

Deliverable #1: Software Requirements Specifications (SRS)

TaxiShare

SE 3A04: Software Design II - Large System Design

T02 Group 01

Jordan Bowman (1133519)

Sadisha Galappatti (1220208)

Peggy Le (0961176)

Chenwei Wang (1065601)

Jack Witek (1211058)

Yunpeng Zhao (1212654)

Table of Contents

0 Revision History	3
1 Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, and Abbreviations	4
1.4 References	5
1.5 Overview	5
2 Overall Description	5
2.1 Product Perspective	5
2.2 Product Functions	6
2.3 User Characteristics	6
2.4 Constraints	6
2.5 Assumptions and Dependencies	6
2.6 Apportioning of Requirements	7
3 Functional Requirements	7
4 Non-Functional Requirements	12
4.1 Look and Feel Requirements	12
4.2 Usability and Humanity Requirements	12
4.3 Performance Requirements	13
4.4 Operational and Environmental Requirements	14
4.5 Maintainability and Support Requirements	14
4.6 Security Requirements	14
4.7 Cultural and Political Requirements	15
4.8 Legal Requirements	15
5 Division of Labour	16

0 Revision History

Revision	Revision Date	Description of Change
v1.0	October 6, 2014	Initial Document

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of a taxi carpooling application. The document will brief on the purpose of the application, features, how the application will interface with users and server, constraints on the software being developed, and external factors that will affect the end product. The SRS is intended for the use of stakeholders and developers as a foundation for the project.

1.2 Scope

TaxiShare give users the opportunity to share a taxi with strangers or friends in order to decrease the cost of their taxi fare. Users can offer to share a taxi to anyone who is travelling to the same destination or anywhere on the route, or they can request to share a taxi where they state a final destination and current location via GPS and are matched up with an offerer to share a taxi. Other features include a calendar booking system if taxis need to be reserved for a certain time and a point system to give users discount incentives to carpool and use the application. The system will be designed to allow people to communicate with other users participating in their carpool with the built-in chat system.

TaxiShare will be very applicable since it connects all users together with one common focus: to get from point A to point B. TaxiShare will be able to filter matches to cater to the needs of specific users and pick the best match for each individual. Given the design and production time frame, TaxiShare will be fast and easy-to-use so that users will continually use the application to reap the benefits of carpooling. The goal of TaxiShare is to provide an interface to the taxi dispatcher that the general public will want to use. Benefits will include saving money on taxi fares, request to share a taxi with a quick form, reduce time spent looking for a person to share a taxi with, and provide another environmentally friendly method of travelling.

1.3 Definitions, Acronyms, and Abbreviations

Below is a list of definitions, acronyms and abbreviations to be used in this document.

Definitions

Stakeholder: A person, group or organization that has interest or concern in the outcome of the application design process.

QR code: Two-dimensional matrix barcode which can be read by a camera on a phone. It is abbreviated from quick response code.

TaxiSharer: Users that are sharing a taxi with another user.

TaxiShare: A taxi ride appointment in which more than one user is carpooling. Also the name of the application.

Job: A request or offer from users handled by the server.

Acronyms

API: Application programming interface

SRS: System requirements specification

Abbreviations

App: Short for application

1.4 References

- (1) Google Maps API v2, 29 September 2014, Google:
<https://developers.google.com/maps/documentation/android/intro>
- (2) ZXing 3.1.0, 22 May 2014, ZXing: <https://github.com/zxing/zxing>
- (3) Facebook Login API v2.1, 2014, Facebook:
<https://developers.facebook.com/docs/android/login-with-facebook/v2.1>
- (4) Google+ login API, 17 June 2014, Google:
<https://developers.google.com/+/mobile/android/sign-in>
- (5) Twitter login API, 2014, Twitter Inc.: <https://dev.twitter.com/web/sign-in>
- (6) Google Play Developer Program Policies, 1 April 2014, Google:
<https://play.google.com/about/developer-content-policy.html>
- (7) Google Play Developer Distribution Agreement, 25 September 2014, Google:
<https://play.google.com/about/developer-distribution-agreement.html>

1.5 Overview

The SRS is designed to make it easy for the stakeholders to understand the details of the app. It also provides a general reference of the requirements for the developers to refer to in the design and development of the product. In the following sections, the description of the app will be elaborated on, providing functional and non-functional requirements. Section 2 contains the overall description of the product including the main functions, general assumptions made by the development team, user characteristics, and constraints. Section 3 covers all of the functional requirements of the app, and section 4 describes the relevant non-functional requirements of the design.

