



UNSW

SYDNEY

Software Engineering

DELIVERABLE 2

Paul (Hangi) Park | z5162101 |
Linchen Chen(Gary) | z5163479 |
Patrick | z5161379 |
Shangjie Zhou(Jay) | z5211225 |

Table of Contents

| | |
|--|----|
| Problem Statement..... | 2 |
| Data Source..... | 3 |
| Software Components | 3 |
| Relating Choices of Components..... | 5 |
| Choice of Technology or Framework | 7 |
| Details of the APIs..... | 8 |
| Justification of Choice of Components..... | 9 |
| Choice of Platform..... | 12 |
| Benefits | 12 |
| Requirement 1:..... | 13 |
| Requirement 2: | 16 |
| Requirement 3: | 18 |
| Requirement 4: | 21 |
| Requirement 5: | 28 |
| Requirement 6: | 30 |
| Requirement 7: | 34 |
| Requirement 8: | 37 |
| Conclusion (Team Organisation/Appraisal) | 39 |

PROBLEM AND SPECIFICATION

Problem Statement

For the duration of the course, the team decided to undertake a travel planner themed website. The team realized that other website that offer these services has a clustered website filled with information. So, the team decided to solve this issue by using a minimalistic design to offer a more user-friendly website that allows the user to freely and easily traverse our website. We also found out that most websites that offer travel destinations don't usually provide a timetable to schedule your trip. We as a team thought it was essential to provide a timetable as it will help the user to better manage their time during their stay in the relevant cities. Also, as our website can validate if the timetable is feasible through complex algorithms, it helps the user create a realistic schedule. Like other websites we still provide the essentials that a travel planner website needs such as a search engine, suggested destinations and events, detailed information of the users points of interest and filtering systems to filter through all the data until they can find a place of they're desire. As these are all provided by other websites we decided to not focus on these aspects during our design.

SOFTWARE ARCHITECTURE

Data Source

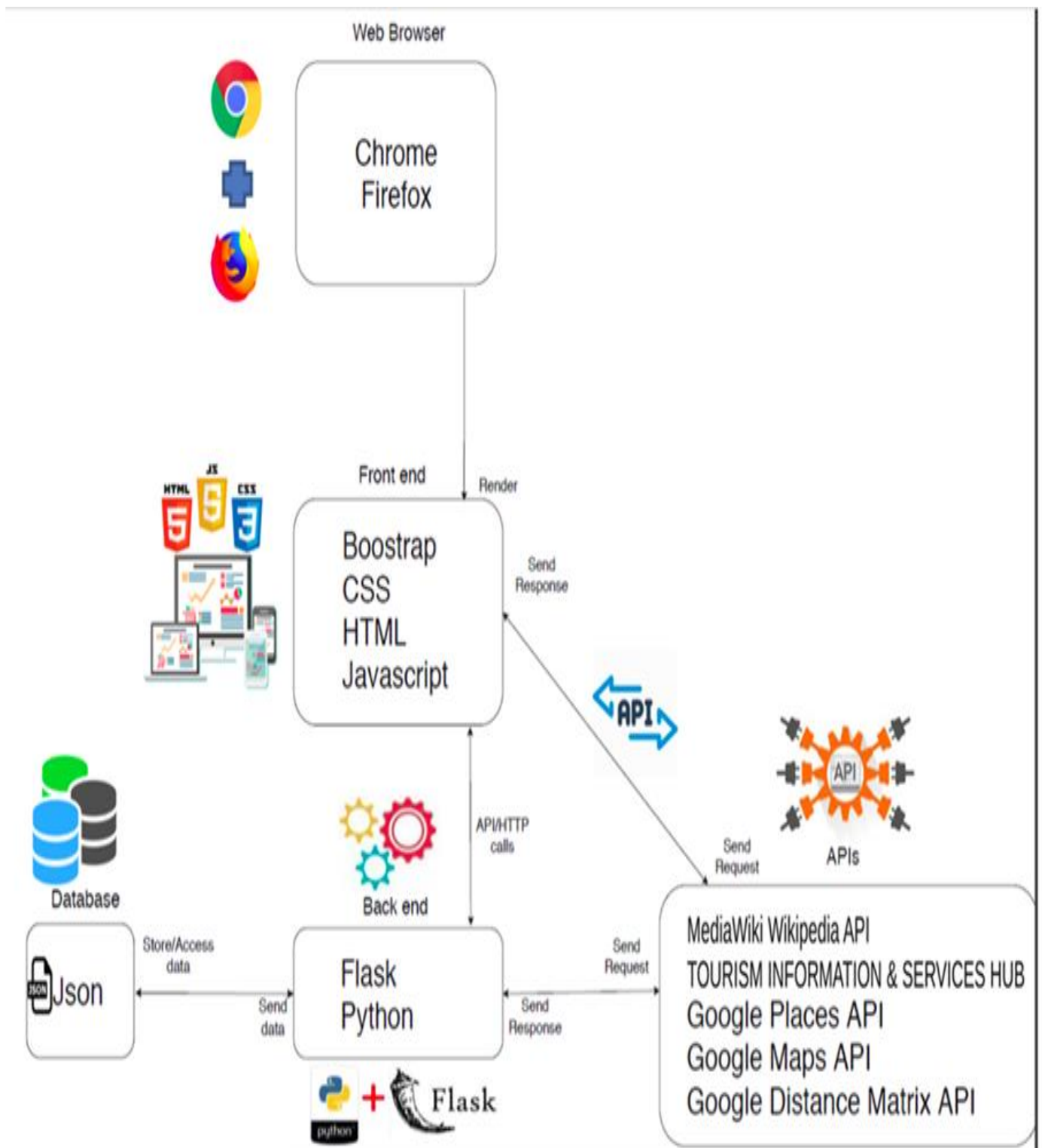
The data source that we will be utilizing will be aa from APIs:

- Google Maps API
- Tourism and Information Hub API (Singapore Government)
- Google Places API
- Google Distance Matrix API
- WikiMedia Wikipedia API

Software Components

The software components used in our project:

- Bootstrap
- CSS
- HTML
- Javascript
- Flask
- Python
- Google Maps API
- Google Places API
- Google Distance Matrix API
- WikiMedia Wikipedia API
- Tourism and Information Hub API (Singapore Government)
- Json(Database)
- Web-browser (Firefox, Chrome)



Relating Choices of Components

Front-end:

Web-browser:

To target a wide-range of customers we are deploying our website on most major web-browsers such as Chrome, Firefox and Internet Explorer.

Bootstrap:

Bootstrap is a giant collection of handy, reusable bits of code written in HTML, CSS, and JavaScript. It's also a front-end development framework that enables developers & designers to quickly build fully responsive websites. Thereby, we will be utilizing it to provide the building blocks of what our webpage design is going to look like.

CSS:

We design some of our own CSS files to make up our HTML page, to make our user interface succinct and attractive.

HTML:

We use HTML to create our web page for the project.

JavaScript:

JavaScript is an object-oriented computer programming language commonly used to create interactive effects within web browsers. Thereby we are utilizing it to offer a more dynamic website as it will provide additional functions that is not provided granted in Bootstrap.

Back-end:

Flask:

Flask is a web framework to support HTTP requests and responses for our front-end and back-end, it also dispatches code that generate HTML. Flask is a base python web framework which is convenient since we use Python as our back-end language and thereby allowing us to easily manage the back-end.

Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. We decided to utilize it as our back-end language for our websites so that it will be the backbone of the back-end implementation.

API/Database:

An API 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. So, we will be using the API to provide the data required for the website

Json:

JSON is short for JavaScript Object Notation and is a way to store information in an organized, easy-to-access manner. Json was chosen to build our own database for the website. It will store the user account information and all destination information that is retrieved from the APIs.

Choice of Technology or Framework

| | Advantage | Disadvantage |
|---|--|---|
| Python | <ul style="list-style-type: none"> • Easy to code and read • Free and Open-Source • High- Level Language • Large Standard Library • Object-Oriented • Extensible • Team already comfortable using python | <ul style="list-style-type: none"> • Speed Limitations • Weak in Mobile Computing and Browsers • Underdeveloped Database Access Layers |
| Flask (Web Framework) | <ul style="list-style-type: none"> • Simple request-response design • Simple templating system • Easy to learn • Flask is API-stable | <ul style="list-style-type: none"> • Flask is not as full-featured • Is difficult to deploy in production. • Loose design principles |
| Bootstrap (Front-End Architecture) | <ul style="list-style-type: none"> • Don't code from scratch • Uniform design throughout webpage • No more adjusting pixel width • Synergy is maintained across the website • All in one framework for HTML/CSS and JavaScript | <ul style="list-style-type: none"> • Restricts your design • Slower to do full updates because it doesn't have an in-memory DOM. |
| Json (Data Structuring) | <ul style="list-style-type: none"> • It supports data structures used in modern languages • JSON is Fast • Schema Support: meaning it has wide-range of browser compatibility • JSON is a great tool for the sharing data of any size even audio, video etc. | <ul style="list-style-type: none"> • Security Risk: lacks robust security from malicious software • JSON has no error handling for JSON calls |
| HTML/CSS (Front-End Framework) | <ul style="list-style-type: none"> • Easy to use • Members all know how to use it | |
| JavaScript | <ul style="list-style-type: none"> • JavaScript is fast • JavaScript is relatively simple to learn and implement • JavaScript plays nicely with other languages and can be used in a huge variety of applications. • Being client-side reduces the demand on the website server. • There are many ways to use JavaScript through Node.js servers. | <ul style="list-style-type: none"> • Because the code executes on the users' computer, in some cases it can be exploited for malicious purposes. • JavaScript is sometimes interpreted differently by different browsers. |

