Ridesharing Web Application

Parth Kapur
Seren Yavuz
Thomas Gorney

Software Requirements Specification Document

Version: 1 **Date: 2/24/2019**

Customer Description and Consent of Use

Application F:
“The software application idea I have is a website. This website is a ride sharing website which differs from ordinary rideshare services like Uber or Lyft. This website’s purpose will allow users around the world to rideshare with each other. The defining difference between Uber or Lyft is, Uber/Lyft does not fulfill the ability to plan ridesharing ahead of time. Uber, Lyft, and other rideshare companies offer on the spot ridesharing. Sometimes, however, that is not what is desired by the user. People may wish to plan their trips out ahead of time for a variety of reasons including following an itinerary or cost saving. Their only option currently is to book a taxi which can be a costly investment and defeats the purpose of planning. My rideshare website will accomplish and fulfill the needs of this target market (planned rideshare). The tools that I am planning to use to build this website include: HTML, CSS, JavaScript, jQuery, Bootstrap, PHP, MYSQL, Google Maps API, AJAX, and JSON. I may not utilize all these tools, but I can imagine seeing a need for each of these languages. The website will work by prompting a user to sign up or log in. If user needs to sign up, upon successful signup the user will receive a push email to active his or her account. A user should also be able to change his email address or reset his password if required. Once this is done, the user should be able to put a start and end destination into the Google Maps API. When this action is completed, there will be a called to the MYSQL database to query all drivers willing to drive from the start destination to the end destination. This query will display drivers name, vehicle, cost per seat, departure time, phone number, etc. The user will also be allowed to add a trip, edit a trip, or delete a trip if the user himself/herself wants to offer a ride sharing service from a destination to an end destination. Additionally, a user should have the ability to update his profile information with pictures, details, etc. As more and more users sign up, I hope to cover the entire area of San Diego!”

Project Owner: Parth Kapur
Email: kapur002@cougars.csusm.edu

Signature: Parth Kapur **Date: 2/24/2019**

1. Introduction

1.1 Purpose

Transportation is an important yet tricky aspect of daily life. If a person does not own a car, transportation through other means can be quite expensive. Public transportation, while an option, is cost effective but at the expense of time. RideShare provides everyone the chance to supply or demand the ride service whilst being both cost & time effective.

Our target users range from those who would like to share their cost when traveling, to those who desire company for the road trip. If the user needs a ride, our website will act as the middleman to those willing to provide the service to the end user!

1.2 Scope

RideShare is inexpensive and convenient application that enhances the carpool lifestyle. Users query their starting point and destination. Users can also add a trip if they wish to offer their carpool services to another user. As a result, users can be someone who is planning a trip or a driver who is willing to offer their vehicle as a method of transportation.

It is all supply and demand base product. It doesn't guarantee the ride. If one user's request matches with the other user's then it's a planned ride.

The RideShare application benefits everyone. Let's say a user is driving to Los Angeles from San Diego. The user shares planned trip in details, such as when, where to and from, type of the car, and available seats and if there is another user who looks for a ride to Los Angeles from San Diego the application will display all matching travelling users.

1.3 Definitions, acronyms, and abbreviations

API: Application Programming Interface; is a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

CSS: Cascading Style Sheets; is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

GPS: Global Positioning System; is a global navigation satellite system that uses at least 24 satellites, a receiver and algorithms to provide location, velocity and time synchronization for air, sea and land travel.

HTML: Hypertext Markup Language; is the standard markup language for creating web pages and web applications.

HTTP: Hypertext Transfer Protocol; is the protocol used to transfer data over the web. It is part of the internet protocol suite and defines commands and services used for transmitting web page data.

JSON: JavaScript Object Notation; is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types.

PHP: Personal Home Page; is a script language and interpreter that is freely available and used primarily on Linux web servers.

VARCHAR: Variable Character Field; is a set of character data of indeterminate length.

1.4 References

Currently not applicable.