2 Overall Description

2.1 Product Perspective

TaxiShare will be a software component of a larger taxi cab system composed of users, taxis, the Taxi Company's dispatcher, Google Maps, QR code scanning and social networking apps.

TaxiShare will interface with Google Maps to allow customers to track the taxi route and present potential routes when requesting or offering taxi carpools. Google Maps is a very popular and easy-to-use mapping service. The Google Maps API is available for use, free of charge, and will be used to integrate the service with Taxi Share with the hopes of providing users a stable and reliable tracking/planning tool.

TaxiShare will also use the ZXing library for QR code reading and decoding. This library allows simple decoding of 2D barcodes. 2D barcodes can encode many types of actionable text including: URLs, contact information and geographic information. For the purposes of the taxicab system, each QR code will represent a unique identifier for each taxi.

Lastly, Taxi Share will allow users to log in with an existing social networking account such as Facebook, Google+ and Twitter. Users will then be able to share taxis with their friends more easily.

2.2 Product Functions

User Registration

Each user will create a profile that can later be edited or removed. Only registered customers may offer or request a taxi carpool.

User Login

Users will use the login credentials they chose when registering to enter the app, or may log in using their existing social network accounts (Facebook, Google+ and Twitter).

Requesting a taxi

Users will be allowed to enter the search options such as origin and destination. User will be shown a listing of available offers to which they can send a request, based on user rating and destination.

Offering a taxi

Users will be able to send out an offer either by scanning a QR code presented within the taxi or booking a future taxi ride with an integrated calendar. The offerer will give an origin, destination, and number of empty spots in the taxi.

Display the fare owed

The fare owed, distance and time travelled will be displayed in a convenient way for users after the taxi ride is done.

Chat system

Users shall be able to chat with other TaxiSharers in their future TaxiShares.

Points system

Points will be given based on distance travelled and number of TaxiSharers while sharing a taxi. Users can redeem these points for a discounted fare.

Integrated map system

Users will be able to use the map to view potential routes and track their taxi during a ride.

Profile Viewing

Users will be able to view, edit or remove their profile. They will be able to modify their email address, password or list of favourite locations.

2.3 User Characteristics

The product will be made for anyone who is capable of paying for a taxi and has an Android device. This includes people of any education background. No technical expertise should be necessary other than knowing how to use the Android operating system.

2.4 Constraints

The project must be completed by December 3rd, 2014. The product must encrypt all transmitted messages using a cryptosystem. The product must also new free and open-source resources.

2.5 Assumptions and Dependencies

The project will be designed for the Android 2.3 and above. It is assumed that the external interfaces will remain available for this project (Google Maps [API](#), ZXing library, and social networking login [API](#)). The

app will also depend on network connectivity. As a result, it is assumed that all users have subscribed to a cellular data provider.

2.6 Apportioning of Requirements

Not available at this time.

3 Functional Requirements

The various viewpoints in this section include:

User: The user is anyone who wishes to use the TaxiShare app. This includes non-registered users, registered users, TaxiShare offerers and TaxiShare requesters. The TaxiShare offerer is the one who is offering to share their taxi ride. The TaxiShare requester is the user who is looking to join a TaxiShare with an offerer.

Dispatcher: The dispatcher is an automated program that sorts the various requests and offers and finds matches between the two. It also stores the relevant information about each taxi in the fleet.

Security/Database: This component is in charge of encrypting and storing all of the user's information such as name, email, username, password, taxi rides, favourite locations and friends list. It is also responsible for decrypting the information for app access.

BE1. App is launched.

