ELEC3609

Internet Software Platform "Router"

System Requirements Analysis & Use Case Modelling

JooHwan Kwon

Vicky Lin

Bryn Lom

Dat Ngyuen

Nick Theng

Table of Contents

2. Introduction

This document is a system requirements analysis and use case modelling of a web application project development called “Router”

Our web app, “Router”, is for travelling, it takes a route for getting from one pl ace to another and specifies the attractions which are along the path. This can help make the destination of travel less important, hence allowing the journey to become more fun. However, not only providing the users with the attractions, it also supports the user by giving them room to store their favourite routes and attractions as well as to view history of their travels.

1.1 Purpose

Allows user to experience scenic/interesting travel in a simple web app, without too h eavily focusing on the destination.

1.2 Intended Audience

- Developers (Frontend + backend)
- Testers
- Project managers
- Executives who may be interested in this project

1.3 Intended Use

Programmers: Checking requirement
Managers: Help for defining deadlines and requirements
Testers: Will be able to check requirements for what they are testing against

1.4 Scope

In scope:

1. Overall description of the work
2. Deliverables
3. Justification for the project
4. Constraints.

5. Assumptions
6. Inclusions/Exclusions

We are creating a scenic travel app focusing on the journey and not the destination. With this we will provide personalised routes to users, including novelty stops and att ractions for the purpose of re-defining the traveling experience. This web app will wor k Worldwide, be available on mobile and desktop computers and take an efficient rou te to reach our destination. We are restricted by the third party maps service we use for accuracy of the service.

1.5 Definitions and Acronyms

Rest API: (Representational State Transfer Application program interface): An easy way to access information from another service using HTTP requests.

UML: Unified Model Language: It is a visual modeling language developed using dia grams to assist the developers to construct, visualize and document the software sys tem.

MVP: Minimum Viable Product: It is a product created through a development techni que that implements just enough features to create early deployment of the product.

1. Overall Description

2.1 User Needs

Simple user interface
Fast UX
Consistent routing
Recommendations consistent with user likes (Initially defined with account)

2.2 Assumptions and Dependencies

Dependencies:

- Google maps API (for location and direction data)
- AWS EC2

Assumptions:

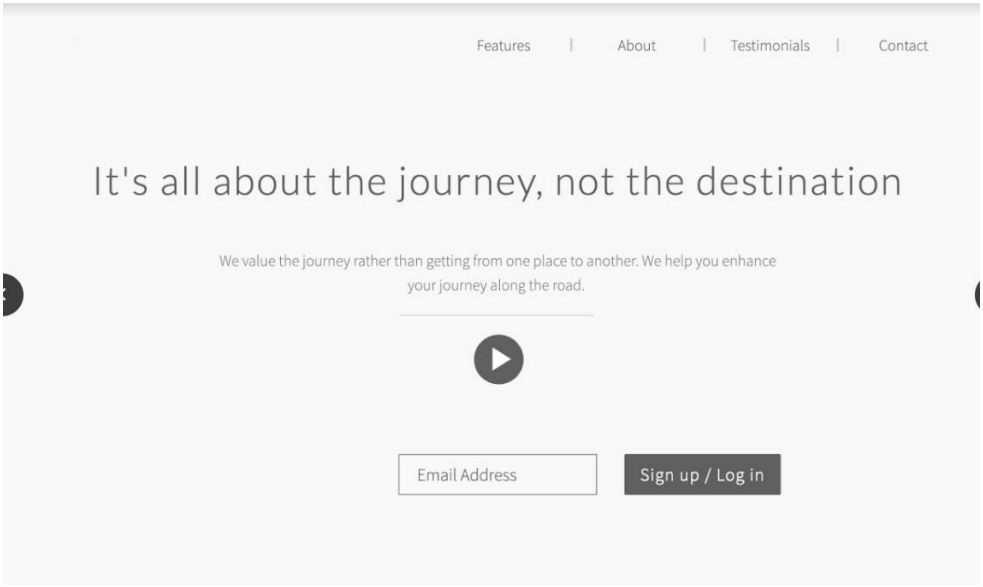
- Google maps API can function worldwide

