



UNSW
AUSTRALIA

SENG2021 - Requirements and Design Workshop
Team Mongoose

Final Report

2021, Term 1

Aadhishrii Patiil	z5251839
Enoch Luu	z5319999
Huiyi Wang	z5261844
Sanojan Thiagaraja	z5207855
Tingzhuang Zhou	z5208105

Part 1: Requirements Analysis

Purpose

When collaborating with friends to plan a trip, individuals tend to undergo two phases: **deciding** and **planning**. In the deciding phase, they find and suggest places of interest to the group, followed by a vote to determine the options. In the planning phase, they organise the chosen activities in a loose schedule that gives the group a rough idea of the order of activities and how much time you'll be spending on each one.

These 2 phases usually occur notwithstanding technological assistance from apps. Currently, apps that aid the trip planning process only accommodate one of these tasks well. Messaging apps like Skype and Messenger offer a text poll, which lets people nominate and vote for activities. However, planning is difficult when scheduling details are sent through messages. On the other hand, dedicated trip planning apps like TripHobo provide an interactive user experience where you can organise your activities clearly on a schedule, but it doesn't allow collaboration when deciding what activities to choose in the first place. As a result, users are forced to use a combination of apps to communicate, research, decide and schedule activities, making the process unnecessarily convoluted and time consuming.

TripCollab provides a holistic solution that lets users decide and plan in one app, with a strong focus on group collaboration.

Problem Statements

1. Many trip planners only allow one user to plan a given trip, rather than multiple users planning the trip together.
2. Many trip planners don't allow members of a group to make suggestions for activities and vote for their favourite one/s.

3. It is difficult to compare travel destination options from a simple Messenger poll without any metrics (e.g. rating, hours) and visual cues.
4. Many trip planners do not suggest options for activities based on the user's interests.
5. Group chats lack an interactive schedule view to block out time throughout the day for chosen activities.
6. Many trip planners don't suggest nearby side-activities such as restaurants to add to the schedule.

User Stories

1. Requirement: Many trip planners only allow one user to plan a given trip, rather than multiple users planning the trip together

Feature: *Login with social network account*

As a: User of online social network services like Facebook and Twitter

So that: I don't have to spend time creating a new account for the web app and have to remember multiple passwords

I want to: be able to login using an existing social network account

GIVEN that I am on the homepage of the web app

WHEN I click on 'Login'

THEN I should see a 'Sign in with Google' button

WHEN I click on 'Sign in with Google'

THEN I should see the Google Sign-In page

AND I enter my Google email and password

THEN I should be logged in

AND return to the home page

As a: Leader of the group trip

So that: My friends can join me in our decision making of the trip's activities

I want to: Create a new group for a trip on a specified date with a specified location

GIVEN that I am on the ‘Make Group’ page after logging in

WHEN I enter a group name, date of the trip and location of the trip

THEN I should be on a blank ‘Nominations’ page for the newly created group

Feature: *Share trip page links*

As a: Member of the group trip

So that: My friends can nominate activities and vote for the option they prefer for the specific group that they are a part of

I want to: Send a link unique to the group to bring them to the Nominations page

GIVEN that I am on the Nominations page of a certain group that I am a part of

WHEN I click on the ‘Add Member’ icon

THEN the URL to the current group’s trip page is copied to my clipboard

AND I can share the URL of the current group’s trip page with friends

WHEN my friends visit the link in their browser

THEN they are on the Nominations page

2. Requirement: Many trip planners don’t allow members of a group to make suggestions for activities and vote for their favourite one/s

Feature: *Search for an activity*

As a: Member of the group trip

So that: My friends can be informed of my interests

I want to: Nominate activities for consideration by the group

GIVEN that I am on the Nominations page of a certain group that I am part of

WHEN I fill in the search box with “Sydney Opera House”

AND I click ‘Enter’ on my keyboard

THEN I am on the ‘Activity Details’ page of Sydney Opera House

Feature: *Nominate an activity*

As a: Member of the group trip

So that: My friends can be informed of my interests

I want to: Nominate activities for consideration by the group

GIVEN that I am on the ‘Activity Details’ page (for “Sydney Opera House”) of a certain group that I am part of

WHEN I click on the ‘Nominate’ button

THEN I can see the ‘Nominate’ button grayed out, indicating that I have nominated it

Feature: *Vote for an activity nomination*

As a: Member of the group trip

So that: I can indicate my support for a particular activity/s

I want to: Vote for my favourite activity nominations

GIVEN that I am on the Nominations page of a certain group that I am part of

AND at least one activity has been nominated

WHEN I click on the plus icon at the corner of the activity card

THEN the plus icon turns into a tick indicating I have voted for this activity

Feature: *Display vote count for each activity*

As a: Member of the group trip

So that: The group can clearly compare and identify the most popular activities (those with the most votes)

I want to: Visualise the number of votes for each activity as a bar graph, organised from most popular at the top and least popular at the bottom

GIVEN that I am on the Poll Count page of a certain group that I am part of

THEN I should see a list of activity names, each consisting of the number of votes it received, and a bar whose length is representative of the percentage of group members who voted for it, ordered from most to least popular

3. Requirement: It is difficult to compare travel destination options from a simple Messenger poll without any metrics (e.g. rating, hours) and visual cues.

Feature: List of activity cards, each consisting of name, image and rating

As a: Member of the group trip

So that: I can have a better understanding of this activity

I want to: See a snapshot of each available option, consisting of a photo of the activity, name and rating

GIVEN that I am on the Nominations page of a certain group that I am part of

THEN I can see all listed nominated activities with activity name, a photo and a rating out of five

Feature: Activity Details view providing more information about a specific activity

As a: Member of the group trip

So that: I don't have to manually research each option through Google

I want to: See more detailed information about an activity

GIVEN that I am on the Nominations page of a certain group that I am part of

WHEN I click on the activity with name "Sydney Opera House"

THEN I am on the 'Activity Details' page of Sydney Opera House

AND I should see detailed information about "Sydney Opera House", including a brief description, images, rating, contact information, opening hours and reviews

4. Requirement: Many trip planners do not suggest options for activities based on the user's interests.

Feature: Auto-generate activity suggestions based on responses to questionnaire

As a: Member of the group trip

So that: I don't have to research online to find new activities

I want to: Rate my interest in different categories of activities and receive suggestions that are tailored to my interest

GIVEN that I am on the 'Nominations' page of a certain group that I am part of

WHEN I click on 'Don't know where to go?'

THEN I should see a questionnaire pop-up asking me to rate my interest in given activity categories via a slider

AND I rate my interest by dragging the slider of my preferred types to the right, in this case, ‘Sightseeing’ and ‘Recreation’

WHEN I click on ‘Find activities’

THEN I should be able to view a three suggested activities based on my response in the questionnaire (in this case, sightseeing and recreation activities)

5. Requirement: Group chats lack an interactive schedule view to block out time throughout the day for chosen activities.

Feature: Add activities from poll to a schedule

As a: Leader of the group trip

So that: I can more easily plan out the day through a visual aid

I want to: Select the most popular activities in the poll to add to the schedule

GIVEN that I am on the Poll Count page of a group that I am part of

WHEN I click on the “Add to Calendar” icon on the ‘Sydney Opera House’ activity

THEN the activity is added to the schedule

WHEN I click on “Schedule” on the sidebar

THEN I am on the Schedule page of the same group

AND I can see that a ‘Sydney Opera House’ time block has been added to the group’s schedule

Feature: Schedule containing chosen activities, with equal amount of time set for each one (by default)

As a: Member of the group trip

So that: I can get a general estimate of how much time we have for each activity

I want to: View the chosen activities in a schedule

GIVEN that I am on the Schedule page of a group that I am part of

THEN I can see auto-generated time-boxes of equal length for all activities I chose in the Poll Count page

Feature: *Each activity's time block can be dragged around, shortened and stretched*

As a: Leader of the group trip

So that: I have greater control on how we spend the day, and minimise the chance of running out of time for certain activities

I want to: Arrange the order of as well as time to spend on each activity

GIVEN that I am on the Schedule page of a group that I am part of

WHEN I drag the 'Sydney Opera House' 9:00 time block and move it to the 11:00 row

THEN I can see the Sydney Opera House time block is now located in the 11:00 row

AND I drag the bottom or top edge of the time block down or up

THEN I can see the 'Sydney Opera House' time block is bounded by the point of release

6. Requirement: Many trip planners don't suggest nearby side-activities such as restaurants to add to the schedule.

Feature: *Auto-suggest options for side-activities closeby to the chosen main activities*

As a: Member of a trip

So that: We have extra options for things to do such as food, lodging etc. in-between main activities

I want to: See a list of side-activities that are nearby to the main activity

GIVEN that main activity/s have been added to the schedule of a group that I am part of

WHEN I click on "Side Activities" on the sidebar

THEN I am on the Side Activity Suggestions page

AND I can see a list of side activities nearby the chosen main activities

Feature: *Add side activity to the schedule*

As a: Leader of a trip

So that: I can make time in the schedule for nearby side-activities

I want to: Add side activity/s to the schedule

GIVEN that I am on the Side Activity Suggestions page of a group that I am part of
AND side activities have been suggested to me in the form of activity cards

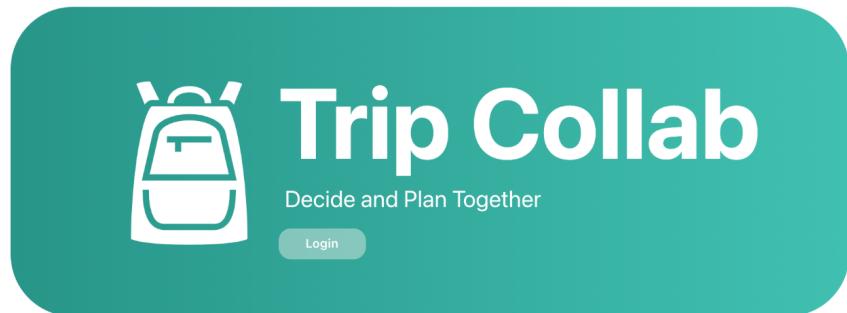
WHEN I click on the plus icon at the corner of an activity card

