



PlusOne

Easy planning, Exciting adventures

Deliverable 5

Mentor: Ali Darejeh

Team: random_group

Stephen Bletsas (z5257110)

Faris Jalal (z5262405)

Vishnu Birudavolu (z5280782)

Rushil Pardasani (z5266512)

Thiagaut Valcke (z5282625)

Table of Contents

Executive Summary	3
Project Context	4
Competitor Analysis	4
Purpose of our system	5
Problem Statements	5
Features of Our System	6
User Stories	7
Frontend	27
Flutter	27
Backend	28
Firebase Cloud Firestore	28
Flask	28
Yelp API	29
Ticketmaster API	29
Facebook Login	29
Google Login	29
Google Maps API	30
OpenWeather API	30
Testing	31
Advanced REST client	31
Summary of Key Benefits and Design Implementation	41
Team Organization and Conclusion	42
Team Roles	42
Communications	42
Tools for Project Management	43
Reflection	44

Executive Summary

PlusOne provides a user-friendly mobile application that aims to address the problems left by the COVID pandemic of a poor social and economic climate, by automating the tedious process of organising events and promoting socialising and recreational activities. Under the guidance of Ali Darejeh, PlusOne was designed by team random_group as part of the SENG2021 21T1 course. This report provides further details of the business value, purpose, creation and technical design of the mobile app. Source code can be viewed at:

[https://github.com/farisjalal/SENG2021_random_group.](https://github.com/farisjalal/SENG2021_random_group)

Project Context

The world has come to a social standstill over the last year during the pandemic, driving people into their homes and increasing physical isolation. A recent study in Australia stated that more than 50% of people reported feeling lonely during the pandemic. Government closures and restrictions on all bars, clubs, cafes, restaurants and entertainment venues have led to major economic downturn for all these industries; with revenue falling for pubs, bars and nightclubs by 18.1% in 2019-20, with a further decline of 16.7% in 2020-21. Restaurants were hit hard with a slump of 15%, with the hardest hit industry being theatres only achieving 25% of their 2019 revenues in 2020.

Thankfully, we are reaching a visible end to the pandemic in Australia with normality slowly coming to prominence over recent months. People are becoming increasingly comfortable having social gatherings and events. New trends have started to emerge as a result of the government lifting restrictions, with one of them being the popularity of the “mystery experience”. Qantas have developed a new “mystery flight” program to encourage Australians to travel interstate for an adventure in a surprise location. The popularity of this trend is evident, with flights being “sold out in just 15 minutes” (Guardian, 2021).

Competitor Analysis

Applications such as **EventBrite** are on a similar line but do not provide users with a full fledged plan for an outing and are mostly used for booking events and do not make reservations in restaurants or book movie tickets. In addition Eventbrite does not organise sequences of events according to preferences.

Ticketebo is another event organiser application, however it does not include a chat function, to let you organise events with friends and does not display weather data to recommend you the best event. Similar to Eventbrite, Ticketebo does not organise a series of events according to your liking.

'I'm In' is a competitor who is focused on organising events for groups. The application lets you create groups, view upcoming events and existing ones in a list and on a map. However this application does not let you book restaurant, movie or concert events. In addition 'I'm in' has in-app advertisements, while our application is completely ad-free (with plans on bringing in revenue through event promotion).

Purpose of our system

Current social trends coupled with the effects of the coronavirus pandemic have left people feeling isolated and alone as a consequence of lockdowns and social-distancing regulations. This has resulted in fewer people visiting small businesses, tourist attractions, entertainment and hospitality (restaurants, bars, clubs, etc.) venues. As we progress, people are becoming increasingly comfortable to have social gatherings but a growth in the trend of working at home, many people find it increasingly difficult to switch off. With a significant amount of the population having very busy schedules, making organising an outing an unnecessary hassle. Having to meticulously create a gathering by considering others' interests can be cumbersome and occasionally frustrating, making the outing itself less enjoyable and may dissuade people from going out in the first place.

To address these problems, we propose an application that automatically develops a fun and exciting recreational agenda for users based on their preferences to make going out with friends and family a simple, enjoyable and exciting experience.

Problem Statements

- Current social trends coupled with the effects of the coronavirus pandemic have left people feeling isolated and alone.
- Due to the pandemic fewer people are visiting small businesses, tourist attractions, and entertainment related venues.
- A significant amount of the population have very busy schedules, making organising an outing an unnecessary hassle.
- Having to manually take into consideration all of one's friends' interests when planning an outing is cumbersome and sometimes frustrating.
- Having to meticulously organise an outing can make the outing itself less enjoyable and dissuade people from going out in the first place.

Features of Our System

To differentiate itself PlusOne features include its ease of usability, event plan automation, venue details, past-plans and a chat system.

To provide a high-quality customer experience, PlusOne's user interface has been designed with ease of use in mind and satisfying mobile UX design principles content prioritisation, intuitive navigation, legible text, visible elements, hand position controls and many more to create a seamless experience.

The mobile application allows users to create an event by inputting filters and preferences, which accesses our ever-updating database to automatically generate a plan consisting a series of events, automating the tedious process of event planning and management.

Within each plan, a user can select a specific event to gather greater details of that venue/location. Details will include the location (provided with a map), weather at location and a small summary of the event.

Users can access past plans with the history feature, sorted initially by new-to-old, so they can have the ability to reorganise a new similar plan, recommend venues to friends or to even just look back at a location they appreciated.

Participants of an event can converse with each other through the app's integrated chat system, removing the need for external social media. This makes finalising a plan much simpler as users can raise issues of event times, certain venues and so on.

User Stories

User Story 1	
Feature	Register a User
Story	<p>As a first-time user,</p> <p>So that I can access and use the application,</p> <p>I want to be able to register for a profile account.</p>
Case	<p>Given that I am on the homepage,</p> <p>When I click on the 'Sign Up/Log In' button,</p> <p>Then I am redirected to a signup form where I can create an account via:</p> <ul style="list-style-type: none">· Connecting through my Google account· Connecting through Facebook· Creating a new account
Priority	1 st Priority
Implemented	Yes

User Story 2

Feature	Creating a plan path and adding constraints
----------------	---

Story	<p>As a user,</p> <p>So that I can organise an outing with friends,</p> <p>I want to be able to create a plan path of what is available given certain filters and updated covid restrictions.</p>
Case	<p>Given that I have signed up as a user,</p> <p>When I click on the 'create plan' button,</p> <p>Then I am redirected to the 'planner filter' page,</p> <p>When I fill in various details:</p> <ul style="list-style-type: none"> · Event name · Start time · Budget · Type of event (date, drinks, chill, etc.) · Number of people, etc. <p>Then I should see an auto-generated plan path with the ability to customize this recommendation,</p> <p>And be able to finalise the plan path.</p>
Priority	1 st Priority
Implemented	Yes

User Story 3

Feature	Editing the plan path after it has been created
----------------	---

Story	<p>As a user,</p> <p>So that I can tweak a few changes in the plan path provided by the application,</p> <p>I want to be able to edit and update the plan path generated.</p>
Case	<p>Given that I am on the specific plan paths page,</p> <p>When I click on the 'Edit path' button,</p> <p>Then I can edit segments of the plan to suit my preferences.</p>
Priority	1st Priority
Implemented	Yes

User Story 4	
Feature	Displaying path details
Story	<p>As a user,</p> <p>So that if I want an overview of a path,</p> <p>I want to be able to view details about the path and the events/venues it contains.</p>
Case	<p>Given that I am on the homepage or events page,</p> <p>When I click on a specific plan path,</p> <p>Then I can view details such as the weather, map and path plan,</p> <p>And When I click on an event or venue,</p>

	Then I can view details about the venue/event also provided with a map.
Priority	2nd Priority
Implemented	Yes

User Story 5	
Feature	Deleting a plan after it has been created
Story	<p>As a user,</p> <p>So that in case of unforeseen circumstances that the event cannot be carried,</p> <p>I want to be able to delete the plan.</p>
Case	<p>Given that I am on the 'plan' page,</p> <p>When I click on the '-' icon,</p> <p>Then a message appears with a message to confirm the deletion of the event,</p> <p>When I click "Yes, I am sure",</p> <p>Then The plan is deleted,</p> <p>And it notifies everyone that the plan has been deleted.</p>
Priority	2 nd Priority
Implemented	Yes

User Story 6	
Feature	Automated booking
Story	<p>As a user,</p> <p>So that I can reserve a placement at a venue,</p> <p>I want to be able to book through the app without direct interaction with the event page/venue.</p>
Case	<p>Given that I am on the specific plan path page,</p> <p>When I click on the 'book' button,</p> <p>Then a selection of events that can be booked appears,</p> <p>When events are selected and confirmed,</p> <p>Then a booking for the selected venue/events is processed.</p>
Priority	2 nd Priority
Implemented	No

User Story 7	
Feature	View event history
Story	<p>As a user,</p> <p>So that I can compare details of plans and to look back on plans or maybe rebook a plan,</p>

	I want to be able to view past events by order of date (recent to oldest).
Case	<p>Given that I am on the homepage,</p> <p>When I click on the 'History' button,</p> <p>Then I am redirected to the 'History' page,</p> <p>And I am able to see the past history of events I attended with details.</p>
Priority	3 rd Priority
Implemented	Yes

User Story 8	
Feature	Chat system
Story	<p>As a user,</p> <p>So that I can organise and coordinate events without needing to access multiple social medias,</p> <p>I want to be able to chat with them within the app.</p>
Case	<p>Given that I have signed up as a user,</p> <p>And my friend has also signed up,</p> <p>When I click on the 'Chat' button,</p> <p>Then I am redirected to the 'Chat' page,</p> <p>When I click on the 'Create Chat' button,</p>

	<p>Then I am able to add a user/s via phone number, account name or email.</p> <p>And a new chat is created</p> <p>And messages can be sent</p>
Priority	2 nd Priority
Implemented	Yes

User Story 9	
Feature	Notify group members
Story	<p>As a group member,</p> <p>So that I can be informed of event time, location changes or any updates within the group,</p> <p>I want to be notified of updates through notifications on my phone.</p>
Case	<p>Given that I am a member of the group,</p> <p>When any changes or updates occur within a group or event,</p> <p>Then I receive a notification of the update so I can be informed.</p>
Priority	2nd Priority
Implemented	Yes

User Story 10	
Feature	Updating profile details/preferences
Story	<p>As a user,</p> <p>So that friends can see what my interests and preferences are through my account,</p> <p>I want to be able to add/update my personal preferences within my profile.</p>
Case	<p>Given that I am on the 'Home' page,</p> <p>When I click on the 'profile' button/display picture,</p> <p>Then I am redirected to the 'My profile' page,</p> <p>When I click on the 'Preferences' button</p> <p>Then I am redirected to the 'preferences' page.</p> <p>And I am able to update my user preferences</p>
Priority	2nd priority
Implemented	Yes

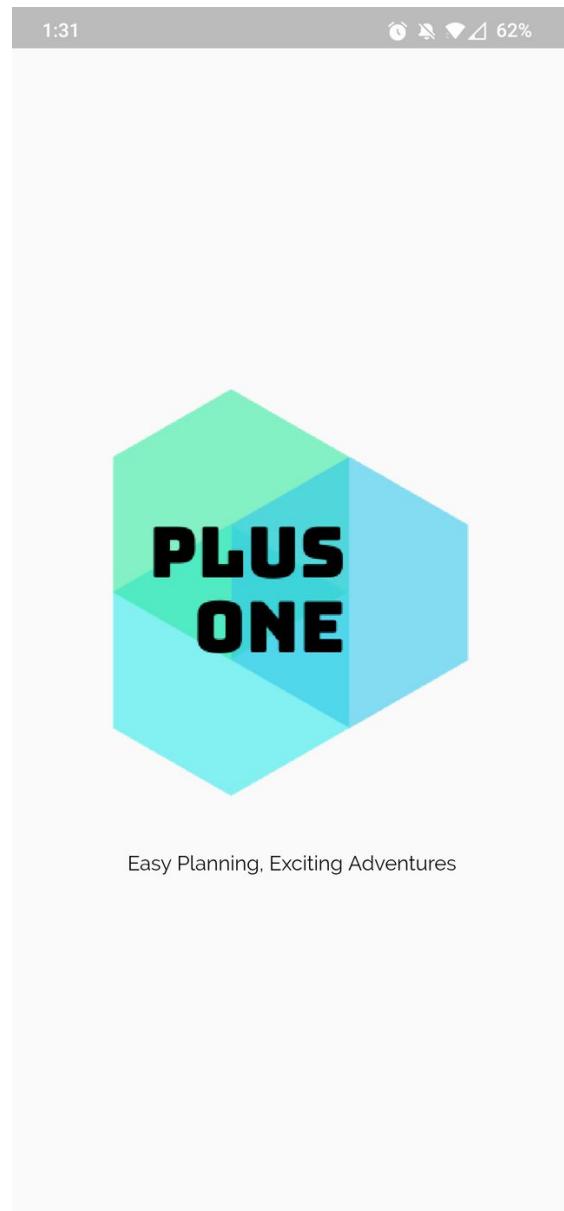
User Story 11	
Feature	Notify plan path
Story	As a user,

	<p>So that I can know when to leave one venue/activity on time to make it to the next one,</p> <p>I want to be sent notifications to my phone to alert me that I need to be at the next venue soon</p>
Case	<p>Given that I have already created a plan path,</p> <p>And the event is currently underway,</p> <p>When one event in the plan path is nearly finished,</p> <p>Then a notification will be sent to the phone to alert the user. (This will be the case for both normal and mystery plans)</p>
Priority	2nd Priority
Implemented	Yes

