page-design - CSS
   1.3. Dynamic content - JavaScript, JQuery, Ajax.
2. Back-end (server based level):
   2.1. Application server - Tomcat
   2.2. Database - MySQL
   2.3. Server logic - Spring Framework
3. Bill board application:
   3.1. Web-pages - JSF
   3.2. JMS - ArtemisMQ
   3.3. Application server - WildFly
   3.4. Server logic - EJB
   3.5. WS - REST
   3.6. Web socket

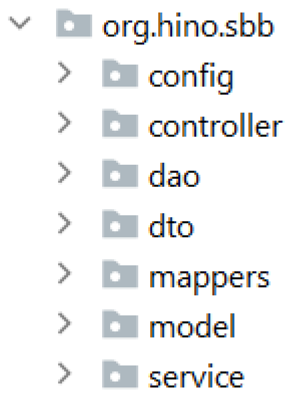
7. System architecture.

Architecture of server-based part presented by MVC - design pattern.

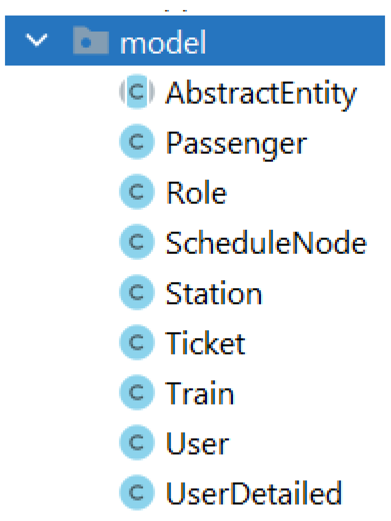


8. Class structure.

According MVC-pattern application has next structure:

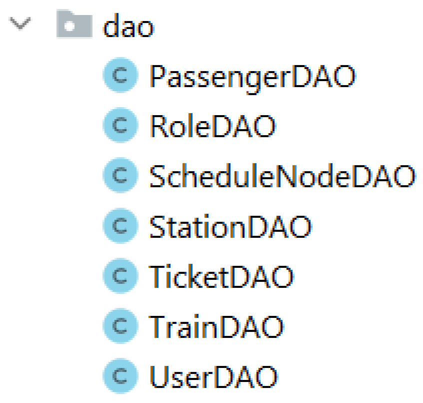

```
org.hino.sbb
    config
    controller
    dao
    dto
    mappers
    model
    service
```

Model level:



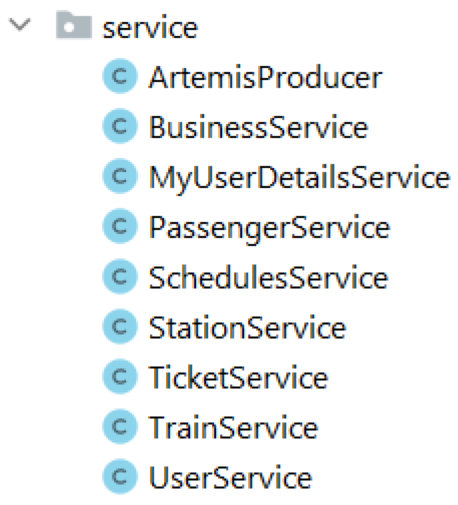
model
AbstractEntity
Passenger
Role
ScheduleNode
Station
Ticket
Train
User
UserDetailed

Model-service level:



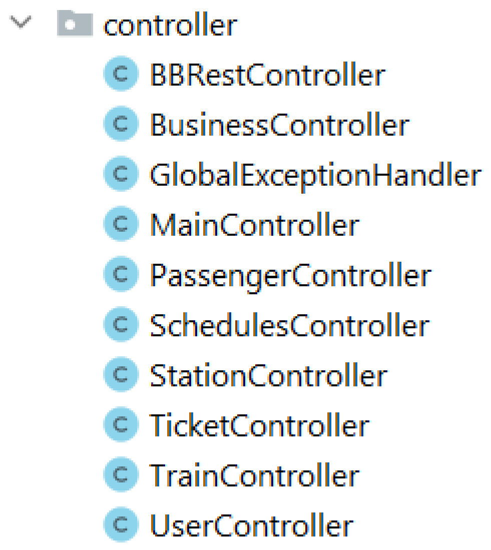
dao
PassengerDAO
RoleDAO
ScheduleNodeDAO
StationDAO
TicketDAO
TrainDAO
UserDAO

