

Group name

QueerCars

Group members

Viktor Mellgren	890309-5274	mviktor@student.chalmers.se
Björn Johansson	840716-7496	bjorn.johansson@gmail.com
Anna Nyström	850903-5906	anna@binbop.se
Jonas Bornold	871006-3275	johonas@gmail.com

General overview over the system

What is this? In which area is the system supposed to be used.

What is it supposed to do? Etc.

The project is about creating a webservice to rent cars.

Possible user roles

Administrators have permissions to edit most aspects of the system such as cars, active rentals e.t.c. as detailed below.

Customers have permissions to create personal accounts and create rentals coupled to their account.

Use cases

Users can register a new account.

Users can log in with their credentials.

Users can view available cars.

Users can view information about the available cars based on the model of the car.

Users can view terms and services.

Users can view service information with pick-up location and pricing.

Users can view news as posted by the staff.

Administrators can log in with their credentials.

Administrators can create new cars.

Administrators can edit existing cars.

Administrators can remove cars.

Administrators can create users.

Administrators can edit users.

Administrators can remove users.

Administrators can add items to the news feed.

Administrators can edit items in the news feed.

Administrators can remove items from the news feed.

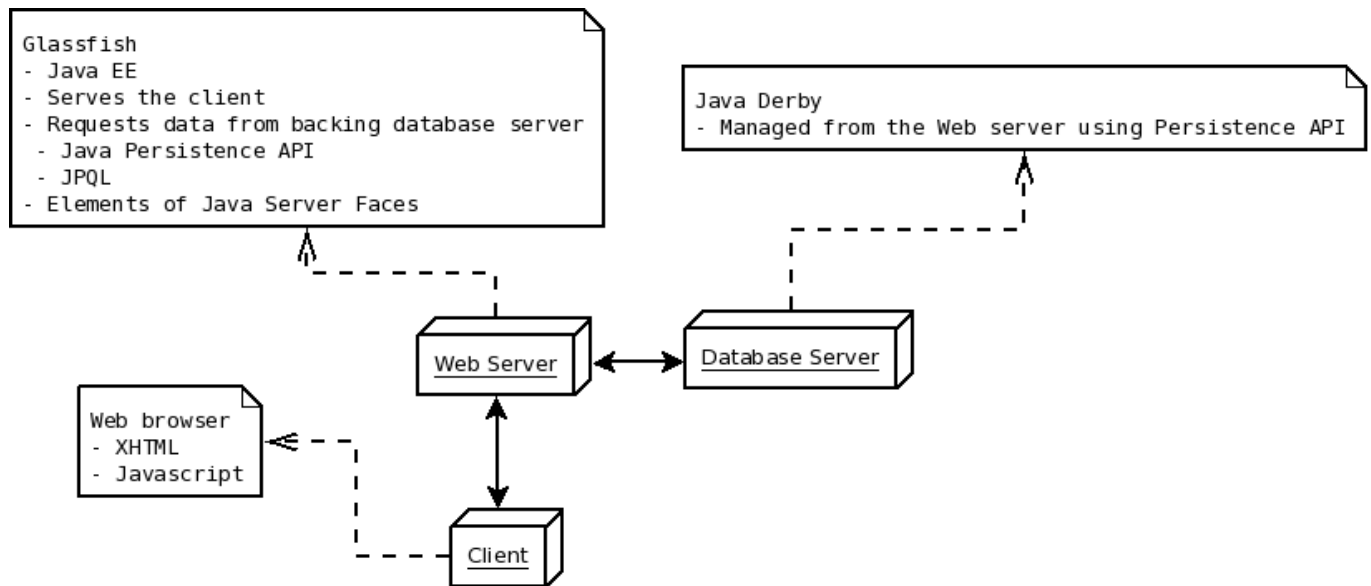
Administrators can create new administrator accounts.

Administrators can edit existing administrator accounts.

Administrators can remove administrator accounts.