Details of the APIs

| API | API address | Description | Cost | Pros | Cons |
|---|---|---|----------------------------------|--|---|
| Google Places API | https://developers.google.com/places/web-service/intro | The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest. Details | Free (Up to 11,500 calls/month) | <ul style="list-style-type: none"> - Large amount of free calls - Includes large amounts of information on locations | <ul style="list-style-type: none"> - Costs money after a certain number of calls |
| Google Maps API | https://developers.google.com/maps/documentation/maps-static/intro | The Maps Static API lets you embed a Google Maps image on your web page without requiring JavaScript or any dynamic page loading. The Maps Static API service creates your map based on URL parameters sent through a standard HTTP request and returns the map as an image you can display on your web page. | Free (Up to 100,000 calls/month) | <ul style="list-style-type: none"> - Large amount of free calls - Shows a physical map that displays surrounding streets and locations | <ul style="list-style-type: none"> - Costs money after a certain number of calls |
| Google Distance Matrix API | https://developers.google.com/maps/documentation/distance-matrix/start | The Distance Matrix API is a service that provides travel distance and time for a matrix of origins and destinations, based on the recommended route between start and end points. | Free (Up to 40,000 calls/month) | <ul style="list-style-type: none"> - Large amount of free calls - Finds both time and distance, including delays such as traffic | <ul style="list-style-type: none"> - Costs money after a certain number of calls - Doesn't display directions to take |
| Tourism information and services hub (content user API) | https://tih-dev.stb.gov.sg/content-api/apis | The Tourism Information & Services Hub (TIH) is a digital resource platform for businesses to access relevant information on Singapore's tourism offerings and travel software services. | Free | <ul style="list-style-type: none"> - Free - Lots of useful Information - Reviews - Can search by keywords - Contains tags to help filter by | <ul style="list-style-type: none"> - Only contains information about Singapore |
| MediaWiki Wikipedia API | https://www.mediawiki.org/wiki/API:Main_page | The MediaWiki action API is a web service that allows access to some wiki-features like authentication, page operations, and search. It can provide Meta information about the wiki and the logged-in user. | Free | <ul style="list-style-type: none"> - Allows access to all of Wikipedia's library - Lots of information | <ul style="list-style-type: none"> - Requires a page to be on Wikipedia - Oftentimes too much unnecessary information |

Justification of Choice of Components

Back-End - Python

Python is an easy to learn programming language that the group has some knowledge of. Due to its large library of functions, it makes building a robust back-end simpler and easier. As python is used extensively in the work-force, we desired to emulate what it feels like to be in a business. Also, as it is free and open-sourced we do not have to worry about licensing problems down the future. Although it has its drawbacks compared to other languages such as Java, Ruby etc. due to its security problems, the simple but robust back-end that Python provides is the perfect tool needed for our start-up website. Also, as the team is very familiar in the uses of python compared to the other languages such as Ruby, C, C++ etc. we decided to utilize python.

Web Framework - Flask

We are using Flask as our web framework that handles HTTP requests and responses from our front-end and backend. As we are using python as our back-end language, flask due to its base python web framework, will work seamlessly along with it. Also, as it is API-stable it will help API's call be fluent and safe making it very desirable to use as our main source of database will be in APIs. We have other alternatives such as Django which is widely used in the industry however due to us being familiar with Flask we have decided to use it.

Front-End Framework - Bootstrap

To provide an interactive and visually appealing website, we have decided to use Bootstrap as the front-end web framework. To do this Bootstrap will provide the HTML and CSS design templates for interface components, it will also supply optional JavaScript library to provide an interactive user interface. As Bootstrap is widely used in industry and how quickly and easily one can use a template for their own website we have decided to use Bootstrap over other front-end framework such as Foundation and other lesser known front-end framework.

JavaScript

JavaScript is an object-oriented computer language that we are using to support the other frameworks we are using. Due to its simplicity, it will be easier on the members to learn whilst also adding additional functions that are not provided from Bootstrap. The versatility and interoperability that JavaScript has also allows us to seamlessly add it to all the other frameworks we are using. Due to all these features which incorporates well with the other front-end framework we have decided to use JavaScript over the CoffeeScript, Dart, TypeScript which cannot provide all the functions together that we mentioned that JavaScript could.

Additional Front-End Framework - w3

W3.CSS is another front-end framework incorporated into our website to help design our website faster, beautiful and responsive. While the JavaScript will add the dynamic aspect of the website, the W3.CSS will be the visual and functions of the website.

Data Structuring - Json

Json will be our main form of back-end database to store new data received from APIs and sort them. As we found out that querying and requesting in APIs can be slow, we desired a faster way to retrieve the same data. Thereby using Json to store previous requests and when the same request is called later we can just send the stored data of the same request, improving the overall time and fluidity of the webpage. Also, as we are storing different arrays of data (maps, customer review etc) Json has great compatibility with our needs. We have decided not to use a formal database system such as SQL or PostgreSQL as our mentor told us that it will provide too much workload to learn and code it up. We have decided not to use other Data Serialization Comparisons as they are less used in industry while also co-operating less with the JavaScript front-end framework we used.

Google Maps and Places API

The reason why we utilized these APIs was because it was free. It also provided a wide range of data about the POIs in which we can utilize to provide the information to the detailed webpage and search webpage. It also provided customer reviews in which we needed to provide the user a more objective understanding of the destination. Lastly it provided images about the POIs which was useful to show the user what the place they are going to look like.

Google Distance Matrix API

This API provided the distance and time it takes from destination A to destination B. This was very useful as it will help provide the important feature of validating the timetable that the user created.

MediaWiki API

Provided a more In-depth explanation about the destinations the user is interested in

Tourism information and services hub (content user API)

This provided all the data we needed about Singapore. This was also a very secure API as it was provided by the Singaporean government so no risk utilizing it. Also it was free unlike the Australian tourism information and service hub, making it easier to access and use. Lastly they provided an in-depth description in how to use its API making it very simple and easy to use. This was one of the main APIs we used, it provided the detail of the POIs, provided the events that coming up, provided the pictures and also the reviews.

Choice of Platform

The platform which the project can be supported by any computers that run on Linux and Windows.

And the supported web-browser are:

- Chrome
- Firefox
- Others wide-range use web-browser

The reason why we decided to allow our website to run on these web-browsers was because we wanted to get wide enough coverage so that a lot of people can benefit from our website.

Benefits

The aim of our project was to create a website that allows the users to plan where to and what to do. To do this our team has utilized various software components and technology to achieve this aim. Firstly, using Bootstrap implemented with CSS3, we were easily able to create a simple, coherent but elegant user interface at tremendous speeds. Also, using JavaScript with its additional functions that are not in CSS3 we were able to create a dynamic whilst also smooth user experience. Due to the use of JavaScript and python, we were able to create a dynamic timetable that can validate if the schedule is plausible or not. This allows the user to be able to better plan and manage their trips so that it will be a less stressful trip. This was possible as python allowed us to create complex algorithms to measure the from destination A to destination B with the help of Google Distance Matrix API. The JavaScript allowed us to understand the position of each events in the timetable so that we can measure the time it takes from each event, whilst also allowing us to seeing if the opening hours matches when the user wants to go there, etc.

Secondly, creating our own database system with Json allows the webpage queries and calls to be quickly responded to, allowing a fluent website. It also allowed us to store the user information such as their email, name and timetable and enabled a way to design a more individualistic website where we can focus on each user's preference. Lastly utilizing python to handle the back-end with its vast library function, helped create a simple but robust framework for the webpage.

INITIAL SOFTWARE DESIGN

Requirement 1: There needs a simple way to search for places by categories.

Feature1: Search for places within a city.

As a: user

So that: I can seek further information about different places in a city.

I want to: search for places that are in the city.

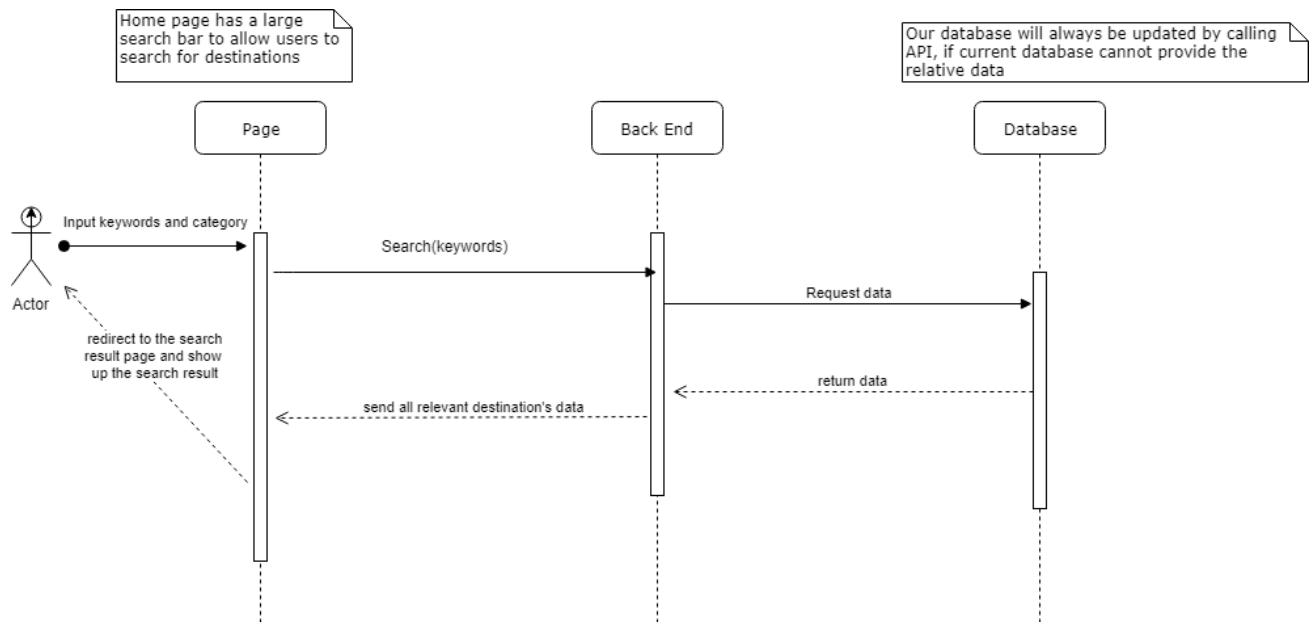
Scenario: the user can search for destinations in that city by typing in relevant key words.

Given: I am in the search page

When: I type in some key words in the search bar

And: I press “Search”

Then: I should see destinations related to the key words in the city I previously selected



1.1 Search for places within a city.

Feature2: Search for places sorted by accommodation, tourist attractions and restaurants.

As a: user

So that: I can seek further information about the destinations I have in mind.

I want to: search for places sorted by accommodation, tourist attractions and restaurants that match the type of destination I have in mind.

Scenario: the user can further search for destinations sorted by restaurants, accommodation or tourist attractions.

