

COSC 4P02 Software Engineering 2

Progress Report 2

1. Names and Roles:

Name	Brock ID	Email	Role(s)
Vincent Zhang	6697114	vz18iu@brocku.ca	Dev team
Hi Leung	6799969	hl19wl@brocku.ca	Dev team
Louis Wang	6116271	lw16bn@brocku.ca	Dev team
Yujia Zuo	6567150	yz18fa@brocku.ca	Scrum Master, Dev team
Hang Li	7319957	hl21bi@brocku.ca	Product Owner, Dev team
Ziyuan Xu	5476650	tx13al@brocku.ca	Dev team
Yifeng Zhang	6593487	yz18el@brocku.ca	Dev team
Jiahao Pang	6366173	jp17gh@brocku.ca	Dev team

2. Features we planned to implement this sprint:

- a. Map zooming feature: Complete
 - As the map is quite long, especially on the first floor, it would be helpful to let the users zoom in and out so they can find things more easily. This would also allow them to focus only on what's relevant (i.e., what's on their screen) instead of having to manually block out irrelevant information on their own.
- b. Map outline: Complete
 - We decided to rework the map implementation. Rather than using an image file like we did in the previous iteration, we decided to redraw the map using lines in order to facilitate our navigation function. These lines will act as walls/barriers to indicate where the navigation search algorithm can/cannot go.
- c. Linking database to Android Studio: Complete (using alternative methods)
 - The linking of the database to the application will allow us to query the data required for functions like logging in, searching, displaying items, navigation, etc.
- d. Login System: Works but needs additional features
 - This feature was intended to allow the museum staff to make changes to the map but restrict guest users from doing so. Having a login will allow us to differentiate the level of control the different types of users will have over the map.
- e. Search Bar: Works but needs additional features
 - Currently, the search bar can search items from the database. This feature allows users to search for specific items on the map. However, the viability of this is dependent on the data the museum has (and their naming scheme is not very make sense).
- f. Navigation: In Process
 - For this feature, we want to provide a shortest path between two locations. We do not have a fully fleshed out idea for this yet since the other prerequisites are being worked on and have a higher priority.

3. Coming features that we plan to implement:

a. Login system:

- We currently are able to log in but there are no differences between being logged in or logged out in terms of features.
- We need to update the UI and provide buttons to access map editing features like adding, removing and editing showcases (displays).
- We also need to implement a separate management screen for admin users to manage staff logins (i.e., create, delete or edit staff accounts).

b. Showcase system:

- Staff members should be able to click on the map and add or remove a showcase or edit it if one already exists at that location. This allows the map to be more dynamic and modifiable if changes are made in the future to the items on display or if showcases get moved around.
- Part of this involves creating a list of items per showcase but the implementation details need further discussion.
- Do a recycled view. After clicking the search list items Feedback

c. Database integration:

- Currently, we have successfully implemented a simple SELECT query (for login purposes). However, we also need to implement UPDATE queries to facilitate the other features like updating and/or creating new information related to staff logins, museum items and museum showcases (displays).

d. Search bar:

- Next, the search bar should allow the user to click on a search result to view more information. Clicking on an item in the search bar is supposed to display more details about that item.
- Create a new activity or fragment that will show the relevant information.
- Decide what information to display and how it should be presented (e.g. text, images, etc.).
- For employees who already know the item ID or name is when an employee wants to find an exhibit quickly, they can simply delete the item or move it to another location on the map and you can see Search for the latest location of exhibits

