

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  
**SINGAPORE**

**CZ2006: Software Engineering**

**Lab Group: BCS3**

**Team: 5 – DMILF**

**Project Name:**

**explore<sup>+</sup>  
SG**

**Team Members:**

|  |           |
|--|-----------|
| Deionna Chee Rui Ping .....            | U2021449D |
| Faith Lee Kai Ling .....               | U2022307C |
| Foo Yong He Herman .....               | U1940005A |
| Hor Chia Yee, Lara .....               | U2021546F |
| Magdeline Ng Xuan Lynn .....           | U2020797L |
| Muhammad Ihsan Bin Mohammad Azmi ..... | U2021160B |



## Table of Contents

|                                    |           |
|------------------------------------|-----------|
| <b>Table of Contents</b>           | <b>2</b>  |
| <b>Project Description</b>         | <b>4</b>  |
| Mission Statement                  | 4         |
| Scope                              | 4         |
| Target audience                    | 4         |
| Assumptions and Constraints        | 4         |
| Initial UI Mockups                 | 5         |
| <b>Functional Requirements</b>     | <b>5</b>  |
| Use Case Descriptions              | 9         |
| Use Case Diagram                   | 33        |
| Class Diagram                      | 34        |
| Sequence Diagrams                  | 35        |
| Registration                       | 35        |
| Login (Username)                   | 36        |
| Login (Google Sign In)             | 37        |
| Login (Email)                      | 38        |
| Retrieve Favourites                | 39        |
| Recommendations                    | 40        |
| Search for places                  | 41        |
| Update Tracker                     | 42        |
| Update Interests                   | 43        |
| Create/Update Review               | 43        |
| Send Invite                        | 44        |
| Accept Invite                      | 45        |
| Decline Invite                     | 46        |
| Change profile picture             | 46        |
| Dialog Map                         | 47        |
| <b>Non-Functional Requirements</b> | <b>48</b> |
| <b>Interface Requirements</b>      | <b>48</b> |
| <b>Architecture Design</b>         | <b>49</b> |
| <b>Data Dictionary</b>             | <b>50</b> |
| <b>Testing</b>                     | <b>51</b> |
| Black Box Testing                  | 51        |
| Registration Test Cases            | 51        |
| Username Login Test Cases          | 52        |
| Email Login Test Cases             | 53        |



|   |           |
|---|-----------|
| Google Login Test Cases                 | 53        |
| Update Tracker Test Cases               | 54        |
| Send invitation Test Cases              | 54        |
| White Box Testing – Control Flow Graphs | 55        |
| Managing User's Interests               | 55        |
| User's Review                           | 56        |
| Invitee Accepts/Rejects Invite          | 56        |
| Search for places                       | 57        |
| Favourite                               | 58        |
| <b>Traceability</b>                     | <b>59</b> |
| <b>Appendix</b>                         | <b>60</b> |



# 1. Project Description

## 1.1. Mission Statement

We aimed to develop an app that suggests things to do or places to visit in Singapore. Hence, we created a platform for people to find places and invite others along with them, then rate and review those places for other users to view. Overall, we hope to allow people to discover unique places that they would otherwise have not known about, as well as help them decide where to go on their next outing.

## 1.2. Scope

Our app allows users to search and get recommendations for new places to go to based on their interests and current location. Every user can create an account free of charge, which can be used to invite fellow users to explore new areas and keep track of their already visited places. User profiles are highly customisable, as users are able to change anything from their display picture to their interests, providing a truly personalised experience. The intuitive but aesthetic UI makes for a pleasant experience in finding out more about these locations. Upon visiting, users are also able to leave a review of the place, giving other users a better picture of their expected experience.

## 1.3. Target audience

Our main target audience consists of couples and friends. Harnessing the power of technology, we aim to be able to bring these people closer together by making the process of connecting seamless and easy.

## 1.4. Assumptions and Constraints

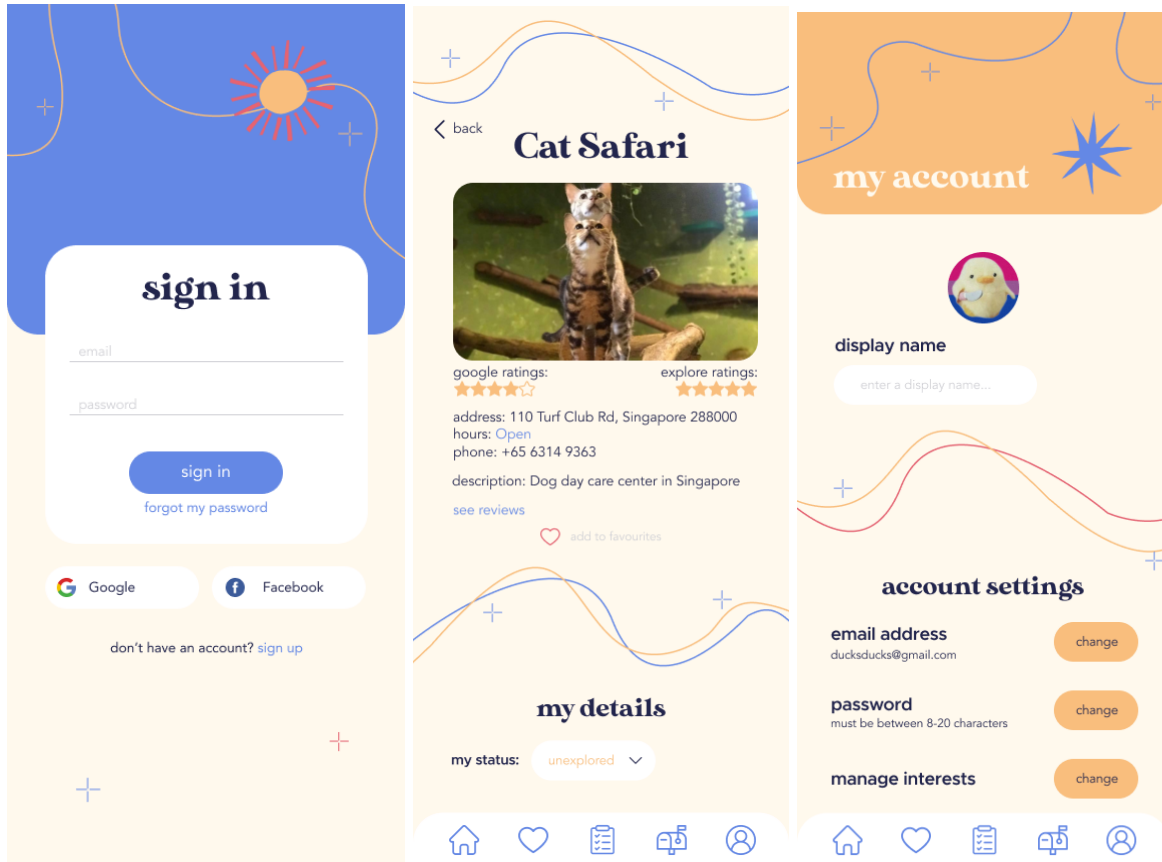
For our app to work, it is assumed that users have GPS enabled, as well as a stable internet connection at all times during use. This is due to the app's heavy reliance on the user's location, as well as needing to fetch currently available data to ensure that the information displayed is up-to-date.

Since it uses the GooglePlaces API, places listed and suggested by our app are limited to those registered on Google. Furthermore, the interests selectable by users are limited to the categories made available by the GooglePlaces API.

Finally, our app is currently only available in the English language.

## 1.5. Initial UI Mockups

To visualise the look and feel, we made use of the free online software Figma to create mockups of the pages and functions that would be available on our app. Figma also supports linking between pages to simulate actual interactions in the app, allowing us to get a better picture of what the final product would be like.



*Fig 1: Initial UI mockups for exploreSG*

In our UI design, we aimed to make it as aesthetically pleasing but as intuitive as possible, providing users with a calm, easygoing feel during use. Through the mockups, we were able to get a good visualisation of how all the functional requirements would come together in the final product (full UI mockups can be found under the link provided in the Appendix).



## 2. Functional Requirements

1. Registration
  - 1.1. When the user first logs in, he must be able to register a new account through our system registration
  - 1.2. Required user information:
    - 1.2.1. Username
    - 1.2.2. First Name
    - 1.2.3. Last Name
    - 1.2.4. Email address
    - 1.2.5. Password
  - 1.3. The user must also have the option to register a new account via their Google account instead
  - 1.4. The system must validate username availability
  - 1.5. The system must validate that all required fields have been entered
  - 1.6. The system must validate that the password is at least 8 characters
  - 1.7. The user must be able to verify their email
    - 1.7.1. The system must send a verification email to the registered email address
    - 1.7.2. The system must validate that email has been verified
2. Login
  - 2.1. The user must be able to login using their username and password
  - 2.2. The user must also have the option to login via email address and password
  - 2.3. The user must also have the option to login via their Google account
  - 2.4. The system must validate that both fields have been entered
  - 2.5. The system must authenticate the login through our account database
  - 2.6. If login credentials are incorrect, the system will display an error message
  - 2.7. If login is successful, the system will redirect the user to the home page
  - 2.8. If the user forgets his password, he must be able to request for change of password
    - 2.8.1. The system must be able to send an email to the user for change of password
3. Home Page
  - 3.1. The system must allow users to search for locations with filters.
    - 3.1.1. The system must have the option to search using keyword, or choose place type from drop down list
  - 3.2. The system must be able to display a list of recommended places.
    - 3.2.1. Displayed under the 'explore' section.
    - 3.2.2. Recommended places include places based on the user's interests.
      - 3.2.2.1. Recommended places will be sorted by ascending distance.
      - 3.2.2.2. If the user has no selected interests, the system will recommend random places sorted by ascending distance.
4. Searching
  - 4.1. The user must be able to choose to search via dropdown list or search bar.



- 4.2. The user must be able to choose the condition to filter the places that are returned from the search.
  - 4.2.1. Filters include:
    - 4.2.1.1. Distance
      - 4.2.1.1.1. The user will set the maximum distance using the slider bar.
      - 4.2.1.1.2. The system will retrieve the location of the user.
      - 4.2.1.1.3. The system will return a list of places based on ascending distance from the user's location.
    - 4.2.1.2. Ratings
      - 4.2.1.2.1. The user sets the range of rating values.
      - 4.2.1.2.2. The system will return a list of places based on ascending ratings within the range of rating values that the user has set.
    - 4.2.1.3. Price
      - 4.2.1.3.1. The user will set the range of price values.
      - 4.2.1.3.2. The system will return a list of places based on ascending prices within the range of price values that the user has set.
- 5. Favourites
  - 5.1. The system must allow the user to add a place to their favourites from the home, place details and search results pages.
    - 5.1.1. The system must add a place to the user's favourites when the heart button is clicked.
  - 5.2. The system must allow the user to remove a place from their favourites from the home, place details, search results and favourites pages.
    - 5.2.1. The system must remove a place from the user's favourites when the heart button is clicked.
  - 5.3. On the clicking of the heart button, the system must save the changes to the user's list of favourite places on the user database.
- 6. Tracker
  - 6.1. The system must add places that the user selects as 'to explore' to the tracker.
  - 6.2. The system must reflect the users that have been invited and have accepted the invitation in the place details.
  - 6.3. After the date and time of the place details has lapsed, the system will shift the place from 'to explore' section to 'explored' section.
- 7. Place Details
  - 7.1. The system must display important place details information, including:
    - 7.1.1. Place name
    - 7.1.2. Place image
    - 7.1.3. Distance from user's current location
    - 7.1.4. Price
    - 7.1.5. Google and exploreSG ratings
    - 7.1.6. Place address



- 7.1.7. Place operation status (Open/Closed)
  - 7.2. The user must be able to favourite a place from the place details page.
  - 7.3. The user must be able to change their personal place status from the place details page.
    - 7.3.1. The user must be able to invite others when the 'to explore' option is selected.
      - 7.3.1.1. The user must be able to select a date, time, and list of usernames for the invitation.
      - 7.3.1.2. The user must be able to select only a date and time for a solo exploration.
    - 7.3.2. The user must be able to leave a rating and review when the 'explored' option is selected.
- 8. Reviews Page
  - 8.1. The system must display all existing 'explorer reviews' for a particular place for all users to view.
  - 8.2. The system must display the user's display name, profile picture, rating, and review for each unique review.
- 9. Profile
  - 9.1. The system must display the following information:
    - 9.1.1. Display name
    - 9.1.2. Profile picture
    - 9.1.3. Username
  - 9.2. The system must allow users to change the following:
    - 9.2.1. Display name
    - 9.2.2. Profile picture
    - 9.2.3. Username
    - 9.2.4. Password
    - 9.2.5. Interests
      - 9.2.5.1. The system must navigate to the user's interests page when the user clicks the manage interests button
    - 9.2.6. The system must allow the user to logout by clicking the 'logout' button under the account settings section.
- 10. Interests
  - 10.1. The system must allow the user to change their interests from the list of interests provided by GooglePlaces API.
    - 10.1.1. The system must select an interest when an unselected interest is clicked
    - 10.1.2. The system must deselect an interest when an selected interest is clicked
  - 10.2. The system must save the user's interests to the user collection in our database when the user clicks the 'save changes' button.
- 11. Retrieve Information
  - 11.1. Places Information (Places API)
    - 11.1.1. The system must be able to retrieve relevant information from Places API
      - 11.1.1.1. Information include:
        - 11.1.1.1.1. Place name





- 11.1.1.1.2. Place address
- 11.1.1.1.3. Place photo
- 11.1.1.1.4. Distance from user's current location
- 11.1.1.1.5. Place price
- 11.1.1.1.6. Place Google rating
- 11.1.1.1.7. Place opening status
- 11.1.1.1.8. Place type
- 11.1.2. The user must be able to retrieve relevant information for recommended places or places of interest
  - 11.1.2.1. Information include:
    - 11.1.2.1.1. Place name
    - 11.1.2.1.2. Place address
    - 11.1.2.1.3. Place photo
    - 11.1.2.1.4. Distance from user's current location
    - 11.1.2.1.5. Place price
    - 11.1.2.1.6. Place Google rating
    - 11.1.2.1.7. Place opening status
    - 11.1.2.1.8. Place type