AND I click on “Schedule” on the sidebar

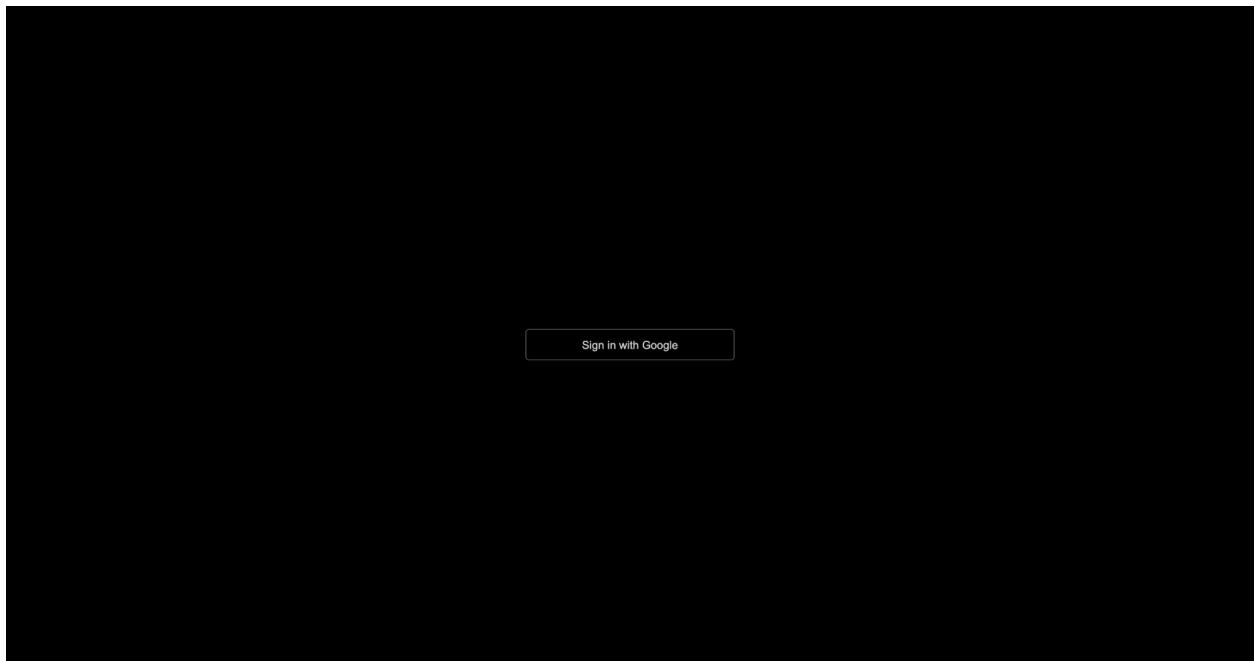
THEN I am on the Schedule page

AND I can see the chosen side activity in the timetable

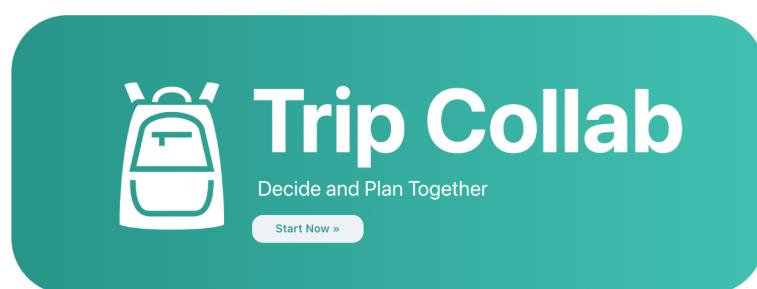
User Interface



This is the landing page of our website. Users are provided with an option to 'Login.'



Users are given an option to Sign in using Google.



Once the user has signed in he/she can click the 'Start Now' button to start using the website.



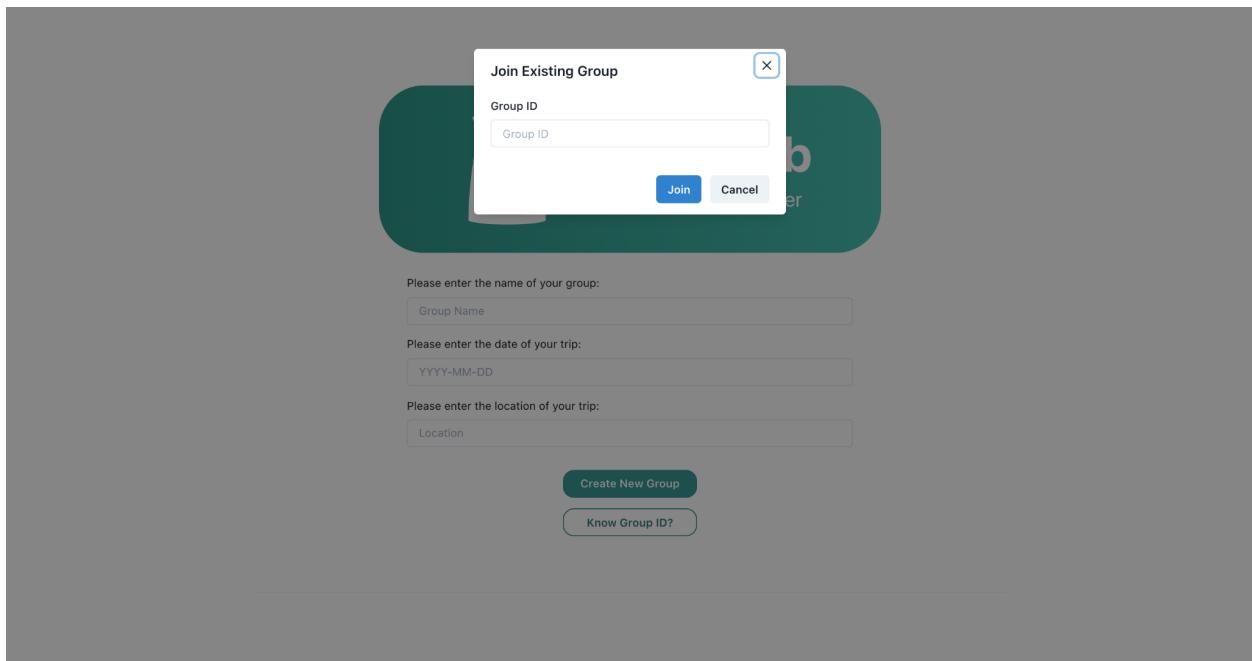
Please enter the name of your group: Team Mongoose
Team Mongoose

Please enter the date of your trip: 2021-05-10
2021-05-10

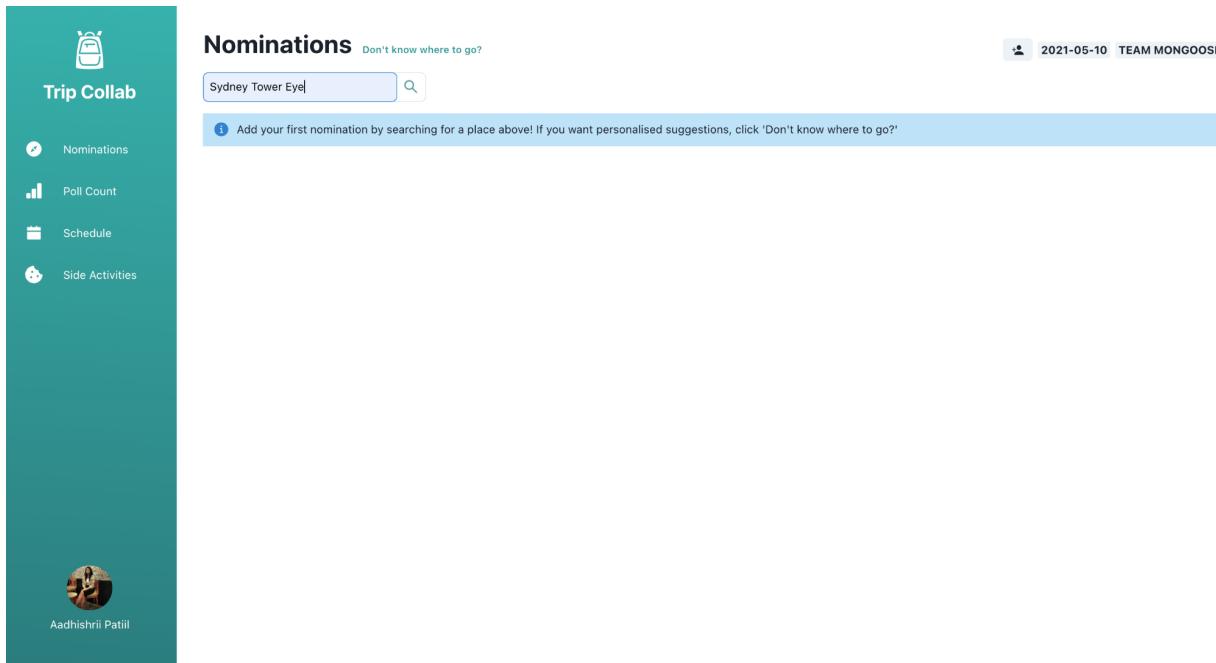
Please enter the location of your trip: Sydney
Sydney

[Create New Group](#)
[Know Group ID?](#)