2. System Features and Requirements

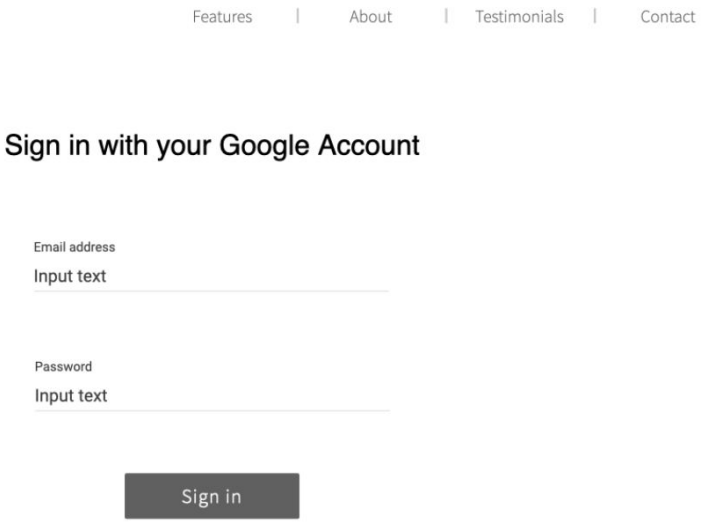
ID	Requirement Description	Source/requestor	Priority	Justification /need	Test Strategy	Notes
1	Can login to the system	Customer	High	Core functionality	Module testing	
2	Can edit travel preferences and save on a logged in user account	Customer	High	Core functionality	Module testing	
3	Can create travel route from one location to another	Customer	High	Core functionality	Module testing	
4	User will be given consistent recommendations on attractions to see during travel	Customer	High	Proof that the recommendation system works	Module testing	
5	Will be able to handle 10 concurrent users	Developers	High	Catering for QOL service for users	Module testing	
6	Output REST API will produce correct results 100% of the time	Developers	Medium	Specific for developers extending on our work	Module testing	
7	Service uptime of > 90%	Customer	High	QOL of the service	User acceptance testing	
8	User specific data is securely encrypted	Customer	Medium	Security of users data	Existence	Not much sensitive data stored ie. why medium security
9	Each code module having at least one line of dedicated comments	Developer	Medium	Quality of code base for extension and readability	Existence	
10	User can view history of previous trips when logged in	Customer	Medium	QOL	Module testing	
11	User can delete trip history data	Customer	Medium	Security	Module testing	

3. Website Wireframe

Each wireframe will have short descriptions below to provide further i nformation.

