

# Sprint 1:

## Database:

- 1) Create database
  1. User stories: As a user, I want all the data to be permanently stored.
  2. Features (functions): created a database to store all the data.
  3. Enhancement: find a way to make the connection safely.
  4. Bug fix: N/A
  5. Prioritization: sprint 1
  6. Estimation: 1-2 weeks to find a good database server and build a service on it.  
The time consumed in the application should be less than 2 sec.
- 2) Use crawler to collect museum item data
  1. User stories: As a user, I want all the data in the database to be real and accurate.
  2. Features (functions): create a crawler to grab all items in the database to export to an excel file
  3. Enhancement:
    - a. before coding the crawler, analyzing the html information of the database website.
    - b. Clean up the data and remove the duplicate items, and categorize the items.
  4. Bug fix: N/A
  5. Prioritization: sprint 1
  6. Estimation: 2 days for coding the crawler and 3 days to clean the data.

## Customer interface:

- 1) Display the map in the main screen
  1. User stories: As a customer, I want to see the interface when the app starts.
  2. Features (functions): layout design for the main activity interface.
  3. Enhancement: Some buttons and search bar layout would be displayed on the main interface.
  4. Bug fix: N/A
  5. Prioritization: sprint 1
  6. Estimation: 1 week. The time consumed to display the interface should be in 1 sec.
- 2) Display the map in the main screen
  1. User stories: As a customer, I want to see the map when the app starts.
  2. Features (functions): draw the map using coordinates.

3. Enhancement: Initiate two floors of maps when the app starts, then use buttons to decide which floor to be visible and invisible (increase processing efficiency) while keeping perspective of the invisible floor. (The button of the selected floor would be displayed as red.)
4. Bug fix: N/A
5. Prioritization: sprint 1
6. Estimation: 2 weeks. The time consumed to display the map should be in 1 sec.

# Sprint 2:

## Database:

- 1) Create some tables in the database for different users
  1. User stories:
    - a. As an employee, I want to login to the system by the correct password with my account name.
    - b. As a user, I don't want to save all the data in my phone.
  2. Features (functions): have a table designed to save all the employee's account number and password information.
  3. Enhancement:
  4. Bug fix:
  5. Prioritization: sprint 2
  6. Estimation: 1 hour to simply create a table in the database.
- 2) Load all the data to the database:
  1. User stories: As a user, I want all the data displayed in the app to be real and accurate from the database.
  2. Features (functions): Save all the data fetched from the crawler in our new database.
  3. Enhancement:
  4. Bug fix:
  5. Prioritization: sprint 2
  6. Estimation: 1 day to add the fetched data to the database.
- 3) Create connection with database.
  1. User stories: As a user, I want to interact with the database. (As not all data can be saved on the device.)
  2. Features (functions): Save data to the database by embedded SQL code written in android studio.
  3. Enhancement: We need to find a safe way to save our password in our file.
  4. Bug fix: They always told us the connection failed, but we didn't know what was wrong until we double checked the password. (password incorrect)
  5. Prioritization: sprint 2 (HIGH HIGH)
  6. Estimation: 1-2 weeks to find a way to find an effective way to make the connection.

## Customer interface:

- 1) The map can be moved, zoomed in and out.
  1. User stories: As a customer, I want to see the map more flexible.

2. Features (functions): move the map with the finger, zoom in / out with fingers.
  3. Enhancement:
  4. Bug fix: N/A
  5. Prioritization: sprint 2
  6. Estimation: 1-2 weeks. The moving job should be fluent and no lagging can be detected by human eyes.
- 2) The associated search bar is built and tested with some given Strings.
1. User stories: As a customer, I want to search for not specific things. (error tolerate)
  2. Features (functions): the associated choices would be displayed under the search bar.
  3. Enhancement:
  4. Bug fix: N/A
  5. Prioritization: sprint 2
  6. Estimation: 1 week to fix what kind of component can implement this function.
- 3) Find the shortest path, with the all the distance information as input
1. User stories: As a customer, I want to know the shortest path to the destination.
  2. Features (functions): output a the shortest path list for drawing the shortest path
  3. Enhancement: Using dijkstra base algorithm to find the path, due to the map size is not big.
  4. Bug fix: N/A
  5. Prioritization: sprint 2
  6. Estimation: 1 week

