Nanyang Technological University

**Lab 2 Deliverables:**

**CZ2006 Software Engineering**

Name:

1. Shen GuangXu U1820675J
2. Elaine Pak U1821405K
3. Choong Han Yi U1820140J
4. Guo WanYao U1822530E
5. Zhang YongZhe U1921423J
6. Goh Jia Rui, Bryna U1822465A

Lab Group: SS9

Team Name: AAAAAA

**Table of Contents**

[1. Mission Statement 3](#_Toc33360722)

[2. Functional Requirements 3](#_Toc33360723)

[3. Non-functional Requirements 4](#_Toc33360724)

[3. Use Case Model 5](#_Toc33360725)

[4. Use Case Descriptions 5](#_Toc33360726)

[4. Entity Class Diagram 25](#_Toc33360727)

[5. Key Classes (Boundary and Control) 25](#_Toc33360728)

[6. Sequence Diagrams 25](#_Toc33360729)

[7. Dialog Map 25](#_Toc33360730)

[5. Data Dictionary 25](#_Toc33360731)

[6. UI Mock-up 27](#_Toc33360732)

# Mission Statement

The AAAAAA team will create a mobile application making Singapore’s parks and greenery convenient to explore. The team considers this project complete when all stated requirements have been satisfied. This project supports the NParks objective to promote green recreation as a lifestyle.

# Functional Requirements

1. The application must run on the Android Operating System
2. The application must be able to display the following information within 5km of the user’s position to the user in a map
   1. Parks
   2. Park connector loops
   3. Park facilities
      1. Tracks
      2. Car Parks
      3. Buildings
      4. BBQ Pits
      5. Activity Areas
   4. Heritage trees
   5. Tree conservation area
   6. Skyrise greenery
   7. Heritage Roads
   8. Community Gardens
   9. PSI
   10. Dengue Infection
   11. Weather
   12. The application must allow the user to filter out information types
3. The application must allow the user to instead select a location on the map
   1. The application must display the above information within 5km of the location
4. The application must allow the user to instead display all the above information in a list format
5. The application must allow users to receive notifications based on their current location
   1. The notifications must be on nearby Objects of Interest
   2. The notifications must be on nearby information
6. The application must allow users to log in
7. The application must allow users to view the profile for each Object of Interest on the Map
   1. The application must allow logged-in users to upload photos of Objects of Interest
   2. The application must allow logged-in users to remove their own photos
   3. The application must allow users to view photos the Object of Interest
   4. The application must allow logged-in users to ‘like’ photos
8. The application must allow logged-in administrators to amend Objects of Interest
   1. The application must allow the administrator to add and remove Objects of Interest
   2. The application must allow the administrator to remove photos
9. The application must recommend trails based on inputs as follows
   1. Type of activity (hiking/cycling/running/skating/jogging)
   2. Distance Covered

# Non-functional Requirements

Usability:

1. The application must display help messages in the local language according to the user’s locale (English, Chinese)
2. Every page must have a help button that explains the functions of every button on the page
3. 80% of users must be able to view facilities within 5km in 10 seconds of opening the application’s home page
4. All pages must have a settings button

Reliability:

1. The application must not crash for up to 2 hours of continuous usage
2. The application should not drain more than 10% of battery after 2 hours of continuous usage

