SUTD 2021 50.003 ESC Project Meeting 2 Report C3G7

Team Members:

Darren Loh Zhao Ying

Harshit Garg

Seah Zen Yi

Dong Jiajie

James Raphael Tiovalen

Table of Contents

Changes in Requirement

There have been some refinements in the requirements from Project Meeting 1. All the requirements discussed with the client listed in Project Meeting 1, under the 'Clarification of Requirements' section, still hold and are valid.

However, through our shared Telegram group, our client Mr Calvin Foo has noted to us to **include APs that are hidden**, even though they are fully functional. This can be done by finding their BSSIDs and levels in our scan results.

In addition, Mr Calvin Foo has shared that the best solutions in the market can achieve 2 to 5 meters in tracking accuracy. This means a larger emphasis will be placed on refining and bettering our algorithm to **hit the required accuracy range**.

Currently, the specified requirements entail the implementation of a single-level static/fixed floor plan with a relatively accurate algorithm. The requirements could potentially be changed in the future as the project progresses over time and as we coordinate together with the client.

Formal Documentation of Use Case

ID	**FindMyTag_UC_01**
Name	Area Mapping
Objectives	Accurately mapping the location on the phone and sending it to database to triangulate the location
Pre-conditions	1. Phone must accurately map the location. 2. Phone must be able to send data to the database. 3. The various Wi-Fi APs must be up and available to connect to.
Post-conditions	Success: 1. Area mapping is successful, and the correct data is sent to the database. Failure: 1. Area mapping has failed to identify the locations. 2. Area mapping has succeeded but failed to send the data to the database.

Actors	Primary: 1. Wi-Fi APs 2. Mobile Application Secondary: 1. Database 2. User
Trigger	User presses the ‘Map’ button.
Normal Flow	1. User uploads the floor plan onto the database. 2. Once uploaded, the user may proceed to the starting location. 3. When ready, press the ‘Map’ button and move in a line through the doors. 4. The devices then sense the Wi-Fi signal strength from the different APs connected and conclude the location of the user using triangulation method. 5. The user can pause, choose to re-map, or proceed to map it onto the floor plan. 6. A success message is shown to the user, prompting the user to continue. 7. The user may proceed to another area of the floor to continue mapping.
Alternative Flow	1. Mapping has failed and an error message is shown to the user on why it failed. a. Wi-Fi has failed to detect the location; use case concludes with error message and prompt to re-map. b. Uploading to database has failed; use case concludes with error message for user to temporarily keep data locally and fix the database.
Interacts With	Notify User, Data Collection, Wi-Fi Information Collection use cases.
Open Issues	1. How detailed should the error message be? a. Inform user ways to debug. b. Simple troubleshooting guide. 2. Will there be an admin and common user separation? a. Required if other users are expected to access the mapped data but cannot upload the data.

ID	FindMyTag_UC_02
Name	Data Collection for Database
Objectives	To collect and record the location data in a database.
Pre-conditions	1. Internet connection must be sufficiently strong. 2. The database must be connected to the internet and ready to receive the data. 3. Data must have been collected through the application.
Post-conditions	● Success 1. Places visited are recorded and can be checked any time. ● Failure 1. Location data collection is a failure. 2. Location data collection succeeded but information is not stored in the database.
Actors	Primary: 1. Database 2. Mobile Application Secondary: 1. User 2. Wi-Fi APs
Trigger	User presses ‘Upload’ to send the data to the database.
Normal Flow	1. After mapping has been completed in the app, User presses the ‘Upload’ button. 2. The app does a primary check on if there is an existing local data before requesting a connection with the database. 3. The database authenticates the app before receiving the data from the app. 4. Data received is then prepared and put into the cloud function to determine the user location using the algorithm.
Alternative Flow	1. User is unable to connect to the internet; use case concludes with error notification to the user. 2. App is unable to contact the database; use case concludes with error notification to the

	user. 3. Database fails to authenticate the app; use case concludes with error notification to the user. 4. Database fails to receive the data from the app; use case concludes with error notification to the user.
Interacts With	Notify User, Area Mapping use cases.
Open Issues	1. How will the data be prepared for the cloud function after getting taken from the app? 2. How will we fetch the data stored locally after prior failed attempts?

ID	**FindMyTag_UC_03**
Name	Wi-Fi Information Collection
Objectives	Collecting location information using the Wi-Fi
Pre-conditions	1. Wi-Fi must be enabled. 2. Signal strength must be sufficiently strong.
Post-conditions	Wi-Fi information collected upon request.
Actors	Primary: 1. Wi-Fi APs Secondary: 1. Mobile Application
Trigger	User presses the 'Map’ function on the application.
Normal Flow	1. The Wi-Fi emitter sends the signal to the device consistently. 2. The signal strength is then used by the device for mapping purposes.
Alternative Flow	1. One of the Wi-Fi emitters cannot be detected; use case concludes with slightly less accurate data collected.
Interacts With	Area Mapping use case.
Open Issues	1. (Alternative Flow 1) Might need an initial set-up to detect all possible Wi-Fi APs or beacons. 2. How will we troubleshoot the aforementioned issue?