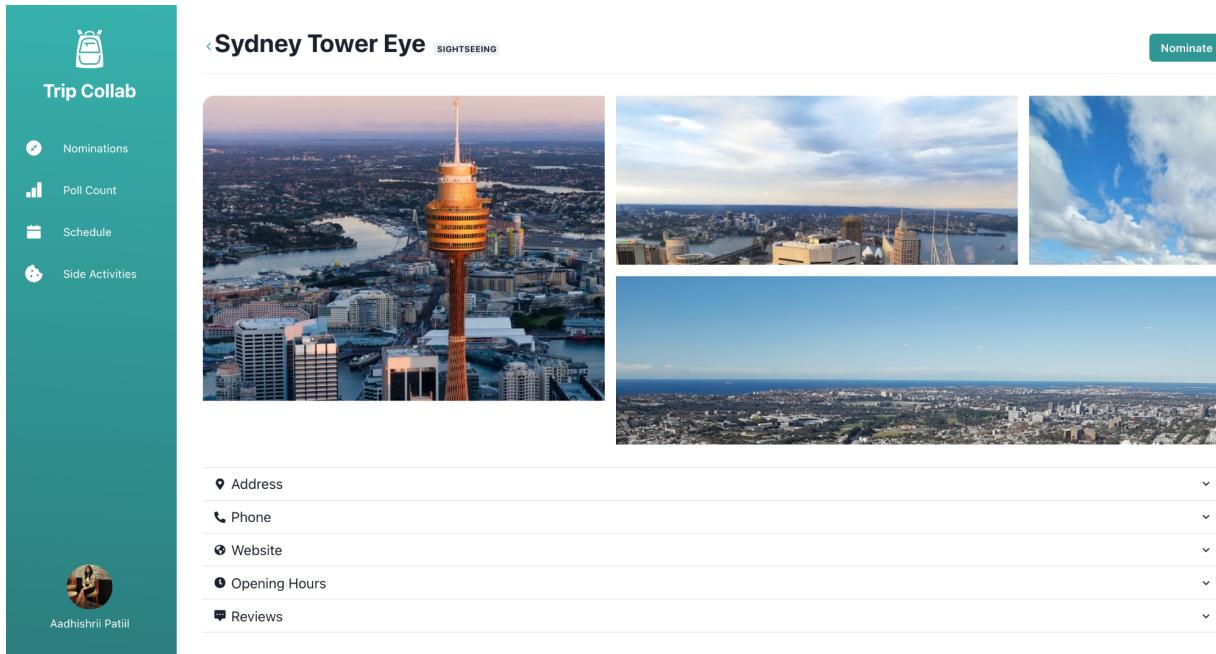
Users can create a new group by filling the form (which will make them the owner of the group).



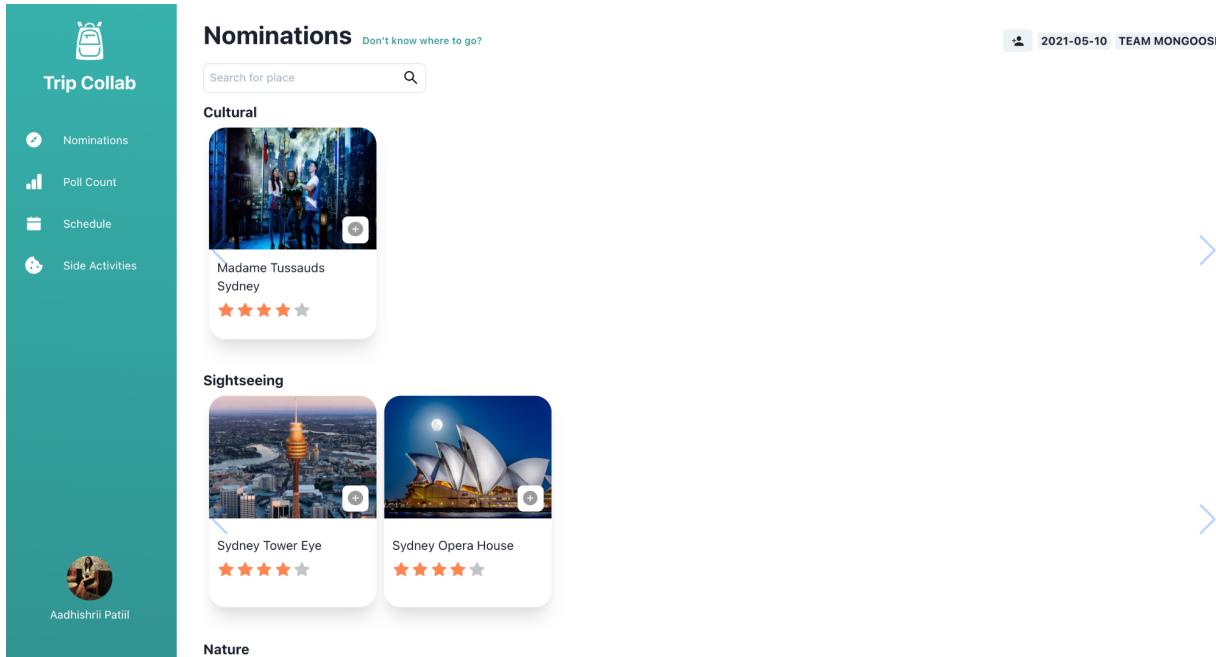
Alternatively, users can join an existing group by clicking on the 'Know Group ID?' button and entering the Group ID provided by the owner.



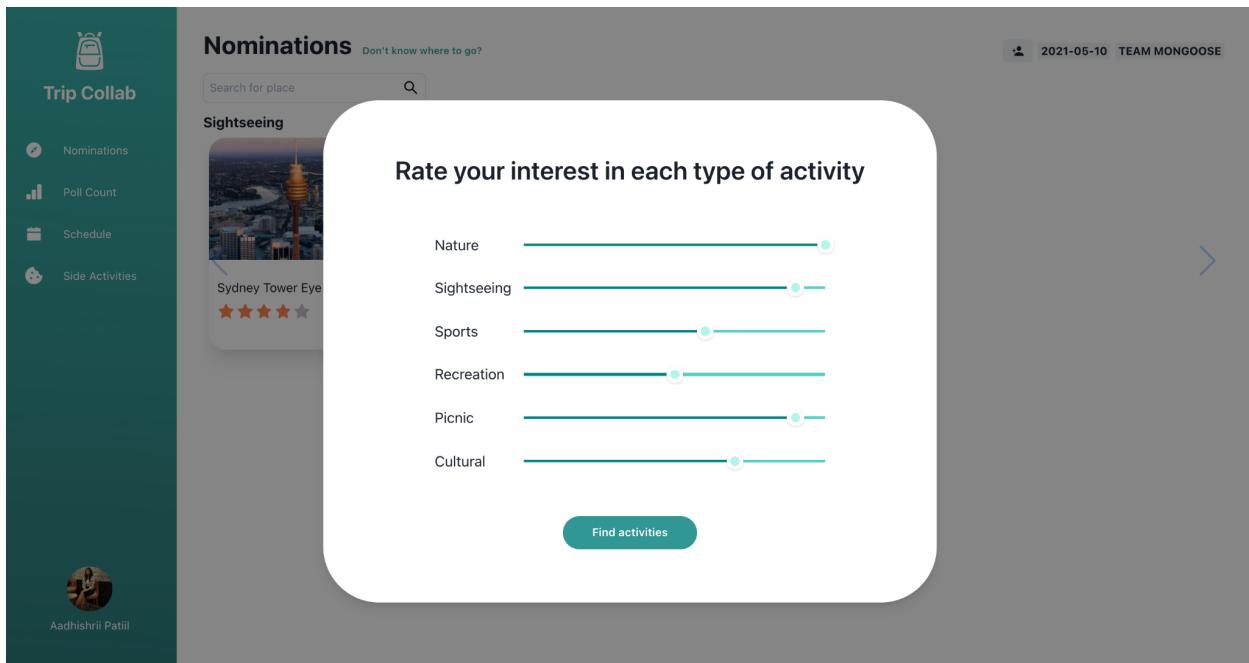
When the user logs in for the first time the nominations page is empty as there are no nominated activities. The user can search for a location by typing the name of the location in the search box.



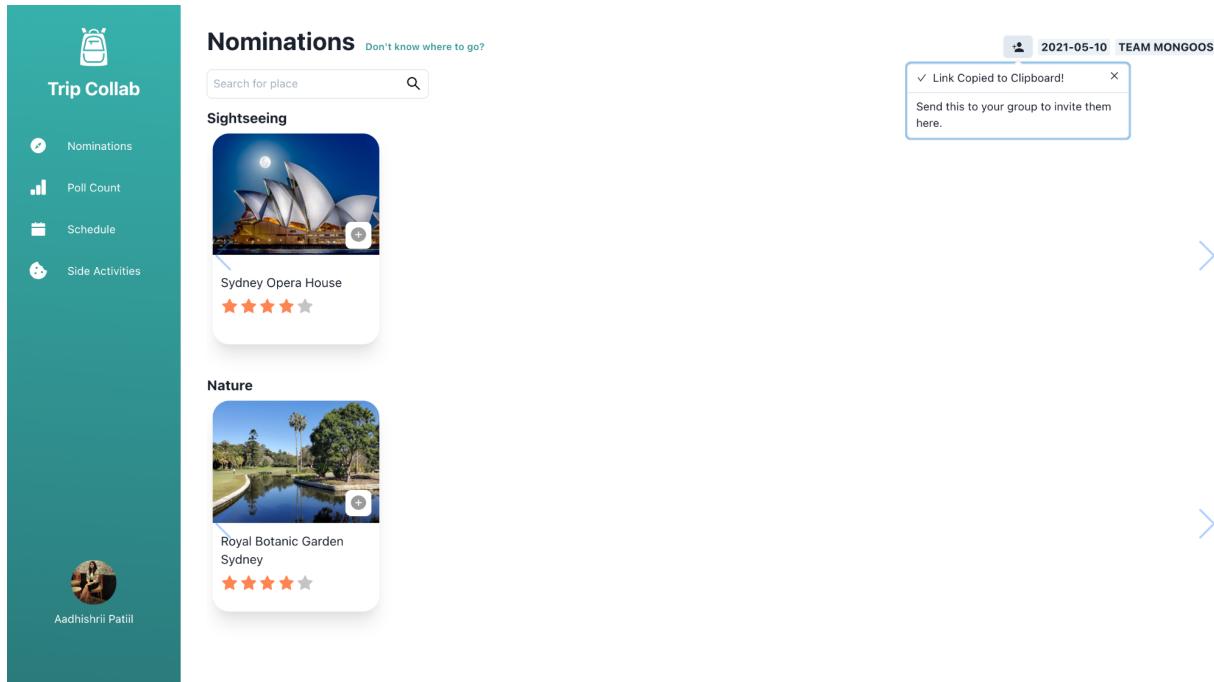
On clicking search for a particular location the details of the location show up (Images, Address, Phone Number, etc). If the user is interested, he/she will click on the 'Nominate' button.



