

Software Requirements Specification

Road Trip Advisor Web Application

Version 1.0

Printed: 10/02/2018

Road Trip Advisor Team

Beverly Ackah, Shaila Hirji, Frederick Wirtz

Bellevue College

Supervisor: Fatma Serce

Revisions

Version	Primary Author	Description of Version	Date Completed
1.0	Beverly ACKAH, Shaila HIRJI, Fred WIRTZ	Creating initial draft of SRS as a description for the final product	October 15th 2018
1.1	Beverly ACKAH	Updates of the Functional Requirements of the project.	November 5th 2018
1.2	Beverly ACKAH	Removed the Admin section	November 21st 2018
1.2.1	Shaila HIRJI	Edited Meal Preferences image	February 17th 2019
1.2.2	Shaila HIRJI	Edited landing page image, meal preference image	March 4th 2019

Table Of Contents

1. Introduction	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms & Abbreviations	6
1.4 References	7
1.5 Overview	7
2. Overall Description	7
2.1 Product Perspective	7
2.2 Product Functions	7
2.3 User Characteristics	8
2.4 Constraint	8
2.5 Assumptions and Dependencies	9
3. Specific Requirements	10
3.1.1 User Interfaces	10
3.1.2 Hardware Interfaces	12
3.1.3 Software Interfaces	12
3.1.3.1 User Information	12
3.1.4 Communications Interface	12
3.1.4.1 Communication between UI and User	12
3.1.4.2 Communication to UI and API	13
3.2 Functional Requirements	13
3.2.1 User Functionalities	14
3.2.1.1 User access to website	14
3.2.1.2 Register a profile	14
3.2.1.3 Planning a trip	15
3.2.1.4 Login to profile (edit, plan, view saved trips)	15
3.3 Performance Requirements	15
3.3.1 Real Time updates	15
3.3.2 Accuracy	16
3.3.3 Compatibility	16
3.3.4 Safety and Security	16
3.4 Logical Database Requirements	16
3.5 Design Constraints	16
3.5.1 Standards Compliance	16
3.5.2 Web Based Product	16
3.5.3 APIs	16
3.6 Software System Attributes	16
3.6.1 Availability	16
3.6.2 Portability	17

1. Introduction

1.1 Purpose

This document proposes the software functionalities and requirements for version 1.0 of a road trip advisor web service capable of giving the user detailed information about the trip they are planning. It will also explain system constraints, interface, and interactions with other external applications.

1.2 Scope

The Road Trip Advisor is a web application that helps people plan amazing road trips. After entering a starting and destination point, the web application will suggest a couple of road trip alternatives overlaid with different restaurant, hotels and activities options optimized with user preferences. The user preferences will be determined based on the available reviews provided. The users shall be able to view road suggestions for their trips, be able to create a new trip by entering starting and destination points and trip constraints and be able to login and save their trip info.

Routes

- Add a starting location and an ending location along with date and time
 - The application will use the shortest route between two points and not give the user alternative routes.
- Overlays on the map restaurants and lodging along the route within 25 miles from the route.
 - You can hover over the restaurants/hotels to see name/location
 - Change it later to allow the option to change the distance, change selected restaurants on route to other restaurants within the parameter
- Create a route and show places to eat where you would expect the user to be at breakfast, lunch, and dinner. [I think this is the main feature]
- Transfer the route to your phone [like add it onto GoogleMap or Map app]

User accounts

- Logins with site specific, Facebook, Google, or Amazon accounts.

Extra Features:

- Allow users to enter multiple destinations to create a route
- Give road trip alternatives optimized with user preferences.
- Have a 'Profile' page that will save all the reservations (hotel, restaurants, flights, etc..)
- Weather config on trip days, what weather will be like at time of planned event in itinerary
- Speed monitor (only if creating mobile app)
- Budgeting trip 9 unique feature compared to RT)
- Tips with machine learning
- Share trips on social media
- Activities todo
- Cheap gas in the area
- Multiple destinations

- Expected fuel cost
- Suggest destinations based on the duration of the trip (if user plans a 3 days trip, he/she will only go as far as San Francisco, but not France for example)
- Display the minimum and maximum time needed to go to one destination (if the user enters Las Vegas as the ending point, he/she will need at least 18 hours to reach by car)
- Personalized itinerary for each days
- Social Media version: connect travelers. Don't necessarily have to be fb friends to be connected
- Transfer the route to your phone
- Break a multi-day trip into sub trips based on how much a user wants to drive per day (hours or miles) (show lodging options only at places they would stop to spend the night)
- Share routes
 - Allow others to edit a shared route (and an option to NOT allow people to edit a shared route). Make it similar to google docs?
 - Multiple people can edit the route at the same time
- Create a route and show places to eat where you would expect the user to be at breakfast, lunch, and dinner.
- Routes
 - Allow users to enter multiple destinations to create a route
- Select travel options (Car, Bike, Plane, Train, Transit, Etc)
- Create affiliate accounts to make money if someone goes from our website to purchase a hotel reservation or rental car.
- Add Hotels/restaurants/other locations to the trip. Similar to roadtrippers.