VP1.1. User

- i. A login/register screen shall be displayed if the user chose not to save their credentials. The login/register screen shall contain a username field, password field, "Remember my credentials" checkbox, "Forgot Password?" button, "Log In" button, "Log In using [social network]" buttons and "Register" button.
- ii. The main menu shall be displayed, if the user chose to save their credentials prior to exiting the app. The main menu shall contain the buttons "Offer Taxi Carpool", "Request Taxi Carpool", "View my Profile", "View Future Taxi Rides", "Log Off".
- iii. If the user chose to save their credentials, but the saved credentials were incorrect, the login/register screen shall be displayed.
- iv. If the app is launched for the first time on the device, a small tutorial shall be displayed for the user after logging in. The tutorial shall point to the various buttons in the app and explain what each button does.

VP1.2. Dispatcher

None.

VP1.3. Security/Database

- i. If the user chose to save their credentials, the saved credentials shall be sent to the database and checked with the information in the database to ensure that the correct information was provided. The database shall notify the app whether the correct or incorrect information was provided.

BE2. User wants to log in at the login screen.

VP2.1. User

- i. The user shall fill out the "Username" and "Password" fields at the login screen. Once done, they shall press the "Log In" button. If the login information was correct, the user

shall be taken to the main menu. If the login information was incorrect, an error notification shall be displayed to the user by the app and they shall then be taken back to the login screen.

- ii. The user shall be able to choose whether or not they want their credentials saved with the “Remember my Credentials” checkbox.
- iii. The user shall be able to login using an existing social networking account (Facebook, Google+ or Twitter).
- iv. If the user enters the incorrect password 5 times in a row, the user shall be notified that their account has been locked on that device until successful password reset. A notification shall appear asking the user if they want to reset their password. If they chose to reset their password, an email shall be sent to them. Refer to BE11.

VP2.2. Dispatcher

None.

VP2.3. Security/Database

- i. The submitted login information shall be checked against the list of registered users. If a match is found, the user shall be allowed to proceed to the main menu. If a match is not found, the app shall be notified that the login information was incorrect.

BE3. User wants to register for a new account.

VP3.1. User

- i. The user shall press the “Register” button at the login/register screen to begin registration of a new account.
- ii. The user shall be brought to a form containing the following required fields: first name, last name, username, password, confirm password, email, and gender.
- iii. Upon successfully filling out all of the fields, the user shall press the “Register” button at the end of the form. They shall be notified that their account was successfully created. Then, they shall be taken back to the login screen.
- iv. The user shall not leave a field blank, enter a username or email already in use, or enter non-matching password and confirm password fields. If any of the above have occurred, they shall be notified of the error.

VP3.2. Dispatcher

None.

VP3.3. Security/Database

- i. The database shall confirm that all required fields are valid (username or email is not taken, and the password and confirm password fields match). If the fields are valid, the information shall be stored in the database. If the fields are not valid, the app shall be notified that the fields are invalid.

BE4. User wants to request taxi carpool.

VP4.1. User

- i. The requester shall select “Request Taxi Carpool” at the main menu to request a taxi carpool.
- ii. The requester shall be taken to a calendar where they can select the day of the desired taxi ride. They shall then be taken to a form to fill out the details of the request. They shall need to enter the origin and destination. Optional search criteria shall include sort by nearest cab, highest rating first, find specific offerer, gender, and on my route.
- iii. The user shall be able to enter the destination by typing into a text box or by selecting from their list of favourites.

- iv. Once the requester finishes filling out the search criteria, they shall press the “Search for Taxi Carpool” button. A list of results satisfying their search shall then be listed.
- v. The requester shall be able to view the details (origin, destination, price, start time, expected time of arrival, number of people involved in taxi share) of each search result. They shall also be able to view the offerer’s route on Google Maps and the modified route if they decide to send a request. If the requester is interested in the selected search result, they shall be able to send a request for it using a “Request to TaxiShare” button.
- vi. The requester shall be able to cancel the search or search using new criteria.
- vii. Once the request has been confirmed by the offerer, the requested taxi carpool shall be added to the requester’s list of future taxi rides.
- viii. The offerer shall receive a notification of the list of requesters for their TaxiShare offer. The offerer shall be presented with the new estimated fare, distance and time. They shall then confirm whether the requester(s) could join their taxi ride. To help the offerer decide whether it is optimal to accept or refuse the request, the percent change in fare and time shall be presented to them. They shall also be able to view the modified route on a map.
- ix. The offerer shall be able to view the updated information of their taxi ride after confirmation of the request.

