UNSW
AUSTRALIA

SENG2021 - Requirements and Design Workshop
Team Mongoose

Deliverable 2

2021,Term 1

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

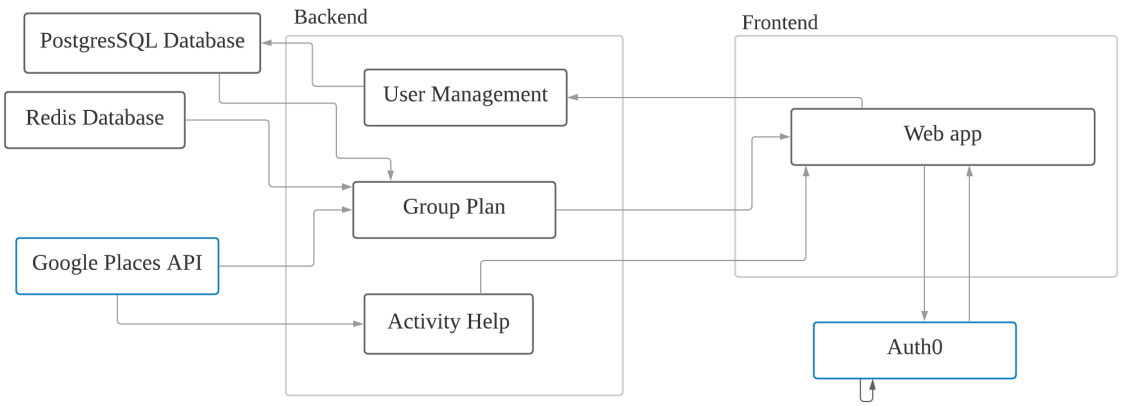
Part 1: Software Architecture

External Data Sources

We will be using 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.

The **Auth0** API will provide an authentication and authorisation platform for our web app. We believe that this will make it both easier for us as developers as well as the end-user. As developers, we can avoid having to implement authentication manually instead focusing our time and effort on the distinguishing features of our project. It also shrinks the size of our database hence reducing request/query time to boost performance.

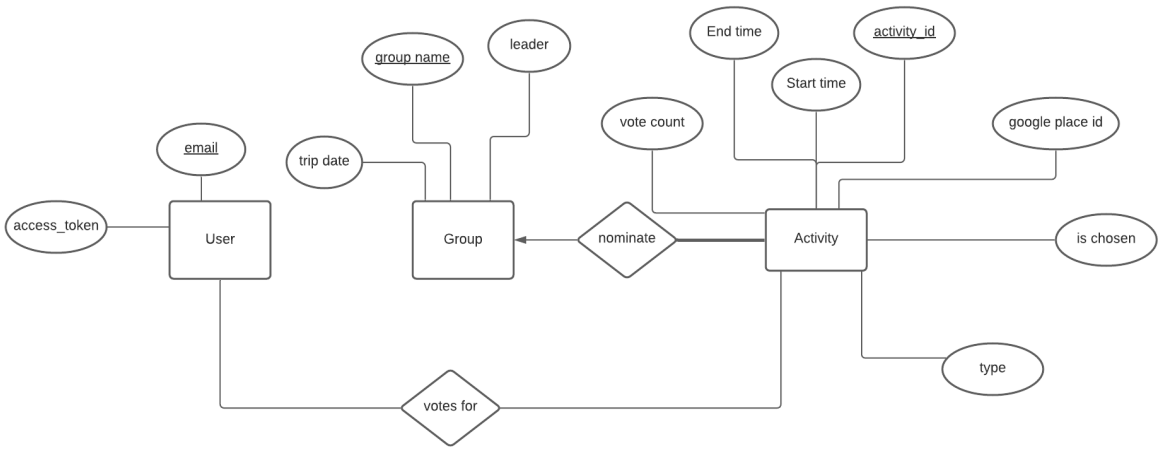
Furthermore, the end-user will have the convenience of securely logging in through their existing social accounts such as Google, Twitter, Github, (securely through OAuth 2.0 Authorisation) instead of having to create a new account with our app. In addition, since Auth0 only returns an access token, the user cannot be identified on the frontend, ensuring anonymity in the voting and nominating process.

The Auth0 API links to our '**User Management**' component in our backend. This handles storing user emails in the database and updating the user's access token after expiry. Doing this ensures that users are authorised when using the web app.

Our '**Group Trip'** component consists of the key functions that group members should be able to do: nominate and vote for activities, view the poll results, and schedule the activities. This information needs to be stored persistently in the database. When it needs to display the activities on the frontend, it will make requests to the **Google Places API** (discussed in previous section) which provides information like images, location name and rating. The Google Place API is an effective way of retrieving existing online resources from an information archive.

Our **'Activity Help'** component consists of secondary functions that group members can use to inform their decisions. This includes the questionnaire which asks users for their interest, as well as the activity details page to find out more about a specific activity. Both these features will make a request to the Google Places API and receive information regarding specific places.

The **'PostgreSQL Database'** component stores information that is specific to each group. This includes the group name, trip date, as well as the activities that are chosen (and the votes from each user).



ER Diagram for PostgreSQL Database

The **'Redis Database'** component stores the basic information for all activities, including activity name, rating 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 and we don't have to send requests to Google Places API for this information repetitively. We will use the hash data structure supported by Redis for the information storage. Concatenating group name and activity id as the hash key and each key associated with field name, rating and photo_reference allows us to quickly retrieve the information needed for displaying activities on the "Nomination" page for each group.

Language/Framework Choices

The web stack of our software:

- Frontend Tier: **HTML, React JS, Styled-Components**
- Backend Tier: **Python, Flask**
- API Tier: **Google Places API, Auth0**
- Data Tier: **PostgreSQL, Redis**

We have decided to build 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. We will also use the **Styled-Components** react library, which allows us to write CSS in Javascript files. This CSS-in-JS solution makes styling easier to manage than traditional CSS stylesheets.

Python will be used as the language for our backend components as it is easy to use and all team members are proficient in it. Furthermore, we will use **Flask,** one of the leading Python web development frameworks. We chose Flask over the popular Python framework Django as Flask 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, it is only partially SQL compliant, which can make the syntax 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 instead of 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 can then send a request to the Google Places API for the information as all activities and their Google Places IDs 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.

Platform Choice

The choice of server distribution will be Linux on Amazon AWS AMI. It is highly secure, stable and reliable. Linux is also the most secure existing kernel.
In terms of browser platforms, we will be aiming to ensure compatibility with the most popular web browsers such as Google Chrome and Firefox. In particular, the user experience will be optimised for desktop/laptop-sized devices as we believe the research-focused nature of our app would be better suited to a user base on desktop/laptop computers as opposed to mobile devices. This strategy will be more feasible given the time we have to implement our web app.

Summary:

From the discussions and negotiations of the selection of language and frameworks, the method for communicating with API, the construction/refinement of database, and the MVP test for compatibility, this has led us to a software architecture design with high cohesion and low coupling.