ID	FindMyTag_UC_04
Name	Notify User
Objectives	From the collection of errors, pick out ones that suits the error that the user is facing
Pre-conditions	An error has occurred in one of the use cases.
Post-conditions	Error has been rectified or the user can troubleshoot the issues.
Actors	Primary: 1. Mobile Application Secondary: 1. User 2. Database
Trigger	A connection error or data collection error in other activities has occurred.
Normal Flow	1. The activity that receives an error sends the error code to the overseer. 2. From the collection of error codes, a most suitable one is sent back to be displayed.
Alternative Flow	NIL
Interacts With	Area Mapping, Data Collection and Wi-Fi Information Collection use cases.
Open Issues	1. How will we further develop this use case?

ID	FindMyTag_UC_05
Name	Data Testing
Objectives	From the data collected in the prior use cases, test the accuracy of the data.
Pre-conditions	1. The collected data exists. 2. The data has been mapped.
Post-conditions	1. The data collected during mapping is accurate and usable. 2. Data accuracy has an error allowance of approximately 2~5m.
Actors	Primary:

	1. User 2. Mobile Application 3. Database Secondary: 1. Wi-Fi APs
Trigger	The user presses the 'Test' button on the mobile application.
Normal Flow	1. The user presses the 'Test' button on the app. 2. Based on the Wi-Fi APs, the application fetches the floor plan from the database based on the initial estimated location. 3. The floor plan is then displayed on the mobile app, with a dot to indicate the user location. 4. The user can then choose to continue testing by pressing a 'Next' button to check his location at another spot. 5. The data collected is accurate and the testing concludes.
Alternative Flow	1. There is no initial data collected from Area Mapping use case; use case concludes with an error notification to the user. 2. The data collected is inaccurate; use case concludes, user can attempt to re-map the floor. 3. There is trouble fetching data from the database; use case concludes with an error notification to the user.
Interacts With	Notify User, Data Fetching, Wi-Fi Information Collection use cases.
Open Issues	1. Should we make it more user friendly by implementing extra buttons such as 're-map' during the testing phase?

ID	**FindMyTag_UC_06**
Name	Data Fetching
Objectives	Fetch the data from the database and display onto the application.
Pre-conditions	1. There must be existing data in the database. 2. The application must be connected to the

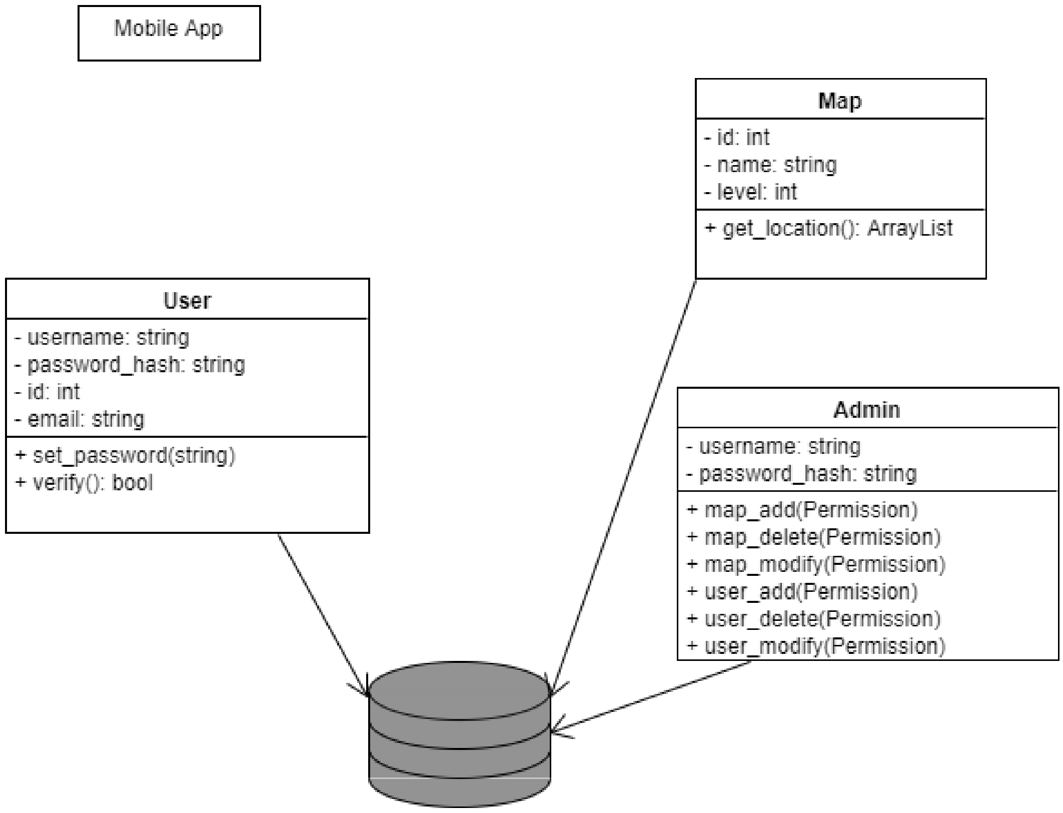
	database.
Post-conditions	The data fetched from the database is accurately displayed on the application for the user to view.
Actors	Primary: 1. Mobile Application 2. Database Secondary: 1. User
Trigger	1. ‘Test’ button is pressed. 2. ‘Pull’ button is pressed
Normal Flow	1. The user presses a button to try and pull the data from the database. 2. The application then collects the current location data and sends it to the database. 3. The database then proceeds to authenticate and send the data requested by the application. 4. Upon receiving the data, the application then displays the data onto the application for the user.
Alternative Flow	1. There is no initial data collected from Area Mapping use case; use case concludes with an error notification to the user. 2. There is trouble fetching data from the database; use case concludes with an error notification to the user.
Interacts With	Notify User, Data Testing use cases.
Open Issues	NIL

ID	**FindMyTag_UC_07 (NEW)**
Name	Hidden AP Search
Objectives	Search for hidden APs whose SSIDs cannot be searched by WifiManager scans.
Pre-conditions	1. Location permissions must be allowed. 2. Location must be turned on. 3. Presence of Wi-Fi APs.
Post-conditions	Hidden APs detected and also accounted for in our data collection and testing.