1.3 Definitions, Acronyms & Abbreviations

Acronym	Meaning
API	Application Programming Interface
ETA	Estimated Time of Arrival
HTTPS	Hyper Text Transfer Protocol Secure
RTA	Road Trip Advisor
SSL	Secure Socket Layer
UML	Unified Modeling language

1.4 References

1.5 Overview

2. Overall Description

2.1 Product Perspective

Road Trip Advisor (RTA) is a web application for planning road trips. Enter a starting point and destination, and it will provide potential routes with pit stops selected along the way. To allow for easy navigation, users can transfer the routes to their cell phone for turn by turn directions. The goal of this web service is to minimize the number of different applications or services a user would need to bounce between in order to have a successful road trip. We give the user an option to pre plan their trips before they hit the road. The application will not only provide routing options for the user but in fact prepare a whole trip itinerary for them on a day to day basis for the duration of their vacation.

The application will be built upon several different APIs including but not limited to Navigation APIs, Location API, Reviews API and Current Events API. The application will also be relying on some AI technology to enable planning more personalized itineraries based on user's input.

2.2 Product Functions

Road Trip Advisor will have the following core functionalities:

Plan a Trip Module:

- Create a trip
- Save a trip
- Delete a trip
- Edit a trip

Optimization Module:

- Find optimized roads based on user profile

Map Module:

- Display paths on map
- Display details on the trip

User Modeling Module:

- Create user profile (Input meal, accommodations, activities preferences, number of travelers)

Database Module:

- Store user's preferences based on profile
- Store optimized paths

Authentication Module:

- Allow user to log in

Multiple Path Module:

- Allows user to get three different paths and pick the alternative that is the most convenient

Detailed information about the product functions can be found in 3.2 section of this document.

2.3 User Characteristics

There are essentially two classes of users for the Road Trip Advisor: the user of the internal application and an on-site technician. Each of these two users have a different use of the system as well as their own requirements. As for now, we will only refer to the user:

- User: Must be older than 13 years old and must possess the technical know-how to use the interface for creating trips. The user must have a login to be able to save the selected trip. Users can use any other functionalities without having a login . Once the authentication is a success, the user can plan, modify, save, delete or oversee the trip by picking from three different trip alternatives. Different suggestions will be available to give the user more options. In order to get the optimized paths based on user preferences, the user will need to input preferences under the profile page.

2.4 Constraint

APIs

- Localisation must be implemented using Navigation APIs.
- APIs use real-time accurate data.
- Final output may differ based on weather and traffic forecast.

Implementation language restrictions

- The programming language shall be JavaScript for the web application.
- The programming language shall be SQL for the cloud database

Resource limits

- The users' device shall have a working data plan or wifi connection to receive trip information.
- Cloud capacity to hold data .
- Reliability, security, dependability and efficiency of the database.
- Needs to be compatible on most devices.
- Needs to be compatible with most browsers.
- The Google API can only be used for a specific number of time per day

2.5 Assumptions and Dependencies

It is assumed that the following features will be in place:

- The user will use a computer to enter the information needed.
- The data will be exported on the Google Map, so that the user can have access to the details of the trip.
- The data of the final output will be available regardless of the wifi connection.

The final output will depend and differ based on the following:

- Accuracy of the different APIs (directions, locations, etc.)
- Efficiency and Security of the Databases
- WiFi access.
- Weather and traffic forecast,
- Devices used
- Browser geolocation API-- based off the wifi on the computer

3. Specific Requirements

3.1.1 User Interfaces

Returning and first time users, when visiting the Road Trip Advisor website will be presented with an initial landing page (figure 1). This screen offers the user the option of entering a starting point as well as a destination point. The user can also enter date and a time of the trip. The ‘Search’ button will take the user to another page where a map will be displayed.

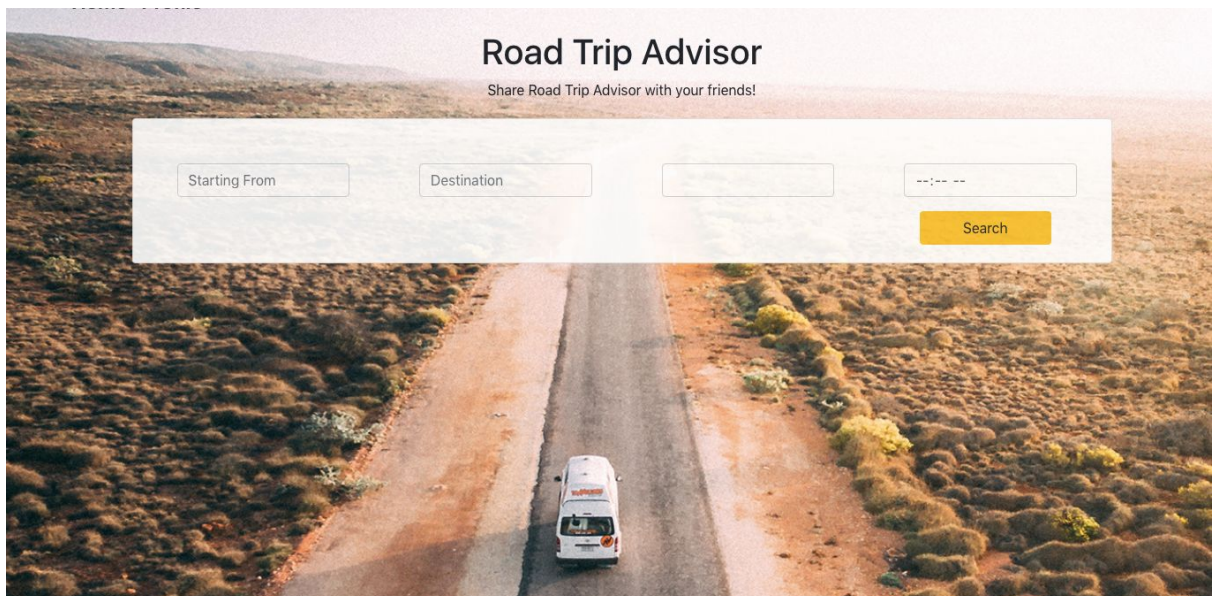


Figure 2 displays the screen the user will see after hitting the 'Search' Button. The user will be able to visual where the starting and ending point will be. The markers both indicate the intended locations.