1.5 Overview

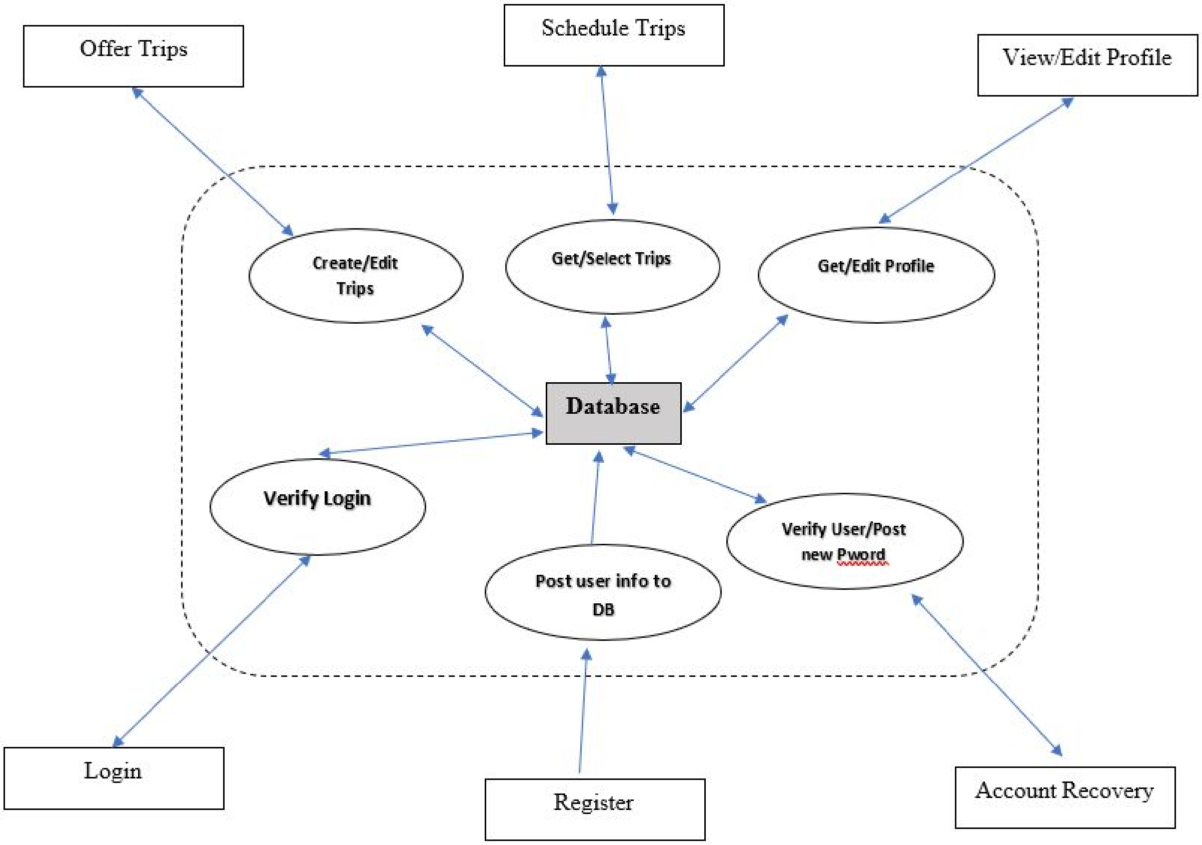
The report consists of the overall description including product perspective; system interfaces, user interfaces, hardware interfaces, software interfaces, and communication interfaces. Following these, the report will include the product functions, user characteristics, constraints, assumptions and dependencies, apportioning of requirements. In conclusion, specific requirements will be covered including external interfaces, functions, performance requirements, logical database requirements, design constraints, software system attributes, and organizing the specific requirements.