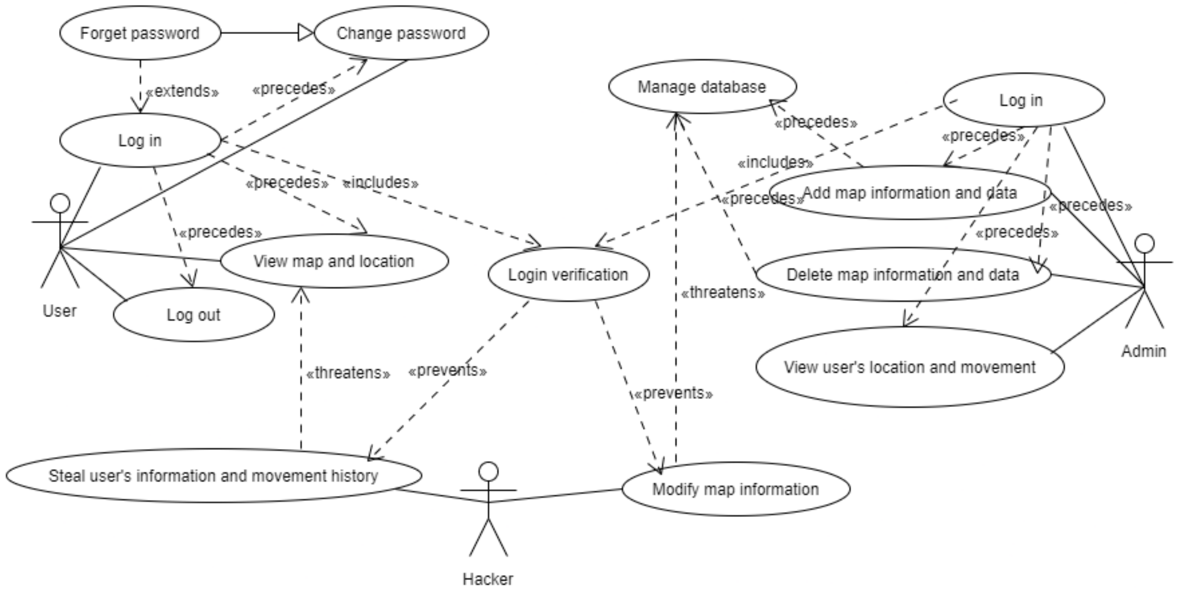
Actors	Primary: 1. Wi-Fi AP 2. User Secondary: 1. Database
Trigger	When the ‘Map’ button is pressed, WifiManager starts to scan for Wi-Fi APs and collect the BSSID + level.
Normal Flow	1. User presses the Map button. 2. WifiManager starts scanning for Wi-Fi APs. 3. Wi-Fi APs BSSID and level is collected and stored. 4. Changes in level are stored and sent to the database for calculations.
Alternative Flow	1. User presses the map button. 2. No Wi-Fi APs detected OR location settings are turned off. 3. Notify users of failure.
Interacts With	Notify User, Data Testing, Wi-Fi Information Collection use cases.
Open Issues	NIL