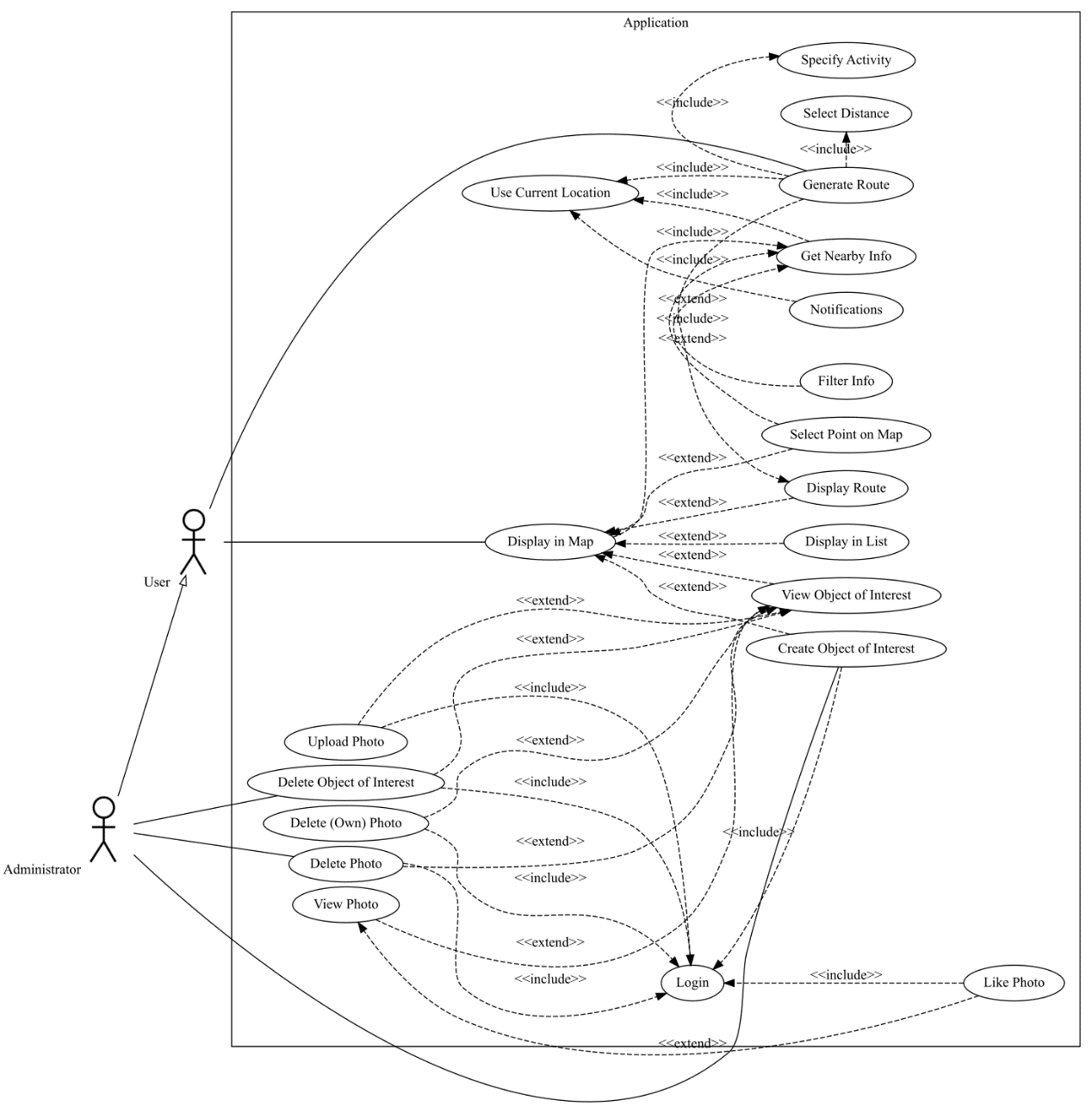
Performance:

1. The application should respond within 0.5s after the user taps on the screen
2. The application should boot up from user tapping on the app icon on the phone home screen within 5 seconds.
3. The application should show a visualized location on the map.
4. The application should return the estimated travelling time within 5 seconds.

