SRS - Software Requirements Specification

September 17, 2018

Contents

Document history

Version	Date	Resp	Description
0.x	2018-09-13	SG	Draft version

1 Introduction

2 Reference documents

There are no referenced documents for this version.

3 Background and goals

3.1 Main goals

The main goal of the system is to make it easier for commuters to car pool. This would hopefully reduce traffic in heavily trafficked areas and make the the commuting less stressful for everyone.

3.2 Actors

There are two main actors using the system

User: The users can be divided into two different roles, the driver and the commuter. Each physical person can have either role, although one person never can have both roles at the same time.

Administrator: The administrator is a special case user of the system. The administrator can add and remove users, rides and locations from the system.

4 Terminology

username: The unique name of a user inside the system.
password: A word or phrase used for authentication by the user.
ride: Information about who is going to drive as well as from where to where, departure time, arrival time, and the number of seats available.

5 Context diagram

The context diagram displayed in Figure 1 shows the interaction of the system's components and users on a high level.

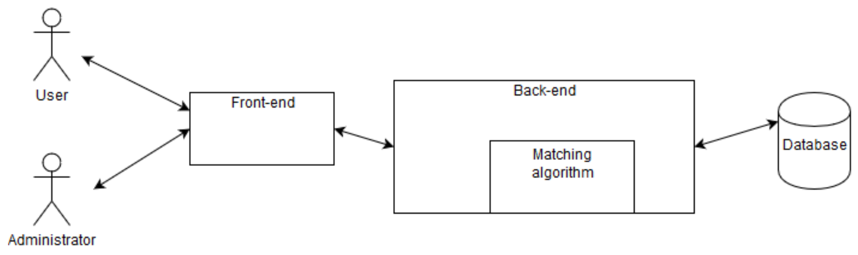

Figure 1: Context diagram

6 Functional requirements

6.1 Creating account

Scenario 6.1.1 The user creates an account. Precondition: User is not logged in.

1. The program starts
2. The user presses the "new account"-button.
3. The user is asked to provide username, email and password.
4. The user provides username, email and password and presses new account.
5. An account with the given information is saved to the database. The information is stored encrypted.

Scenario 6.1.2 If the users enters tries to create an account with an illegal username, email and password(see 6.6), and/or does not enter either a username, email or password, an account is not created and a message is displayed stating "Invalid information given" aswell as a "ok"-button. The user presses the ok-button and is taken back to the "new account"-screen.

Scenario 6.1.3 If the users tries to create an account with an already occupied email or username, an account is not created and a message is displayed stating "Email already in use" or "Username already in use" aswell as a "ok"-button. The user presses the ok-button and is taken back to the "new account"-screen.

6.2 Log in and log out

Scenario 6.2.1 The user logs in. Precondition: User is not logged in.

1. The program starts
2. The user is asked to provide username and password
3. The user provides username and password

4. The user is showed the website functionality.

Scenario 6.2.2 The user fails to log in. Precondition: User is not logged in.

1. The program starts.
2. The user is asked to provide username and password
3. The user provides username and/or password incorrectly.
4. "Incorrect username or password" is displayed to the user.
5. The user repeats step 3, 4 more times.
6. "Account temporarily blocked" is displayed before the user

Requirement 6.2.3 If the user fail log in 5 times, the user won't be displayed the log in for 5 minutes regardless of which machine the user is using. On login the user will instead be displayed "Account temporarily blocked".

Scenario 6.2.4 The user logs out. Precondition: The user is logged in.

1. The user is presented with a "log out"-link.
2. The user presses the log out link.
3. The user is taken back to the "log in"-screen.

6.3 Creating rides

Scenario 6.3.1 The user creates a ride. Precondition: The user is logged in.

1. The user clicks the "create ride" tab.
2. The user is asked to provide departure-time, arrival-time, number of seats, departure-location and arrival-location.
3. The user provides departure-time, arrival-time, number of free seats, departure-location and arrival-location.
4. The ride is saved and contains departure-time, arrival-time, number of free seats, departure-location, arrival-location and the username of the ride creator.

Scenario 6.3.2 If the user provides illegal departure-time, arrival-time, number of free seats, departure-location and/or arrival-location(see 6.6). The ride is not created, and a message is displayed stating "The information entered was not correct".

Scenario 6.3.3 The user can cancel the creation of a ride by pressing the "cancel"-button when asked to provide provide departure-time, arrival-time, number of free seats, departure-location and arrival-location. If the button is pressed, the user is taken back to the home-page, and no ride is created.

Scenario 6.3.4 If the user tries to enter a departure and/or arrival-time which has already passed , then a ride is not created , a message is displayed stating "Departure and/or arrival time has already passed" and an "ok"-button is displayed.

Scenario 6.3.5 If the user tries to enter a arrival-time that is before the departure-time, a ride is not created, a message is displayed stating "Arrival-time needs to occure before the departure-time" and an "ok"-button is displayed. The user clicks the "ok"-button and taken back to "create a new ride"-screen.