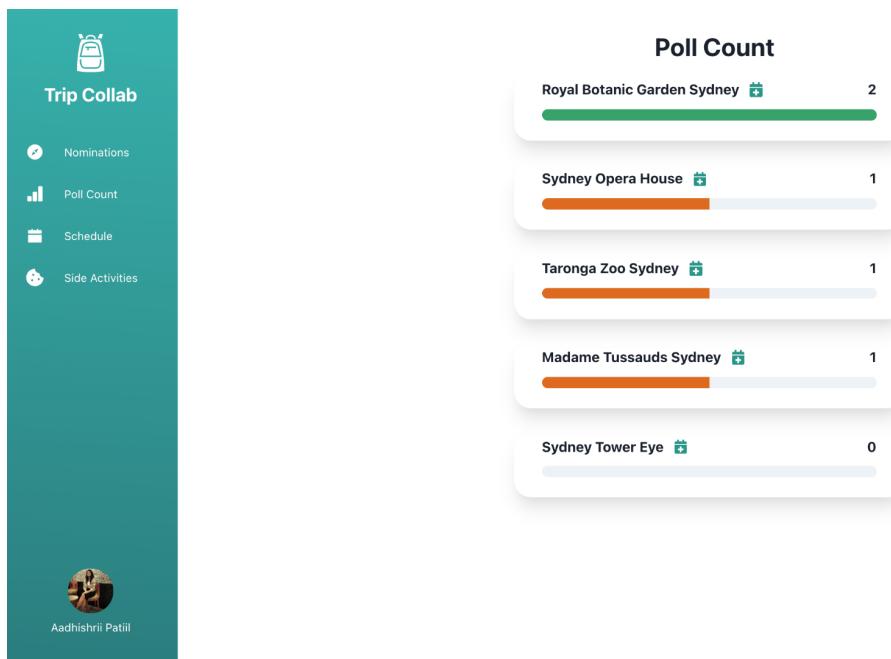
All the nominated activities appear on the nominations page and are visible to everyone who is a part of this group. All the members are free to vote for their favourite activities.



The user can click “Don’t know where to go?” if they don’t have a specific place they’d like to search. They can fill this form and find activities according to their interest.



The owner can also add members to the group using the “Add members icon” on the top right which copies the link to the group page. This can be shared with other users who wish to join the group.



The Poll Count Page displays the number of votes received by each nominated activity. The owner of the group can add activities with the highest number of votes to the schedule.

The Schedule Page displays the plan for the day. This can only be modified by the owner of the group.

The Side Activities Page displays all the activities that are in the vicinity of the selected main activities. The owner of the group can add these activities to the schedule if he/she wishes to do so.

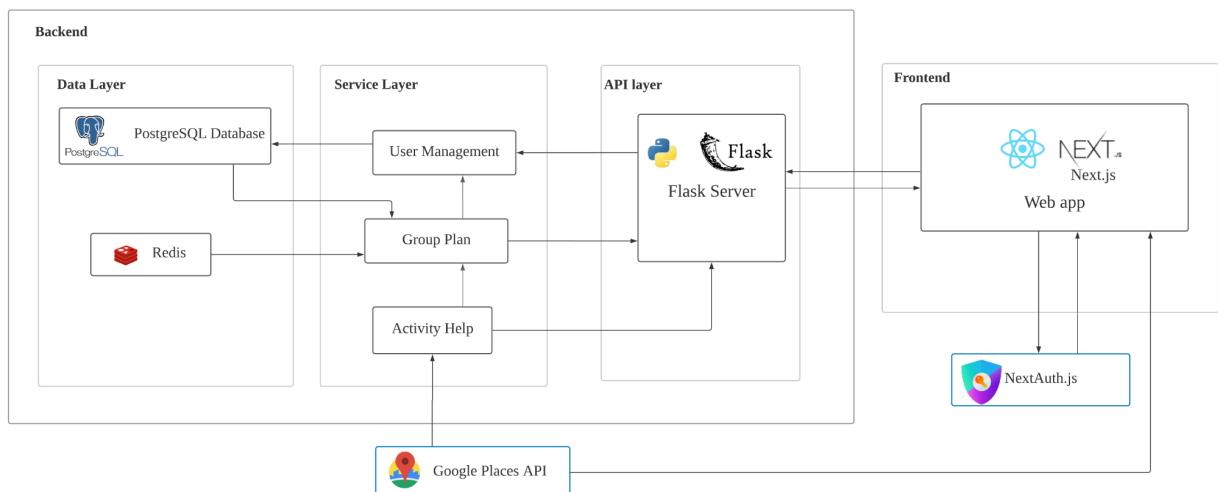
Part 2: Software Architecture and Design

External Data Sources

We used the Google Places API to show location information about activities that users nominate. This includes the name, relevant images, category, opening hours, contact information and reviews.

Google Maps provides 99% coverage of the world so it will likely have data on any location that a traveller is interested in. In addition, with over 1 billion monthly active users, it will have many up-to-date reviews on most activity locations. Finally, its \$200 free credit allows us to access its entire data set at no charge (for our usage).

Software Components



Software Architecture Diagram

Software components will consist of our own as well as those from a third party.

NextAuth.js provides authentication and authorisation for our web app. By utilising its default APIs, we can avoid having to implement authentication manually instead focusing

our time and effort on the distinguishing features of our project. It also allows the end-users to securely log in through their existing social accounts instead of having to create a new account with our app. In addition, identifying users by their access tokens only across different pages on the frontend ensures anonymity in the voting and nominating process.

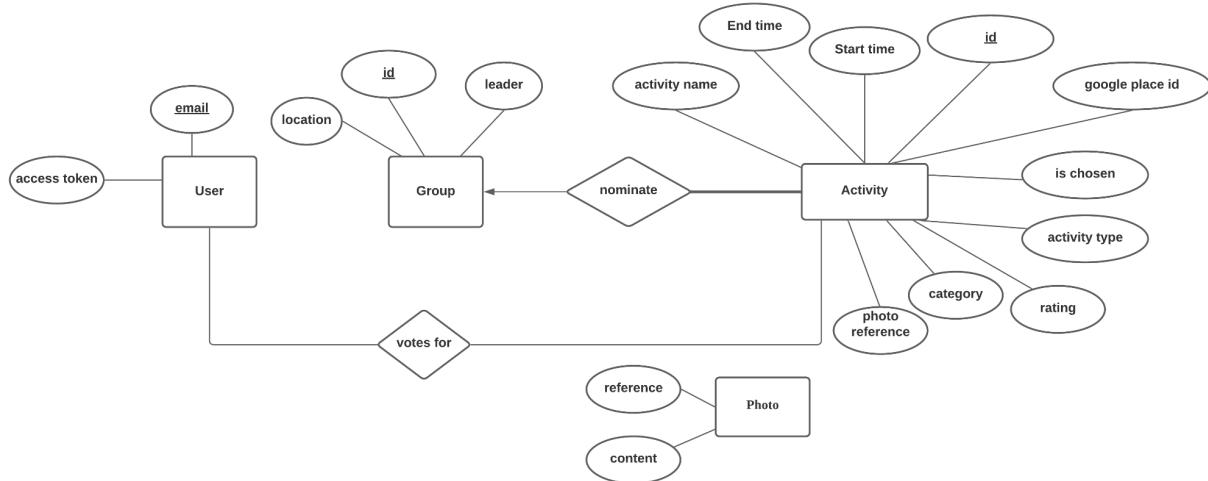
The NextAuth.js component links to our web app in the frontend. The default ‘signIn’ API provided by NextAuth.js redirects the authenticated user to our home page, returning the user’s email. We store this email, alongside a UUID that we generate on the frontend as the access token, in the *Postgres database*. This is handled by the **User Management** component. In each session, the access token is updated. Doing this ensures that users are authorised when using the web app.

Our ‘**Group Plan**’ component handles the key functions that group members should be able to do: create a group, nominate and vote for activities, view the poll results, and schedule the activities. Checks are also made to ensure that the user has the correct permission to carry out certain tasks; only the group owner can add activities to the schedule and manipulate the schedule. To do this, we connected the ‘*User Management*’ component to retrieve the email of the current user (using their access token), and verifying if this was the group leader. In terms of data storage, nominations are stored in the *Redis*, while group details, activities and votes are stored in the *Postgres Database*.