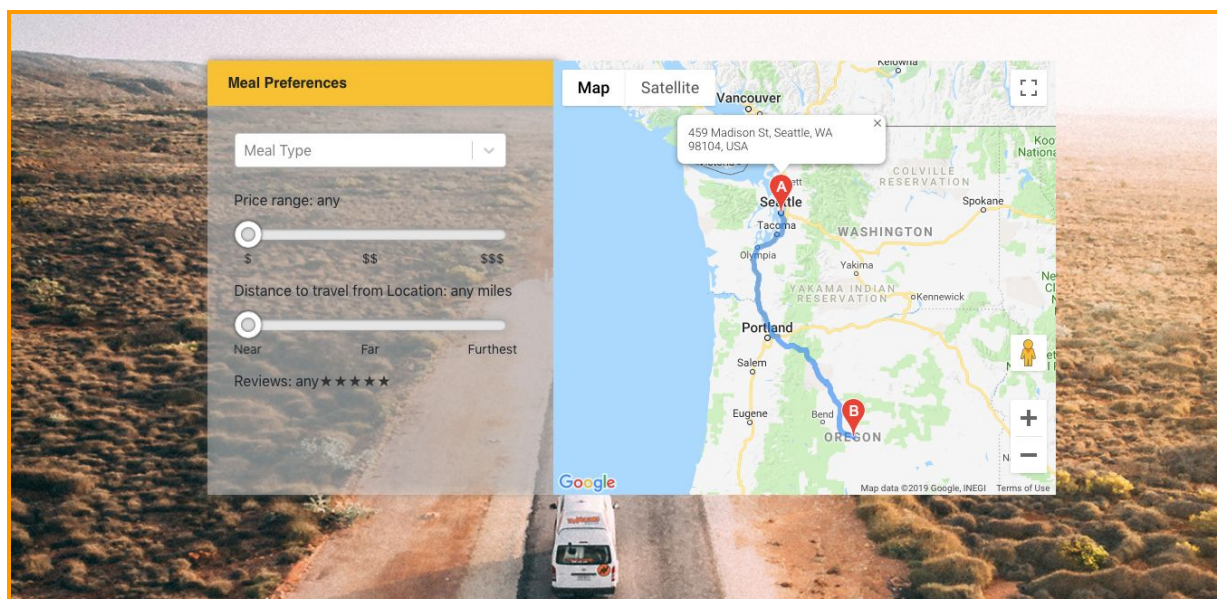
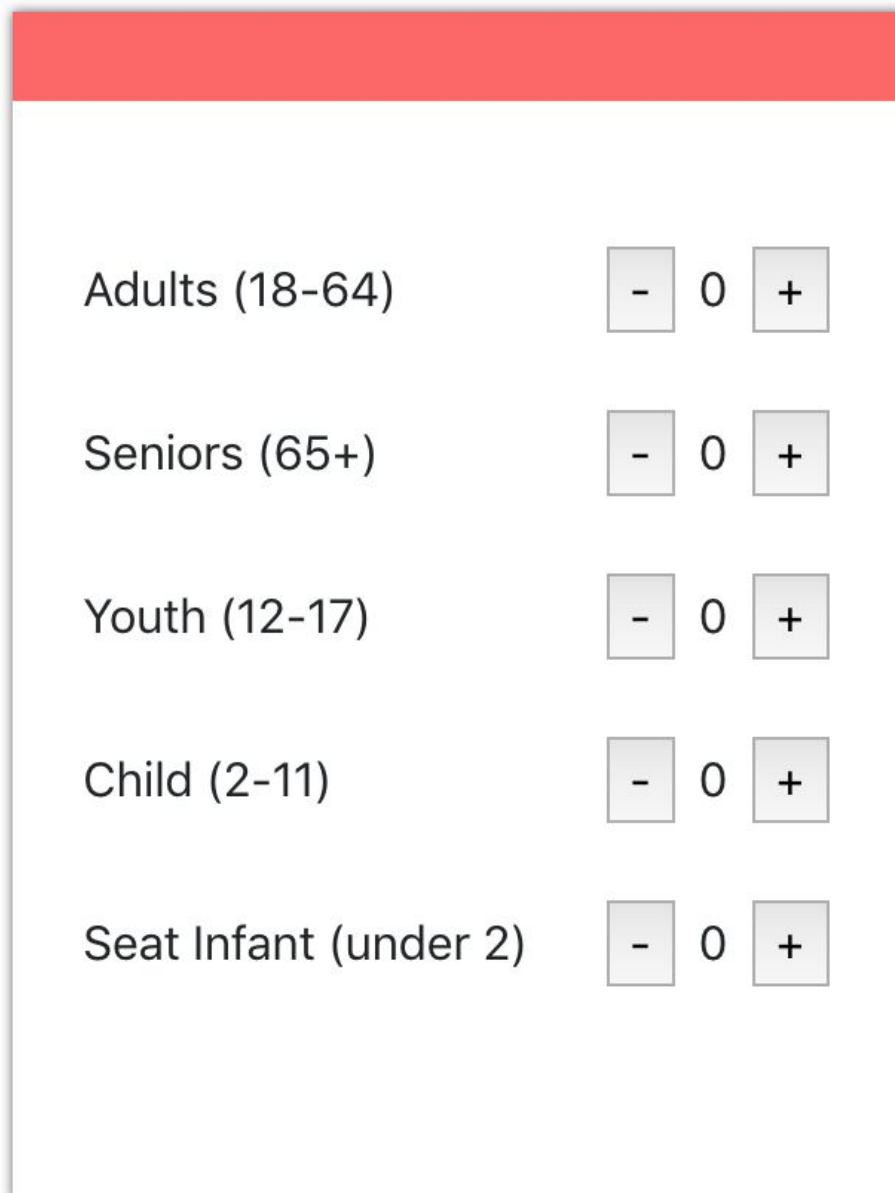