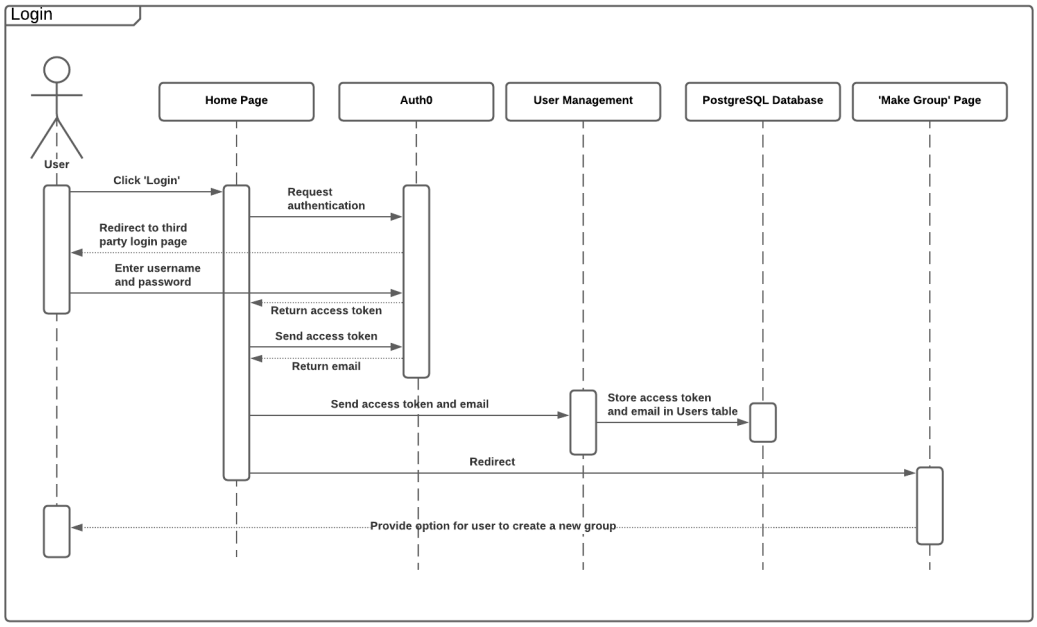
Part 2: Initial Software Design

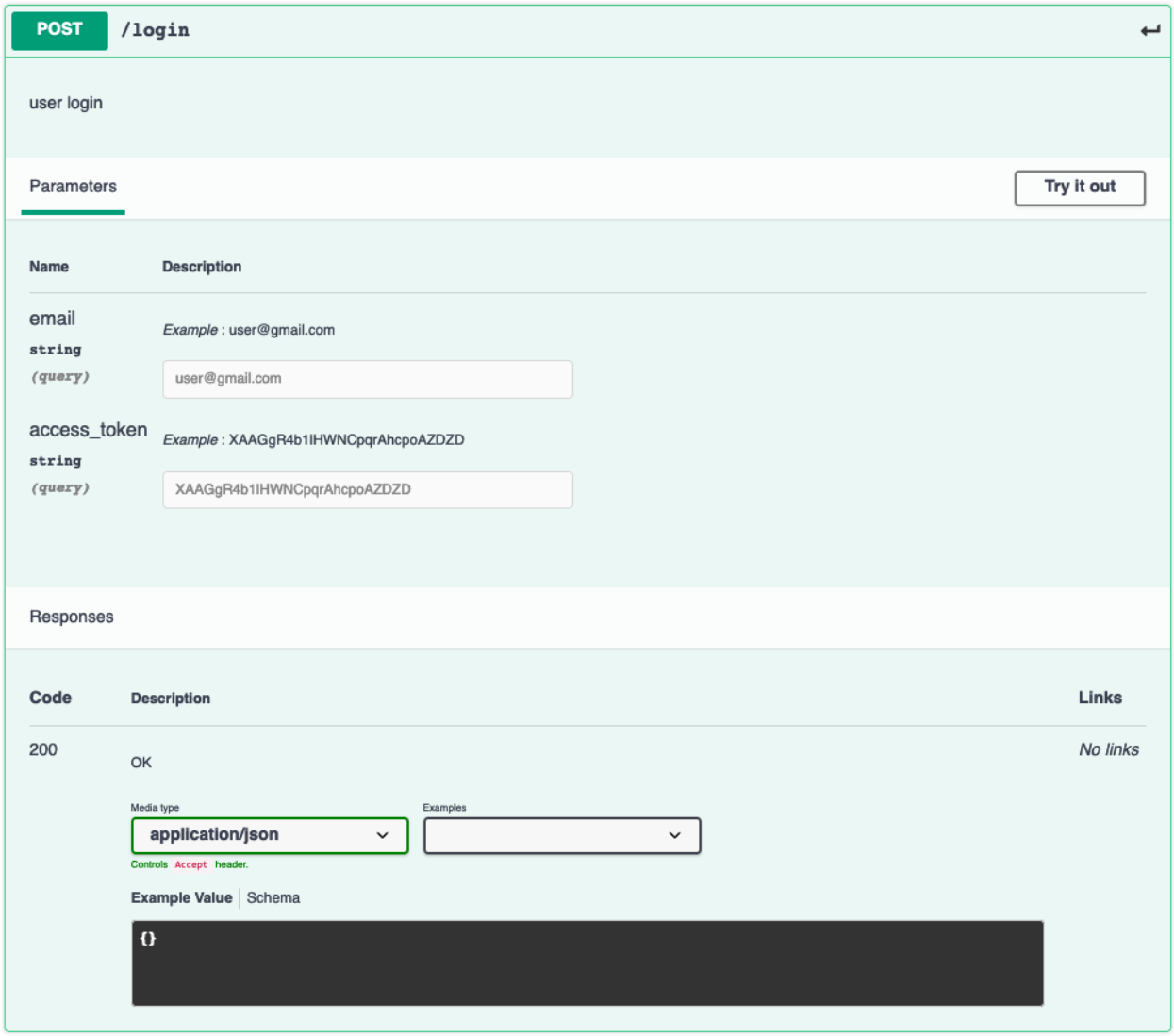
User Stories with Sequence Diagrams

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 dialog box offering multiple options to login with popular social network providers
WHEN I click on 'Login with Facebook'
AND enter my Facebook email and password
THEN I should be logged in
AND see an option to create a new group

