Software Requirements Specification
CabShare™

Monday November 3rd, 2014
Tutorial 1
SE 3A04

Team Members

Kemal Ahmed	1150377
Mitchell Spector	1208006
Sahajmeet Bhutta	1223708
Xue Lin	1151043
Yuchen Xiao	1212583

Table of contents

Revisions

Version	Date	Sections changed/added	Summary of changes
Version 0	Monday, October 6th 2014	Software Requirements Specification Document created	N/A
Version 1		Business events section References section corrected Nonfunctional requirements	Redundant and incorrect business events removed. A minor semantic error was corrected in the references section. Nonfunctional requirements which were outside the scope were removed.

1 Introduction

1.1 Purpose

The intent of this software requirements specification (SRS) is to clarify the functional requirements, nonfunctional requirements, features, and the purpose of the proposed product, an Android **application** named **Cabshare™**. The intended audience of this SRS are the stakeholders, namely the cab company which will be using the **application**. It will also serve as a guide for the development team to ensure the **application** is created as originally intended.

1.2 Scope

CabShare™, herein referred to as 'the **application**', will allow its users to carpool in cabs in order to lower an individual customer's cab fare by splitting the fare. This should help increase the number of customers for the cab company using the **application** by offering competitive rates.

The **application** will allow a **cabpooler** to request a cab by beginning a search. The search will ask for criteria, such as the **cabpooler's** current location, which can be relayed to the **application** by the mobile device's GPS, whether or not they would require an accessible cab, how far the **cabpooler** is willing to walk, how long the **cabpooler** is willing to wait, and destination. The requested search will be encrypted and sent to the **dispatcher** which will return results matching the search criteria. The **cabpooler** will be able to browse through search results, view profiles of **commuters** in cabs returned in the result, sort the results by ratings of **commuters**, amount of money saved, amount of time until the cab could arrive, and if the destination of the **commuter** matches the destination of the **cabpooler**. Once the **cabpooler** has chosen a cab, that choice is encrypted and sent to the **dispatcher** who would send a request to the **commuter**. The **commuter** may chat with the **cabpooler** and accept or deny the request. Once the request has been accepted, the **dispatcher** would relay the information to the cab company which would communicate with the cab; this is outside the scope of the **application**.

This **application** will not leak the name, contact information, or any other private information of any of its users. It will also not allow **commuters** or **cabpoolers** to contact the cab drivers directly.

1.3 Definitions, Acronyms, and Abbreviations

A.P.I.: Application programming interface;
application: The product proposed by this document

cabpool: A carpool between a **cabpooler** and a **commuter**.
cabpooler: A person at somewhere else waiting for an offered cab carpool
CabShare™: The name of our product; used interchangeably with 'the **application**'
commuter: A registered passenger currently in a **CabShare™** cab, sending the cab **cabpool** offer
dispatcher: A system designed to stores all relevant information about each cab in the fleet and decides how to handle offers and requests for **cabpools**.
foreign source: An individual or electronic device that is not the current user of the **application** or the company receiving information.
overseer: A person under the employ of the cab company. They oversee the smooth operation of the dispatcher and communicate with the cab company and system.
potential cabpooler: A registered user who does not yet have the **application** open, but has intentions of using the **application**
potential commuter: All non-registered users who have interacted with the system by entering the cab, or starting a trip. Once the register as a user they would be considered "potential **cabpoolers**" (see above).
Reasonable Time: The time estimated by a map engine such as google maps to travel a certain specified distance in a specific mode of transportation

1.4 References

In the initial stages of understanding our competition, the team looked into existing applications offering similar services. These included the applications Uber and Didi Dache. We also looked into using the Uber API, but ultimately decided against using it because it violates their terms of service as a competitor to their business. The original website for the Didi Dache (Didi Taxi) application, an application created in China, was written in Mandarin and was viewed after Google Chrome translated its pages. The Didi application was designed to connect drivers and passengers, where our **application** is designed to connect potential passengers to current passengers.