Given: I am in the search page

When: I click on “Category”

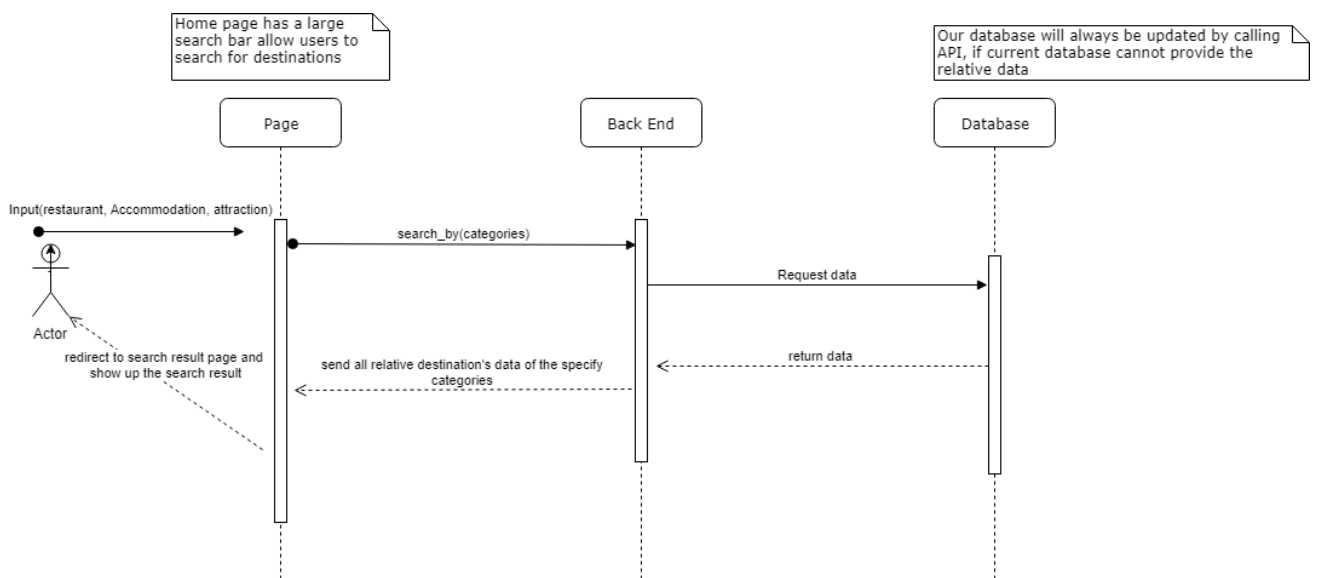
Then: I should see 3 selections: “Restaurants”, “Tourist attractions”, and “Accommodations”

When: I click on one of the selections

And: I type some key words in the search bar

And: I press “Search”

Then: I should see results related to the key words and my selection



1.2 search destination by categories on home page

Feature3: To be able to further categorize the search using filters

As a: user

So that: I can narrow down the search results according to my interest, in order to seek information about the destinations more efficiently.

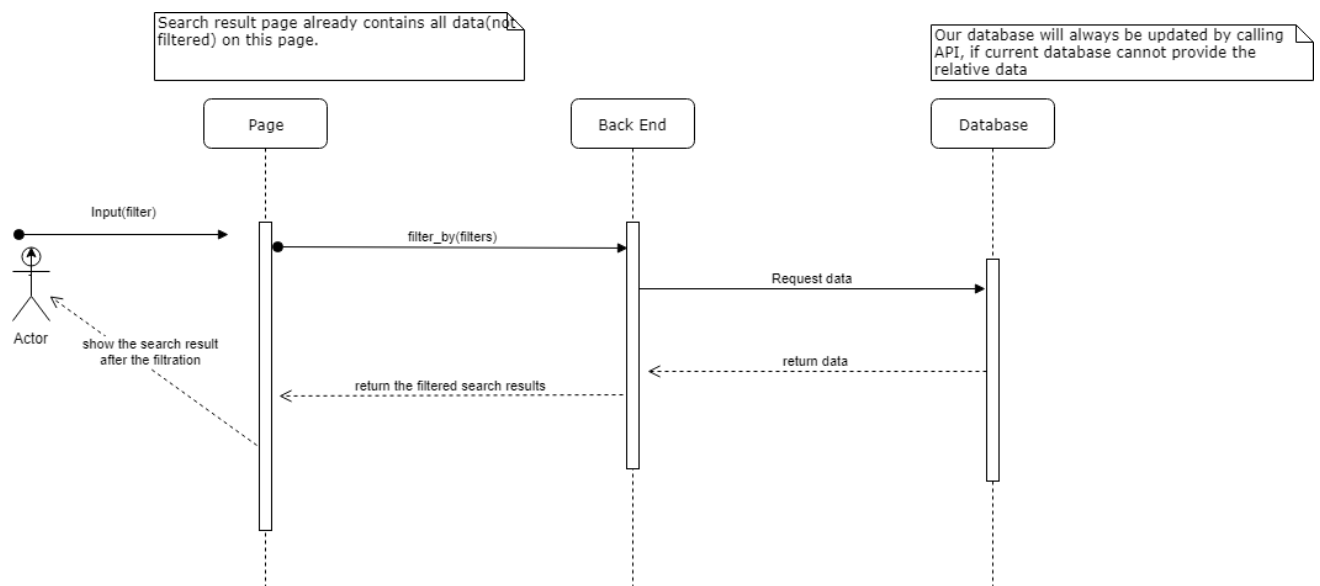
I want to: search by further categories after the search produced some results

Scenario: Within the search result page there are filters on the left-hand side in which you can further specify what you are looking for

Given: I am in the search result page

When: I have selected a filter(e.g. low cost)

Then: It will further sort the destination according to the filter.



1.3 filter the search results on search result page

Requirement 2: present information of each search result in the search result page in 2 different levels of specificity.

Feature1: present only the very-essential (e.g. name, picture etc.) information for each search result.

As a: user

So that: I can decide whether to look further into the search results.

I want to: see the basic information about each search result

Scenario: After search result completes, small containers of basic information about each individual destination is presented.

Given: I am in the search page

When: I press the search button with a city and “Restaurant” selected

Then: It should load a new page with all the relevant search results

And: Each result should contain the very basic information for the restaurant, such as its name, cuisine, average rating and cost.

Feature2: present further succinct information for each search result with a hover box.

As a: user

So that: I can quickly compare multiple destinations without the need to load each of them.

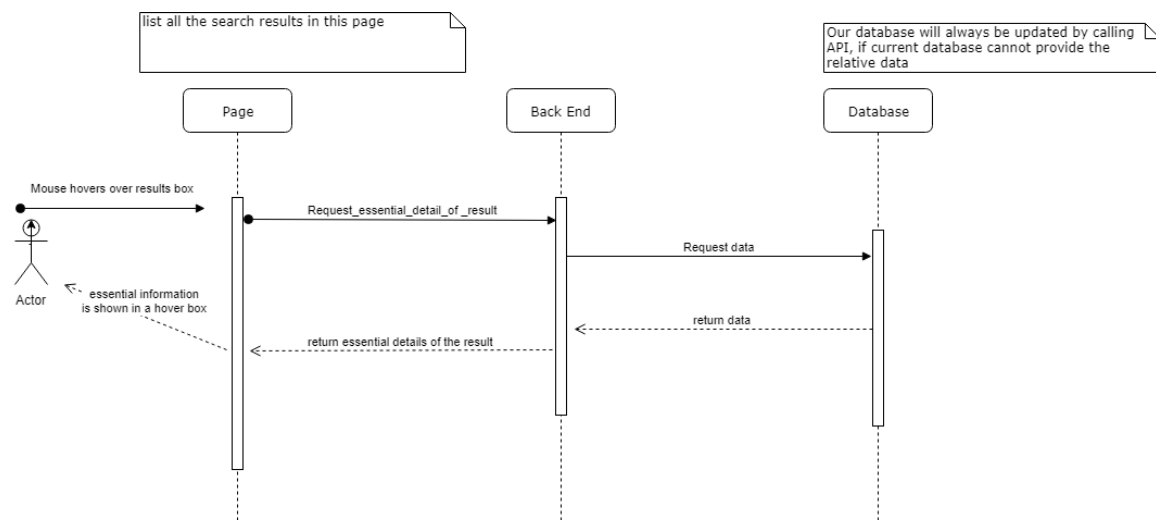
I want to: see more but not full relevant information about the destinations I am interested in.

Scenario: when I move my mouse to a search result, I can see more information about it in a hover box.

Given: I am in the search result page of “restaurants in the selected city”

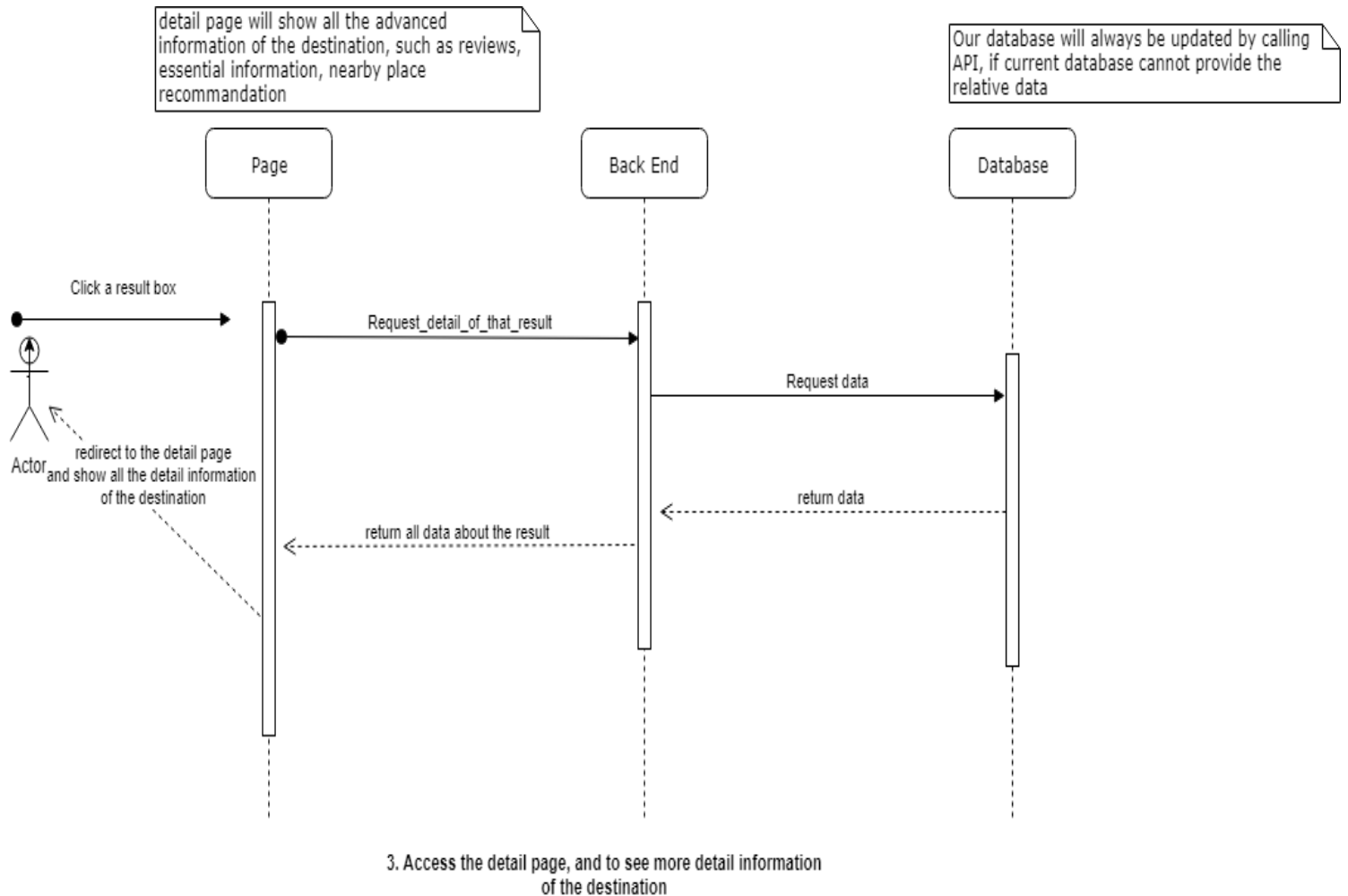
When: I move my mouse to one of the results.