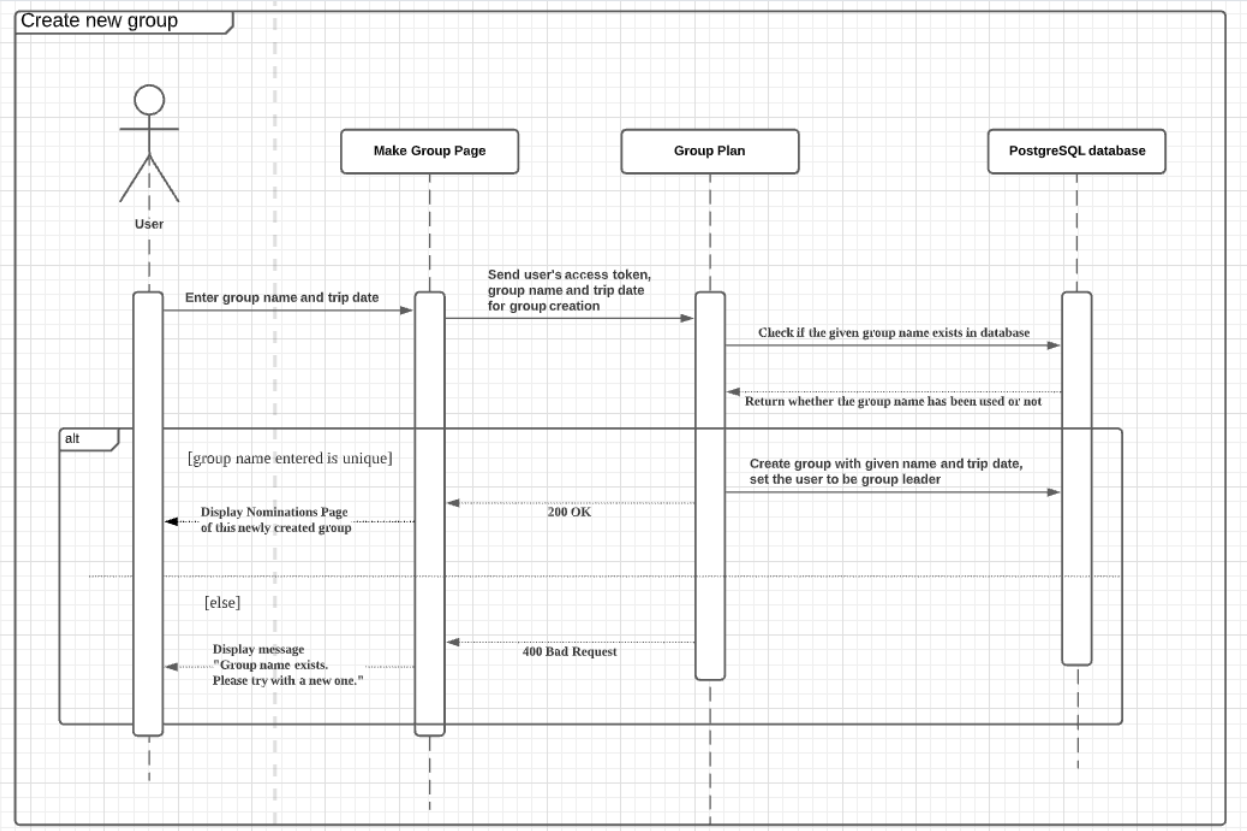


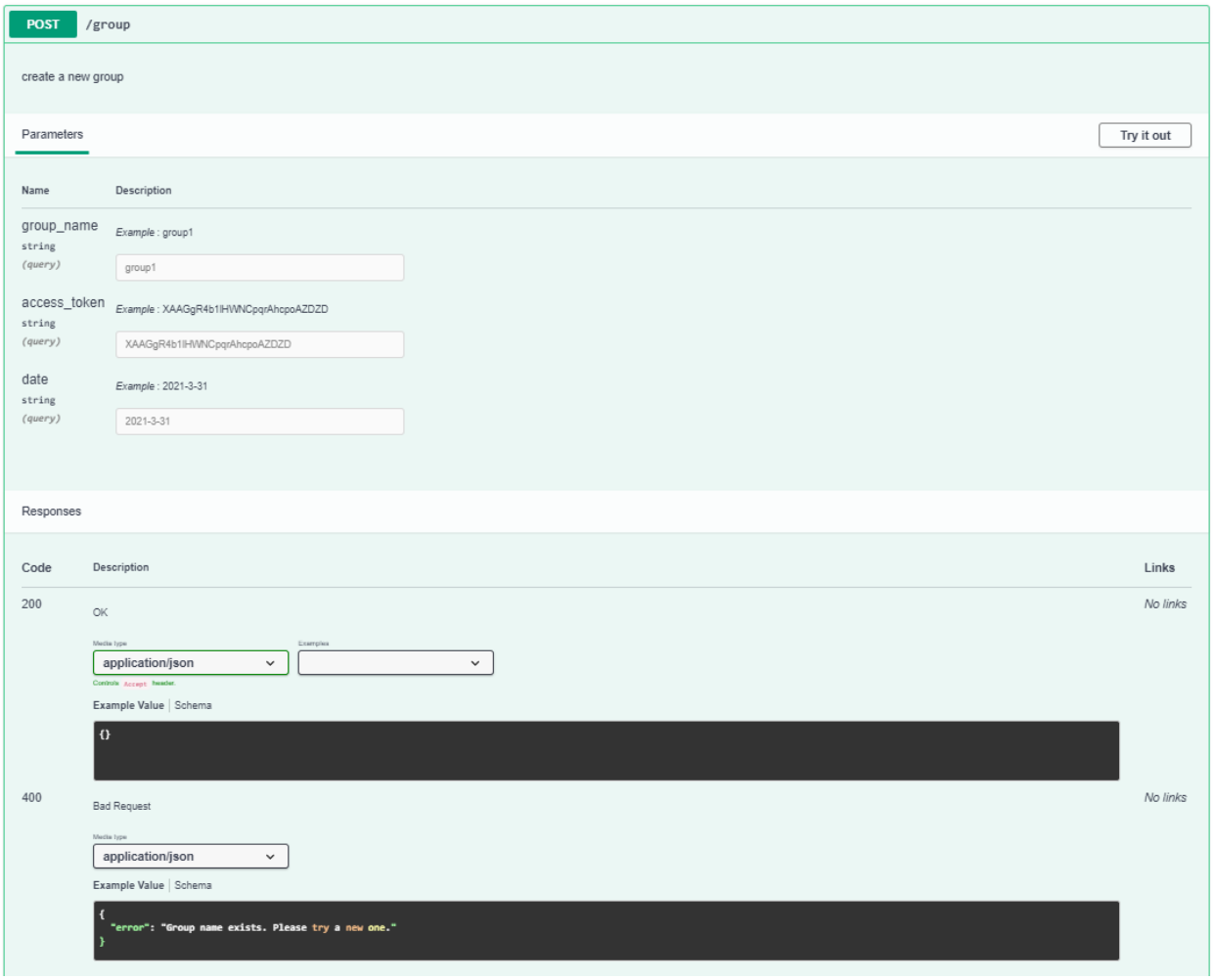

POST
/login
user login
Parameters
Try it out
Name
Description
email
string
(query)
Example : user@gmail.com
user@gmail.com
access_token
string
(query)
Example : XAAGgR4b1lHWNCpqrAhcpoAZDZD
XAAGgR4b1lHWNCpqrAhcpoAZDZD
Responses
Code
Description
Links
200
OK
No links
Media type
application/json
Examples
Controls 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

GIVEN that I am on the 'Make Group' page of the web app after logging in
WHEN I enter a unique group name and date of the trip
THEN I should be on a blank 'Nominations' page for the newly created group