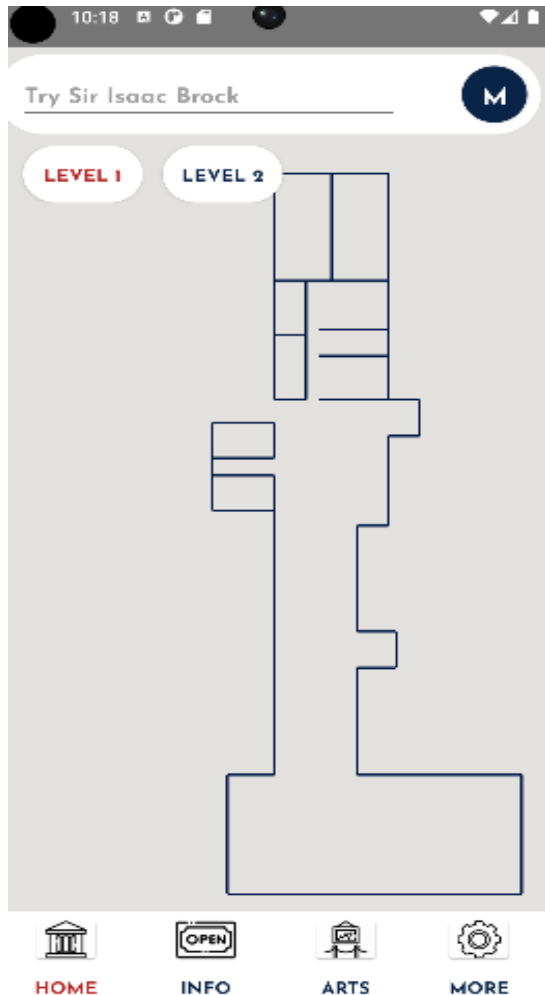
e. Navigation:

- Currently, the algorithm used to find the shortest path between two selected points on the map has been implemented.
- Need to show the path on the screen.
- Need to define the distance between two adjacent points.

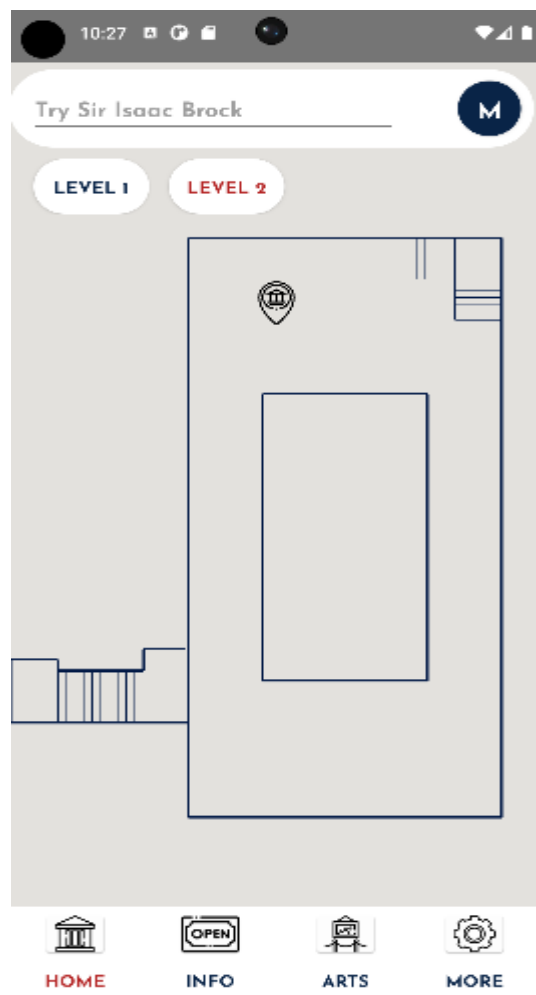
4. App screenshots:

Screenshots taken from Nexus 4 Emulator

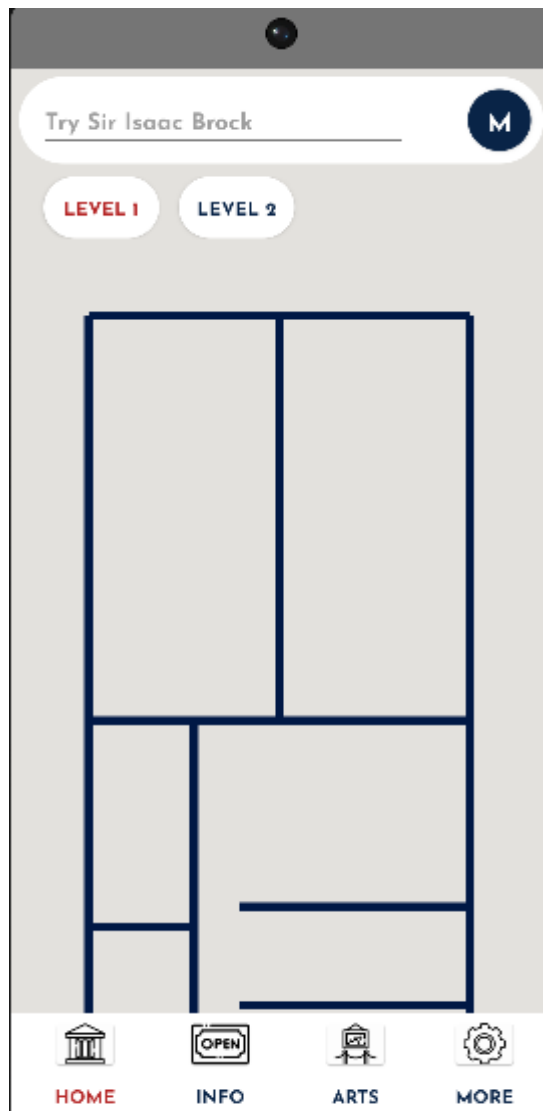
Portrait Layout of Floor One



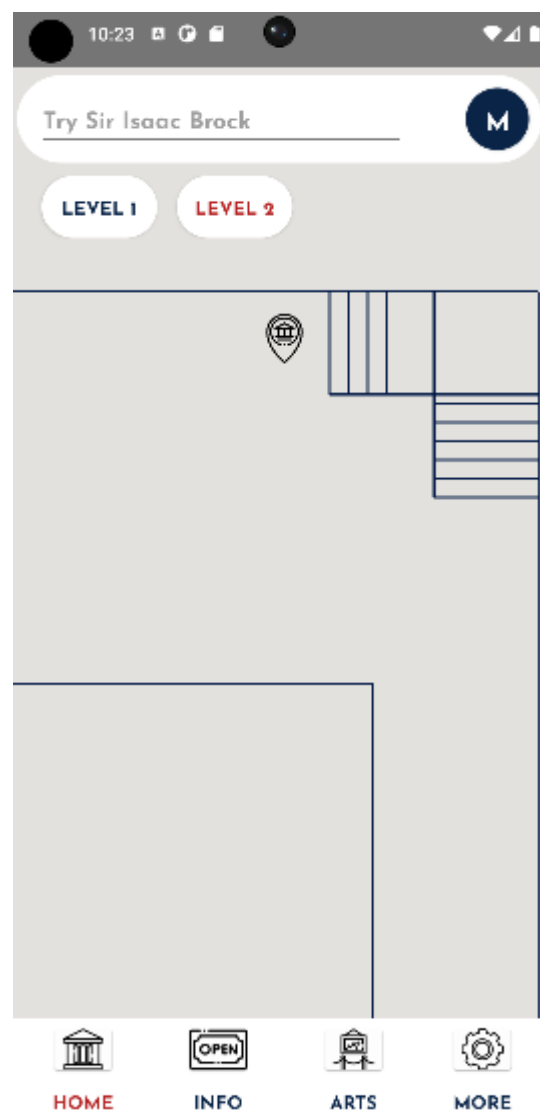
Portrait Layout of Floor Two with ImageButton