2. Overall Description

2.1 Product Perspective

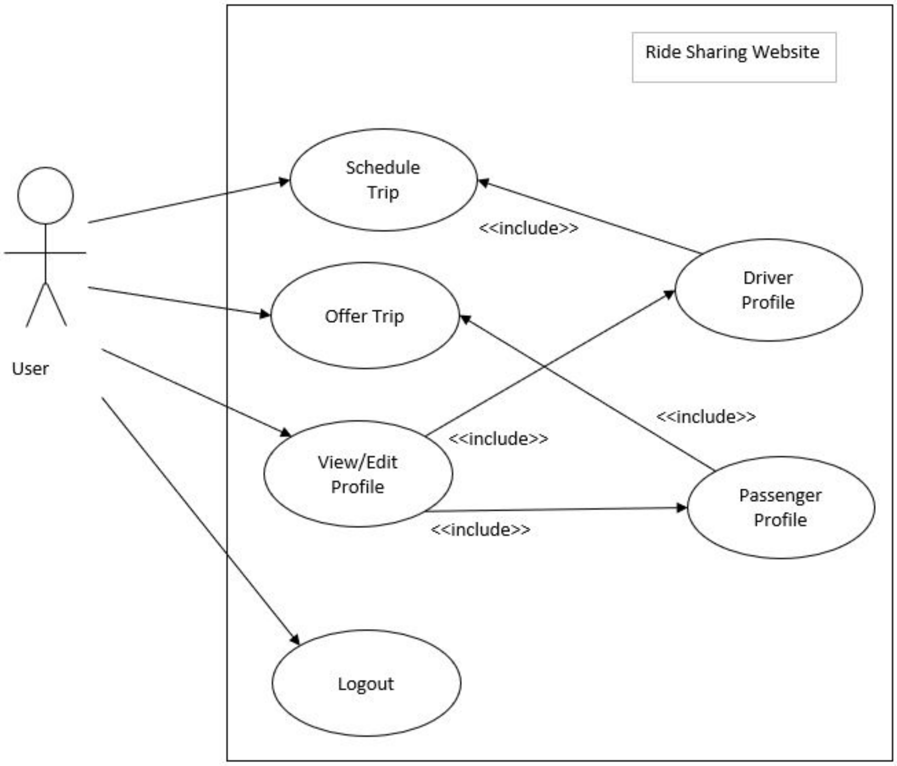
This ride sharing web application differentiates itself from other competing ride sharing applications by allowing users to schedule their ride sharing in advance. This service will be provided through a web application that can be used easily on the phone. Users will be able to login, customize their profile, and schedule rides.

2.1.1 System Interfaces



2.1.2 User Interfaces

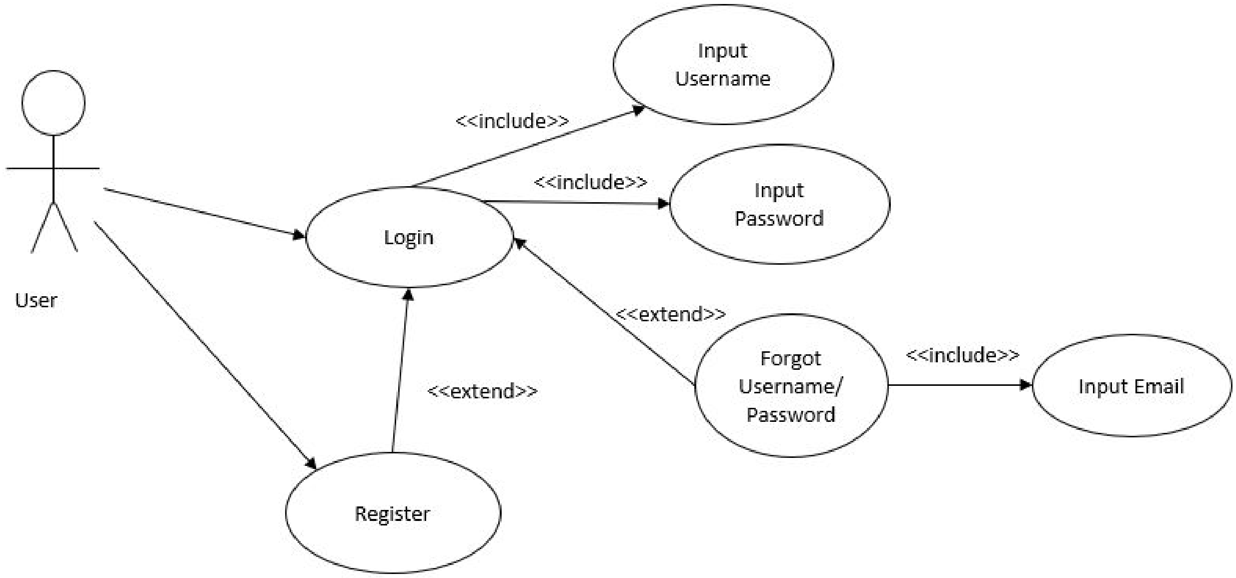
There will be 3 main web pages in this web application. There will be a portal view, where users can login or sign up. There will be a Ride Sharing page, where users can sign up for rides or offer rides, and there will be a profile page, where users can view other users' profile pages.