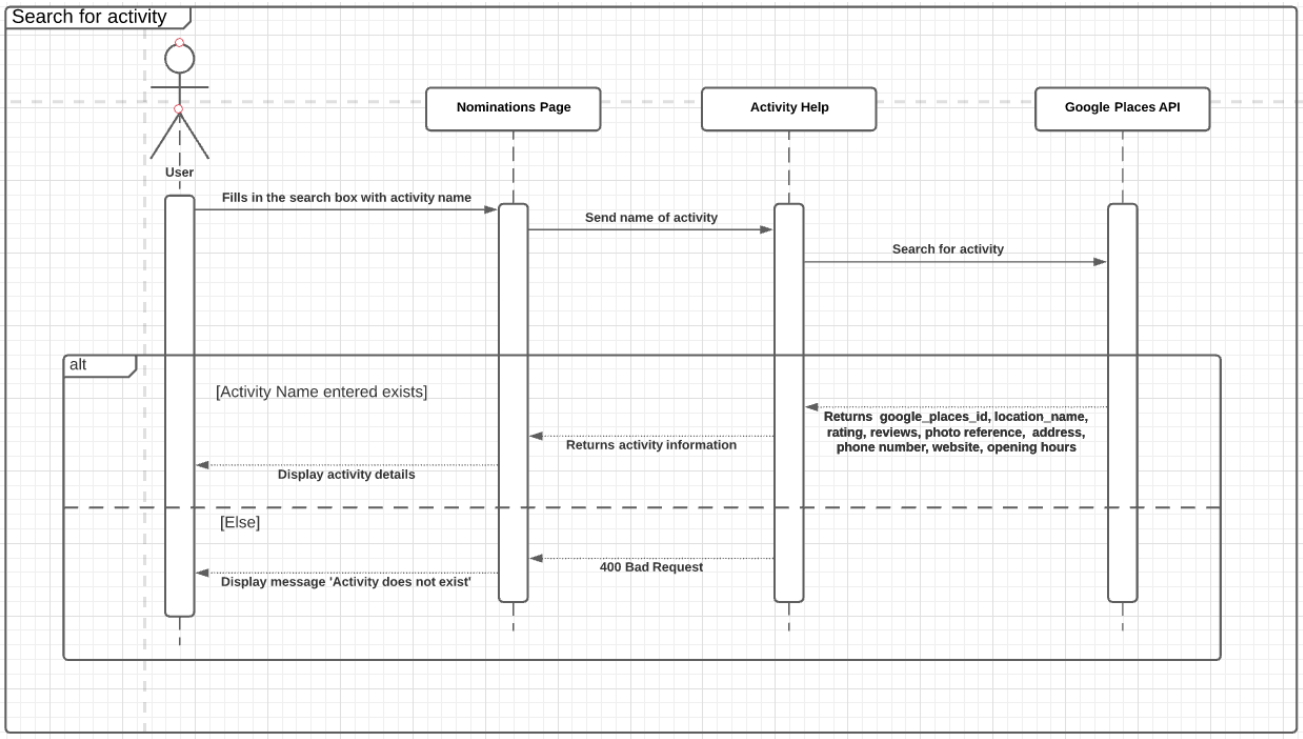


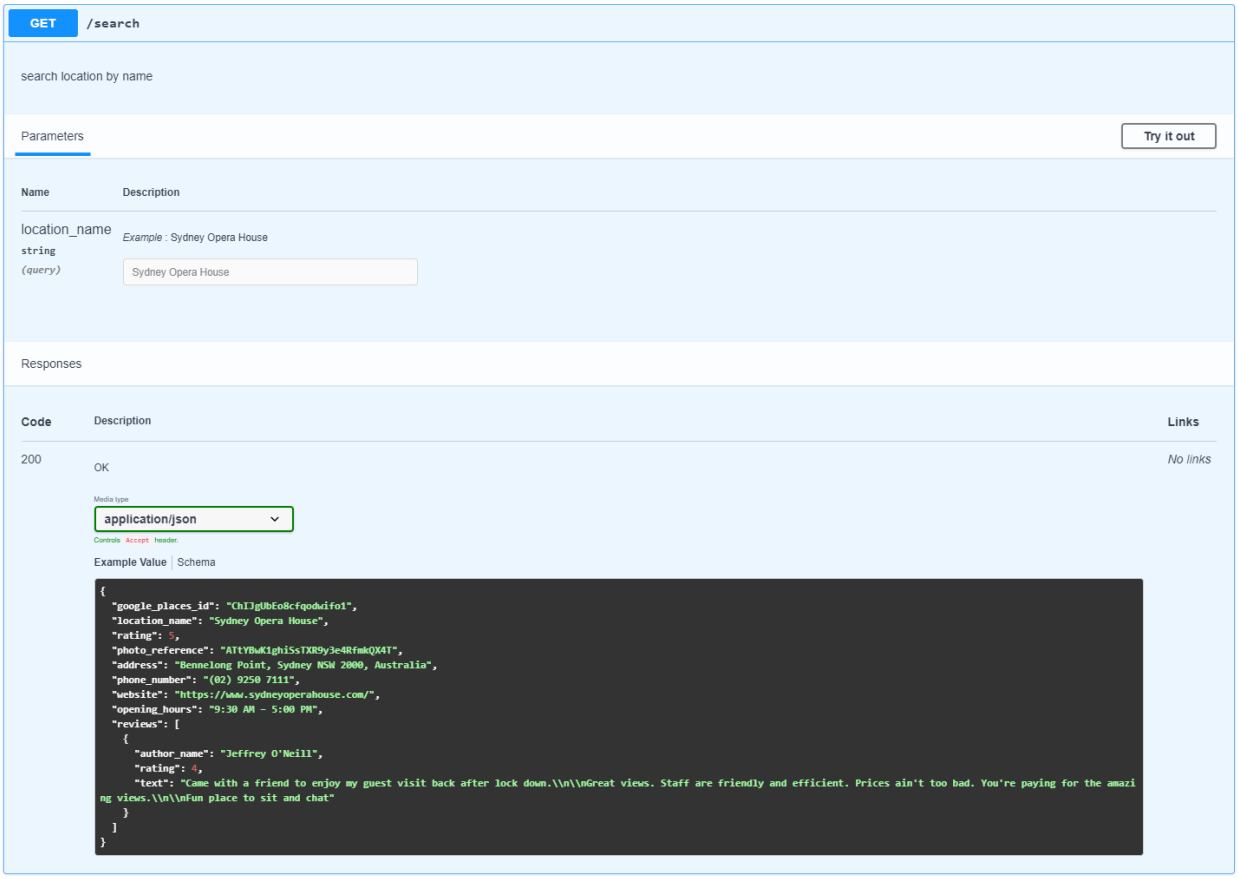

POST
/group
create a new group
Parameters
Try it out
Name
Description
group_name
string
(query)
Example : group1
group1
access_token
string
(query)
Example : XAAGgR4b1lHWNCpqrAhcpoAZDZD
XAAGgR4b1lHWNCpqrAhcpoAZDZD
date
string
(query)
Example : 2021-3-31
2021-3-31
Responses
Code
Description
Links
200
OK
No links
application/json
Example Value | Schema
{}
400
Bad Request
No links
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