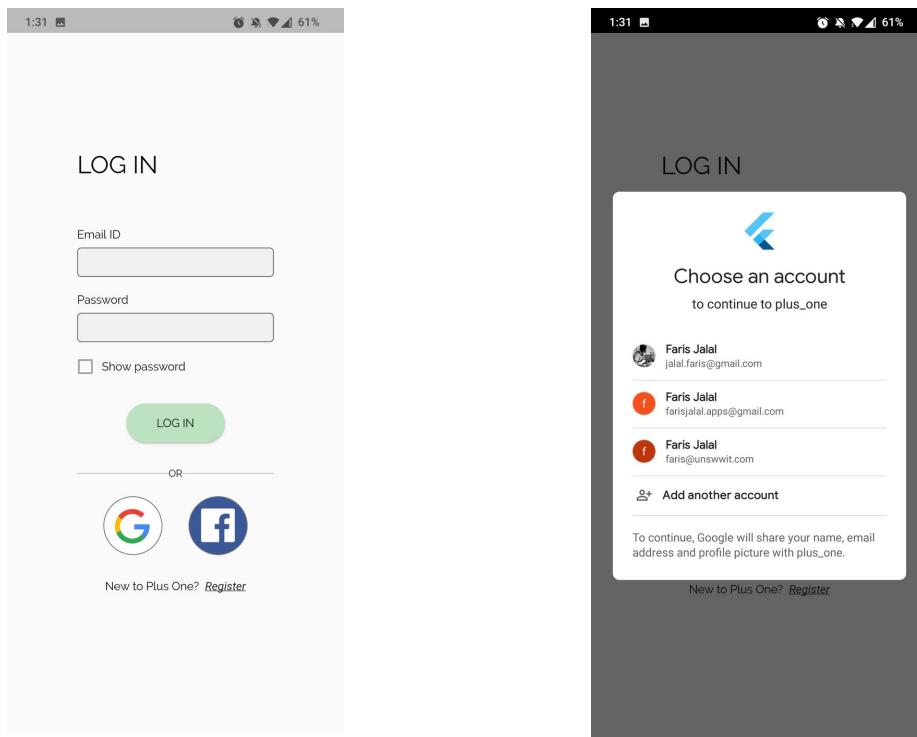
User Story 12	
Feature	Getting Location Specific Details
Story	<p>As a user,</p> <p>So that I can travel hassle-free to the event,</p> <p>I want to be able to get more data about travelling to particular locations.</p>
Case	<p>Given that I have already created a plan path,</p> <p>When it is time to leave for the event,</p> <p>Then a notification will be sent to the phone to alert the user.</p>

	From the view path screen the user can click on the map and this will redirect the user to a google maps website where they can get more information.
Priority	2nd Priority
Implemented	Yes

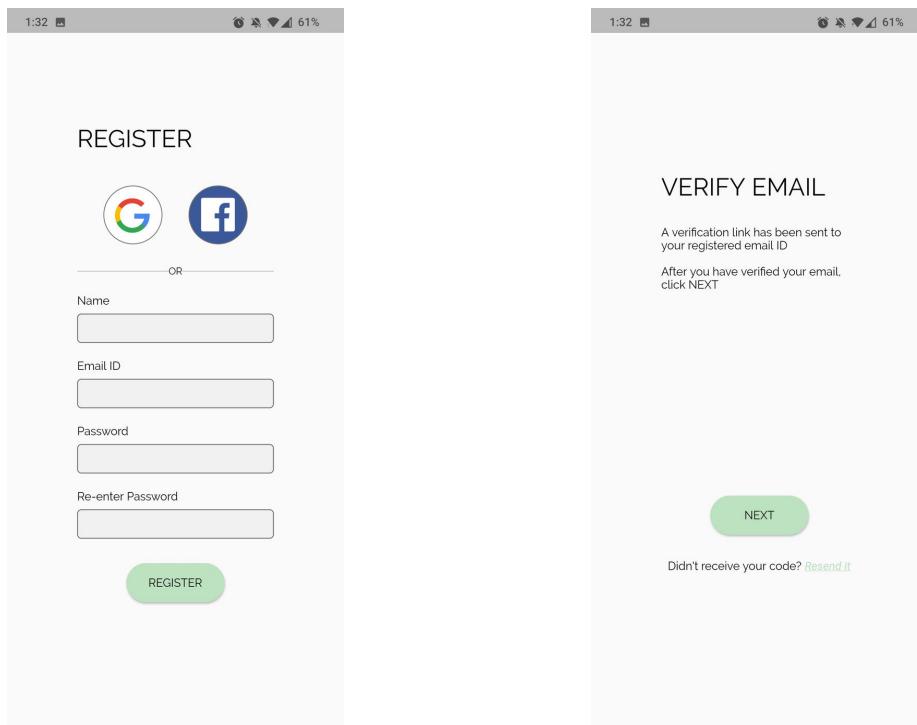
User Interfaces



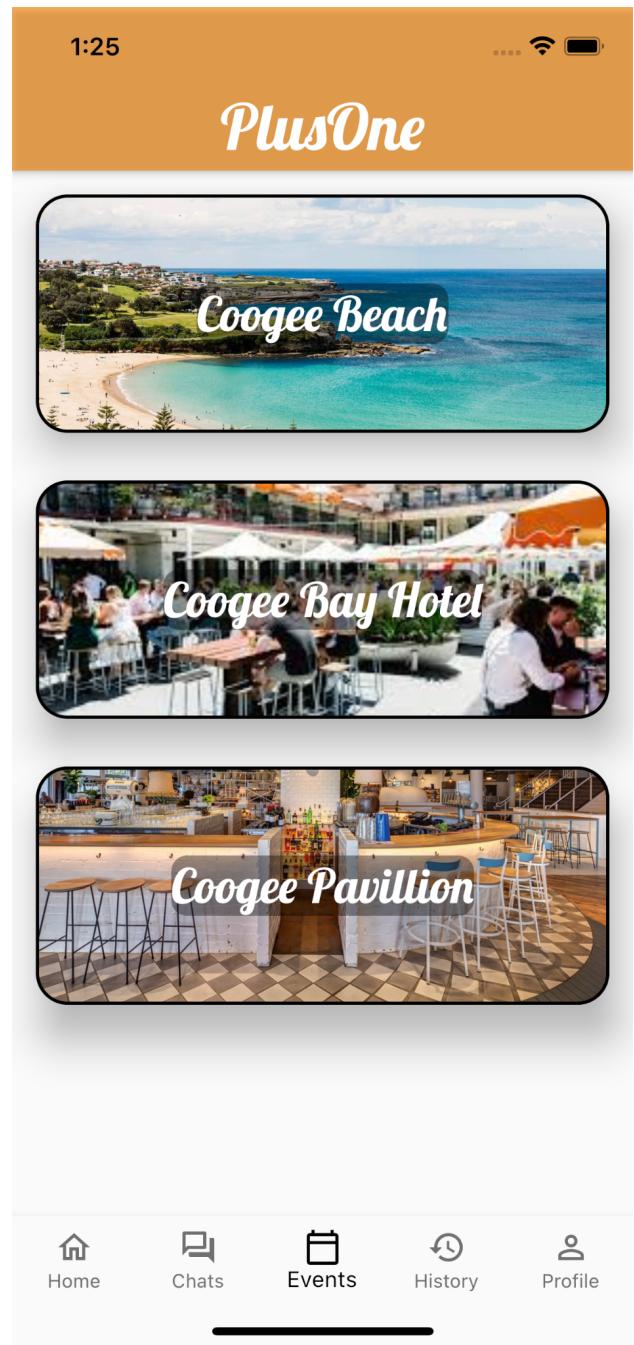
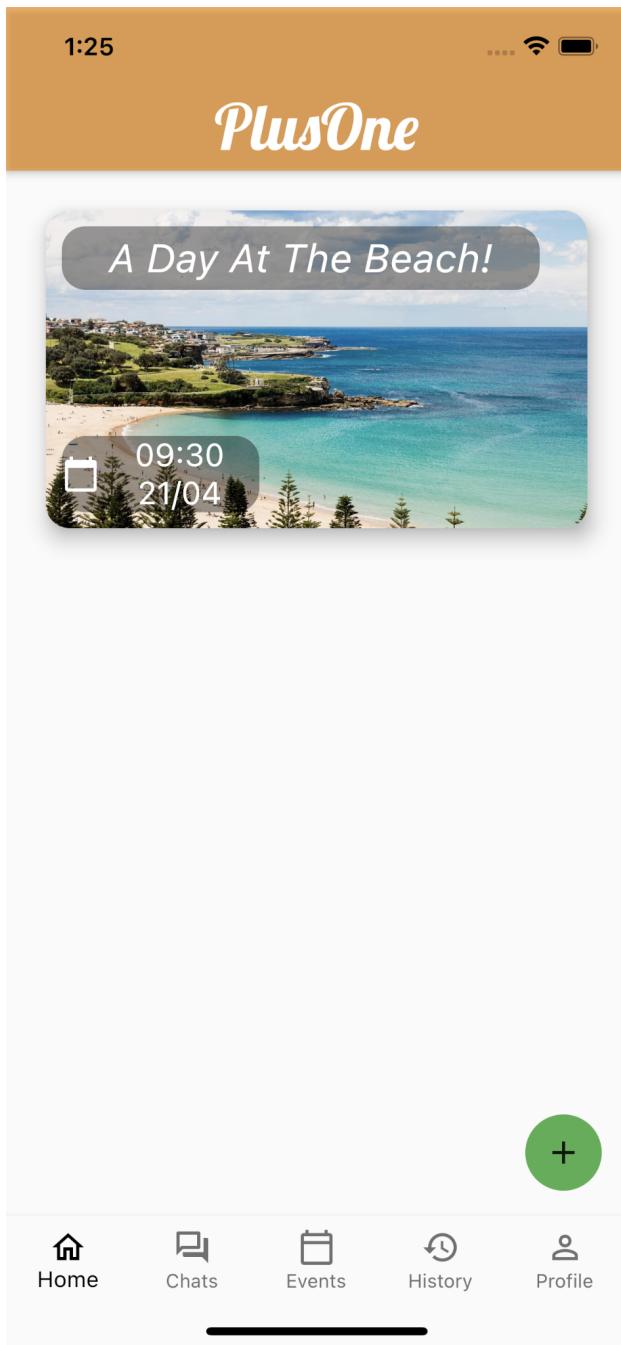
Title screen - When opening the mobile app, users are greeted with a title screen with our logo and our company motto "Easy Planning, Exciting adventures".



Login screen - After the title screen, existing users can login through either email and password. If logging in via external accounts, such as Google or Facebook, a pop-up screen will appear prompting the users to select an account.

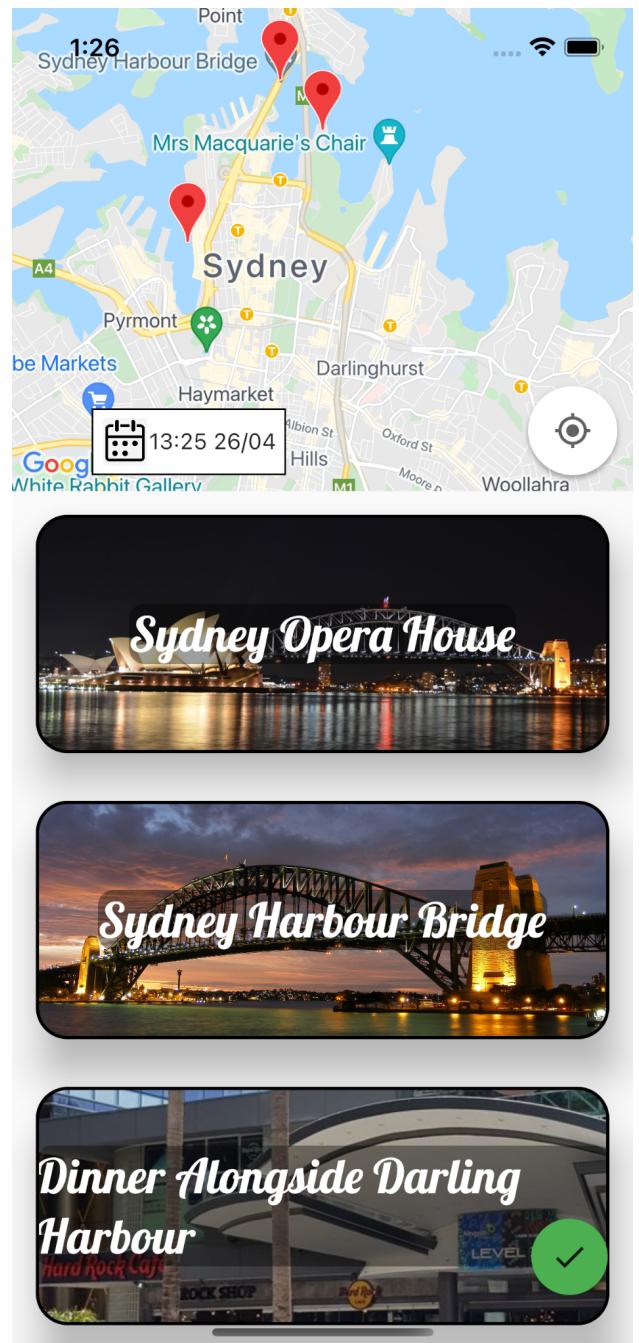
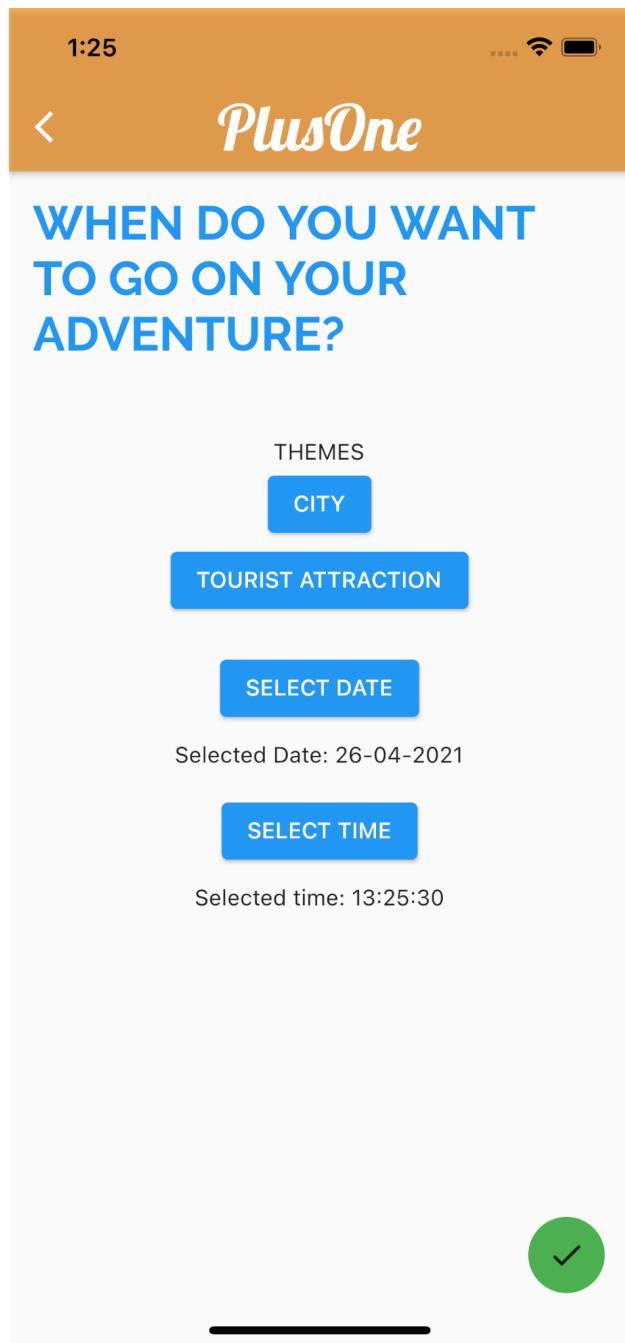


