# COSC 4P02 Software Engineering 2

Final Report

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, Main writer |
| 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. Login system: Mostly Complete
      - Employees can use the account name and the password to log in to the modifying system. (incorrect password or incorrect username which cannot match the data in our database would be notified.)
      - When staff log-in, they are able to modify exhibits closets which are also called showcases in our app (see below for more).
      - Additionally, they are also able to add and remove artefacts (which we referred to as items during development) from showcases.

   b. Showcase system: Mostly complete
      - Showcases are now displayed on the map (edges and pin buttons together) allowing both visitors and staff to interact with them in different ways depending on whether or not they are logged in.
      - Visitors are able to interact with the showcases by clicking on them and viewing the items inside of them. This also gives them access to the navigation system (see below for more).
      - On the other hand, staff members are able to create new showcases, change the details (position and size) of existing showcases, and remove showcases.This allows the map to be more dynamic and modifiable if changes are made to the museum in the future through changes in the items on display or if showcases get moved around.
      - We've also implemented an algorithm to prevent overlapping showcases
      - Although the UI has not changed much since the last report, the functionality was only implemented this sprint.

   c. Database integration:
      - Since the last sprint, where we only had SELECT queries, some UPDATE queries have been implemented allowing us to make changes within the app and have them reflected in the database. This allows for the adding, removal, and editing of showcases and items.

   d. Search bar:
      - We have a functioning search bar now where users can search for the items (artefacts) by name. This brings up a list of results showing the name and pictures for easier identification. When the user clicks on it, a pop-up will come up giving them information about the item.
      - Additionally, if the artefact is on display, the showcase will also be opened in the background and skip to that floor so that users can immediately access the navigation features or view the other showcase items upon closing the pop-up.

e. Navigation: Complete but could use some improvements
    - The shortest path algorithm has been implemented and works by selecting two points (start and end) using the showcases, bathroom entry and exit. After selecting the two points, a line will be drawn showing the shortest path between them without crossing the obstacles and the edge of the map.

**Features we did not plan to implement but had to be implemented as they were seen as high-priority features to have.**

f. Offline mode: Complete but could use some improvements
    - When opening the app without an internet connection, the app is put into offline mode meaning interactions are quite limited. However, the app still functions as a map telling users where the toilets, exits and staff counter is.

g. User manuals: Complete (see links below in section 4)
    - We added user manuals to help the less tech-savvy museum visitors who may want to use the app to enhance their experience at the museum. These manuals are to be maintained regularly, including updating the screenshots within the manual, whenever changes are made (e.g., updates to current features or new features).

## 3. Coming features that we plan to implement (hypothetically):

a. Login system:
   - Whilst staff are able to add and remove items, we still need to implement the ability for them to edit items from within the app.
   - We also need to implement a separate management screen for admin users to manage staff logins (i.e., create, delete or edit staff accounts). Currently, this can only be done through the database.
   - The device can remember the account number and password as previous successfully login.

b. Showcase system:
   - Currently, the showcase only allows them to view the items. Thus, we need to give staff members the ability to edit the item's contents from within the app.
   - Furthermore, when showing the items (in the pop-up), it would also be beneficial to add a section stating whether the item is displayed or not. If it is displayed, we would also inform the user about where it is located (e.g., floor number and which part of the building it is in).

c. Navigation:
   - Currently, the shortest path line shows exactly the shortest path. This means that if we have to go around a corner, the line will go to that corner creating a diagonal. This is not how it is typically represented in other map applications so work will have to be done to smoothen out this line so that the lines look more natural (e.g., 90 degrees when turning a corner).