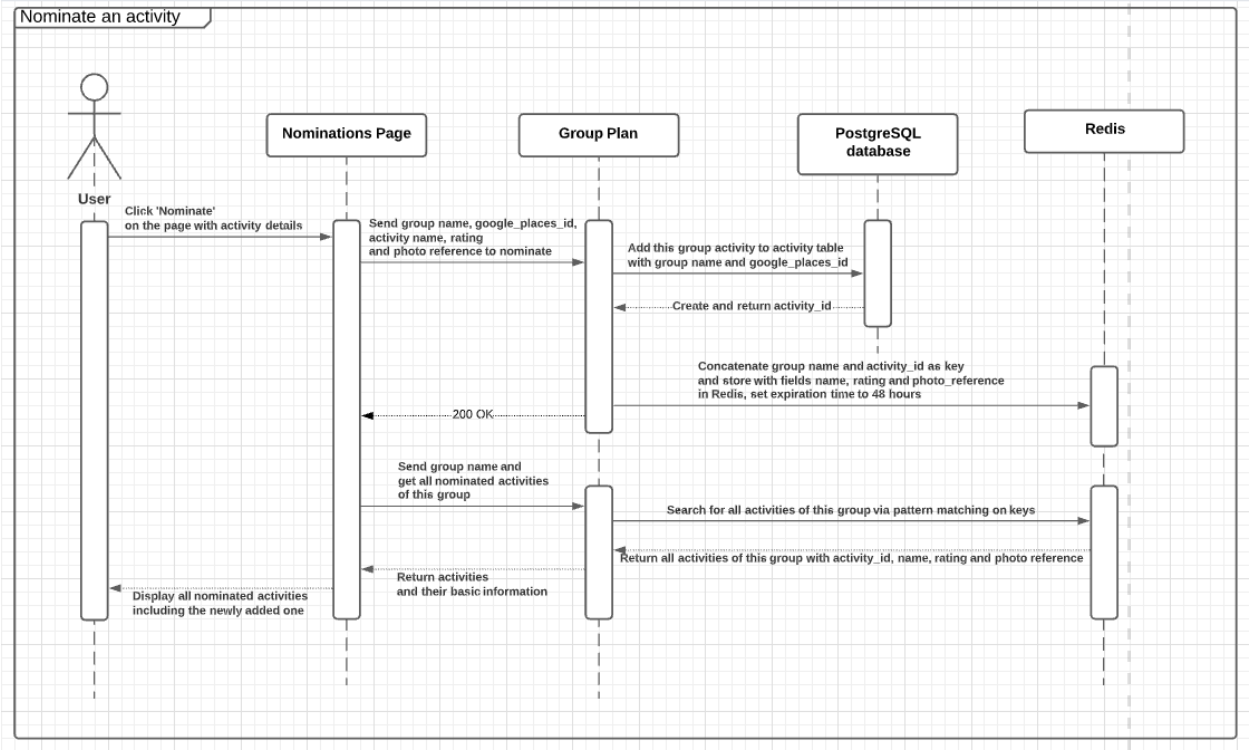


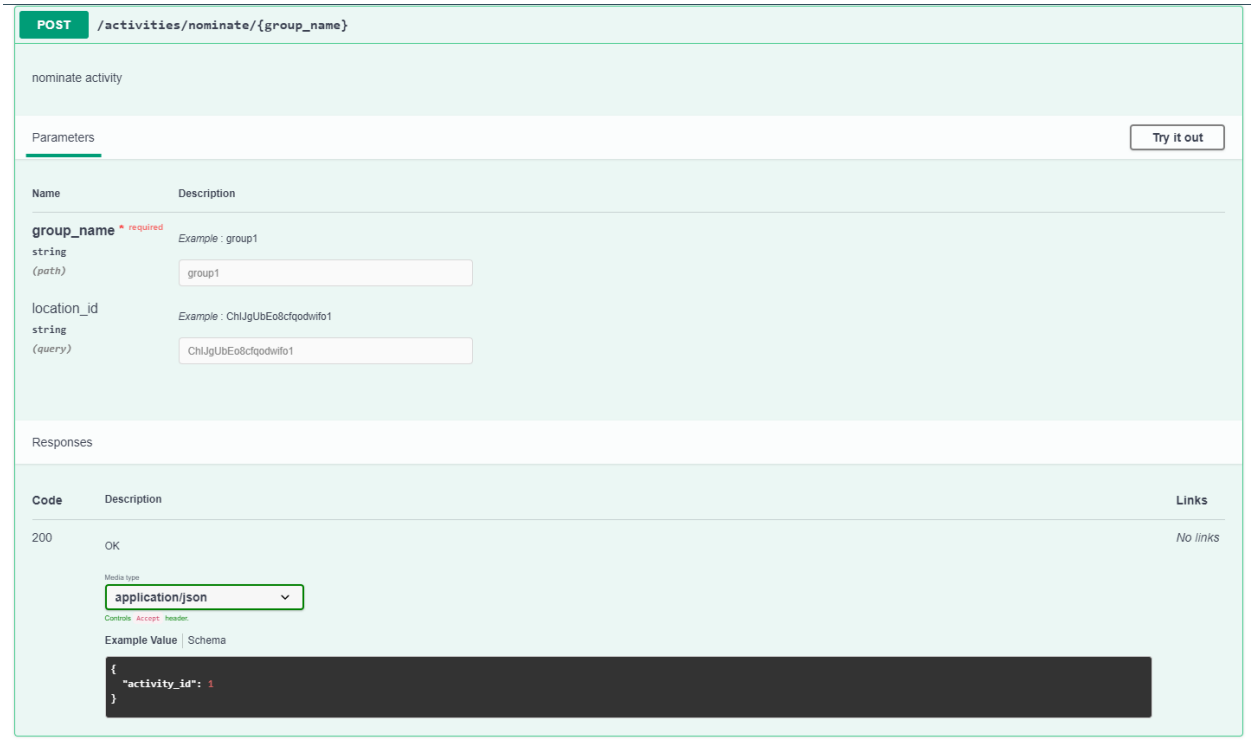

GET
/search
search location by name
Parameters
Try it out
Name
Description
location_name
string
(query)
Example : Sydney Opera House
Sydney Opera House
Responses
Code
Description
Links
200
OK
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
{
"google_places_id": "ChIJgUbEo8cfqodwifo1",
"location_name": "Sydney Opera House",
"rating": 5,
"photo_reference": "AItYBwK1ghiSsTXR9y3e4RfmkQX4T",
"address": "Bennelong Point, Sydney NSW 2000, Australia",
"phone_number": "(02) 9250 7111",
"website": "https://www.sydneyoperahouse.com/",
"opening_hours": "9:30 AM – 5:00 PM",
"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 amazi
ng views.\\n\\nFun place to sit and chat"
}
]
}

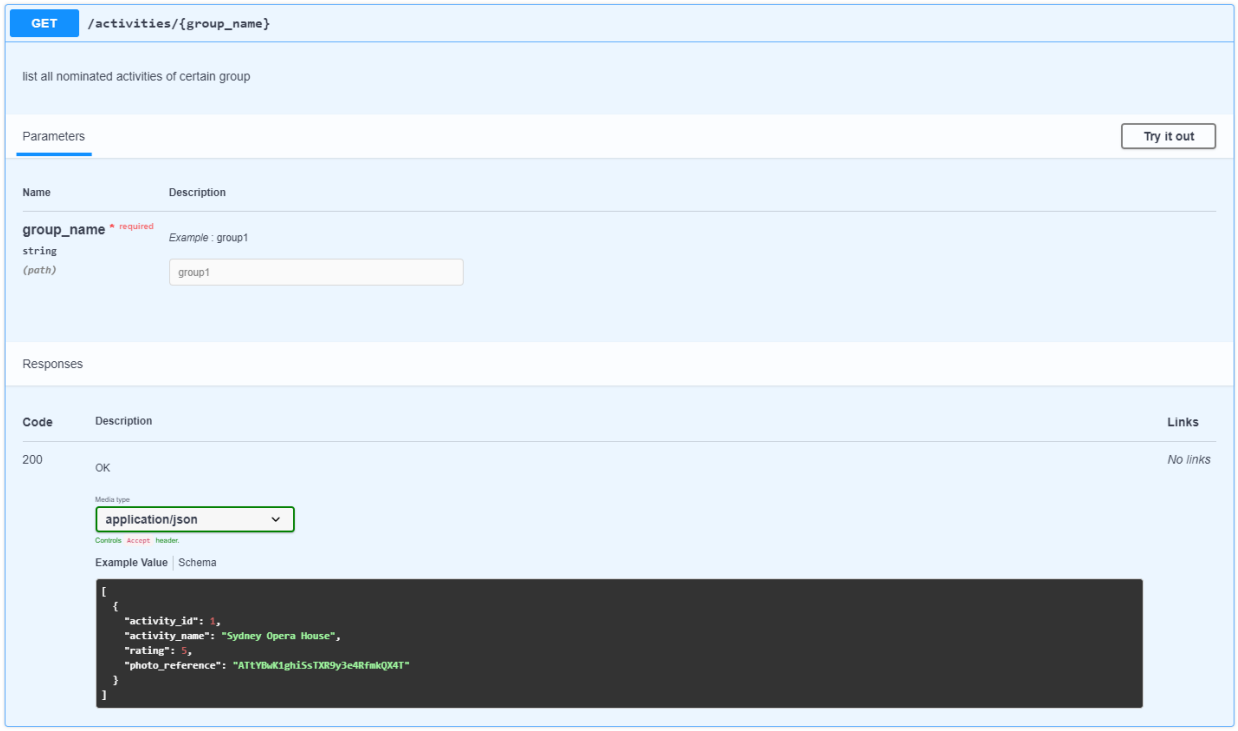
Feature: Nominate an activity

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 should go back to the 'Nominations' page
AND I can see an activity card titled "Sydney Opera House" on the list of nominations