Our ‘**Activity Help**’ component consists of secondary functions that involve invoking the Google Places API to provide information about locations to group members. These functions include the questionnaire which asks users to rate their interest in various types of activities, as well as the activity details page to find out more information about a specific activity. Both of these features will make requests to the Google Places API and return information regarding activity details, suggestions based on the questionnaire

response and photographic content through a photo reference. The process does not send any information to the database until an activity has been nominated.

The '**PostgreSQL Database**' component stores information that is specific to each group. This includes the group name, trip date, activities that are nominated as well as photo data of each place the activity is taken place.



ER Diagram for PostgreSQL Database

The '**Redis Database**' component stores the basic information for all activities, including activity name, rating, activity category and a reference to the activity photo in Google Places API. This component is used to improve the performance of reading basic information about activities in “Nomination Page”, since we assume users will visit this page frequently when they search, nominate and vote for activities sending requests to Google Places API and PostgreSQL database for this information repetitively may be slow. We use the hash data structure supported by Redis to store the activity name, rating, photo reference and category. This allows us to quickly retrieve the information needed for displaying activities on the “Nomination” page for each group.

Web Stack Choices

The web stack of our software:

- Frontend Tier: **ReactJS, NextJS**
 - Libraries: Chakra-UI, Swiper, React Scheduler, NextAuth.js
- Backend Tier: **Python, Flask**
- API Tier: **Google Places API**
- Data Tier: **PostgreSQL, Redis**

We have built a Single-Page Application (SPA) in order to produce a highly responsive and intuitive web app for a more pleasant user experience. To achieve this, we have chosen **React JS**, a popular and powerful Javascript library, which has a robust set of features such fast rendering times, high stability and reusable components. We also considered alternative frontend frameworks such as Vue and Angular, however we preferred React's quicker learning curve and our frontend engineers' familiarity with it.

In addition, we decided to use **Next.js**, an open-source React front-end framework that enables server-side rendering and generating static websites for React-based web applications. There are several advantages of using this instead of React on its own:

- *Server Side Rendering:* When a user interacts with a website, it needs to be responsive in order to maintain the user's attention. Using server-side rendering, Next.js builds the HTML and serves the pre-rendered page to the browser with minimal Javascript code. Once the page is loaded by the browser, the Javascript runs to make the page interactive. This leads to better performance.
- *Automatic Code Splitting:* Next.js only loads the CSS and Javascript needed for a given page. This increases the performance as there is less to download and the user can see the page quicker.
- *Page-based Routing:* Next.js provides an intuitive page-based routing system which allows wherein files inside the pages directory are automatically made available via

the URL. Furthermore, it supports dynamic routing which is useful for passing in the unique group_id URL parameter in our pages.

Python is the language we chose for our backend components as it is easy to use and all team members are proficient in it. Furthermore, we use **Flask**, as the web development framework. We chose Flask over the popular Python framework Django as it is light-weight, more flexible and extensible than Django. It is designed to make getting started quick and easy. Since our web project doesn't rely on heavy use of backend functioning, we are unlikely to benefit from the structured blueprint provided by Django. Finally, as Flask is taught in the prerequisite course COMP1531, it is familiar to all members. Thus, Flask was the better choice for us.

In the data tier, we chose **PostgreSQL**, a relational database management system that is reliable, secure and is largely SQL compliant. We also considered MySQL which is known for its ease of setup and use. However, as it was only partially SQL compliant, we thought the syntax may be more difficult to understand. Furthermore, our members were familiar with PostgreSQL, having studied it in COMP3311. Ultimately, these factors led to our decision to use PostgreSQL for our relational database.

We also decided to use **Redis**, a NoSQL in-memory database. Redis will be used to cache our basic activity information in addition to the existing PostgreSQL database since reading from cache is much faster than from a relational database and allows more flexible data storage. This speed is beneficial for the 'Nominations' page that will be frequently visited. However, Redis is an in-memory data structure store and will use much more RAM. Despite this, we can avoid storing too much data by setting an expiration time (48 hours) for each activity's data. We assume that most users are able to finish the activity decision process in two days and it is unlikely for them to request the activity information after that. In the event that they do, we are still able to retrieve the data as all

activities and their information are stored in the PostgreSQL database persistently. Also, compared to other NoSQL databases like MongoDB, Redis has a relatively simple data structure and is extremely easy to set up. It also outperforms MongoDB in storing and reading with the smaller database size that we expect to have for our use case.

Chakra UI is a simple, modular and accessible component library that gives developers the building blocks they need to design their React Application. It provides some key advantages over manually styling through CSS.

- It provides tools for writing and styling components quickly. This made the coding process for our frontend much simpler and faster.
- The library provides few useful hooks which allowed us to reuse the logic for managing pop ups and modals in our project.
- The components system is highly modular and provides a flexible way of building styled components.

Swiper

Swiper is a modern touch slider API that provides intuitive transitions between slides. We decided to use this to display the list of activity cards on our ‘Nominations Page’ and ‘Side Activities Page’ in each category. It is particularly useful if the list of cards is longer than the viewport width as the user can swipe through to see each card fully. Adding in a Swiper slider was much easier and time-efficient than trying to implement a similar feature manually though Javascript.

NextAuth.js

NextAuth.js is a complete open-source authentication library for Next.js applications. It provides built-in support for passwordless OAuth2.0 authentication with a large range of services, including Google and Facebook. As we mentioned in the software component section, it handles the user authentication process in our web app, which makes the user

management in our app simpler and more secure. We chose NextAuth.js instead of Auth0 since it is built specifically for Next.js app and thus integrates more easily.

React Scheduler

DevExtreme React Scheduler is an API that provides a React component to represent scheduled data and allows a user to manage it. We used this to build our interactive schedule page. It provides several intuitive features such as:

- *Day View:* Appointments are arranged from top to bottom. This results in a clean and flexible day view
- *Drag-and-Drop Editing:* This feature enables users to intuitively drag and drop appointments. Furthermore, by pulling the edge of the top/bottom of the appointment, the length of the appointment is updated. This feature greatly enhances the user experience and would be difficult to code manually.

Tool Choices

Postman

We used Postman for backend API testing. Postman is one of the most popular tools for manual API testing as it supports all HTTP methods. It also allows us to save the API information including sample request and response, which makes testing much easier and more reusable. Testing via Postman allows us to see the response data returned by the backend and quickly determine which API went wrong.

Docker

Docker is used in the deployment process of our application. Docker containers allow us to package up the application with all the dependencies it needs, including services and libraries, in one go. It ensures that the application can be run on any machine with docker installed regardless of different operating systems and customised settings. The

consistency of performance is guaranteed in Docker containers as each container runs in an environment that is isolated and standardized. By tagging Docker images with different versions, managing the product of each iteration becomes more convenient. We chose to use Docker over deploying on Heroku since Heroku requires us to run in its own cloud environment. Docker containers are more flexible and allow us to easily deploy on various platforms later on.

Platform Choice

In terms of browser platforms, we ensured compatibility with the most popular web browsers such as Google Chrome and Firefox. In particular, the user experience is optimised for desktop and laptop-sized devices as we believe the research-focused nature of our app would be better suited to these users as opposed to mobile devices.

User stories with Sequence Diagrams

Feature: Login with social network account

As a: User with a Google account

So that: I don't have to spend time creating a new account for the web app and have to remember multiple passwords

I want to: be able to login using an existing Google account

GIVEN that I am on the homepage of the web app

WHEN I click on 'Login'

THEN I should see a 'Sign in with Google' button

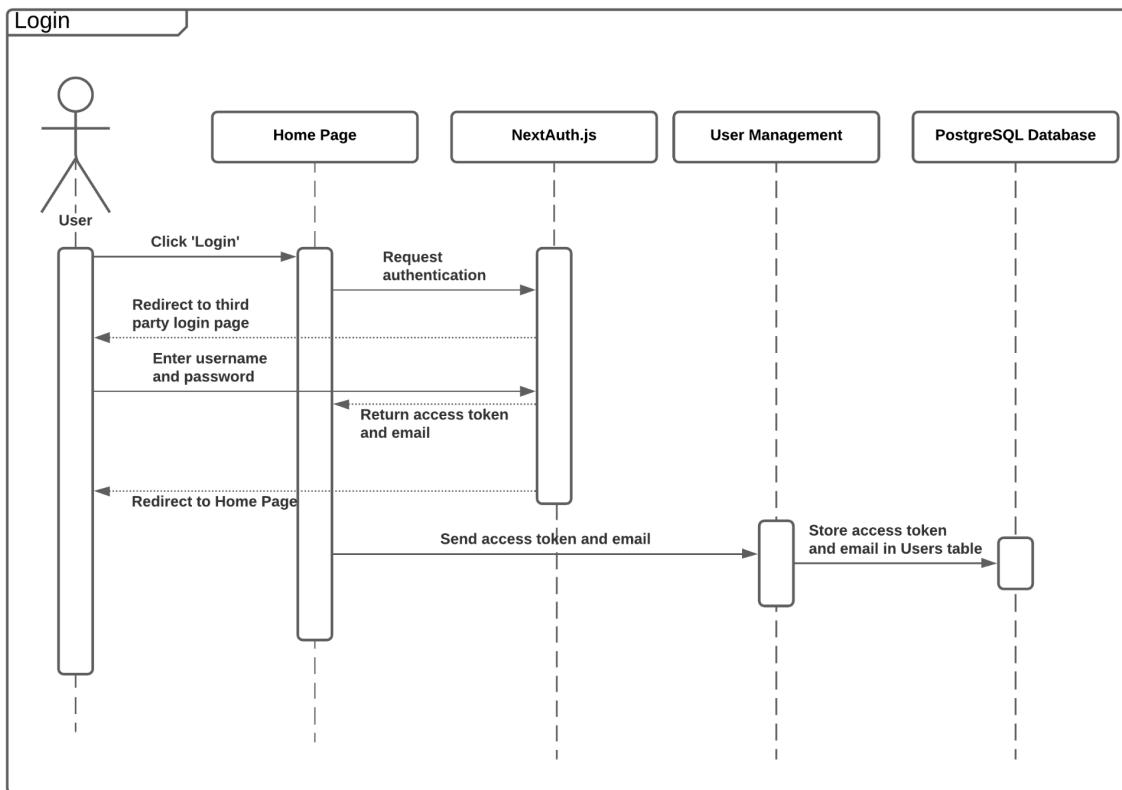
WHEN I click on 'Sign in with Google'