Supportability:

1. The application must support both KML and JSON datasets
2. The application must allow the user to transfer their data across multiple devices via logging in

# Use Case Model



# Use Case Descriptions

|  |  |
| --- | --- |
| **Use Case Name:** | Use Current Location |
| **Actor:** | Application |
| **Description:** | Application gets user’s current location |
| **Preconditions:** |  |
| **Postconditions:** |  |
| **Priority:** | Low |
| **Frequency of Use:** | Very High |
| **Flow of Events:** | 1. Application uses phone’s GPS to get user’s current location 2. Application uses phone’s network to place user on map |
| **Alternative Flows:** | User has not granted app permission to use phone’s GPS   1. Dialogue appears requesting for GPS access |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Display Nearby Information |
| **Actor:** | Application |
| **Description:** | Application displays nearby information based on user’s current location |
| **Preconditions:** | Application has user’s current location |
| **Postconditions:** |  |
| **Priority:** | Low |
| **Frequency of Use:** | Very High |
| **Flow of Events** | 1. Application gets all information within 5km of a user’s position 2. Application gets all Objects of Interest within 5km of a user’s position |
| **Alternative Flows:** | User filters out information  User drags a pin onto the map   1. Application gets all information within 5km of that location 2. Application gets all Objects of Interest within 5km of a that location |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Display in Map |
| **Actor:** | Application |
| **Description:** | Application displays a map |
| **Preconditions:** | Application has nearby information available |
| **Postconditions:** |  |
| **Priority:** | Low |
| **Frequency of Use:** | Very High |
| **Flow of Events:** | 1. Application gets map of Singapore from Google Maps API 2. Application displays information as icons on the map 3. Application displays Objects of Interest as icons on the map |
| **Alternative Flows:** | User has a route selected   1. Application overlays route on top of map   User requests application to display information in a list format   1. Application switches to list view   User clicks on Object of Interest icon   1. Application shows object of interest’s profile   User drags a pin onto the map   1. Application displays all information within 5km of that location 2. Application displays all Objects of Interest within 5km of a that location |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Notifications |
| **Actor:** | Application |
| **Description:** | Application pushes notifications to user |
| **Preconditions:** | Application has user’s current location |
| **Postconditions:** |  |
| **Priority:** | Low |
| **Frequency of Use:** | High |
| **Flow of Events:** | 1. Application gets all information within 5km of a user’s position 2. Application gets all Objects of Interest within 5km of a user’s position 3. Application pushes a notification to phone if information or Object of Interest is present within 5km of a user’s position 4. Application adds the notification to the notification tab |
| **Alternative Flows:** | User has chosen not to display notifications |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Display in List |
| **Actor:** | User |
| **Description:** | User views nearby Information and Objects of Interest in a list format |
| **Preconditions:** | Information and Objects of Interest Displayed in Map |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. User selects toggle to display in list format 2. Application displays information and objects of interest previously visible on the map, sorted by category |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Filter Info |
| **Actor:** | User |
| **Description:** | User chooses which types of Information and Objects of Interest they wish to see |
| **Preconditions:** | Map has Displayed Information |
| **Postconditions:** |  |
| **Priority:** | High |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. User selects “Set Filters” in the left menu 2. User chooses which Information and OI categories that they want to see on the map/list 3. Application saves user’s filters when they press back to return the map/list view 4. Application updates Map |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Viewing OI’s profile |
| **Actor:** | User |
| **Description:** | User views the OI’s profile if he taps his finger onto the OI’s icon |
| **Preconditions:** | OI’s profile visible on Map |
| **Postconditions:** |  |
| **Priority:** |  |
| **Frequency of Use:** |  |
| **Flow of Events:** | 1. User taps the OI Icon on the Map 2. Application displays OI’s profile, which includes: Name of the OI, the Hashtag assigned, Date Created, and Location 3. Application displays the first few photos in the OI’s profile |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Upload Photos |
| **Actor:** | User |
| **Description:** | User uploads photos about the OI |
| **Preconditions:** | User entered OI’s profile  User is logged-in |
| **Postconditions:** |  |
| **Priority:** | High |
| **Frequency of Use:** | Low |
| **Flow of Events:** | 1. User presses upload button on the OI’s profile page 2. Application opens phone’s camera 3. User takes photo 4. Application checks if the photo size is less than 5MB 5. Application uploads photo |