VP4.2. Dispatcher

- i. The dispatcher shall receive the requester’s search criteria to check against the available offers. The dispatcher shall then send the relevant offers to the app.
- ii. After the offerer confirms a request to an offer, the dispatcher shall update the information of the taxi ride. The updated information shall then be sent to the app.
- iii. The dispatcher shall receive the request from the app. It shall then forward the request to the offerer.

VP4.3. Security/Database

- i. The location and time of the carpool for the user shall be saved in the database.

BE5. User wants to offer taxi carpool.

VP5.1. User

- i. The offerer shall select “Offer Taxi Carpool” on the main menu to offer a taxi carpool.
- ii. The offerer shall have two options: scan a QR code presented in the taxi if they have hailed a taxi or book a future ride using the calendar.
- iii. If the offerer chose to scan a QR code, the offerer shall scan the code with their phone’s camera. The code shall represent the taxi ID. The offerer shall then enter the destination and number of empty spots of the ride. Once done, the offerer shall press the “Offer Taxi Now” button.
- iv. If the offerer chose to book a future ride, a calendar shall be displayed. The offerer shall select a day on the calendar for their future ride. They shall then be taken to a form where they shall fill out the details of the taxi ride such as time, origin, destination, and number of empty spots in TaxiShare. Upon completing the form, the user shall press the “View Potential Routes” button. The user shall be able to view potential routes on Google Maps. The user shall then press a “Book Future Taxi Ride” button after selecting the desired route.
- v. The user shall be able to enter the destination by typing into a text box, or by selecting from their list of favourites.

VP5.2. Dispatcher

- i. The dispatcher shall receive the information of the taxi ride after a QR code is scanned or a form for a future ride is filled out. The information of the new taxi ride shall be added to the list of taxi rides.
- ii. If a QR code was scanned, the taxi corresponding to the taxi ID shall have its status changed to “On route”.

VP5.3. Security/Database

- i. The information of the new taxi ride shall be saved in the database under the user’s information.

BE6. Registered user wants to log off of app.

VP6.1. User

- i. The user shall select the “Log Off” button in order to log off of the app.

VP6.2. Dispatcher

None.

VP6.3. Security/Database

None.

BE7. User wants to view or edit their profile.

VP7.1. User

- i. The user shall select the “View my Profile” button at the main menu in order to view their profile. They shall be able to view their name, email, number of points, and list of favourite locations.
- ii. The user shall select the “Edit Profile” button at the profile viewing page in order to edit their profile. They shall be able to change their email, password, and list of favourite locations (add or remove locations). Once done, they shall press the “Save” button to save their changes.
- iii. At the editing profile page, the user shall have the option to remove their profile with the “Remove Profile” button.
- iv. After the user presses “Save” or “Remove Profile”, they shall be asked to re-enter their password to confirm changes to or removal of their profile. In the case of removing the profile, the user shall confirm that they understand that everything shall be removed and the information cannot be retrieved again. If the correct password was provided, the user shall be notified that their profile has been updated or removed. If the incorrect password was provided, the user shall be notified that their profile could not be updated or removed due to invalid credentials.
- v. The user shall select the “View Friends List” button at the profile viewing page in order to view their friends list. The user shall then be able to add or remove friends from this list. To add a friend, the user can search for a username or email, or import from Facebook, Google+ or Twitter if they logged in using these social networking accounts.

VP7.2. Dispatcher

None.

VP7.3. Security/Database

- i. Any changes to the user’s profile shall be updated and saved in the database.
- ii. If the user chose to remove their profile, their profile shall be removed from the database.
- iii. The provided password shall be checked against the database. The database shall notify the app whether the password was correct or incorrect.

BE8. Taxi ride appointment begins.