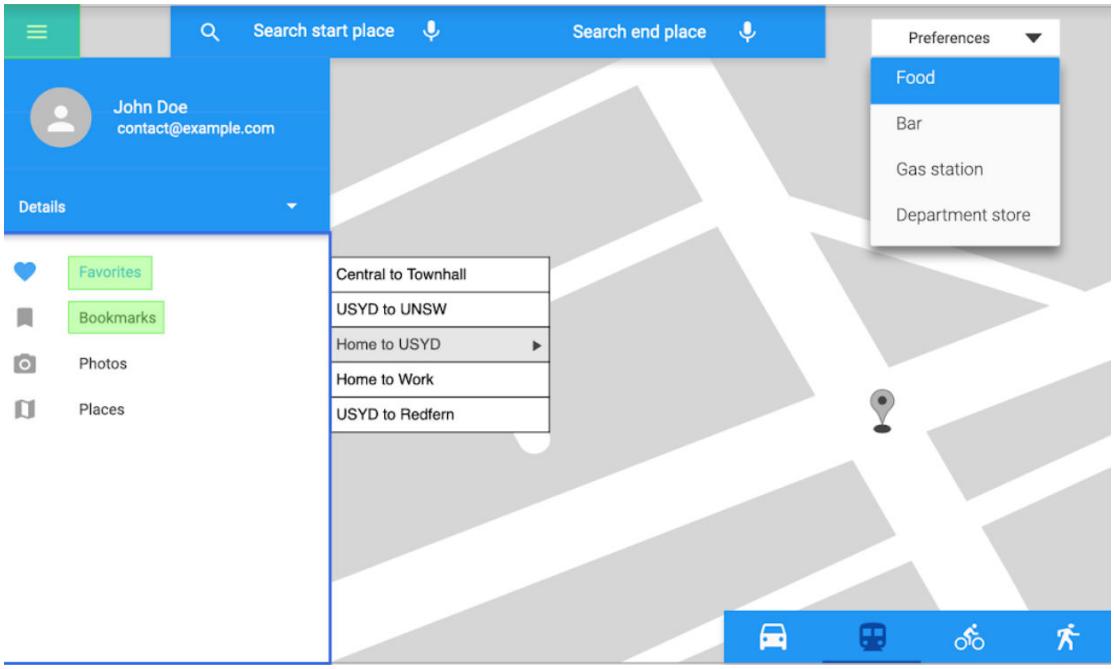


This screenshot shows the homepage when a user first visits the website without login. I t shows our description of the web-app and its feature to encourage users to sign up

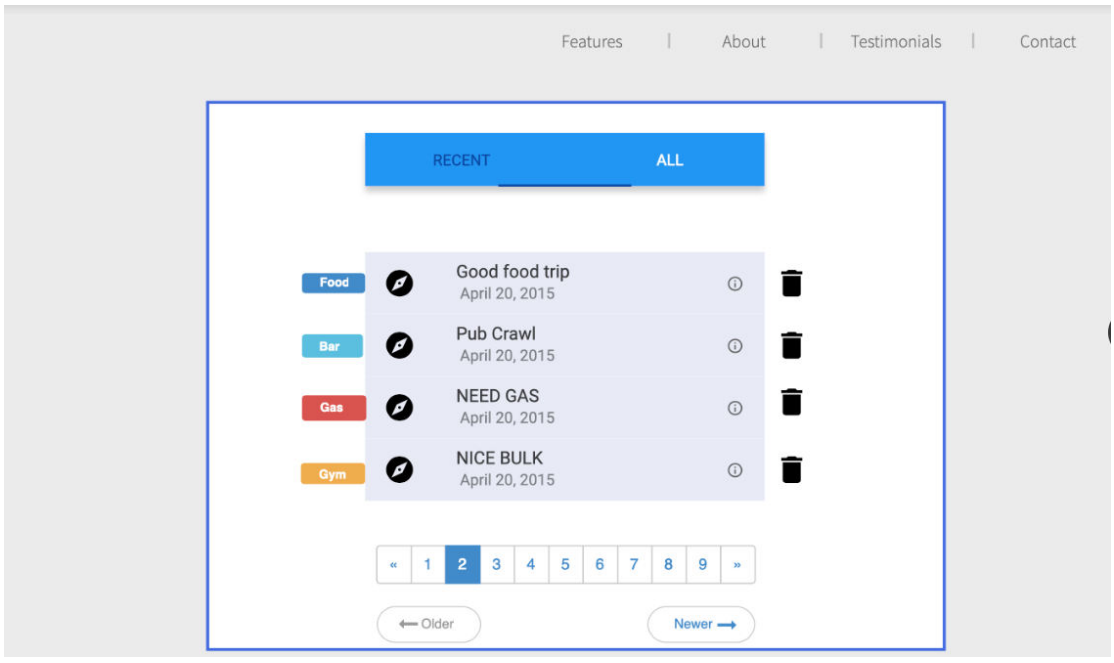


This screenshot shows the page where user first sign up/ sign in, they will be using their Google account to sign in. As it can be seen, all pages will have shortcut menus at the to