Figure 3 displays the profile the user can make that allow them to customize their trip based on their preferences. The user shall enter information such as: the number of people traveling, their meal preferences, the range of the age of the travelers, the type of events they prefer.

Profile



The profile form features a red header bar at the top. Below it, there are five rows, each representing an age category. Each row consists of a text label on the left and a set of three buttons on the right: a minus sign (-), the number 0, and a plus sign (+). The categories and their corresponding buttons are:

Adults (18-64)	-	0	+
Seniors (65+)	-	0	+
Youth (12-17)	-	0	+
Child (2-11)	-	0	+
Seat Infant (under 2)	-	0	+

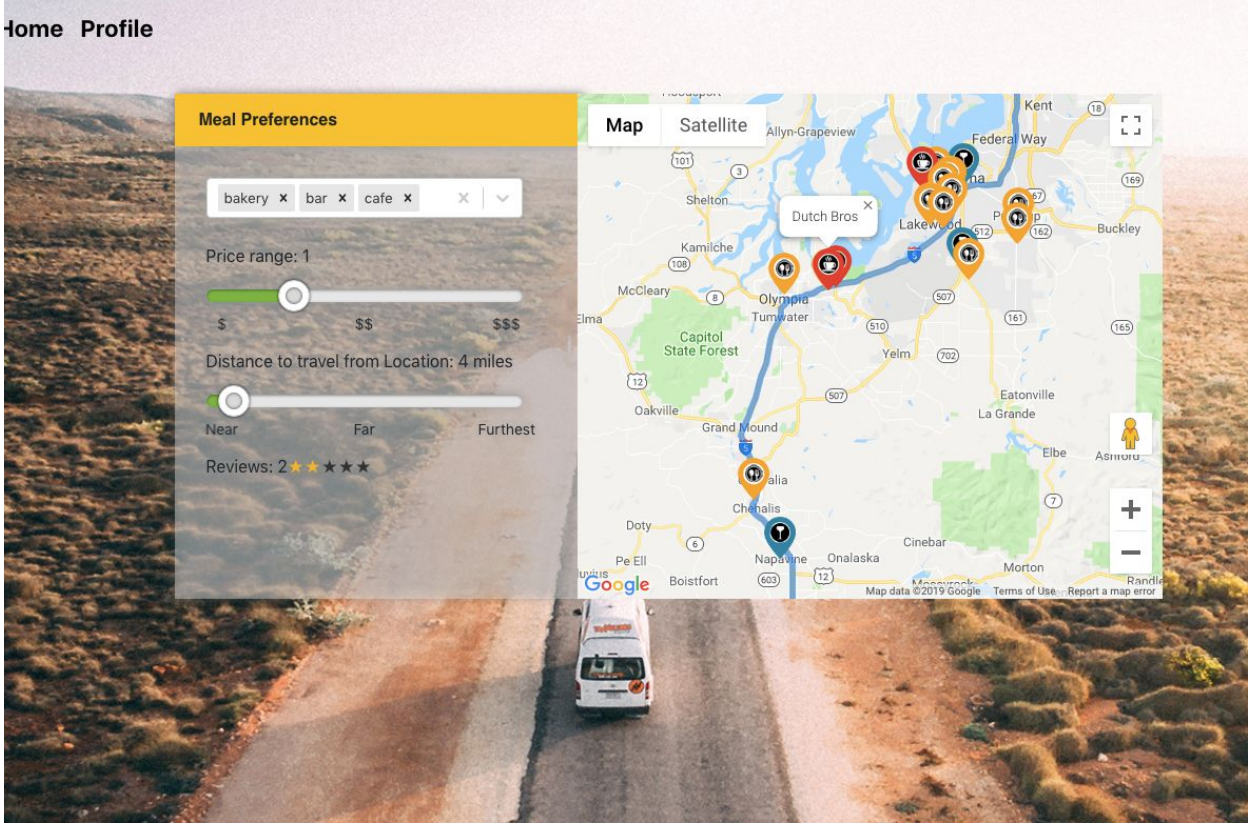


Figure 4 displays the UI for user to enter their meal preferences options. From the select bar they can select the type of places they are interested in for meals, the price range for how expensive they want their meal to be, how far away from their route can the restaurant be and how much the review or ratings for the restaurant should be.

