Software architecture design

Taxi Service System is web-based tool based on Java Enterprise Edition platform that should be deployed on the Application Server. So it was decided to deploy our application on Wildfly (formely JBOSS) Application Server v9.0.CR1.

In order to package our application to war archive we used Maven is a build automation tool.

For data persistence we used RDB called PostgreSQL v9.4 with PostGIS extension for storing geometric information for our system.

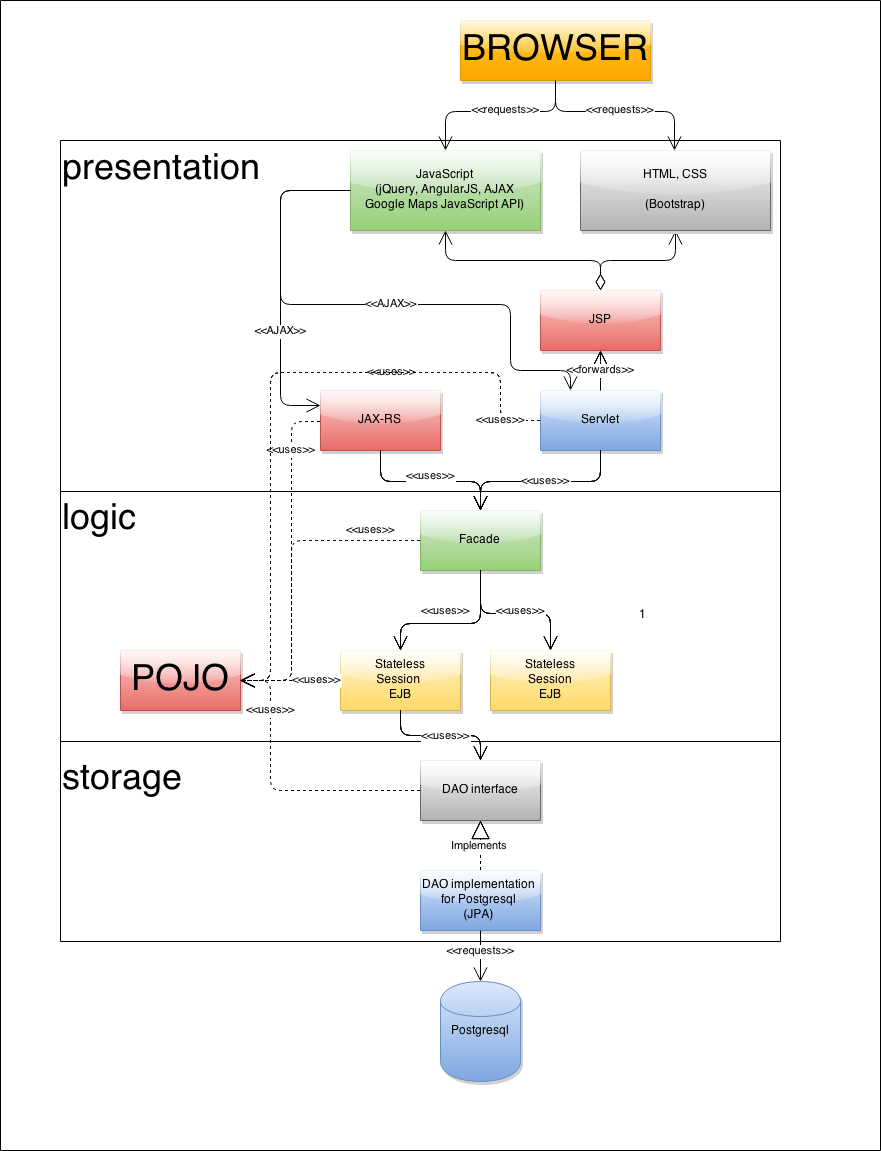
We used N-Tiered development model in our application. So I decided to separate our system order to make tiers to have low cohesion.

It was decided to separate system in 3 Maven modules:

1) Presentation module – Presentation layer module

2) Logic module – business logic layer module

3) Storage module – data representation module



# Presentation module requirements

1) Scalability and low cohesion:

a) JSP pages used for View

b) Servlets and JAX-RS RESTful web services used as Controllers.

c) POJO used as model in order to provide easy JSON conversion possibility.

2) Security and role administration. In order to implement this requirement JAAS security domain was configured at Wildly. The configurations jboss-war.xml.

3) Fast access to system functionality. In order to implement this requirement taxi order page link can be accessed ASAP. from the main page or via URL without any registration. To prevent data overhead and slow loading webpages perform AJAX requests for JSON objects. jQuery and AngularJS JavaScript libraries was used for AJAX.

4) Attractive design. The system should provide attractive view for taxi service system users in order to increase system popularity. The view uses bootstrap css styles in order to make webpages attractive and adaptive (accessible on any device).

# Logic module requirements

1) Full business operation stack support. This requirement was implemented using EJB 3.2 technology.

2) High scalability and low coupling with other system modules. In order to implement these requirement 5 system facades was created:

1. Driver Facade;
2. Customer Facade;
3. Order Facade;
4. Admin Facade;
5. Report Facade.

These Stateless Session Beans are used as entering point to all application logic from the presentation module.

3) Data validation. To implement this requirement regular expressions and Java EE Bean Validation was used.

# Storage module requirements

1) Safe data persistence mechanism. The biggest part of this requirement is delegated to the PostgreSQL database. Except the PostgreSQL, in order to provide save ACID transactions JTA CMP has been used.

2) Providing realization independent object-oriented interface. DAO pattern was implemented in order to encapsulate the persistence type. And to simplify mapping of database tables JPA native SQL API was used for object relation mapping.

3) Possibility to save geometric types. This requirement was implemented using PostGIS extension for PostgreSQL database. For ORM Hibernate Spatial with JTS datatypes was used.