Register screen - New users can create an account by entering their email and password or through Google/Facebook. Users will need to verify their email.



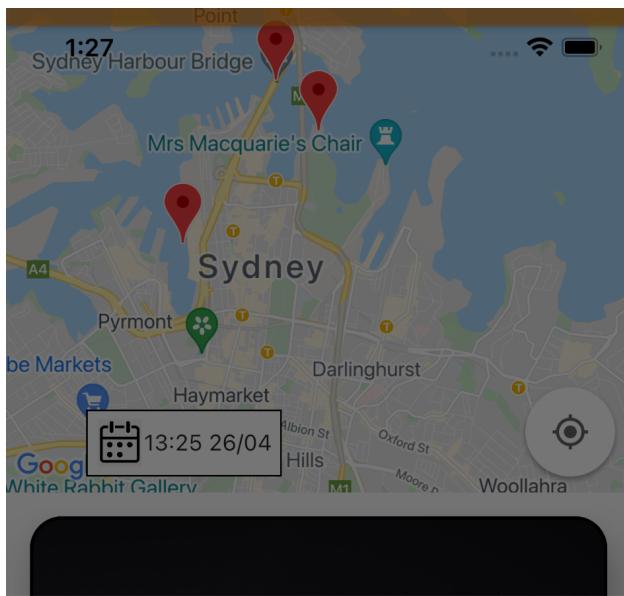
Home screen - This displays the upcoming plans for a user, displaying details such as name of event, date and time. In addition, there were going to be pre-generated plans that users can suggest to friends/family but due to time restraints were not able to be complete.

Event page - A user can click into the listed plan to display the organised path with a backdrop of the venue/location and name. A user can select an event to gain further details (which will be looked over another screenshot).



Plan creation - This page is the initial page for a user to create a plan. They are greeted with a list of themes that are randomly generated. A user can select a date and time, which will be considered by the database when generating a plan.

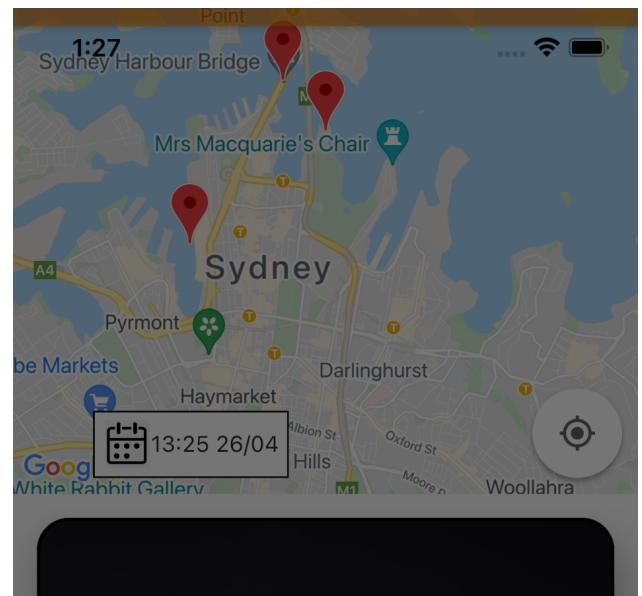
Generated plan - After the user has input their preferences, a plan is generated. This displays the list of events shown through cards of a name and backdrop of the location. It is accompanied by a map that points the location of each event, in which users can select to identify which point refers to what.



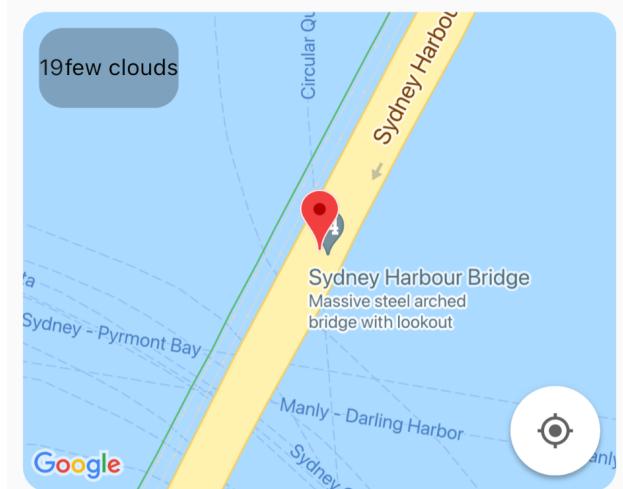
Sydney Harbour Bridge

The Sydney Harbour Bridge is an iconic landmark spanning one of the finest natural harbours known to mankind. Opened in 1932, the bridge is fondly nicknamed the Coathanger by Sydneysiders. You can walk and cycle across the bridge or climb to the top for stunning views.

Description Map



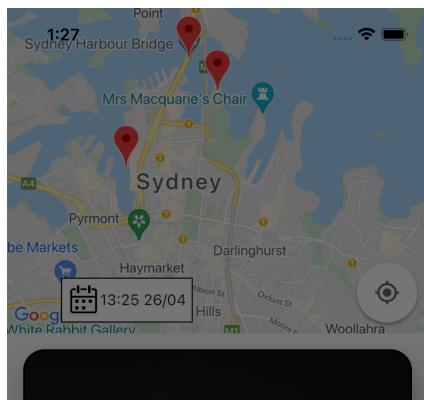
Sydney Harbour Bridge



Event details (description) - When an event card is selected it displays in-depth details about the venue/location. This is to provide participants with extra information if the generated event is unfamiliar to them.

Event details (map) - Users can access a map that is zoomed in closer to the location so users can clearly identify its whereabouts. On the top left corner is the real time weather in that area, identifying temperature (in Celcius) and conditions (e.g., few clouds).

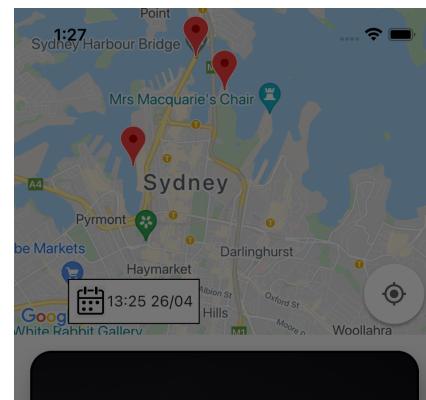
Many similar pages like these were developed for other events, with a few more examples below.



Dinner Alongside Darling Harbour

Lavish dinner at the famous Hard Rock Cafe alongside Darling Harbour. Enjoy the spectacular view!

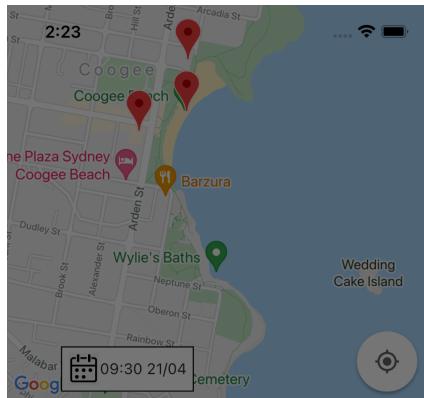
Description Map



Dinner Alongside Darling Harbour



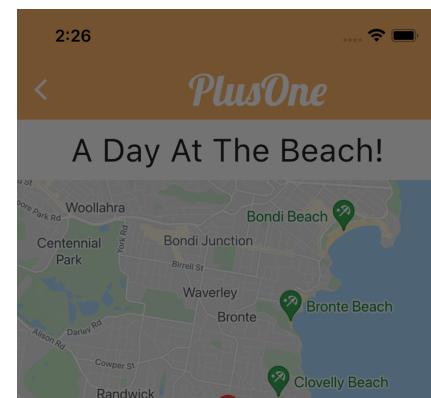
Description Map



Coogee Beach

Coogee Beach is a great beach with calm surf and is family-friendly. The promenade area has restored historic buildings and nurtured parklands. Most of the facilities are located at mid-beach with showers, change rooms and toilets near the Arden Street bus stop. Upmarket restaurants now jostle with fish and chip shops and an increasing number of boutiques.