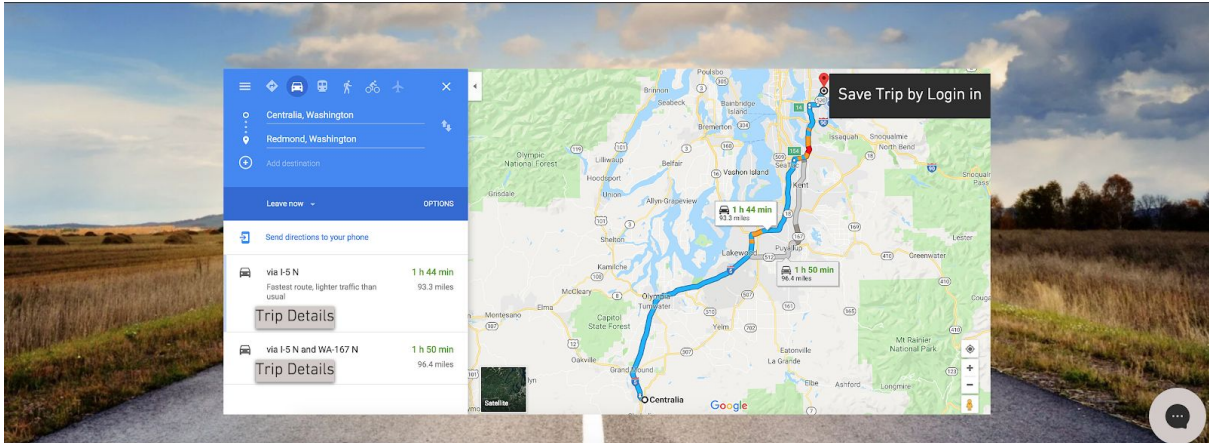


Figure 6. displays the trip details.

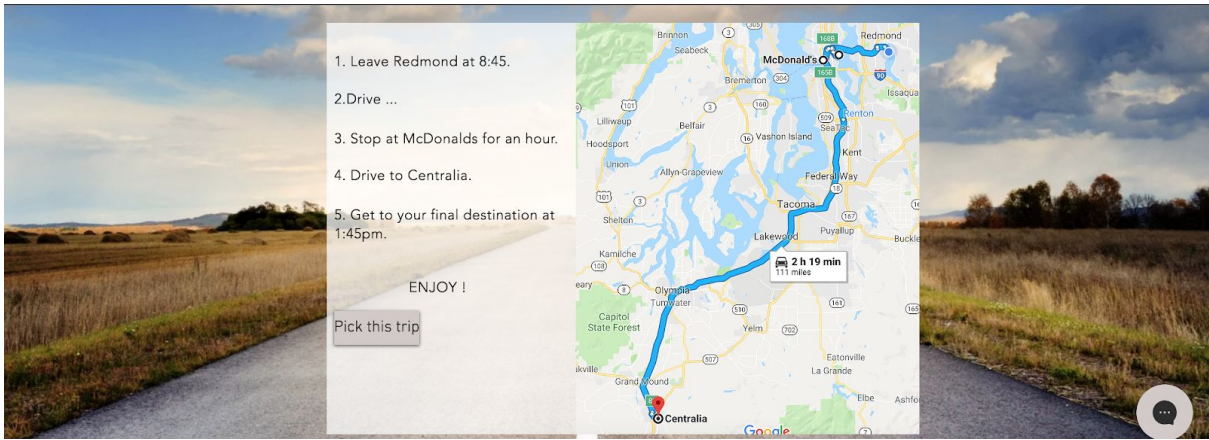
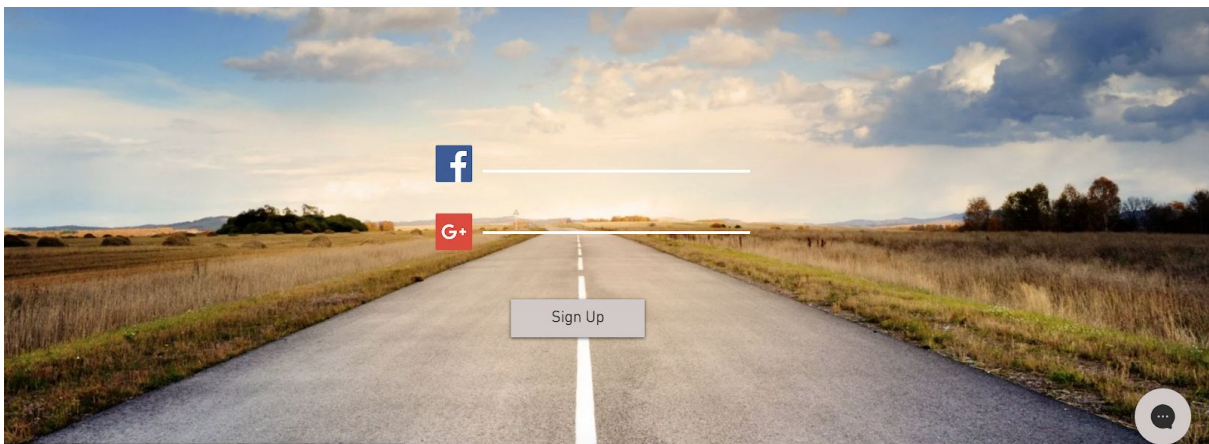
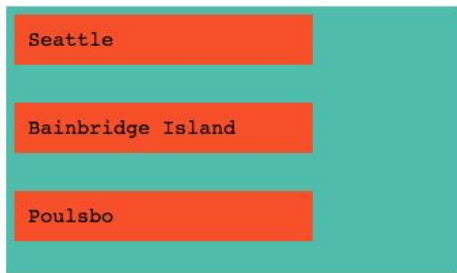


Figure 7. displays the login page for a user to be authorized to save the trip on RTA.