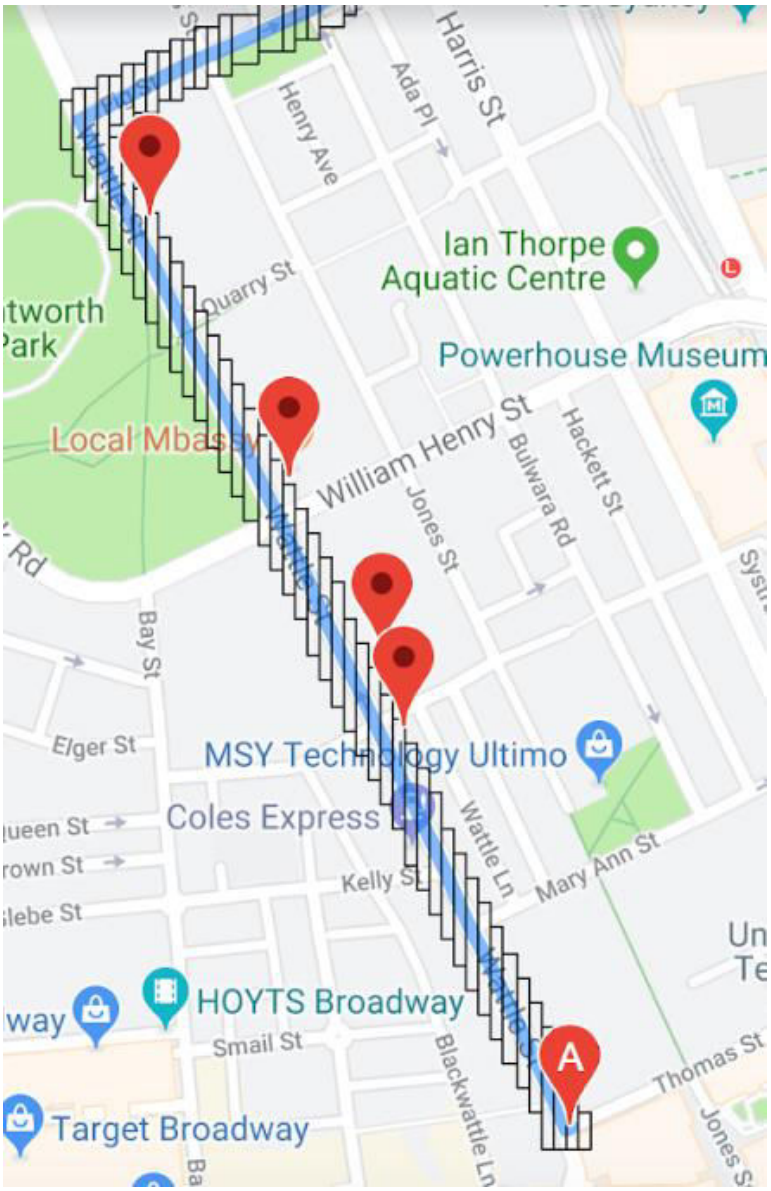
p for navigation purposes.



This is the main feature of the web-app. It shows Google map and search bar for place A and B. User can set preference for the attractions they want to see along the route an d the side bar can be used to navigate the users to their favourites and bookmarked rout es. The user preferences include type of attraction, departure and destination locations a nd type of transport they are using.



This wireframe shows that the users can delete or customize their bookmarked routes u sing tags to give the users more sense of ownership. Each route can be reviewed again by the users.



This is the result of a search along the route. It display marker for each places that user can click and it will link to place to Google. The boxes won't be shown to the users as thi s is in developers view. Each boxes represent boundaries for search limit for the set rout e.

4. Use Case Modelling

To see the flow of events for use cases (Unified Modeling Language), see Appendix 7.1

Here are the list of use cases covering normal cases, malicious (misuse) cases a nd invalid cases of “Router”

Case 1: Logs in

Case 2: Choose starting/destination/preference

Case 3: Change starting/destination/preference

Case 4: Check history

Case 5: Save route

Case 6: Change account details

Case 7: Delete account

Case 8: Check favourite

Case 9: Use favourite

Case 10: Logs out

Case 11: Delete favourite

Case 12 Delete history

Case 13 Register account

Case 14 (Malicious): Start and destination inputs are 1000 characters long

Case 15 (Invalid) Start and destinations are too far apart

*For further detailed description, see appendix 7.