1. Wei Cheng. "Didi Dache Home Page". Available: http://www.xiaojukeji.com/website/index.html [2014-09-15].
2. "Uber API Terms of Use". https://developer.uber.com/v1/terms/, 2014-09-12 [2014-09-15].

1.5 Overview

Sections 1 through 3 are written in IEEE format and section 4 is written using the Volare format.

The SRS describes the **application**, the functional and non-functional requirements, and how the tasks were distributed by the team. The next section gives the general overview of the product itself, brief descriptions of its functions, constraints on the system, any assumptions held about the intended users while creating the product, and any features which can be implemented in later versions of the software. As the document continues, it will describe the functional requirements and general factors that affect the product in more detail. These requirements are perceived from different viewpoints and describe how the system will react to certain external stimuli. Further in the document, all other requirements are outlined which are not at the core of the **application's** main reason for existence. It outlines requirements about security, the look and feel of the **application**, usability, performance, the product's operation and environment, and maintainability and support. The final section of the document outlines the contributions of each member to this document and, in later versions, towards the completion of the product as a whole. The document as a whole should provide a thorough overview of what this product aims to accomplish.

2 Overall Description

2.1 Product Perspective

This **application** will allow cab customers to arrange **cabpools** to minimise the cost per customer. Its intended users are customers who are willing to share a cab with others. Some products that the development team used for reference include Uber and Didi Dache, which operate cab services with a mobile application interface that includes some of the features of our **application**. The difference is that our **application** is focused on the customer already in the cab, making it efficient and easy to carpool with people in your area while the previous products are driver-focused, helping drivers that would offer cab services connect to more potential customers. The **application** itself will not be self-contained as it will need access to Google maps for location data and PayPal will be the prefered and only payment method through the **application**. Also, the **application** will be associated with a particular cab company for their exclusive use to encourage sales of their service.

2.2 Product Functions

CabShare™ will contain two modes, "Request Carpool" mode for **cabpoolers**, and "Offer Carpool" mode for **commuters**. In "Request Carpool" mode, the **application** allows for the **cabpooler** to enter their current location and destination along with a number of optional search criteria. After the search, the **application** returns a list of potential matches according to the **cabpoolers'** search criteria. When the **cabpooler** chooses a match, the **application** forwards the request to the potential **commuter**, along with an optional short message from the **cabpooler**.

In "Offer **Cabpool**" mode, the **application** allows the **commuter** to scan a QR code (representing the Cab ID) that is provided by the cab. After scanning the QR code, the **application** allows the **commuter** to enter relevant information (destination, cab ID, maximum number of customers to share with) and makes it an offer then sends it to potential **cabpoolers**. The **commuter** would then be able to receive requests from **cabpoolers** for the duration of his or her trip to share the cab. In addition, the **application** will allow users to make, edit, and remove their own profiles. Once a **commuter** has reached his or her destination, the **commuter** will be asked to first rate the other **commuters**, which they may skip if they choose, before the **commuter** will be taken to the payment page.
The search criteria, such as accessible cab, maximum wait time, and maximum walking distance for pick up,the users in the friend list and number of **commuters** to share with is sorted by nearby cab. The highest rating is displayed first with the option to find a specific offeror,etc)

2.3 User Characteristics

The user base for this system will consist of two types of people: **commuters** and **cabpoolers**. Because the basic development environment is under Android, the users must be able to operate an Android device. The **commuter** and **cabpooler** must know how to use the GPS on their mobile device or Google Maps to locate their location and destination, so that the system can quickly and conveniently provide a matched request for the users. Users are assumed to be knowledgeable in the use of the Google maps **application** on Android devices as this **application** will make use of it and have a user interface of a similar flow and design.

2.4 Constraints

The **application** must be running on an Android device that is version 3.0 or later, and it should be able to handle a maximum capacity of 100,000 users. Running the **application** must not cause permanent damage to the device. This **application** must be fully functional in the Greater Toronto Area and it is to be completed before December 3rd, 2014. To be competitive, the **application** should not use more than 10.5 MB of space in the phone's memory.