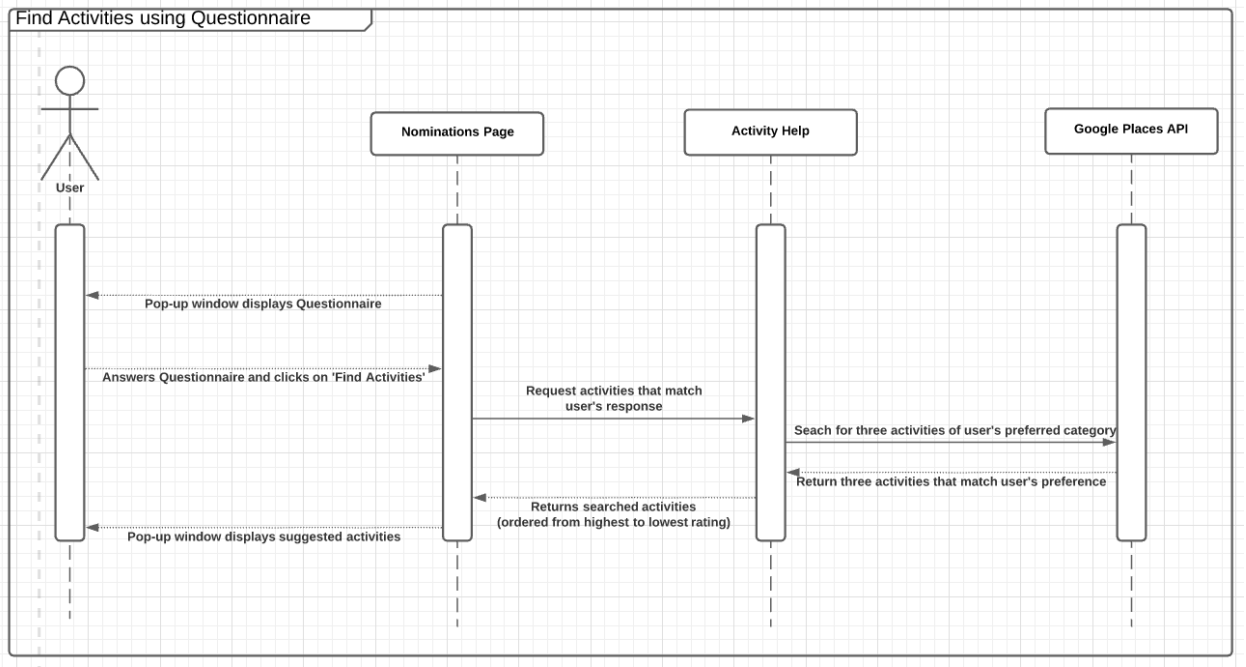

POST
/activities/nominate/{group_name}
nominate activity
Parameters
Try it out
Name
Description
group_name * required
string
(path)
Example : group1
group1
location_id
string
(query)
Example : ChIJgUbEo8cfqodwifo1
ChIJgUbEo8cfqodwifo1
Responses
Code
Description
Links
200
OK
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
{
"activity_id": 1
}

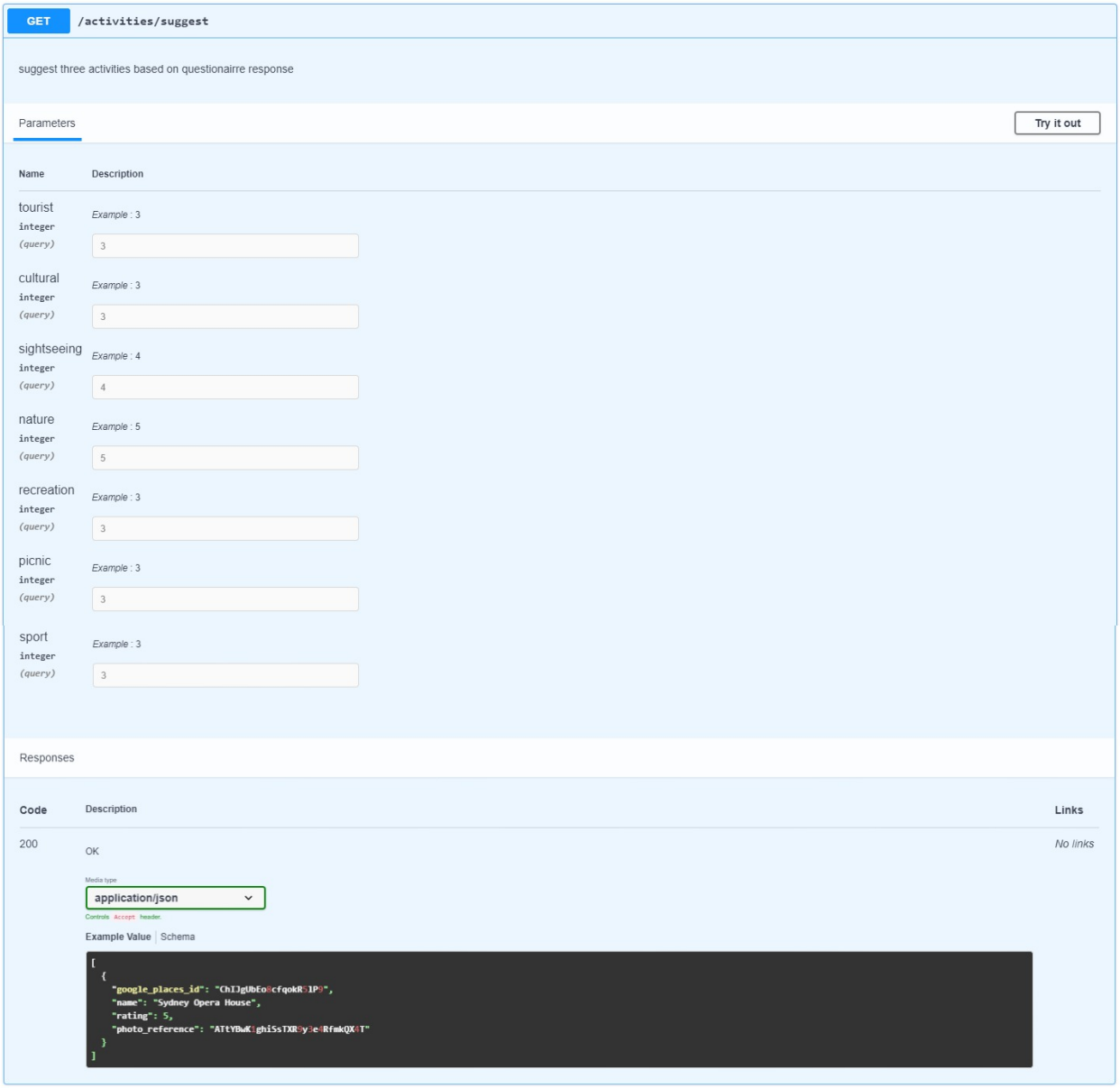

GET
/activities/{group_name}
list all nominated activities of certain group
Parameters
Try it out
Name
Description
group_name * required
string
(path)
Example : group1
group1
Responses
Code
Description
Links
200
OK
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
[
{
"activity_id": 1,
"activity_name": "Sydney Opera House",
"rating": 5,
"photo_reference": "ATtYBwK1ghiSsTXR9y3e4RfmkQX4T"
}
]

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
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 from 1 to 5
WHEN I submit my rating of interest for each activity category such as 3 for "Nature", 5 for "Recreation" and 1 for the other categories
THEN I should be able to view a list of suggested activities based on my response in the questionnaire (in this case, likely 2 recreation and 1 nature activity)