5. Project plan

6.1 Team responsibilities and roles

Member	***Strength***	***Language***	***Preference***	***Role***
Dat Nguyen	Working with RESTful API and Socket communication Experience with web development	Golang Python C/C++ Javascript	Frontend	Designing and writing web services (RESTful API) QA testing (UI and UX)
Vicky Lin	Fast learner Good communicator	Java Python SQL	Backend	Writing AJAX and SQL statements Assisting
Nick Theng	Fast and flexible learner Flexible programmer Experience with web development	Java C/C++ Javascript Python	Frontend/ Backend	Designing UI, UX and general web functions Assisting server-side framework – python
Bryn Lom	Good communicator Good at writing documentation Ability to	Java Python SQL	Backend	Using Django framework to communicate with database Assisting writing AJAX and SQL

JooHwan Kwon	Good tracker of project Good at coordinating milestones of project	Java SQL HTML, CSS, Javascript	Frontend	Writing HTML, CSS, Java script using react.js

6.2 Milestones

To see the Gantt Chart of whole project with milestone and dependencies of the tasks, see appendix 7.3

List of milestones used to check project progress

Write SRS - Expected start date: 20/08/2019 Expected end date:21/08/2019

Write user stories - Expected start date: 25/08/2019 Expected end date:26/08/2019

Create wireframe - Expected start date: 29/08/2019 Expected end date:30/08/2019

Code frontend - Expected start date: 31/08/2019 Expected end date:14/09/2019

Code backend - Expected start date: 31/08/2019 Expected end date:14/09/2019

Create draft mock-up - Expected start date: 15/09/2019 Expected end date:21/09/2019

Initial prototype developed - Expected start date: 30/09/2019 Expected end date:11/10/2019

System testing - Expected start date: 15/10/2019 Expected end date:18/10/2019

Pre-launch review - Expected start date: 19/10/2019 Expected end date:21/10/2019

Site launch - Expected start date: 26/10/2019 Expected end date:27/10/2019

QA (Quality Assurance) testing - Expected start date: 29/10/2019 Expected end date:31/10/2019

Evaluation - Expected start date: 1/10/2019 Expected end date:3/10/2019

6.3 MVP features

Minimum viable product, “Router”, proposed in this documentation is a web application that e laborates a commonly used map service (Google map API) and user’s interests together to d isplay a result of existing places to visit along the route that matches user’s specific interests and preferences.

Router differentiates itself from the existing market products through implementation of simpl e features added onto existing products to further support user’s convenience and interest. Minimum viable product feature that it implements is a map that shows a route with places th at could be visited under classification of user’s interest.

Other features that further outstands Router is the login system for the user to utilise favourit es and history feature for future references. The users can come back to Router to revisit an y places or routes they have bookmarked or liked previously.