Initial Design

Class Diagram

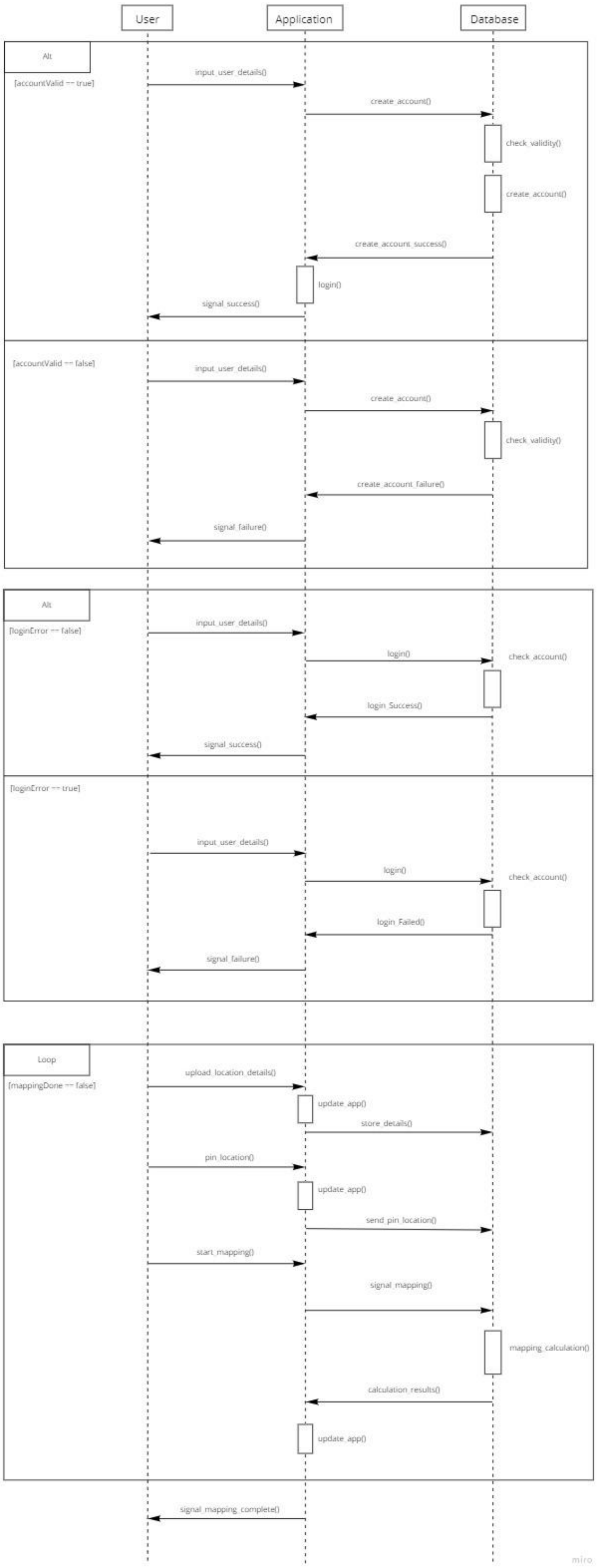


Use Case Diagram

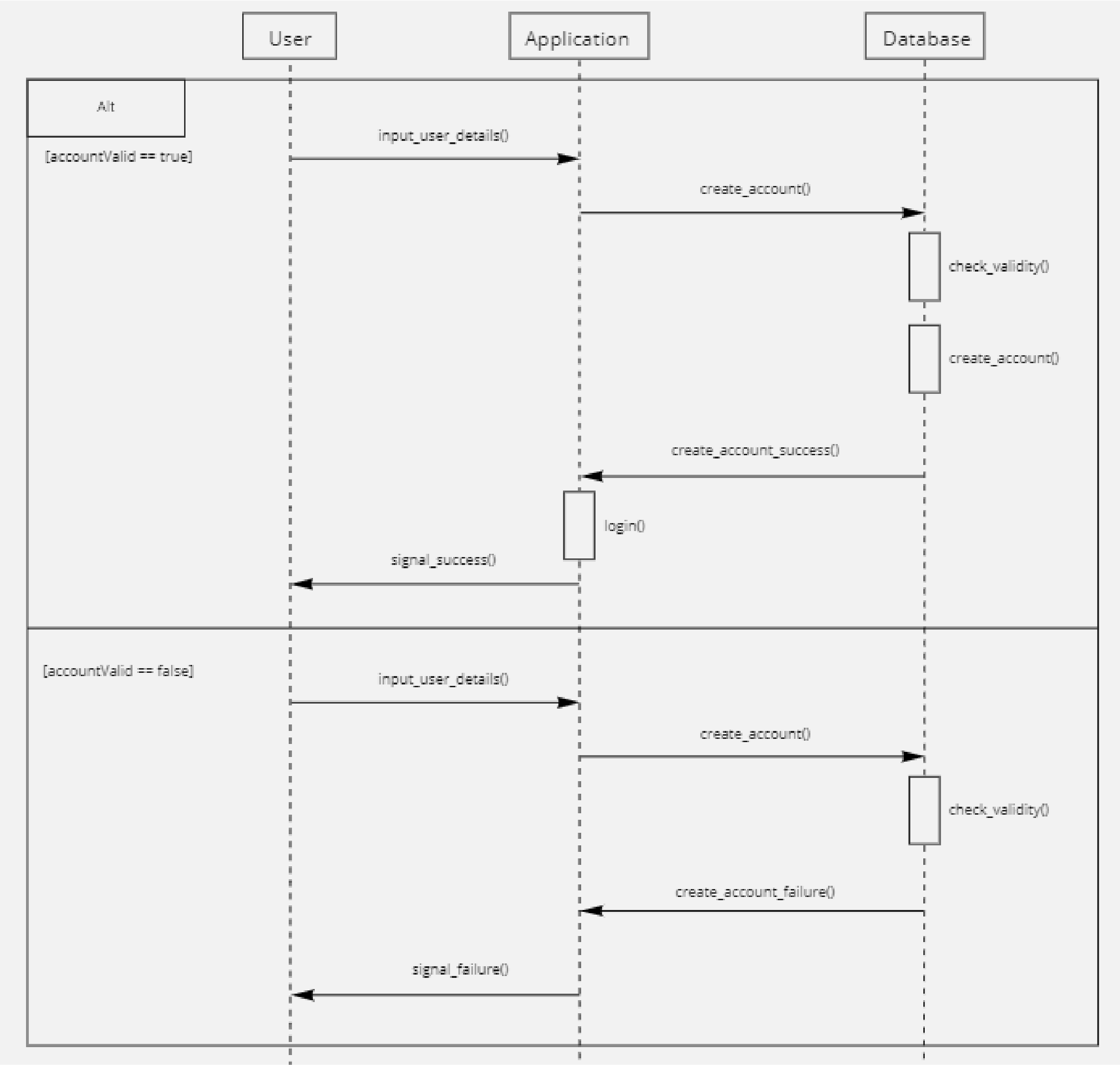


Sequence Diagram