## 2.1. Use Case Descriptions

|                |                            |                    |           |
|----------------|----------------------------|--------------------|-----------|
| Use Case ID:   | 1                          |                    |           |
| Use Case Name: | Register for a new account |                    |           |
| Created By:    | Deionna Chee               | Last Updated By:   | Lara Hor  |
| Date Created:  |                            | Date Last Updated: | 17/4/2022 |

|                 |  |
|-----------------|--|
| Actor:          | User   |
| Description:    | Registration for a new user account.   |
| Preconditions:  | <ol style="list-style-type: none"> <li>1. User's account must not already exist in the database.</li> <li>2. User's device must be connected to WiFi/Mobile Data.</li> </ol> |
| Postconditions: | <ol style="list-style-type: none"> <li>1. A verification email will be sent to the user's registered email address.</li> </ol>   |
| Priority:       | Medium   |



|                       |   |
|-----------------------|---|
| Frequency of Use:     | Rarely  |
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. User enters a valid username.</li> <li>2. User selects the "check username" button.</li> <li>3. System confirms username is available.</li> <li>4. User enters a valid first name, last name, email address and password.</li> <li>5. User selects the "register" button.</li> <li>6. System will send verification email to the registered email address.</li> <li>7. User verifies their email and successfully registers their account</li> </ol>  |
| Alternative Flows:    | <p>AF-S2: System detects an invalid (existing) username.</p> <ol style="list-style-type: none"> <li>1. System displays an error message "Username taken. Change to another unique username!"</li> <li>2. Return to Step 1.</li> </ol> <p>AF-S5: System detects any field is empty.</p> <ol style="list-style-type: none"> <li>1. System displays error message "_____ cannot be empty"</li> <li>2. Return to step 4.</li> </ol> <p>AF-S5: System detects email is not in the correct format.</p> <ol style="list-style-type: none"> <li>3. System displays error message "Enter a valid email"</li> <li>4. Return to step 4.</li> </ol> <p>AF-S5: System detects password is less than 8 characters.</p> <ol style="list-style-type: none"> <li>5. System displays error message "Password must contain at least 8 characters"</li> <li>6. Return to step 4.</li> </ol> |
| Exceptions:           | -   |
| Includes:             | <ol style="list-style-type: none"> <li>1. Validate the account availability.</li> </ol>   |
| Special Requirements: | -   |
| Assumptions:          | -   |
| Notes and Issues:     | -   |

|              |   |
|--------------|---|
| Use Case ID: | 2 |
|--------------|---|



|                |                             |                    |           |
|----------------|-----------------------------|--------------------|-----------|
| Use Case Name: | Register via Google account |                    |           |
| Created By:    | Deionna Chee                | Last Updated By:   | Lara Hor  |
| Date Created:  |                             | Date Last Updated: | 17/4/2022 |

|                    |   |
|--------------------|---|
| Actor:             | User  |
| Description:       | Registration for a new user account via the user's Google account.  |
| Preconditions:     | <ol style="list-style-type: none"> <li>1. User's account must not already exist in the database.</li> <li>2. User's device must be connected to WiFi/Mobile Data.</li> <li>3. User's device is already logged in to their Google account.</li> </ol>  |
| Postconditions:    | -   |
| Priority:          | Medium  |
| Frequency of Use:  | Rarely  |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User selects the "sign in with google" button.</li> <li>2. System will validate the account availability with respect to the database.</li> <li>3. User will be prompted to grant permission for System to access User's Google account.</li> <li>4. User will grant the permission by selecting the "Continue" button</li> <li>5. User chooses the google account they want to use.</li> <li>6. System will extract User data from User's Google account and save them in the database.</li> <li>7. User will be logged in successfully using their Google account.</li> </ol> |
| Alternative Flows: | <p>AF-S2: System detects user's mobile is not logged in to the Google account.</p> <ol style="list-style-type: none"> <li>1. System will prompt User to enter their credentials to login to their Google account.</li> <li>2. User provides their credentials to login.</li> <li>3. Return to Step 5.</li> </ol>  |
| Exceptions:        | -   |



|                       |                                  |
|-----------------------|----------------------------------|
| Includes:             | 1. Validate account availability |
| Special Requirements: | -                                |
| Assumptions:          | -                                |
| Notes and Issues:     | -                                |

|                |                               |                    |           |
|----------------|-------------------------------|--------------------|-----------|
| Use Case ID:   | 3                             |                    |           |
| Use Case Name: | Validate account availability |                    |           |
| Created By:    | Lara Hor                      | Last Updated By:   | Lara Hor  |
| Date Created:  |                               | Date Last Updated: | 17/4/2022 |

|                    |   |
|--------------------|---|
| Actor:             | System  |
| Description:       | System validates account availability after the user verifies their email, and creates the account in the database.   |
| Preconditions:     | <ol style="list-style-type: none"> <li>1. User's device must be connected to Wifi/Mobile data.</li> <li>2. Account cannot already exist in the database.</li> </ol>   |
| Postconditions:    | <ol style="list-style-type: none"> <li>1. Account is successfully created in the system's database.</li> <li>2. User is able to login to the system using the created account.</li> </ol>   |
| Priority:          | Medium  |
| Frequency of Use:  | Rarely  |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User receives the verification email.</li> <li>2. User clicks to verify their email via Firebase.</li> <li>3. System checks that the email is verified and does not exist in the database.</li> <li>4. System creates the account in the database.</li> </ol> |
| Alternative Flows: |   |



|                       |   |
|-----------------------|---|
| Exceptions:           | - |
| Includes:             | - |
| Special Requirements: | - |
| Assumptions:          | - |
| Notes and Issues:     |   |

|                |                             |                    |           |
|----------------|-----------------------------|--------------------|-----------|
| Use Case ID:   | 4                           |                    |           |
| Use Case Name: | Request for password change |                    |           |
| Created By:    | Ihsan                       | Last Updated By:   | Lara Hor  |
| Date Created:  |                             | Date Last Updated: | 17/4/2022 |

|                    |  |
|--------------------|--|
| Actor:             | User   |
| Description:       | Allow User to reset or change their password   |
| Preconditions:     | 1. User's device must be connected to WiFi/Mobile Data   |
| Postconditions:    | 1. User's password has to be updated in the database.<br>2. User must be able to login with new password   |
| Priority:          | Medium   |
| Frequency of Use:  | Rarely   |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User enters registered email and submits request</li> <li>2. System checks if there exists an account with that email on the database</li> <li>3. System sends email for User to reset password to User's email address</li> <li>4. User resets their password via Firebase</li> <li>5. System saves users new password in the database</li> </ol> |
| Alternative Flows: | -  |



|                       |   |
|-----------------------|---|
| Exceptions:           | - |
| Includes:             | - |
| Special Requirements: | - |
| Assumptions:          | - |
| Notes and Issues:     | - |

|                |                          |                    |              |
|----------------|--------------------------|--------------------|--------------|
| Use Case ID:   | 5                        |                    |              |
| Use Case Name: | Verify Login Credentials |                    |              |
| Created By:    | Herman                   | Last Updated By:   | Deionna Chee |
| Date Created:  |                          | Date Last Updated: | 17/4/2022    |

|                   |  |
|-------------------|--|
| Actor:            | System   |
| Description:      | Account authentication process through username and password.  |
| Preconditions:    | <ol style="list-style-type: none"> <li>1. User's account must already exist in the database.</li> <li>2. User's device must be connected to WiFi/Mobile Data</li> </ol>  |
| Postconditions:   | <ol style="list-style-type: none"> <li>1. The user is able to add items to their list of favourites.</li> <li>2. The user is able to retrieve items from their list of favourites.</li> <li>3. The user is able to remove items to their list of favourites.</li> <li>4. The user is able to use the to-go list feature.</li> <li>5. The user is able to add and remove items to the to-go list list.</li> <li>6. The user is able to send recommendations to others (via email).</li> </ol> |
| Priority:         | High   |
| Frequency of Use: | Rarely   |
| Flow of Events:   | <ol style="list-style-type: none"> <li>1. User tap login via username.</li> </ol>  |



|                       |   |
|-----------------------|---|
|                       | <ol style="list-style-type: none"> <li>2. User enters their credentials in the login interface.</li> <li>3. User's select the login button.</li> <li>4. System validates the account by checking the user's credentials with the database.</li> <li>5. System authenticates the user to login successfully.</li> </ol>  |
| Alternative Flows:    | <p>AF-S1: User tap login via Google account.</p> <ol style="list-style-type: none"> <li>1. Return to Step 4.</li> </ol> <p>AF-S3: System detects empty username or password fields.</p> <ol style="list-style-type: none"> <li>1. System displays an error message "Username or password fields cannot be empty."</li> <li>2. User filled up the required field(s) for username and password.</li> <li>3. Return to Step 3.</li> </ol> <p>AF-S4: User enters the wrong credentials</p> <ol style="list-style-type: none"> <li>1. System displays an error message "The password is invalid or the user does not have a password"</li> <li>2. Return to Step 2.</li> </ol> |
| Exceptions:           | -   |
| Includes:             | -   |
| Special Requirements: | -   |
| Assumptions:          | - Database can be referred to both System's database and Google's database.   |
| Notes and Issues:     | -   |

|                |        |                    |           |
|----------------|--------|--------------------|-----------|
| Use Case ID:   | 6      |                    |           |
| Use Case Name: | Search |                    |           |
| Created By:    | Ihsan  | Last Updated By:   | Magdeline |
| Date Created:  |        | Date Last Updated: | 17/4/2022 |

|        |      |
|--------|------|
| Actor: | User |
|--------|------|



|                       |  |
|-----------------------|--|
| Description:          | Search for places of interest/Activity spots using filters   |
| Preconditions:        | <ol style="list-style-type: none"> <li>1. User's device must be connected to Wifi/Mobile data.</li> <li>2. User's location services must be enabled.</li> </ol>  |
| Postconditions:       | Based on user's search conditions and specified filters, a list of places will be displayed  |
| Priority:             | High   |
| Frequency of Use:     | Often  |
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. User chooses "keyword search" using slider bar.</li> <li>2. User enters text to be queried into the search bar</li> <li>3. User selects the condition to filter by in "filter by" dropdown bar. <ol style="list-style-type: none"> <li>a. Available filters: Price, Distance radius, Ratings</li> </ol> </li> <li>4. User clicks on the "go" button.</li> <li>5. System displays a list of places based on user's input and specified filter.</li> </ol>   |
| Alternative Flows:    | <p>AF-S1: User chooses "dropdown list" using slider bar.</p> <ol style="list-style-type: none"> <li>1. User selects a place from the dropdown list displayed.</li> </ol> <p>AF-S2: User does not enter text in search bar.</p> <ol style="list-style-type: none"> <li>1. User clicks on the "go" button.</li> <li>2. System displays a list of places based on "filter by" condition, returning places within 10km of the user's location.</li> </ol> <p>AF-S2: User does not select anything from the "filter by" dropdown bar.</p> <ol style="list-style-type: none"> <li>1. By default, results are sorted by prominence if no filters/sort are selected</li> </ol> |
| Exceptions:           | -  |
| Includes:             | <ol style="list-style-type: none"> <li>1. Retrieve place information</li> <li>2. Retrieve user location</li> <li>3. Verify login credentials</li> </ol>  |
| Special Requirements: | -  |
| Assumptions:          | -  |
| Notes and Issues:     | -  |





|                |                |                    |           |
|----------------|----------------|--------------------|-----------|
| Use Case ID:   | 7              |                    |           |
| Use Case Name: | Review a place |                    |           |
| Created By:    | Faith Lee      | Last Updated By:   | Faith Lee |
| Date Created:  |                | Date Last Updated: | 16/4/2022 |

|                    |  |
|--------------------|--|
| Actor:             | User   |
| Description:       | Leave a rating and review for a place after visiting that place.   |
| Preconditions:     | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> <li>3. User's place status must be set to 'explored' (either user set or the time of invitation has lapsed)</li> </ol>  |
| Postconditions:    | <ol style="list-style-type: none"> <li>1. The review can later be viewed on the place details page</li> <li>2. The review can also be found on the 'explorer reviews' page</li> </ol>  |
| Priority:          | Medium   |
| Frequency of Use:  | Moderate   |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User chooses a rating and types out a review</li> <li>2. User presses the submit button</li> </ol>   |
| Alternative Flows: | <p>AF-S1: System detects an empty rating field.</p> <ol style="list-style-type: none"> <li>1. System displays an error message "please fill out rating &amp; review field".</li> <li>2. Return to Step 1.</li> </ol> <p>AF-S2: System detects an empty review field.</p> <ol style="list-style-type: none"> <li>1. System displays an error message "please fill out rating &amp; review field".</li> <li>2. Return to Step 1</li> </ol> <p>AF-S3: System detects the user has already made a review.</p> <ol style="list-style-type: none"> <li>3. System updates the previous review.</li> </ol> |
| Exceptions:        | -  |
| Includes:          | <ol style="list-style-type: none"> <li>1. Verify login credentials</li> </ol>  |



|                       |   |
|-----------------------|---|
| Special Requirements: | - |
| Assumptions:          | - |
| Notes and Issues:     | - |

|                |                        |                    |           |
|----------------|------------------------|--------------------|-----------|
| Use Case ID:   | 8                      |                    |           |
| Use Case Name: | Retrieve place reviews |                    |           |
| Created By:    | Faith Lee              | Last Updated By:   | Faith Lee |
| Date Created:  |                        | Date Last Updated: | 16/4/2022 |

|                       |  |
|-----------------------|--|
| Actor:                | System   |
| Description:          | Display all explorer reviews for a particular place  |
| Preconditions:        | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol>             |
| Postconditions:       | -  |
| Priority:             | Medium   |
| Frequency of Use:     | Often  |
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. User clicks on the 'see explorer reviews' link</li> <li>2. System returns all existing reviews</li> </ol>                        |
| Alternative Flows:    | AF-S1: System detects that there are no reviews for the place.<br><ol style="list-style-type: none"> <li>3. System displays the message "no reviews to show..."</li> </ol> |
| Exceptions:           | -  |
| Includes:             | <ol style="list-style-type: none"> <li>1. Retrieve place information</li> <li>2. Verify login credentials</li> </ol>   |
| Special Requirements: | -  |



|                   |  |
|-------------------|--|
| Assumptions:      | All users who have created a review exist, such that all necessary user information can be retrieved, based on user UID linked to each review. |
| Notes and Issues: | -  |

|                |                                |                    |              |
|----------------|--------------------------------|--------------------|--------------|
| Use Case ID:   | 9                              |                    |              |
| Use Case Name: | Add place to user's favourites |                    |              |
| Created By:    | Deionna Chee                   | Last Updated By:   | Deionna Chee |
| Date Created:  |                                | Date Last Updated: | 16/4/2022    |

|                    |   |
|--------------------|---|
| Actor:             | User  |
| Description:       | To add location to user's list of favourite places.   |
| Preconditions:     | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol>  |
| Postconditions:    | <ol style="list-style-type: none"> <li>1. User's selected place will be added into the list of favourite places and saved into the user database.</li> </ol>  |
| Priority:          | High  |
| Frequency of Use:  | Often   |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User navigates to the 'Place Details' page.</li> <li>2. User clicks the heart outline button to add the selected place to their favourites.</li> <li>3. Heart outline button will turn into a filled heart button.</li> <li>4. System will save the user's favourite place into the user database.</li> </ol>                                 |
| Alternative Flows: | <p>AF-S1: User navigates to the 'Home' page.</p> <ol style="list-style-type: none"> <li>1. User scrolls to find a place to add to favourites.</li> <li>2. Return to step 2.</li> </ol> <p>AF-S1: User navigates to the 'Search Results' page.</p> <ol style="list-style-type: none"> <li>1. User scrolls to find a place to add to favourites.</li> <li>2. Return to step 2.</li> </ol> |



|                       |  |
|-----------------------|--|
| Exceptions:           | -  |
| Includes:             | <ol style="list-style-type: none"> <li>1. Verify Login Credentials</li> <li>2. Retrieve place information</li> </ol> |
| Special Requirements: | -  |
| Assumptions:          | -  |
| Notes and Issues:     | -  |

|                |                               |                    |              |
|----------------|-------------------------------|--------------------|--------------|
| Use Case ID:   | 10                            |                    |              |
| Use Case Name: | Remove user's favourite items |                    |              |
| Created By:    | Deionna Chee                  | Last Updated By:   | Deionna Chee |
| Date Created:  |                               | Date Last Updated: | 16/4/2022    |

|                   |   |
|-------------------|---|
| Actor:            | User  |
| Description:      | To remove user's favourite items.   |
| Preconditions:    | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol>  |
| Postconditions:   | <ol style="list-style-type: none"> <li>1. User's selected place will be removed from the list of favourite places and saved into the database.</li> </ol>   |
| Priority:         | High  |
| Frequency of Use: | Moderate  |
| Flow of Events:   | <ol style="list-style-type: none"> <li>1. User navigates to the 'Place Details' page.</li> <li>2. User clicks the filled heart button to remove the selected place from their favourites.</li> <li>3. Filled heart button will turn into a heart outline button.</li> <li>4. System will remove the user's favourite place from the user database and save the change.</li> </ol> |



|                       |  |
|-----------------------|--|
| Alternative Flows:    | <p>AF-S1: User navigates to the 'Home' page.</p> <ol style="list-style-type: none"> <li>1. User scrolls to find a place to add to favourites.</li> <li>2. Return to step 2.</li> </ol> <p>AF-S1: User navigates to the 'Search Results' page.</p> <ol style="list-style-type: none"> <li>1. User scrolls to find a place to add to favourites.</li> <li>2. Return to step 2.</li> </ol> <p>AF-S1: User navigates to the 'Favourites' page.</p> <ol style="list-style-type: none"> <li>1. User scrolls to find a place to add to favourites.</li> <li>2. User clicks the filled heart button to remove the selected place from their favourites.</li> <li>3. System will remove the selected place from the list of favourite places displayed on the 'Favourites' page.</li> <li>4. Return to step 4.</li> </ol> |
| Exceptions:           | -  |
| Includes:             | 1. Verify Login Credentials  |
| Special Requirements: | -  |
| Assumptions:          | -  |
| Notes and Issues:     | -  |