THEN I should see the Google Sign-In page

AND I enter my Google email and password

THEN I should be logged in

AND return to the home page



POST /login

user login

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "email": "user@gmail.com",
  "token": "XAGgr4bilhWNcpqrAhcpoAZDZD"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Content Accept header:

Example Value | Schema

```
{}
```

Feature: Create a new group

As a: Leader of the group trip

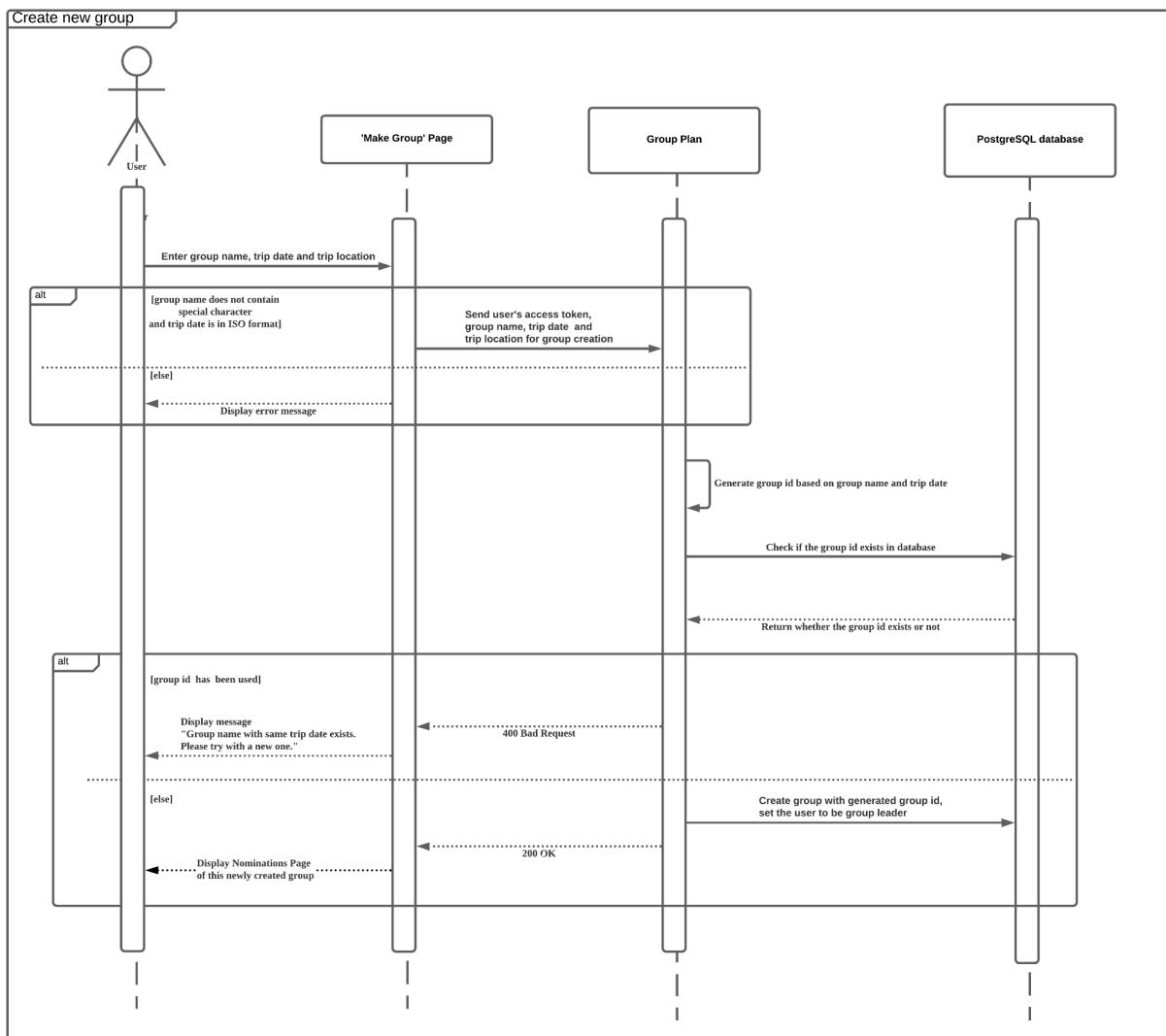
So that: My friends can join me in our decision making of the trip's activities

I want to: Create a new group for a trip on a specified date with a specified location

GIVEN that I am on the 'Make Group' page of the web app after logging in

WHEN I enter a group name, date of the trip and location of the trip

THEN I should be on a blank 'Nominations' page for the newly created group



POST /group

create a new group

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{ "group_name": "group1", "token": "XAAggrRb1HmCpqrAhcpoAZDZD", "trip_date": "2021-3-31", "location": "Sydney" }
```

Responses

Code	Description	Links
200	OK	No links
400	Bad Request	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{}
```

Media type

application/json

Example Value | Schema

```
{ "error": "Group name exists. Please try a new one." }
```

Feature: Search for an activity

As a: Member of the group trip

So that: I can find information about an activity that interests me

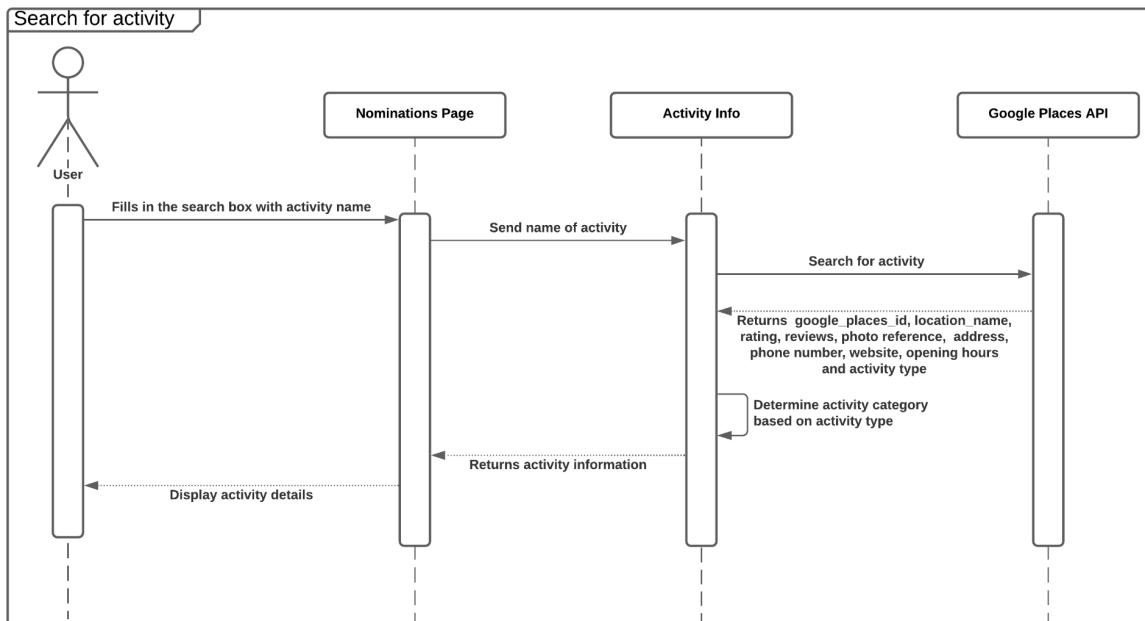
I want to: Search for specific activities

GIVEN that I am on the Nominations page of a certain group that I am part of

WHEN I fill in the search box with “Sydney Opera House”

AND I click ‘Enter’ on my keyboard

THEN I am on the ‘Activity Details’ page of Sydney Opera House



GET /search

search place by name

Parameters

Name	Description
activity	Example : Sydney Opera House string (query) Sydney Opera House

Responses

Code	Description	Links
200	OK	No links

Media type: application/json

Example Value | Schema

```
{
  "google_places_id": "ChIJgUbEo8cfqodwfoI",
  "place_name": "Sydney Opera House",
  "rating": 5,
  "photo_references": [
    "ATyBwUighis5tXKBy3e4RfmkQX4T"
  ],
  "address": "Bennelong Point, Sydney NSW 2000, Australia",
  "phone_number": "(02) 9250 7111",
  "website": "https://www.sydneyoperahouse.com/",
  "opening_hours": [
    "Monday: 9:30 AM - 5:00 PM"
  ],
  "category": "nature",
  "reviews": [
    {
      "author_name": "Jeffrey O'Neill",
      "rating": 4,
      "text": "Came with a friend to enjoy my guest visit back after lock down.\n\nGreat views. Staff are friendly and efficient. Prices ain't too bad. You're paying for the amazing views.\n\nFun place to sit and chat"
    }
  ]
}
```