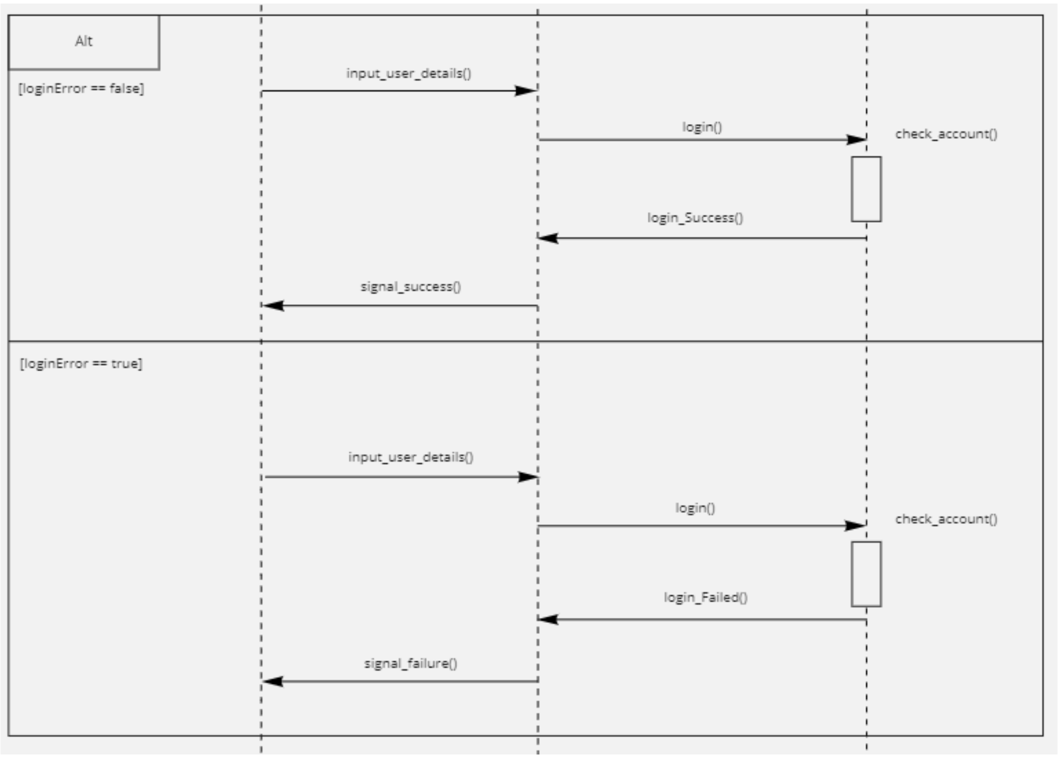
Overall Flow



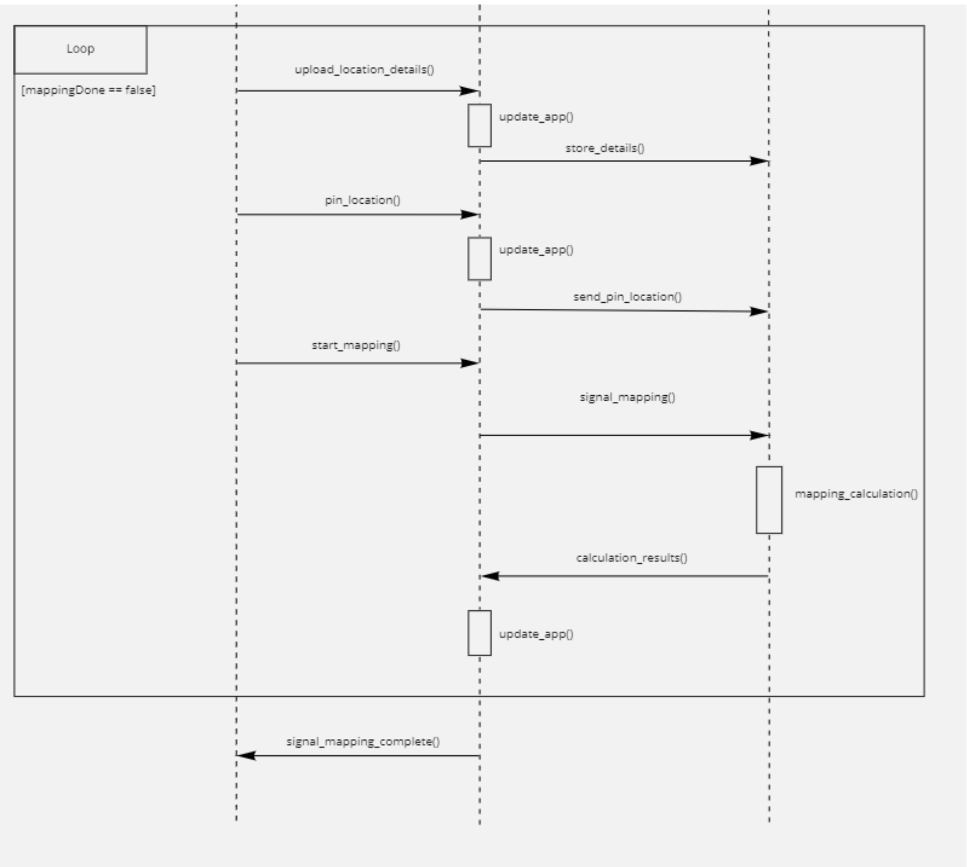
Registration



Login



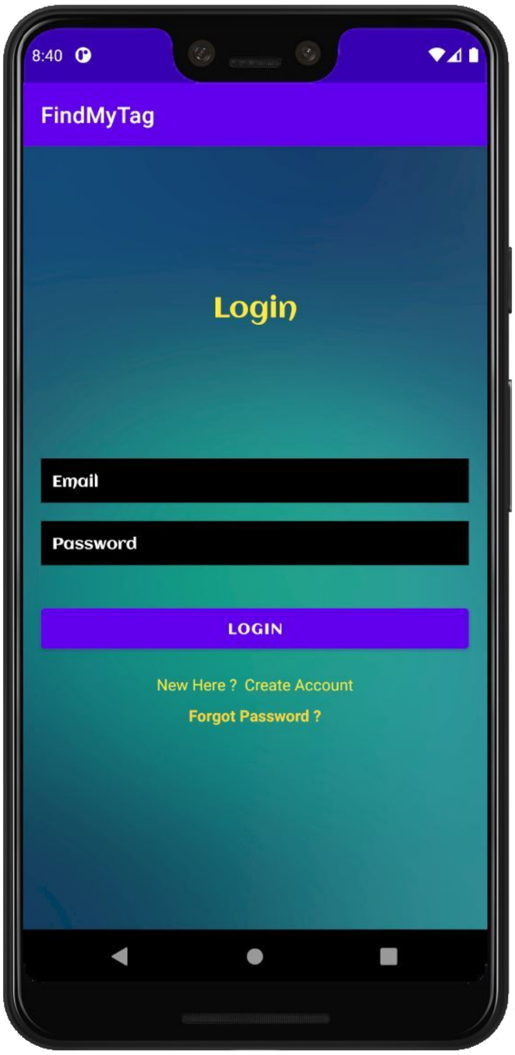
Mapping