| **Alternative Flows:** | User chooses to take photo from album instead  User chooses a photo that is larger than 5MB   1. Application blocks upload 2. User chooses to cancel or take another photo |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | View Photo |
| **Actor:** | User |
| **Description:** | User views a photo in the OI’s profile |
| **Preconditions:** | User is in OI’s profile  OI has one or more photos in its profile |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. User clicks photo in OI’s profile 2. Application expands photo to full screen |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Delete own photo (User) |
| **Actor:** | User |
| **Description:** | User removes photos which they have uploaded previously |
| **Preconditions:** | User is in OI’s profile  User has uploaded one or more photos to OI’s profile before  User is logged-in |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Low |
| **Flow of Events** | 1. Application displays a cross notation on the top right corner of photos user has uploaded 2. User taps on the cross 3. Application displays a confirmation window 4. User taps ‘yes’ choice in confirmation window 5. Application removes photo from OI’s profile |
| **Alternative Flows:** | User taps ‘no’ choice in confirmation window   1. Application returns to OI’s profile without removing photo |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Create New OI (Admin) |
| **Actor:** | Administrator |
| **Description:** | Administrator creates new Object of Interest (OI) |
| **Preconditions:** | Administrator is in Map View  Object of Interest does not exist at that specific location  Administrator is logged-in |
| **Postconditions:** |  |
| **Priority:** | High |
| **Frequency of Use:** | Low |
| **Flow of Events** | 1. Admin taps on the “Add” button 2. Admin drops a pin at specific location on the map 3. Admin selects a Hashtag and unique name for the OI 4. Application adds in date created and location 5. Admin confirms creation of OI 6. Application creates OI and profile |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Delete Object of Interest (OI) |
| **Actor:** | Administrator |
| **Description:** | Administrator removes existing Object of Interest (OI) |
| **Preconditions:** | Administrator is in Object of Interest’s Profile  Administrator is logged-in |
| **Postconditions:** |  |
| **Priority:** | High |
| **Frequency of Use:** | Low |
| **Flow of Events** | 1. Admin taps on the cross of the OI’s profile. 2. Application displays confirmation dialogue 3. Admin taps ‘yes’ choice 4. Application removes OI |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Delete Photos (Admin) |
| **Actor:** | Administrator |
| **Description:** | Administrator removes photos that belong to an existing Object of Interest (OI) |
| **Preconditions:** | Administrator is in Object of Interest’s Profile  Administrator is logged-in |
| **Postconditions:** |  |
| **Priority:** | High |
| **Frequency of Use:** | Low |
| **Flow of Events** | 1. Admin taps on the cross of the photo to delete it 2. Application displays a confirmation window 3. Admin taps ‘yes’ choice 4. Application removes photo |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Generate route |
| **Actor:** | User |
| **Description:** | Users select a path if they want a route recommendation within the app. |
| **Preconditions:** | Application has user’s current location |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. User chooses Hiking/Jogging activity 2. User clicks confirmation button 3. Application calculates routes within 500m of the distance users have specified, with the starting point being the user’s location 4. Application generates at least three recommended routes using available information on Park Tracks 5. Application displays the 3 recommended routes on the map |
| **Alternative Flows:** | User chooses Cycling/Skating activity   1. Application generates at least three recommended routes using available information on Park Connectors   User chooses no distance   1. Application generates at least three recommended routes of average distance using highest number of Objects of Interest crossed |
| **Exceptions:** |  |
| **Includes:** | Specify activity  Select distance |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Specify Activity |
| **Actor:** | User |
| **Description:** | Users choose the type of tracks that recommended by the system |
| **Preconditions:** | User selects “generate route” |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. Application displays default as “hiking/jogging” activity 2. User confirms choice |
| **Alternative Flows:** | User selects “cycling/skating” tab |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Select Distance |
| **Actor:** | User |
| **Description:** | User selects the distance of the routes they would like the application to generate |
| **Preconditions:** | User selects “generate route” |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. User chooses the distance for the application to generate routes from drop-down menu |
| **Alternative Flows:** | User chooses no distance |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