Technical design of the system in the following order (UML where appropriate)

Physical set up (tiers)



Participating software components

Client tier

Provides a user interface to the customers and administrators for the application and displays data provided from in addition to facilitating modifications to the system in a user-friendly manner. Communicates with the Web server.

Web tier

Provides backing logic for the client interface. The web tier queries the database tier and interprets the returned data into meaningful formatting for the client tier. This tier also submits modifications requested from the client tier to the Database tier.

Database tier

Provides backing persistent data storage that holds the state of the application and its users, cars active rentals e.t.c.

Packages

edu.chl.queercars

Contains the various Entities in use in the application such as Cars, Customers, Administrators and Rentals.

edu.chl.queercars.administrativeTools

Contains a mail handler used for informing customers and staff about rentals.

edu.chl.queercars.beans

Contains Beans used in the JSF components of the application.

edu.chl.queercars.dbhandlers

Contains Database handlers that provide easy interfaces to the database for the rest of the application.

edu.chl.queercars.servlets

Contains the applications servlet classes which provide interfaces to the client tier of the application. These servlets are generally called using HTTP-methods.

edu.chl.queercars.validation

Validators used to validate inputs from the user and ensure that input data is as expected.

Layered view

View (Browser)

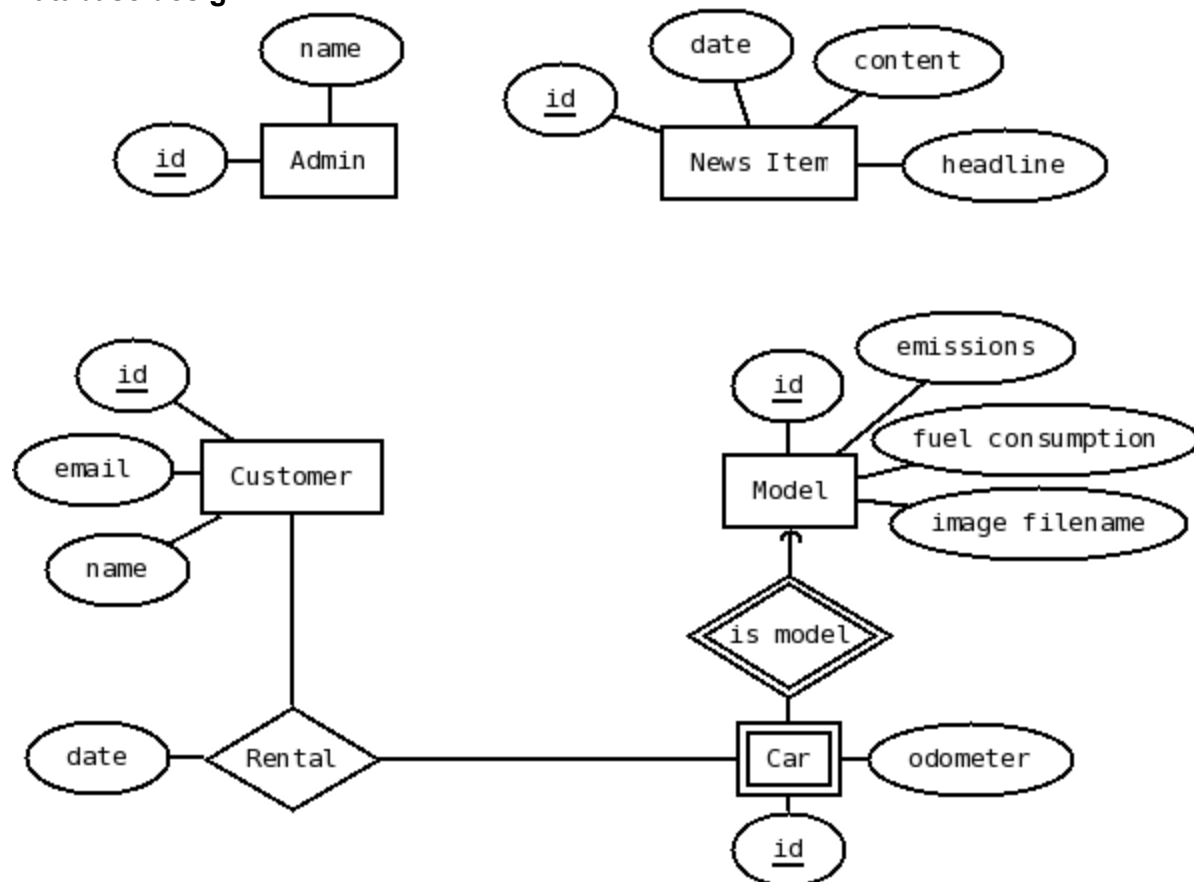
Control (Servlets)