Portal

The portal page provides various user interfacing with account login or creation. The portal page will take you to the RideSharing main page.

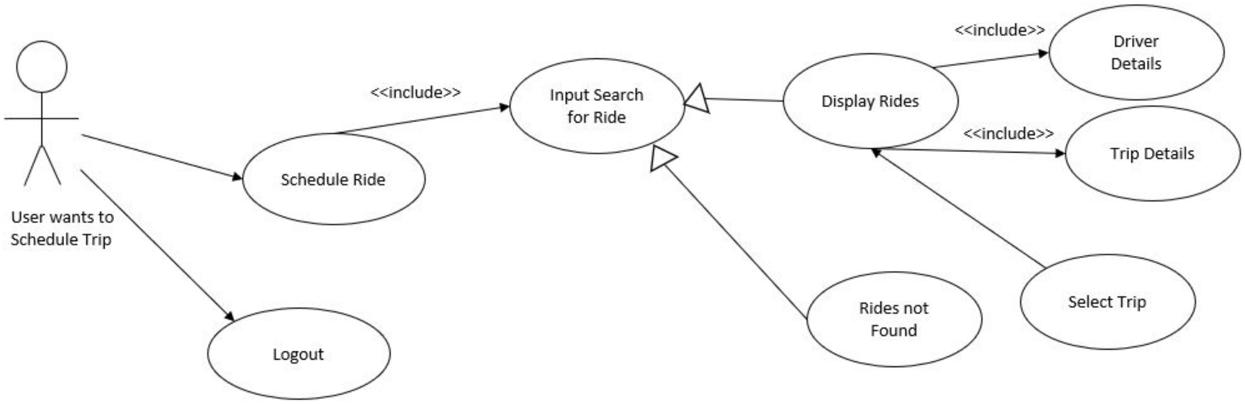
- Login: The user user will login to a user portal. Which will direct them to the ride sharing home.
- Forgot Username: The user will click this button if they forget their username. This will redirect them to another page where they can input their credentials to verify their account and retrieve their username..
- Forgot Password: The user will click this button if they forget their password. This will redirect them to another page where they can input their credentials to verify their account and retrieve their password.
- Create Account: The user will click on this button to create a new account for the website if they have no account already. Upon successful creation, the user will receive an email to verify their account.



RideSharing Page

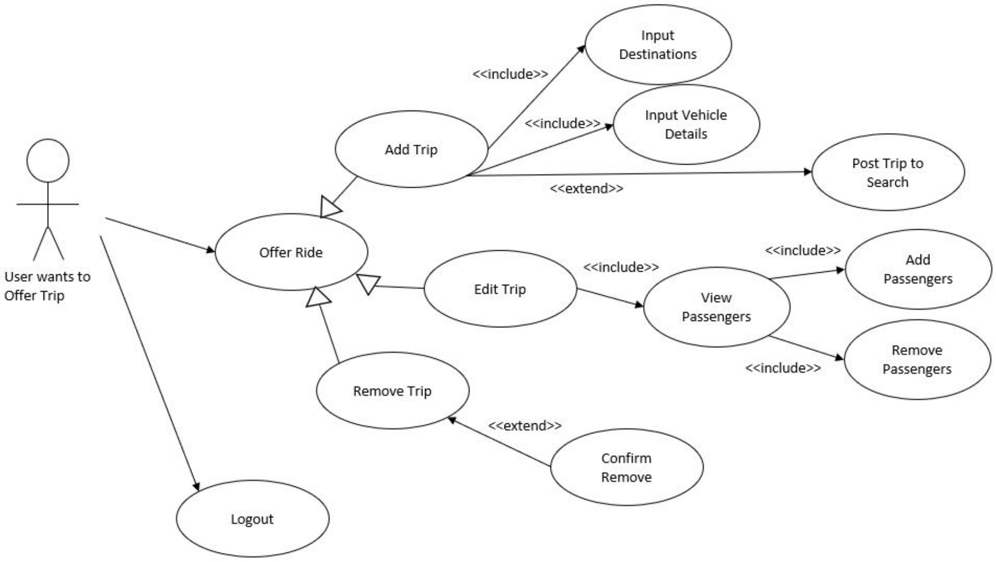
The RideSharing page is where users will interface with various buttons and prompts in order to schedule a ride share.