Service level:



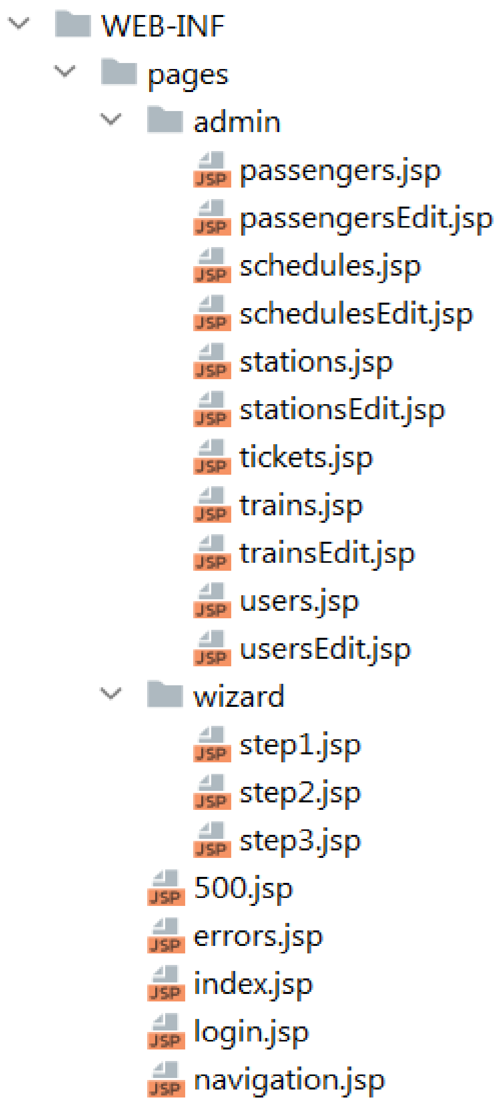
service
ArtemisProducer
BusinessService
MyUserDetailsService
PassengerService
SchedulesService
StationService
TicketService
TrainService
UserService

View-service level:



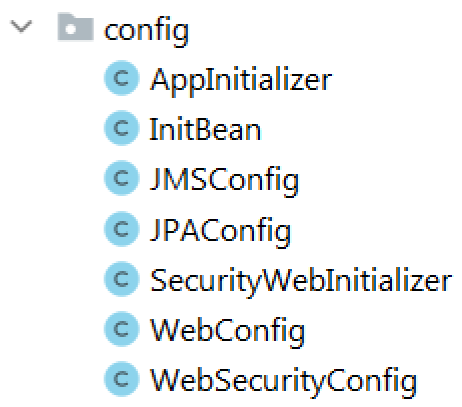
controller
- BBRestController
- BusinessController
- GlobalExceptionHandler
- MainController
- PassengerController
- SchedulesController
- StationController
- TicketController
- TrainController
- UserController

View level:

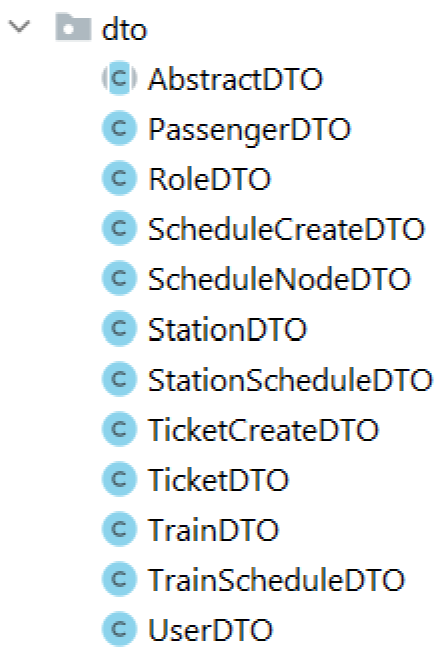


- WEB-INF
  - pages
    - admin
      - passengers.jsp
      - passengersEdit.jsp
      - schedules.jsp
      - schedulesEdit.jsp
      - stations.jsp
      - stationsEdit.jsp
      - tickets.jsp
      - trains.jsp
      - trainsEdit.jsp
      - users.jsp
      - usersEdit.jsp
    - wizard
      - step1.jsp
      - step2.jsp
      - step3.jsp
    - 500.jsp
    - errors.jsp
    - index.jsp
    - login.jsp
    - navigation.jsp