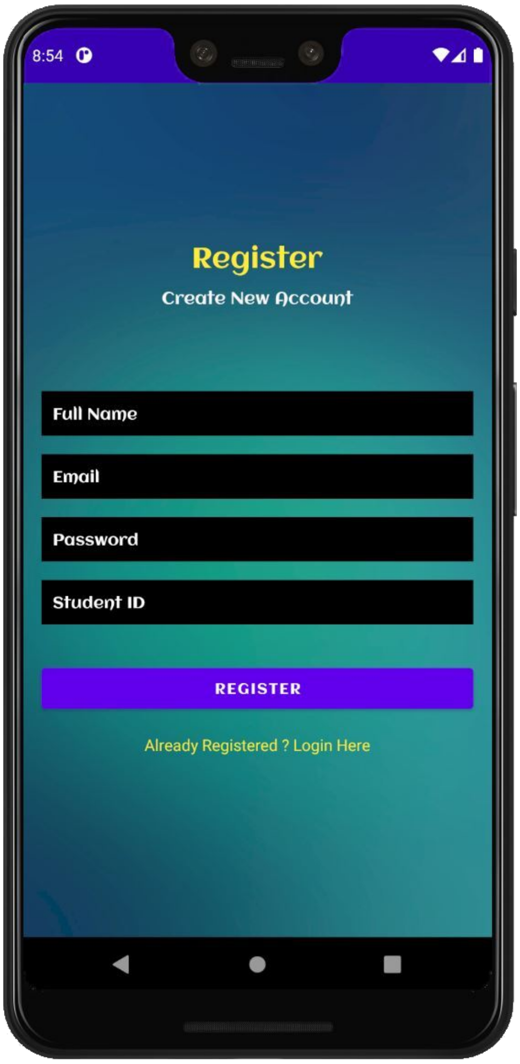


Implementation

Demo has been successfully conducted with our instructor, Professor Sudipta, to showcase some of our application's basic features.