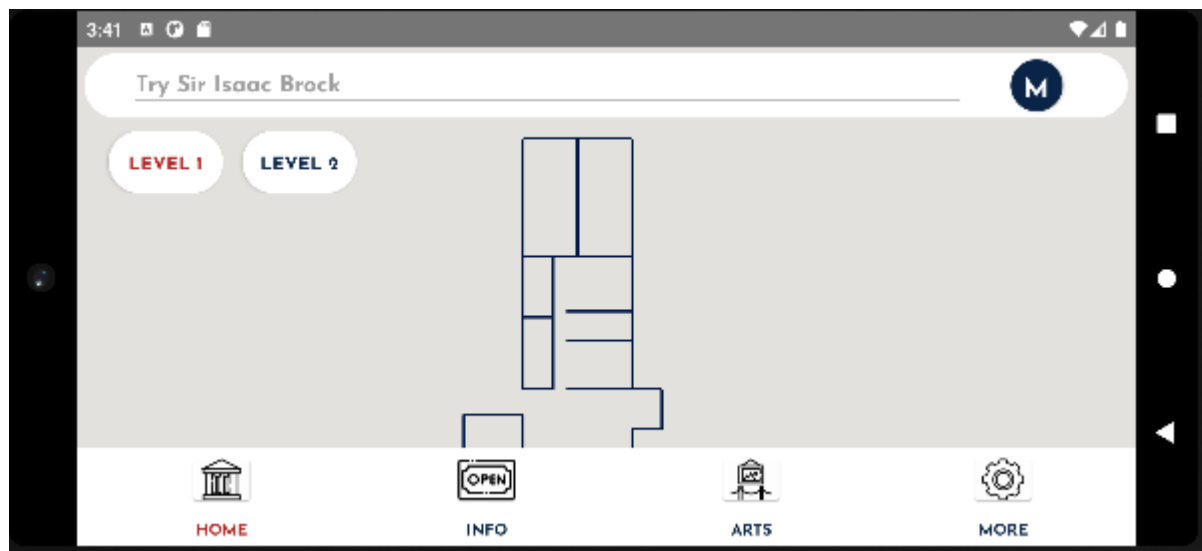
Floor One After Zooming in



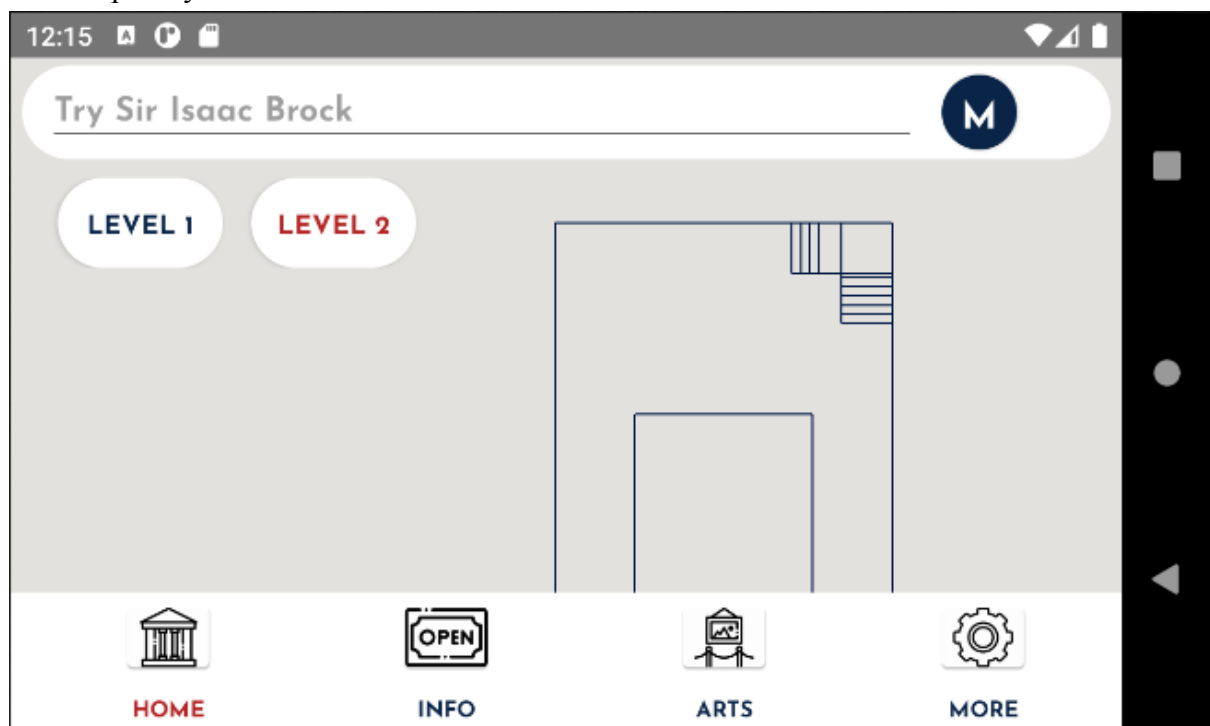
Floor Two After Zooming in



Landscape Layout of Floor One