Then: I should see a slightly more detailed description container of this destination, showing further information such as a preview of customers’ reviews, location of the restaurant, opening hours, etc.



2.2. show essential information in a hover box on search result page

Requirement 3: present in-depth descriptions for destinations of all kind. (if available)



Feature1: present detailed description after each search result.

As a: user

So that: I can obtain more detailed information about my destination of choice.

I want to: be provided with more detailed information about the destination and further personal detail about the place.

Scenario: After one clicks on the short succinct information for further information. They are loaded into a webpage filled with further description about the place.

Given: I am in the “search results” page

When: I click on one of the search results

Then: A new page comes out with further description about the destination

Feature2: present customer reviews in the detailed destination page under the description

As a: user

So that: I can obtain more genuine information about the place, by comparing the customers’ review to the its official description on the website.

I want to: be provided with fellow customers’ opinion about the destination I have in mind

Scenario: When I scroll down from the description, I can see the customer review.

Given: I am in the “detailed destination” page

When: I scroll down from the description area

Then: I see the customer reviews

Feature3: present the location of the destination in a map with its personal details

As a: user

So that: I can plan my way to get there

I want to: see the location and personal detail of the destination

Scenario: There is a map with the destination's address, phone number and website next to the description.

Given: I am in the "detailed destination" page

Then: I can see a map with the destination's personal details

Feature4: present all nearby places to the destination, grouped by accommodation, attraction and restaurant

As a: user

So that: I can plan my trip after visiting the current destination

I want to: see all the nearby places to the destination

Scenario: below the customers' reviews, there shows all nearby places to the destination, grouped by accommodation, attraction and restaurant

Given: I am in the "detailed destination" page

When: I scroll down below customers' reviews

Then: I can see all nearby places to the destination.

Requirement 4: To provide a function that allows the user to plan their trip

Feature1: To have an interactive, personal timetable in the profile page.

As a: user

So that: So that I can easily view my timetable

I want to: see the interactive, personal timetable

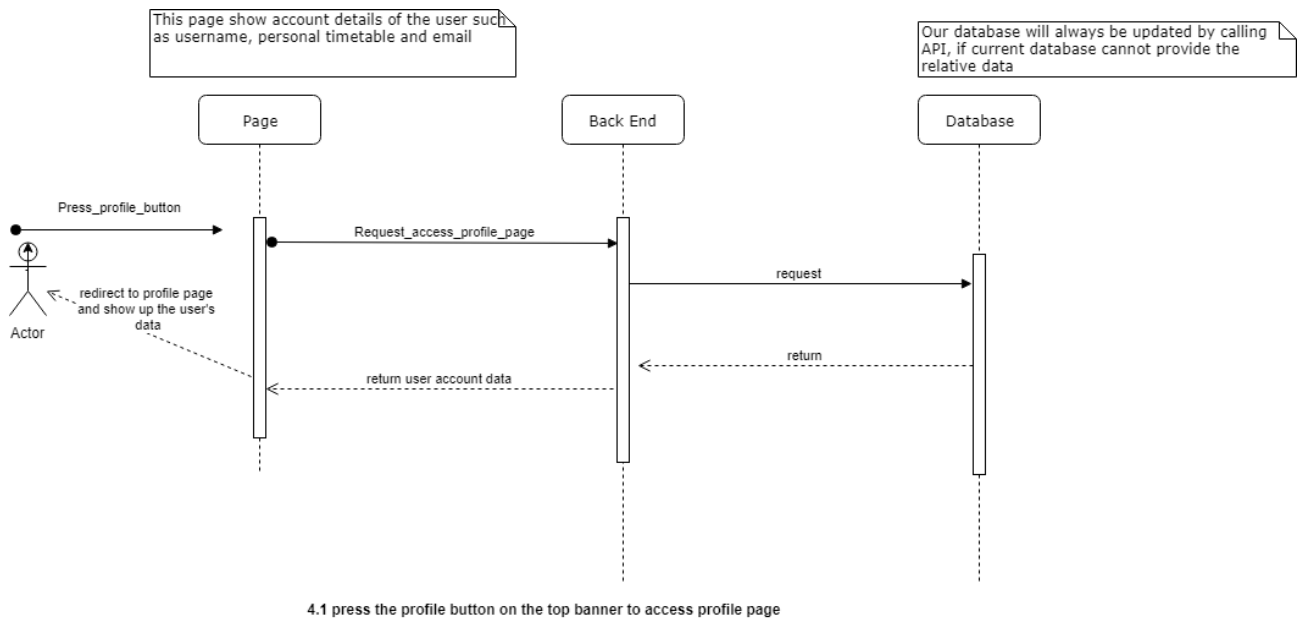
Scenario: After they click on the profile pick, they will see their details and the timetable underneath it.

Given: I am in the search page

When: I press the profile button

Then: It should load the profile page

And: The page will have their username, email and underneath that information will be the timetable



Feature2: Be able to add the destination into a bank where you can later use to timetable your trip.

As a: user

So that: easily access the destination I have in mind onto the timetable

I want to: see the destination I saved within a bank

Scenario: After they click on the profile pick, they will see on the right-hand side the bank of destination they saved from the detailed destination webpage

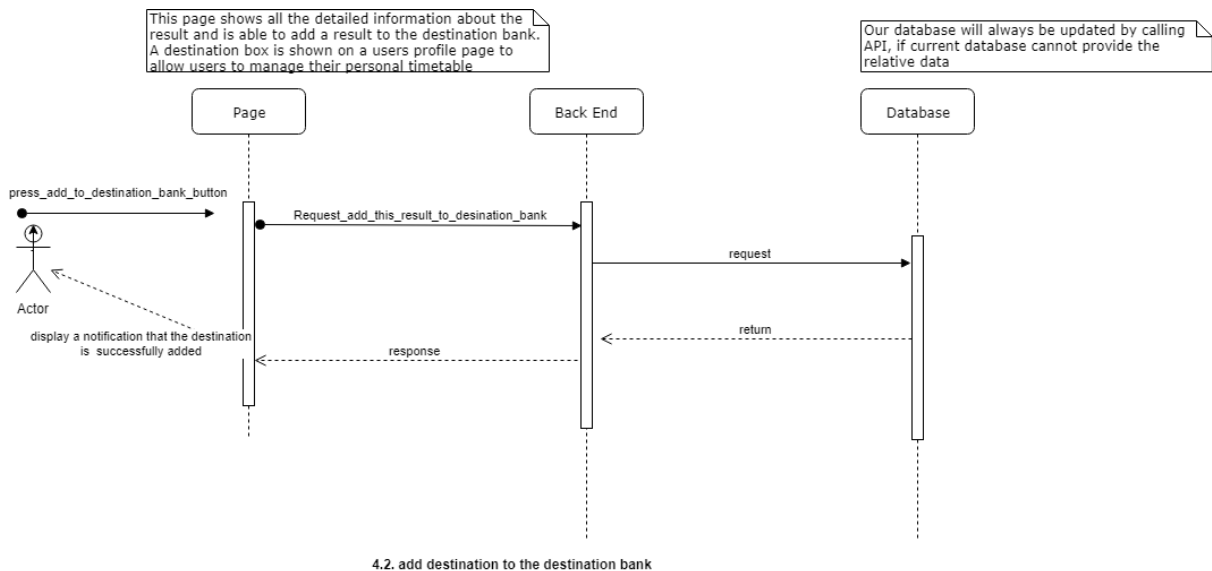
Given: I am in the detailed destination page

When: I press the 'Add to timetable'

Then: It should save in the bank

When: I enter the Profile page

And: on the right-hand side there should be a new "destination box" that is stored in the bank.



Feature3: To be able to add a destination from the bank.

As a: user

So that: to schedule my trip for the week.

I want to: add a destination to any timeslot in the timetable.

Scenario: When a user enters the profile page, he can move the destination boxes from the bank into the any timeslots in the timetable.

Given: I am in the profile page

When: I drag a destination box from the bank into one of the timeslots in the timetable

Then: That timeslot will be filled with the destination box

And: No other destination boxes can be placed into that timeslot

And: The added destination box will still stay in the bank

And: That destination can be still be added to some other empty timeslots.

Feature4: To be able to delete a destination in its timeslot.

As a: user

So that: I can change my schedule when I no longer want to visit the destination at that time

I want to: delete a destination from its timeslot.

Scenario: When a user no longer wants a destination to be in the bank, he can delete it in the bank.

