SOFTWARE REQUIREMENTS SPECIFICATION

Rodeo Town Taxi Mobile Application

Rodeo Speedwagon: Tyler Lloyd, Junyu Lu, Austin Richardson, Jose Rodriguez, Stephen Stengel

Table of Contents

Revision history

Version 1: December 4, 2019

1. Overview

1.1 Company and mission

The company in which this app is to be developed is the Rodeo Town Taxi LLC located in Ellensburg, WA. This company has been around for many years providing numerous services in addition to a traditional taxi service. Rodeo Town Taxi also provides delivery services, lockouts, and jump starts. Their main focus is to provide a safe, fast, and friendly service for any user and to help in the fight against drunk driving in their local area.

1.2 Purpose

The purpose of this SRS document is to outline and describe the software process, requirements, and tools used in the development of the Rodeo Town Taxi mobile application. This document is a blueprint and description of all details related to the application for our software team and the client to reference and to follow as a guideline during the development process and can always be referenced for necessary information

1.3 Scope

The Rodeo Town Taxi Mobile Application's objective is to provide a convenient way for customers to call for a taxi at their location. Instead of calling the Rodeo Town Taxi dispatch phone number, customers can open the app and click a button to automatically request a taxi to the user's location. For taxi drivers, the application will provide an easy way to locate where to pick up the customer.

1.4 Definitions, acronyms, and abbreviations

API: An acronym for Application Program Interface
App: An abbreviation for mobile application
Portrait mode: A term used when the device in question is in a vertical orientation

1.4 References

Rodeo Town Taxi's website:
https://rodeo-town-taxi-llc.business.site/

IEEE SRS Std-830-1998:
http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf

1.5 Overview

The rest of this SRS will provide and overall description of the Rodeo Town Taxi App as well as necessary requirements for the app. The remaining two sections of this SRS are organized to reflect this.

2. Overall description

2.1 Product perspective

The Rodeo Town Taxi App is a standalone app, meaning it is not a part of a larger application or system. However, this does not mean that the application will not rely on external components like a database to communicate with and store driver and user data, the server to host this database, and the APIs and libraries that will make the functionality of this application possible. In fact, it is these components, which independently are not functional, that come together to form a larger system; the taxi application.

2.1.1 User interfaces

The app will require the user's device to be in portrait mode. Making an action within the app will refresh the current screen layout to accommodate the new actions that the user can take. Additionally, the app will make use of a clean and minimalistic user interface. The screen that allows the user to login will only contain necessary elements: text fields and a submit button. The main screen for a customer would contain a map and a button to request a driver. The main screen for a driver would simply contain a map showing where they need to go to pick up the customer. Drivers will occasionally be prompted with a customer request, which is distributed to all active drivers. Drivers can accept customer requests on a first come first serve basis. Another key characteristic of the minimalistic layout is the use of short error messages because the target audience for this app will likely not want to be overwhelmed with long and complex error messages. To summarize, the user interface needs to be as intuitive as possible because the software is meant to decrease the difficulty of calling a taxi.

2.1.2 Hardware interfaces

The Rodeo Town Taxi App will be compatible with mobile phones running the latest versions of the iOS and Android operating systems. The app will make use of the device's GPS antenna, Wi-Fi antenna, and Cellular antenna if the user is not currently connected to a Wi-Fi network.

2.1.3 Software interfaces

This app will be built with Google Cloud, Firebase, Google Maps API, and React Native. Google Cloud will be used for creating the server used by the app. Additional information on Google Cloud can be found at https://cloud.google.com. Firebase will be used as the Google Cloud server's database management system, which will store data for each user account. More information on Firebase can be found at https://firebase.google.com/docs. Google Maps API will be used to locate the customer requesting the cab as well as the driver when he is on his way to the user. Additional information on Google Maps is available at https://developers.google.com/maps/documentation. Lastly, React Native 0.61 will be used to design the user interface for the Rodeo Town Taxi app. This will allow the developers of the Rodeo Town Taxi app to develop for iOS and Android simultaneously. Documentation for React Native can be located at https://facebook.github.io/react-native/docs/getting-started.