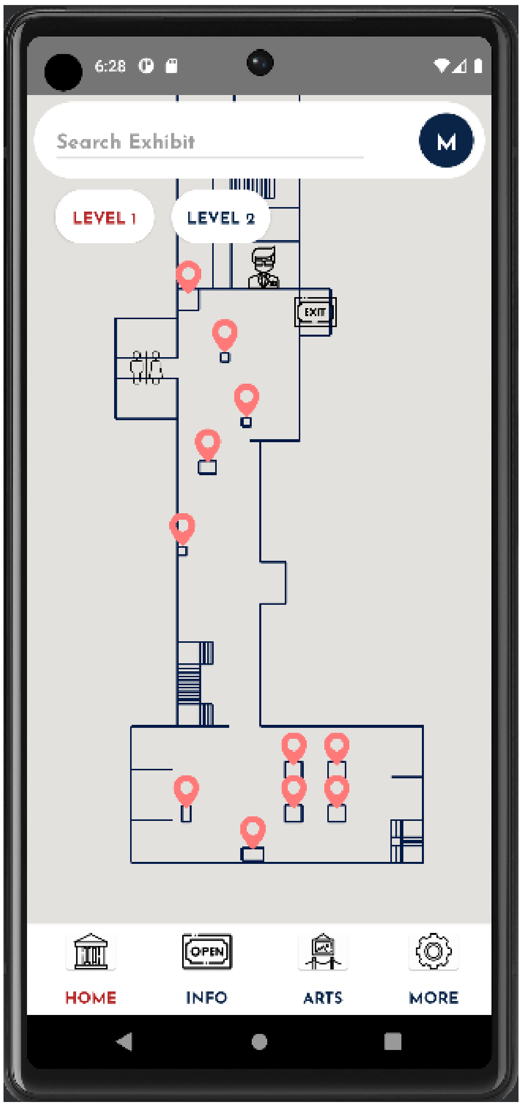
d. Offline mode:
   - Currently, the offline mode is not very interactive as many of the functions have been disabled due to the lack of internet connection meaning information cannot be retrieved from the database. To improve the interactivity, we would like to make the navigation features available with the pre-loaded POIs (toilets, exits, etc.) as they are static right now and cannot be interacted with.

4. **App screenshots:**

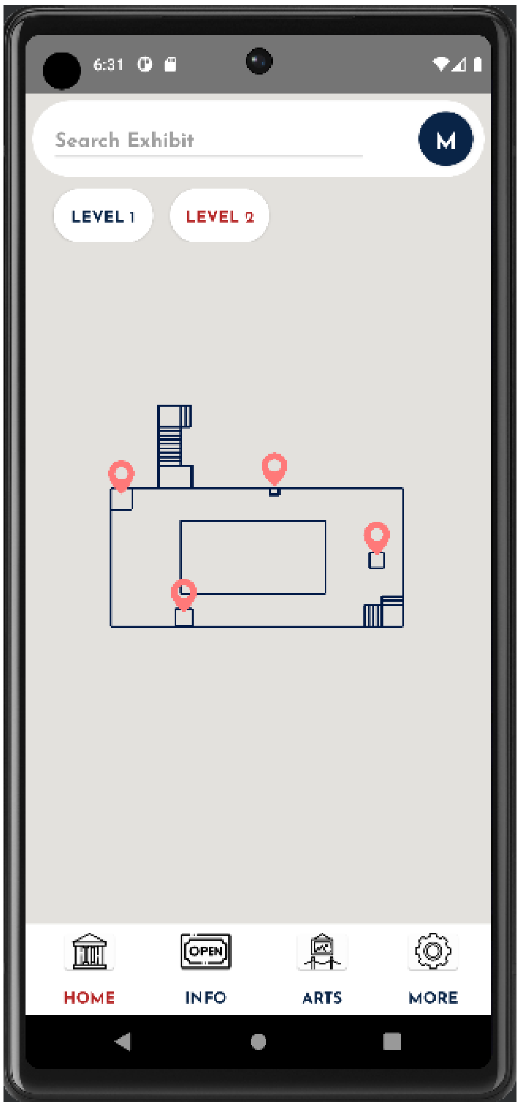
Screenshots taken from Pixel 6 Emulator

**Home Screen**
Since the last sprint, we have added the ability to display showcases (exhibits) onto the map. These are represented by the orange pins. We also included the locations of the toilet, exit, and staff counter on the map as these are valuable locations in case of an emergency.

Floor One



Floor Two



To have a more in-depth understanding of the features available in the application, please refer to the user manuals as they have more detailed explanations and annotated screenshots accompanying those explanations:

1. Visitor version
2. Staff version

## 5. Sprint

Sprint 4: (5/4 - 30/4)

- Sprint 4 has been very productive allowing us to implement a lot of the features from our backlog which have been pushed back due to external software issues. Now that those software issues are resolved and troubleshooting is not the main focus of this sprint, we were able to focus primarily on providing our users with the core functions of the app.

- Review: We have finally reached a point where we have a working, deliverable application. Although there is much room for refinement, the core features are all available to both the visitor and staff user groups. Between the 3rd sprint and now, we have gone through multiple iterations of the UI design to provide a better interactive experience as the buttons are more visible (i.e., indicative of their functions). This should help the user have a smoother experience. Nevertheless, we are still providing visitors with a user manual which can be accessed from within the application itself (but the staff manual has to be accessed from github).

