**Requirements which have been propose**

The following functional requirements of the Car Sharing system have been stated, fully and should be able to be implemented and tested.

1 User registration with the Car Sharing system for TWO user types:

**Lifters** and **Leftie** users

2 Login

3 car registration

4 Registration of a vehicle

5 Trip registrations for leftie

6 Delete cars

7 Delete trips

8 Remove users

9 Ban users

10 Trip search

11 Edit trip data

12 Edit user data

13 Edit vehicle data (leftie and lifter)

14 Administrator functionality (for example user searching)

15 Calendar functionality which facilitates user date input

**Functionality Specification functionality (Note: functionality was checked during the last week of meeting:**

**Login in the system**

A user provides a username and password.

The system checks if all data has been provided by the user.

User input is checked for special characters. Such as “””, “\”, “/”.

Password and username are verified if they have sufficient amount of

characters (5 and more characters).

The username is checked for uniqueness.

If the username and password matches the data in the database, the user may

login in the system.

For security and statistical purposes the system keeps the date of the last

login and the IP address of the computer from which the user logged in last

time.

**Updating personal details**

User logins in the system.

User changes his/her personal details.

The system checks the provided data.

Forename and surname fields can include only letters. The gender fields can

be only “m” or “f”. Status filed can be: lifter or liftie.

The user cannot change the values of the following fields: the date of

registration, username, the date of the last login.

If the user provides valid data his/her data are updated.

**Removing personal details from the system**

User logins into the system.

User accomplishes steps to remove account.

The system makes the user’s data inactive.

The system checks if the user has a car and created trips.

The system makes the user car and trips inactive.

**Registration of a car**

User logins into the system.

The user provides the details of his/her car.

The system checks if the user enters the car registration number. The fields

car registration number, car type and car model can have numbers, letters

and spaces.

The system supposes that one user can have only one car and one car

belongs to one user.

The system checks if a car with the same registration number is registered

with website as the car of another user and is active.

Then, the users car is registered with the website.

**Updating details of user’s car**

User logins into the system.

The user changes the details of the car.

The user can change the registration number of the car. The system

supposes that the user changes the car, but still has the same trips.

The system checks if the user enters the car registration number. The fields

car registration number, car type, car model can have numbers, letters and

spaces, the field seats should have a numeric value.

The system checks if a car with the same registration number is registered

with the website as the car of another user and is active.

Then, the system updates the car details.

**Trip registration**

The user logins into the system.

The user enters the trip information.

The system checks to see if the user provides all the necessary information.

A trip point has to have a start and a end points. An intermediate point is

optional. Trip points can have city, postcode and street or city and street or

postcode values.

A trip has to have trip type. As the user can offer a lift after he/she registers

the car, the system checks if a user has a car.

A trip has to have the start time. The return time is optional. A trip has to

have a frequency value (daily, weekly, occasional). A weekly trip has to

have a week day, an occasional trip has to have a date. The system checks if

the car has trips.

The system registers the trip.

In order to collect statistical information about the trips, the system sends

the feedback letter to the user in the 5 days after the trip is registered.

**Updating details of trips which the user has registered**

The user logins in the system.

The user searches for his/her trips, then the user updates the trip

information.

The system checks if the user provides all necessary information.

A trip point has to have a start and a end points. An intermediate point is

optional. Trip points can have a city, postcode and street or city and street or

postcode values.

A trip has to have a trip type. As the user can offer a lift after he/she

registers the car, the system checks if a user has a car.

A trip has to have the start time. The return time is optional. A trip has to

have frequency value (daily, weekly, occasional). A weekly trip has to have

a week day, an occasional trip has to have date. The system checks if the car

has trips.

The system registers the trip.

**Removing trips**

User logins into the system.

User searches for his/her trips and deletes the trip.

The system makes the trip inactive.

**Searching for a trip**

User logins into the system.

The user accomplishes steps to search his/her trips.

The uses trips are displayed.