Scenario 6.3.6 The user cannot create a ride at a time where the user has indicated through previous rides that the user should be traveling as a passenger or a driver.

Scenario 6.3.7 If the user tries to enter a departure-time that is considered unreasonable based on previous rides(ex: the user has a arrival-time in Helsingborg at 13.00 and tries to create a ride with a departure time at 13.05 from Stockholm) the ride is not created. A message is displayed stating "Departure-time considered unreasonable based on ride history" aswell as a "ok"-button. The user presses the "ok"-button and is brought back to the create a ride screen.

6.4 Searching and joining rides

Scenario 6.4.1 The user searches for a ride and joins it. Precondition: The user is logged in.

1. The user clicks the "search ride" tab.
2. The user is asked to provide departure-time, departure-location and arrival-location.
3. The user provides provides departure-time, departure-location and arrival-location.
4. The user presses the "search"-button.
5. The user is displayed a list of available rides(according to). Each ride displays number of seats, number of free seats, username of all users who've signed up, roles of all users who've signed up, arrival-time, departure-time, arrival-location and departure-location.
6. The user is presses the join ride button attached to the ride.
7. The ride is added to the users pending rides.

Requirement 6.4.2 The user cannot join two rides that takes place during the same time.

Requirement 6.4.3 When joining a a ride, the user can see all of the usernames of the people added to the ride

Requirement 6.4.4 The driver can kick and ban users based on usernames from joining the ride. Preventing them to join the ride.

Requirement 6.4.5 When searching for rides the rides are displayed according to the following rules:

1. The system will only display rides which have a later ride departure-time than the given departure-time.
2. The system will only show rides which contain the a maximum 3 km difference between ride departure-location and given departure-location.
3. The system will only show rides which contain the a maximum 4 km difference between ride arrival-location and given arrival-location.
4. The higher the ride rank, the higher up on the page it is displayed.
5. First rides are ranked based on the difference between given departure time and ride departure time.
6. Secondly rides are ranked based on the distance between given arrival-location and ride arrival-time and distance between given departure-location and ride departure-time.

6.5 Displaying joined rides and deleting a ride

Scenario 6.5.1 The driver deletes a ride. Precondition: The user is logged in and hosting a ride.

1. The user clicks on the "My rides"-tab.
2. The user is displayed a list of each ride the user has joined. Each ride displayes the role of all users, username of all users who've joined the ride, total number of seats, number of free seats, arrival-time, departure-time, arrival-location and departure-location.
3. The user presses the "delete"-button attached to one of the users ride.
4. The ride is removed from the "My rides"-tab for all users who signed up to that ride.

6.6 Data

Requirement 6.6.1 User names and password should consist of 3-10 characters, only numbers and traditional English letters are allowed.

Requirement 6.6.2 Email should consist of 3-40 characters, only numbers,ascii-characters and traditional English letters are allowed. **Requirement 6.6.3.** When searching for or creating a ride arrival time and departure should follow the format AB/CD-XY:ZN, where A,B,C,D,X,Y,Z and N is each an integer. AB represents the day of the month, CD represents the month, XY the hour and ZN the minutes.

Requirement 6.6.4 When searching for or creating a ride, arrival and departure-location should only be made up of letters, numbers and ascii-characters.

Requirement 6.6.5 A user cannot be a driver and passanger at the same time on any specific ride.

6.7 Administration

Requirement 6.7.1. Admin users should be able to add users to the system.

Requirement 6.7.2. Admin users should be able to ban users from using the application.

Requirement 6.7.3 Admin users should be able to remove users from the system.

Requirement 6.7.4. Admin users should have full access to all tabs on the system.

Requirement 6.7.5. Admin users should be able to delete existing commute rides.

7 Quality requirements

Requirement 7.1. At least 80% of the code should be covered by unit-tests

Requirement 7.2. Knowledge corresponding to the knowledge goals of EDAA45 / EDA011 / EDA016 / EDA017 / EDAA20, and a basic knowledge of SQL and JavaScript should be sufficient in order to understand, maintain, and further develop the system

Requirement 7.3. 4 out of 5 students studying computer science at LTH should understand how to use the system within 10 seconds when sitting at a computer with the web-app opened.

Requirement 7.4. The application should work with
Firefox, Chrome, Opera and Edge on computers running Windows 10,
Firefox, Chrome, Opera and Safari on computers running MacOs 10.12 ”Sierra” and
Firefox and Chrome on computers running Ubuntu 18.04.

Requirement 7.5. When using the system is used in one of the computer rooms in ”E-Huset”, LTH, the response to any request should in at least 90% of all cases be given within 2.0 seconds.

Requirement 7.6 The system should be large enough for at least 10000 accounts to be created.

8 Project requirements

Requirement 8.1. The system's back-end should be developed in Java

Requirement 8.2. The system's front-end should be developed in JavaScript

Requirement 8.3. A SQL database should be used to store data.