Technical Solution Description

CONTENT

DESCRIPTION. Application “UberBahn” represents an information system of a company, which provides passenger rail transportation. It allows adding stations, routes and getting information about trains and registered passengers for its employees. Application provides clients with information about trains, station timetables and gives opportunity to buy tickets.

USED TECHNOLOGIES AND FRAMEWORKS.

IDE – IntelliJ Idea 16.1

JDK 1.8

Apache Tomcat 8.5.2

Servlet/JSP

MySQL Server 5.7

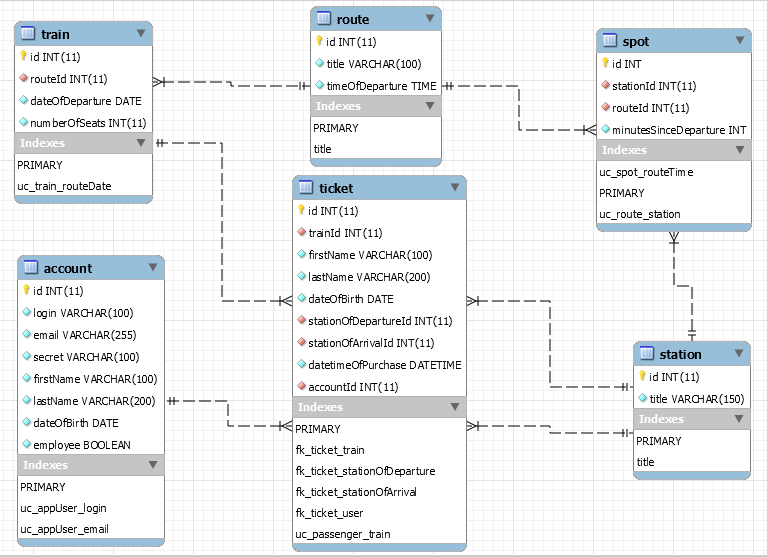
Apache Mavev 3.3.9

JPA 2.0 / Hibernate 5.1

HTML, CSS

JavaScript, jQuery

SCHEME OD DATABASE.



The scheme of database consists of 6 tables: train, route, spot, station, ticket and account.

Each table includes primary key – id.

Table station contains information about station titles (unique), table route contains information about route title (unique) and time of departure, table account includes login (unique), email (unique), secret (or password), first name, last name, date of birth and information whether a user is an employee or not.

Table train contains information about route (relation between table train and table route Many-To-One), date of departure and number of seats. Relation between tables route and station Many-To-Many, to resolve this problem table spot is added, which includes routeId and stationId (relation between tables spot and route or spot and station Many-To-One), minutes since departure. Unique constrains are routeId and stationId, routeId and time since departure. Table ticket provides information about train (relation between tables ticket and train Many-To-One), passenger first name, last name, date of birth (unique constraint), stations of departure and arrival (relation between tables ticket and station Many-To-One), date and time of purchase and account (relation between tables ticket and account Many-To-One).

MODULES OF THE APPLICATION. Application implements 3-tier architecture. View-tier represented by controllers, JSP; Service-tier by services, transports; Data-tier by repositories, entities.

FUNCTIONALITY.

For clients:

* find train from station A to station B;
* get timetable per each station
* buy ticket (include train search)

For employees:

* add new stations, routes and trains
* find trains by route
* view registered passengers (include train search)

IMPROVEMENT IN THE NEXT RELEASES

- Exception handling and logging

- Unit-test

- UI for different types of users

- Dependency Injection pattern