- **Schedule a Ride Share**: Users can schedule a ride share by inputting their start and end destination into the Google API.
  - Search: The user will use this field to input their start and end destination and date(s) to travel.
  - Display: The search will then output a display of all the drivers that are willing to make that trip including the driver's name, vehicle, cost per seat, departure time, phone number, etc.
  - Select: Users will then be able to make a selection based on their Display results. Users will select their preferred driver and the driver's vehicle will be updated with a taken seat.



- **Offer a Ride Share**: Users can choose to offer a ride share instead if they have a suitable vehicle.

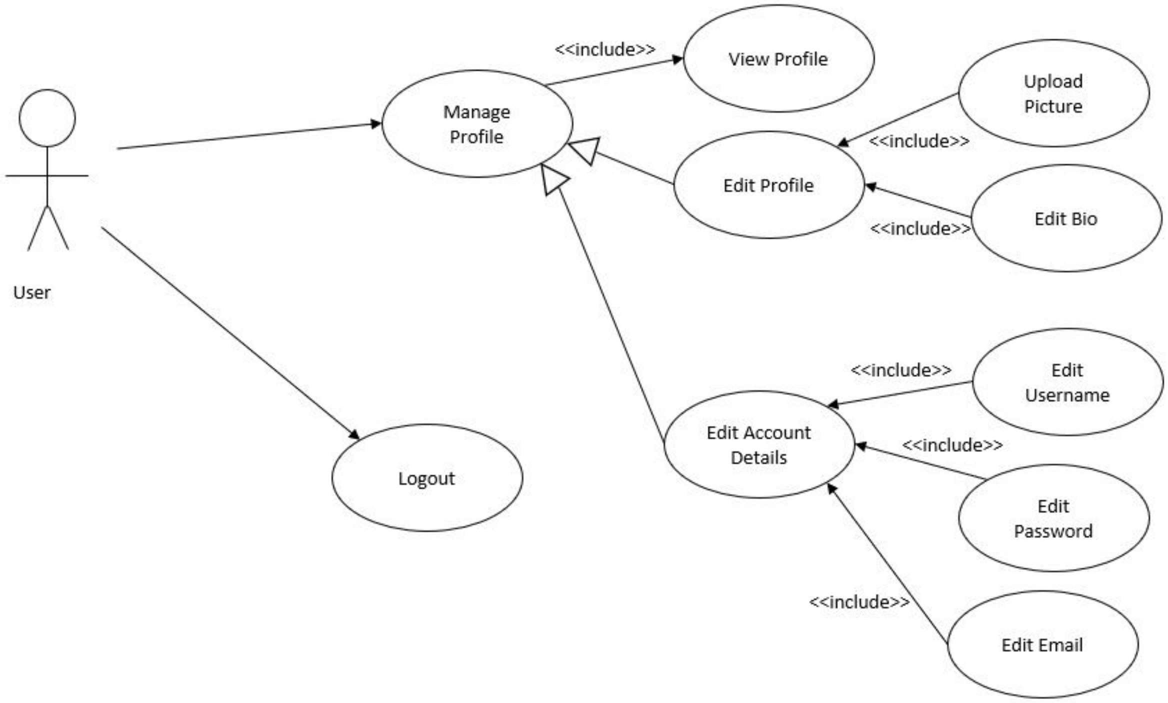
- ○ Add Trip: The user can add a trip by inputting their start and end destination, their vehicle details, and dates.
- ○ Remove Trip: The user can remove their added trips through a delete trip button.
- ○ Edit Trip: The user can edit their trip details such as vehicle details, dates and destination.
- ○ View Users: The driver can view the users' profiles that have signed up for their trip.



Profile

The profile page is so that users can look at other users profiles to see if they wish to rideshare with them.