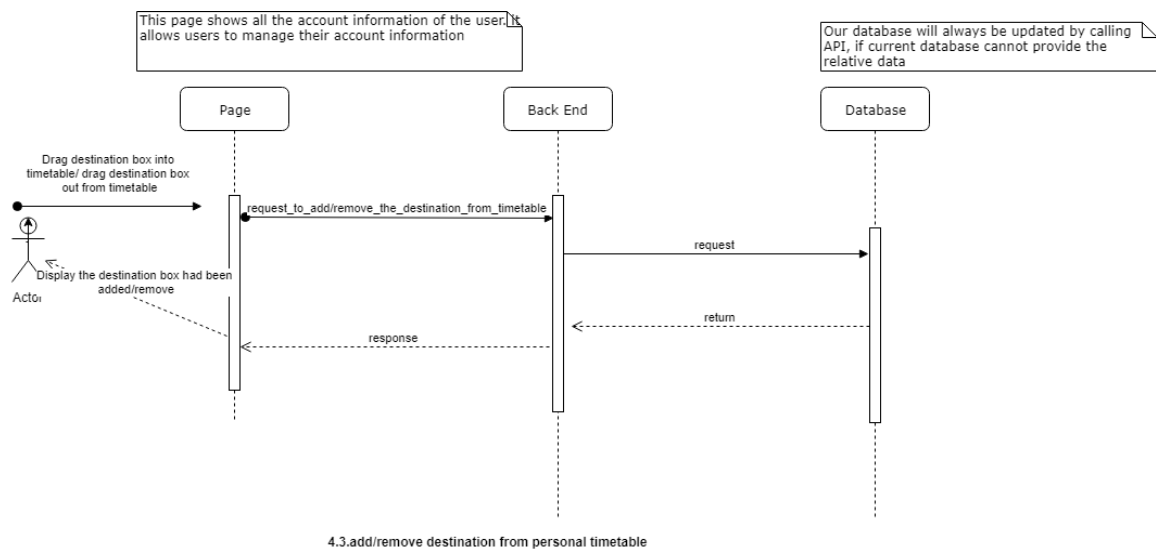
Given: I am in the profile page

Given: A timeslot is filled with destination "A"

When: I double click on the timeslot filled with destination "A"

Then: That destination will be deleted from the timeslot

And: The emptied timeslot can now be filled with another new destination.



Features: To be able to remove the destination from the bank.

As a: user

So that: I can remove destinations from the plan if I change my mind.

I want to: remove the destination box from the bank.

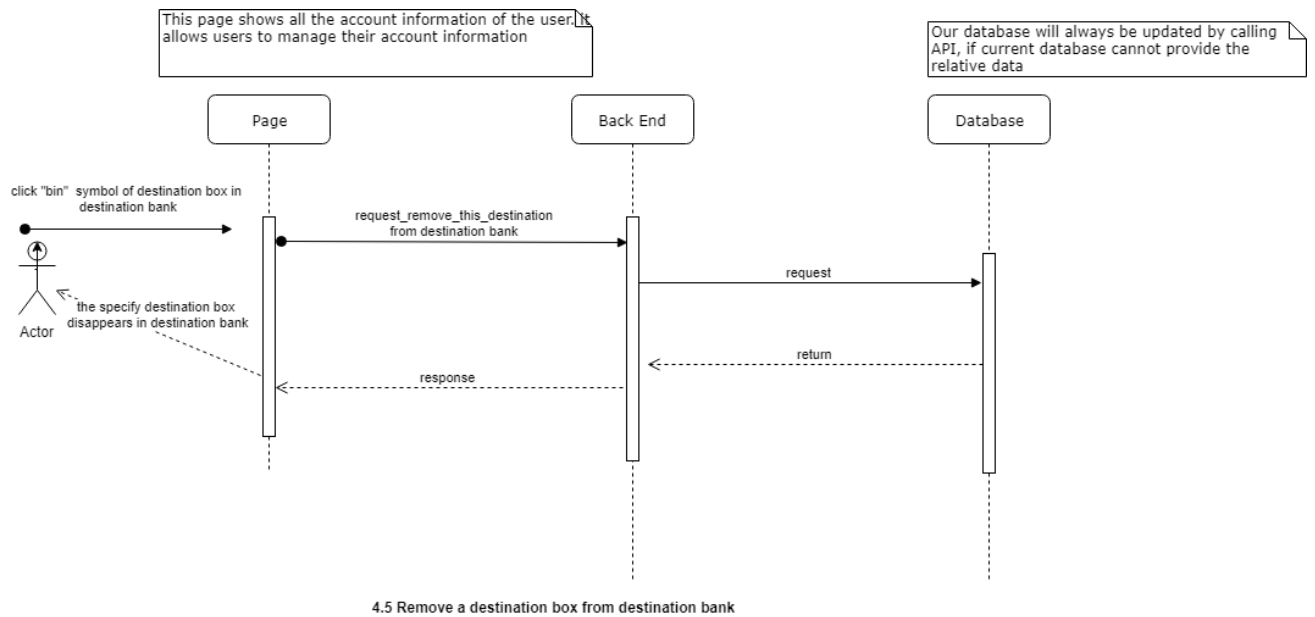
Scenario: When they enter the profile page, they can remove the destination boxes within the bank with a button.

Given: I am in the profile page

When: I press the 'bin' button on the destination boxes in the bank

Then: the destination box is removed from the bank

And: I can re-add the destination box from the detailed destination page.



Feature6: To generate the plausibility of the timetable

As a: user

So that: everything I planned will go smoothly

I want to: see if what I planned in the timetable is possible

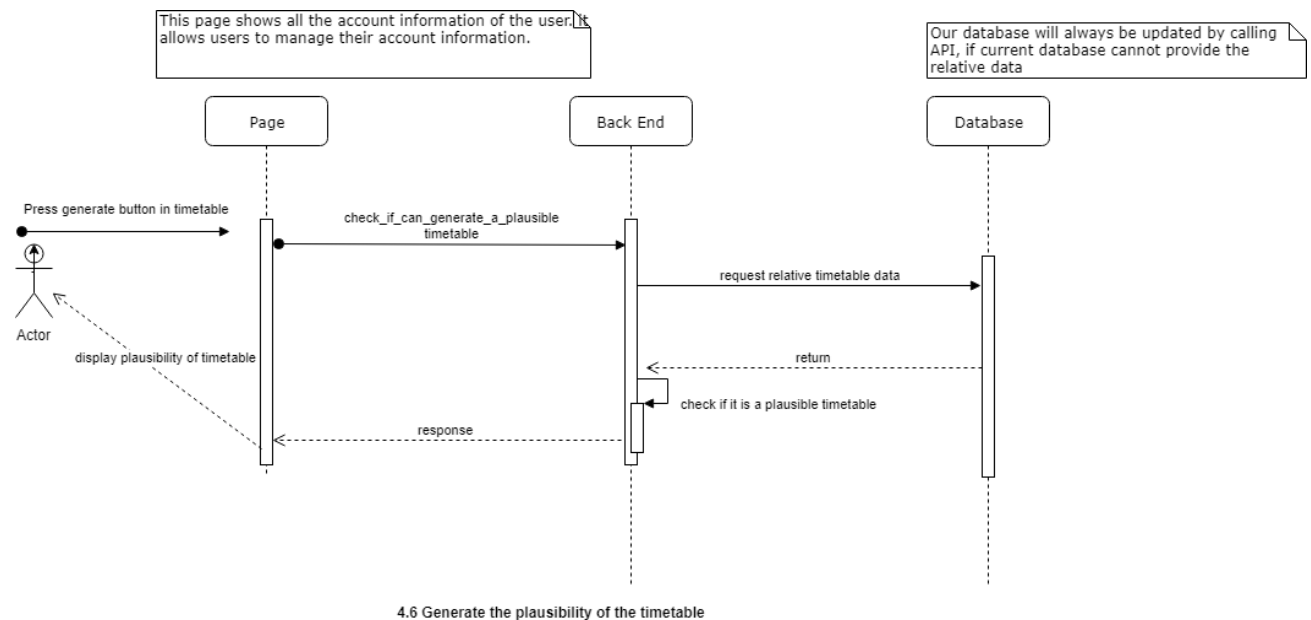
Scenario: When they press the 'generate' button, it will demonstrate by turning the bar of the timetable green if it is possible, otherwise turn red to show it is not possible.

Given: I am in the profile page

When: I press the 'generate' button at the bottom of the timetable

Then: the timetable bar will turn green/red

And: tell the user the timetable they created is possible/impossible.



Feature7: To be able to pick a specific week for the timetable

As a: user

So that: the timetable will be more realistic as it matches the date of my trip

I want to: be able to select a week with specific dates for my timetable

Scenario: When the user selects a date from the calendar and press “Get timetable” button, I will see a weekly timetable for the week that contains the date I selected.

Given: I am in the profile page

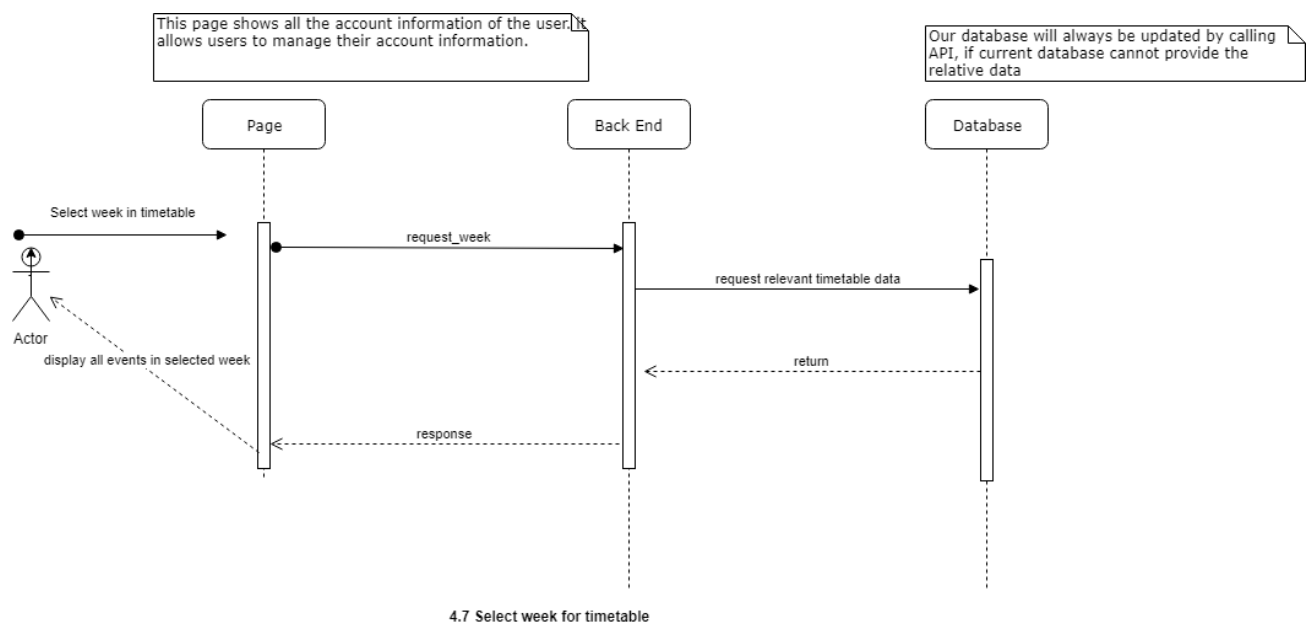
When: I click on the date selection bar

Then: I should see a dynamic calendar of the current month

When: I click on a date in the calendar

And: I click on the “Get timetable” button

Then: I should see a weekly timetable for the week that contains the date I selected, which has specific dates displayed under each day of the week.