|                |  |                    |              |
|----------------|--|--------------------|--------------|
| Use Case ID:   | 11                                       |                    |              |
| Use Case Name: | Retrieve user's list of favourite places |                    |              |
| Created By:    | Deionna Chee                             | Last Updated By:   | Deionna Chee |
| Date Created:  |  | Date Last Updated: | 16/4/2022    |

|                |  |
|----------------|--|
| Actor:         | User   |
| Description:   | To retrieve user's list of favourite places.   |
| Preconditions: | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol> |



|                       |   |
|-----------------------|---|
| Postconditions:       | 1. A list of user's favourite places will be displayed on the 'Favourites' page.  |
| Priority:             | High  |
| Frequency of Use:     | Often   |
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. User navigates to the 'Favourites' page.</li> <li>2. System will retrieve the list of user's favourite places from the database.</li> <li>3. System will display a list of user's favourite places on the 'Favourites' page.</li> </ol> |
| Alternative Flows:    | AF-S2: User has no favourite places<br>1. System displays "No favourites to show"   |
| Exceptions:           | -   |
| Includes:             | <ol style="list-style-type: none"> <li>1. Verify Login Credentials</li> <li>2. Retrieve place information</li> </ol>  |
| Special Requirements: | -   |
| Assumptions:          | -   |
| Notes and Issues:     | -   |

|                |  |                    |              |
|----------------|--|--------------------|--------------|
| Use Case ID:   | 12                                       |                    |              |
| Use Case Name: | Send invite to another user/other users. |                    |              |
| Created By:    | Deionna Chee                             | Last Updated By:   | Deionna Chee |
| Date Created:  |  | Date Last Updated: | 16/4/2022    |

|                |  |
|----------------|--|
| Actor:         | User   |
| Description:   | To send invite to another user   |
| Preconditions: | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol> |



|                    |   |
|--------------------|---|
| Postconditions:    | <ol style="list-style-type: none"> <li>1. User's selected place will be added to the list of 'toExplore' items and saved into the database.</li> <li>2. Invited user(s) receive(s) invitation in inbox.</li> </ol>  |
| Priority:          | High  |
| Frequency of Use:  | Moderate  |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User navigates to the 'Place Details' page.</li> <li>2. User clicks the dropdown menu and changes 'my status' to 'to explore'.</li> <li>3. User clicks on the calendar icon and selects a date.</li> <li>4. User clicks on the clock icon and selects a time.</li> <li>5. User enters another user's username in the 'friends' field and clicks the 'add' button.</li> <li>6. User repeats step 5 until all friends are added.</li> <li>7. User clicks the 'invite' button to invite friends.</li> <li>8. System adds an 'invite' item to the invited user's inbox, containing the place, date and time details, and saves it to the database.</li> <li>9. System adds a 'toExplore' item to the user's 'Tracker' page, containing the place, date and time details, and saves it to the database.</li> </ol>   |
| Alternative Flows: | <p>AF-S3: User selects an invalid date (date before current time)</p> <ol style="list-style-type: none"> <li>1. System shows an 'invalid date and time' error, prompting the user to select a valid date.</li> <li>2. Return to step 3.</li> </ol> <p>AF-S4: User selects an invalid time (time before current time)</p> <ol style="list-style-type: none"> <li>1. System shows an 'invalid date and time' error, prompting the user to select a valid date.</li> <li>2. Return to step 4.</li> </ol> <p>AF-S5: User keys in invalid username and clicks the 'add' button.</p> <ol style="list-style-type: none"> <li>1. System shows an 'invalid username' error message, prompting the user to key in a valid username.</li> <li>2. Return to step 5.</li> </ol> <p>AF-S5: User keys in own username and clicks the 'add' button.</p> <ol style="list-style-type: none"> <li>1. System shows a 'you cannot invite yourself' error message.</li> <li>2. Return to step 5.</li> </ol> |
| Exceptions:        | -   |
| Includes:          | <ol style="list-style-type: none"> <li>1. Verify Login Credentials</li> </ol>   |



|                       |   |
|-----------------------|---|
| Special Requirements: | - |
| Assumptions:          | - |
| Notes and Issues:     | - |

|                |                |                    |              |
|----------------|----------------|--------------------|--------------|
| Use Case ID:   | 13             |                    |              |
| Use Case Name: | Update tracker |                    |              |
| Created By:    | Magdeline Ng   | Last Updated By:   | Magdeline Ng |
| Date Created:  |                | Date Last Updated: | 17/4/22      |

|                    |  |
|--------------------|--|
| Actor:             | User   |
| Description:       | Change places from “unexplored” to “to explore” and “explored”   |
| Preconditions:     | <ol style="list-style-type: none"> <li>1. User’s device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol>   |
| Postconditions:    | <ol style="list-style-type: none"> <li>1. User’s selected place will be added to the list of ‘toExplore’ items and saved into the database.</li> <li>2. After the time of exploring that user has set lapsed, system will shift the place from ‘toExplore’ to ‘explored’</li> </ol>                    |
| Priority:          | High   |
| Frequency of Use:  | Moderate   |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. System adds a ‘toExplore’ item to the user’s ‘Tracker’ page, containing the place, date, people going and time details.</li> <li>2. After the time has lapsed, the system shift the place from the ‘toExplore’ section to the ‘explored’ section.</li> </ol> |
| Alternative Flows: | AF-S2: User explores the place before set date and time. <ol style="list-style-type: none"> <li>1. User can set the place from ‘toExplore’ to ‘Explored’.</li> <li>2. Place will be displayed under the ‘explored’ section.</li> </ol>   |





|                       |                             |
|-----------------------|-----------------------------|
| Exceptions:           | -                           |
| Includes:             | 1. Verify Login Credentials |
| Special Requirements: | -                           |
| Assumptions:          | -                           |
| Notes and Issues:     | -                           |

|                |                     |                    |              |
|----------------|---------------------|--------------------|--------------|
| Use Case ID:   | 14                  |                    |              |
| Use Case Name: | Respond to invites. |                    |              |
| Created By:    | Deionna Chee        | Last Updated By:   | Deionna Chee |
| Date Created:  |                     | Date Last Updated: | 17/4/22      |

|                   |   |
|-------------------|---|
| Actor:            | User  |
| Description:      | To accept/reject invite from another user.  |
| Preconditions:    | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol>  |
| Postconditions:   | <ol style="list-style-type: none"> <li>1. 'toExplore' items are updated accordingly (add or not adding new profiles) and saved into the database.</li> <li>2. Invites are removed from inbox and changes are saved into the database.</li> </ol>  |
| Priority:         | High  |
| Frequency of Use: | Moderate  |
| Flow of Events:   | <ol style="list-style-type: none"> <li>1. User navigates to the 'Inbox' page.</li> <li>2. User clicks on 'accept' to accept the invitation.</li> <li>3. User's profile is added to the 'toExplore' item for all users who have also accepted the invite and saved to the database.</li> </ol> |



|                       |  |
|-----------------------|--|
|                       | 4. 'Invite' item is removed from the list of invites on display and the change is saved into the database. |
| Alternative Flows:    | AF-S2: User clicks on 'decline' to decline the invitation.<br>1. Return to step 4.                         |
| Exceptions:           | -  |
| Includes:             | 1. Verify Login Credentials  |
| Special Requirements: | -  |
| Assumptions:          | -  |
| Notes and Issues:     | -  |

|                |                            |                    |           |
|----------------|----------------------------|--------------------|-----------|
| Use Case ID:   | 15                         |                    |           |
| Use Case Name: | Retrieve place information |                    |           |
| Created By:    | Herman Foo                 | Last Updated By:   | Faith Lee |
| Date Created:  |                            | Date Last Updated: | 16/4/2022 |

|                   |  |
|-------------------|--|
| Actor:            | Google Places API  |
| Description:      | Retrieving place information from Places API and our own database (for exploreSG rating)   |
| Preconditions:    | <ol style="list-style-type: none"> <li>1. User's device must be connected to Wifi/Mobile data</li> <li>2. User must be logged in to their account</li> <li>3. exploreSG ratings/reviews are linked to a place via its place UID</li> </ol> |
| Postconditions:   | <ol style="list-style-type: none"> <li>1. User will be able to view the place information</li> </ol>   |
| Priority:         | High   |
| Frequency of Use: | Often  |



|                       |   |
|-----------------------|---|
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. System gets place details (name, picture, Google rating, address etc.) from Places API</li> <li>2. System gets exploreSG rating and total number of ratings from our place database</li> <li>3. System displays all place information on the particular place details page</li> </ol> |
| Alternative Flows:    | AF-S1: No image available for a particular place<br><ol style="list-style-type: none"> <li>1. Default question mark image is displayed instead</li> </ol>   |
| Exceptions:           | -   |
| Includes:             | -   |
| Special Requirements: | -   |
| Assumptions:          | A unique ID exists for every place from Places API  |
| Notes and Issues:     | -   |

|                |                     |                    |           |
|----------------|---------------------|--------------------|-----------|
| Use Case ID:   | 16                  |                    |           |
| Use Case Name: | Change display name |                    |           |
| Created By:    | Ihsan               | Last Updated By:   | Ihsan     |
| Date Created:  |                     | Date Last Updated: | 16/4/2022 |

|                 |  |
|-----------------|--|
| Actor:          | User   |
| Description:    | Users can change their display name  |
| Preconditions:  | <ol style="list-style-type: none"> <li>1. User's account must already exist</li> </ol>   |
| Postconditions: | <ol style="list-style-type: none"> <li>1. User's display name will be changed and user's collection in database is updated</li> <li>2. User's display name will be changed for all uses</li> </ol> |



|                       |  |
|-----------------------|--|
| Priority:             | Medium   |
| Frequency of Use:     | Rarely   |
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. User enters new display name in the two text fields (first name, last name)</li> <li>2. User taps the “change” button</li> <li>3. System updates user’s display name on database and saves changes</li> <li>4. System displays updated display name on profile page</li> </ol> |
| Alternative Flows:    | AF-S2: User left at least one field blank <ol style="list-style-type: none"> <li>1. System displays “Field is empty” under empty field</li> <li>2. Return to step 1</li> </ol>   |
| Exceptions:           |  |
| Includes:             | <ol style="list-style-type: none"> <li>1. Verify login credentials</li> </ol>  |
| Special Requirements: | -  |
| Assumptions:          |  |
| Notes and Issues:     | -  |

|                |                        |                    |           |
|----------------|------------------------|--------------------|-----------|
| Use Case ID:   | 17                     |                    |           |
| Use Case Name: | Change profile picture |                    |           |
| Created By:    | Ihsan                  | Last Updated By:   | Ihsan     |
| Date Created:  |                        | Date Last Updated: | 16/4/2022 |

|                |  |
|----------------|--|
| Actor:         | User   |
| Description:   | Users can change their profile picture   |
| Preconditions: | <ol style="list-style-type: none"> <li>1. User’s account must already exist</li> </ol> |



|                       |   |
|-----------------------|---|
| Postconditions:       | <ol style="list-style-type: none"> <li>1. User's profile picture will be changed and user's collection in database is updated</li> <li>2. User's profile picture will be changed for all uses</li> </ol>  |
| Priority:             | Medium  |
| Frequency of Use:     | Rarely  |
| Flow of Events:       | <ol style="list-style-type: none"> <li>1. User clicks on "change" profile picture button on profile page</li> <li>2. System displays filepicker, displaying png and jpg files</li> <li>3. User selects new profile picture from system files</li> <li>4. System uploads file to database and saves changes</li> <li>5. System updates users picture attribute on database with URL of file and saves changes</li> <li>6. System displays updated profile picture on profile page</li> </ol> |
| Alternative Flows:    | AF-S3: User does not select a file, presses back<br><ol style="list-style-type: none"> <li>1. System displays "No file selected"</li> </ol>   |
| Exceptions:           |   |
| Includes:             | <ol style="list-style-type: none"> <li>1. Verify login credentials</li> </ol>   |
| Special Requirements: | -   |
| Assumptions:          |   |
| Notes and Issues:     | -   |

|                |                 |                    |           |
|----------------|-----------------|--------------------|-----------|
| Use Case ID:   | 18              |                    |           |
| Use Case Name: | Change username |                    |           |
| Created By:    | Ihsan           | Last Updated By:   | Ihsan     |
| Date Created:  |                 | Date Last Updated: | 16/4/2022 |

|        |      |
|--------|------|
| Actor: | User |
|--------|------|



|                       |  |
|-----------------------|--|
| Description:          | Users can change their password  |
| Preconditions:        | 1. User's account must already exist   |
| Postconditions:       | 1. User's username will be changed and user's collection in database is updated<br>2. User can login with the updated username   |
| Priority:             | Medium   |
| Frequency of Use:     | Rarely   |
| Flow of Events:       | 1. User clicks on "change" username button<br>2. System displays dialog pop up box with text field<br>3. User enters new username<br>4. User clicks on submit button<br>5. System updates user's username on database and saves changes<br>6. System displays updated username on profile page |
| Alternative Flows:    | AF-S4: User click on "cancel" button<br>1. System returns to profile page<br>AF-S4: Text field is empty<br>1. System displays "Field is empty"<br>2. Return to step 3<br>AF-S4: Entered username is already taken<br>1. System displays "Username taken"<br>2. Return to step 3                |
| Exceptions:           |  |
| Includes:             | 1. Verify login credentials  |
| Special Requirements: | -  |
| Assumptions:          |  |
| Notes and Issues:     | -  |



|                |                  |                    |           |
|----------------|------------------|--------------------|-----------|
| Use Case ID:   | 19               |                    |           |
| Use Case Name: | Manage interests |                    |           |
| Created By:    | Faith Lee        | Last Updated By:   | Ihsan     |
| Date Created:  |                  | Date Last Updated: | 16/4/2022 |

|                    |   |
|--------------------|---|
| Actor:             | User  |
| Description:       | User can edit their account interests, which are used to recommend them places on the home page   |
| Preconditions:     | <ol style="list-style-type: none"> <li>1. User's device must be connected to WiFi/Mobile Data.</li> <li>2. User must be logged in to their account.</li> </ol>  |
| Postconditions:    | <ol style="list-style-type: none"> <li>1. User's interests will be updated on the database</li> <li>2. User's recommendations will be updated to follow the newly selected interests</li> </ol>   |
| Priority:          | Medium  |
| Frequency of Use:  | Rarely  |
| Flow of Events:    | <ol style="list-style-type: none"> <li>1. User clicks on manage interest button on profile page</li> <li>2. System displays manage interest page with user's previous interests</li> <li>3. User selects or deselects desired interest chips</li> <li>4. User clicks "save changes" button</li> <li>5. System updates user's interests on database and saves changes</li> <li>6. System displays profile page</li> </ol>              |
| Alternative Flows: | <p>AF-S1: User registers and signs in for the first time</p> <ol style="list-style-type: none"> <li>1. System displays manage interest page with no selected interests (New user)</li> <li>2. User selects desired interest chips</li> <li>3. Return to step 3</li> </ol> <p>AF-S6: User indicating interests for the first time after registering</p> <ol style="list-style-type: none"> <li>1. System displays home page</li> </ol> |
| Exceptions:        | -   |



|                       |                             |
|-----------------------|-----------------------------|
| Includes:             | 1. Verify login credentials |
| Special Requirements: | -                           |
| Assumptions:          |                             |
| Notes and Issues:     | -                           |