- Retrospective: As we all have different courses, many of which do not overlap, it was difficult for us all to meet despite having the allocated lecture times. As the project was due later in the term, many of us decided to allocate our resources towards our own work (i.e., work from other courses or work outside of school). Nevertheless, we still were able to come together again towards the end and produce work at a rapid pace. We were significantly behind on our initial plan from the previous sprints. However, we were able to mostly catch up to our initial project proposal.

  Unfortunately, given our scattered schedules, we had some difficulty with understanding each others' code and with knowing who did what. Although we have become more proficient at using collaborative tools like GitHub, it does not change the fact that there is a steep learning curve to Android development which means that one person working on one thing may not understand what others are working on. For example, a backend developer may not understand how the UI is organised and implemented. However, whenever we encountered issues, our members were very cooperative, patient, and willing to explain things to others. Thus, even if a backend developer does not know how to implement front-end things, they are still able to provide assistance in terms of critical and creative thinking to help fix unexpected issues.

Admittedly, we could have done a better job at using Jira. Our group mainly used it as a way of tracking the backlog items and who was in charge of which item. As the interface was not very intuitive to use, we mostly kept track of our progress outside of Jira (e.g., within our meeting minutes) and updated our Jira at the end of each split. We also faced issues regarding sprint management within Jira as we were unable to edit our backlog/roadmap page despite looking for resources on the internet for how to do that. This was likely due to the interface being different on the free cloud version as opposed to the interface shown on the resources we found online. Thus, for future projects, we may use an alternative solution to keeping track of our backlog and to record our progress.

In the final sprint, we discovered that Scrum is effective in managing collaboration and the allocation of tasks. As discussed above, if we were more familiar with our Scrum tools, or alternatively if the Scrum tools fit better with our development, we would have had more success with the methodology as opposed to a traditional waterfall style of development. This was partly due to it being very difficult to implement the component of Scrum/Agile which states that all members may switch roles with each other and be able to function as normal. Perhaps this could be attributed towards the nature of app development where there are significant differences between the syntax of back-end and front-end code. Thus, for the future, it may be worthwhile to spend more time planning out development prior to starting the sprint so that we would be able to better understand what each member is doing and how we can adequately support each other.

As stated in the previous reports, we had a significantly slower start given the time taken for us to decide between creating an Android application and website. Furthermore, the majority of our group was unfamiliar with both types of implementations. In order to best fit the user’s needs, we settled on creating an Android application which we all had to learn how to do from scratch. Thus, we were not aptly prepared in advance for this project in terms of knowledge or experience. To combat this, we decided to go into programming earlier to learn the tools and knowledge required to put together the application. This made planning and management very difficult as we all learned how to use Android studio and AWS at different rates causing us to fall significantly behind on development. Given that we realistically only had a month to put together the application after preparing all the prerequisites (i.e., database, android programming knowledge, etc.), we believe we did a good job on getting the application to where it is now. However, we do acknowledge that because of this, we were not able to follow the Scrum methodology as closely as we had hoped to.

Lastly, we did not perform any unit testing. This was an intentional decision as much of our testing required hands-on (literally) testing which required interacting with the Android emulator. This would be difficult to simulate as a unit test. Thus, the majority of our testing was done manually. Although this might lead to poorer efficiency, we were able to make up for that with creative testing which let us discover bugs that were missed during development. For example, we encountered a bug when trying to navigate from point A to B, and then B to C which caused a crash on point B when switching between the pair of points. Thus, although we believe that unit testing is important, especially for more back-end heavy programs, since our program is mainly interactive, and therefore front-end heavy, manual testing was the better route to take.

- After the review and retrospective, we determined that our next steps are just to refine our existing features. Polishing up the features will lead to better performance when using the app and therefore provide users with a better user experience. We may also choose to implement some of our lower priority features that were not implemented due to time constraints. (P.S. The paperwork is much more important than the program itself. Good paperwork can save a lot of development time.)

6. **Issues:**

<u>Solved:</u>

a. Dynamically layout for the item lists for each showCase.

<u>New issues:</u>

a. The application could not execute without the internet after some features were added, so we need to check what happened to that.
b. The item editing function should be able to modify the image of the item, but the images of items were saved as a URL of the image from the internet. We need to create a URL for the uploaded image.

7. **Contributions and achievement:**

<u>Scrum Master</u> (Yujia Zuo):

1. Organize all sprint plans, sprint reviews, and sprint review meetings to record project progress.
2. Supporting the Product Owner to ensure that the product backlog is well-maintained and prioritized, and shows everything in Jira.
3. Through twice-weekly meetings, a collaborative environment is fostered that encourages open communication, mutual assistance, collaboration and continuous improvement between teams. Facilitate project discussions, drive team cohesion, and resolve conflict or disagreement through face-to-face meetings.

