

Software Requirements Specification

7th February 2019

Journey Sharing

Neeraj Sanjay Athalye

Hallikera Sree Harsha

Amogh Pandit

Xingyu Qiu

Abbas Ali Nadakkavil

Aditi Dubey

Submitted in partial fulfillment of the requirements of Advanced
Software Engineering (CS7CS3)

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Scope.....	3
1.3 References	3
2. Overall Description.....	3
2.1 Application Functions.....	3
2.2 User Characteristics	3
2.3 Constraints	3
2.4 Assumptions.....	3
3. Functional Requirements.....	4
4. Non-Functional Requirements.....	8

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Journey Sharing application. The goal of this project is to develop an app for travelers to find compatible fellow-travelers to share a journey. After each trip, each traveler will rate fellow travelers. A journey route will be devised taking all final destinations into account.

1.2 Scope

This application is intended for two types of people: Those who want to share their daily commute and those who want to share their journey with other passengers travelling on the same route. It considers everyone's preferences when matching them together. This application can also connect people without needing an internet connection.

1.3 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

This section will describe the overall behavior of the application and its workings along with the functions, user characteristics, constraints and assumptions

2.1 Application Functions

In the application, users will be able to look for passengers to share their journey and will also connect them via a chat window and calculate a starting location for everyone to meet.

2.2 User Characteristics

The app can be used by people of all ages. It facilitates two types of users: Those who are looking to share their daily commute, and those who want to share their journey with other passengers travelling on the same route.

2.3 Constraints

The app is constrained by the device's memory and power limitations. The app will work for a limited radius when it is not connected to the internet. It will also be constrained by the capacity of the database and the accuracy of the GPS location of the device.

2.4 Assumptions

The assumptions that have been made are that the device will have enough memory to store the app along with its data, it will have enough computing power to smoothly run the app. We have also assumed that the device will have a stable internet connection and have WIFI Direct enabled. We have also assumed that the devices are within a 500m radius of each other when using a peer to peer connection. (Offline mode)

3. Functional Requirements

The following tables showcase the functional requirements of the app in the form of use cases.

Use Case 1	View/Edit Profile
Description	The user can view or edit their profile. This includes being able to view/change the profile picture, name, gender, email address, phone number, home address and work address. You can also reset your password
Actor	Passenger
Precondition	The account must already be created
Trigger	Clicking on the View/Edit Profile option in the navigation drawer
Outcome	The profile details may be modified

Use Case 2	Sign In
Description	The user enters their username/email and password to gain access to the rest of the application. This serves as the gateway to the app. If the user has forgotten their password, they can choose the forgot password option
Actor	Passenger
Precondition	The account must already be created
Trigger	Opening the app or signing out from an existing account
Outcome	The user is granted access to the app

Use Case 3	Create Account
Description	The user is asked to enter personal details like name, gender, phone number, email address, password, home address and work address. An account is created with the above details and the user is sent to the sign in screen.
Actor	Passenger
Precondition	The user must have a valid email address and phone number
Trigger	Opening the app for the first time
Outcome	An account has been created for the user

Use Case 4	Manage/View Booking History
Description	The user can view the bookings that they have made in the past and can modify or cancel current bookings. Changes can be made to the preferences mentioned during booking
Actor	Passenger
Precondition	There must be existing bookings
Trigger	Clicking on the Manage bookings option in the navigation drawer
Outcome	The booking details may be modified

Use Case 5	Set Preferences
Description	The user can set their preferences for fellow passengers. These include setting the preferred gender, start time, number of passengers, mode of transport, distance to the starting point
Actor	Passenger
Precondition	A journey is being searched for/ created.
Trigger	Clicking on the Set Preferences option
Outcome	Preferences for the journey are set

Use Case 6	Search for Journey
Description	The user can search for a journey by entering their destination and giving their preferences for fellow passengers.
Actor	Passenger
Precondition	The account must already be created
Trigger	Clicking on New Journey option
Outcome	The app searches for journeys

Use Case 7	Send Journey Request
Description	If the user finds a journey in the previous step, a request can be sent to be added to the journey

Actor	Passenger
Precondition	The user must have searched for a journey
Trigger	A journey has been found by the user
Outcome	A request is sent to the creator of the journey

Use Case 8	Create New Journey
Description	If no journey is found after performing a Search, then a new journey is created with the user's preferences. Alternatively, a user can simply create a new journey without searching for one too.
Actor	Passenger
Precondition	The user must have searched for a journey
Trigger	The new journey option is chosen, or a journey is searched for, but no journey is found.
Outcome	A new journey is created

Use Case 9	Create/Join a journey group
Description	If a journey request was accepted, the user is added to the journey group and a group chat is opened. Else, if the user has created their own journey, a journey group is created, and passengers are added to it when their requests are accepted.
Actor	Passenger
Precondition	A journey must be created, or a journey request must have been accepted
Trigger	A journey request was accepted, or a new journey was created
Outcome	A journey group is created, and a chat is opened for the members

Use Case 10	Send/Receive Message in Chat
Description	The user can send/ receive messages in the group's chat window to communicate with the other passengers
Actor	Passenger
Precondition	The group must have been created

Trigger	The group has been created
Outcome	The messages will be sent/received

Use Case 11	Agree on a Starting Point
Description	After the group has been created, a central starting location is calculated based on everyone's current locations. Each member must then accept the starting point.
Actor	Passenger
Precondition	The journey group must be created.
Trigger	The journey has been created
Outcome	A central starting point is assigned for everyone to meet at.

Use Case 12	Start journey
Description	Once all the members have reached the starting point, the journey is started which follows the shortest path to the destination
Actor	Passenger
Precondition	Everyone must have reached the starting point
Trigger	Everyone must click on the Start Journey option
Outcome	The journey starts

Use Case 13	End Journey
Description	The journey is ended when the user has reached their destination
Actor	Passenger
Precondition	The journey must have already started
Trigger	Clicking on the End Journey button
Outcome	The journey is ended

Use Case 14	Rate Fellow Passengers
-------------	------------------------

Description	After the journey has been completed, the user must rate all the other passengers.
Actor	Passenger
Precondition	The trip must have ended
Trigger	This option is provided to the user after the trip has ended
Outcome	The passengers are assigned a rating

4. Non-Functional Requirements

The database and the Java web service should be hosted on an AWS server using high speed internet connection. The application should have an interactive and easy to use GUI. The waiting time for finding a journey should not be too long.