## **Employee interface:**

- 1) The first floor and the second floor should have the same unit number.
  1. User stories: As an employee, I want to use the same unit to add showcases for both floors. (meters or inches).
  2. Features (functions): unify the unit in different floor
  3. Enhancement:
  4. Bug fix: N/A
  5. Prioritization: sprint 2
  6. Estimation: 1-2 days
- 2) Staff can login by clicking the right top button.
  1. User stories: As an employee, I want to log in with a dialog.
  2. Features (functions): The login system can check if the password matches the saved data in our database.
  3. Enhancement: Pop some notification if the login is failed or successful.
  4. Bug fix: N/A
  5. Prioritization: sprint 2
  6. Estimation: 1 week for the layout design and the connection with the database.

### 3) Employee can view the map

1. User stories: As an employee, I want to see the changes I've made and what the visitor will see in the employee interface.
2. Features (functions): employees will see what the visitor will see and all the uploads and changes will be shown on the map.
3. Enhancement: The changes will be react and show on the visitor side after the valid input. Some ideal functions are setting as buttons on the interface.
4. Bug fix:
5. Prioritization: sprint 2
6. Estimation: 3-4 days

# Sprint 3:

## Database:

- 1) The device should connect to the database.
  1. User stories: As a customer, I want to fetch the data from the database to show the exhibits in my device.
  2. Features (functions): connection successfully!
  3. Enhancement:
  4. Bug fix:
    - a. The connection always fails with the error message, internet used in the main thread. Hi added a policy to allow internet access in the main thread.
    - b. For efficiency, Hang moved the connection to another thread to make the program with better performance.
  5. Prioritization: sprint 2 (EMERGENCY)
  6. Estimation: 1 - 2 weeks.
- 2) Screen can rotate when the phone is changed from vertical to horizontal.
  1. User stories: As a customer, I want to view the map horizontally when I move my phone to hori.
  2. Features (functions): add a layout for horizontal screen design.
  3. Enhancement:
  4. Bug fix: N/A
  5. Prioritization: sprint 3
  6. Estimation: 1 day. At most 2 seconds to reflect.
- 3) Map pins can be added to the map dynamically and move with the map (moving or scale)
  1. User stories: As an employee, I want to see the map pin indicate the showCases and can move with the map.
  2. Features (functions): map pin indicating the showCases can be created dynamically and be moved with the map.
  3. Enhancement: The edges of the showcases are surrounding the showCase button.
  4. Bug fix:
    - a. The pin cannot be created dynamically. Terry tried many methods to do that, but not working. Yifeng finally found a way to solve this problem. After that, Hang implemented it on the map.
    - b. The pin can move with the map, but is buggy when the map is scaling.
  5. Prioritization: sprint 3
  6. Estimation: 2 weeks. No lagging can be detected by the human eyes.
- 4) Customer can search from the real items using search bar:

1. User stories: As a customer, I want to search items by the real names from the database.
  2. Features (functions): search bar can use the names of all items from the database.
  3. Enhancement:
  4. Bug fix:
  5. Prioritization: sprint 3
  6. Estimation: 2-3 days.
- 5) The map pin can display a list of items by clicking.
1. User stories: As a customer, I want to see a list of items in a showCase.
  2. Features (functions): The items in a showCase can be displayed as a scroll view as there might be many exhibits in one showCase. And for each item, it should display an item and the name of that item.
  3. Enhancement: Set the item picture and the name clickable for future usage.
  4. Bug fix:
    - a. The displaying items would not disappear, but would stack again and again.
    - b. Item name cannot be viewed at the center horizontal.
    - c. The item list cannot be turned off even by clicking some other buttons on the map.
    - d. The image is out of scale.
  5. Prioritization: sprint 3
  6. Estimation: 1-2 weeks
- 6) Customer can see the information about the museum
1. User stories: As a customer, I want to know about the information about the museum like location, ticket price, open hours and so on.
  2. Features (functions): In InfoPage, customers can review the basic information about the museum.
  3. Enhancement: users can click on the address of the museum to get route information from their map app on their phone. Users can call the museum by clicking on the phone number and email to the museum by clicking on the email address on the page.
  4. Bug fix:
  5. Prioritization: sprint 3
  6. Estimation: 3 days to design the UI and implementation.
- 7) Display the item details by clicking the item from the displayed item list
1. User stories: As a customer, I want to see the detail of the item in a showCase.
  2. Features (functions): Use a popup dialog to show the item detail page, including all the item information and url to the webpage.
  3. Enhancement:
  4. Bug fix:

5. Prioritization: sprint 3
6. Estimation: 1 week

## **Employee Interface:**

- 1) The floors are displayed with the fixed coordinates.
  1. User stories: As an employee, I want to add the showCases with its size and coordinates set with inches.
  2. Features (functions): Recalculate the first and the second floor with inches and repaint the map with the new calculated edges.
  3. Enhancement: Separate the edges to inner edges and outer edges. (As inner edges are for navigation and the conflict detection for adding the showCase.
  4. Bug fix: visually see the incorrect edges.
  5. Prioritization: sprint 3
  6. Estimation: 2-3 days
- 2) Employees can add a new showcase (and map pin) to the map and database.
  1. User stories: As an employee, I want to add a new showcase to the map.
  2. Features (functions): input detail value to create a new showCase, and add the showCase information to the database, and add the showCase to the map displaying now.
  3. Enhancement: New showcase can't add outside the map or overlap the existing showcase.
  4. Bug fix:
    - a. After creating a showCase, the map is not updated. (new showCase button and the edges would not display on the map.)
    - b. The button can be viewed but unclickable.
    - c. The showcase might overlap other showcases.
    - d. The showcase cannot be created if any edge is stuck to the wall. sharing with the wall or other showcases.
    - e. The new map pin is not movable with the map.
  5. Prioritization: sprint 3
  6. Estimation: 3 days. (Exceed the time slot a lot as there are so many bugs.)
- 3) Employees can log out from the login interface.
  1. User stories: As an employee, I want to log out after finishing the modification.
  2. Features (functions): get back to the mainActivity.
  3. Enhancement: The data sharing with the mainActivity to avoid the re-request for the database. (Can use a singleton to implement this function)
  4. Bug fix: N/A
  5. Prioritization: sprint 3
  6. Estimation: 1-2 days



# Sprint 4:

## Customer Interface:

- 1) Making the buttons and UI more beautiful.
  1. User stories: As a customer, I want to see a better look UI.
  2. Features (functions): Change some layout and the ratio of the distances between components.
  3. Enhancement:
  4. Bug fix: N/A
  5. Prioritization: sprint 4
  6. Estimation: 2-3 days.
- 2) Navigation:
  1. User stories: As a customer, I want to see a line with different color from other edges between the locations I selected. Also, the line must be the shortest.
  2. Features (functions): Using a dijkstra base algorithm to find the shortest path due to the map is not big. Need to find a method to define the actual distance without crossing the obstacles. Need to find a method to be able to navigate between floors. Need to present the path on the screen. Add a button in the item list scroll view, so the users can click that for navigation.
  3. Enhancement: Abstract the edges on the map for efficient input. Change the pin of the start point and the end point. (start point to a green pin, end point to a flag) The navigation icon in the item list will change if it is selected as a start/end point.
  4. Bug fix:
    - a. when the start point and end point are the same point, the algorithm would crash. Fixed by avoiding calling the algorithm.
    - b. The navigation icon in the showcase item list cannot change with the start/end point selection made.
    - c. There are some tiny paths between showcases which people cannot walk through. (To be fixed in next sprint)
  5. Prioritization: sprint 4
  6. Estimation: 1 week.
- 3) Customer can change the font size
  1. User stories: As a customer, I want to be able to change the font size to read the details about the items
  2. Features (functions): In MorePage, customers can adjust font size by clicking on different sizes of buttons to change the font size for the app.
  3. Enhancement: three buttons that provide small, regular, and large sizes of font that the customers can choose from.
  4. Bug fix: Buttons overlap the screen, so change the layout to LinearLayout and add the weightsum to allocate to each button 1 weight.
  5. Prioritization: sprint 4