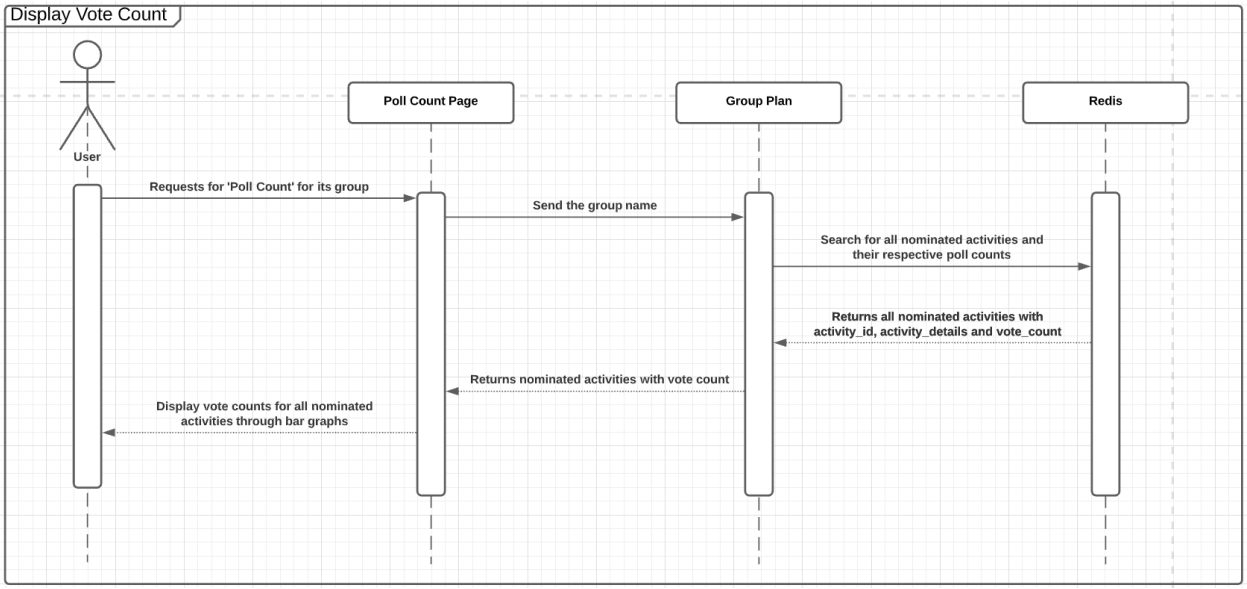


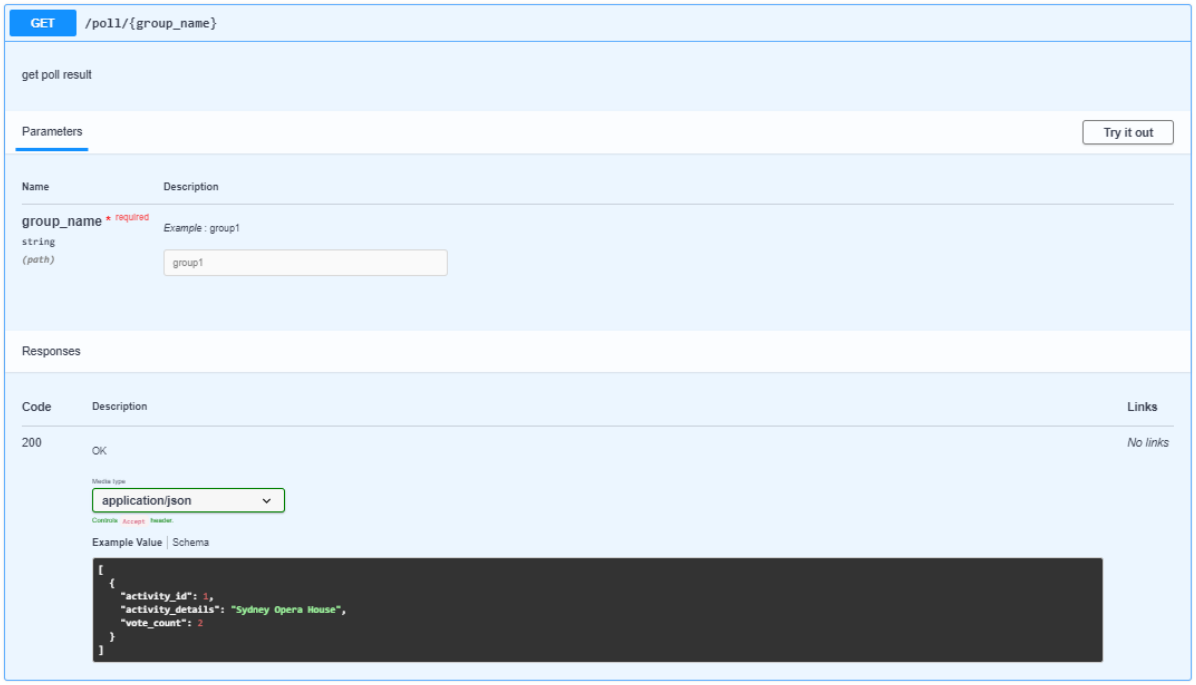

GET
/activities/suggest
suggest three activities based on questionairre response
Parameters
Try it out
Name
Description
tourist
integer
(query)
Example : 3
3
cultural
integer
(query)
Example : 3
3
sightseeing
integer
(query)
Example : 4
4
nature
integer
(query)
Example : 5
5
recreation
integer
(query)
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
Controls Accept header.
Example Value
Schema
[
{
"google_places_id": "ChIJgUbEo
cfqokR
1P
",
"name": "Sydney Opera House",
"rating": 5,
"photo_reference": "ATtYBwK
ghiSsTXR
y
e
RfakQX
T"
}
]

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 group that I am part of
THEN I should see a list of activity names and bars whose length is indicative of the number of the votes for that activity