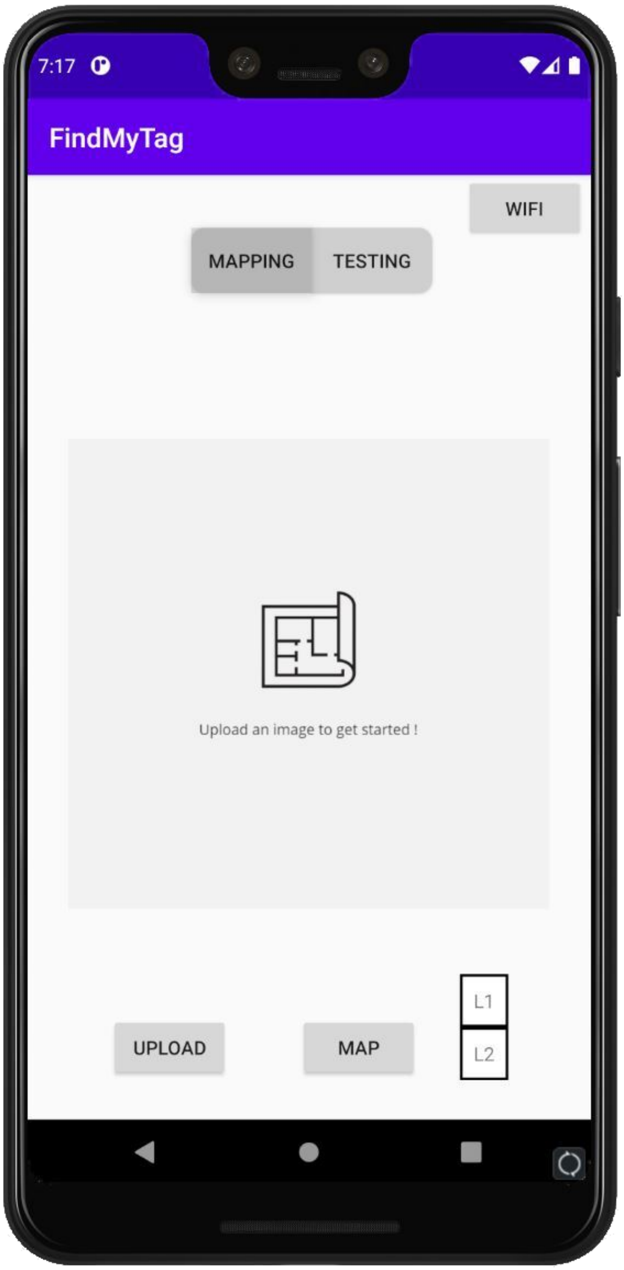
Login Page





The login page has been created with a functional email authentication login format while the account register page has been created that would allow for the creation of an account that would be reflected on the Firebase.

Mapping Page

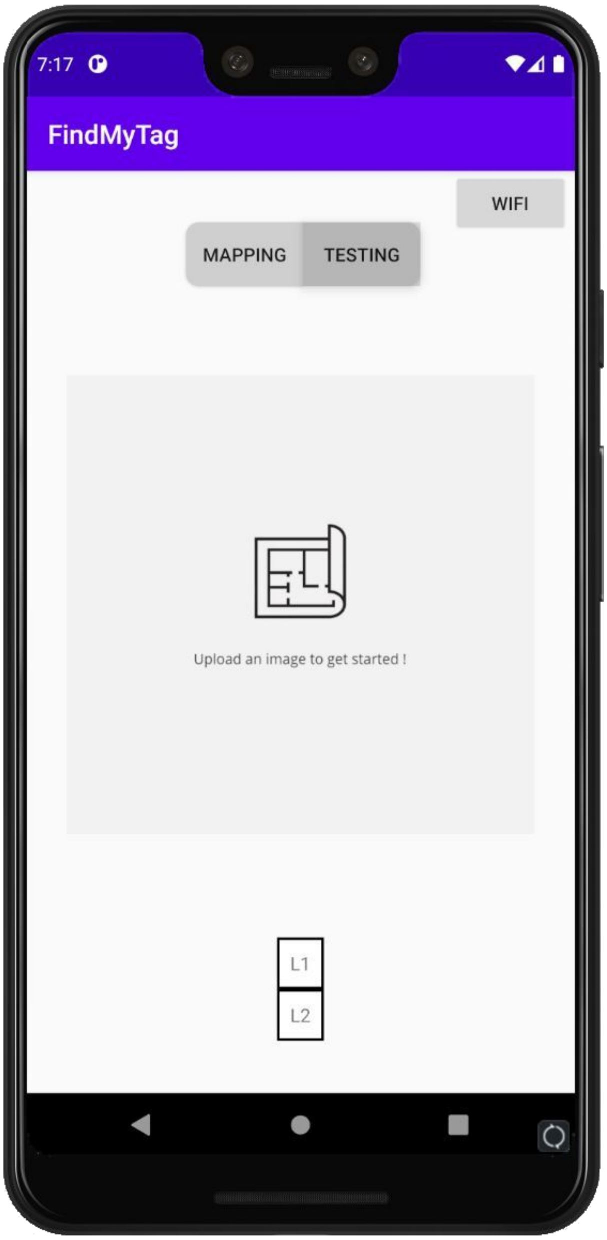


An initial mapping page has been created with a functioning upload button to allow for the upload of a layout picture.

Upcoming Implementations:

- The image would be uploaded to a database to be stored.
- A dialog popup would be added upon clicking the upload button to allow users to input other information that would be relevant.
- The floor section would be updated along with the inputted user information and would change accordingly depending on the selected layout.