- Edit Profile: Users edit their profile details to include a picture of themselves as well as any information the wish to provide.
- Edit Account Details: Users edit their account details in the profile page as well. These details include, username, email, and password.



2.1.3 Hardware Interfaces

There are only two hardware interfaces in this web application. There is the interface through a desktop computer to connect to the web application, and there is interface through a phone to connect to the web application. Considering the structure of the internet, users should be able to use any desktop computer (Windows, Linux, Apple, etc), or any phone to connect to the web application (iOS, Android, etc.).

2.1.4 Software Interfaces

We will be using various tools, libraries, and integrations in order to provide this ride sharing service:

- HTML/CSS/JavaScript: Will be used for the general structure of the web application. This will help control page layout and style.
- jQuery : JavaScript library used to simplify HTML traversal, event handling, animating, and Ajax interactions for rapid development.
- Bootstrap: HTML/CSS/JavaScript framework used to develop responsive, mobile-first websites.
- PHP: Scripting language used for server-side scripting which will help interface between the database and web application.
- MySQL: Relational database management system that will be used to store account details, and trip details.

- Google Maps API: This will be used to allow us to use Google Maps as our mapping tool when scheduling and finding routes.
- AJAX: Set of web development techniques used to create asynchronous applications. We may use this to communicate with the database without having to reload the web page.
- JSON: We may be using JSON in order to transmit structured data over a networked connection.

2.1.5 Communications Interfaces

Some communications interfaces will be used to help transmit data over a network and provide some interaction with the user and database.

- Web Browser: Users will interact with our web application through a web browser of their choice. All major web browsers should be supported on a desktop or phone.
- HTTP and HTML: HTTP and HTML are used to send and display web pages from the web server to the user.
- Email: Email will be used to send out account verification and creation requests.

2.2 Product Functions

WIth this web application the users will be able to ride share to specific destinations. There will be account functionality so that users can login, logout, and edit account details. There will also be a profile page so that users can view other users' profiles.

Portal
The first page will bring the user to a portal page. In this page the user will be able to Login to their account, create an account, or verify account in order to retrieve their username or password.

Rideshare
After logging in the user will be brought to the ride sharing page. In this page the user will be able to search for a possible rideshare that they can join and eventually be brought to their location. You can logout of your account here and be brought back to the portal page, you can also view profiles to help screen drivers or other participants on the trip.

You will also be able to create your own trips if you are the person wanting to rideshare. Users can create a trip and others can sign up for the trip. The user can also edit the trip details, such as destination, and car information. The user can also delete their trip if they are no longer able to make the trip.

Profile

If the user is viewing their own personal profile, then they will be able to edit their details such as, name, age, date of birth, bio, picture, etc. If the user is viewing another user's profile then they will be able to view these public details to see if they would like to share a ride with that person.

2.3 User Characteristics

There are two types of users that will be interacting with the site. Users that wish to join a trip, and users that wish to create a trip. These two different types of users both have the same general functionality in the website. The portal is the exact same, as well as the basic rideshare main page.

Users who wish to join a trip will login through the portal and then they will be able to search for a trip. The search output any rides that are close to their destination. The user can then click to join a specific trip.

Users who wish to create a trip have the same functionality as users wishing to join and vice versa. However, they can create trips, have people join their trip, and ideally be paid upon delivering their customers to their locations.

Any user is either a person looking to join a trip, or a person looking to setup a trip. Therefore, everyone has the functionality to join or create trips, but not everyone will want to. Some will just join trips, while others will just create trips.

2.4 Constraints

The overall look and feel of the website is constrained by the web browser and mobile phone you are using. Considering everyone uses different web browsers, and different phone manufacturers; the interfaces may look slightly different on each one of them.

The mapping and GPS functionality are constrained by Google Maps API. As we have decided to use Google Maps API we are forced to use google maps as our map for the web application, which means other mapping tools when using GPS functionality may not work.

Internet connection is also a constraint on the web application. Since our application is a web service, an internet connection is required to use it. If there is not internet connection then we cannot access google maps or the database. The web application is also constrained by the database.