2.1.4 Communication interfaces

The app will make use of the device's Wi-Fi and Cellular antennas to establish a connection to the app's server via the internet. The antenna that will be used at any moment in time will depend on how the user is connected to the internet at that moment. If they are connected via Wi-Fi, the app will use the Wi-Fi antenna. If not, it will use the Cellular antenna. Additionally, The GPS in the device will be used to broadcast the user's location to the driver if they are a customer or the customer if they are a driver.

2.1.5 Operations

The server that the Rodeo Town Taxi connects to will perform regular backups to protect the data in the database. Restorations will be made in circumstances where it will be necessary.

2.2 Product functions

The application will ask a guest for their name. There is a desirable feature where the user will be able to create an account to store additional information such as number of rides and rewards and this may be included considering time constraints. The guest will be able to access the application to either call the dispatch or request a ride as would a user with an account. The Rodeo Town Taxi app will give the customer the ability to request a taxi ride at the user's current location. They will be presented with a map showing where the driver currently is. The driver will be able to see where the customer is on their map after accepting a customer ride request. The operator of Rodeo Town Taxi will have special access to the database of his drivers, giving them the ability to add and remove driver profiles from the database when needed.

2.3 User characteristics

The typical user of the Rodeo Town Taxi App is one who possesses the skills required to comfortably use a smartphone running iOS or Android. The application is to be built in a way that any user with the knowledge of using a mobile device and connecting to the Internet will be able to download the application and follow all the steps to request any service from Rodeo Town Taxi. The app does not require any other special skills.

The customer is the default user of the Rodeo Town Taxi. They will be able to request taxi rides and other Rodeo Town Taxi services. They will lack the enhanced privileges that the driver and administrator roles have.

The driver is an employee of Rodeo Town Taxi who will be responsible for picking up the customer. someone who has experience using apps similar in complexity to the Rodeo Town Taxi App. For example, the driver is already familiar with an app used as a payment system for the business and should be able to understand the driver interface of this app easily.

The administrator is someone who is tech savvy but is not able to reprogram the database system or the app themselves. They will be provided a manual which will act as a reference for the use and maintenance of the system. The administrator is assumed to be the owner of Rodeo Town Taxi, but could also be a trusted employee. The administrator has experience using similar apps and apps similar in complexity.

2.4 Operating environment and tools considered

The operating environment for the Rodeo Town Taxi application will include the following aspects:

- Cloud Server – Google Cloud
- Database – Firebase
- Client / Server interactions between the application and the cloud server
- Cloud Server Operating System – Linux
- User Platforms – Android and iOS (Mobile)
- Languages – React Native JavaScript and JSON

2.5 Constraints

2.5.1 Hardware limitations

The Rodeo Town Taxi app will be designed for devices running the latest versions of the Android and iOS operating systems. If the device in question is not able to run those operating systems, it may have difficulty running the app to its full potential. Additionally, the user's phone will be running other applications at the same time as this app and its resources will be limited in terms of memory and processing power and storage.

2.5.2 Interfaces to other applications

The Rodeo Town Taxi App will provide a shortcut that allows the user to dial the Rodeo Town Taxi dispatch phone number. This will require interfacing with the default iOS and Android phone dialing apps.

2.5.3 Reliability requirements

Should the app lose connection to the database, every feature within the app except the dial function will become inaccessible by the user until a connection to the database is reestablished. The user will be notified of this should they try to use the app without a connection to the database. The app will try to reconnect as long as it is open.

2.5.4 Safety and security considerations

Because the Rodeo Town Taxi App will allow drivers to create accounts with their email address and a password, the database for the app will need to store this data securely. This will involve encrypting user passwords, which stores the passwords as an undecipherable string rather than plain text. This is necessary to keep user login information secure should someone gain unauthorized access to the database. Additionally, the server containing the database will require software such as Fail2Ban to prevent unauthorized access through brute force attempts. Lastly, any actions made in the database, such as login attempts and changes to values, will be recorded in a log file which will include time stamps and the IP address the user made changes from.