Requirement 5: To be able to login your profile page

Feature1: To be able to sign-up

As a: user

So that: I can save the results and timetable.

I want to: be able to sign-up onto the website

Scenario: In the home/search page I can sign-up through the button of the profile at the banner

Given: I am in the search/home page

When: I press profile button

Then: It should load a pop-up page,

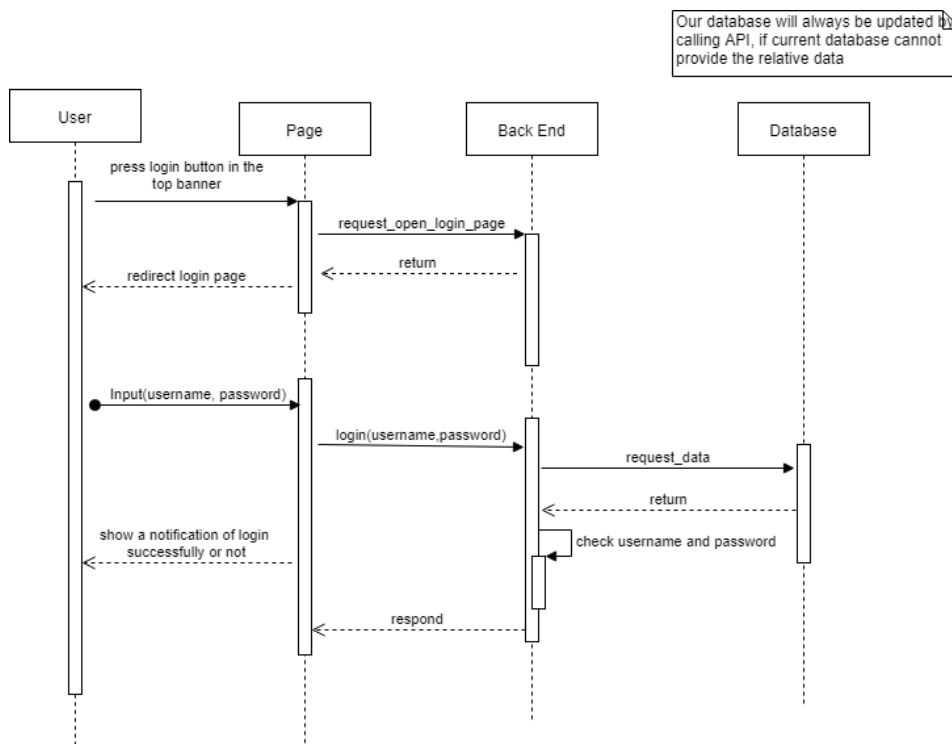
When: I press the “sign-up” button

Then: It loads the information needed to sign-up

When: I fill out all the information needed for the sign-up

Then: Press the sign-up button

And: I am logged in the websit



5.1. access login page and login

Feature2: To be able to login

As a: user

So that: I can re-load the saved timetable and research.

I want to: be able to sign-in onto the website

Scenario: In the home/search page I can sign-in through the button of the profile at the banner

Given: I am in the search/home page

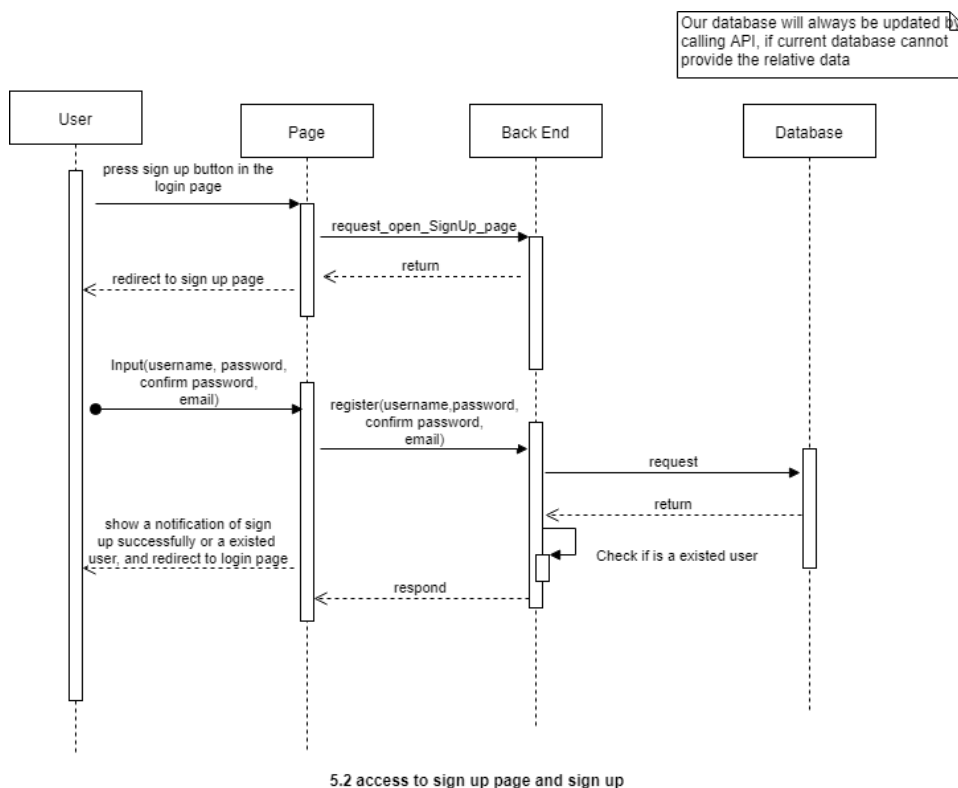
When: I press profile button

Then: It should load a pop-up page, with the information needed to sign-in

When: I fill out all the information needed for the sign-in

Then: Press the sign-in button

And: I am logged in the website.



Requirement 6: To always have direct access to different pages.

Feature1: To have a banner throughout the website

As a: user

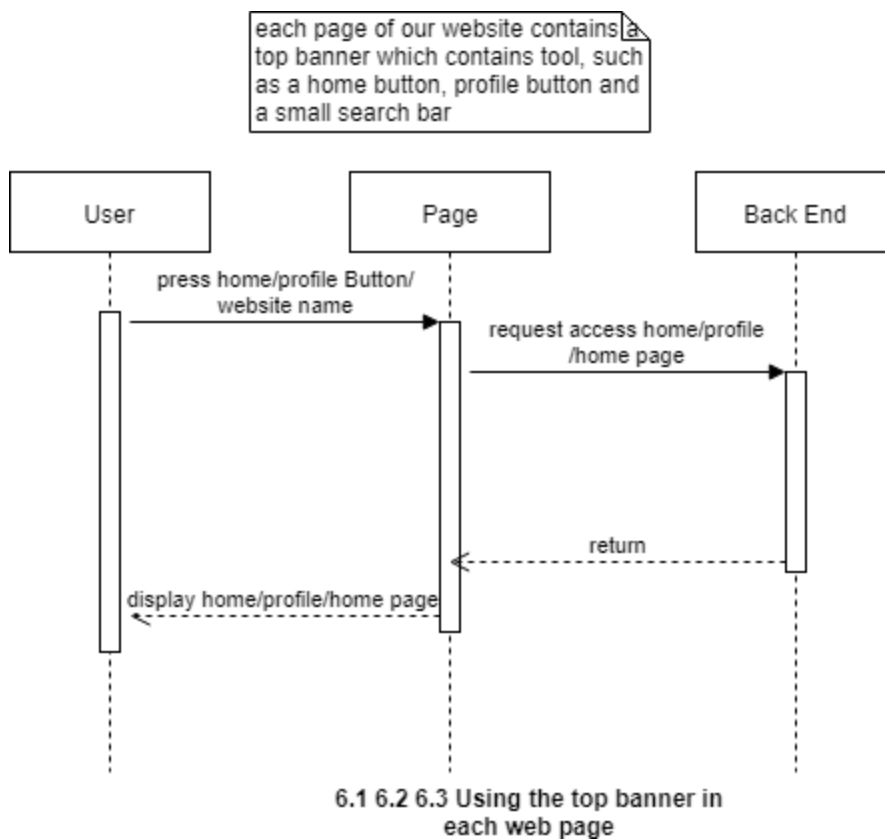
So that: I can navigate to different pages immediately

I want to: always be able to see a banner at the top of my screen while browsing

Scenario: I should see the banner at the top of my screen anywhere in the website.

Given: I am on the home/detailed-destination/search-result/profile page

Then: I should see a banner at the top of my screen.



Feature2: Have a button at the banner to transfer to the profile page

As a: user

So that: I can easily access my trip schedule and edit my profile.

I want to: be able to go to my profile page anywhere on the website

Scenario: Once I click on the profile button on the banner at the top of the screen, I will be navigated to my profile page.

Given: I am on the home/detailed-destination/search-result page

When: I click on the profile button on the banner at the top of the screen

Then: I should be navigated to my profile page

Feature3: Have a button at the banner to transfer to the home page

As a: user

So that: I can easily access the home page.

I want to: be able to go to the home page anywhere on the website

Scenario: Once I click on the home button on the banner at the top of the screen, I will be navigated to the home page.

Given: I am on the profile/detailed-destination/search result page

When: I click on the home button on the banner at the top of the screen

Then: I should be navigated to the home page

Feature4: To have a search bar at the banner

As a: user

So that: I do not have to go to the home page every time I need to search for a destination.

I want to: be able to search for destinations anywhere on the website

Scenario: Once I click on the search button on the banner at the top of the screen, there will be a search bar in which I can type in key words to search for destinations.