1

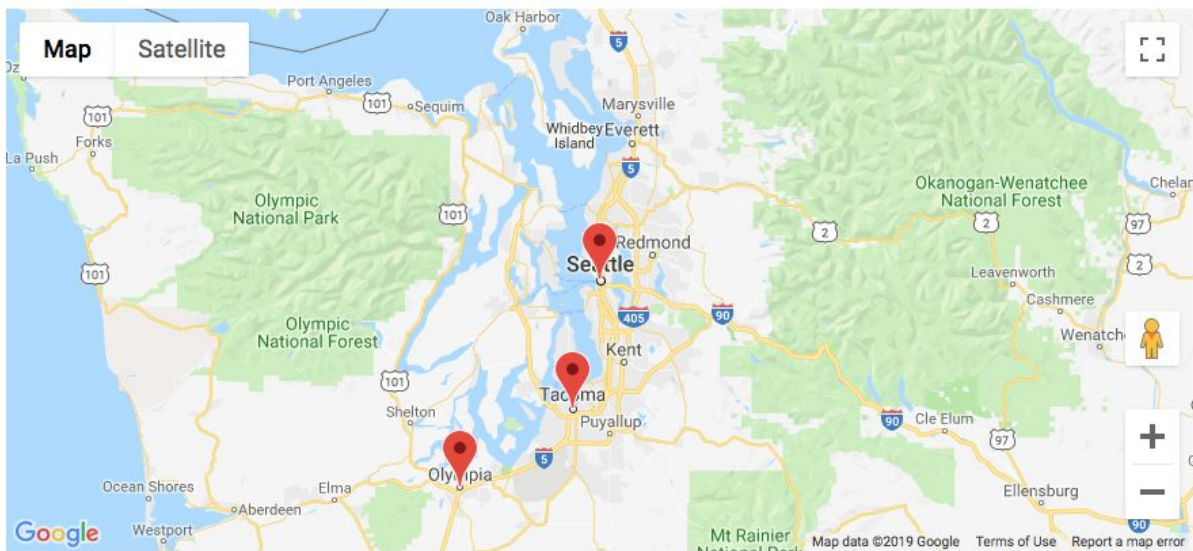


Figure 8 shows the UI for picking a route to display. Clicking on an itinerary will display the that itinerary on the map and hide of other routes. Map will be moved to be beside or behind the routes shown.

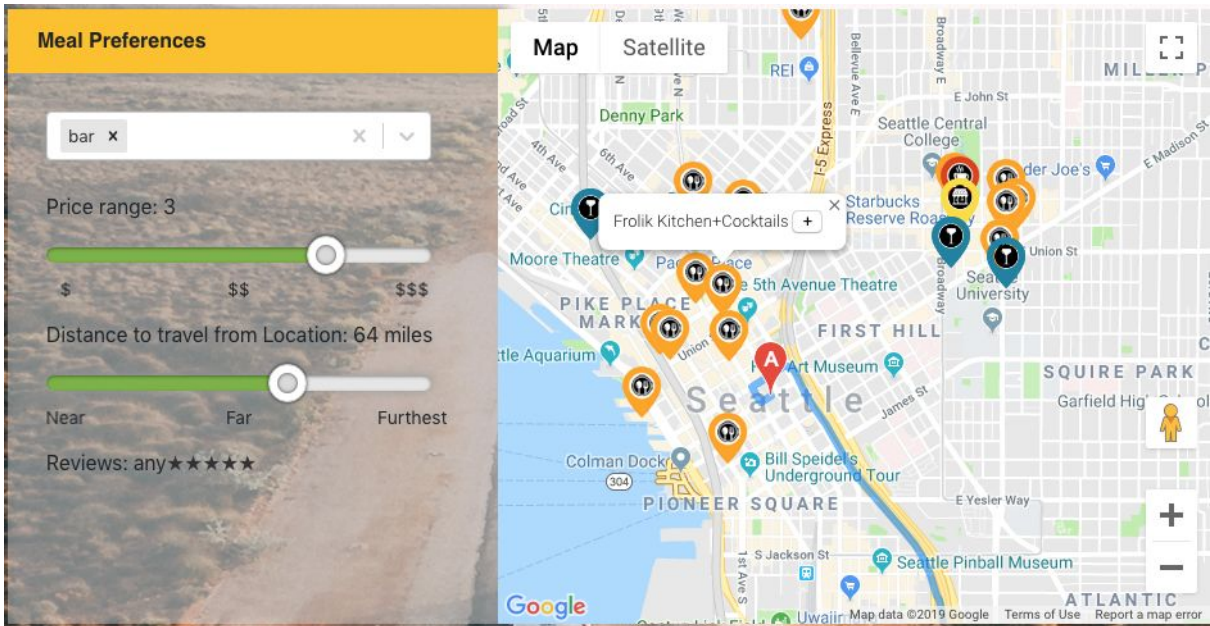


Figure 9 shows the ability to add a selected location to a route.