Product Owner (Hang Li):

1. Requirements collection.
2. Split jobs to everyone.
3. Ask 2 people to work together as a group in case someone messes up something (or quit).
4. Solve the technique questions with the team. (Collecting the ideas and deciding which one is good.)
5. Ask for a refactor if it is not compatible with other features.

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. (UI: Louis Wang, Back-end logic: Hi Leung + Hang Li)

- Map Display Interface: Show the layout of the map (First floor, colour matching, and design: Ziyuan Xu; Second floor: Yifeng Zhang; Combine: Vincent Zhang; Modified by Hang Li; Jiahao Pang to fit the future features)

- Efficiency and make the code tidy. (e.g. Merge some reused code; Ask for a limited number of data from the database every time.) (Hang Li)

- Implemented item, showcase, map pin objects, including creating, editing, removing and accessing methods. (Yifeng Zhang)

- The map pin can be put on the first and the second floor dynamically. (Yifeng Zhang)

- Create some singleton for the sharing data between different activities. (P.S. avoid re-request data from the database.) (Hang Li)

- The map pin on the map can be interacted with to view a list of items in it. (Hang Li)

- The items in the list can be viewed with more details. (Hang Li)
- Adding showCase function. (Louis Wang; Hang Li)
- ShowCase confliction checking algorithm. (Yifeng Zhang; Hang Li)
- Deleting showCase function (Hang Li)
- Moving showCase function. (Hang Li)
- Search for an item to add to a showCase function. (Hang Li)
- Deleting item from a showCase function. (Hang Li)
- Navigation Interface design. (Pin would be changed after selecting start or terminating.) (Hang Li)
- Washrooms and staff and exits can be displayed and interactive on the map. (Hi Leung, Ziyuan Xu and Hang Li)
- Employee logout function. (Hang Li)
- Some manual testing after feature release. (Hang Li; Yifeng Zhang; Ziyuan Xu; Yujia Zuo;  Louis Wang; Hi Leung; Vincent Zhang)
- Information page for showing the museum information like operating hours, ticket price, address, parking information, and contact information. (Ziyuan Xu)
- Different font sizes function. (Ziyuan Xu)
- All arts display page. (Ziyuan Xu)
- Added the visitor and staff user manuals into the application under the “info” menu in the visitor view and the “more” menu in the staff view respectively. (Hi Leung)
- Overall members: Yujia Zuo; Jiahao Pang; Louis Wang; Vincent Zhang; Yifeng Zhang; Ziyuan Xu; Hi Leung

<u>Back-End:</u>

- Efficiency
    - Make the code abstract, use a static function in the DatabaseHelper, and all the functions in databaseHelper have no side effects of any code after executing. (Hang Li)

- Staff log-in
    - The database interaction job while the staff add/delete items in the showcase or showcases on the map. (Hang Li)
    - An algorithm to detect if the new/moved showCase is located at an available position. (Hang Li)
- Navigation
    - The navigation algorithm was implemented this sprint by Jiahao. This allows the user to pick the starting point and ending point (both are exhibits) so the app can automatically generate the shortest path for them without crossing obstacles and the edge of the map. (Jiahao Pang)
    - Upon testing, Hi discovered that setting an existing endpoint to be the next starting point will cause the app to crash. This issue was resolved by Jiahao and Hang. (Jiahao Pang, Hang Li, Hi Leung)
- POIs (points of interests)
    - The POIs were added towards the end of the sprint as we realised that this is a necessary feature we overlooked. The POIs included the toilet, exit, and staff counter - which should all be displayed on the map.
    - At first, Ziyuan implemented this as a static image but upon further discussion between Hang and Hi, we decided that these should be re-implemented as a hard-coded showcase. This would allow users to use the navigation feature with the POIs either by going between 2 POIs or between a POI and an exhibit. (Ziyuan Xu, Hang Li, Hi Leung)
- Offline mode
    - We also added this feature quite last minute (around half-way through the sprint) as we had not previously considered what would happen when an internet connection was not available.
    - Previously, trying to access the app without an internet connection would lead to a crash due to the app trying to access the database but not being able to. This would prevent users from being able to use the app in case of an emergency and an internet connection is unavailable.
    - Hi created a “offline mode” for the application by disabling features that involved database interactions. This meant that the only features available were those which are hard coded (e.g., moving the map around, changing floors, zooming in and out, etc.). As this wasn’t very useful, Hi decided to reuse Ziyuan’s previous code from the POI to create static images of the POIs when a connection was not available. (Hi Leung, Ziyuan Xu)
    - Optimally, these should be presented as showcases so that the navigation functions would work with it. However, as these POIs are all located on the ground floor and within the central area of the museum, we decided this was not an urgent implementation and could be deferred to a later sprint.
- Overall members: Hang Li; Hi Leung; Jiahao Pang; Ziyuan Xu