Given: I am on the search-result/home/profile/detailed-destination page

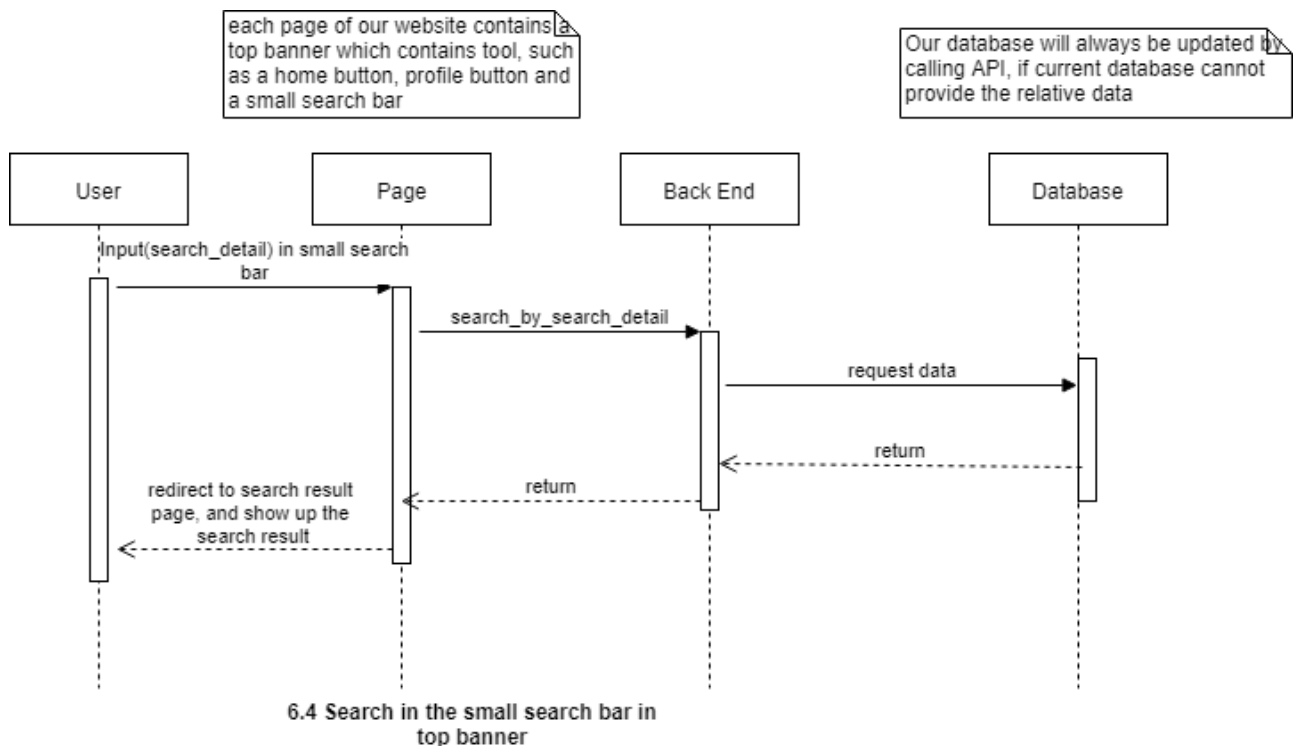
When: I click on the search button on the banner at the top of the screen

Then: I should see a search bar next to the search button

And: I type in key words “Indian Restaurant”

And: I press the search button

Then: I should see all “Indian Restaurants” in all the available cities.



Features5: To have an upwards arrow at the end of every page

As a: user

So that: I want to navigate around the webpage easily

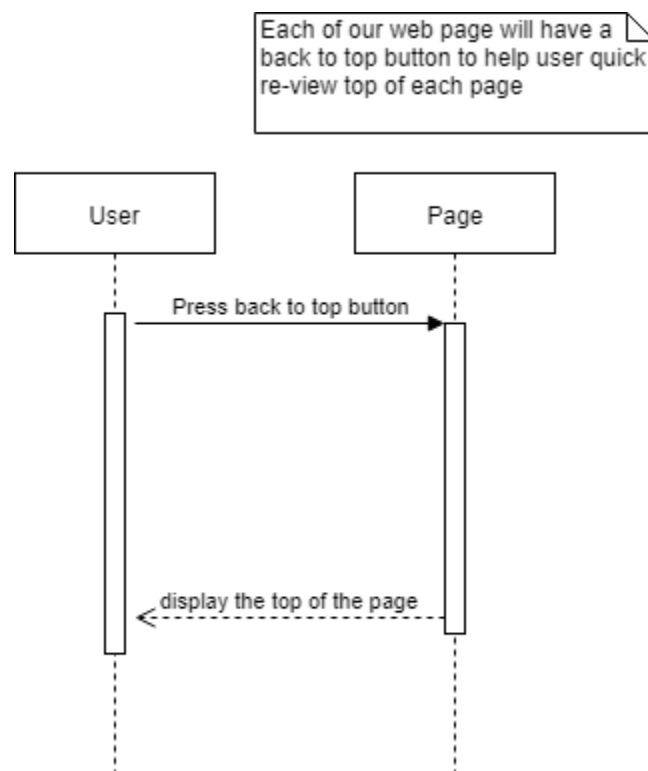
I want to: be able to reach the top of the page with a click of a button

Scenario: Once I click on the “upwards” arrow page I am able reach the top of the webpage

Given: I am at the bottom of the search-result/home/profile/detailed-destination page

When: I click on the “upwards arrow” button on the banner at the top of the screen

Then: I should be at the top of the webpage



6.5 Using back to top button

Requirement 7: To be able to suggest what possible destination there are for the user

Feature1: To have 3 suggested destinations for the city

As a: user

So that: I can decide where to go when I have no destinations in mind

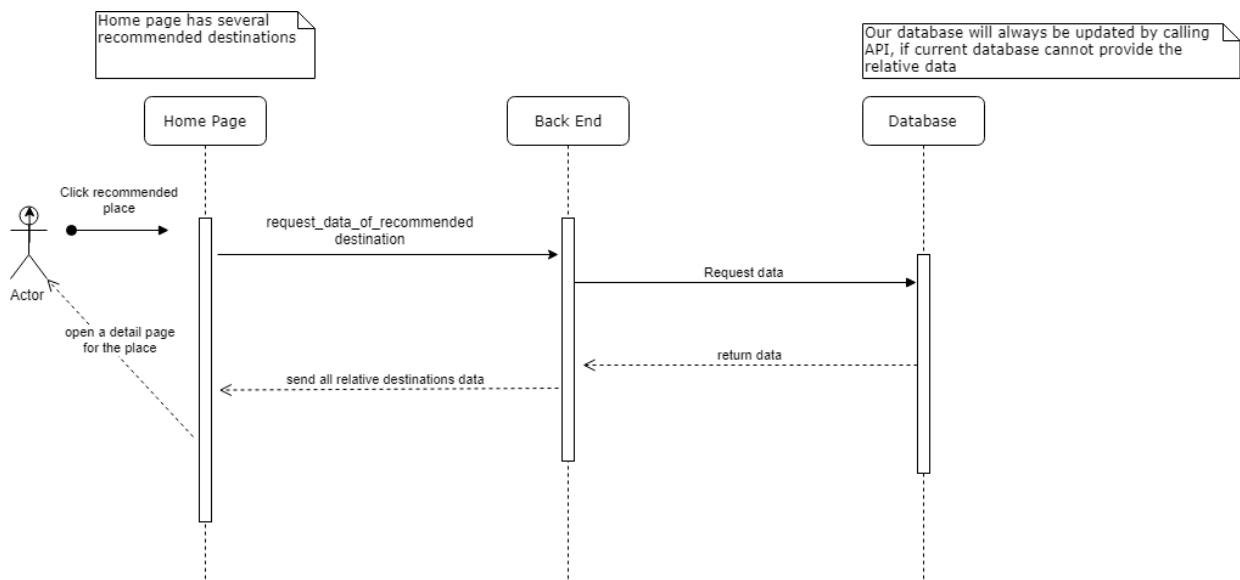
I want to: see recommended places for all the available cities

Scenario: When I scroll down from the home/search page I will be able to see some recommended destinations

Given: I am in the home/search page

When: I scroll down

Then: I see 3 recommended destinations.



7.1 Click on recommendation place on home page

Feature2: To be able to see 3 of the top-visited destinations on the website

As a: user

So that: I can have a guide to some popular destinations if I have no destinations in mind

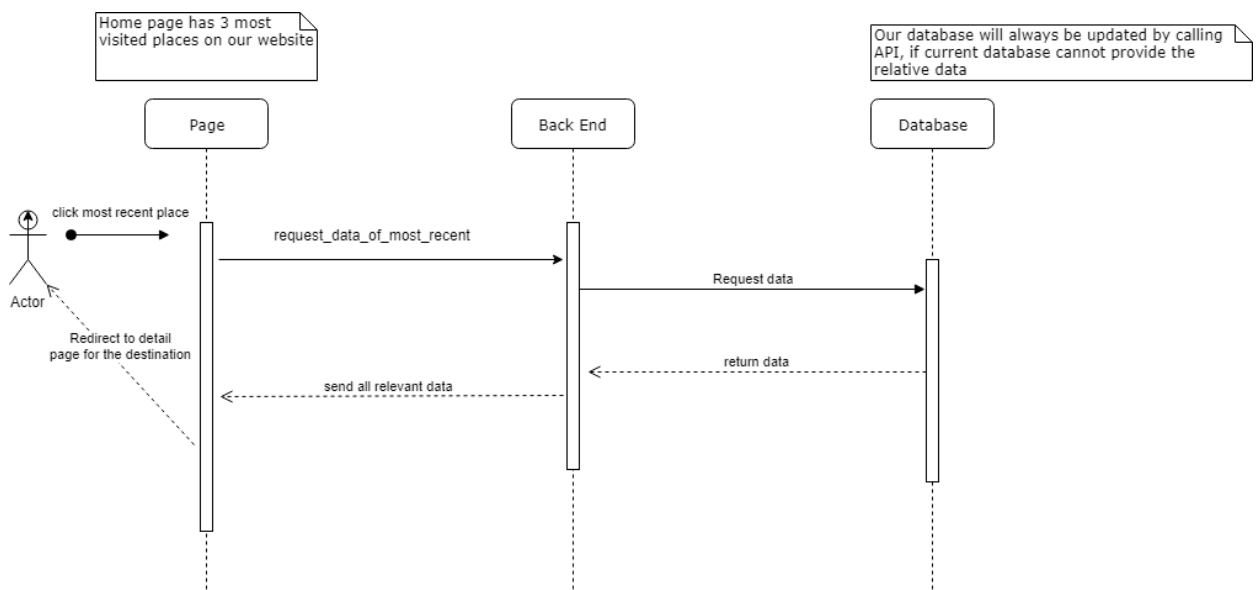
I want to: see the most visited places on the website

Scenario: When I scroll down from the home/search page I will be able to see 3 of the most visited destinations on the website