6. Estimation: 1 day to design the UI and implement the actual function. 1 day to make the button fix different screen sizes and does not overlap the screen.
- 4) Customers can review all items
1. User stories: As a customer, I want to see all the items in the museum
  2. Features (functions): In ArtPage, customer can browser all the items
  3. Enhancement: use a RecyclerView to show all the items and users can scroll the screen to see more items. Each item can only show 3 lines of description. But if the description is more than 3 lines, there is a more button to show more descriptions to the users.
  4. Bug fix: the items list from the database keeps throwing null pointer exceptions. To fix the bug I have to communicate with Heng to get the items from the database.
  5. Prioritization: sprint 4
  6. Estimation: 3 days to design the UI layouts and implement the actual function. 1 day to get items from the database and improve the item lists that show in the page.
- 5) A manual add in the info page:
1. User story: As a customer, I want to see how to use the application by a manual.
  2. Features (functions): Add a webpage at the bottom of the info page.
  3. Enhancement:
  4. Bug fix:
  5. Prioritization: sprint 4
  6. Estimation: 2-3 days to edit the manual.
- 6) Search bar can be used to display the associated items if exists:
1. User story: As a customer, I want to see a list of items after inserting some key words of the item I am looking for. After selecting the item I want, I want to see the details of that item and the item list of that item if it is in a showcase.
  2. Features (functions): Add an activity with a scroll view to view all the items. (selectable as well)
  3. Enhancement:
    - a. Add a slider for the search bar activity to get abc to the caller activity in case no item I want from the list.
    - b. Implement the enter input function for the search bar as well.
  4. Bug fix: The user interface would not change to the floor containing that item. (By calling a function with an if condition to solve this problem)
  5. Prioritization: sprint 4
  6. Estimation: 2-3 days to edit the manual.

### **Employee interface:**

- 1) Employee can delete existing showcase(map pin)

1. User stories: As an employee, I want to delete an existing showcase from the map.
  2. Features (functions): showcase can be deleted after selected. And the database would delete the showcase and the item links to that showcase.
  3. Enhancement: notification will pop when no showcase is selected. The showcase's item list displaying would be clear after removing. The edges and the map pin of that showcase would be removed directly.
  4. Bug fix:
    - a. After deleting the showCase, the map can still see the showCase on the map.
    - b. After deleting the showCase, the customer's interface can still see that showcase.
  5. Prioritization: sprint 4
  6. Estimation: 3-4 days
- 2) Employee can edit the coordinate and size of the existing showcase(map pin)
1. User stories: As an employee, I want to move an existing showcase to another location. (Maybe with a different size of the showcase.)
  2. Features (functions): popup a dialog to request the data that needs for the location of the new showcase.
  3. Enhancement:
    - a. the coordinates cannot be outside the map or overlap another existing showcase.
    - b. A notification will popup if no showcase has been selected.
    - c. The popup dialog would fill the contents with the selected showcase's information.
  4. Bug fix:
    - a. The old showCase is not removed from the previous position. (Duplicate showCases)
    - b. The edges of the showCase were kept on the map.
    - c. One of the edges of another showCase was removed. (Rewrite the function for getting the edges for the showcase selected.)
    - d. Moving a showCase from one floor to another left the edges of the showCase at the previous location. (Fixed by changing an if to else if)
  5. Prioritization: sprint 4
  6. Estimation: 3-4 days.
- 3) Employee can add new exhibits to existing showcase
1. User stories: As an employee, I want to add new exhibits to a showcase.
  2. Features (functions): exhibit can be added into the showcase item list if the exhibit exists inside the database. If the item is in another showcase, it would be removed from that showcase. After adding, the displaying item list can see the item added visually. A button to access the adding function. Also, we need to implement a search dialog for users to add an item to the showCase.

3. Enhancement:
  4. Bug fix:
    - a. The layout is out of the boundary of the dialog.
    - b. The picture layout is overlapping with the text layout, so the images cannot be viewed clearly.
  5. Prioritization: sprint 4
  6. Estimation: 3-4 days
- 4) Employees can delete items from showcase
1. User stories: As an employee, I want to remove items from a showcase.
  2. Features (functions): item will be removed from the item list of the selected showcase. Item link with that showCase would be deleted in the database, and the displaying item list for the showcase can see that the item was removed.
  3. Enhancement:item list delete option layout, making it more bigger for clicking
  4. Bug fix: N/A
  5. Prioritization: sprint 4
  6. Estimation: 2-3 days