Services (Database Handlers)

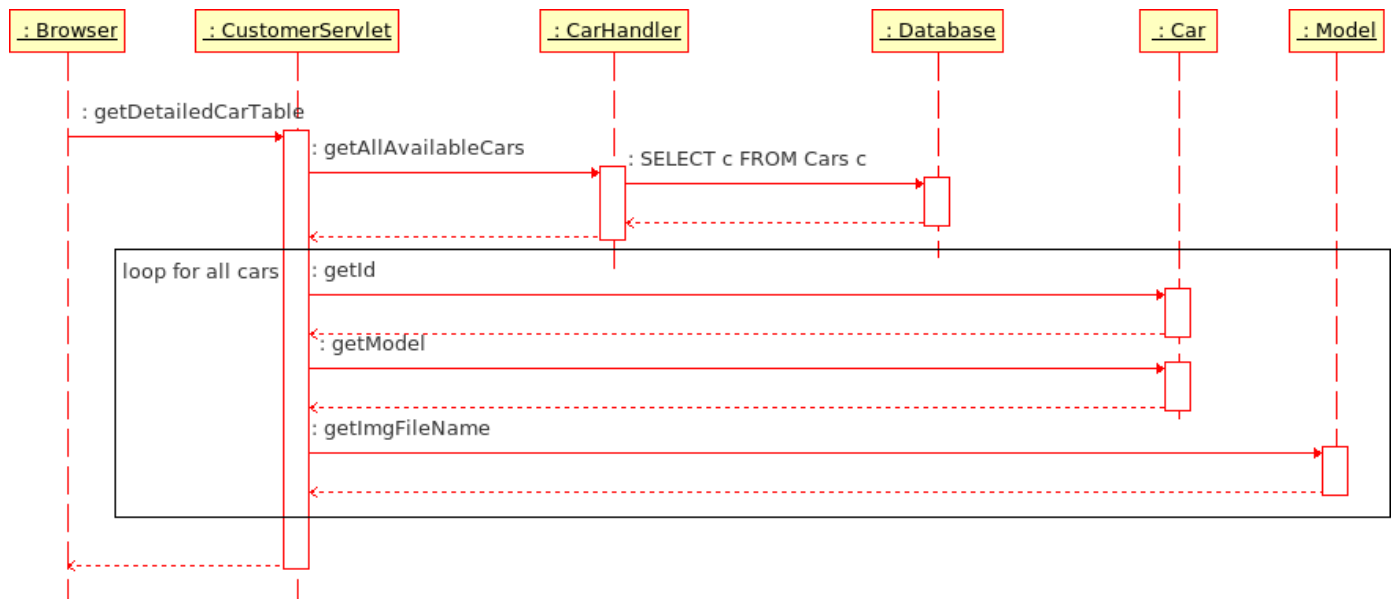
Persistence Layer (JPA)

Model (Database)

Database design



Sequence diagram for the use case “User can view all available cars”



Which technologies are used, where? A list.

Web server (Glassfish):

- JavaEE6
- Java Persistence API
- JPQL
- Jav Server Faces
- JavaMail API

Database server (Java Derby):

- Java Persistence API
- JPQL

Client:

- Javascript
- jQuery
- XHTML
- AJAX
- Google Maps API

Point out the strong parts of your application!

- (.) (.) <--- boobiez
-).(
- (Y)