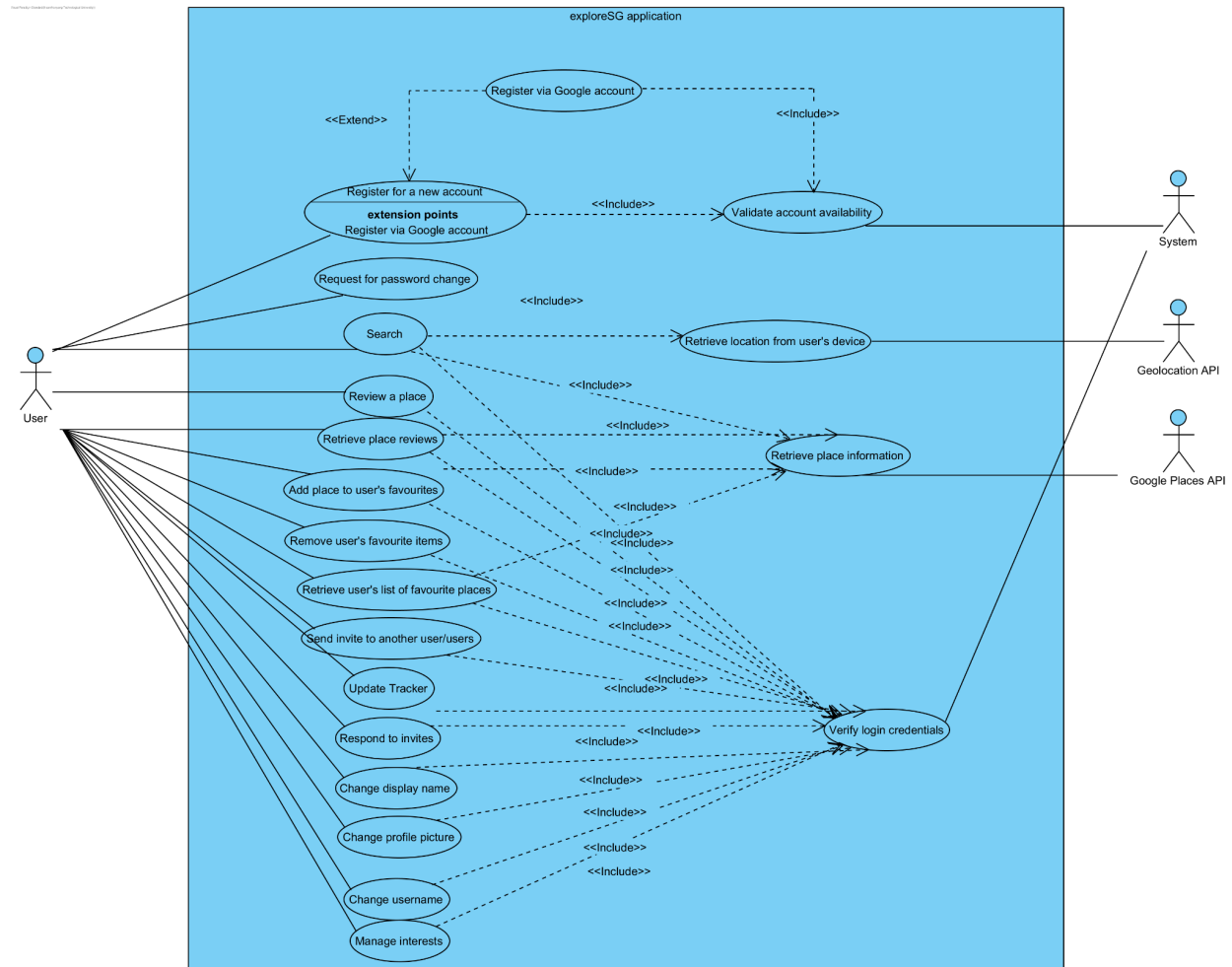
|                |                          |                    |           |
|----------------|--------------------------|--------------------|-----------|
| Use Case ID:   | 20                       |                    |           |
| Use Case Name: | Retrieve user's location |                    |           |
| Created By:    | Faith Lee                | Last Updated By:   | Faith Lee |
| Date Created:  |                          | Date Last Updated: | 16/4/2022 |

|                       |   |
|-----------------------|---|
| Actor:                | Geolocator  |
| Description:          | Retrieve current location coordinates (latitude and longitude) of user's device       |
| Preconditions:        | 1. User's device must be connected to WiFi/Mobile Data.                               |
| Postconditions:       | -   |
| Priority:             | High  |
| Frequency of Use:     | Often   |
| Flow of Events:       | 1. System requires locator<br>2. User's current location coordinates are returned     |
| Alternative Flows:    | AF-S1: Null location returned<br>1. Display error message "Location permission error" |
| Exceptions:           | -   |
| Includes:             | -   |
| Special Requirements: | -   |



|                   |   |
|-------------------|---|
| Assumptions:      | - |
| Notes and Issues: | - |