Given: I am in the home/search page

When: I scroll down

Then: I see 3 of the most visited destinations on the website



7.2 Click on most recent place on home page

Feature3: To have recommendations for some events held in the city

As a: user

So that: I can add more plans with a greater variety in my schedule

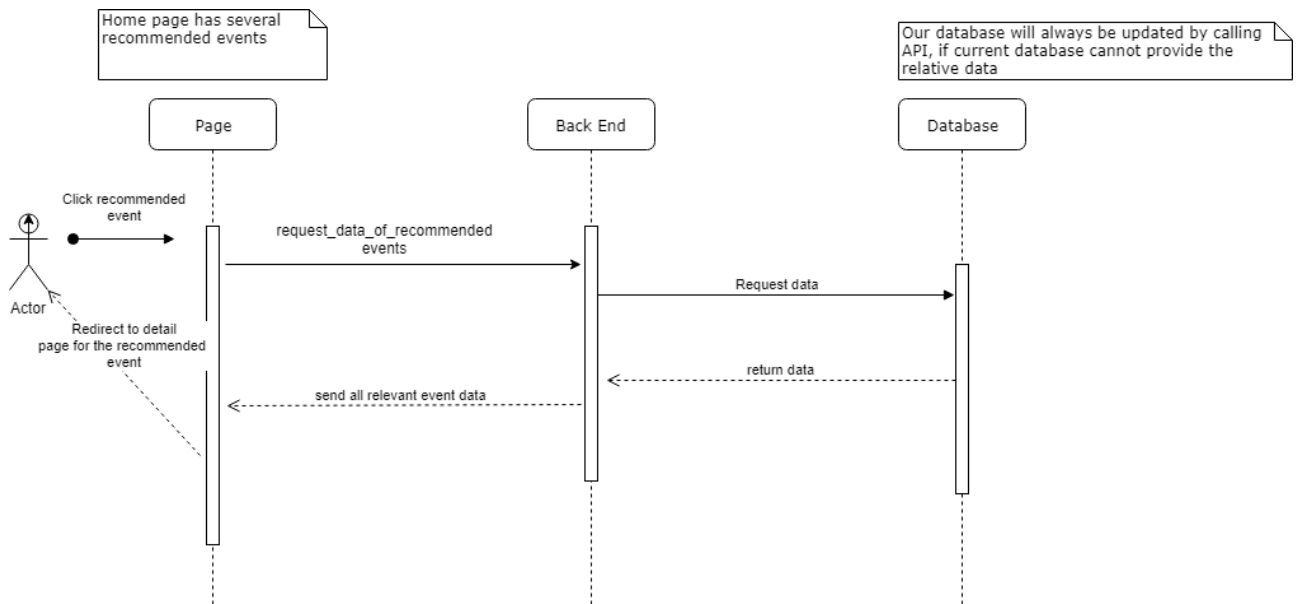
I want to: know some of the events that I can go to in the city

Scenario: When I scroll down from the home/search page I will be able to see recommendations for some events

Given: I am in the home/search page

When: I scroll down

Then: I see recommendations for some events that I can go to



7.3 Click on recommended event on home page

Requirement 8: To be able to edit your profile

Feature1: To have an edit button on the user's profile page

As a: user

So that: I can change my email address and username when needed

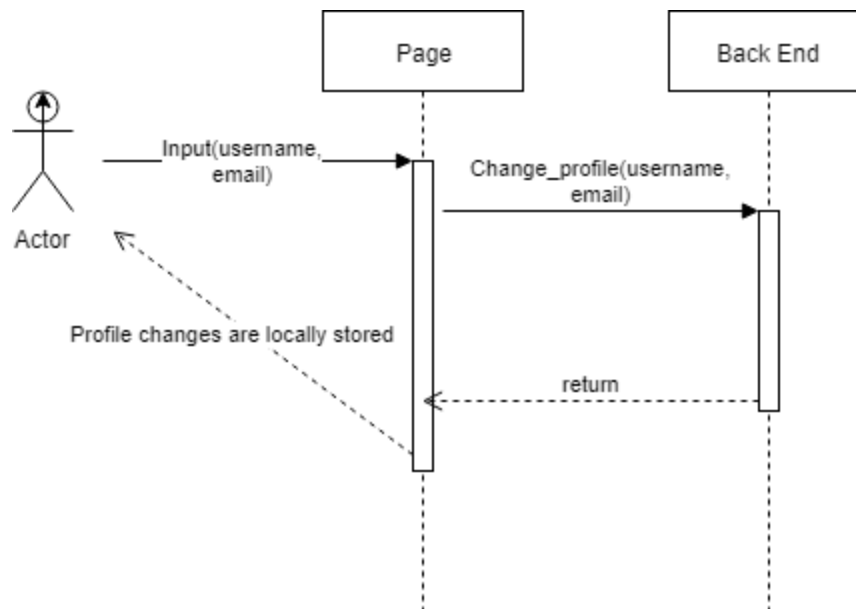
I want to: be able to edit my profile

Scenario: When I click the “edit profile” button, I will see a pop-up window in which I can change my email address and username.

Given: I am in the profile page

When: I click “edit profile”

Then: I should see a pop-up window, in which I can input my new name and email address.



8.1. Edit profile on edit profile page

Feature2: To have a save button on the changing profile window.

As a: user

So that: I can have new profile information linked to my account

I want to: be able to save my changes to my profile

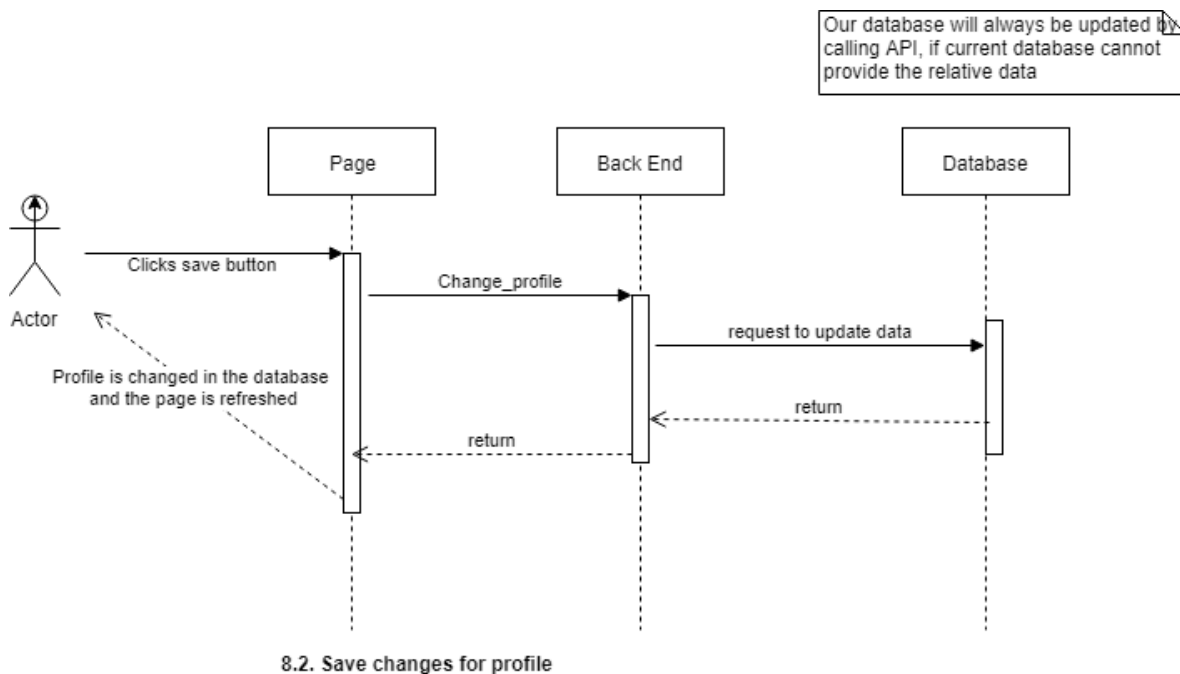
Scenario: When I click the “Save” button, my changes will be saved.

Given: I am in the changing profile window

When: I input a new name and email address

And: I click “Save”

Then: The pop-up window should disappear, and I can see my new name and email address on my profile page.



Conclusion (Team Organisation/Appraisal)

General Team Roles

For the duration of the course, the team utilized some of the key concepts of the scrum process to develop the project. The process we took from it was that we meet regularly to update each other on what we must do and where we are up to and re-developing our plans to match our current process speed. We also did not have an overall team leader, instead we discussed some of the key points of the project and casted votes in what we should do next. Lastly each of the members of the team had a designated webpage that they had to complete because we as a team wanted everyone to have relevant experience in coding.

Project Specific Roles

In terms project specific roles, the team worked on different parts of the prototype as follows:

- Gary oversaw the creation of the backend, this included how the data will be stored and queried out.. However due to other members needing help in their designated pages, Gary became the overall overseer of the project where he helped the struggling team members. He also created the homepage.
- Jay help develop the backend with Gary, this included how the data will be stored and queried out. He was also in charge of detailed-destination page.
- Paul worked on displaying the timetable and developing the profile page. He also oversaw the reports and documentation of the projects.
- Patrick supervised on the search-page and provided help with the reports.

Organisation

Scrum meetings were held at least once a week to discuss the tasks that were completed, tasks that needed to be completed in the foreseeable future, as well as any factors that might hinder our ability to complete the tasks assigned by the given deadline. This was all documented so that members can see what each person had to do and when they must complete the designated tasks. We used trello, where people can post up task and websites that can help further the project.

How Did the Project Go?

In the team's unanimous opinion, we felt that although time constraints were a little heavy for the big project, we set out to develop, we ultimately are satisfied with the prototype that we have developed. The team feels that this experience has bestowed many valuable lessons, not only gaining experience on working collaboratively within a group environment but also gaining exposure to the many languages needed to develop the prototype for this project.

Issues

The issues encountered was that we had people who were less experienced in coding which resulted to Gary helping others in their designated task. Though this was great as the members can improve upon their coding with the help of an experienced coder, it put heavy burden on Gary. Also due to the time constraints provided by the new trimester it felt like all courses were rushed and we did not have time to fully digest the learning and research more heavily in what web stack we should have used or what languages and framework would have made the coding easier. Due to getting assignment and test every week we had less time to gather together and fully discuss as a cohesive unit in what we needed to do. Even though we had a weekly meeting we rarely had a full gathering due to the heavily workload that UNSW is now providing.

Would we do anything different?

What we would change is that we would have liked to use JQuery with our other front-end framework as it would provide an easier design process of the dynamic website we desired. Also, we due to the change in semesters we should have begun our prototype at the very beginning when we realized what we wanted to create. However, because we thought it will be in one of our deliverables we did not, so what we would have liked to change was to start our project earlier so that we would not be pushed for time.

Conclusion

Looking back on the project and the events that transpired, we are happy in what we have achieved and glad that we made new friends along the way. We have learnt valuable lessons in teamwork, where the environment such as the time constraints and heavy workload was tight. In the end we believe that we have enjoyed the sleepless nights and get together after every task was done as it demonstrated that when we all work together,

we can achieve things we could not individually while also creating lasting friendships along the way.