|  |  |
| --- | --- |
| **Use Case Name:** | Display Route |
| **Actor:** | Application |
| **Description:** | Application overlays generated routes on map for user to choose |
| **Preconditions:** | Users have confirmed route generation |
| **Postconditions:** |  |
| **Priority:** | Moderate |
| **Frequency of Use:** | Medium |
| **Flow of Events:** | 1. Application displays generated routes 2. User selects a specific route 3. Application permanently overlays route on Map View |
| **Alternative Flows:** |  |
| **Exceptions:** |  |
| **Includes:** |  |
| **Special Requirements:** |  |
| **Assumptions:** |  |
| **Notes and Issues:** |  |

# Entity Class Diagram

# Key Classes (Boundary and Control)

# Sequence Diagrams

# Dialog Map

# Data Dictionary

|  |  |
| --- | --- |
| **Data Dictionary** | |
| **Term** | **Definition** |
| Park | All Singapore National Parks |
| Tracks | Tracks: includes cycling tracks, footpaths, roads (kml20), step (kml46), boardwalk (77), earth tracks (78), carpark (235) - stopped at 561 |
| Park facilities | Any facilities within the park’s premises |
| Objects of Interest (OI) | Any object/facility in the park decided by the administrator to attract the interest of its users, such as heritage trees/fitness equipment/barbeque pits  The administrators must assign one hashtag to each of the OI on creation |
| Hashtag | Shows each Object of Interest’s category |
| Object of Interest’s Profile | Holds   * Hashtag assigned to the OI * Unique Name of the OI * Date Created by Admin * Location * Uploaded Photos |
| ‘Like’ Function | User who are viewing the photos about the OI will be able to ‘like’ the photo |
| User | People who are using the app |
| Guest | People using the app with restricted functionalities without needing to log in (will not be able to ‘like’/upload photos or sync their data across devices) |
| Administrator | Users with increased permissions |
| Login | Allows users to access their unique accounts by entering a username and a password |
| Logged-In | A user who has successfully completed their log-in procedure and has access to like, upload, and remove photo permissions |
| Map View | Default view of map will display:   * Parks (shaded in green) * Park Connector Loops (shaded in dark grey) * Tracks (shaded in light grey) * Carparks (shaded in beige) * Icons to represent OI   Users can customise Map View at the Filtering Page  Each information type will have their own unique icon and colour |
| Filtering Page | Users can override the default view of Map View by selecting information to see on the map  Some examples include:   * Do not display ‘Car Parks’ * Display ‘BBQ Pits’ |
| List View | Users will be able to see all information on the Map View displayed in a list format, sorted according to distance from the user. |
| Park connectors | Park connectors connect two different parks  There are 30 park connectors total in Singapore |
| Park track | Refer to the regular walking paths within a park |
| Cycling/skating activity | Cycling/skating routes only use park connector tracks |
| Hiking/jogging activity | Users can hike/jog on both tracks and park connectors |
| Distance | Users choose distance from a dropdown menu, determining how long the planned route should be  The unit of measure is kilometre |
| Recommended Routes | The system should recommend the three most optimized routes for users to choose from  The length of the recommended routes should be within 500m of the distance specified by users  For hiking/jogging activities, the starting point should be the user’s location (assuming the user is already in a park)  For cycling/skating activities, the starting point should lie on a recommended park connector |

# UI Mock-up

Link to UI Mock-up: <https://xd.adobe.com/view/5814c79d-7a0a-4e48-74d2-066ec8faf77f-2e25/>