Beyond the code

- User Manual
    - The user manual was written alongside development with changes made as new features were added/modified. For example, originally only the visitor's user manual was accessible in the app. However, Hi thought that this would be inconvenient for the staff who need to refer back to the manual so he added the staff's manual to the app too.
      (Hi Leung)
- Overall members: Hi Leung

## 8. Backlog and Work Allocation

Backlog Documents:

1. Product Backlogs (Hang Li, Jiahao Pang, Terry Xu, Yifeng Zhang, Louis Wang):
   Product Backlog

2. Meeting Backlogs (Yujia Zuo; Hi Leung):

   a. 14/01
   b. 24/01 and 28/01
   c. 02/02
   d. 09/02 (Review & Retrospective)
   e. 16/02

   We did not meeting during the reading week and exam day (Feb. 21st, Feb.23rd, Feb.27th)

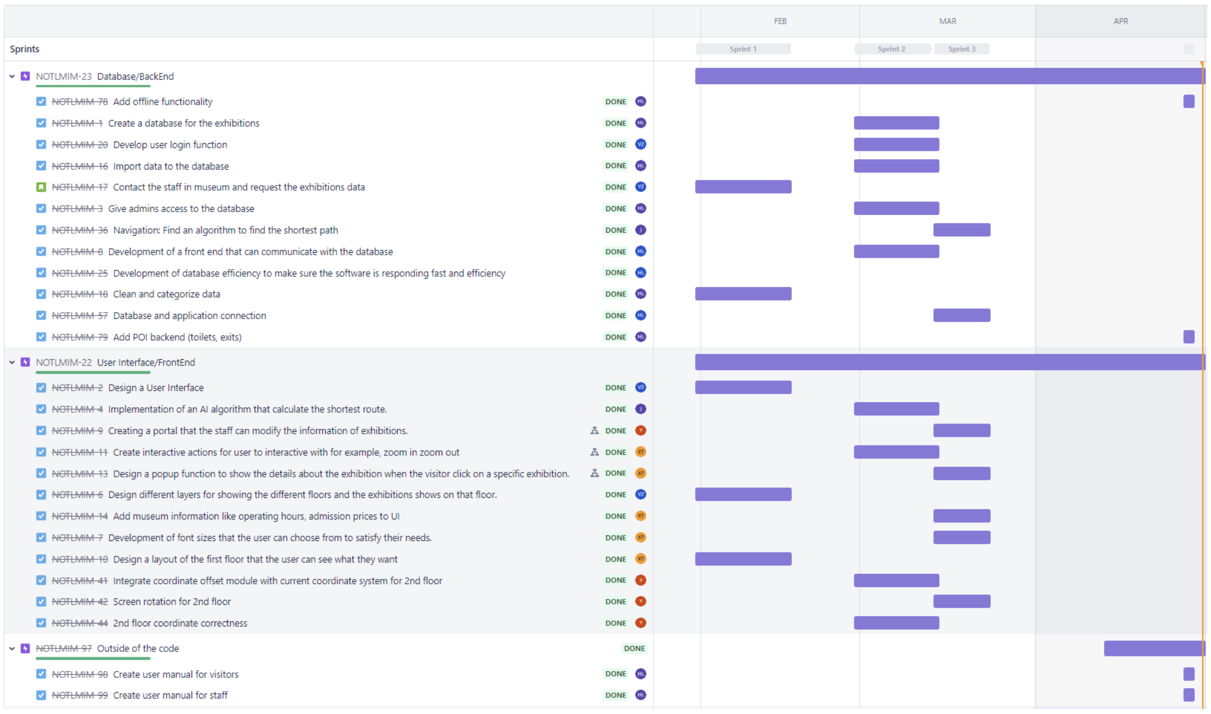
   f. 02/03 (Review & Retrospective)
   g. 07/03
   h. 14/03
   i. 23/03 (Review & Retrospective)
   j. 28/03
   k. 01/04 (Small Group Meeting)
   l. 04/04
   m. 08/04
   n. 15/04

   We did not hold a meeting on the week starting on 17/4 due to exams and assignments. Also, due to exams and Assignments, it is hard for Scrum Master to protect teams from external distractions or outages.

   o. 27/04 (Review & Retrospective)

Jira

This is a screenshot of our roadmap from Jira. This also acted as a pseudo-burndown chart as we did not have access to a dedicated burndown chart.