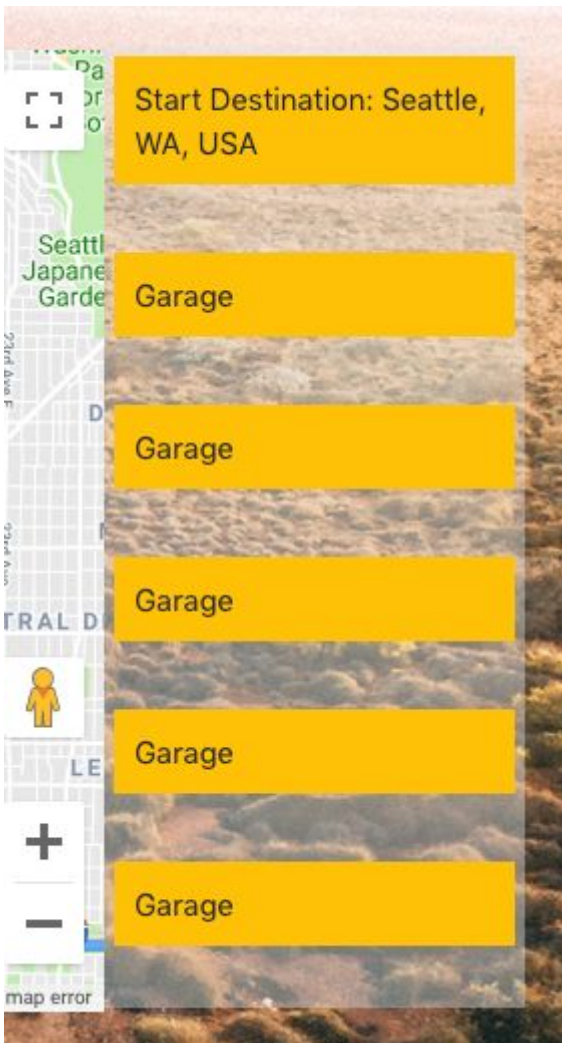


Figure 10 shows that after clicking on the button in the info window, the locations is added to the route.

3.1.2 Hardware Interfaces

The application shall function on any Internet-enabled computer with a Google Chrome, Firefox, Safari or Internet Explorer 11 browser.

3.1.3 Software Interfaces

The application shall make use of several software interfaces to interact with external software systems. It shall also store and query user preferences and path information in a mongoDB database hosted on Amazon web services RDS. Queries will be made using JavaScript and NodeJS. Additionally, the application will use Google Map API for its geolocalisation as well as TripAdvisor API for restaurants and hotels recommendation.

3.1.3.1 User Information

Any information given to the system by the user such as their login information, preferences while planning a specific trip or a trip itself will be saved onto a Database(--maybe mongo)

- Adding/updating login information
- Saving planned trips
- Saving user preferences in regards to meals, activity preferences etc

Outgoing data from the database consists of SELECT commands, such as user trying to retrieve one of their saved trips

3.1.4 Communications Interface

3.1.4.1 Communication between UI and User

All data and requests sent over to the database will be encrypted secured over HTTPS with SSL. The application shall communicate with the various database and software services via API function calls. Constant suggestion updates will be made to existing trips if shorter or better routes are available. User will also have the option of changing their meal preferences and seeing real time updates in their selected trips. Additionally, confirmation emails will be sent to the user once they have created a login. The website will also provide users with the option of resetting their forgotten password.

3.1.4.2 Communication to UI and API

APIs will be accessed by a web page using HTTPS with SSL.

3.2 Functional Requirements

The application shall have the following functional requirements:

Plan Trip Module:

- Create a trip
- Save trip
- Update trip info
- List Trips

Refer to 3.1.1. User Interface Figure 2. and Figure 4. to see the flow of this functional requirement.

Optimization Module:

- Find optimized roads based on user profile

Refer to 3.1.1. User Interface Figure 3. to see the user preference and Figure 4. to see the trip obtained based on the user profile.

Map Module

- Display paths on map

Refer to 3.1.1. User Interface Figure 5. to see where the map is displayed.

Database Module

- Add/Update/Delete user's preferences based on profile
- Store optimized paths

Authentication Module

- Log in
- Log out

Let's have a couple of user stories to capture a description a couple of features RTA will have.

User Story 1:

As a user, I want to plan a trip from Seattle, WA to Los Angeles, CA to know in details what path to follow.

In this case, the user will be given three different road alternative based on start and end destination and user preferences. The user will be able to pick one of the three options and save it to the profile associated to the login.

User Story 2:

As a user, I want to save the details of the trip to have access of the trip's details.