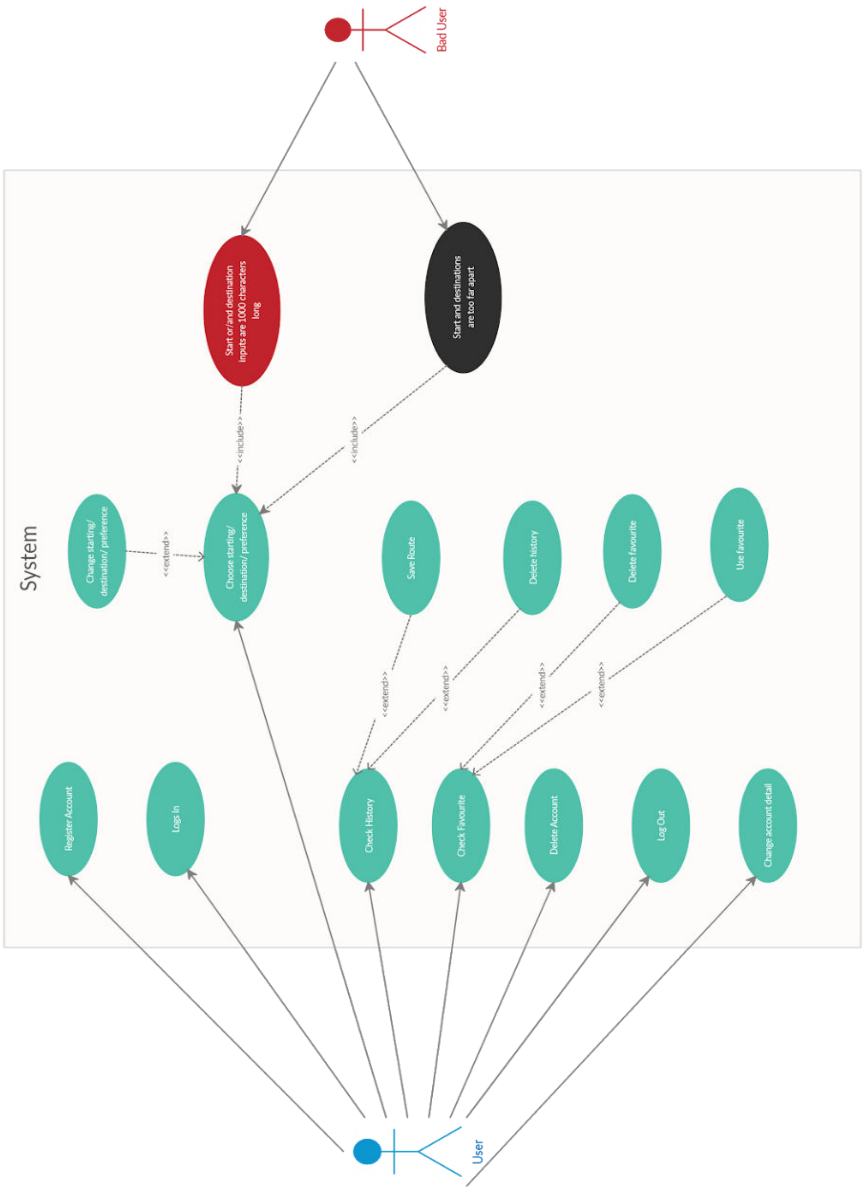
However, there are other features or functionalities that have been abandoned due to time, fi nancial and system constraints.

Given limited time of 11 weeks and as a team of amateur programmers with lack of experien ce and knowledge, there were some features brought up in the design phase but has been a bandoned due to the team capabilities. This included stronger registration process which the team assumed to consume an excessive amount of time.

Another constraint was the limited number of recommendations displays on the page or time taken to display these recommendations, especially for a long-distance travel. Due to the us e of Google API, there exists a limit set by Google to the number of calls of places at a time, which limits the displays of places. Hence, Router implements maximum limit on the distanc e that can be inputted by the user.

6. Appendix

7.1 Flow of events for use cases (combined)



6.2 Detailed Use cases

I. Normal Cases

UC1 : Logs in

1.1 Preconditions:

None

1.2 Main Flow:

When the users try to use the application, the application prompted a log-in screen for the user to e nter their username [E1] and their username's password [E2].

1.3 Subflows:

None

1.4 Alternative Flows:

Error 1: The user entered a username that the system cannot queried from the database, a messag e popped up on the page asking the user to revalidate their input.

Error 2: The user entered a username that the system can query but however the password mismat ched, a message popped up on the page asking the user to revalidate their password.

1.5 Actors

User, Router's database

UC2 : Choose starting/destination/preference

2.1 Preconditions:

User has logged in

2.2 Main Flow:

User has a view of a map and a few input forms for their choice of starting point, destination, and a field for their choices for type of places they would want to visit along the route. The user then input s in starting point [E1], destination [E2] [E3] [E4], and keywords / preferences [E5]. The app then st arts to route the path from source to destination and any places in close proximity to the path with t he desired keywords will be pinned across the map.

2.3 Subflows:

None

2.4 Alternative Flows:

Error 1: The user entered a starting location that cannot be found, a prompt telling the user of the er ror popped on the page and ask the user to re-input.

Error 2: The user entered a destination location that cannot be found, a prompt telling the user of th e error popped on the page and ask the user to re-input.