VP8.1. User

- i. All users shall scan the QR code at the beginning of a ride appointment presented in the taxi in order to check in to the taxi share.
- ii. During the ride, the users shall be able to track the taxi location using Google Maps. The time it takes for the taxi to arrive at each user's origin shall be displayed.
- iii. At the end of the taxi ride, the actual fare owed and the points earned during the ride for each user shall be displayed on the app. The user shall be able to redeem their accumulated points for a discounted taxi ride.
- iv. After the payment is received, a short (two questions and comments section: 140 characters) rating form for the TaxiSharers and driver shall be displayed. There shall also be an "Add to Friends" button so that the TaxiSharer could be added to the user's friends list.
- v. The user shall receive 1 point per kilometer multiplied by the number of TaxiSharers in the TaxiShare.

VP8.2. Dispatcher

- i. The actual fare owed for the taxi ride shall be received from the taxi by the dispatcher. The actual fare owed shall then be sent to the app.

VP8.3. Security/Database

- i. The appropriate number of points shall be added to the profile of the TaxiSharers.
- ii. If the user selects the "Add to friends" button, the person is added to their friends list in their profile.

BE9. User wants to chat with other users who are sharing a taxi with them.

VP9.1. User

- i. The user shall select the "Chat with [Username]" button at the future taxi ride viewing page to chat with the desired TaxiSharer.
- ii. The user shall select the "Chat with Everyone" button at the future taxi ride viewing page to chat with all of the TaxiSharer(s) in the ride.

VP9.2. Dispatcher

None.

VP9.3. Security/Database

None.

BE10. User wants to view or drop out of future taxi rides.

VP10.1. User

- i. The user shall select the "View Future Taxi Rides" button at the main menu to view future taxi rides. They shall be able to view the details of the ride (origin, destination, estimated fare, other TaxiSharers, and number of empty spots in TaxiShare).
- ii. The user shall select the "Drop Ride" button at the future taxi ride viewing page to drop out of the taxi ride.
- iii. A notification shall be sent to the other TaxiSharer(s) that the user has dropped out of the ride. The notification shall also display the modified route that will be taken as a result of the modification.

VP10.2. Dispatcher

- i. The dispatcher shall receive the information that a TaxiSharer has dropped out of the TaxiShare. The information shall be updated and shall be sent to the remaining TaxiSharers in the ride.

VP10.3. Security/Database

- ii. The taxi ride information shall be removed from the user's profile in the database.

BE11. User wants to recover forgotten password.

VP11.1. User

- i. The user shall select the "Forgot Password?" button at the login/register screen. They shall then enter the email that they registered with and click "Confirm". The user shall be notified to check their email for a randomly-generated code created by the app. A screen displaying "Enter Code" shall be shown to the user. They shall then enter the code, received by email, in the "Enter Code" field. The user shall then be able to generate a new password with the password and confirm password fields. Once done, the user shall be taken back to the login screen.

VP11.2. Dispatcher

None.

VP11.3. Security/Database

- i. The randomly-generated code shall be saved as part of the user's information which shall only be valid for 24 hours. After 24 hours, the code shall be deleted from the database.
- ii. The database shall update the user's password only when the provided code is correct.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

- LF1. The product shall have a simple and organized look.
- LF2. The buttons shall be large enough so that users can easily see and select them.
- LF3. The colour scheme shall incorporate the colours used in the product logo (yellow and black).
- LF4. Text shall be black and use an easy-to-read font (Arial).

4.1.2 Style Requirements

- LF5. The product shall be attractive to young adults who would be using taxis more often.
- LF6. The product shall make the user feel safe.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

- UH1. The product shall make the users want to use it.
- UH2. Users between 16 and 65 years old should easily understand how to use the app.

4.2.2 Personalization and Internationalization Requirements

None.

4.2.3 Learning Requirements

- UH3. 90% of the registered users shall be able to use the app after viewing the tutorial that is provided after the app is run for the first time.

4.2.4 Understandability and Politeness Requirements

None.

4.2.5 Accessibility Requirements

UH4. The user shall be allowed to change the font to small, medium or large (sizes 8, 12 and 16, respectively).

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PR1. The app shall open the login/register screen or main menu with a maximum response time of 2 seconds.

