Parking Lot Specification

Author: Amir Hadzhiev

Non-Functional Requirements:

The project shall fulfill all requirements requested from the client.

The project will include the use of Spring, JPA, Hibernate, Html, CSS, ThymeLeaf, MySQL, Intellij (Java Version 17).

The system must be user-friendly and provide information for all events which could be triggered. The application will be web-orientated and must target a fast response time.

Functional Requirements:

This section provides detailed information, on how specific events shall behave when triggered from the user. The specification consists of all features which are requested from the client and shall be implemented by the developer.

1.0 Parking tasks

- ✓ 1.1 Insert - The button Add Parking must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, save the new information with an Auto Incremented Id into the database. If data is not valid, the application must return a negative response.

- ✓ 1.2 Update - The button Update must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, update the new information on the specific Id item from the database. If data is not valid, the application must return a negative response.

- ✓ 1.3 Delete - The button Delete Parking must “fire” event which, deletes item with specific id from the database. The functionality must check if the id requested from the user is valid. If id is not valid the application must return a negative response.

- ✓ 1.4 Show parking by id and Show parking zones by parking id - The button Park Info combines these two tasks and trigger the expected action, providing information regarding parking’s ids and zones.

2.0 Parking Zone tasks

- ✓ 2.1 Insert - The button Add Parking Zone must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, save the new information with an Auto Incremented Id into the database. If data is not valid, the application must return a negative response.

- ✓ 2.2 Update - The button Update must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, update the new information on the specific Id item from the database. If data is not valid, the application must return a negative response.

- ✓ 2.3 The button Delete Parking Zone must “fire” event which, deletes item with specific id from the database. The functionality must check if the id requested from the user is valid. If ID is not valid the application must return a negative response.

- ✓ 2.4 Show Zone by id and Show parking places by zone id - The button Park Info combines these two tasks and trigger the expected action, providing information regarding parking’s ids and zones.

3.0 Parking Place tasks

- ✓ 3.1 Insert - The button Add Parking Place must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, save the new information with an Auto Incremented Id into the database. If data is not valid, the application must return a negative response.

- ✓ 3.2 Update - The button Update must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, update the new information on the specific Id item from the database. If data is not valid, the application must return a negative response.

- ✓ 3.3 Delete - The button Delete Parking Place must “fire” event which, deletes item with specific id from the database. The functionality must check if the id requested from the user is valid. If ID is not valid the application must return a negative response.

- ✓ 3.4 Show parking place by id and Show car by parking place id - The button Parking Place Info combines these two tasks and triggering the expected action.

4.0 Car tasks

- ✓ 4.1 Insert - The button Add Car must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, save the new information with an Auto Incremented Id into the database. If data is not valid, the application must return a negative response.

- ✓ 4.2 Update - The button Update Car must “fire” event which, checks if user has fulfilled all required input field boxes, and if so, update the new information on the specific Id item from the database. If data is not valid, the application must return a negative response.

- ✓ 4.3 Delete - The button Delete Car must “fire” event which, deletes item with specific id from the database. The functionality must check if the id requested from the user is valid. If id is not valid the application must return a negative response.

- ✓ 4.4 Show specific car by indicating id. The application shall provide information including the car item, parking place, parking zone and parking all retrieved from the database. If the requested id is invalid, response must be negative.

5.0 Parking a car

- ✓ 5.1 Park Car - The button Park Car must “fire” event which, checks if user has fulfilled all required input field boxes. The user must provide specific car id and indicate on which parking the car must be parked. The user shall click “Validate Input”, which will trigger a select box indicating all parking zones in requested parking. After the user has indicated a zone, by clicking “Validate Input”, the application will list all available parking places. If the

user has chosen a parking by id and fire “Park Car” event which shall trigger a writing into the database with the specific information where the car is parked. If data is not valid a negative response must be returned.

- ✓ 5.2 Unpark Car – The button Unpark Car must “fire” events which validates the indicated car id. If car id is valid, the information regarding where the car is park shall be erased and the car shall be set to “not parked”. If id is not valid or car is not parked response shall be negative.

Applications properties are:

```
Data Source Properties
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/parking_lot?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=123456789
#JPA Properties
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.format_sql = TRUE
spring.jpa.hibernate.ddl-auto = update
spring.jpa.open-in-view=false
```