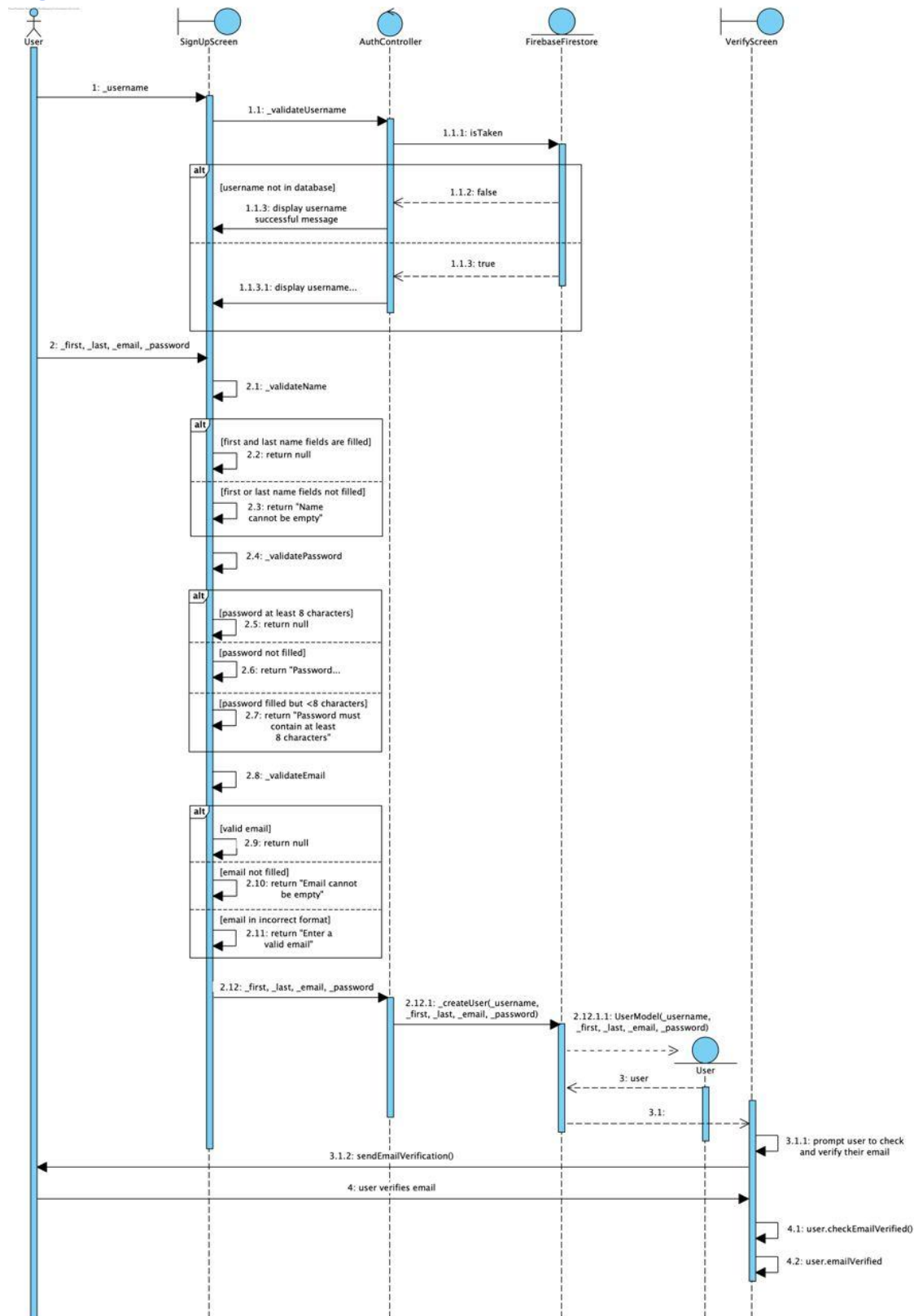
## 2.2. Use Case Diagram



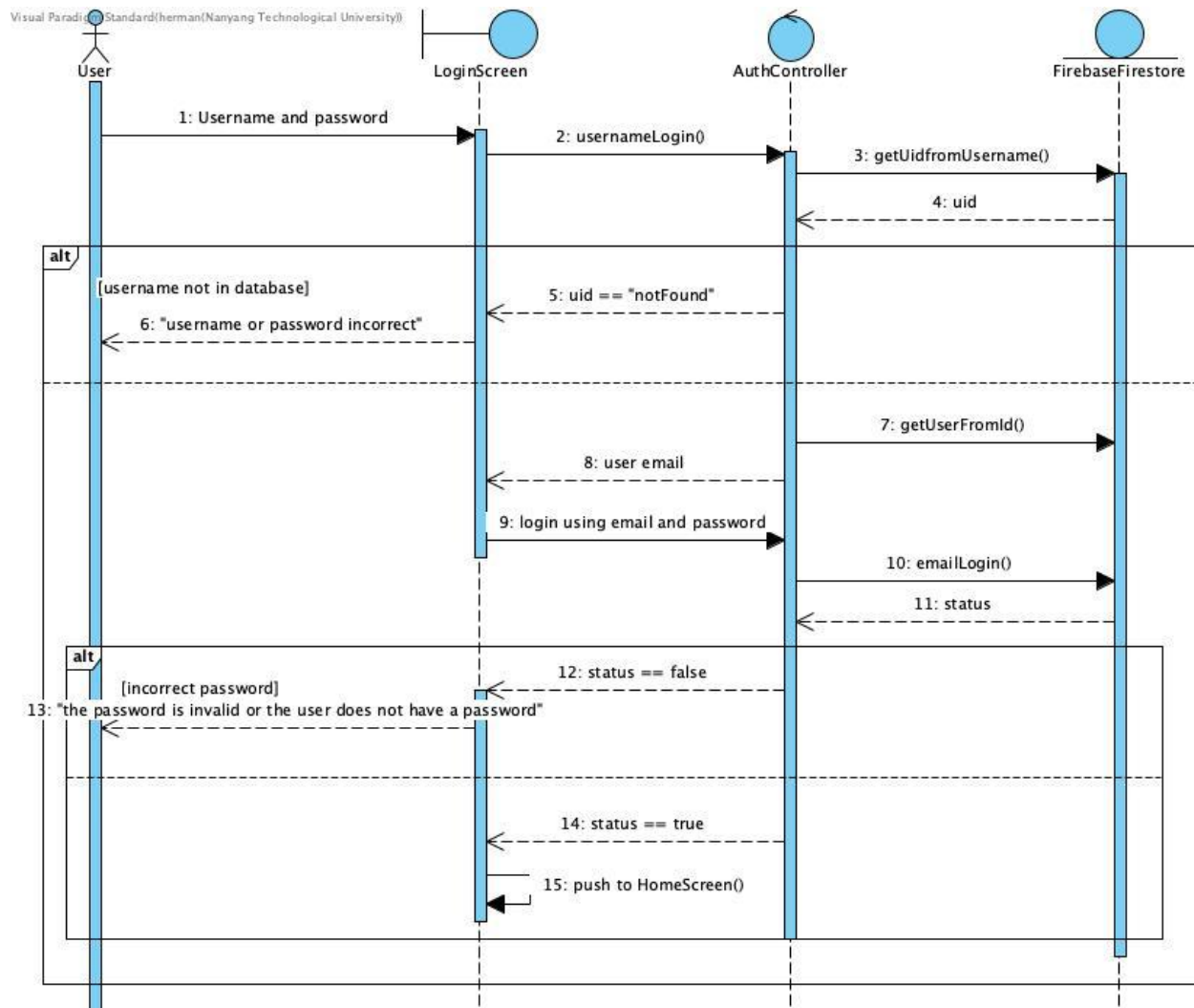


## 2.4. Sequence Diagrams

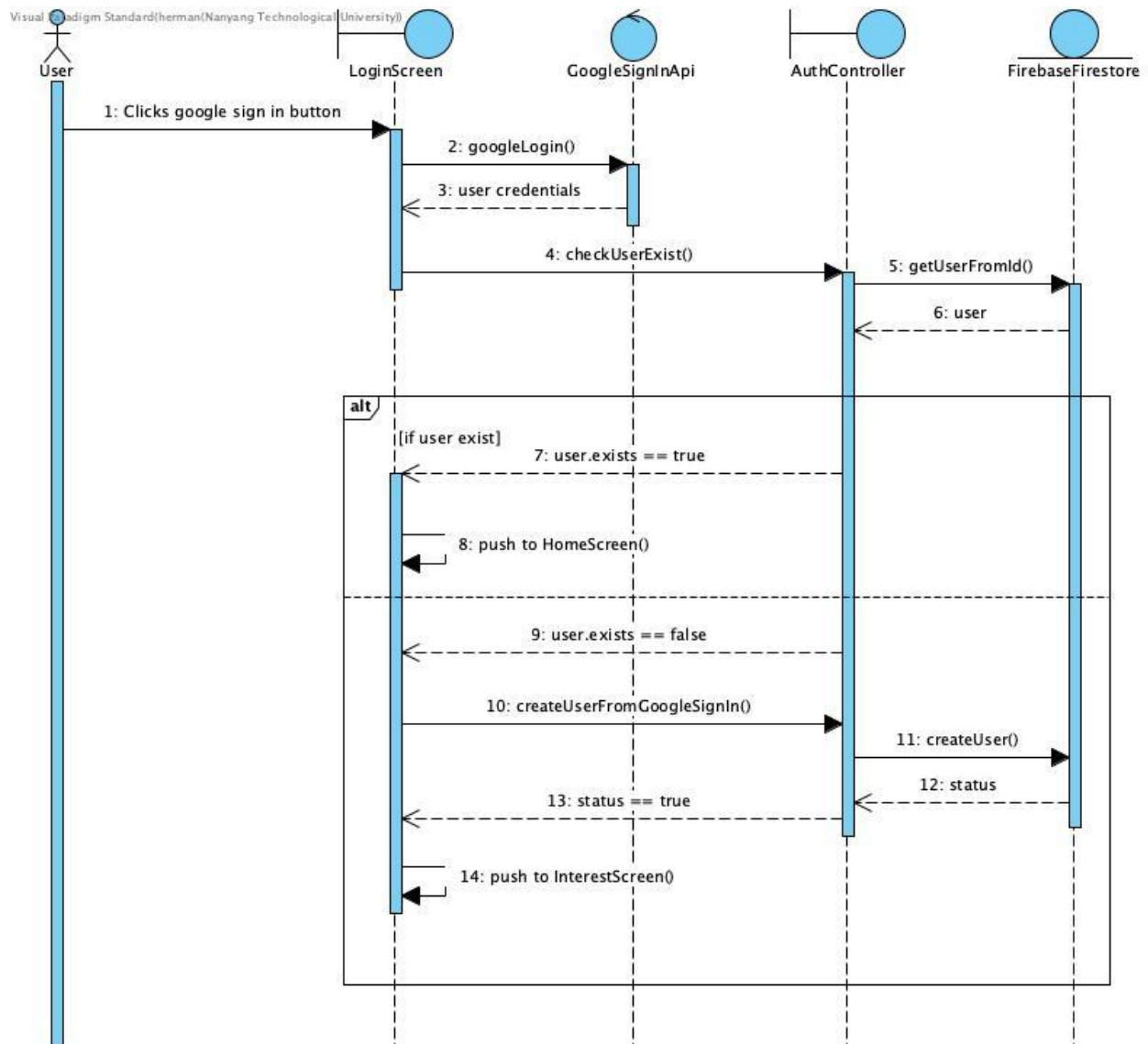
### Registration



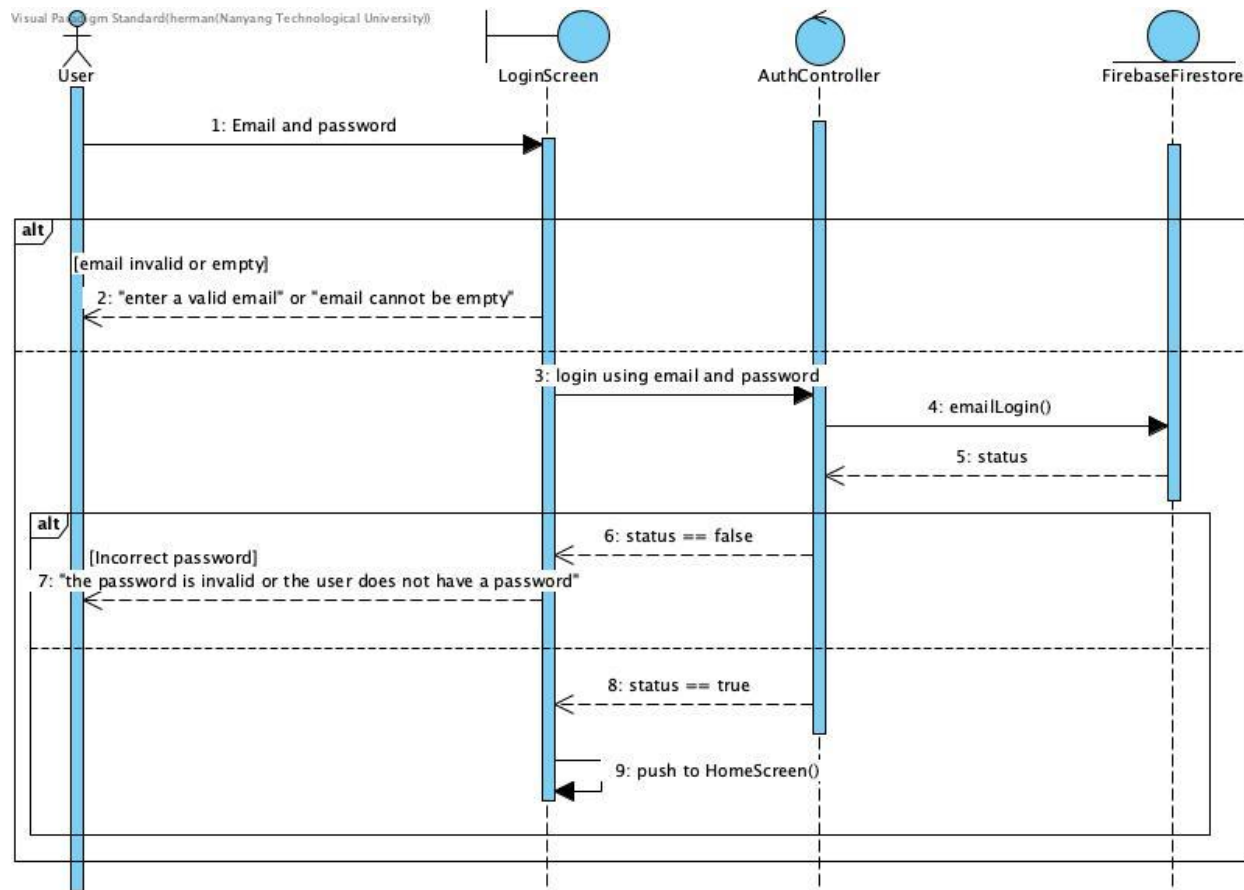
## Login (Username)



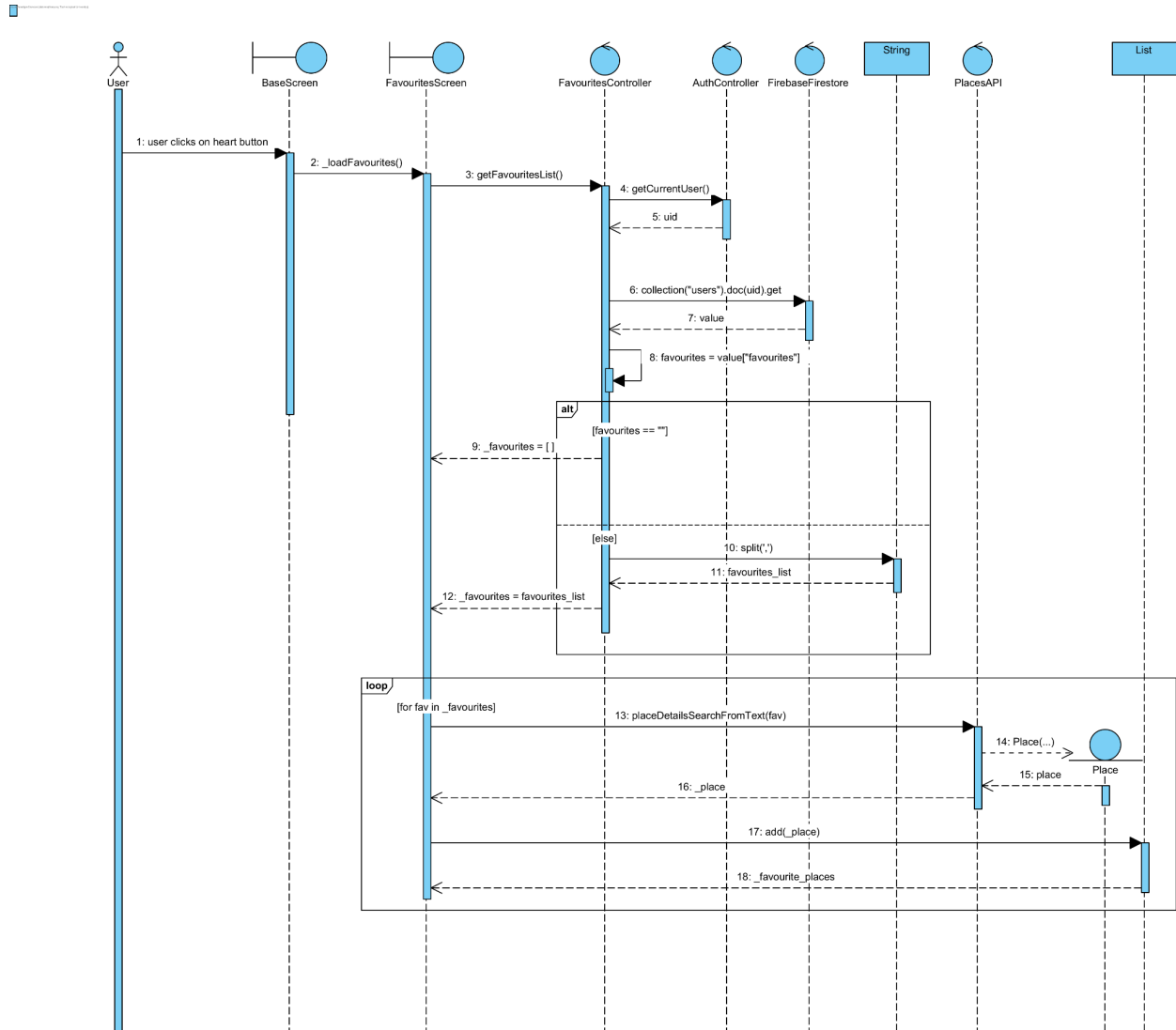
## Login (Google Sign In)



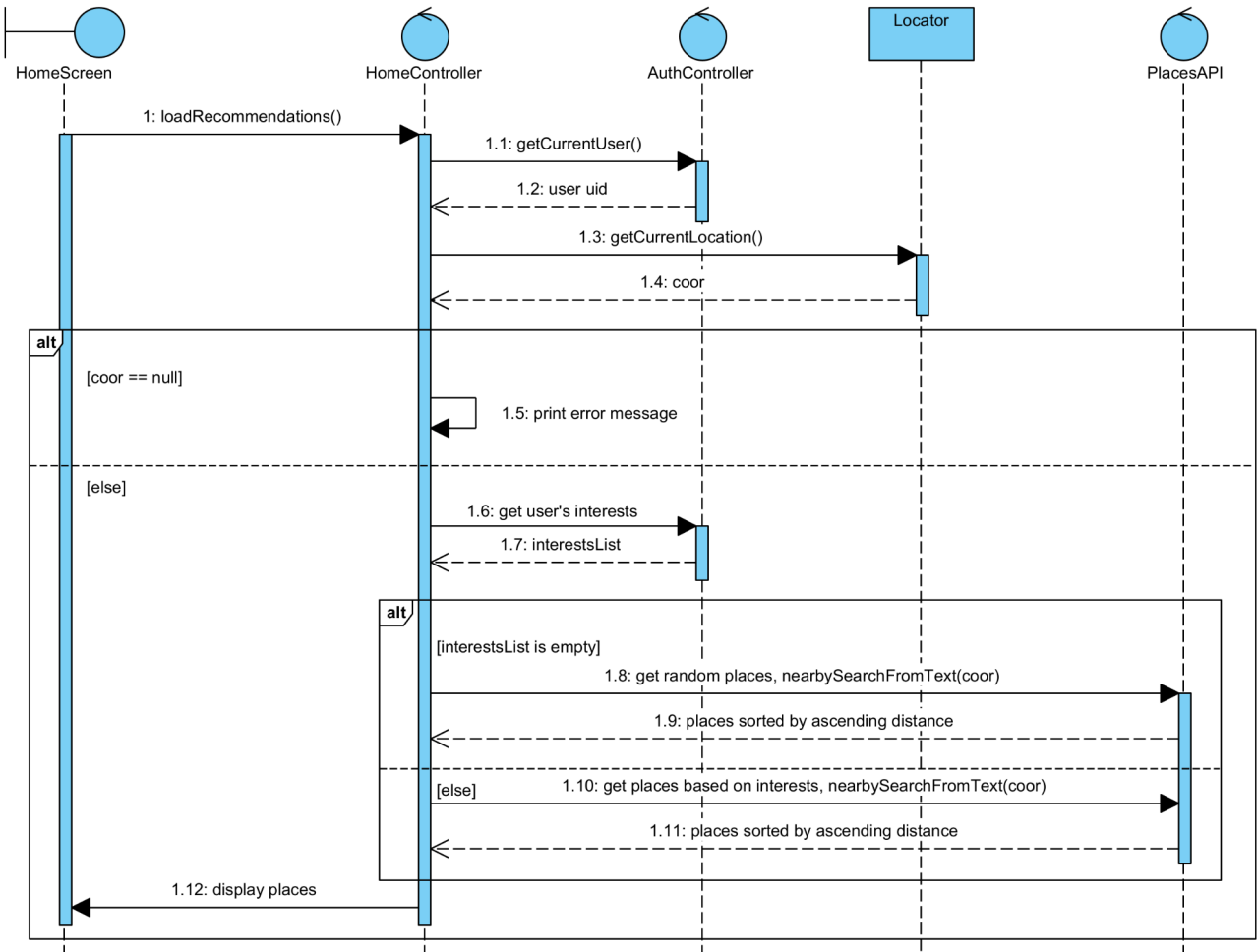
## Login (Email)