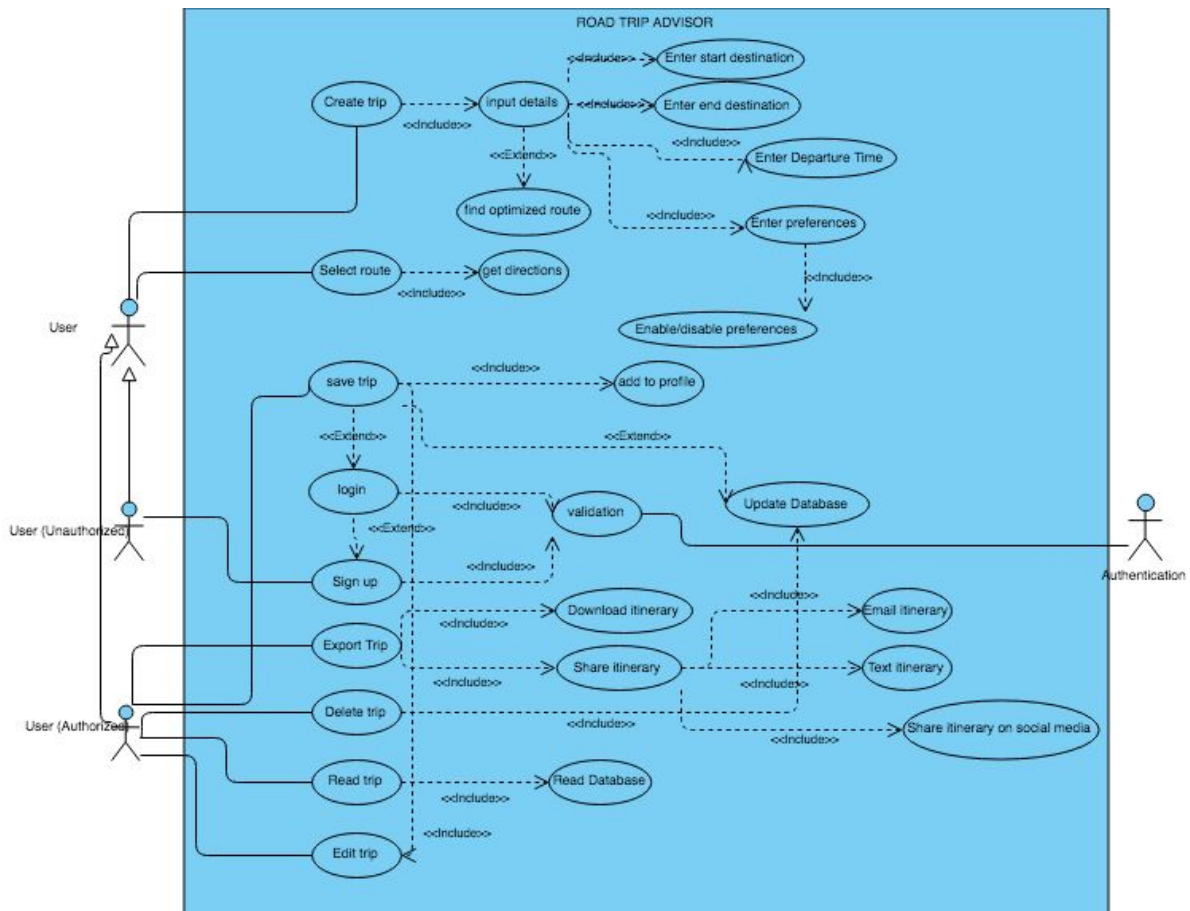
To do so, if the user can save the trip in the Profile section as long as the user has a login on RTA.

User Story 3:

As a user, I want to modify my preferences to indicate the number of people I am traveling with.

To do so, the user can go back to the profile section and adjust the number of people traveling for a specific trip and save the information.

The following UML displays the actors and the functionalities they have.



3.2.1 User Functionalities

3.2.1.1 User access to website

1. A user shall be able to access the Road Trip Advisor website from a device with internet access and be able to start planning a trip, login, or register.

3.2.1.2 Register a profile

1. Users have the option to register a profile with the Road Trip Advisor service.
2. Select “Register” from the landing page, and follow the prompts to fill out and complete the registration.
 - a. Information will be stored on the database so the website can allow customers to plan/save/edit/delete trips in the future.

3.2.1.3 Planning a trip

1. Users will be able to save a trip.
2. Users fill in required information, such as their source and destination, meal preferences, time preferences, number of people they are travelling with, age group their travelling with, purpose of the travel and what type of activities would they prefer on the trip.
3. Users can enter the source and destination(s) manually or by selecting points on a map.
4. Users can modify (change/delete/add) suggested trip options.

3.2.1.4 Login to profile (edit, plan, view saved trips)

1. From the homepage, users, once registered, can login to their profile to

- a) plan a trip
 - b) edit their profile details
 - c) view, edit, or cancel any saved trips.
- 1.

3.3 Performance Requirements

3.3.1 Real Time updates

The application is required to be fast and provide real time updates. After clicking on any button, the user shall receive a response within 10 seconds. The user shall be able to display the optimized paths within 15 seconds. The duration of the trip shall be based on the time user intends on taking the trip. If there are any sorts of delays such as closed intersections or accidents, user shall be notified.

3.3.2 Accuracy

The application should be accurate with the geolocation being used throughout the trip and provide authentic information about hotels and restaurants e.g. reviews and operating hours.

3.3.3 Compatibility

Once a trip has been planned, user has the ability of importing the route onto their preferred navigation application such as Google Maps or Maps.

3.3.4 Safety and Security

Any user can use the application without a login, so they will not need to prove their identity.

3.4 Logical Database Requirements

User and trip information shall be stored like the following table:

Data	Attributes
User	➤ Todo
Route	➤ Todo

3.5 Design Constraints

3.5.1 Standards Compliance

All language used in the software (including manuals and documentation) should comply with current Bellevue College guidelines for decency and equal opportunities.

3.5.2 Web Based Product

Compatible web browser must be installed on computer.

3.5.3 APIs

Our data will be limited to free publicly available APIs

3.6 Software System Attributes

3.6.1 Availability

GUI must be intuitive for users to use.

Database shall be available to user at all time.

3.6.2 Portability

Must work with all recent versions of major browsers (Chrome, IE, Firefox, and Safari)

3.6.3 Reliability

Monthly Uptime Percentage of at least 99.99%