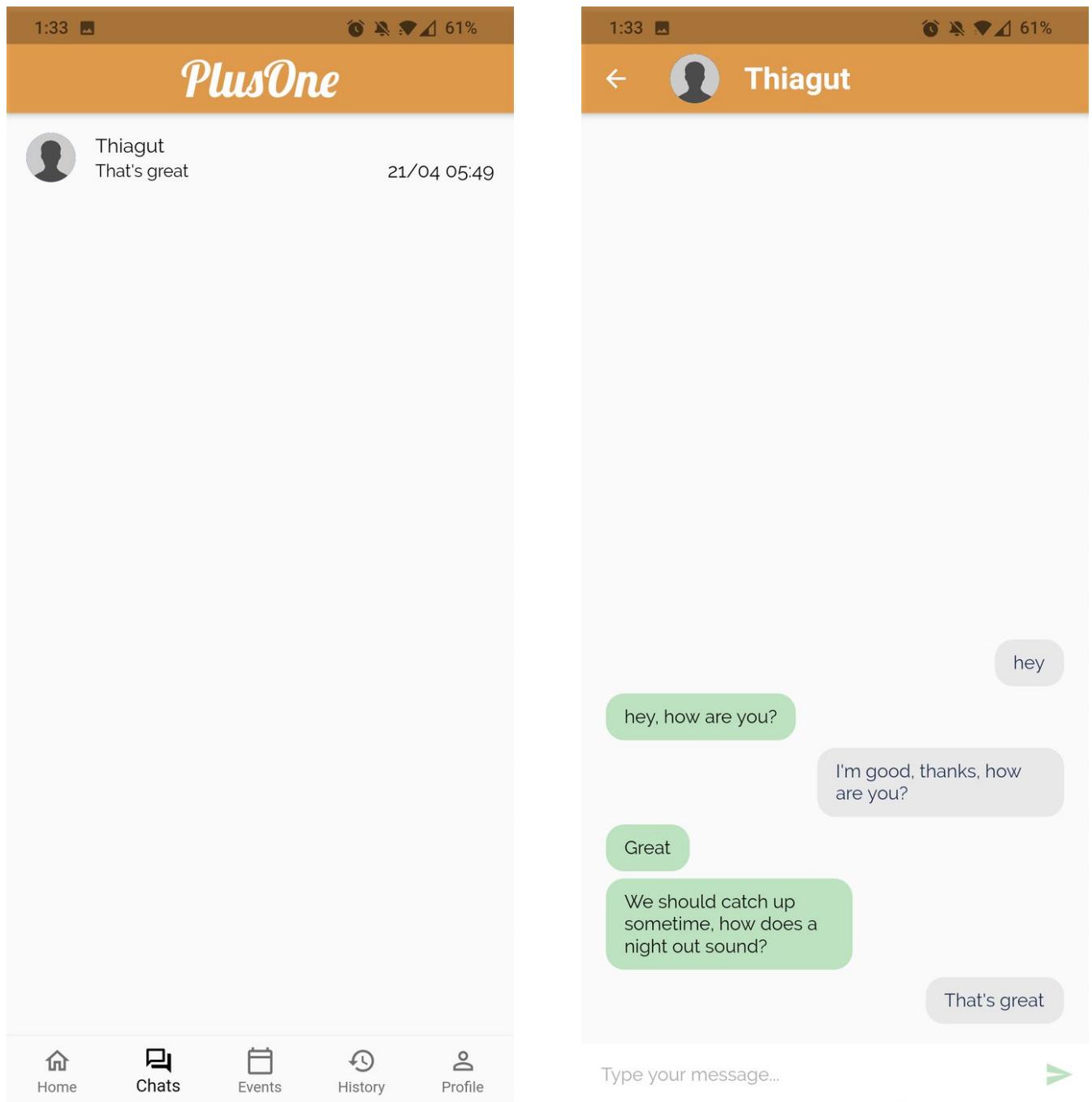
Description Map



Coogee Beach

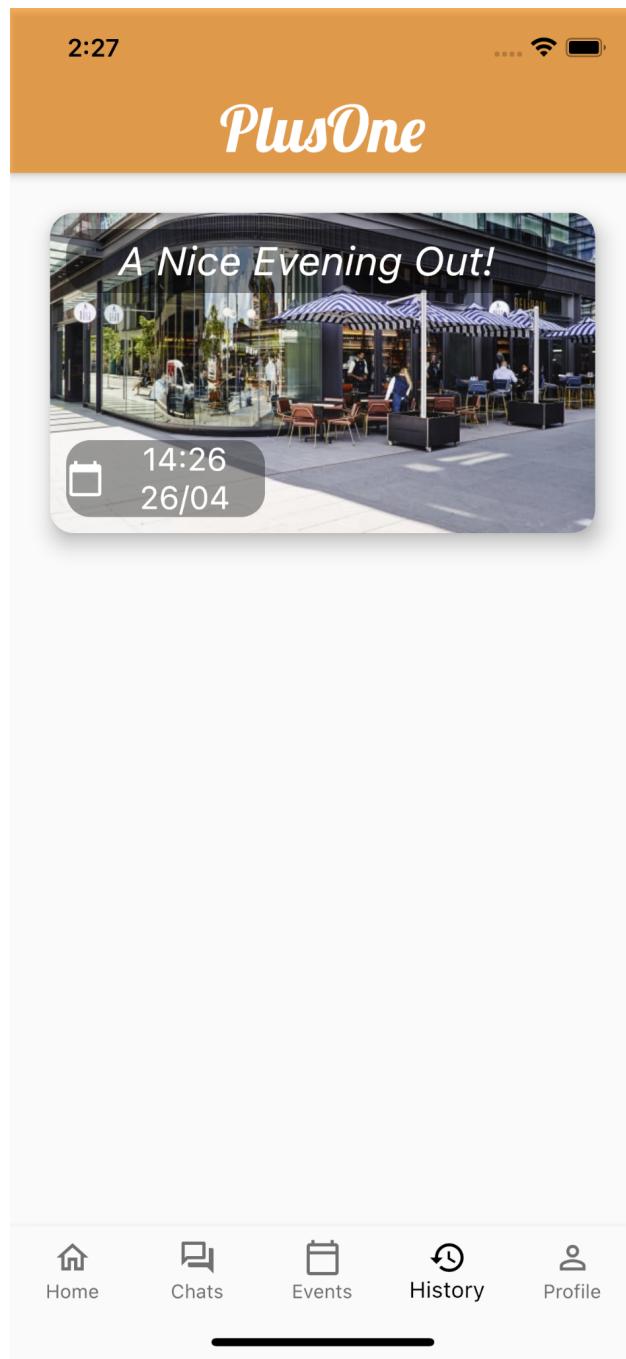


Description Map

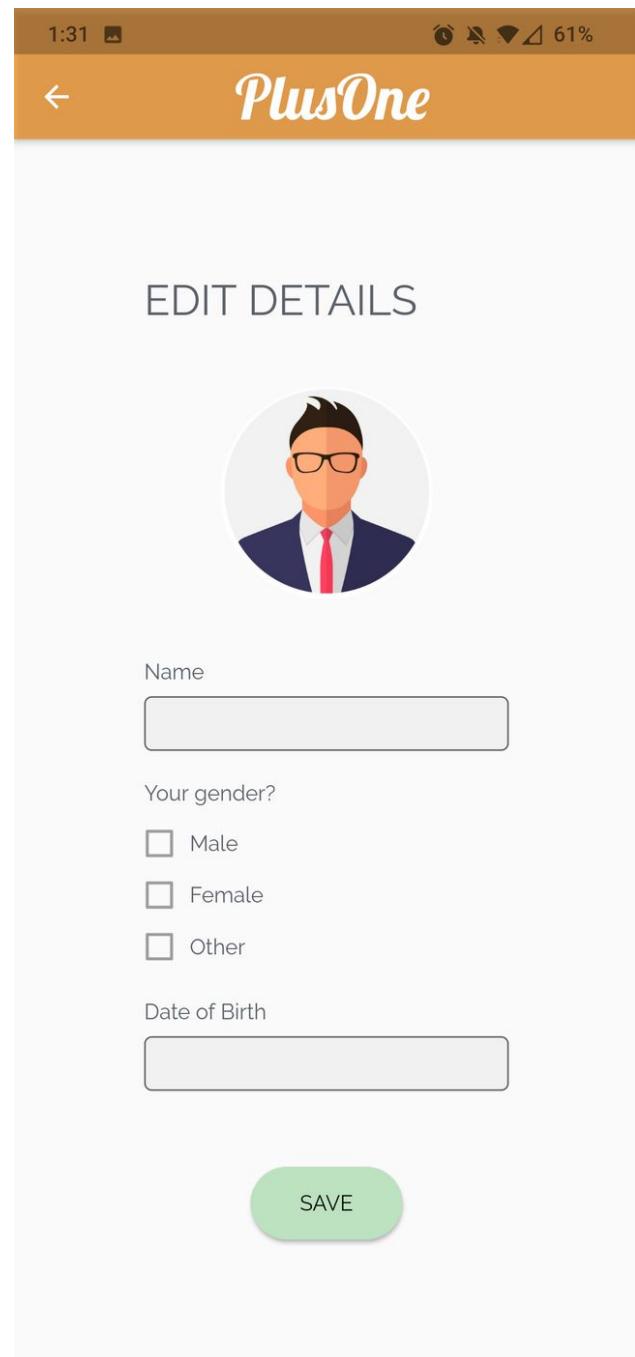
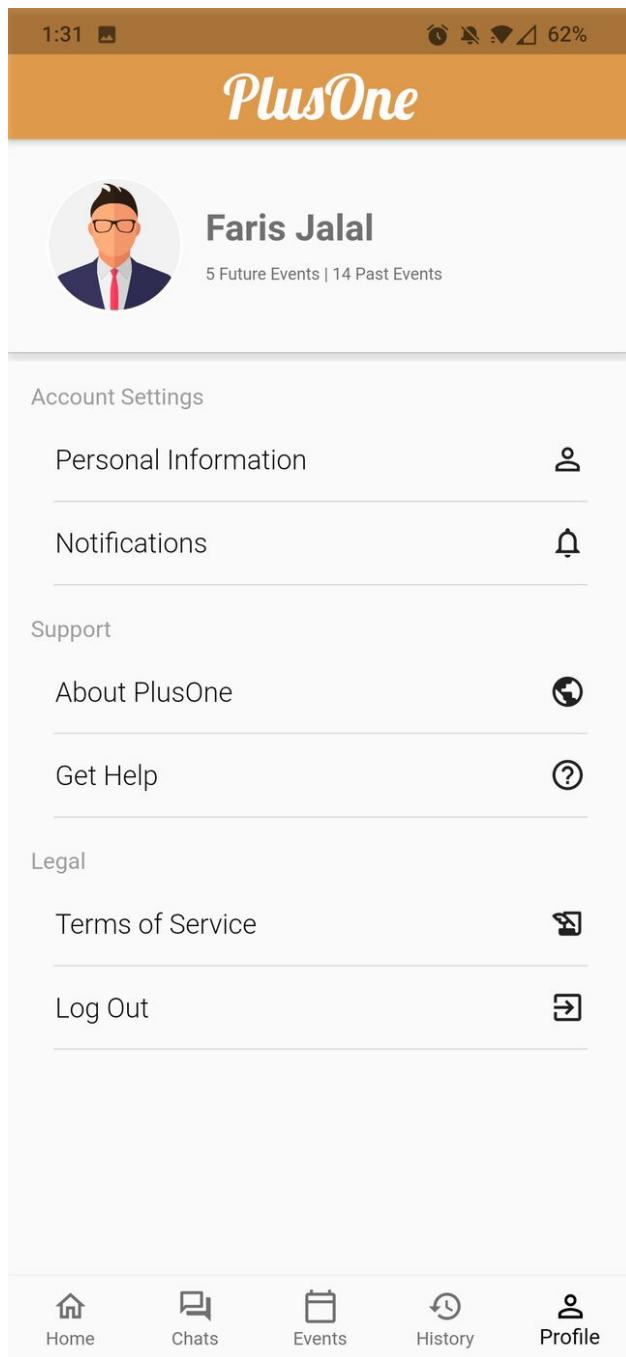


Chat - The chat screen displays a list of active chats, either group or individual, in which users can message others.

Chat (clicked in-chat) - This displays the actual chat between users, with the user (on the right), sending to the recipient (on the left)



History - This page displays a list of the past plans the user has participated in, organised by newest to oldest.



Profile - Displays the user information, settings and support pages users can have access to. Users also have the added capability to edit personal details.

Software Architecture

Diagram 1.1 represents the software architecture of the PlusOne mobile application which will be available on iOS and Android platforms.

It is split into two distinct sections, the frontend and the backend consisting of the selected mobile application stack, external data sources, programming platforms, languages and SDK's that will be implemented in the application.

A brief description of the frontend, backend and the advantages and disadvantages of their respective components can be found below the diagram.

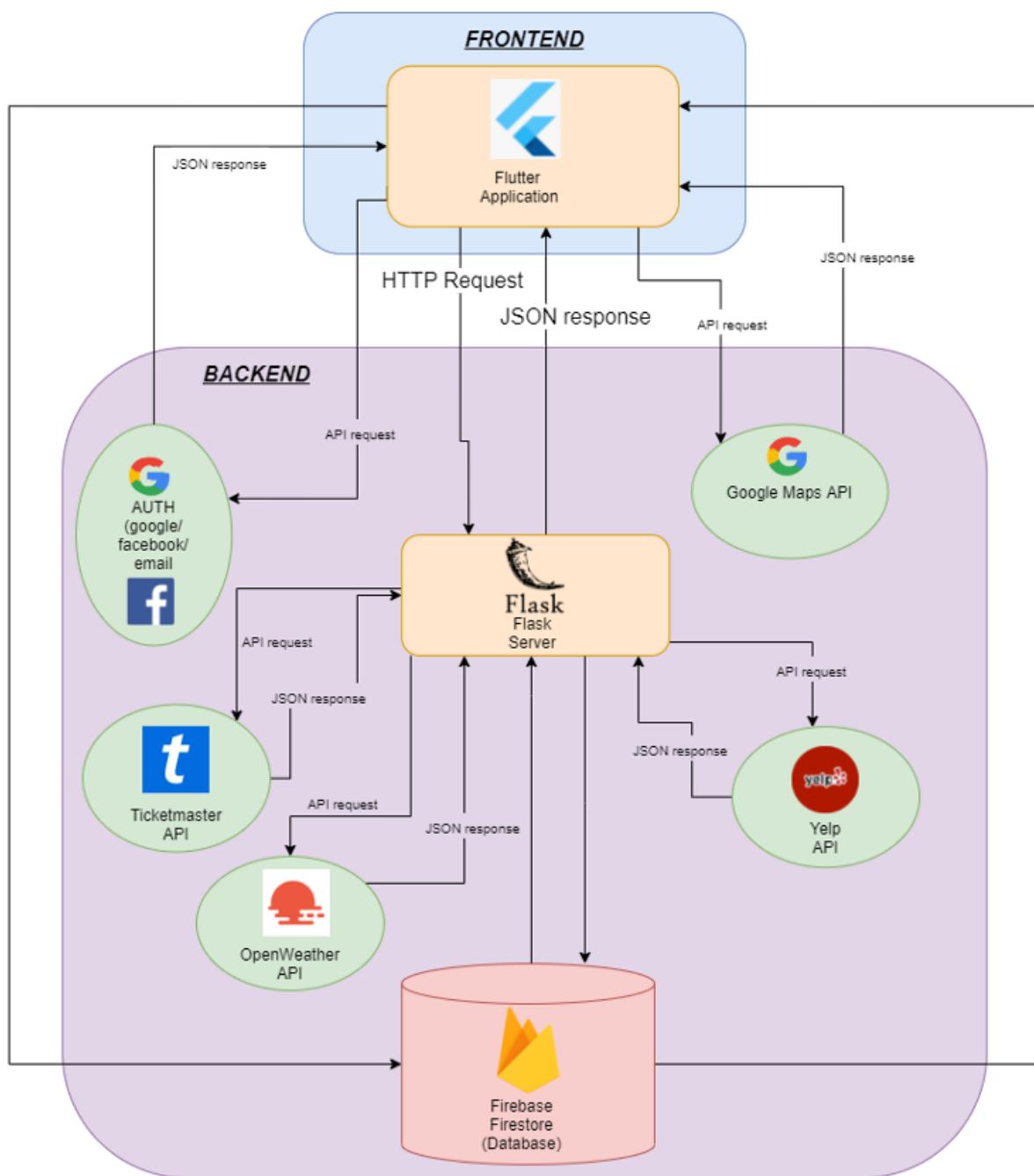


diagram 1.1

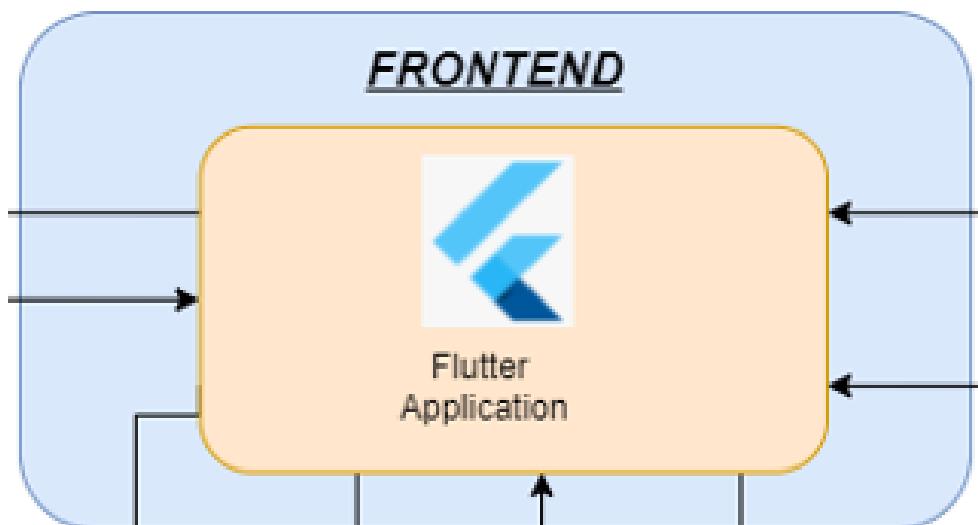
Frontend

Flutter:

It is an open-source SDK, developed by Google and written in Dart, C, and C++. We chose to use Flutter to develop our application as it allows us to create apps for Android and iOS devices with just one codebase written in Dart. Furthermore, flutter seamlessly incorporates many material design features making it easier for developers to create beautiful apps without spending unnecessary amounts of time on UI and styling. Flutter also incorporates ahead-of-time(AOT) compilation allowing apps to achieve higher performance. If we expand PlusOne to a web platform, the latest flutter update allows us to use the same codebase(with minimal changes) to deploy the application to the web too.

Alternatives to flutter that we considered were React Native and Xamarin. These offer similar attributes but third-party libraries are far from perfect, not compatible for high-performance applications, especially with Xamarin as it is written in .NET, which will have API limitations. A main reason flutter was chosen was due to our developers having experience writing in dart, over other languages such as C# (Xamarin), .NET (Xamarin) and JavaScript (React Native).

With a vast developer community, excellent support, and regular updates, Flutter is the ideal platform to develop the app.



Backend

Firebase Cloud Firestore:

Firestore is a fast NoSQL database designed for scaling with the powerful infrastructure of Google Cloud Platform offering automatic horizontal scaling (in and out) as per the application's load, offering strong reliability as it distributes data across multiple data centers in distinct regions.

It features built-in offline support for mobile (iOS, Android) and uses local cache to serve and store data. Hence, an app built using Cloud Firestore remains responsive regardless of network latency or internet connectivity. The app can read, write, listen to, and query data even when the device is offline. If changes are made in the offline state, the same will be synced to the cloud when the client comes back online.

Firebase requires less data flattening or denormalization as it is more structured and utilizes nesting of objects. This supports indexed queries, thus it enables you to combine filtering and sorting on a property in a single query. The database offers atomic (all or nothing executes) write and transaction operations which also means that the transactions will keep repeating until completion, automatically.

Amazon Web Service was a heavy consideration, but was rejected as through research we found Firebase allowed us to get a minimum viable product operational relatively quickly and is better for our intended uses of user authentication and pushing notifications.

All these advantages make Firestore an ideal database solution for a mobile application like PlusOne.

Flask:

Flask is a lightweight WSGI web development framework that is used with Python to build backend systems. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It is one of the most popular Python web application frameworks, and hence has a lot of readily available resources and documentation. It is lightweight, versatile, and can be used easily along with Firebase Cloud Firestore, our database solution. The backend Python server will need to handle API calls as well as process and store information in the database and hence Flask is the most suitable framework available at our disposal.

Other frameworks we could have implemented were Django and FastAPI. The reasons we didn't choose them:

FastAPI is relatively new and has limited documentation, so the learning curve is very high. And for Django, in terms of speed, ease of use and flexibility, it lacks behind Flask. All in all, the ease of use and familiarity of the Python language was a main reason we chose Flask.

[Yelp API:](#)

The Yelp API allows developers to access a database of the newest and most exhaustive restaurant content to power applications with. It covers and returns the best local content and user reviews from millions of businesses across 32 countries. The API allows us to list restaurants by parameters such as type, name, location, ratings along with detailed information such as location coordinates, discounts.

This allows PlusOne to have a wide variety of dinner options at the best rated restaurants across town. It also returns responses in JSON format, which proves beneficial to us as it ties in with the NoSQL database which stores data in JSON-like documents and collections.

We had previously decided to implement the Zomato API, but were unaware that the company shut down its developer page, resulting us to use the Yelp API which offers the same features and functionality as Zomato.

[Ticketmaster API:](#)

Ticketmaster is a popular event management and ticketing platform that allows users to find forthcoming local occasions and receive personalized recommendations. It is used by millions of users. The API allows us to list events, ticket classes, discounts, attendees, venues along with detailed information such as location coordinates, reviews, organizers, formats, pricing, display settings, and media. This allows PlusOne to have a wide variety of event options across town. It too, like the Yelp API, returns responses in JSON format, which proves beneficial to us as it ties in with the NoSQL database which stores data in JSON-like documents and collections.

[Facebook Login:](#)

Signing up and logging in with Facebook has been included as it has a large user base and allows new users to create a new PlusOne account quickly and easily.

[Google Login:](#)

The option where a user can signup and login into the PlusOne app using their google account has been included as Google is very popular and has an extensive user base. It is also very secure as it uses the OAuth 2.0 API and integrates seamlessly with Flutter.

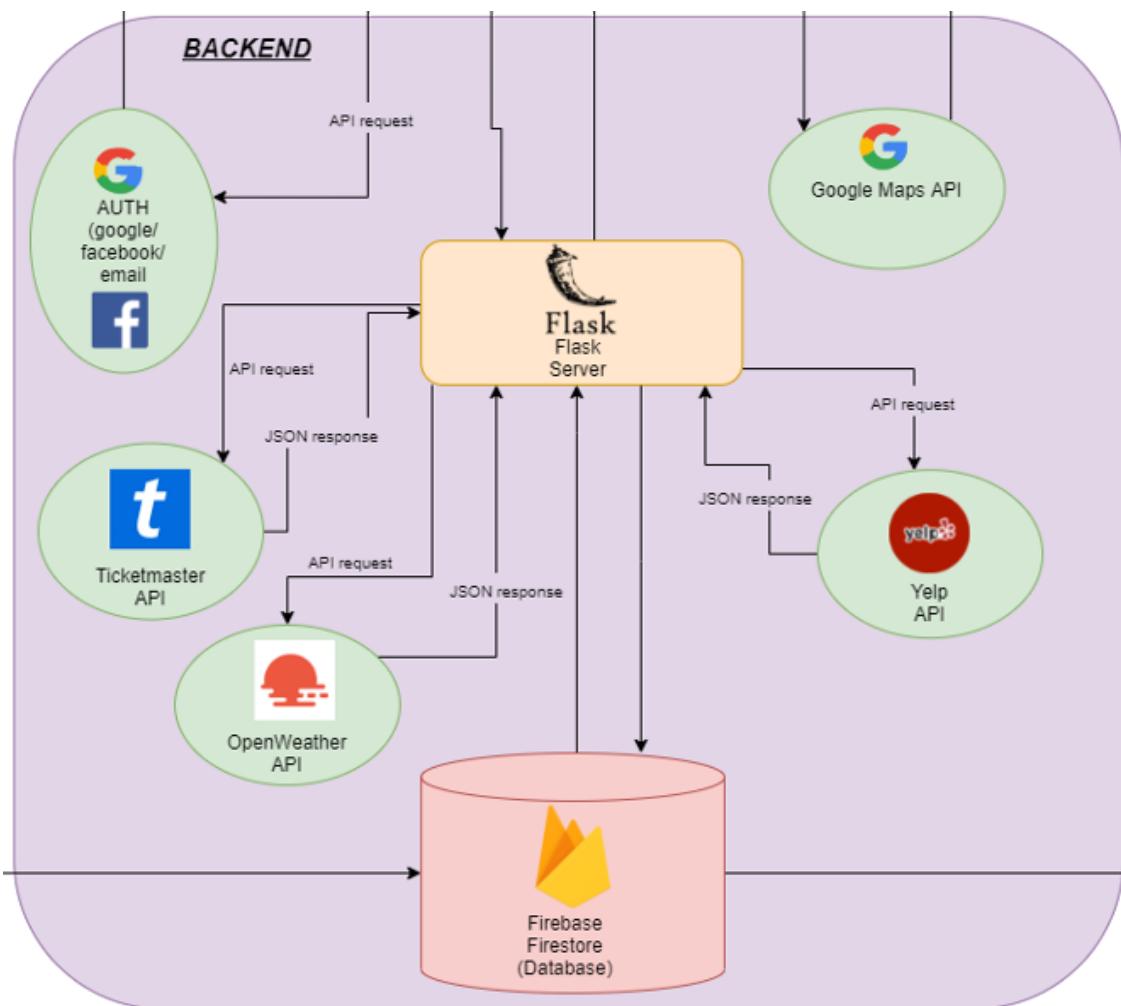
Google Maps API:

The Google Maps API is used to provide the user with location data like the geographical locations of particular events or venues and travel-related data like optimal routes from venue to venue. It also has integrated ride-share application support (Uber, Didi, Lime, etc) which allows users to travel between event venues with ease.

The choice to use the Google Maps API was obvious as it is an industry-standard in providing geographical location data with 99% coverage of the world, 25 million updates, and approximately 1 billion monthly active users.

OpenWeather API:

We will be using this API to obtain data about the weather at particular locations. We chose to use the OpenWeather API as it provides current weather data, forecasts, location-based weather data and is widely used in web and mobile applications to provide weather data.



Overall, these architectural choices ensure that our application is responsive and that all components can seamlessly integrate with each other.

Testing

Advanced REST client:

Tests were written in ARC to see whether the backend API's were functioning consistently and correctly, as per the project specifications. This tests the API endpoints and validation is performed based on the return value, to ensure the frontend is provided with the correct outputs. This aided the development process as issues were found and fixed quickly. Tests on non-functional requirements such as response speed were also tested to maintain a robust front-end.

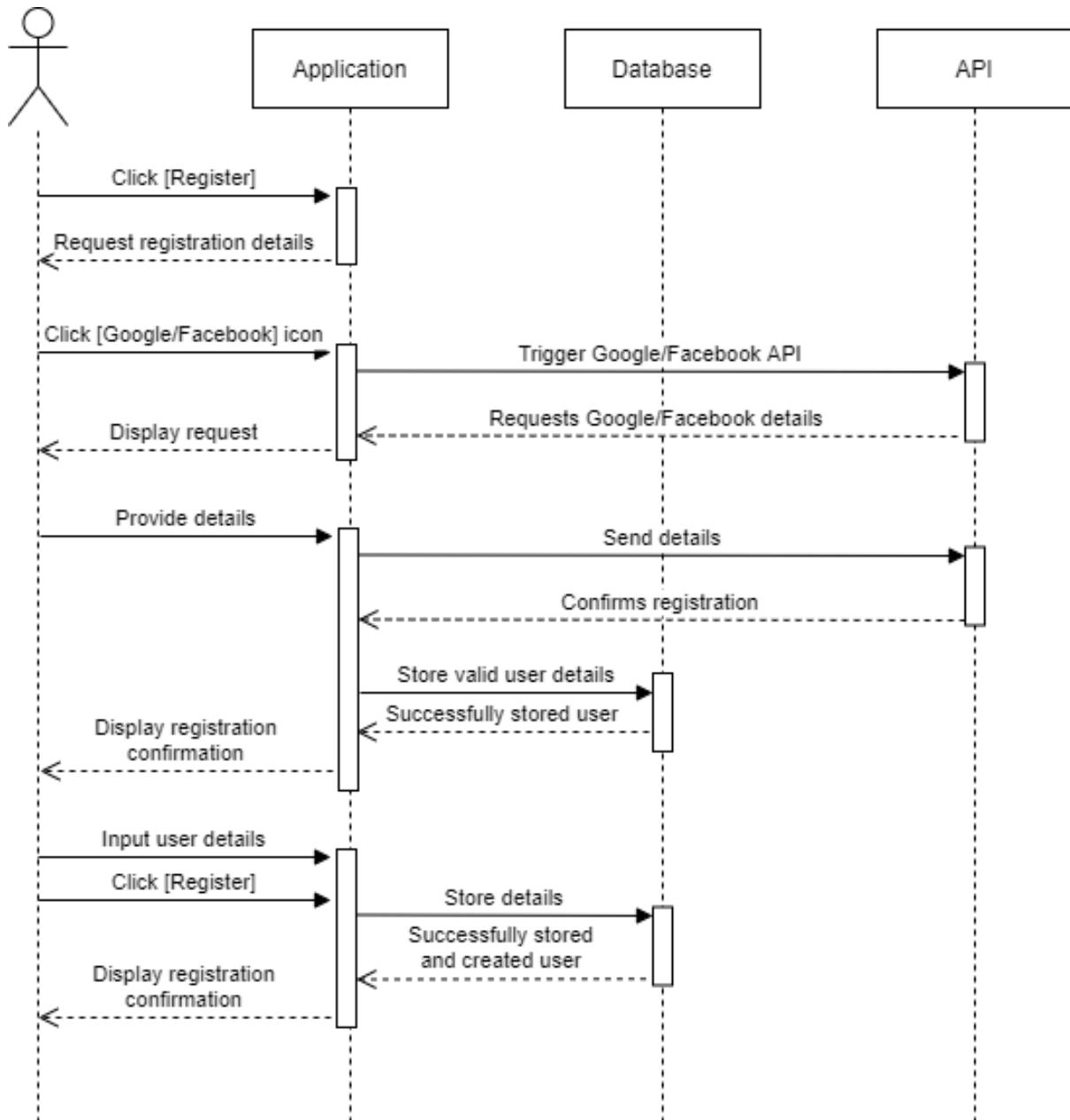
Sequence Diagrams

Application component: Refers to both frontend and backend of software architecture.