Sprints

NOTLMIM-23 Database/BackEnd
- ~~NOTLMIM-78~~ Add offline functionality DONE HL
- ~~NOTLMIM-1~~ Create a database for the exhibitions DONE HL
- ~~NOTLMIM-20~~ Develop user login function DONE VZ
- ~~NOTLMIM-16~~ Import data to the database DONE HL
- ~~NOTLMIM-17~~ Contact the staff in museum and request the exhibitions data DONE YZ
- ~~NOTLMIM-3~~ Give admins access to the database DONE HL
- ~~NOTLMIM-36~~ Navigation: Find an algorithm to find the shortest path DONE J
- ~~NOTLMIM-8~~ Development of a front end that can communicate with the database DONE HL
- ~~NOTLMIM-25~~ Development of database efficiency to make sure the software is responding fast and efficiency DONE HL
- ~~NOTLMIM-18~~ Clean and categorize data DONE HL
- ~~NOTLMIM-57~~ Database and application connection DONE HL
- ~~NOTLMIM-79~~ Add POI backend (toilets, exits) DONE HL

NOTLMIM-22 User Interface/FrontEnd
- ~~NOTLMIM-2~~ Design a User Interface DONE VZ
- ~~NOTLMIM-4~~ Implementation of an AI algorithm that calculate the shortest route. DONE J
- ~~NOTLMIM-9~~ Creating a portal that the staff can modify the information of exhibitions. DONE Y
- ~~NOTLMIM-11~~ Create interactive actions for user to interactive with for example, zoom in zoom out DONE XT
- ~~NOTLMIM-13~~ Design a popup function to show the details about the exhibition when the visitor click on a specific exhibition. DONE XT
- ~~NOTLMIM-6~~ Design different layers for showing the different floors and the exhibitions shows on that floor. DONE VZ
- ~~NOTLMIM-14~~ Add museum information like operating hours, admission prices to UI DONE XT
- ~~NOTLMIM-7~~ Development of font sizes that the user can choose from to satisfy their needs. DONE XT
- ~~NOTLMIM-10~~ Design a layout of the first floor that the user can see what they want DONE XT
- ~~NOTLMIM-41~~ Integrate coordinate offset module with current coordinate system for 2nd floor DONE Y
- ~~NOTLMIM-42~~ Screen rotation for 2nd floor DONE Y
- ~~NOTLMIM-44~~ 2nd floor coordinate correctness DONE Y

~~NOTLMIM-97~~ Outside of the code DONE
- ~~NOTLMIM-98~~ Create user manual for visitors DONE HL
- ~~NOTLMIM-99~~ Create user manual for staff DONE HL

As you can see, a bug occurred with the software preventing us from changing the start and end dates of each sprint (especially with sprint 4 which started on April 5th but is only shown to be starting on April 27th on the chart). We were also unable to find a way to present all of the backlog tasks on this roadmap even after going through all of the buttons and settings available on the Jira interface. However, we were able to obtain a full list of the backlog which can be found in the Backlog folder on Github along with the full backlog grouped by individual members and individual backlogs. Most of the names in the backlog are self explanatory but for clarity, Terry Xu refers to Ziyuan Xu and yz refers to Yifeng Zhang.

As mentioned before in the sprint retrospectives, our usage of Jira was mostly retrospective (i.e., we added things after the sprint ended rather than before). Thus, you may notice that the ascending ordering of the backlog by key does not match up with the sprint order. Unfortunately, when we tried to query the backlog by sprint, it did not order itself correctly. Thus, it would make the most sense to order it by key. Adding items to the backlog retrospectively also meant that we did not fully capture all of the tasks we did during each sprint and we acknowledge that this is an improvement we have to work on for the future.

Furthermore, the epics (purple icons) are marked as incomplete because we are leaving it open for future sprints as we still have improvements to make, as discussed above. These tasks are to be added and allocated at the start of the next (hypothetical) sprint. On the backlog list, these epics are also left as unassigned because each epic/task can only be assigned to one user. However, multiple members contributed to each of these epics meaning it would not make sense to have it assigned to only one member.