Testing Page

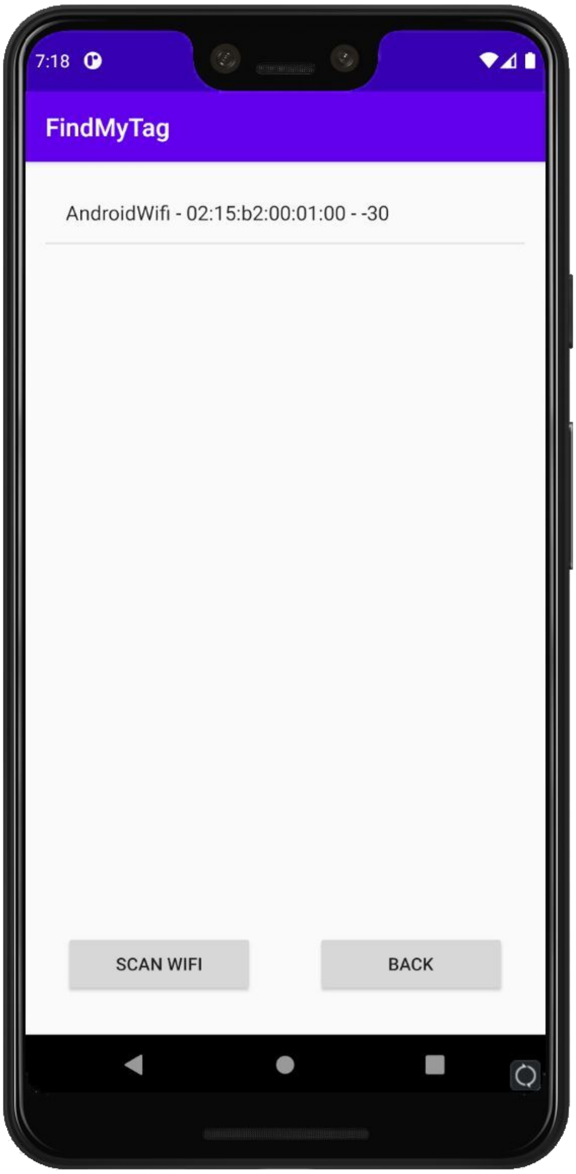


An initial testing page has been created for user tracking purposes.

Upcoming Implementations:

- The image would be updated when a floor plan image was uploaded in the previous mapping activity.
- The floor would also be updated accordingly after the user has inputted the required information and the mapping has been done.
- The user mapping relative to the floor plan image would be shown in this activity.

Wi-Fi Page



AndroidWifi - 02:15:b2:00:01:00 - -30

An activity is created to list out all Wi-Fi APs found.

Upcoming Implementations:

- The Wi-Fi APs that were found will be sent to the database to be used in the Wi-Fi algorithm.
- The important data to be sent would be the BSSID and signal strength in decibels.

Testing Plan

For our testing plan, we need to prioritize the different components accordingly since it is unfeasible and impractical for us to try to achieve 100% line coverage (in terms of white-box testing) for the sake of achieving 100% line coverage within the given limited time frame. As such, we prioritize the testing of the key features of our application, the algorithm, and the database management, over UI interactions, potential connectivity issues and unhandled exceptions invoked by certain user-centric actions.

Black Box Testing	1. Login a. Button press logs the user in and opens location activity. 2. Registration a. Button press opens the Registration page for users to key in their particulars. b. Once registered, automatically logs users in (persistence for login data). 3. Mapping a. Button press opens Mapping fragment. i. Users can press upload and pick the image to upload to the database. Images selected must appear in the designated area. ii. Users can press map and begin the mapping process. 4. Testing a. Button press opens Testing fragment. i. Users can see the image mapped on the screen. ii. Accuracy of the user's current location relative to the floor plan is within the given specifications of 2-5 meters error threshold.
White Box Testing	1. General a. Ensure that stable and compatible connectivity between Android app, database and cloud function can be established (authentication and I/O data format). b. Ensure that the necessary permissions required for the application to run smoothly and properly are given to the Android app and approved by the user. c. Ensure that the User Interface is stable and do not throw any exceptions/errors. d. Ensure that any possible/potential errors/exceptions are handled properly, and

	meaningful feedback messages are provided back to the user. 2. Login a. Test email and password authentication i. Valid email and password format is sent to the database for authentication. ii. Checks if the account exists in the database. 1. If the account exists, login is successful. 2. Else, login fails. iii. Valid email and password combination logs the user in. iv. Invalid email and/or password prevents the user from going to the next activity. 1. If the account exists, go to the password field. 2. Else, prompt the user to register for an account. 2. Registration a. Checks the validity of the email address and password. i. If the email already exists in the database, prevent the user from creating a second account. ii. If the email or password format is invalid, prompt the user to edit their registration particulars. b. Creates a new field in the database to store a newly created account. i. Newly created accounts can be used to login to the app. 3. Mapping a. The uploading process should result in the image being stored in the database and accessible for later use. i. Image size/format must be valid. ii. Image should appear in the database stored as a URL. 4. Testing a. There should exist an image in the database to be pulled by the user and displayed on the app. i. Image must be available when the user pulls from the database to the application. ii. Images used for mapping must be correctly displayed in the application.

Fault-Based Testing	1. Login a. Test invalid email ID (email address format). b. Test invalid email ID (empty field). c. Test invalid password (string.length < 6). d. Test invalid password (empty field). e. Test with valid email and password but without Internet connectivity. f. Else, valid email and password, login successful. 2. Registration a. Invalid email address (empty field or wrong format). b. Invalid password (empty field or string.length < 6). c. Account already exists (check for existing email ID in the database of list of existing accounts). d. Else, valid email and password, registration successful. 3. Mapping a. Invalid image file format (headers). b. Invalid image size. c. Attempt to upload an empty file (check content of file if it is zero bytes in size). d. Attempt to submit an image of 1-by-1 pixel. e. Attempt to map null images. f. Attempt to upload a floor plan without Internet connection. g. Else, attempt to upload a valid floor plan, should appear on the interface. Success. 4. Testing a. No image in database. b. No Internet connection. c. No available Wi-Fi APs around the user. d. The user moved to a separate, incompatible area different from the area specified by the floor plan.

Workload Distribution

Evidence of our group members' commit records can be seen on our version-controlled repository on GitHub here: https://github.com/jamestiotio/FindMyTag