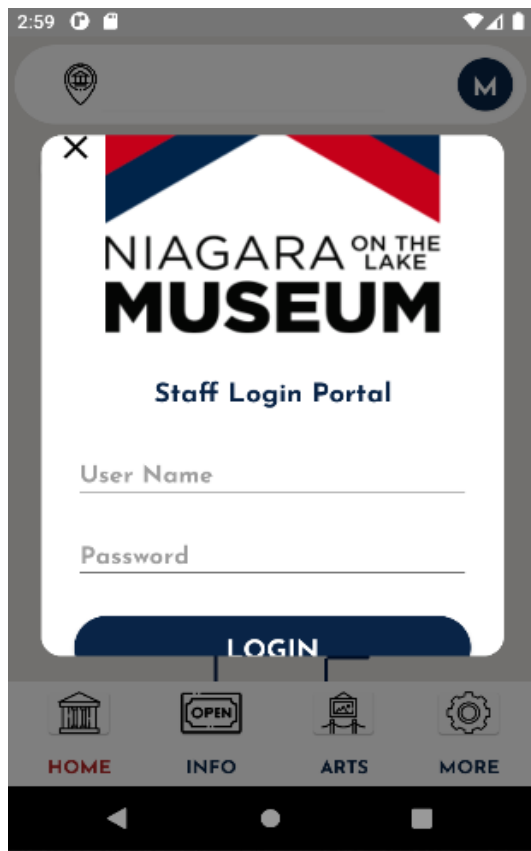


Landscape Layout of Floor Two



Login screen

- This screen is accessed by clicking the top right circle button (M button)