2.5 Assumptions and Dependencies

An assumption is that all users will be accessing our web application on platforms that are powerful enough and supported. Users may not be able to access our site using very old or out of date browsers/phones. We must also assume that the devices connecting to our site all have the potential to use Google Maps API.

Users that access our site should also have internet service, otherwise they would not be able to connect to the website. We can assume that if the user is able to connect to our site's portal then they are able to interact with everything else on the site.

Another assumption is that the driver of the trip will handle all payments. As we do not have a pay service setup, it is up to the driver to handle all payment interactions with their customers.

2.6 Apportioning of Requirements

A major function of the site that should be delayed to another future version is the payment service. Currently we do not have any plans on implementing a payment service for this phase of the site. The idea is to create the structure and general functionality of the site. Any real pay service associated with the site will need to be secure and legal. This would require work outside the scope of this phase of the project.

3. Specific Requirements

3.1 External Interfaces

a) Name of Item	Login
b) Description of Purpose	This will allow the user to login to the system
c) Source of input or destination of output	Login.php
d) Valid range, accuracy, and/or tolerance	100% Accuracy, Zero Tolerance
e) Units of measure	Text
f) Timing	N/A
g) Relationships to other inputs/outputs	Successful login grants access to Offer Trips, Schedule Trips, View/Edit Profile, Account Recovery
h) Screen formats/organization	Desktop, mobile friendly
i) Window formats/organization	N/A

j) Data formats	PHP, MYSQL
k) Command formats	N/A
l) End messages	Login Successful/Unsuccessful

a) Name of Item	Register
b) Description of Purpose	This will assign a user account to a user.
c) Source of input or destination of output	Signup.php
d) Valid range, accuracy, and/or tolerance	100% Accuracy, Zero Tolerance
e) Units of measure	Text
f) Timing	N/A
g) Relationships to other inputs/outputs	Grants access to Login.php
h) Screen formats/organization	Desktop, mobile friendly
i) Window formats/organization	N/A
j) Data formats	HTML, PHP
k) Command formats	N/A
l) End messages	Registration Successful -> Activation Link in Email

a) Name of Item	Account Recovery
b) Description of Purpose	This will reset password for user.
c) Source of input or destination of output	Forgotpassword.php
d) Valid range, accuracy, and/or tolerance	100% Accuracy, Zero Tolerance
e) Units of measure	Text
f) Timing	N/A
g) Relationships to other inputs/outputs	Successful recovery grants access back to user account
h) Screen formats/organization	Desktop, mobile friendly
i) Window formats/organization	N/A

j) Data formats	HTML, PHP
k) Command formats	N/A
l) End messages	Activation Link Sent To Specified Email

a) Name of Item	Offer Trips
b) Description of Purpose	This will allow the user to offer trips
c) Source of input or destination of output	AddTrips.php
d) Valid range, accuracy, and/or tolerance	100% Accuracy, Zero Tolerance
e) Units of measure	Text
f) Timing	N/A
g) Relationships to other inputs/outputs	Updates Database
h) Screen formats/organization	Desktop, mobile friendly
i) Window formats/organization	N/A
j) Data formats	HTML, PHP
k) Command formats	N/A
l) End messages	Add trip successful

a) Name of Item	Schedule Trips
b) Description of Purpose	This will allow the user to search trips
c) Source of input or destination of output	Index.php
d) Valid range, accuracy, and/or tolerance	Accuracy based on driver availability
e) Units of measure	N/A
f) Timing	N/A
g) Relationships to other inputs/outputs	Updates Database
h) Screen formats/organization	Desktop, mobile friendly
i) Window formats/organization	N/A
j) Data formats	HTML, MYSQL, Google Maps API

k) Command formats	N/A
l) End messages	Show query of desired trips and drivers

a) Name of Item	View/Edit Profile
b) Description of Purpose	This will allow a user to make changes to his/her profile.
c) Source of input or destination of output	Profile.php
d) Valid range, accuracy, and/or tolerance	Accuracy dependent on user provided information
e) Units of measure	N/A
f) Timing	N/A
g) Relationships to other inputs/outputs	Updates Database
h) Screen formats/organization	Desktop, mobile friendly
i) Window formats/organization	N/A
j) Data formats	HTML, MYSQL
k) Command formats	N/A
l) End messages	Profile updated successfully

3.2 Functions

The system shall (for Login):

a) Perform necessary validity checks on inputs

b) Call login process form, which will then query database. Upon successful query, user is directed to profile page.

c) If abnormal situation, generate generic error to notify user login is not functioning.

d) Login parameters will be verified with database

e) Upon proper input, grant access to all features of the website including: scheduling trips, offering trips, view/edit profile, account recovery.