Feature: Nominate an activity

As a: Member of the group trip

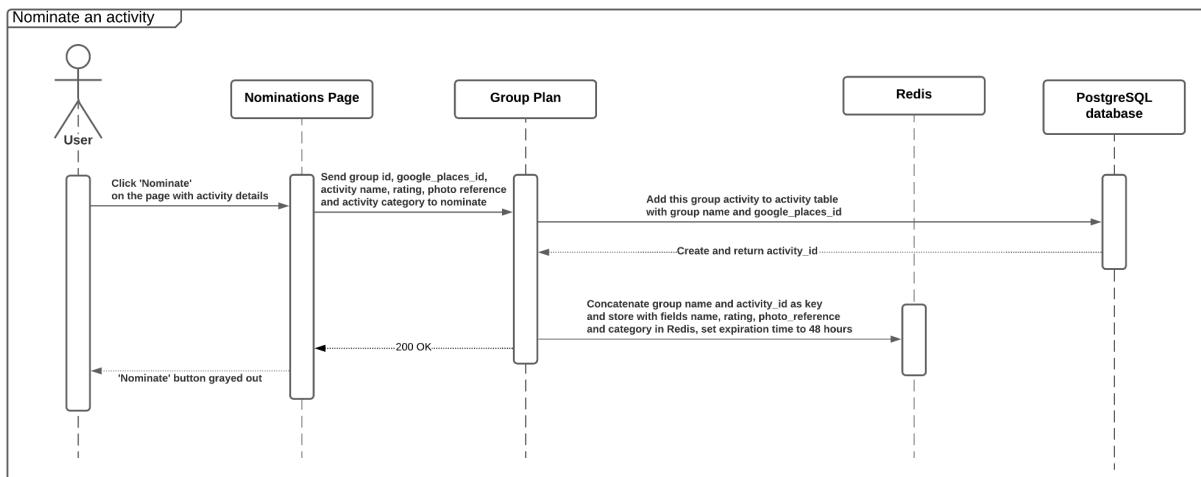
So that: My friends can be informed of my interests

I want to: Nominate activities for consideration by the group

GIVEN that I am on the ‘Activity Details’ page (for “Sydney Opera House”) of a certain group that I am part of

WHEN I click on the ‘Nominate’ button

THEN I can see the ‘Nominate’ button grayed out, indicating that I have nominated it



POST /activities/nominate

nominate activity

Parameters [Try it out](#)

No parameters

Request body required [application/json](#)

Example Value | Schema

```
{
  "group_id": "group1_2021-04-11",
  "google_places_id": "ChIJgUbEoIcFqodwifo!",
  "activity_name": "Sydney Opera House",
  "rating": 5,
  "photo_reference": "ATTyYBwKghisSTXRsyeRfakQX4T",
  "category": "sightseeing"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type [application/json](#)

Contains [Accept](#) header

Example Value | Schema

```
{}  
[{"id": 1, "name": "Activity 1"}, {"id": 2, "name": "Activity 2"}]
```

Feature: Auto-generate activity suggestions based on responses to questionnaire

As a: Member of the group trip

So that: I don't have to research online to find new activities

I want to: Rate my interest in different categories of activities and receive suggestions that are tailored to my interest

GIVEN that I am on the 'Nominations' page of a certain group that I am part of

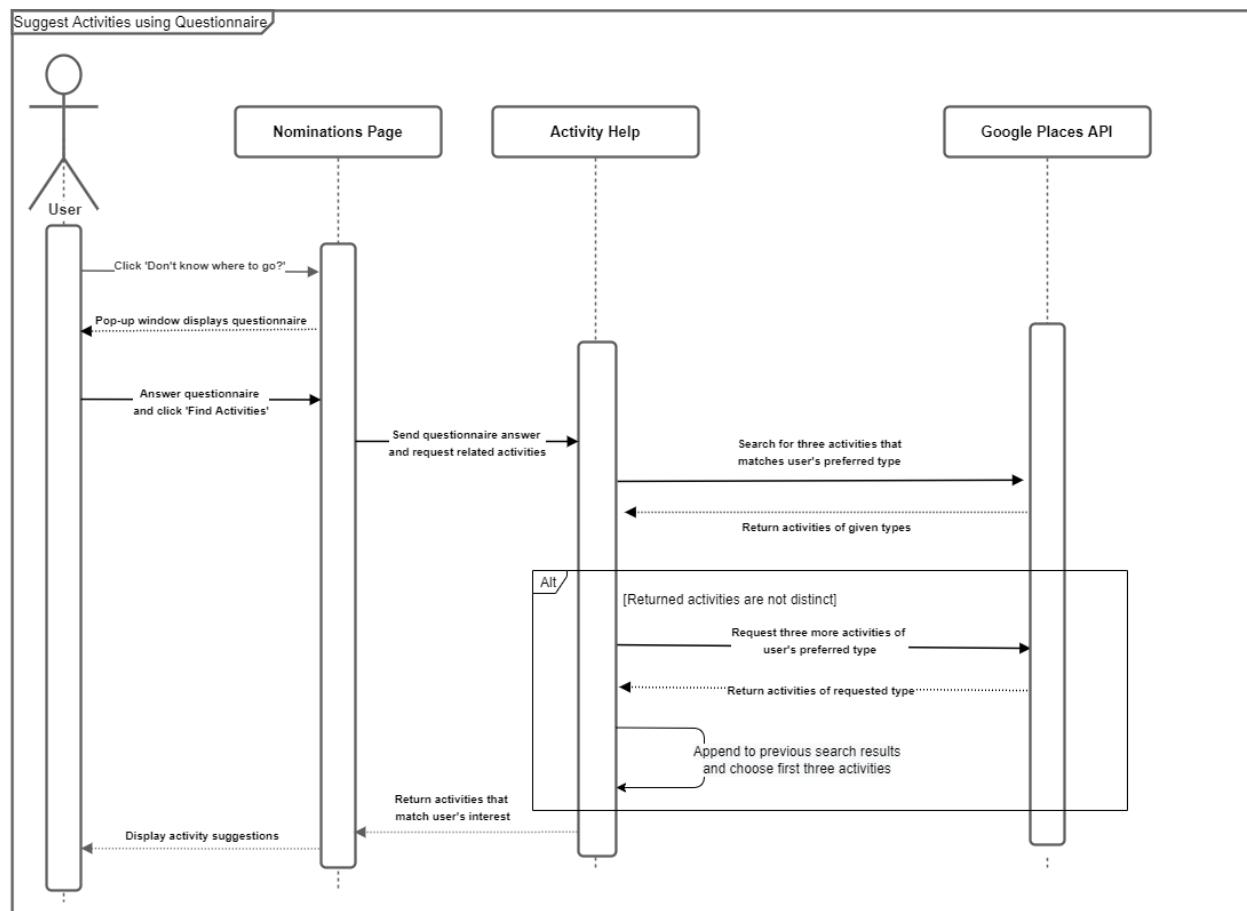
WHEN I click on 'Don't know where to go?'

THEN I should see a questionnaire pop-up asking me to rate my interest in given activity categories via a slider

AND I rate my interest by dragging the slider of my preferred types to the right, in this case, 'Sightseeing' and 'Recreation'

WHEN I click on 'Find activities'

THEN I should be able to view a three suggested activities based on my response in the questionnaire (in this case, sightseeing and recreation activities)



GET /activities/suggest

suggest three activities based on questionnaire response

Parameters

Name	Description
location string (query)	Example : Sydney Sydney
cultural integer (query)	Example : 3 3
sightseeing integer (query)	Example : 4 4
nature integer (query)	Example : 5 5
recreation integer (query)	Example : 3 3
picnic integer (query)	Example : 3 3
sport integer (query)	Example : 3 3

Responses

Code	Description	Links
200	OK	No links

Media type: application/json

Example Value | Schema

```
[{"google_places_id": "ChIJgtbEoIcfqoktS1Pw", "activity_name": "Sydney Opera House", "rating": 5, "photo_reference": "AtYBwLghLssTXRyjeRfmQX4T", "photo_content": "/31/AAQSkZ2RgABAQAAAQABAM/-0qjDhpZgAASlkqAGA...", "category": "sightseeing"}]
```

Feature: Display vote count for each activity

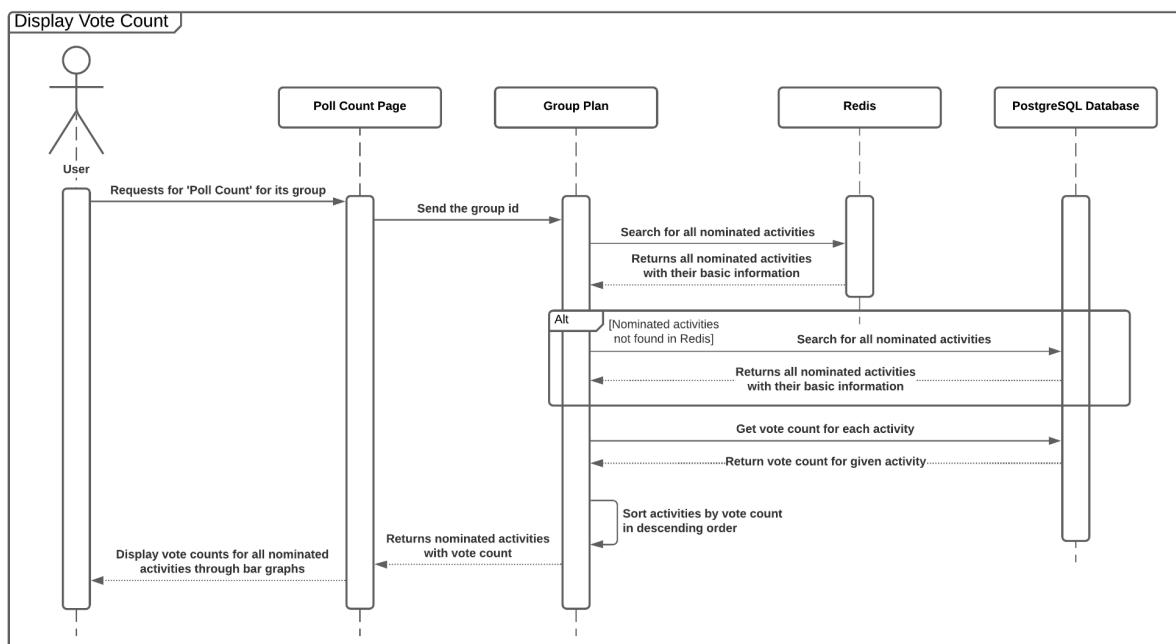
As a: Member of the group trip

So that: The group can clearly compare and identify the most popular activities (those with the most votes)

I want to: Visualise the number of votes for each activity as a bar graph, organised from most popular at the top and least popular at the bottom

GIVEN that I am on the Poll Count page of a certain group that I am part of

THEN I should see a list of activity names, each consisting of the number of votes it received, and a bar whose length is representative of the percentage of group members who voted for it, ordered from most to least popular



GET /poll/{group_id}

get poll result

Parameters

Name Description

group_id * required Example : group1_2021-04-11
string (path) group1_2021-04-11

Try it out

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{  
  "active_user": 3,  
  "activity_poll": [  
    {  
      "activity_id": 1,  
      "activity_name": "Sydney Opera House",  
      "vote_count": 2  
    }  
  ]  
}
```