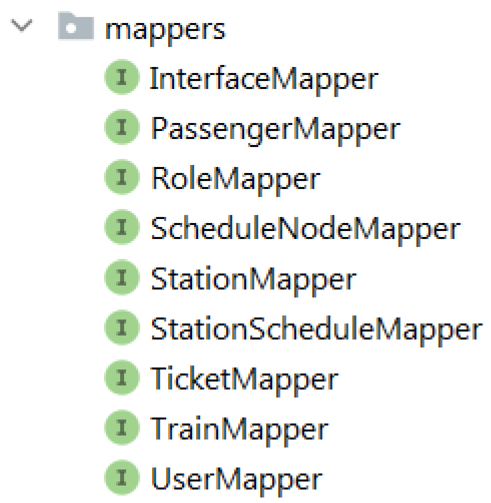
Configuration:


config
AppInitializer
InitBean
JMSConfig
JPAConfig
SecurityWebInitializer
WebConfig
WebSecurityConfig

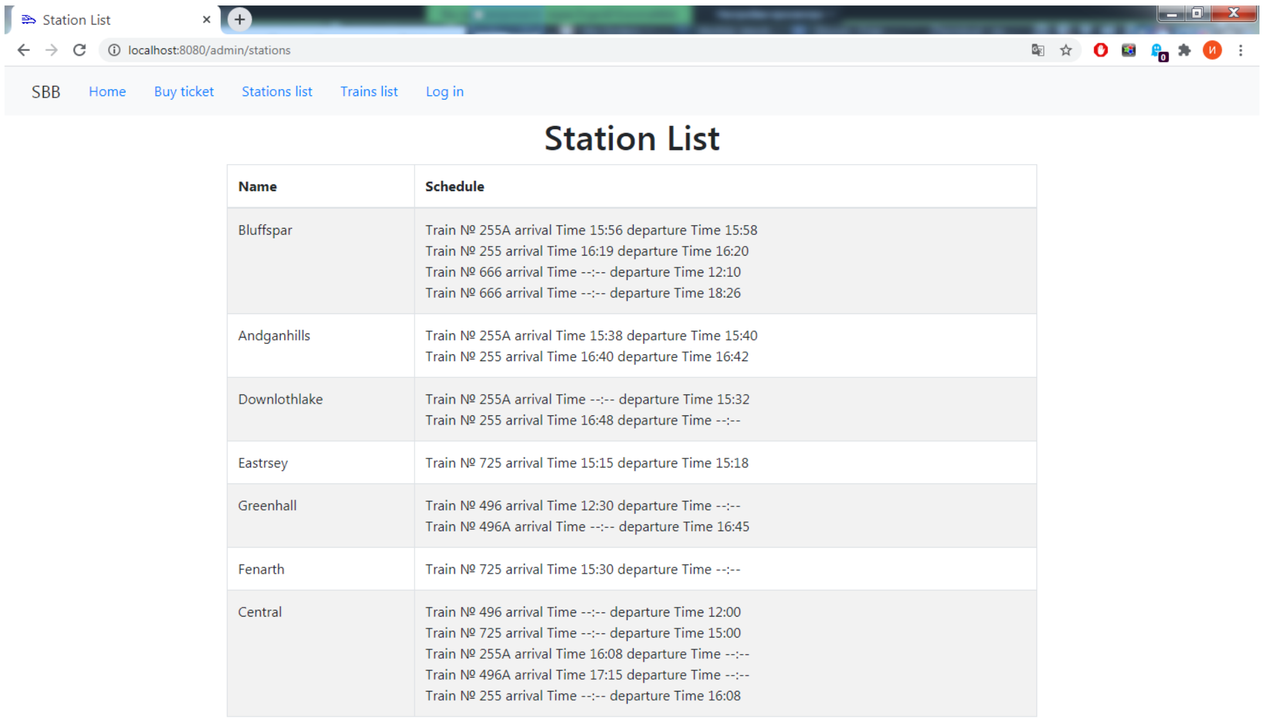
DTO:


dto
AbstractDTO
PassengerDTO
RoleDTO
ScheduleCreateDTO
ScheduleNodeDTO
StationDTO
StationScheduleDTO
TicketCreateDTO
TicketDTO
TrainDTO
TrainScheduleDTO
UserDTO

Mappers:


mappers
InterfaceMapper
PassengerMapper
RoleMapper
ScheduleNodeMapper
StationMapper
StationScheduleMapper
TicketMapper
TrainMapper
UserMapper

9. UI



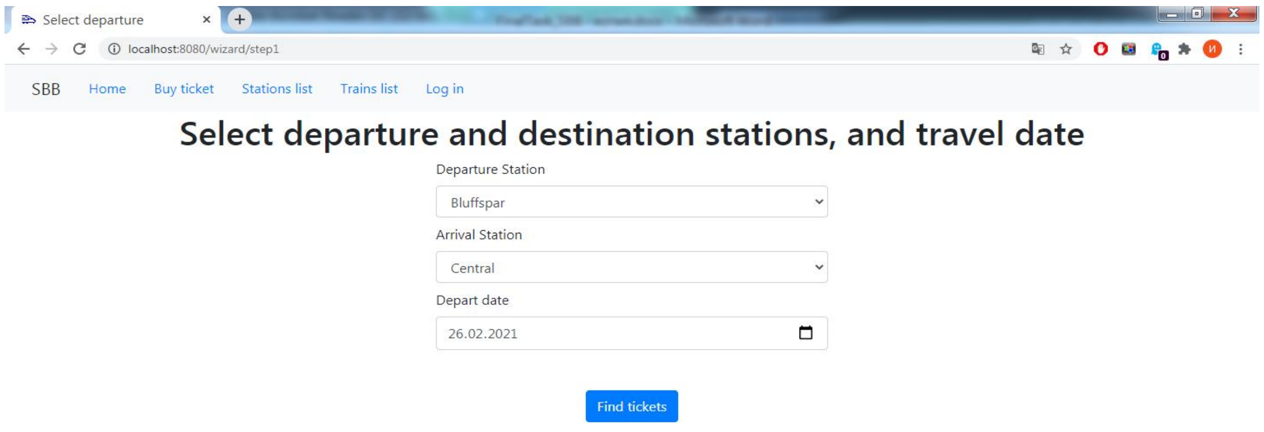
Station List

localhost:8080/admin/stations

Station List

Name	Schedule
Bluffspar	Train № 255A arrival Time 15:56 departure Time 15:58 Train № 255 arrival Time 16:19 departure Time 16:20 Train № 666 arrival Time --:-- departure Time 12:10 Train № 666 arrival Time --:-- departure Time 18:26
Andganhills	Train № 255A arrival Time 15:38 departure Time 15:40 Train № 255 arrival Time 16:40 departure Time 16:42
Downlothlake	Train № 255A arrival Time --:-- departure Time 15:32 Train № 255 arrival Time 16:48 departure Time --:--
Eastrsey	Train № 725 arrival Time 15:15 departure Time 15:18
Greenhall	Train № 496 arrival Time 12:30 departure Time --:-- Train № 496A arrival Time --:-- departure Time 16:45
Fenarth	Train № 725 arrival Time 15:30 departure Time --:--
Central	Train № 496 arrival Time --:-- departure Time 12:00 Train № 725 arrival Time --:-- departure Time 15:00 Train № 255A arrival Time 16:08 departure Time --:-- Train № 496A arrival Time 17:15 departure Time --:-- Train № 255 arrival Time --:-- departure Time 16:08