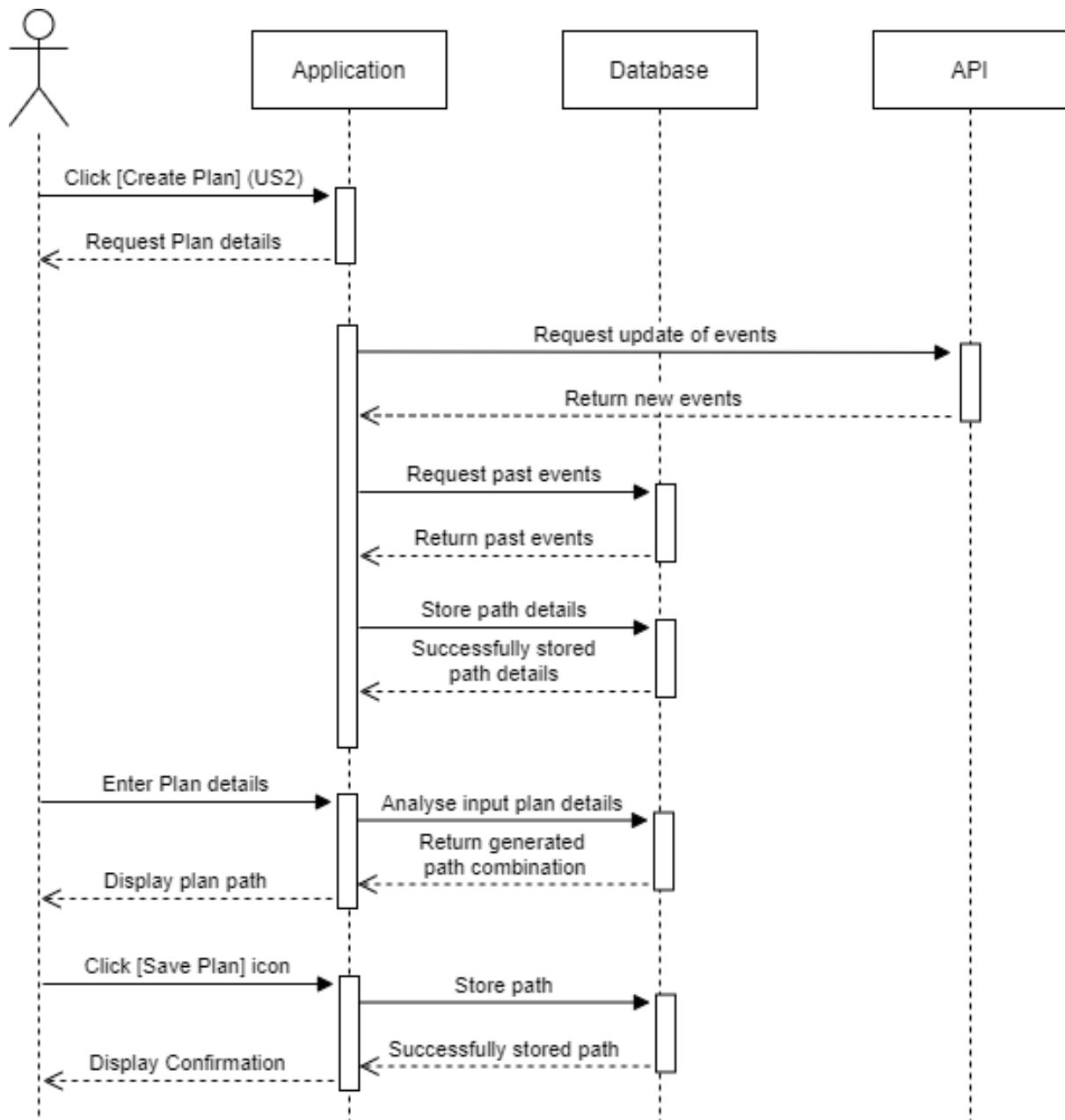
Database component: Refers to the Firebase Cloud Firestore database.

API component: Refers to the various API's implemented in the software architecture.

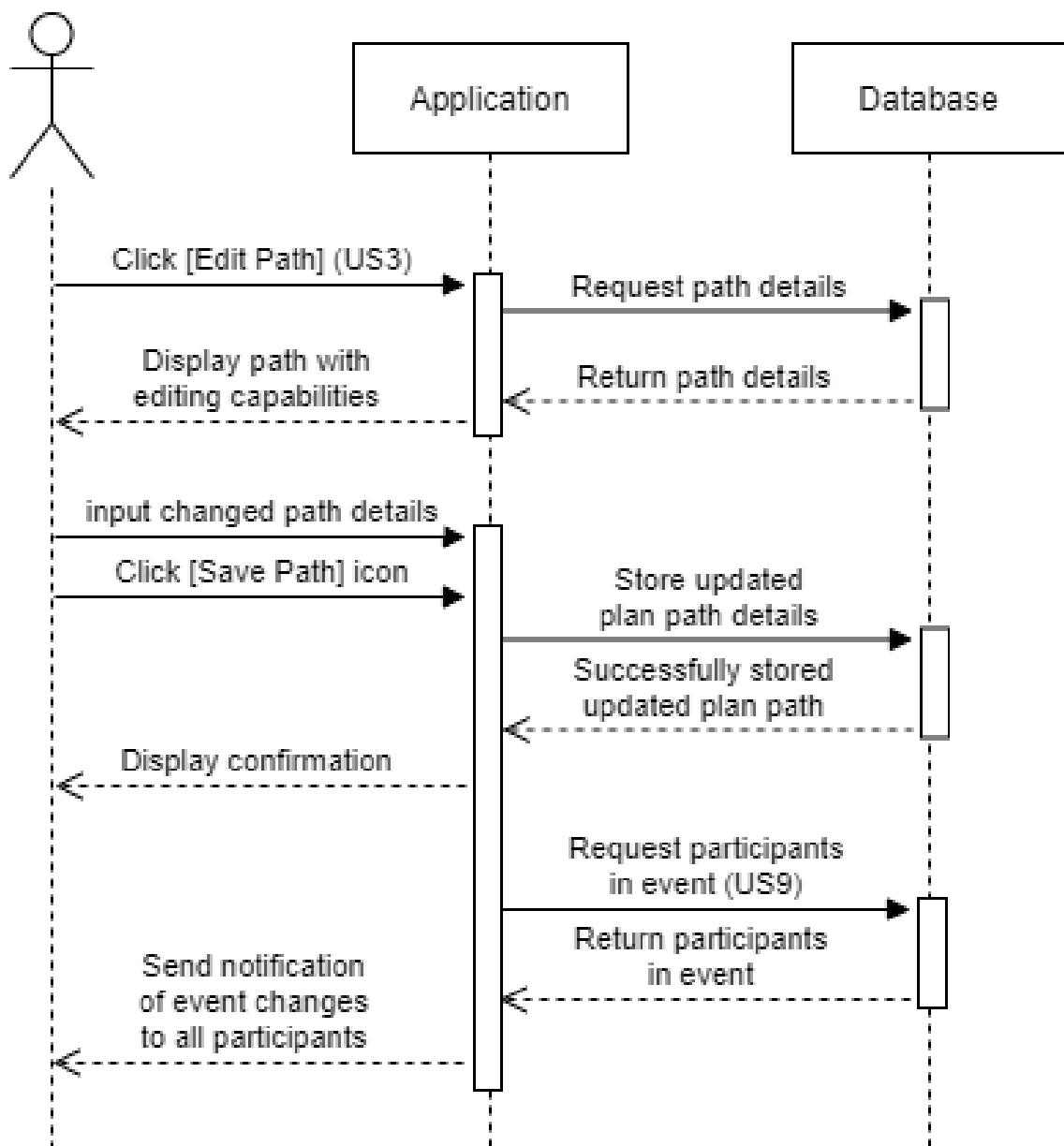
Epic story 1: As a first-time user, so that I can access and use the application, I want to be able to register for a profile account. (US1)



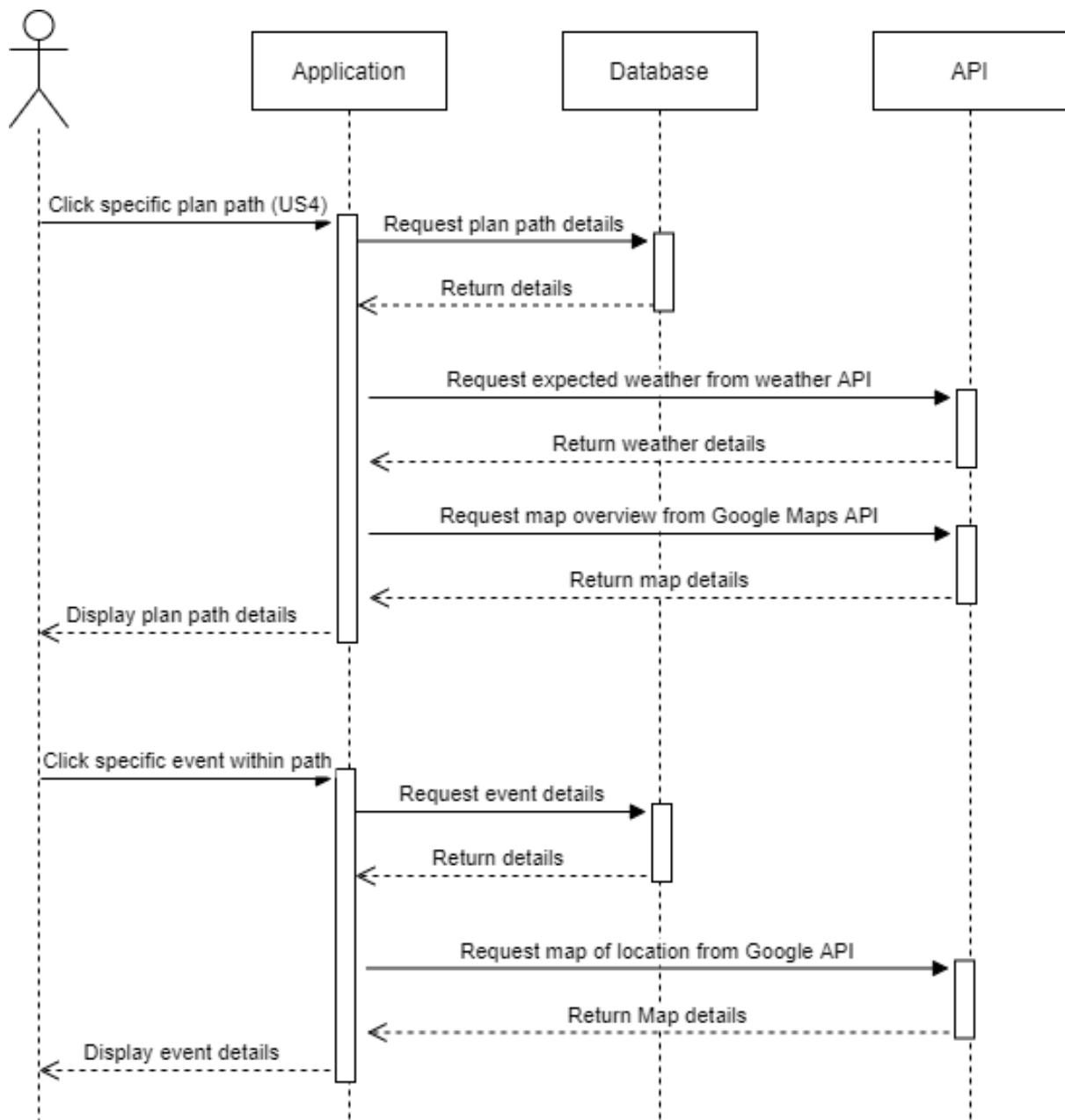
Epic story 2: As a user, I want to be able to create a plan, view plan details, book venues and also have the ability to edit and delete these plans notifying all event participants of any updates and changes. (US 2,3,4,5,6,9)