After Login Layout

- Staff can add and delete exhibitions show up on the APP

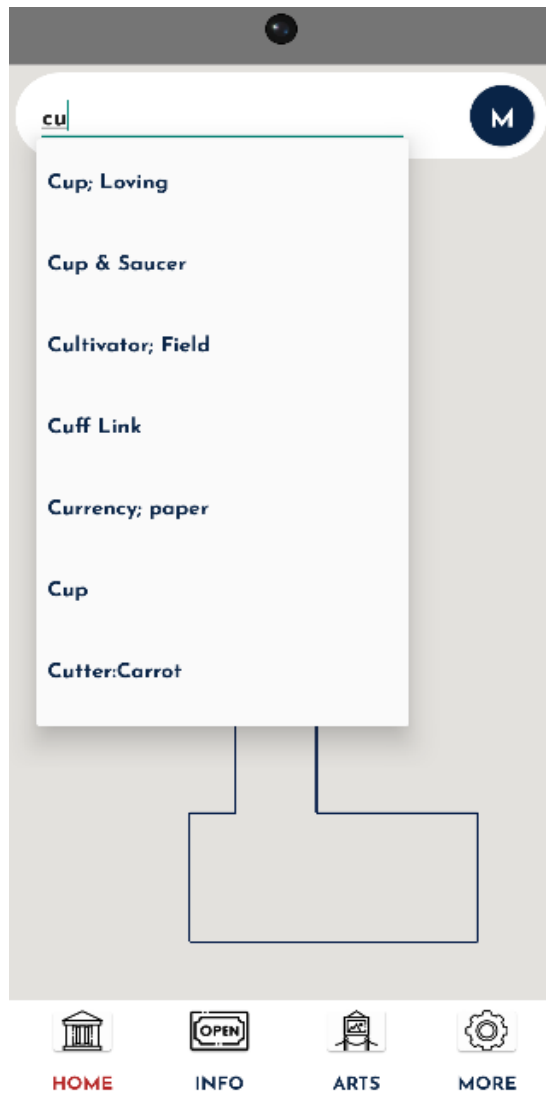


- Staff can also Logout by clicking the same Button as the login button

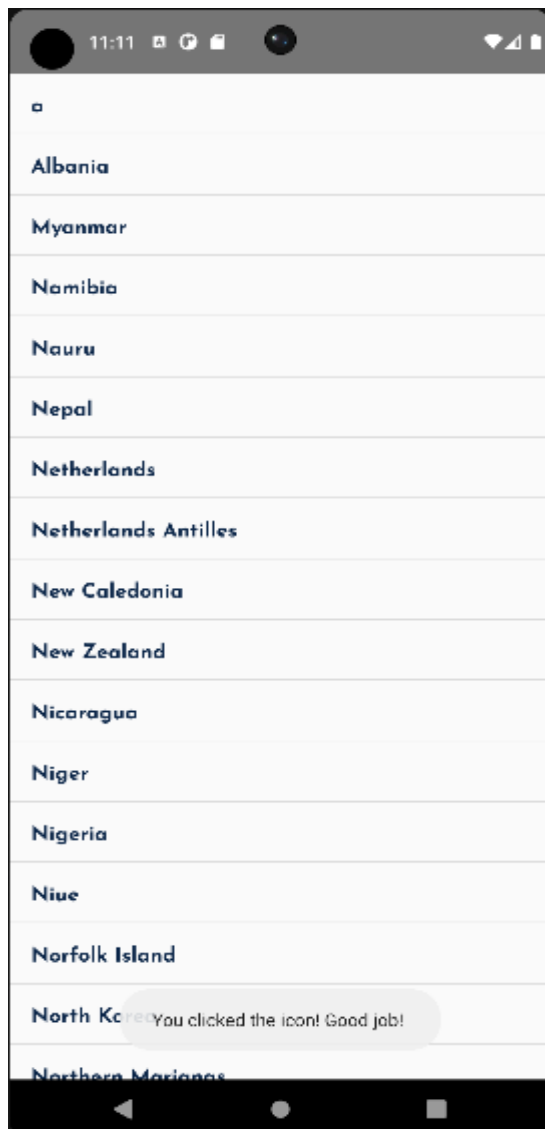


SearchBar Layout

- User can search items from database



After Click the ImageButton



5. Sprint

Sprint 3: Architectural redesign (Date)

- A lot of the progress in sprint 3 has been hindered due to the lack of database integration. This issue has been resolved now so development in sprint 4 should be more focused on feature implementation rather than technology integrations. Nonetheless, we have still made some progress regarding the redesigning of how we will store/implement the map. Details regarding all of this will be explained in more detail in section 7.
- Review: In terms of a functioning map, we do not have a working deliverable yet. However, we have begun implementing the basic features of the map application which shows great promise for the future. We have also successfully connected the database to the application which we are able to demonstrate through a working login system.
- Retrospective: Given our tight schedules with other courses during this sprint, we think our slower velocity compared to the other sprints are justified. We were also limited due to the issues involving the database, but the group members involved in resolving that issue have done the most they could with the time they had left based on their schedules. Meanwhile, the group members that have not been able to proceed with their original tasks have completed prototypes to test their code or pivoted to other tasks. Thus, our sprint overall has been as productive as we think it could be given external limitations.

Each member has also tried their best to arrive at every scheduled scrum meeting. However, we understand that life circumstances and course loads beyond the current course has made it harder for us to coordinate a meeting with every member present. Thus, we also tried to use our discord chat to provide small-scale updates on progress and used our scrum meetings to provide in-person demos of these updates or troubleshoot issues together. Although we have also started using our GitHub more often (which was a goal from the previous sprint retrospective), these in-person demos have been majorly beneficial as we are able to give live feedback and suggestions to each other. Overall, this change in our meeting strategy was helpful for keeping all the members up to date on the progress of our application and also in planning what to do between meetings.

- After the review and retrospective, we determined that our next steps were to continue implementing the methods that we have planned to implement. As there are no major system-wide implementations left, this was the logical way to go.

6. Issues:

Solved:

- a. Our biggest setback from the previous sprint was not being able to connect the database to the application. This has been solved and the details will be explained in more detail in section 7. However, the general overview involves scrapping the more secure “REST API” implementation that was too convoluted and time consuming to implement for a more direct connection. To offset this lower security level, we have implemented a way to hide our connection information (more explained below).
- b. The search bar successfully connects to the database and searches for items from the database.
- c. After logging in, we can see different layouts, and the dialog to add the closet to display the exhibits.
- d. Adding an Imagebutton without the previous external layout.