Feature: Add activities from poll to a schedule

As a: Leader of the group trip

So that: I can more easily plan out the day through a visual aid

I want to: Select the most popular activities in the poll to add to the schedule

GIVEN that I am on the Poll Count page of a group that I am part of

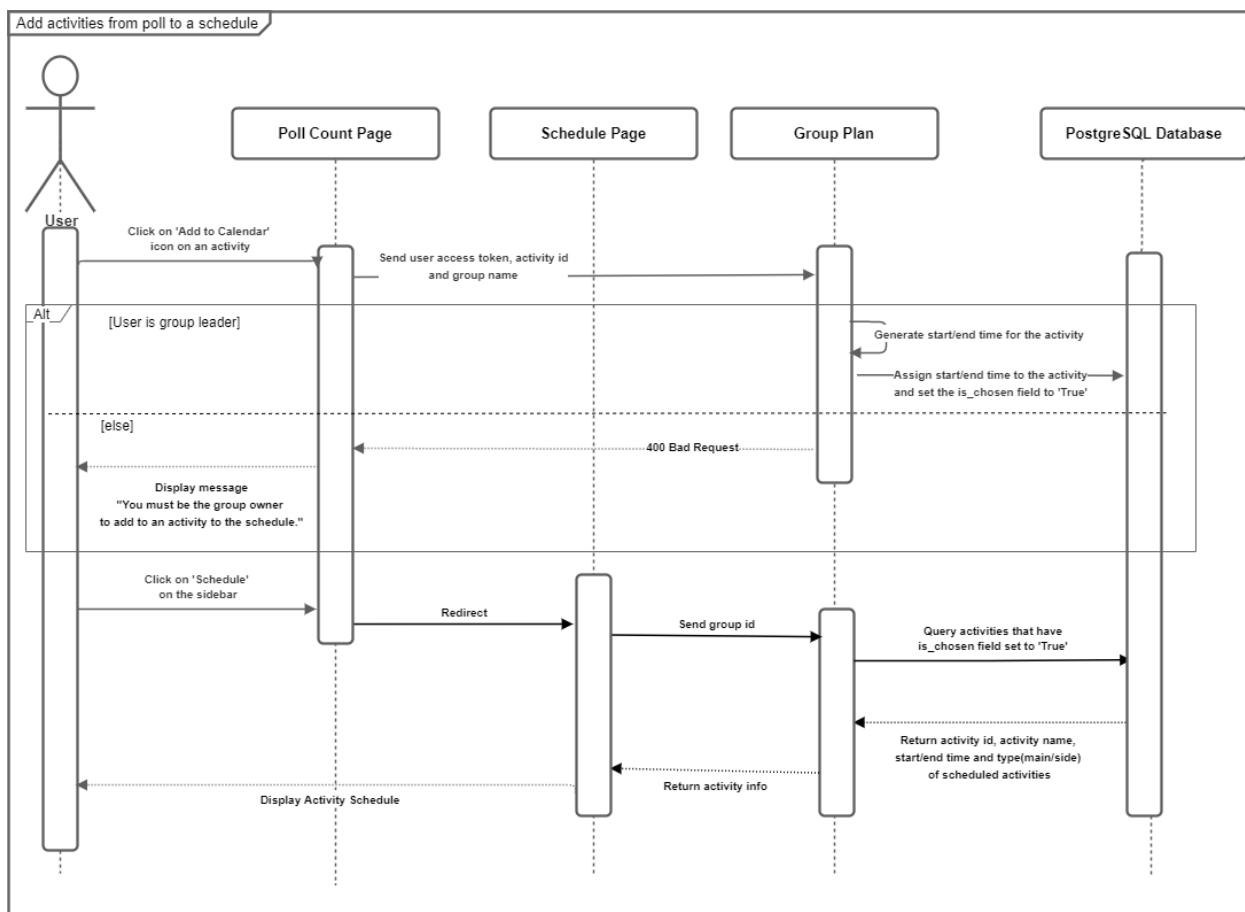
WHEN I click on the “Add to Calendar” icon on the ‘Sydney Opera House’ activity

THEN the activity is added to the schedule

WHEN I click on “Schedule” on the sidebar

THEN I am on the Schedule page of the same group

AND I can see that a ‘Sydney Opera House’ time block has been added to the group’s schedule



POST /schedule

choose activity and add to schedule

Parameters

No parameters

Request body **required**

application/json

Example Value | Schema

```
{
  "group_id": "group1_2021-04-11",
  "activity_id": 1,
  "token": "AAAGgRlb1HNCpqrahcpoAZDZ"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{}
```

GET /schedule/{group_id}

get schedule for chosen activities

Parameters

Name Description

group_id * required Example : group1_2021-04-11
string
(path) group1_2021-04-11

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "activity_id": 1,
    "activity_name": "Sydney Opera House",
    "start_time": "13:00",
    "end_time": "15:00",
    "type": "main"
  }
]
```

For the full backend API, see our Swagger documentation here:
<https://app.swaggerhub.com/apis/tanyawhy/SENG2021/0.1#/>

Part 3: Team organisation and Appraisal

Team roles and responsibilities

Team Member	Role	Responsibilities
Aadhishrii Patiil	Designer / Frontend Engineer	<ul style="list-style-type: none"> - UX/UI design - Styling web app
Enoch Luu	Frontend Engineer	<ul style="list-style-type: none"> - Setting up modals - Styling web app
Huiyi Wang	Lead Backend Engineer	<ul style="list-style-type: none"> - Implemented backend API - Integrated Google Places API - Deployed using Docker
Sanojan Thiagaraja	Lead Frontend Engineer	<ul style="list-style-type: none"> - Styling web app - Connected backend to frontend - Implemented NextAuth
Tingzhuang Zhou	Full-stack Engineer	<ul style="list-style-type: none"> - Implemented scheduling API - Styling schedule page

Reflection

Although the task was challenging, as team members had varying levels of experience, our team overcame these issues to produce a cohesive final product. We found that though the planning process was long and extensive, it was instrumental to the implementation phase as we had anticipated and resolved potential issues. In particular, we found that database design was very useful in the way we approached our backed API. Overall, the project introduced and consolidated technical and soft skills such as clear communication, providing our team a holistic experience of the software requirements and design phase during development.

What we learnt

1. Working as a team in an agile development environment which required adapting and responding to feedback provided by the mentor
2. Furthered understanding of back and front end development libraries, frameworks and languages
3. Application of design concepts in the planning phase of developing software architecture
4. Articulating ideas and plans into report writing to understand system architecture and interaction of elements
5. Relevant communication skills derived from pitching ideas to peers

Positives

- Attained additional experience in honing teamwork skills
- Promoted us to adapt quickly and learn new programming skills

Negatives

- Lack of in person communication during group meetings due to the advent of COVID-19
- Time constraints limit the scope of the project and implementation of features

Suggestions for Improvement

Although our team was satisfied with the final product, our experience highlighted several areas for improvement. We would have benefited from starting coding earlier and doing more testing as we countered a few issues that occurred during the demonstration that required addressing. This issue was further compounded by the fact that we simultaneously made pushes near the due date, making them more difficult to resolve.

Moreover, prior research into libraries and frameworks would allow for a smoother transition into the developing phase as we encountered issues connecting Auth0 to our Next.js app. In order to remedy this, we switched to NextAuth. However, we were still unable to fetch an access token from Google and thus, had to generate our own. In terms of future additions, we would like to implement an option for trips that spanned across multiple days rather than a singular day trip and provide Google calendar integration for the schedule.

References:

- <https://www.tecmint.com/why-linux-is-better-than-windows-for-servers/>
- <https://www.cleveroad.com/blog/web-application-architecture#how-does-web-app-application-work>
- <https://agilie.com/en/blog/reasons-to-use-auth0-for-authentication-in-your-application>
- https://cloud.google.com/maps-platform/?_ga=2.94143261.180287175.1616579914-1905309929.1614833556
- <https://steelkiwi.medium.com/flask-vs-django-how-to-understand-whether-you-need-a-hammer-or-a-toolbox-39b8b3a2e4a5#:~:text=other%20web%20attacks.-,Use%20cases,to%20create%20smaller%20apps%20faster.>
- <https://auth0.com/docs/protocols/protocol-oauth2>
- <https://www.postgresqltutorial.com/postgresql-vs-mysql/>
- <https://www.simplilearn.com/tutorials/sql-tutorial/postgresql-vs-mysql>
- <https://www.stitchdata.com/resources/postgresql-vs-mysql/>
- <https://medium.com/swlh/server-side-rendering-with-next-js-56f84f98f9bd>