The system shall (for Register):

a) Perform necessary validity check on register form

b) Call activation email process. Once account is activated, update database to reflect this to allow access to all website features.

c) If abnormal situation, generate generic error to notify registration is not functioning.

d) Registration form parameters will be verified with database

e) Upon proper input, activate user account

The system shall (for Account Recovery):

a) Perform necessary validity check on email form

b) Call reset password process. Once called, send email to user’s account email to reset password. Once linked is clicked, allow password to be reset.

c) If abnormal situation, generate generic error to notify account recovery is not functioning.

d) Recovery form parameters will be verified with database

e) Upon proper input, reset user password

The system shall (for Offer Trips)

a) Perform necessary validity check on adding trip

b) Push to database with added trip, add to database, display success to user.

c) If abnormal situation, generate generic error to notify offering trips is not functioning.

d) Location parameters will send information to database regarding trips being added.

e) Offering trips will make changes to the data stored in the database

The system shall (for Schedule Trips)

a) Perform necessary validity check on user query

b) Call from database the query, display allocated trips to the user

c) If abnormal situation, generate generic error to notify querying trips is not functioning.

d) Starting and ending destination parameters will send information to database to aid in querying relevant trips.

e) Query will change content of webpage.

The system shall (for View/Edit Profile)

a) Perform necessary validity checks

b) Call to database to obtain user profile information. Push to database if user makes changes to profile.

c) If abnormal situation, generate generic error to notify querying trips is not functioning.

d) Profile form parameters will serve as both push and call from database.

e) Profile query will change content of webpage.

3.3 Performance Requirements

a) The website should be able to be run on any modern browser, including, but not limited to: Google Chrome, Firefox, and Safari.

b) The database should be able to support up to one hundred users in testing phase. When website goes live, database and server should be updated regularly to handle request requirements.

c) The amount and type of information to be handled will mainly be HTML, PHP, CSS, JavaScript, and Google APIs.

3.4 Logical Database Requirements

a) The types of information to be used by various functions will be: INT, VARCHAR, DATE, and other MYSQL supported data types.

b) The information will be accessed frequently.
c) The user will indirectly access the database through HTML and underlying PHP code.
d) A users table should be able to add many trips and schedule many trips. There should be a one-to-many relationship for most tables in the database.
e) A user should not be allowed to access the database directly, only indirectly.
f) Data must be retained until user account is inactive for 180 days.

3.5 Design Constraints

This website should be modern and able to run on any modern-day web browser. The minimum hardware limitations will match those set for the client's web browser.

3.5.1 Standards Compliance

a) Report format:
- This website is to have database logs for all transactions done on the website for 365 days.
b) Data naming:
- This website is expected to encrypt all critical data.

3.6 Software System Attributes

3.6.1 Reliability

This website should be expected to continue operation year-round, only ceasing operation for routine maintenance.

3.6.2 Availability

This website should be able to be rebooted from server if need be. All user data should be kept on the database for 180 days incase of the need of a checkpoint or restore point.

3.6.3 Security

This website should protect all user critical data with MD5 encryption level hashing.

3.6.4 Portability

This website should be able to run on any system as long as the browser is supported by the website.

3.7 Organizing the Specific Requirements

3.7.1 System Mode

This website should have a guest mode for those who are not website users. Functionality in this mode is limited with only the ability to see trips, but not see which driver is providing these trips. This website should also have a user mode which will unlock all functionality of the website.

3.7.2 Feature

This website should have SMTP push email features in order for login, account recovery, and profile pages to function effectively. There should also be a contact us form if the user needs assistance.

3.7.3 Response

The Google Maps API feature of the website will be reliant on data entered by users who are offering rideshare services to other users. The activation and recovery emails are response functions based on what the user enters into the respective webpages.