Fig. 1



Select departure

localhost:8080/wizard/step1

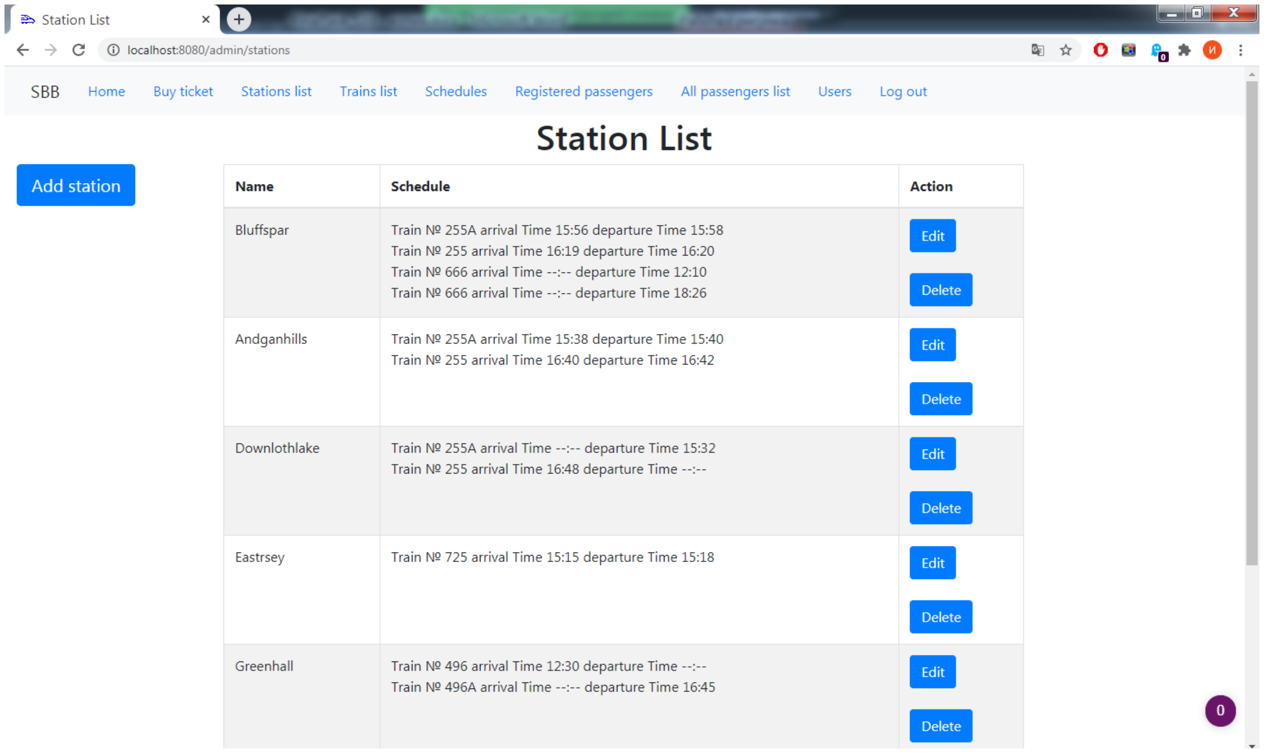
Select departure and destination stations, and travel date