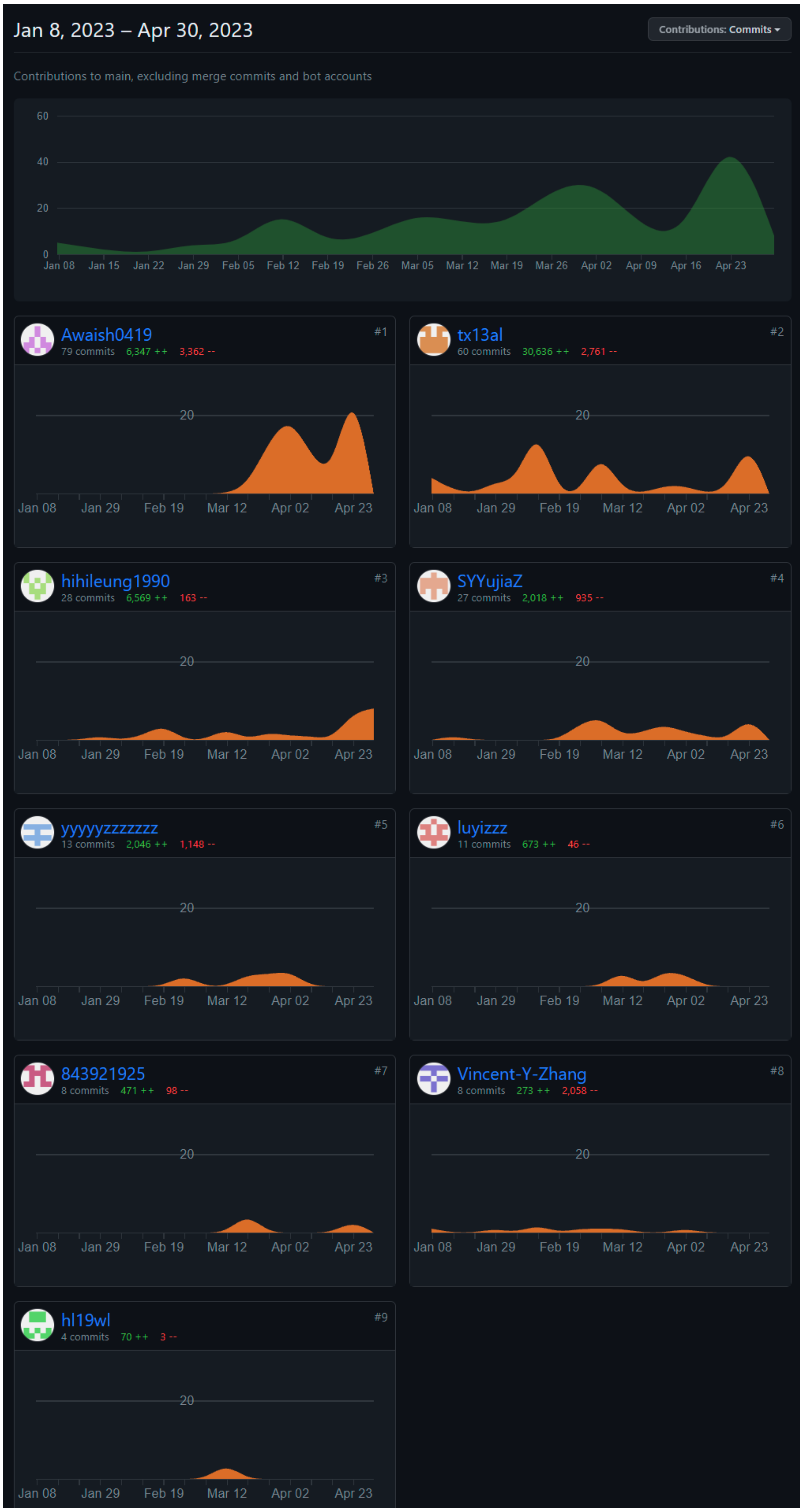
Also note that our sprint only started in February because January was spent as the brainstorming and planning stage including the creation of user stories, creating mockup UIs and coming up with the application requirements that would inform our development.

GitHub

**Names**

- Awaish0419 = Hang Li
- tx13al = Ziyuan Xu
- hihileung1990 and hl19wl = Hi Leung
  - There are commits on the second account because an unexpected error occurred on the GitHub desktop app causing commits to be made from that account rather than the main account.
- SYYujiaZ = Yujia Zuo
- yyyyyzzzzzzz = Yifeng Zhang
- luyizzz = Louis Wang
- 843921925 = Jiahao Peng
- Vincent-Y-Zhang = Vincent Zhang

A screenshot of the contribution, by commits, can be seen below. For the latest version, please refer to the GitHub Insights tab.