New issues:

- a. When implementing the zooming functions, the platform’s built-in zooming functions did not work how we intended it to with our coordinate system. Clicking on the system would work based on the screen’s absolute coordinates rather than the coordinates relative to the map.
- b. The first floor and the second floor painting jobs are so different, we are trying to make them similar to be compatible with other common components.
- c. For the Imagebutton, we can set it on the map without previous layout now, but the position of the button would not be modified after zooming in or zooming out.
- d. We can only save the URL of exhibits’ images in our database, so we need to find some way to display those images in our apps based on the url.

7. Contributions and achievement:

Front-End:

- Created a search bar function that allows for easy access to the exhibition list from our database table. (Yujia Zuo)
- Navigation function: the application can find the shortest path from one position to another, and display it on the map. (Jiahao Pang)
- Log-in System: Employees can login with the username and password. After logging in, the system enters an “edit mode” where the employee can change

the locations of showcases in the exhibits. Doing so will also modify the information within the database. (Louis Wang)

- Map Display Interface: Show the layout of the map (First floor: Ziyuan; Second floor: Yifeng Zhang; Combine: Vincent Zhang)
- Efficiency and make the code tidy. (Hang Li)
- Overall members: Yujia Zuo; Jiahao Pang; Louis Wang; Vincent Zhang; Yifeng Zhang; Ziyuan Xu

Back-End:

- Database Connection: connecting the Android app to the Postgres database hosted on AWS.
 - As the REST API was not set up successfully (even after 2 weeks of troubleshooting since the last sprint ended), Hi Leung decided to use the alternative of a direct JDBC connection. This was our backup plan suggested by Hang Li in case the more secure implementation (REST API) did not work. (Hi Leung and Hang Li)
 - Thus, we added the Postgres JDBC drivers into our application and tried to set it up but instead ran into some problems. There were some driver incompatibility issues which were not apparent at first since the error codes suggested an issue regarding a missing file path. However, after Hi and Hang tried to fix this error, we determined that this was not the case. It was not until Hi luckily came across a stackoverflow post stating that this error may have been caused by driver issues and thus he tried many drivers until one worked (42.2.5). This allowed us to finally connect the database for the first time. (Hi Leung and Hang Li)
 - As for the security of our connection information, Hi found a guide online that shows a method for hiding keys. Although this is typically only used to hide API keys, Hi has adapted it to also hide the connection details (URL, database name, username, password, port). Although this is suboptimal, it is a better compromise than hard coding our connection details into the class and exposing it on GitHub. One advantage of this hiding method is that it gets “ignored” by GitHub when pushing meaning it is not pushed to GitHub. Thus, although we are not able to provide a secure connection through REST APIs, we still managed to provide a slightly less but still secure alternative. (Hi Leung)
 - Key hiding implementation source:
<https://yfujiki.medium.com/how-to-store-use-sensitive-information-in-android-development-bc352892ece7>

- One of the other fixes that allowed us to successfully make the connection involved thread permissions. Android does not allow for external connections from threads for security reasons so we had to make modifications to thread permissions. At first Hi Leung settled on a method involving “Strict Mode” but Hang Li has found a better implementation afterwards and has updated the code to fit that implementation. (Hang Li and Hi Leung)
 - Strict mode fix source:
<https://stackoverflow.com/questions/16666619/postgresql-connection-with-database-in-java-fail>
 - For the thread connections, we decided to have it connect before each query and disconnect after it. This is because our program is not large and the database is limited to less than 2000 rows meaning disconnecting from the database server can release the pressure of the database. (Hang Li and Hi Leung)
 - Set the database class and methods in it to be static as it is convenient for everyone to call it directly from anywhere (abstract enough). (Hang Li)
- Overall members: Hi Leung; Hang Li

8. Meeting minutes (logs)

[07/03](#)

[14/03](#)

[23/03](#)

[28/03](#)

[04/04](#)