2.5 Assumptions and Dependencies

This product is requires access to a working mobile device which has Android version 3.0 and above.

2.6 Apportioning of Requirements

Future versions of the **application** may include a number of additional features. It may be possibly to expand the **application** to include more than one cab company, which may or may not include a business merging or partnership. Different language options may be provided, and allow search criteria to include **commuters** and drivers who speak a preferred language. Functionality may be included to allow swiping left to decline a search result and a share request or right to add it to a list of considered search results and to accept a share request.

3 Functional Requirements

Mode 1 Request the Cab

BE1 A registered user wants to hire a cab.

VP1 Cabpooler

I. Application allows secure login
II. Application allows for specification of desired destination.
III. Application gives functionality for user to choose from cabpools in their area

VP2 Commuter

I. N/A

VP3 Health and Security

I. Application reminds user to be courteous to other commuters.
II. Application allows secure login

BE2 A non-registered user wants to hire a cab.

VP1 Cabpooler

I. Application allows registration.
II. Can call a cab (potentially a cab not within the application) directly

VP2 Commuter

I. N/A

VP3 Health and Security

I. Application allows secure login

Mode 2 Offer the Cab

BE1 A registered user wants to offer a cab.

VP1 Cabpooler

I. Have functionality to confirm if a cabpooler has confirmed share request.

VP2 Commuter

I. Allow offering of cab, attaching intended destination of the cab to create a route
II. Allow confirmation of applications to join ride

VP3 Health and Security

I. Keep identities of commuters and cabpoolers private; share comments and ratings, only

BE2 A non-registered user wants to offer a cab.

VP1 Cabpooler

I. N/A

VP2 Commuter

I. Application has functionality to allow for registration.

VP3 Health and Security

I. N/A

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirement

LF1. The **application** will have two modes each with its own icon on the main page for users to select: a “Request Cab Mode” and an “Offer Cab Mode”.

LF2. The interface will be neat and orderly with no overlapping of visual objects in the user interface.

LF3. A screen with an agreed upon colour will indicate to cabs which **cabpooler** is waiting for them. **Cabpoolers** will be instructed to obnoxiously wave the screen at oncoming traffic.

LF4. Search results identify number of current passengers by the following colour scheme: white for 0 passengers, purple for 1 passenger, blue for 2 passengers, green for 3 passengers, yellow for 4 passengers, orange for 5 passengers, and red for 6 passengers.

LF5. The buttons should be large enough to tap easily, but not so large as to appear tacky.

LF6. The search results will display different icons to indicate different types of cabs (cars, vans, handicap vans), and show a small number indicating the number of **commuters** currently in the cab.

4.1.2 Style Requirement

LF7. It should be easy to move from menu to menu, using the buttons.

LF8. The font of the application should be easy to read, sans serif.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

UH1. Cab users between the ages of 15 to 70 years should be able to use this **application.**

UH2. Colours of the background and any overlaying text should be highly contrasted for people with colour blindness.

4.2.2 Personalization and Internalization Requirements

UH4. The default language of this **application** will be Canadian English, but other language options can be chosen.

UH5. The default currency is the Canadian dollar.

4.2.3 Learning Requirements

UH6. The **application** should be easy to learn and pick up for users who have previously used Google maps.

UH7. After signup, a tutorial will run via popup conversation bubbles to show all users the basic features of the **application**. It will take approximately fifteen minutes.

4.2.4 Understandability and Politeness Requirements

UH8. The **application** should say"Please enter your password."

UH9. The **application** should use words and symbols that are identified as gender-neutral by users.

4.2.5 Accessibility Requirements

UH10. The search criteria allows users to search for accessible cabs.

UH11. The buttons will be large enough for elderly users to tap with ease.

UH12. Contrast between text and the background behind the text shall be contrasting to cause less strain on the eyes of general users, and those with colour blindness.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PR1. The **application** will list available cabs and relative times of arrival to the **commuter** within 30 (thirty) seconds.