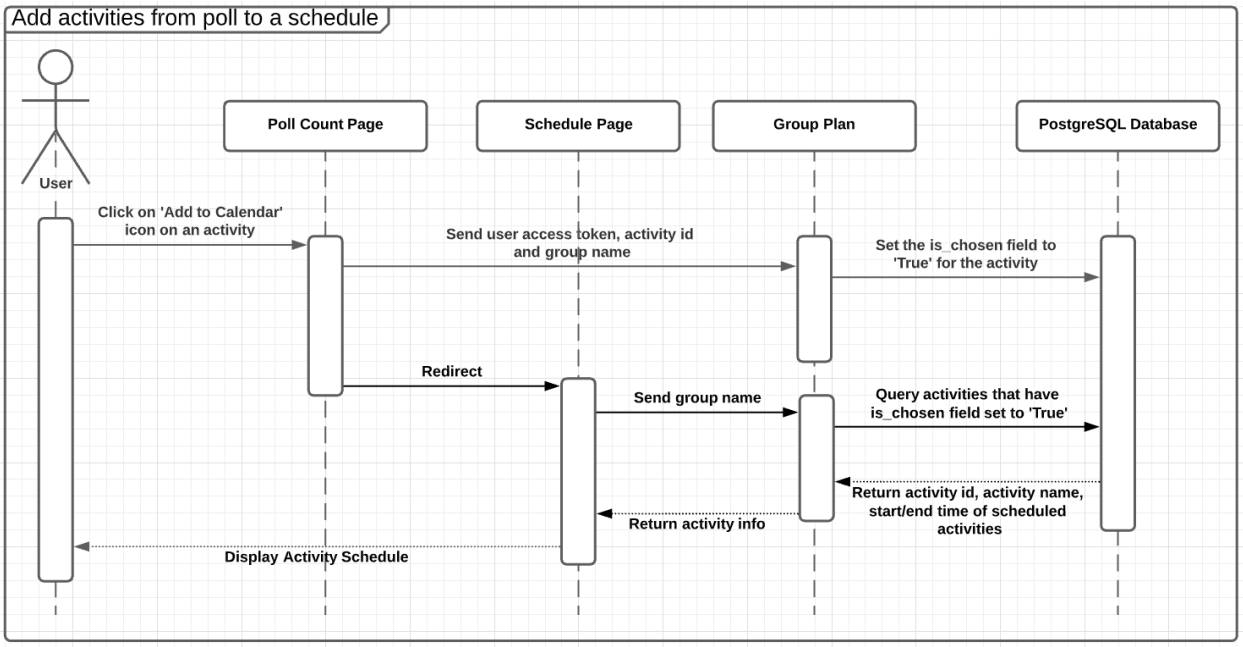


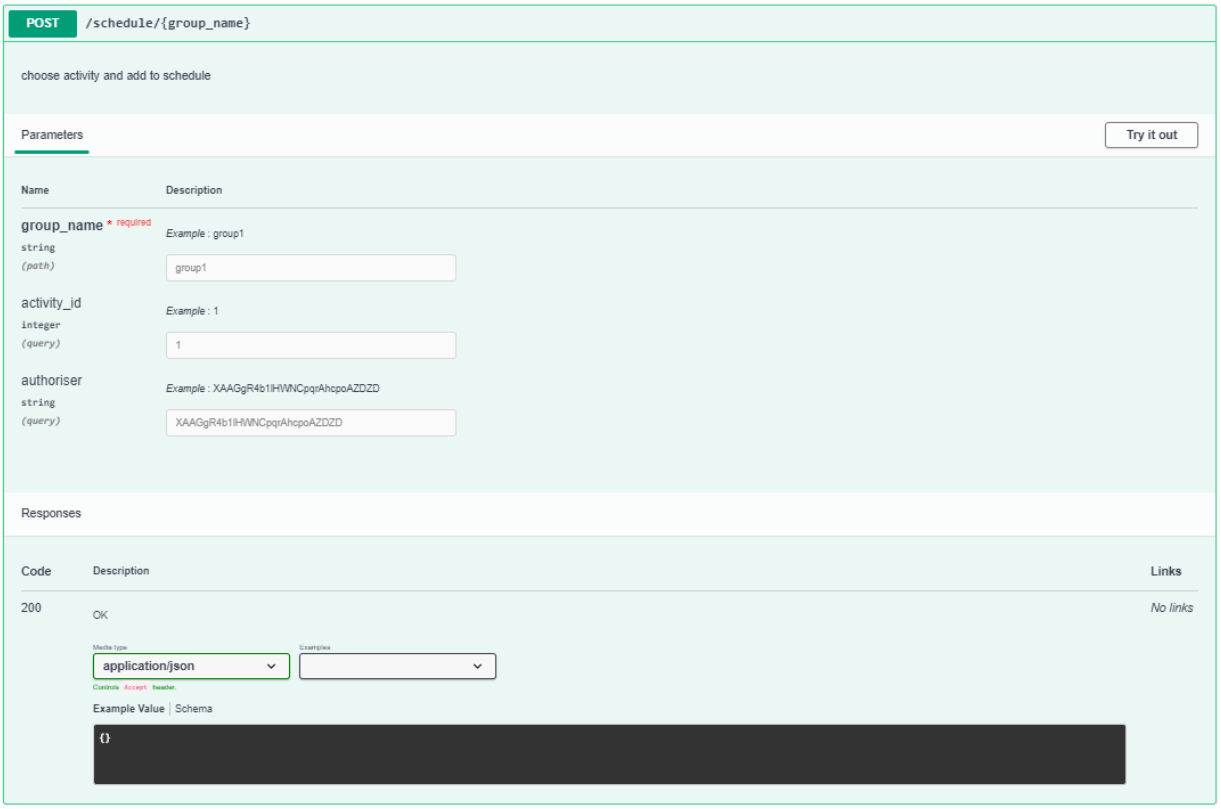

GET
/poll/{group_name}
get poll result
Parameters
Try it out
Name
Description
group_name * required
string
(path)
Example : group1
group1
Responses
Code
Description
Links
200
OK
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
[
{
"activity_id": 1,
"activity_details": "Sydney Opera House",
"vote_count": 2
}
]

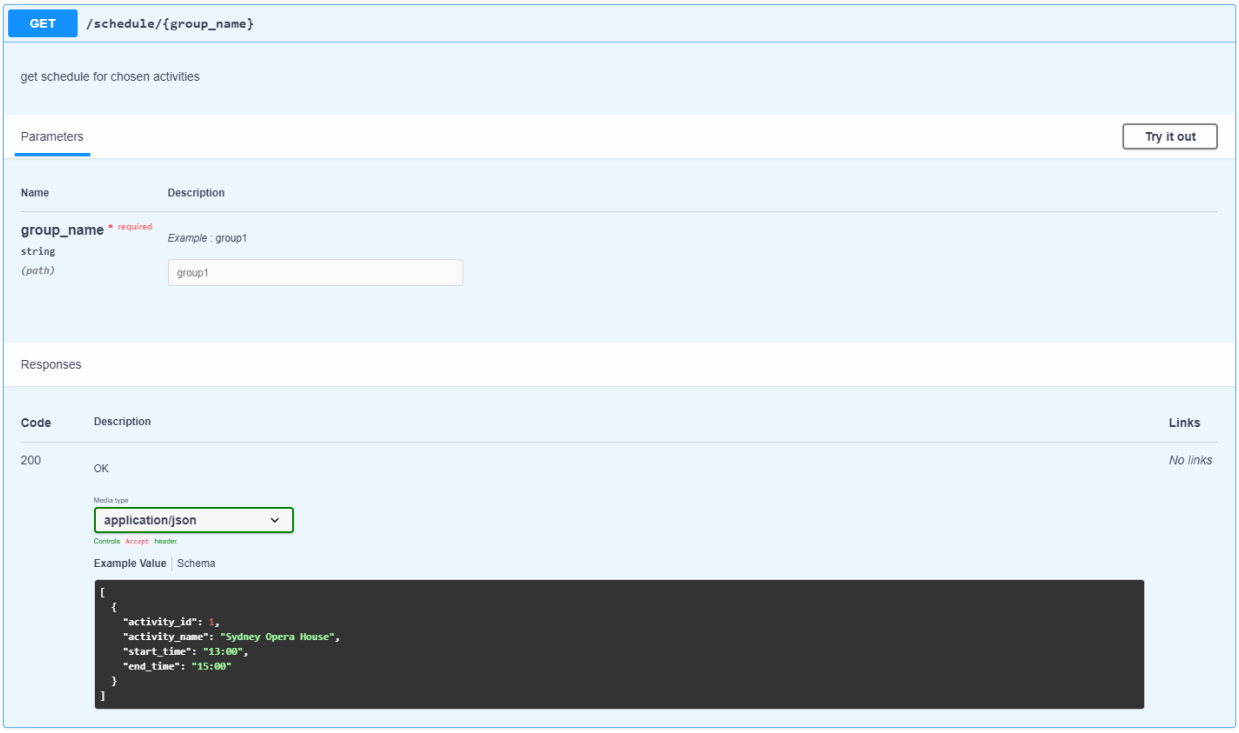
Feature: Add activities from poll to a schedule

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 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/{group_name}
choose activity and add to schedule
Parameters
Try it out
Name
Description
group_name * required
string
(path)
Example : group1
group1
activity_id
integer
(query)
Example : 1
1
authoriser
string
(query)
Example : XAAGgR4b1lHWNCpqrAhcpoAZDZD
XAAGgR4b1lHWNCpqrAhcpoAZDZD
Responses
Code
Description
Links
200
OK
No links
Media type
application/json
Controls Accept header.
Examples
Example Value | Schema
{}


GET
/schedule/{group_name}
get schedule for chosen activities
Parameters
Try it out
Name
Description
group_name * required
string
(path)
Example : group1
group1
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"
}
]

See our Swagger documentation here:
https://app.swaggerhub.com/apis/tanyawhy/SENG2021/0.1#/

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/