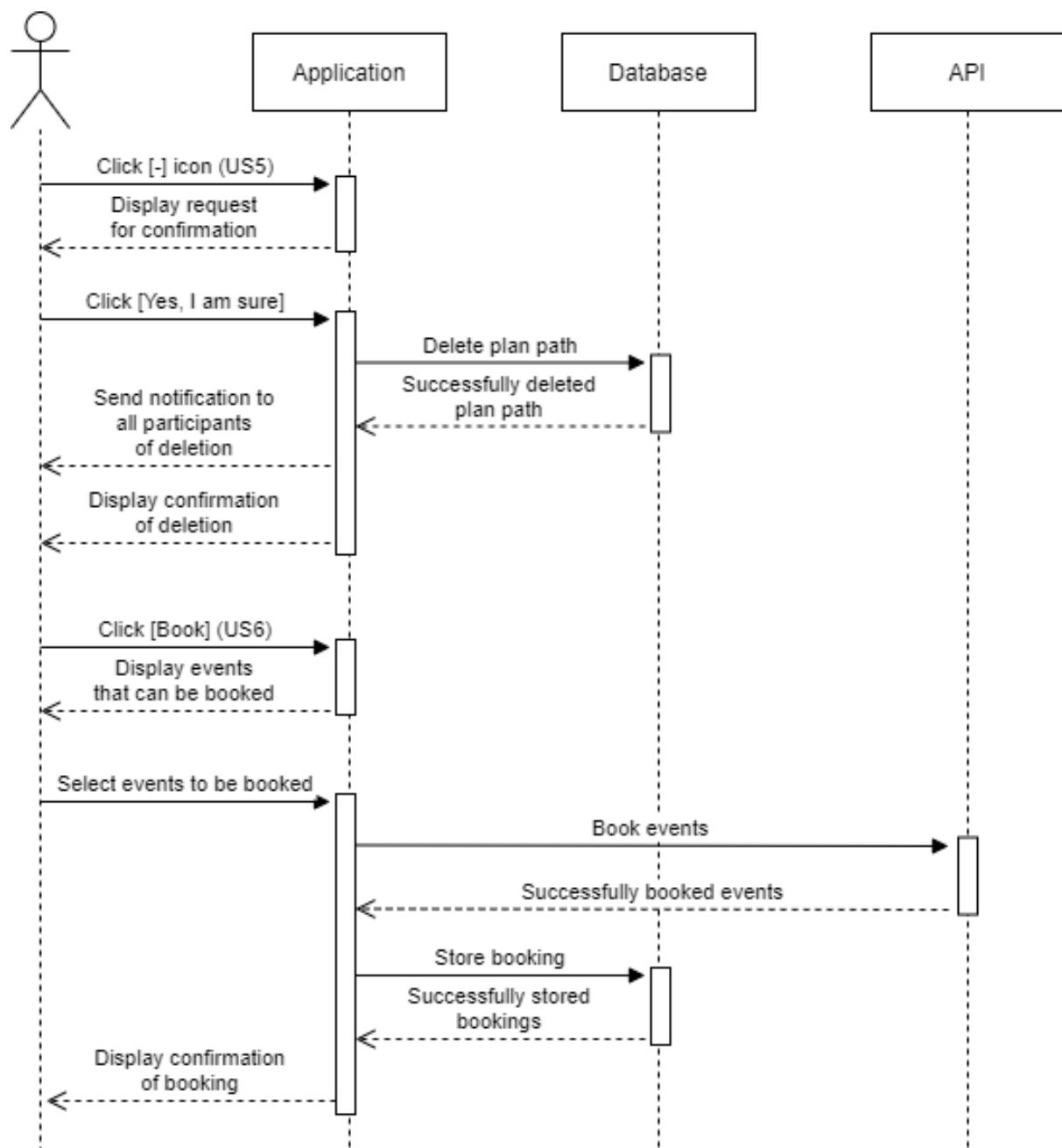
Epic story 2 (Cont.)



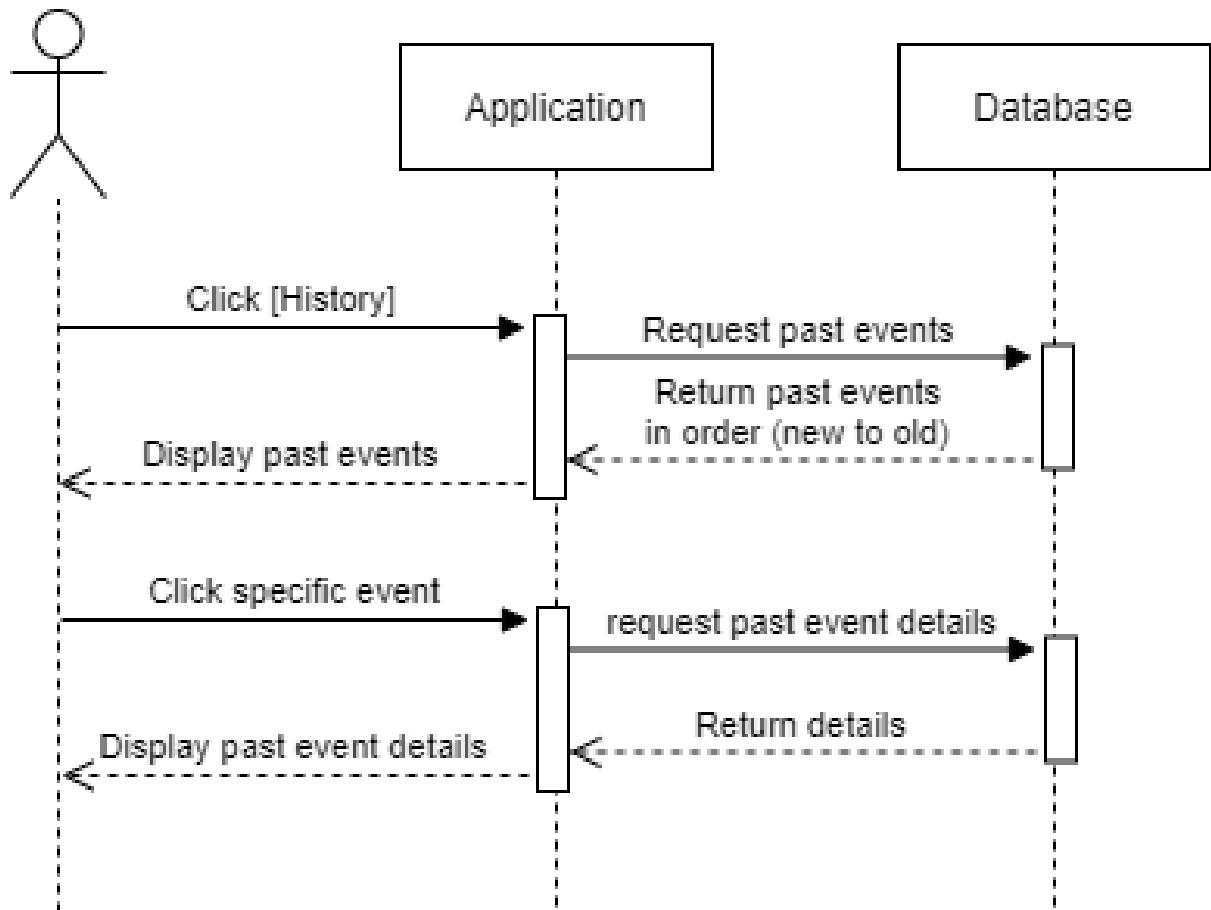
Epic Story 2 (Cont.)



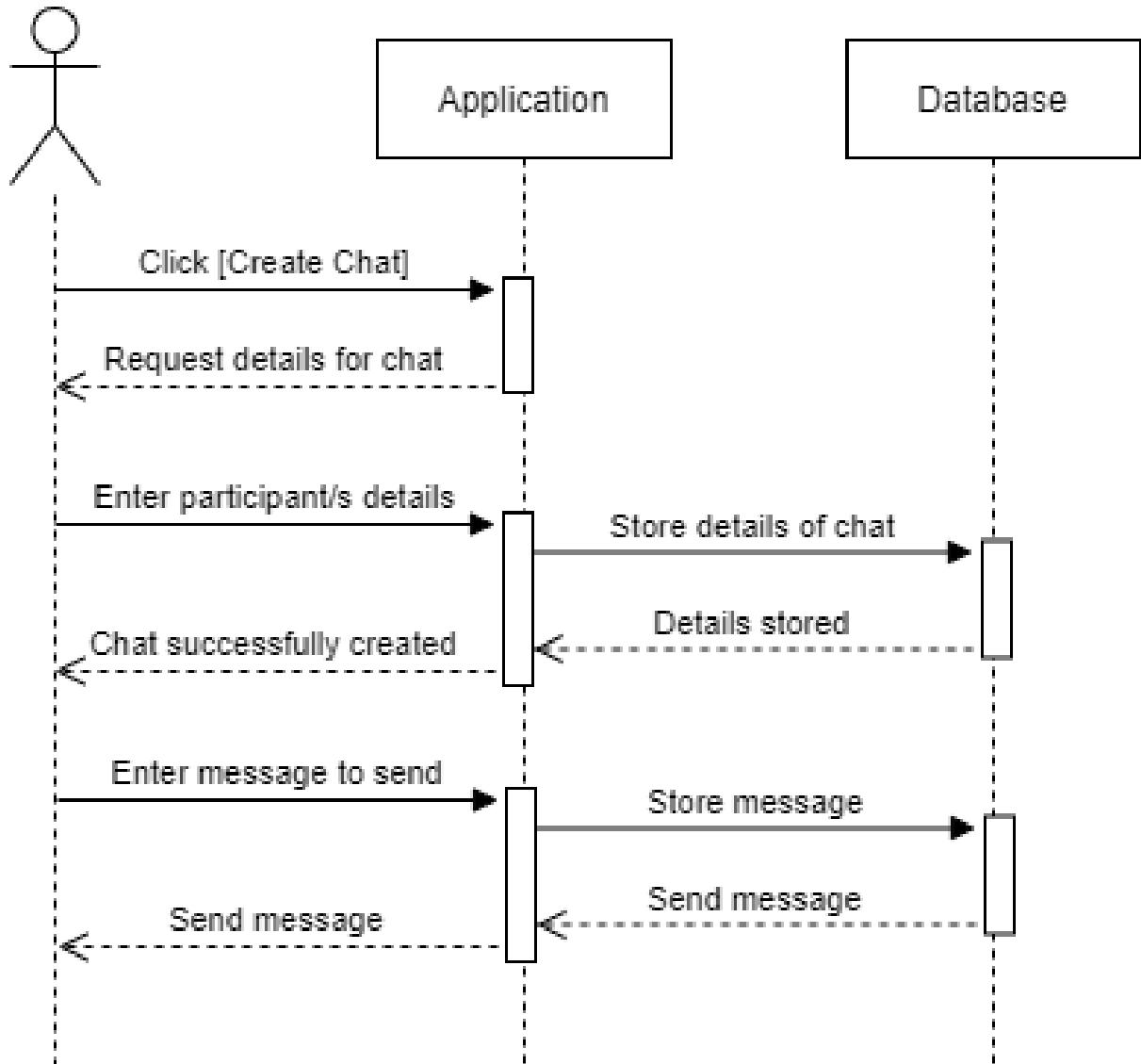
Epic story 2 (cont.)



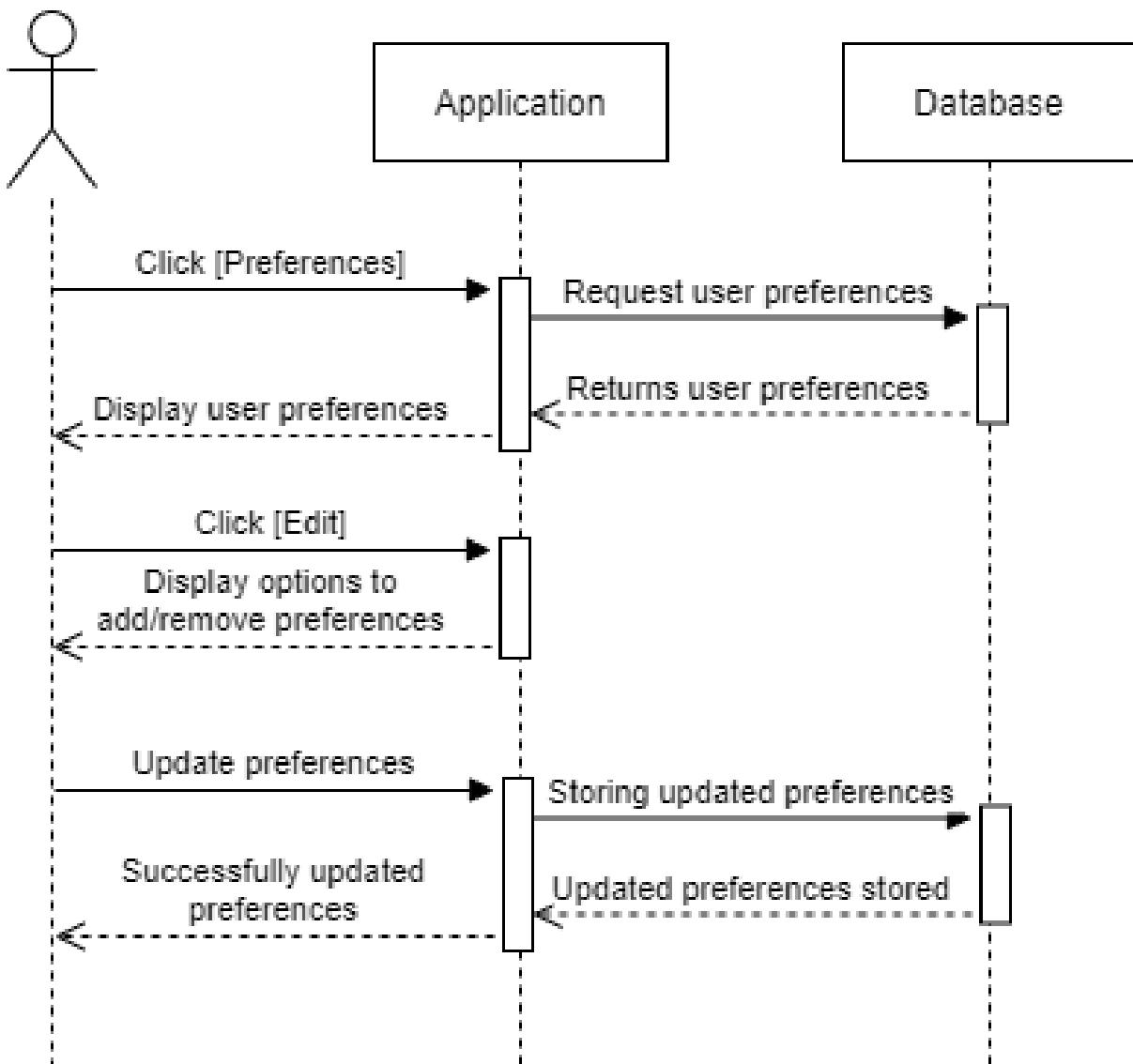
Epic Story 3: As a user, I want to be able to view past events by order of date so that I can compare details of plans and to look back on plans. (US7)