Departure Station: Bluffspar

Arrival Station: Central

Depart date: 26.02.2021

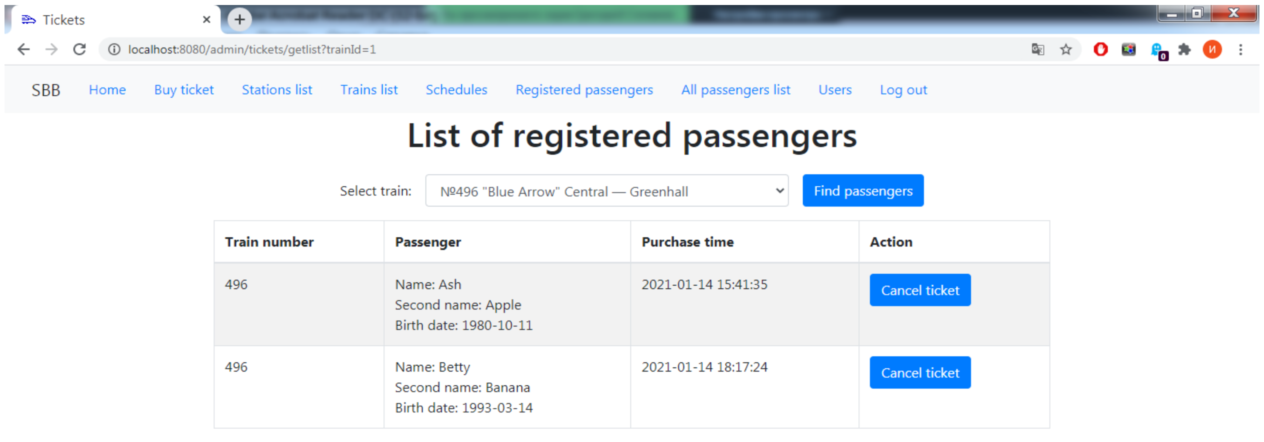
Fig. 2



Station List

Name	Schedule	Action
Bluffspar	Train № 255A arrival Time 15:56 departure Time 15:58 Train № 255 arrival Time 16:19 departure Time 16:20 Train № 666 arrival Time --:-- departure Time 12:10 Train № 666 arrival Time --:-- departure Time 18:26	Edit Delete
Andganhills	Train № 255A arrival Time 15:38 departure Time 15:40 Train № 255 arrival Time 16:40 departure Time 16:42	Edit Delete
Downlothlake	Train № 255A arrival Time --:-- departure Time 15:32 Train № 255 arrival Time 16:48 departure Time --:--	Edit Delete
Eastrsey	Train № 725 arrival Time 15:15 departure Time 15:18	Edit Delete
Greenhall	Train № 496 arrival Time 12:30 departure Time --:-- Train № 496A arrival Time --:-- departure Time 16:45	Edit Delete

Fig. 3



List of registered passengers

Select train: №496 "Blue Arrow" Central — Greenhall

Train number	Passenger	Purchase time	Action
496	Name: Ash Second name: Apple Birth date: 1980-10-11	2021-01-14 15:41:35	Cancel ticket
496	Name: Betty Second name: Banana Birth date: 1993-03-14	2021-01-14 18:17:24	Cancel ticket

Fig. 4

10. Logging.

```
2021-02-24 16:54:02 INFO  AppInitializer:23 - SBB start -------------------------
2021-02-24 16:54:16 INFO  naming:57 - WildFly Naming version 1.0.13.Final
2021-02-24 16:54:16 INFO  security:55 - ELY00001: WildFly Elytron version 1.14.1.Final
2021-02-24 16:54:16 INFO  xnio:95 - XNIO version 3.8.4.Final
2021-02-24 16:54:16 INFO  nio:58 - XNIO NIO Implementation Version 3.8.4.Final
2021-02-24 16:54:16 INFO  threads:52 - JBoss Threads version 2.4.0.Final
2021-02-24 16:54:16 INFO  remoting:99 - JBoss Remoting version 5.0.20.Final
2021-02-24 16:54:23 INFO  ArtemisProducer:29 - jMSProducer created
2021-02-24 16:54:26 INFO  InitBean:24 -  onApplicationEvent running
2021-02-24 16:54:26 INFO  ArtemisProducer:34 - message "init update" was sent to
ActiveMQQueue[jms.queue.SBB]
2021-02-24 16:54:27 INFO  ArtemisProducer:29 - jMSProducer created
2021-02-24 16:54:33 INFO  BBRestController:35 - Sent by REST:
[{"trainNumber":"255A","arrivalTime":"15:56","departureTime":"15:58"},{"trainNumber":"2
55","arrivalTime":"16:19","departureTime":"16:20"},{"trainNumber":"666","arrivalTime":"  -
-:-- ","departureTime":"13:20"}]
```

11. Future improvement.

1. Adding new functionality (payment system, etc.).

2. Refactoring and optimization code.