PR2. Dispatcher will be able to process at minimum 30 executing jobs per second.

PR3. The user app shall process an offer or request within 10 seconds.

PR4. Delivery of chat messages shall be within 3 seconds.

4.3.2 Safety-Critical Requirements

None.

4.3.3 Precision or Accuracy Requirements

PR5. The system shall be able to calculate the time to arrive at pick up location in the minutes.

PR6. All monetary amounts shall be accurate to two decimal places. The sum of the TaxiSharers' fare should equal the amount displayed in the taxi. The amounts shall be rounded to the nearest hundredth (if thousandth value is greater than or equal to 5 then round up to the nearest hundredth; otherwise round down).

4.3.4 Reliability and Availability Requirements

PR7. User's information shall be available 24 hours per day, 365 days a year.

PR8. The system shall be properly functioning at all times except during the maintenance period (Refer to MS1).

PR9. Offering and requesting TaxiShares shall not be available during the maintenance period. Users booking a future taxi ride shall not be able to select the time slot on the calendar.

PR10. The user app shall only be unavailable when the product is downloading and installing an update on the device.

4.3.5 Robustness or Fault-Tolerance Requirements

None.

4.3.6 Capacity Requirements

PR11. The product shall allow a maximum of 6 TaxiSharers to a TaxiShare.

4.3.7 Scalability or Extensibility Requirements

PR12. System will be able to process 3000 executing jobs per second within three years of its launch.

4.3.8 Longevity Requirements

PR13. The system shall be properly designed to be available on the Google Play market from 2014 to 2017, before new taxi carpool demand is required in the public.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

None.

4.4.2 Requirements for Interfacing with Adjacent Systems

OE1. The product must interface with Google Maps, QR code scanning, and social networking sites.

OE2. The product shall allow installation and operation on Android smartphones or tablets with a minimum system requirement of Android 2.3 or above.

4.4.3 Productization Requirements

OE3. The product shall be available free of charge on Google Play.

4.4.4 Release Requirements

OE4. Each release shall not cause previous features to fail.

OE5. Maintenance releases shall be offered to users two times a year.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. The maintenance period shall be within 30 minutes, 2 times a month, on a Wednesday from 4:00AM to 4:30AM.

4.5.2 Supportability Requirements

MS2. The product shall let user to view the tutorial again, if needed.

MS3. Users shall be able to send an email to TaxiSharehelp@gmail.com for all support queries.

4.5.3 Adaptability Requirements

None.

4.6 Security Requirements

4.6.1 Access Requirements

SR1. The product shall require the user to possess their own login account and password in order to access the entire app.

SR2. Only account owner can reset their own password by validating email when they forget their password.

4.6.2 Integrity Requirements

SR3. Any data stored on the database shall be encrypted in order to prevent modification or interception of data.

4.6.3 Privacy Requirements

SR4. The user can only view their own data and no one else's.

4.6.4 Audit Requirements

None.

4.6.5 Immunity Requirements

None.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

None.

4.7.2 Political Requirements

None.

4.8 Legal Requirements

4.8.1 Compliance Requirements

LR1. The product shall comply with the Google Play Developer Program Policies and the Google Play Developer Distribution Agreement.

4.8.2 Standards Requirements

None.

5 Division of Labour

Everyone came up with an innovative feature and presented it at the second group meeting. The last group meeting was a group revision meeting. Everyone gave insightful comments on all of the sections. Due to this, everyone contributed to all of the sections in the SRS document, but focused on their respective parts as well.

Jordan Bowman

- Innovative feature: points system
- Section 3: Functional Requirements

Sadisha Galappatti

- Innovative feature: logging in with social networking credentials
- Section 1: Introduction

Peggy Le

- Innovative feature: points system
- Section 3: Functional Requirements

Chenwei Wang

- Innovative feature: points system
- Section 4: Non-Functional Requirements

Jack Witek

- Innovative feature: integrated calendar system to book or search for future taxi rides
- Section 2: Overall Description

Yunpeng Zhao

- Innovative feature: chat system among TaxiSharers
- Section 4: Non-Functional Requirements