Jan 8, 2023 – Apr 30, 2023

Contributions to main, excluding merge commits and bot accounts

Awaish0419 #1
79 commits 6,347 ++ 3,362 --

tx13al #2
60 commits 30,636 ++ 2,761 --

hihileung1990 #3
28 commits 6,569 ++ 163 --

SYYujiaZ #4
27 commits 2,018 ++ 935 --

yyyyyzzzzzzz #5
13 commits 2,046 ++ 1,148 --

luyizzz #6
11 commits 673 ++ 46 --

843921925 #7
8 commits 471 ++ 98 --

Vincent-Y-Zhang #8
8 commits 273 ++ 2,058 --

hl19wl #9
4 commits 70 ++ 3 --

As you can see from the overall contribution graph, development was quite sprint-like. There is a skew towards the end, but this was due to us trying to understand how to use Android Studio at the earlier sprints. As seen on Hang Li's graph, the majority of the feature development began around late March after we were able to integrate the database with Android Studio. In the following paragraphs, we will recap briefly what each sprint contained in order to explain the allocation of tasks.

The first sprint/spike in the chart in early February represents our starting period. In this sprint, Ziyuan Xu created a web crawler in Python to extract information from the museum's database. She then cleaned the data and passed it to Hi Leung who did a second clean and reformatting of the data before importing that into our database. Hi Leung used AWS RDS (Amazon Web Services' Relational Database Service) to host a Postgres database so that all members would be able to access information from the database at any time. Ziyuan also was in charge of setting up the Android studio environment during this period as she was the repository owner. During this split, we were still having discussions about whether to commit to a website, Android app or both. Thus, we did not do much programming beyond the web crawler.

At the end of the first split, we decided that we were going to commit to creating an Android app. In the second split, we focused on learning how to use Android Studio, creating early versions of the UI, and trying to get the database connected to Android Studio. Whereas some members were learning how to use Android studio through making changes on our repository (Ziyuan Xu, Yujia Zuo, and Vincent Zhang), others decided to learn on their own using their own local files (Hang Li, Yifeng Zhang, Louis Wang, Jiahao Pang). This explains why there was an imbalance of commits around this period. Furthermore, Hi Leung was trying to create a Rest API using Lambda functions and other AWS services to try and create a secure connection between the database and the application. Thus, he did not make any commits to GitHub as everything was done remotely on AWS. However, this did not work out in the end and in the third sprint, we decided to use JDBC to connect to the database instead.

Moving onto the third sprint, we prioritised getting the database connected and building the rest of the functions that would work without the database connection. Starting with the database connection, Hi Leung and Hang Li focused the majority of their effort on making this connection. Whilst Hi was in charge of making the physical connection through JDBC and making sure the database connection credentials were securely stored within the app, Hang directed his attention towards making sure the database integration was efficient. After Hang finished that, he joined the rest of the dev team and began programming, whereas Hi had to learn Android Studio as he spent the majority of the earlier sprints using AWS, Postgres, and writing documents. He achieved this by using threads to maintain temporary connections allowing for more efficient resource management and better responsiveness. On the other hand, the rest of the members were focused on getting the front-end of the application functioning. Not only does this include multiple different interfaces for each "activity" (function), they also re-made the map so that the map was drawn using lines allowing for boundaries to be set. This laid the foundation for future features like navigation and the placing of showcases (exhibits) on the map.

This brings us to the final sprint where we made significant strides towards reaching the requirements we created at the end of January. Although we were behind from the previous sprints, the velocity of this sprint was a lot higher than expected. This was mostly thanks to Hang's overall understanding of Java and Android development and Ziyuan's understanding of the front-end side of Android development. In this sprint, we can see that there is a significant imbalance in terms of commits. This is because our team decided to leverage Hang's understanding and utilised pair programming. Whereas the other members already had the logic of their functions planned out, it was with the help of Hang that they were able to implement it successfully. Namely, these pairs were:

- Hang + Jiahao: Navigation
- Hang + Yifeng and Hang + Louis: Staff features (adding, editing, and removing exhibits, etc.)
- Hang + Yujia: Search bar and searching features in general
- Hang + several members: UI fixes

In addition to this, we decided that offline mode, the inclusion of user manuals within the app, and the availability of important points of interests (POIs) such as toilets, exits, and staff counters were crucial features to add. This was a relatively last minute addition that Hi, Hang, and Ziyuan worked on. However, due to the limited time in the presentation, the offline functionality was not shown in the Teams call. Details on this can be found in the user manual. For more details about who did what in this last sprint, please refer to section 7.