## Retrieve Favourites

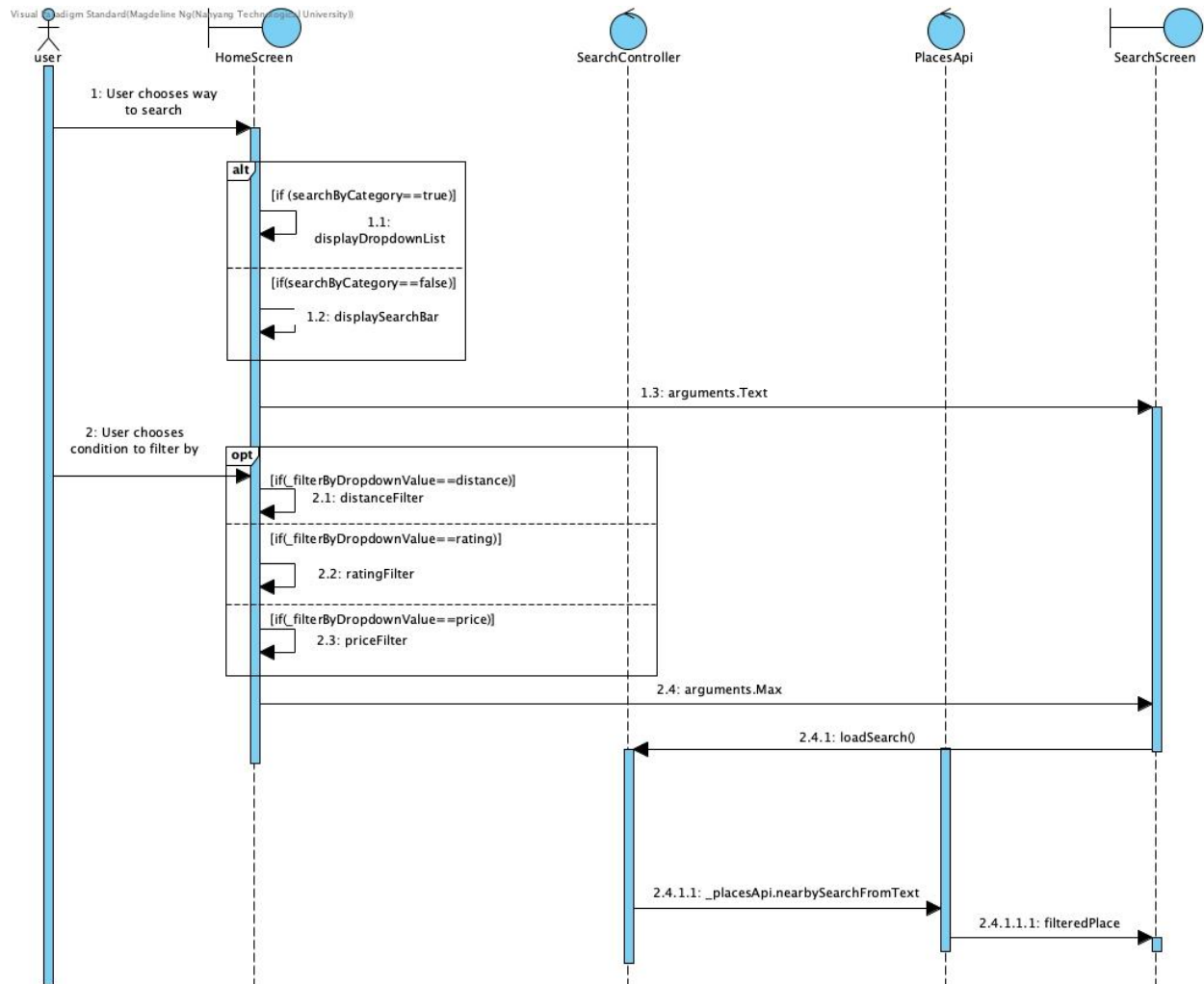


## Recommendations

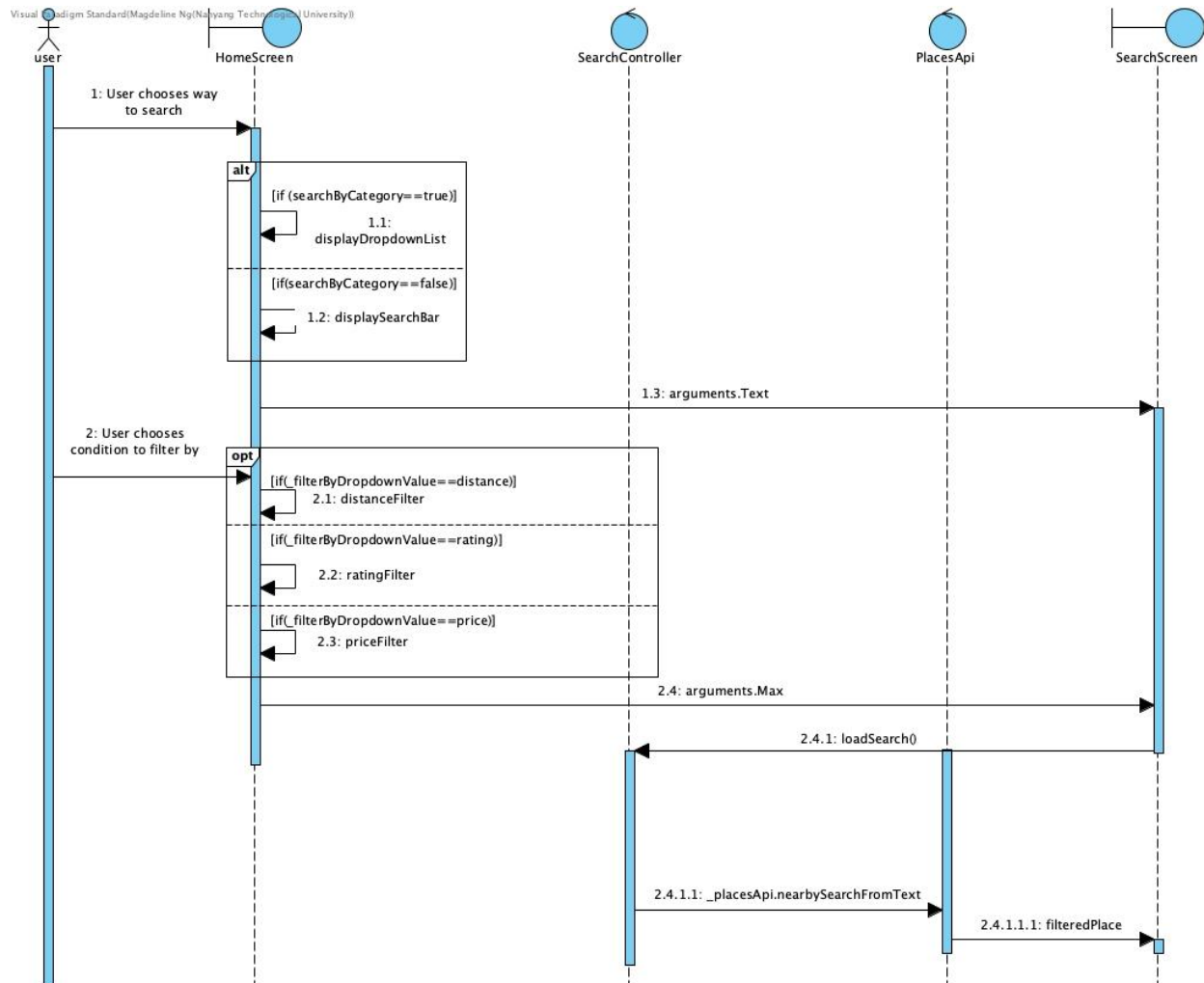




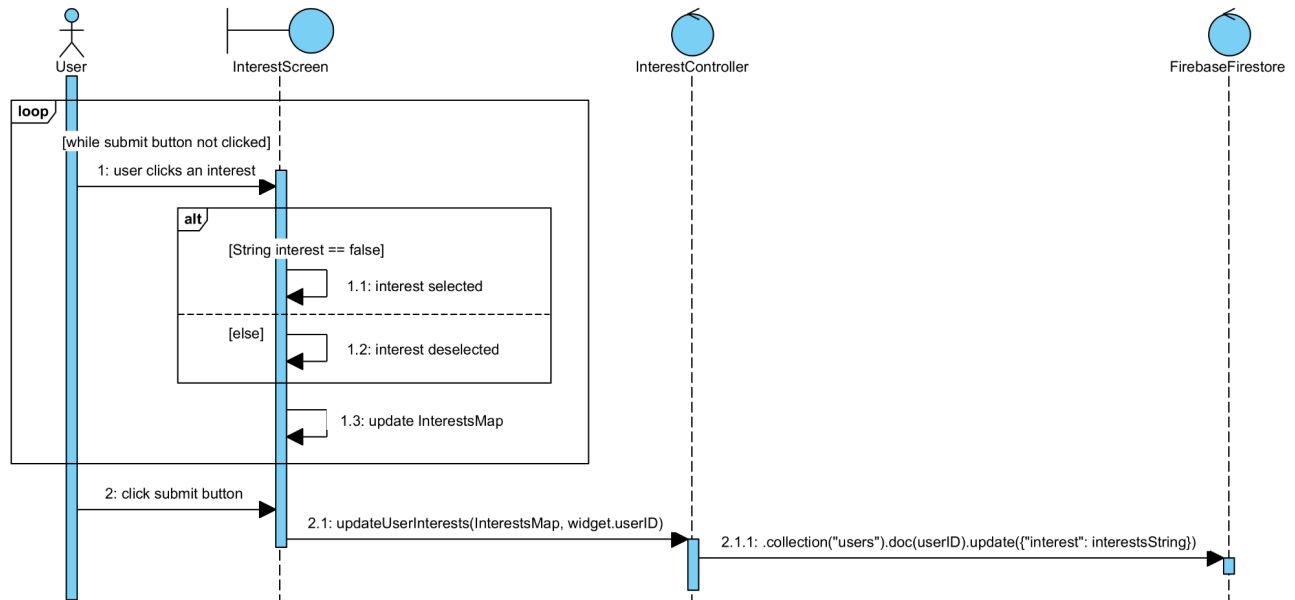
## Search for places



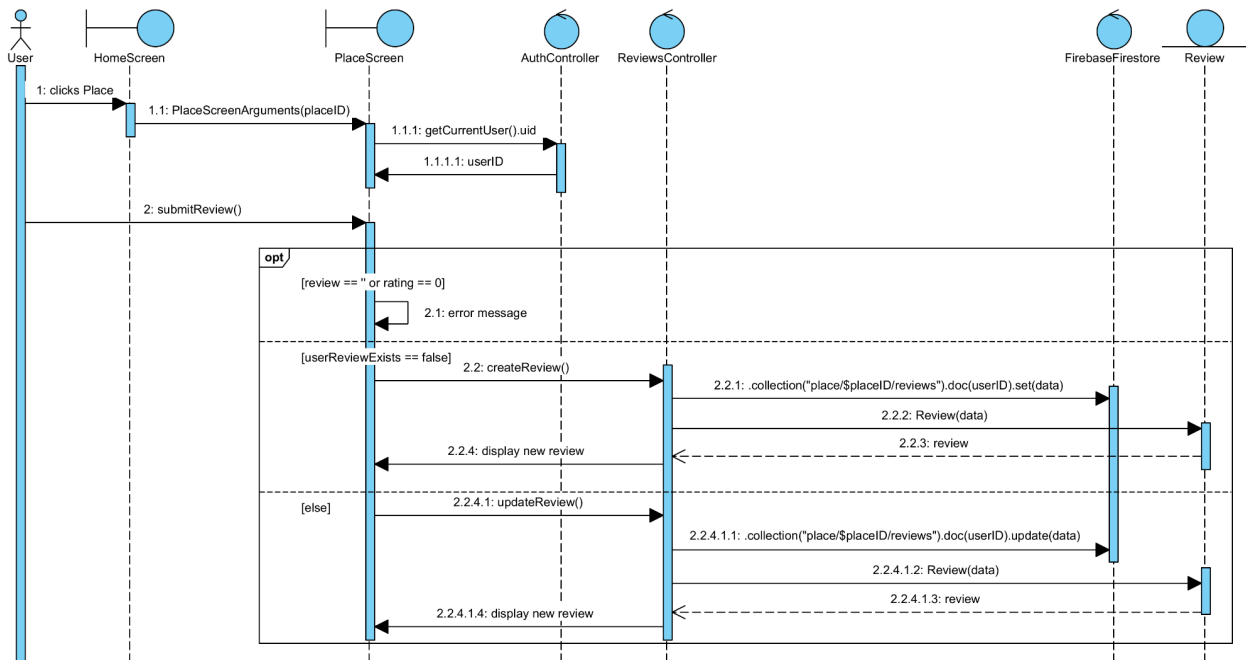
## Update Tracker



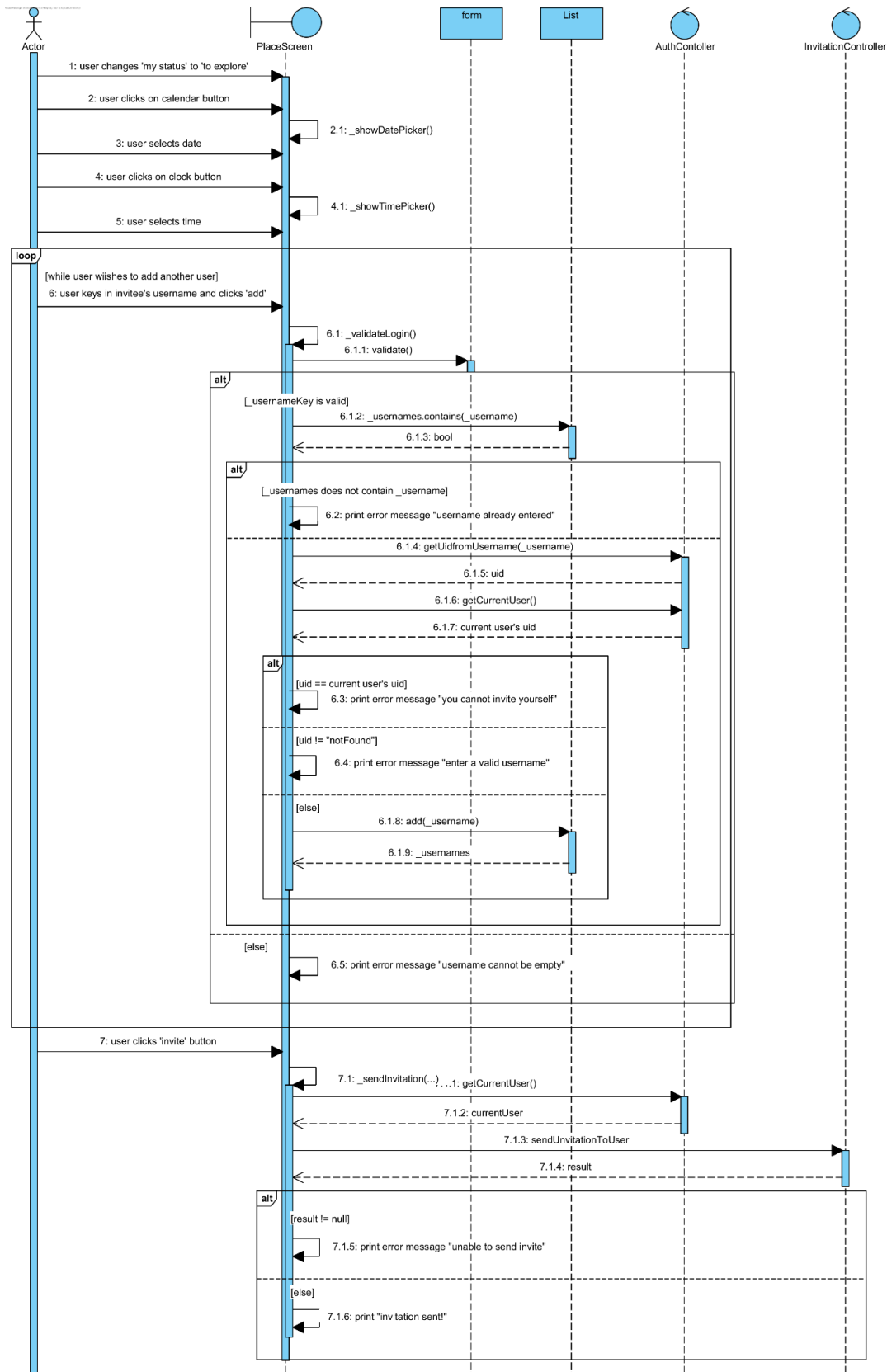
## Update Interests



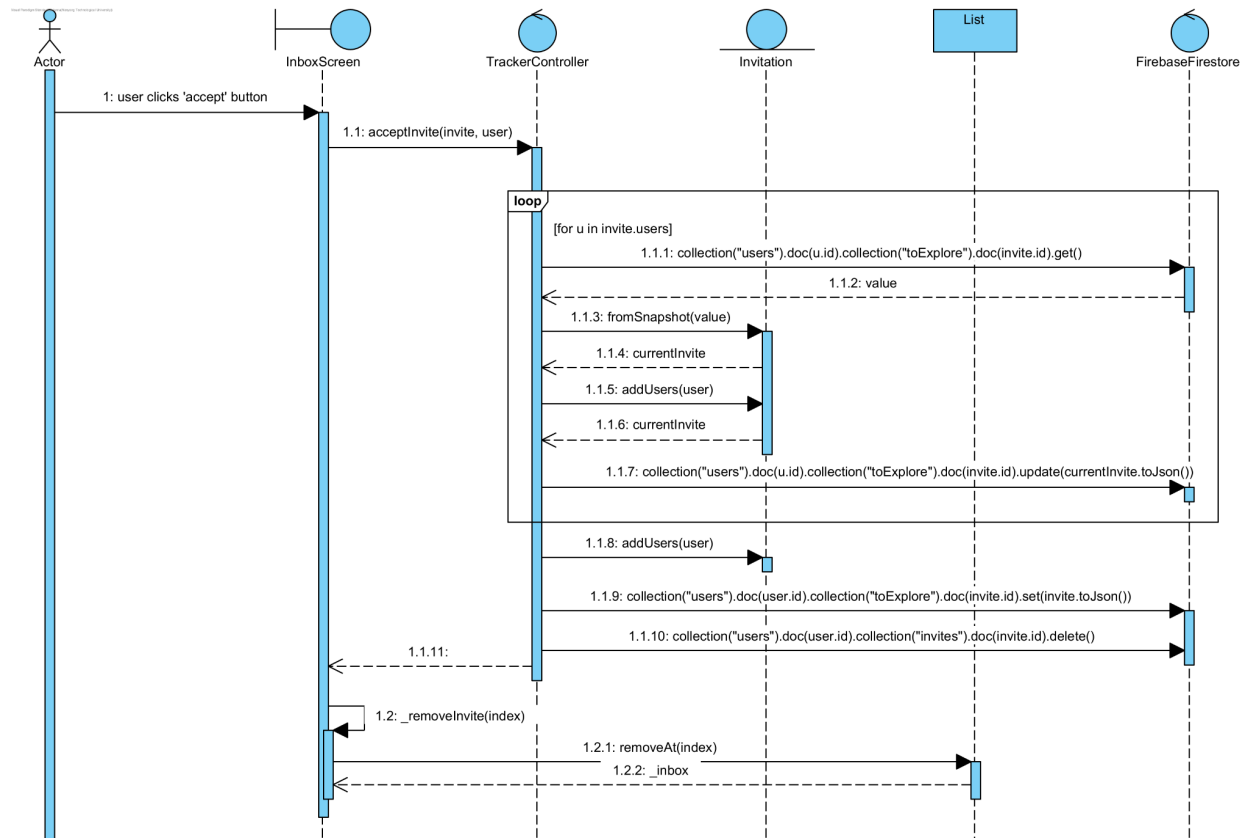
## Create/Update Review



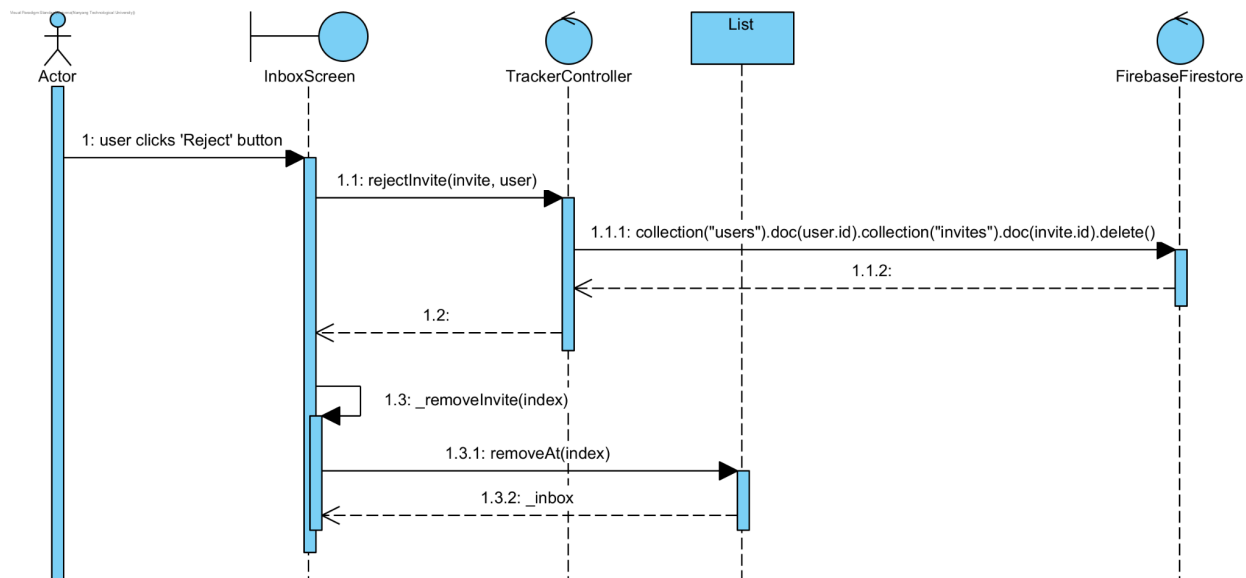
## Send Invite



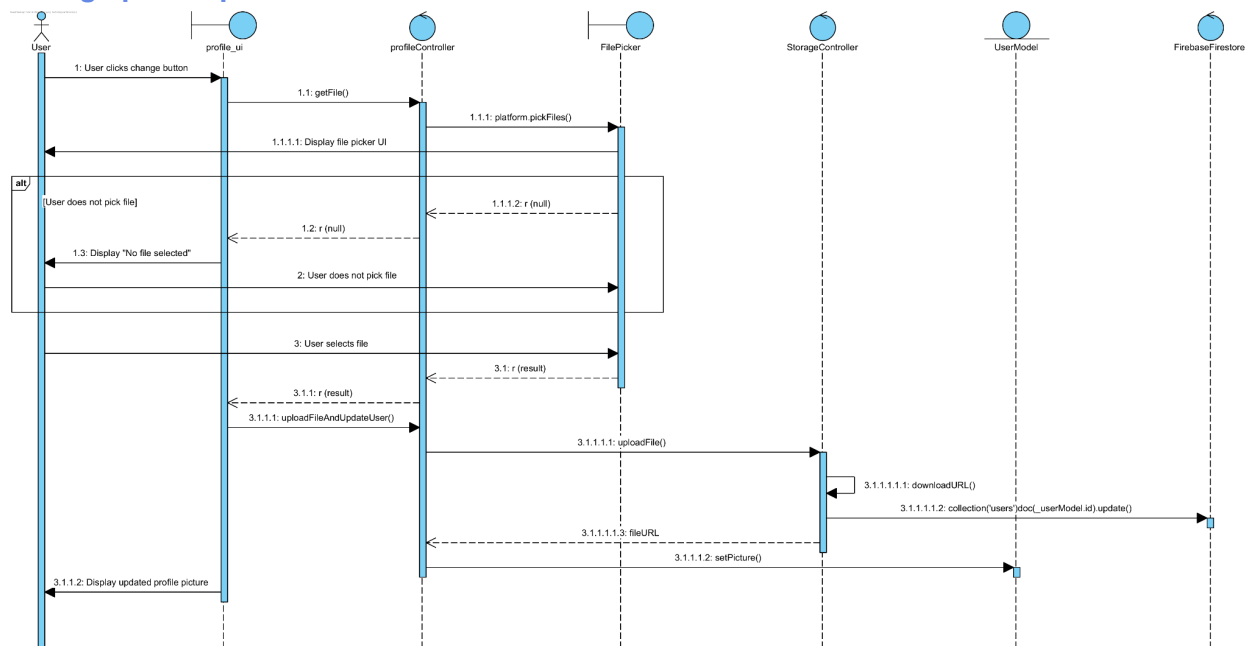
## Accept Invite



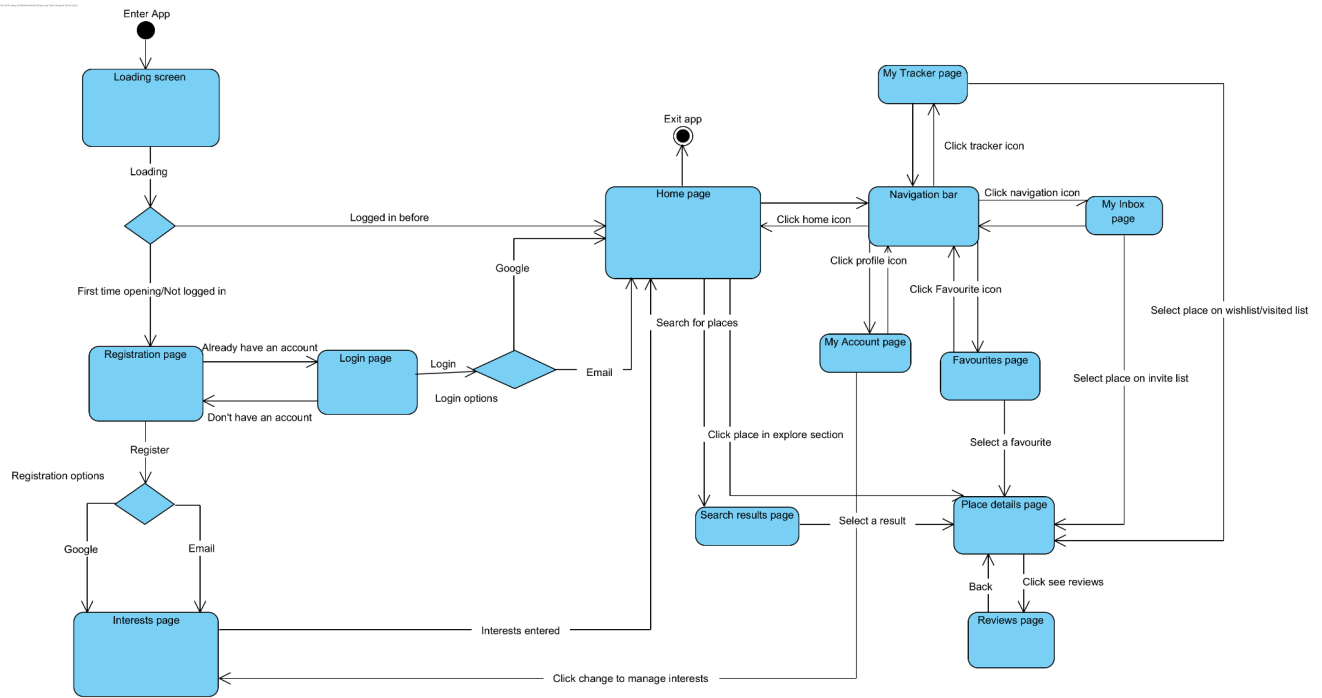
## Decline Invite



## Change profile picture



## 2.5. Dialog Map





### 3. Non-Functional Requirements

| Requirement         | Description   |
|---------------------|---|
| Reliability         | 1. System must not crash regardless of number of users at a time  |
| Recoverability      | 1. The system will return to being functional during a system restart   |
| Usability           | 1. Site interface must be intuitive, and must provide tips/pointers for first time users<br>2. Average rating of experience with the system should be at least %.<br>3. There must be no unhandled exceptions from user input |
| Security            | 1. Users must log in with their credentials and password to access the system.<br>2. For account recovery, user must click the link sent to their email to verify their identity  |
| Understandability   | 1. 90% of new users must be able to perform major use cases without additional help   |
| Speed/Response time | Results for searching with/without filters must take no more than 2 seconds   |

### 4. Interface Requirements

#### User

exploreSG can be used by a variety of users who wish to find new places to explore in Singapore.

#### Hardware

exploreSG requires users to enable location service while using the app to get the user's current location, for getting place distance from the user.

#### Software

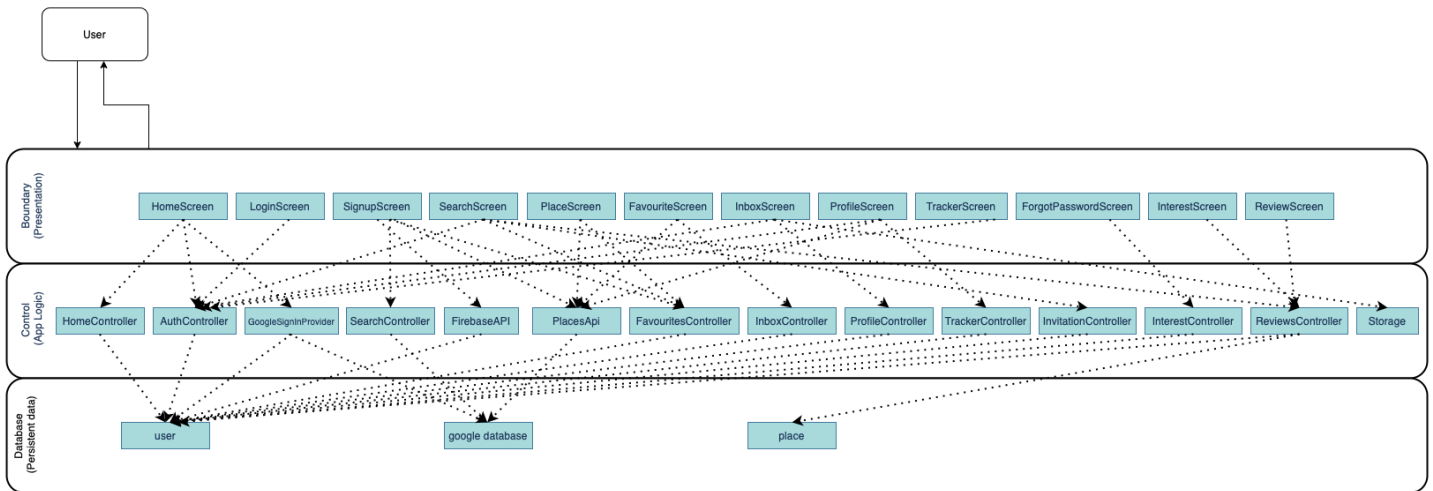
exploreSG is made with flutter, which can run on iOS and Android natively.

#### Communication

The app requires an internet connection to run. All features of the app will be accessible only with an internet connection.



## 5. Architecture Design



We used a three-layer system architecture design, in which we can see that dependencies flow downwards, from our UI classes, to their utilised controllers, and finally to our persistent databases.

We also ensured that each of our classes assumed only one responsibility, allowing for good separation and categorisation.

These are in accordance with the dependency inversion principle, as well as the single responsibility principle, allowing us to achieve high cohesion, and hence loose coupling in our code.

It is important to follow such principles during implementation, to aid with maintenance as well as adding functionality in the future. For example, in the future we may want to make it possible for users to add multiple places in a single invitation sent to friends. This can be implemented by simply adding the functionality on to the relevant control classes without having to modify existing entity classes.



## 6. Data Dictionary

| Term         | Definition  |
|--------------|---|
| User         | User refers to the person using the application to discover and find places of interest to go to based on their own interests   |
| System       | System refers to the mobile application   |
| Place        | Place refers to sites/attraction in Singapore where users can go for activities   |
| Favourite    | A user is able to favourite a place of interest to bookmark it for future references. All favoured places are compiled in a list that is accessible to the user once logged in.                         |
| Tracker      | A user is able to change the status of a place, then view 'to explore' and 'explored' places in their tracker. All tracked places are compiled in a list that is accessible to the user once logged in. |
| Inbox        | A user is able to view pending invites from friends in their inbox. They can choose to accept or reject invites through their inbox.  |
| Search       | Search is a feature that allows users to find places based on keyword or by type of place   |
| Filter       | Filter is a function that restricts the information presented by excluding results that do not fit the specified criteria   |
| 'unexplored' | Status of a place that has not been visited yet   |
| 'to explore' | Status of a place that is going to be visited; invitation has been created (either for solo exploration or exploration with others)   |
| 'explored'   | Status of a place that has been visited.  |

## 7. Testing

### 1. Black Box Testing

#### Registration Test Cases

| Test Input                 |            |           |                     |                       | Expected Output                            | Actual Output                              |
|----------------------------|------------|-----------|---------------------|-----------------------|--|--|
| Username                   | First Name | Last Name | Email Address       | Password              |  |  |
| Valid username             | Not empty  | Not empty | Valid email address | At least 8 characters | User created                               | User created                               |
| Empty                      | Not empty  | Not empty | Valid email address | At least 8 characters | Username cannot be empty                   | Username cannot be empty                   |
| Contain special characters | Not empty  | Not empty | Valid email address | At least 8 characters | Username cannot contain special characters | Username cannot contain special characters |
| <4 characters              | Not empty  | Not empty | Valid email address | At least 8 characters | Username cannot be less than 4 characters  | Username cannot be less than 4 characters  |
| Existing username          | Not empty  | Not empty | Valid email address | At least 8 characters | Change to another unique username          | Change to another unique username          |
| Valid username             | Empty      | Not empty | Valid email address | At least 8 characters | Name cannot be empty                       | Name cannot be empty                       |