Error 3: The system found that the distance between the two locations the user has input exceeded the limit imposed by the system, a prompt telling the user of the error popped on the page and ask the user to re-input either the destination or the starting location.

Error 4: The system could not find a route between the two locations, the system then send a mess age telling the user of the problem on the page and ask the user to re-input their locations.

Error 5: The user has input in an empty string which is invalid, the system tells the user that they ha ve to have a preference to do the search

2.5 Actors

User, Router's database, Google API

UC3 : Change starting/destination/preference

3.1 Preconditions:

User has successfully performed a search

3.2 Main Flow:

The user will change the starting point [E1], destination [E2] [E3] [E4], or keywords / preferences [E 5]. The app then re-route the path from source to destination and any places in close proximity to th e path with the desired keywords will be pinned across the map, or the map keep the same route b ut pinned a new set of places.

3.3 Subflows:

None

3.4 Alternative Flows:

Error 1: The user entered a starting location that cannot be found, a prompt telling the user of the er ror popped on the page and ask the user to re-input.

Error 2: The user entered a destination location that cannot be found, a prompt telling the user of th e error popped on the page and ask the user to re-input.

Error 3: The system found that the distance between the two locations the user has input exceeded the limit imposed by the system, a prompt telling the user of the error popped on the page and ask the user to re-input either the destination or the starting location.

Error 4: The system could not find a route between the two locations, the system then send a mess age telling the user of the problem on the page and ask the user to re-input their locations.

Error 5: The user has input in an empty string which is invalid, the system tells the user that they ha ve to have a preference to do the search

3.5 Actors

User, Router's database, Google API

UC4 : Check history

4.1 Preconditions:

User has logged in

4.2 Main Flow:

User choose to view their route history, the system then displays information regarding the user's ro ute history, this includes the date of the search, search parameters such as starting location, destin

ation and search keyword.

4.3 Subflows:

None

4.4 Alternative Flows:

None

4.5 Actors

User, Router's database

UC5 : Save route

5.1 Preconditions:

User has made a search

5.2 Main Flow:

Once a search is successfully performed, the user can have the option of saving that search to their favourite. Once the favourite action is performed, the user will be able to access that route search anytime from their favourite tab.

5.3 Subflows:

None

5.4 Alternative Flows:

None

5.5 Actors

User, Router's database

UC6 : Change account details

6.1 Preconditions:

None

6.2 Main Flow:

The user change their password [E1] or username [E2] and input their current password [E3] [E4]. The system then updates their new details.

6.3 Subflows:

None

6.4 Alternative Flows:

Error 1: Password is an empty string or exceed character limit imposed by the database schema. T he user is notified of the problem and asked to input a new password.

Error 2: The username is an empty string or exceed character limit imposed by the database schem a. The user is notified of the problem and asked to input a new username.

Error 3: The current password Password is an empty string or exceed character limit imposed by th

e database schema. The user is notified of the problem and asked to input their current password.

Error 4: The password mismatched their current password. The user is notified of the problem and i s asked to re-input their password.

6.5 Actors

User, Router's database

UC7 : Delete account

7.1 Preconditions:

User has logged into the account.

7.2 Main Flow:

A user can choose to delete their account [S1]. The system will purge all of the information regardin g the users from the system database.

7.3 Subflows:

Subflow 1: Once the user chose to delete their account, the system will ask the user to input their u sername [E1], password [E2] and click on the confirmation button.

7.4 Alternative Flows:

Error 1: The user entered a username that the system cannot queried from the database, a messag e popped up on the page asking the user to revalidate their input.

Error 2: The user entered a username that the system can query but however the password mismat ched, a message popped up on the page asking the user to revalidate their password.

7.5 Actors

User, Router's database

UC8 : Check favourites

8.1 Preconditions:

User has logged into the account.

8.2 Main Flow:

User choose to view their favourite routes and the user can see their favoured routes from the past and informations regarding those routes which may include date, source-destination, search words.

8.3 Subflows:

None

8.4 Alternative Flows:

None

8.5 Actors

User, Router's database

UC9 : Use favourite

9.1 Preconditions:

User has at least one favorite route.