2.6 Assumptions and dependencies

An important factor that may affect our ability to complete the requirements would be if the customer were not able to provide us with the hardware or software to host a server/database. This would mean that there would be no way to store user information and therefore no way to distinguish between customers, drivers, and administrators; leaving many of the main features impossible to accomplish with a single application. Another important factor that would reduce the functionality of the application would be if the customer was not able to provide us with the services offered by the Google Maps API or other systems like it. Since the customer requested the ability for the driver to locate the customer on the map and the customer to track the driver, a service like Googles Maps API is mandatory. Finally, our application depends completely on the user (driver, customer, or administrator) having an internet connection and GPS functionality on the device so removing either of these would prevent the application from communicating with the server and locating the user on the map

3. Specific requirements

3.1 External interface requirements

3.1.1 User interface for the customer

Upon starting the Rodeo Town Taxi App, the customer will be presented with a map showing their current location, assuming they have established a connection to the server. After that, they will be able to request a driver to pick them up at their current location by pressing a button at the bottom of their screen labeled “Request Ride.” Once the customer requests a ride, the map will show the location of the driver that will pick up the customer. The screen will also be populated with the driver’s name and the make and color of the car they are driving. Once the driver has picked up the customer, the customer interface will return to its default state. The default customer interface will also allow the user to press an on-screen button to dial the Rodeo Town Taxi dispatch phone number.

3.1.2 User Interface for the driver

When starting the Rodeo Town Taxi app, the driver will be able to login through a dedicated menu within the app to verify that they are indeed a driver. When a driver logs into the Rodeo Town Taxi App, they will be presented with a map showing their current location, assuming they have established a connection to the server. When they are assigned a customer to pick up, the map will show the location of the customer that the driver has to pick up. Once the driver has picked up the customer, the driver interface will return to its default state.

3.1.3 Communication interface

In addition to establishing a connection to the server, the device running the Rodeo Town Taxi App will require an internet connection to receive app updates through the operating system’s app store.

3.2 Functional requirements

3.2.1 Driver role

3.2.1.1 Receiving a customer

When logged in as a driver, the user will wait for a customer request. These requests are shown to all drivers who are not already serving a customer. They will be able to accept the requests on a first come first serve basis. Should the driver in question be the first person to accept the request, they will then see the customer's location on the map and drive to it to pick them up. The rest of the drivers will no longer see the customer request on their screen and will have to wait until a new request is sent.

3.2.1.2 Validity checks on inputs

When the driver wants to login to the app, they will need to supply two pieces of information: an email address and a password. To check if an email address is valid, the app will check if there is an '@' followed by a domain name. For example, rodeo_town@gmail.com is a valid entry, but rodeo_town, rodeo_town@gmail, @gmail.com, @.com, and @com are not valid. For passwords, the user must enter one that is at least eight characters long and contains at least one uppercase letter, lowercase letter, and digit. Should the user fail to correctly supply a valid email address or password, they will be prompted to re-enter the information correctly.

3.2.2 Customer role

3.2.2.1 Requesting a driver

The primary action that a customer will be able to make is requesting a driver. This is the act of having a Rodeo Town Taxi driver pick the customer up at their current location by pressing an on-screen button. When the customer activates the corresponding button on their screen, their location is broadcast to the driver that will pick them up.

3.2.2.2 Dialing Rodeo Town dispatch

This is a function that allows the user to dial the phone number for Rodeo Town Dispatch at the press of the corresponding number on the screen. This can be useful if the server functionality of the app is currently inaccessible.

3.3 Nonfunctional requirements

3.3.1 Modularity

In the development of this application the software will be split into multiple programs and functions so that there are less areas that may cause a catastrophic failure or crash of the app. Any error can be pin pointed to certain locations in the software. The application is to be broken down into the client and driver side, server manipulation, database manipulation, and Google Maps location use. Within these classes there will be numerous functions, for example: creating a marker on the map for the client, creating a marker on the map for the driver, requesting and assigning a driver, assigning a client to a driver, and updating the map with the marker for the driver. Being separated the code can be easily tested and updated if necessary.

3.3.2 Scalability

The scalability of this application can be done through manipulating the server on Google Cloud to better suit the client's needs later if their priorities change. The memory and storage can be updated with existing data on the server depending on the usage of the application. The code is being developed in a modular and maintainable fashion so if there are needs to add features or remove certain features it can be done in an efficient and easy way to quicker scale the application to the client's needs.