| Valid username             | Not empty  | Empty     | Valid email address | At least 8 characters | Name cannot be empty                       | Name cannot be empty                       |
| Valid username             | Not empty  | Not empty | Empty               | At least 8 characters | Email cannot be empty                      | Email cannot be empty                      |
| Valid username             | Not empty  | Not empty | Invalid format      | At least 8 characters | Enter a valid email                        | Enter a valid email                        |

|                |           |           |                     |               |   |   |
|----------------|-----------|-----------|---------------------|---------------|---|---|
| Valid username | Not empty | Not empty | Valid email address | Empty         | Password cannot be empty                    | Password cannot be empty                    |
| Valid username | Not empty | Not empty | Valid email address | <8 characters | Password must contain at least 8 characters | Password must contain at least 8 characters |

### Username Login Test Cases

| Test Input         |   | Expected Output  | Actual Output   |
|--------------------|---|--|---|
| Username           | Password  |  |   |
| Existing username  | Correct password (at least 8 characters)                        | Successful login to app  | Successful login to app   |
| Existing username  | Incorrect password (at least 8 characters)                      | Failed login with error message  | Failed login with error message   |
| Existing username  | Password does not meet required length (less than 8 characters) | Failed login with error message prompting the user to key in at least 8 characters | Failed login with error message ("Password must contain at least 8 characters") |
| Incorrect username | Correct password (at least 8 characters)                        | Failed login with error message  | Failed login with error message ("Username or password incorrect")              |
| No username filled | Correct password (at least 8 characters)                        | Failed login with error message prompting the user to key in username              | Failed login with error message ("Username cannot be empty")                    |
| Existing username  | No password filled  | Failed login with error message prompting the user to key in password              | Failed login with error message ("Password cannot be empty")                    |



### Email Login Test Cases

| Test Input      |   | Expected Output  | Actual Output   |
|-----------------|---|--|---|
| Email           | Password  |  |   |
| Existing email  | Correct password (at least 8 characters)                        | Successful login to app  | Successful login to app   |
| Existing email  | Incorrect password (at least 8 characters)                      | Failed login with error message  | Failed login with error message   |
| Existing email  | Password does not meet required length (less than 8 characters) | Failed login with error message prompting the user to key in at least 8 characters | Failed login with error message ("Password must contain at least 8 characters") |
| Invalid email   | Correct password (at least 8 characters)                        | Failed login with error message  | Failed login with error message ("Enter a valid email")                         |
| No email filled | Correct password (at least 8 characters)                        | Failed login with error message prompting the user to key in email                 | Failed login with error message ("Email cannot be empty")                       |
| Existing email  | No password filled  | Failed login with error message prompting the user to key in password              | Failed login with error message ("Password cannot be empty")                    |

### Google Login Test Cases

| Test Input  | Expected Output   | Actual Output   |
|---|---|---|
| Existing google account that is linked to an existing exploresg account     | Successful login to app   | Successful login to app   |
| Existing google account that is not linked to an existing exploresg account | Successful creation of account linked to google account and login to homepage | Successful creation of account linked to google account and login to homepage |



### Update Tracker Test Cases

| Test Input  | Expected Output  | Actual Output  |
|---|--|--|
| User changes place status from 'unexplored' to 'to-explore', chooses valid date and time, then presses 'add to-explore' | Place appears at user's tracker, under 'to-explore' with chosen date and time, with only the user under 'people'   | Place appears at user's tracker, under 'to-explore' with chosen date and time, with only the user under 'people'   |
| User changes place status from 'unexplored' to explored   | Place appears at user's tracker, under 'explored'  | Place appears at user's tracker, under 'explored'  |
| User tries to change status from 'explored' to 'unexplored'   | 'unexplored' option is disabled  | 'unexplored' option is disabled  |
| User accepts invite from friend for a place that the user has already 'explored'  | Place status changes to 'to-explore'. Place appears at user's tracker, under 'to-explore' with invitation details. Previous review (if any) is stored and retrievable when status is changed to 'explored' | Place status changes to 'to-explore'. Place appears at user's tracker, under 'to-explore' with invitation details. Previous review (if any) is stored and retrievable when status is changed to 'explored' |
| User creates invite for a place that the user has already 'explored'  | Place status changes to 'to-explore'. Place appears at user's tracker, under 'to-explore' with invitation details. Previous review (if any) is stored and retrievable when status is changed to 'explored' | Place status changes to 'to-explore'. Place appears at user's tracker, under 'to-explore' with invitation details. Previous review (if any) is stored and retrievable when status is changed to 'explored' |