4.3.2 Safety-Critical Requirements

PR2. We take no responsibility for any damages to your phone as a result of waving the phone to signal the cab driver.

4.3.3 Precision or Accuracy Requirements

PR3. The fare amount will be accurate to two decimals.

4.3.4 Reliability and Availability Requirements

PR4. The service should be usable twenty-four hours a day, seven days a week, but the availability of cabs is dependant on the cab company.

4.3.5 Robustness or Fault-Tolerance Requirements

PR5. If there are no available cabs within the requested area, the **application** will expand the search radius incrementally until suitable cabs are found and offer the option of calling the cab company directly.

4.3.6 Capacity Requirements

PR6. There is one cab company using the **application** to promote its services.

PR7. There are up to 256 cabs available for hire through the application.

PR8. Each van cab can hold up to 6 passengers, each car cab can hold up to 4 people, and each accessible van can hold up to 3 people.

PR9. The **application** will display the 10 closest cabs from the **cabpooler**.

4.3.7 Scalability or Extensibility Requirements

PR10. The **application** will be developed in a such way where more cab users can use this **application** to offer and request the cab in the future.

4.3.8 Longevity Requirements

PR11. The **application** will be available as long as the cab companies require it.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

OE1. A Samsung Galaxy S4 will be the targeted device for the physical environment.

OE2. The mobile device's operating system will be run under Android OS version 3.0 or a more up-to-date version.

4.4.2 Requirements for Interfacing with Adjacent Systems

OE3. The **application** should scale to different screen sizes.

OE4. The **application** should have different layouts for horizontal and vertical views.

4.4.3 Productization Requirements

OE5. Any information needed to operate the **application** shall be saved in a data store.

4.4.4 Release Requirements

OE6. The **application** will be available from early 2015 for companies and customers who would like to make use of this **application**.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. The **application** needs to be maintained until April 1st, 2015, the due date of the competition.

4.5.2 Supportability Requirements

MS2. The **application** must support android version 3.0 (Honeycomb) and above.

4.5.3 Adaptability Requirements

MS3. Future versions of the **application** could port to other mobile device OSs, such as blackberry, iPhone, Windows phone, or any other platform the company could desire.

4.6 Security Requirements

4.6.1 Access Requirements

SR1. Users should have read-only access to any other user's ratings through the **application**.

SR2. Users should only have write access to their own profile description.

SR3. Cab users should be able to rate people they share the car with only once per trip.

4.6.2 Integrity Requirements

SR4. All users of system will be managed by the administrator and all users introduced into the system will be verified by the system administrators.

4.6.3 Privacy Requirements

SR5. Ride history (with the exception of reviews of trips) should be private to the rider.

SR6. Chat between users should be allowed.

SR7. Users have authority to encrypt some of their own information, except some basic needed information such as, location and destination.

4.6.4 Audit Requirements

SR8. The data stored will be updated when users verify to change the information.

SR9. The data should be updated when the customers have completed their trips.

SR10. The login and logout will be dated and time stamped.

4.6.5 Immunity Requirements

SR11. Under no situation should a **foreign source** be able to access the account of a **CabShare™** user.

SR12. The data store of the **application** should not be able to be accessed by a **foreign source.**

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

CP1. All text in the program must be written in Canadian English.

CP2. If any inputs are required they need to be of the standard generic type, in which the user interface of another product like Google maps should have applicable knowledge to our **application**.

4.7.2 Political Requirements

CP3. The **application** must not insult or offend any majority or minority group present in the **CabShare™** sales demographic.

4.8 Legal Requirements

4.8.1 Compliance Requirements

N/A

4.8.2 Standards Requirements

N/A

Appendices

A Division of Labour

Sahajmeet edited Nonfunctional requirements.
Mitchell, Kemal, Sahaj corrected business events

Team Members

Kemal Ahmed	1150377	_____
Mitchell Spector	1208006	_____
Sahajmeet Bhutta	1223708	_____
Xue Lin	1151043	_____
Yuchen Xiao	1212583	_____