3.3.3 Availability

This application will be available 24/7 as is the current taxi service without the application. Using a Google Cloud server will help to ensure that the server will not go down through a reputable company unless there is a catastrophic failure. There will be continuous power, data backup, recoverability, and backup power sources using Google Cloud in the development of this application. The only foreseen unavailability is if the user does not have an Internet connect or if their device loses power as they would then not be able to use the application.

3.3.4 Maintainability

This application will be easily maintainable with concise and descriptive documentation in all necessary portions of the programs. If there will be any need to add features, you can simply update the database in the Google Cloud server or increase the storage and performance as necessary. Any bugs you can use the search function to find areas of the code that may be affected and work through functions to address certain errors although we do not expect to need to address this upon the release of the application. There will be vigorous testing, properly formatted code, and automated testing.

3.3.5 Reliability

This application will be extremely reliable and nearly fault free using a Google Cloud server to store and process data. Google Cloud has many backups, is always maintained, and runs 24/7 if there are available funds in the client's Google Cloud account. If there are any issues related to the server, Google Cloud support can be contacted at their current phone number 1-855-817-0841. The application will have checks in place to ensure that the software does not crash. This is especially important for the driver interface of the app, because a crash could interrupt the driver's workflow by up to a minute in restarting the app. The app will be tested to be able to run continuously for a full workday without crashes. The app should be restarted, then, at the start of every workday.

3.3.6 Usability

This application will satisfy the needs of the diverse client base that Rodeo Town Taxi services. With the diverse clientele there will be user testing among many age groups to better determine the ease of use and where certain aspects may need to be updated or slightly modified while developing this application. The dominant client base focuses on college aged students who are very well educated in the use of such mobile technologies, however, the app will be built for persons less educated in this technology as well.

3.3.7 Portability

This application will be extremely portable and widely available on any device. Using a Google Cloud server means that data can be accessed from any device anywhere in the world although the service is only available in the area of Ellensburg, WA. The user can download the application on either iOS or Android while on the go if they are connected to the Internet to hail a cab or request multiple services at any time of the day.

3.4 Logical database requirements

The database associated with the Rodeo Town Taxi App will contain a table called "users," that stores user information. The table will have four columns: name, email (primary key), password (hashed), and role (1 for customer, 2 for driver). The role column will be necessary if the development team has time to implement the desirable but not necessary requirement of letting customers create accounts.

3.5 Requirement consistency check

The requirements given to us by the client are consistent and unwavering. When the client describes what he wants from his taxi app, he does so in a logical and focused fashion. The requirements from the client pass the consistency check because all of the goals match the overall idea of a taxi app.

3.6 Requirement validity check

Because the app is meant to do basic actions like requesting a cab driver, it is easy to validate the requirements for it. Because we are not dealing with information that requires knowledge from someone who works in the taxi industry, we are able to confirm the correctness of the information provided by the client. Thus, we are satisfied with the validity of the data given to us by the client.

3.7 Requirement feasibility check

After reviewing the requirements with the team, the consensus is that the highest priority requirements can be done within our ten-week deadline. Because we are dealing with requirements that rely on basic map and database functions, we should have no trouble getting the app done on time.

3.8 Requirement prioritization

The focus of this SRS has been on the requirements that the client deems absolutely necessary. However, the client has also presented to the team requirements that may be considered when the primary ones are completed. The following is a list of requirements grouped by necessity.

3.8.1 Necessary requirements

- Create a button for the customer that requests a taxi driver
- Have a map that displays the customer's location to the driver and the driver's location to the customer
- Allow the user to request a ride without an account
- Create a user account database and give the client administrator access to it so they can remove users from it if necessary
- Create a button that dials the Rodeo Town Taxi dispatch phone number
- Let the user know if a connection to the server cannot be made

3.8.2 Requirements that are highly desirable but not necessary

- Allow the customer to create an account
- The customer can specify what service they want the driver to perform when requesting them (car lockouts, deliveries, etc.)
- Allow the client to give free rides to a customer account
- Implement a rewards system that gives the user a free taxi ride after having seven rides
- Implement a messaging system between the driver and customer

3.8.3 Requirements that are possible but can be eliminated

- Prompt customers to rate their rides on a 5 star system with optional text feedback
- In-app promotions that affect all users such as discounted rides for a short time period

Index