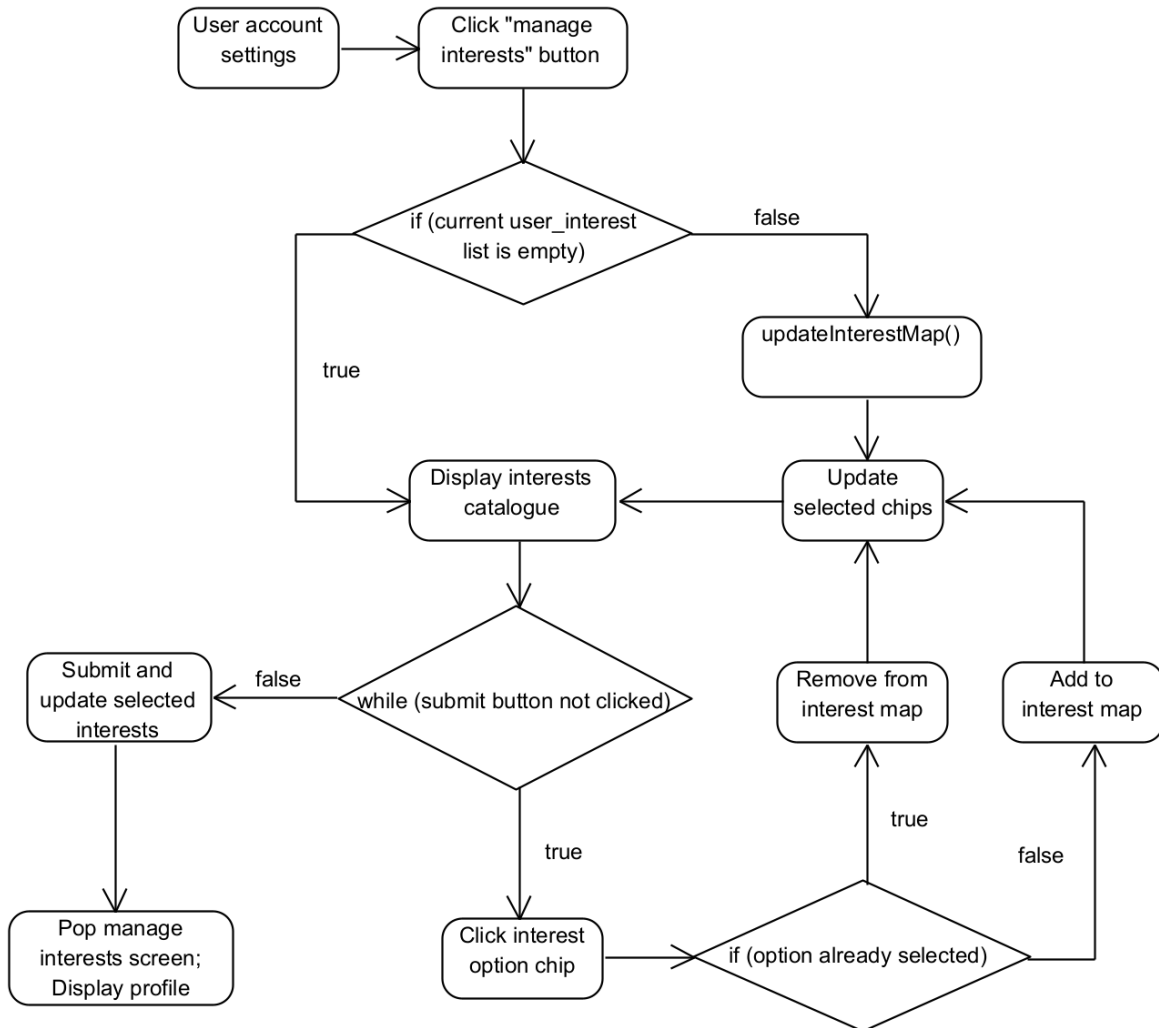
### Send invitation Test Cases

| Test Input |                         |              | Expected Output                      | Actual Output                        |
|------------|-------------------------|--------------|--------------------------------------|--------------------------------------|
| Invitee(s) | Date                    | Time         |                                      |                                      |
| >1         | Selected (After today)  | Not Selected | Prompt to select a valid time        | Prompt to select a valid time        |
| >1         | Selected (Before today) | Selected     | Prompt to select a valid date        | Prompt to select a valid date        |
| 0          | Selected (After today)  | Selected     | Prompt to include at least 1 invitee | Prompt to include at least 1 invitee |

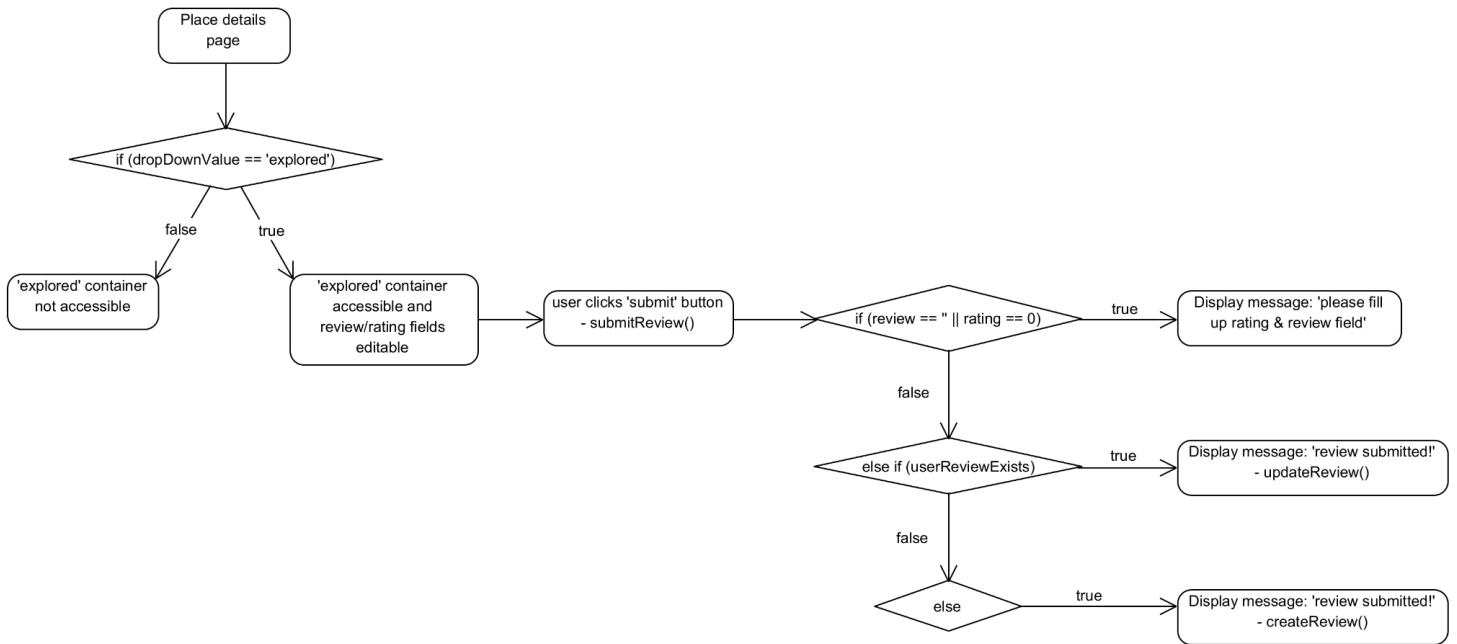
|    |                        |          |                                   |                                   |
|----|------------------------|----------|-----------------------------------|-----------------------------------|
| >1 | Selected (After today) | Selected | Invitation sent out to invitee(s) | Invitation sent out to invitee(s) |
|----|------------------------|----------|-----------------------------------|-----------------------------------|

## 2. White Box Testing – Control Flow Graphs

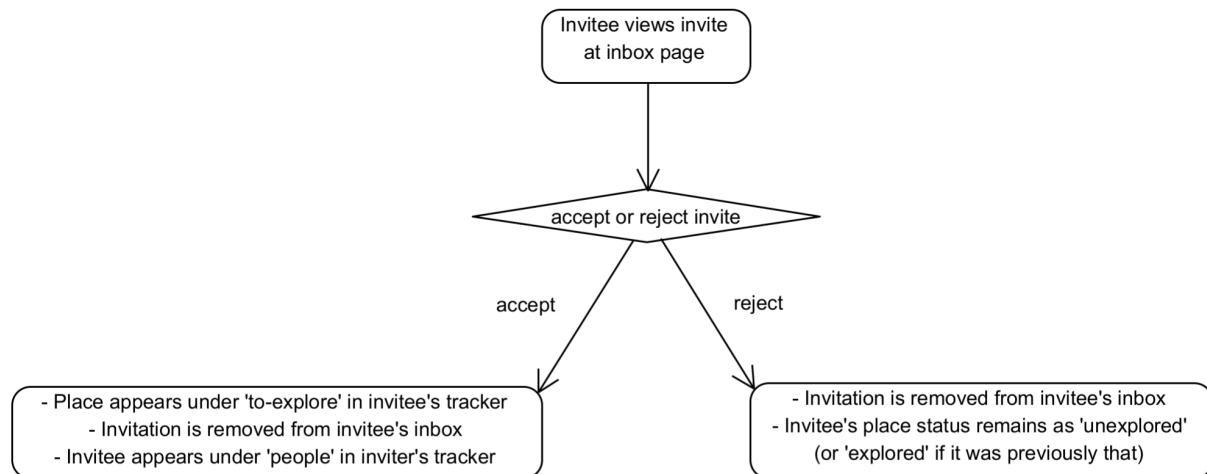
### Managing User's Interests



## User's Review



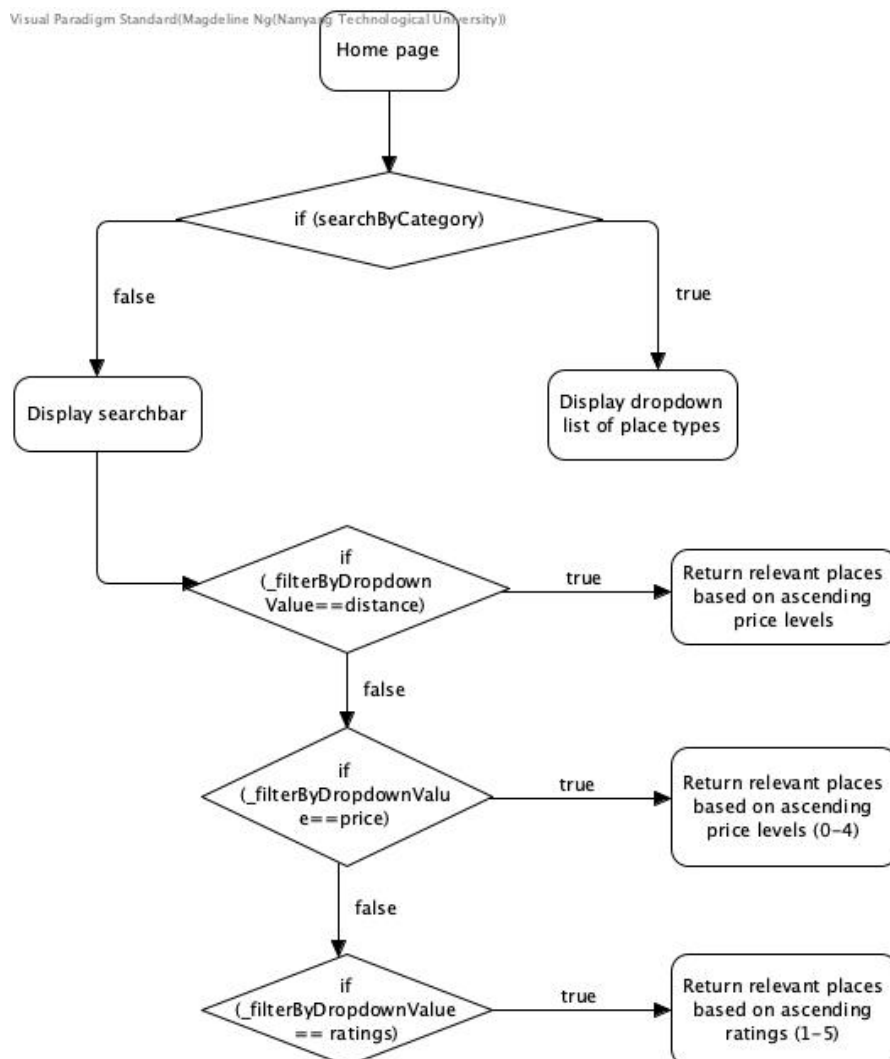
## Invitee Accepts/Rejects Invite



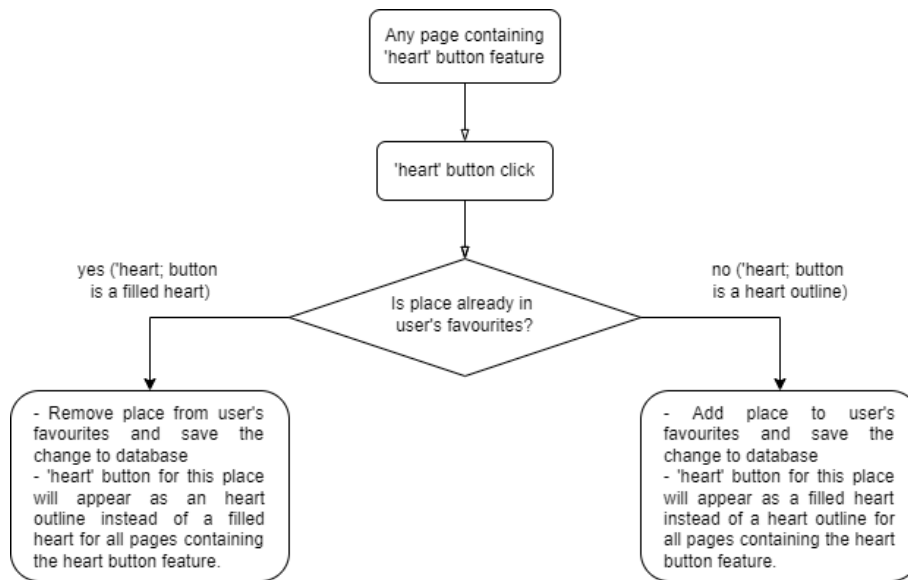


## Search for places

Visual Paradigm Standard(Magdelaine Ng(Nanyang Technological University))



## Favourite





## 8. Traceability

Throughout the project, we practised collective code ownership, in which every team member is allowed and encouraged to make changes to the code either complete a task, repair a defect, or to improve the code's overall structure. In the meanwhile, we still clearly delegated which member works on which part to ensure that we all are working on separate parts of the code, as well as used GitHub branches, to ensure less merge conflicts when merging our own work to the main branch.

Our group also practised continuous integration, where we routinely integrated code changes into the main branch, and tested the changes as early and often as possible to manage integration errors.

Finally, we often ran unit tests to ensure that our code compiled, and our core components work as designed.

As for the software engineering approach we used, we adopted a mix of agile and plan-driven approaches in our project.

Plan-driven development lays out processes in advance to measure progress to keep developers on track.

Agile development embraces change; system requirements may change and our system is designed to accommodate these changes. Program specification, design, and implementation are interleaved in this approach, causing the system to be developed as a series of versions or increments.

In the beginning, we adopted a more plan driven approach to more thoroughly flesh out our ideas before diving into the code. We slowly transitioned into a more agile approach deeper into our app's development to better adapt to and accommodate the changes in our system requirements.

From the very foundations, we elicited our requirements, following software requirement specification (SRS) practices. Then we moved on to create our use case diagram to further specify the functionalities we wanted to implement in our app. From there, we created our class diagram to aid us in understanding the dependencies between our different classes to help us with the code implementation. Then we created sequence diagrams of specific functions, that help to map out the scenario and sequence of our functions, to ensure our logic is sound before implementation, guiding us in writing our code. Finally, we did black/white box testings to uncover exceptions in our program and fix them in our code, to ensure smooth program flow.



## 9. Appendix

1. [Demo Video on Youtube](#)
2. [GitHub repository](#)
  - Master Branch
    - Source code
  - Main Branch
    - Diagrams PNGs
    - Slides
    - README
3. [Figma UI Prototype](#)