9.2 Main Flow:

User chooses a route from the favourite list, the system the reroute a path with found locations with the parameters saved in the favourite route.

9.3 Subflows:

None

9.4 Alternative Flows:

None

9.5 Actors

User, Router's database, Google API

UC10 User logs out of their account:

10.1 Preconditions:

User is logged in

10.2 Main Flow:

The user will open their options to find the option to log out their account, which will end the user's active session on their machine.

10.3 Subflows:

None

10.4 Alternative Flows:

None

10.5 Actors

User, Router's database

UC11 : User deletes favourite routes

11.1 Preconditions:

User has favourite routes and are accessing the list.

11.2 Main Flow:

While users are accessing their list of favourite routes, they may choose the delete option. This will allow all routes in the list to become selectable, which the user can therefore choose which ones to delete. These routes will disappear from the user's list.

11.3 Subflows:

None

11.4 Alternative Flows:

None

11.5 Actors

User, Router's database

UC12 : Delete history

12.1 Preconditions:

User has a non-empty history list and the user is accessing the list.

12.2 Main Flow:

User may choose to clear all routes which were automatically stored within the history list from prev ious searches. This will delete all previously uncleared routes from history.

12.3 Subflows:

None

12.4 Alternative Flows:

None

12.5 Actors

User, Router's database

UC13 : Register account

13.1 Preconditions:

None

13.2 Main Flow:

The user go to the registration site, input in their username [E1] [E2] password [E3] and click confir m. The user will then be able to log in to the website and use the application.

13.3 Subflows:

None

13.4 Alternative Flows:

Error 1: The user entered an empty string the username, this is not allowed by the system to so a p rompt informing the user is displayed on the page and the user cannot continue with registration unt il the problem is fixed.

Error 2: The user entered a username that already existed, the system prompts the user to change their username and does not allow registration.

Error 3: The user entered an empty string the password, this is not allowed by the system to so a pr ompt informing the user is displayed on the page and the user cannot continue with registration unti l the problem is fixed.

13.5 Actors

User, Router's database

II. Misuse Cases

UC14 : Start or/and destination inputs are 1000 characters long (Malicious)

14.1 Preconditions:

None

14.2 Main Flow:

When inputting the destination or the starting location, instead of inputting an appropriate string that the user desires to search for, the user try to overload the input fields with abnormally long string [E1]

14.3 Subflows:

None

14.4 Alternative Flows:

Error 1: The system restrict the input length to be less than 50 characters. The user cannot put in arbitrary long string which may or may not cause undefined behaviour.

14.5 Actors

User

III. Invalid Cases

UC15 : Start and destinations are too far apart (Invalid)

15.1 Preconditions:

None

15.2 Main Flow:

When inputting the destination or the starting location, the user put a valid starting location and a valid destination that are too far from one another [E1]

15.3 Subflows:

None

15.4 Alternative Flows:

Error 1: The system will notify the user that such a long distance is not supported by the application with a message on the web page and ask the user to input another set of valid destination/starting location.

15.5 Actors

User

User Travel Recommendation
ELEC3609 Project start dat 16/08/2019
August
September
October
November
Milestone Description
People assigned
Project Setup
Write project plan
JooHwan
Content research
Everyone
Write SRS
Bryn
Create UML
Dat
Write user stories
Vicky, Dat
UI design
Everyone
UX design
Everyone
Create wireframe
Nick
Coding Phase
Code frontend
Dat, Vicky, Bryn
Code backend
Nick, Joohwan
Create draft mock-up
Everyone
Content delivered/writing
Nick, Joohwan
Create site shells
Dat, Vicky
Initial prototype developed
Everyone
Implementation Phase
Check site compatibility
Vicky, Bryn
System testing
Everyone
Pre-launch review
Nick, Joohwan
Finalise site
Everyone
Site launch
Everyone
Maintenance Phase
Site testing
Everyone
Evaluation
Everyone
Updating system
Dat, Bryn
Implement user feedback
Vicky, Dat
Problem management
Joohwan
Service level management
Joohwan