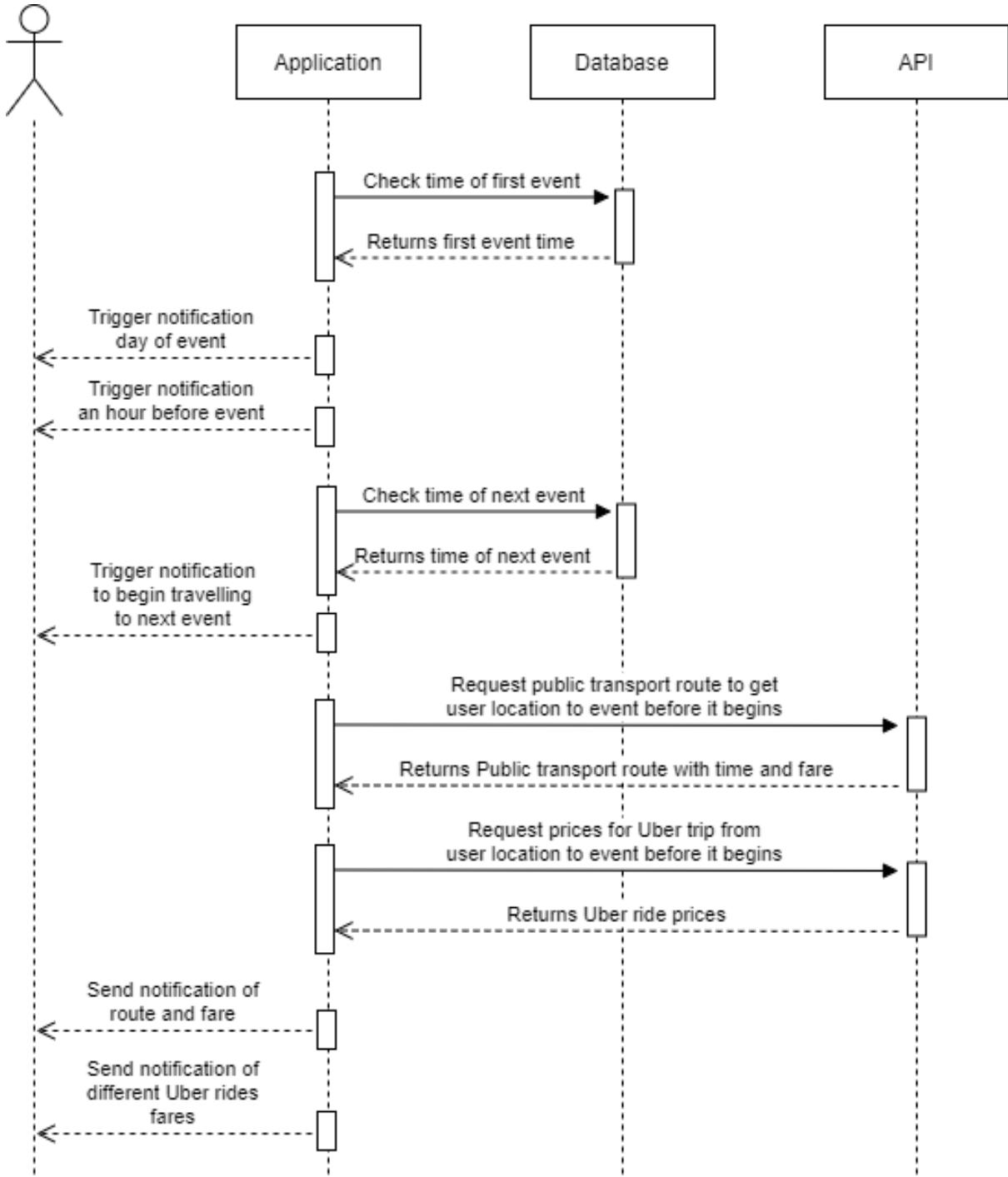
Epic Story 4: As a user, I want to be able to chat with other users within the app so that I can organise and coordinate events without needing to access multiple social media.
(US8)



Epic Story 5: As a user, I want to be able to add/update my personal preferences within my profile so that friends can see what my interests and preferences are. (US10)



Epic Story 6: As a user, I want to be sent notifications to my phone to alert me that I need to be at the next event, also being notified of public transport times and Uber fares (US 11, 12, 13)



Summary of Key Benefits and Design Implementation

A majority of user stories were able to be covered in the development of the mobile application. The few user stories which were not implemented were due to the limited time and complexity relative to the scope of the project, such as a booking system, as there were many variations to consider when booking an event, i.e. purchasing a movie ticket is different to reserving a time at a restaurant. The features/user stories not implemented in this project timeframe are potential additions to future iterations of the application.

Nonetheless, all other functionalities we intended to complete were implemented successfully. Our database, which stores various information of events, locations, attractions and venues is constantly updating by integrating external databases through using multiple API's. This contributes to the application's main feature of ease of use and convenience as it allows the auto-generated plans to be unique and personalised to suit user's preferences, allowing for a greater reach of experiences and limitless possibilities. However, to further satisfy consumers, these automated plans can be edited and configured manually if they have specific locations they want to visit.

In contrast to existing applications, PlusOne offers an in-app chat system that eases the planning process without the need for external social media. This guarantees that all participants within a plan can be contacted, e.g., a user may not have a colleague's phone number. Discussions of suggestions, conflicts and alternatives can be easily conducted with the app. In addition, the implementation of a plan history which consists of previous events a user participated in was designed for users to easily reorganise something similar, favourite a place they enjoyed or even recommend venues to friends.

Team Organization and Conclusion

Team Roles

To reduce conflict throughout the duration of the project, clear roles were allocated. The project required synergy between activities and tasks, also with the limited time and resources provided, having a clear role ensured accountability from each member to perform tasks adequately in their specified field.

<u>Team member</u>	<u>Roles</u>
Faris Jalal	System Architect, Front-end Developer
Rushil Pardasani	Scrum Master, Back-end Developer
Stephen Bletsas	Project Manager, Documentation Lead
Thiagaut Valcke	Dev Tester (front-end), Back-end Developer
Vishnu Birudavolu	Dev Tester (back-end), Front-end Developer

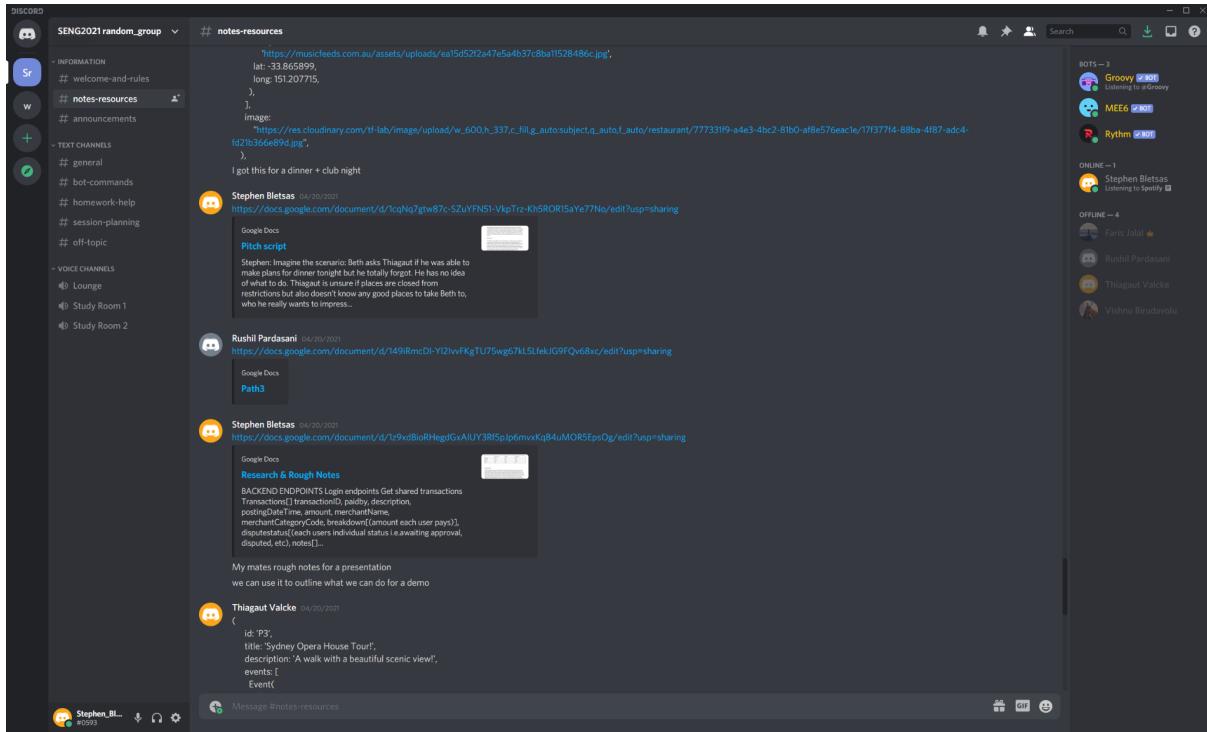
Communications

Using Microsoft Teams and Discord we conducted bi-weekly meetings, which were logically structured and organised by key scrum features. Extra meetings were held closer to deliverable dates to ensure work is completed to a high standard.

The agenda of these meetings/standups focused on:

- Tasks that have been completed since the previous meeting
- Difficulties/restrictions of tasks to be completed
- Next deliverable -> tasks that need to be completed by then
- Division of tasks
- Going through feedback from our mentor -> addressing how we will integrate feedback into the project
- Updating Trello board for completed and uncompleted tasks

In addition to team meetings, we made regular, informal communications via Discord for small clarifications, sharing resources, mitigating risks and for general contact between the project team.



We intended to abide by the five scrum values to maintain an effective agile project management throughout the course, which includes courage, commitment, focus, openness and respect. This resulted in an exceptionally functioning team, effectively enhancing the quality of our final mobile application and deliverables.

Tools for Project Management

To ensure that tasks were properly managed and organised, a Trello board was created. This allowed all members to be aware of tasks, their details and who is assigned to it.

Github was used to share, control and maintain code between members, allowing for development testers to also have access to other member's code.

One drive was used to share documents, research and materials to be used to aid the project and deliverables.

Reflection

Overall the project was a success in advancing our ability to work in a team and apply our knowledge to solve real-world problems. Total independence to build our own personal project allowed us to gain valuable skills to create solutions through understanding the process user requirements and technical design. This was aided by the multiple deliverables which focused on different aspects of the design techniques required to implement a successful project.

A critical aspect of this course was to develop our teamwork and leadership skills, with all members gaining valuable experience as we were continually communicating with each other, delegating tasks to ensure we were organised and meeting deadlines accordingly. Throughout the term, the team was becoming a cohesive unit as we became comfortable with one another, working together to achieve our goals. Dividing roles made tasks and results seem more attainable to achieve, diminishing the mountain of developing a mobile app. Despite delegating tasks/roles, all members were reviewing other members' work, to provide constructive feedback and to also gain knowledge and experience on all aspects of the project.

The very personalised weekly mentor sessions were effective in guiding our project by providing very specific and constructive feedback/suggestions for our project, for example, explaining why external API's are considered part of the backend, even though they don't communicate to any critical components such as the server or database. These tips, accompanied by our design and goals, allowed us to develop our application in an effective manner, as it reduced any blockers or limitations we would come across, effectively enhancing our learning experience.

An issue regarding our API selection became a major issue on the final development stage. Initially, we selected external interfaces based on popularity and well-known characteristics of the company, e.g., Zomato having the largest database of restaurants (1.5 million) globally. However, we had to change a couple APIs as we learnt there were many difficulties regarding the use of some such as paying for use, applying for developer keys, cessation of API service and not providing appropriate information, which in turn would limit our ability to add functionality to the mobile app. This issue

generated some tension within the team, but after consulting each other in an emergency meeting the entire team came into agreement in terms of alternatives and came to a resolution.

An area for improvement to improve productivity during the development phase would have been to put greater effort in the initial research of API alternatives during the software architecture stage, before we began implementing the back-end.

Despite any issues, the project was an overall success which provided the perfect platform to understand the intricate process of software engineering and gaining experience on designing and developing a mobile application in a team environment.