

SC2006: Software Engineering

Lab 2 Deliverables

SCHOOL OF COMPUTER SCIENCE AND
ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

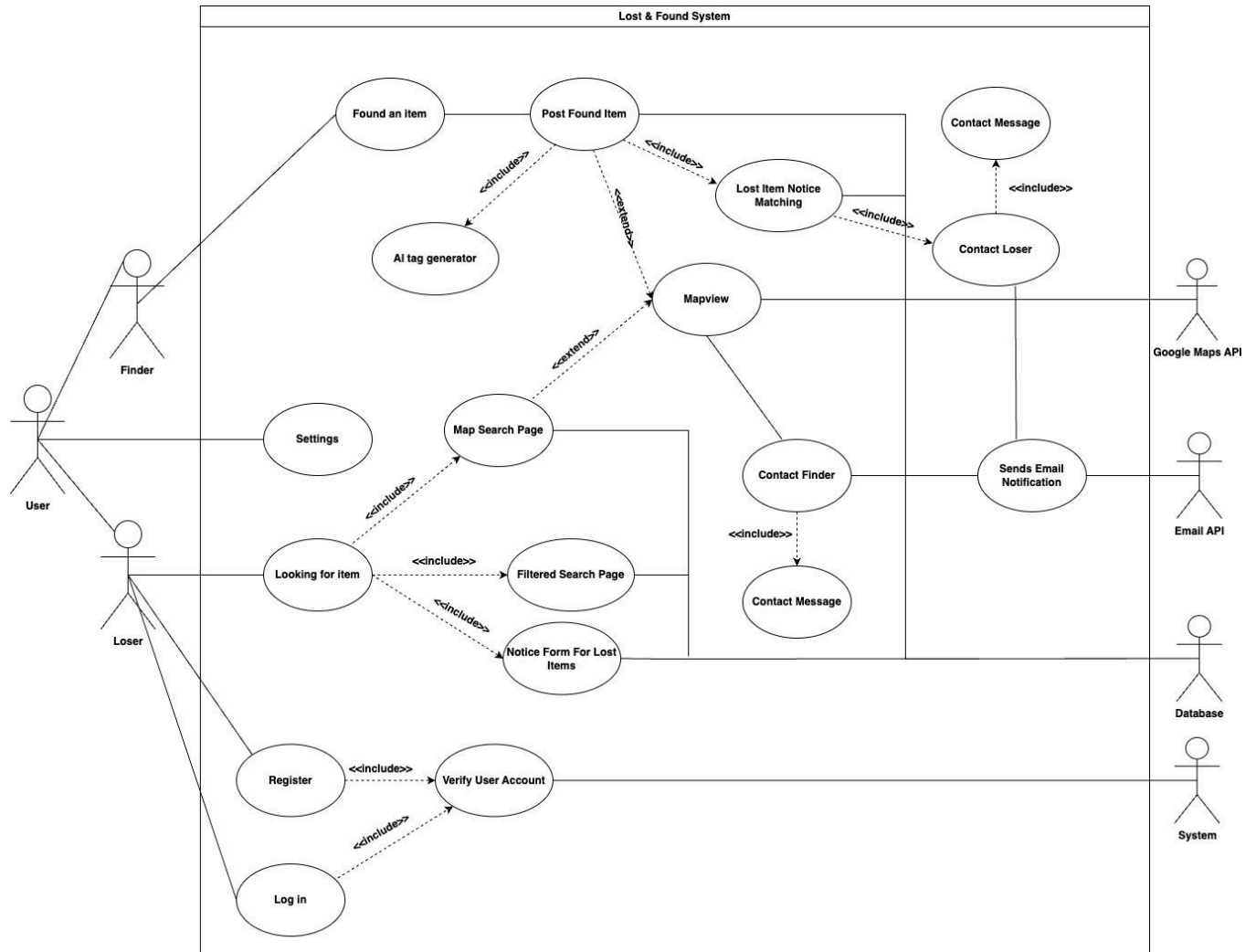
| <u>Group members</u> | <u>Matriculation Number</u> |
|----------------------|-----------------------------|
| Ong Yong Tze Marcus | U2210054L |
| Luo Mao Yuan | U2220239K |
| Poh Qi Bin | U2222890C |
| Phoebe Lee Jia Ying | U2223881K |
| Rhea Susan George | U2220116B |
| Tan Jun Yan Xavier | U2221653J |

Date of Submission: 21/2/24

Document Outline

1. Complete Use Case Diagram
2. Class Diagram of entity classes
3. Key Boundary Classes and Control Classes
4. Sequence Diagrams of some use cases
5. Initial Dialog Map
6. Use Case Descriptions

Complete Use Case Diagram



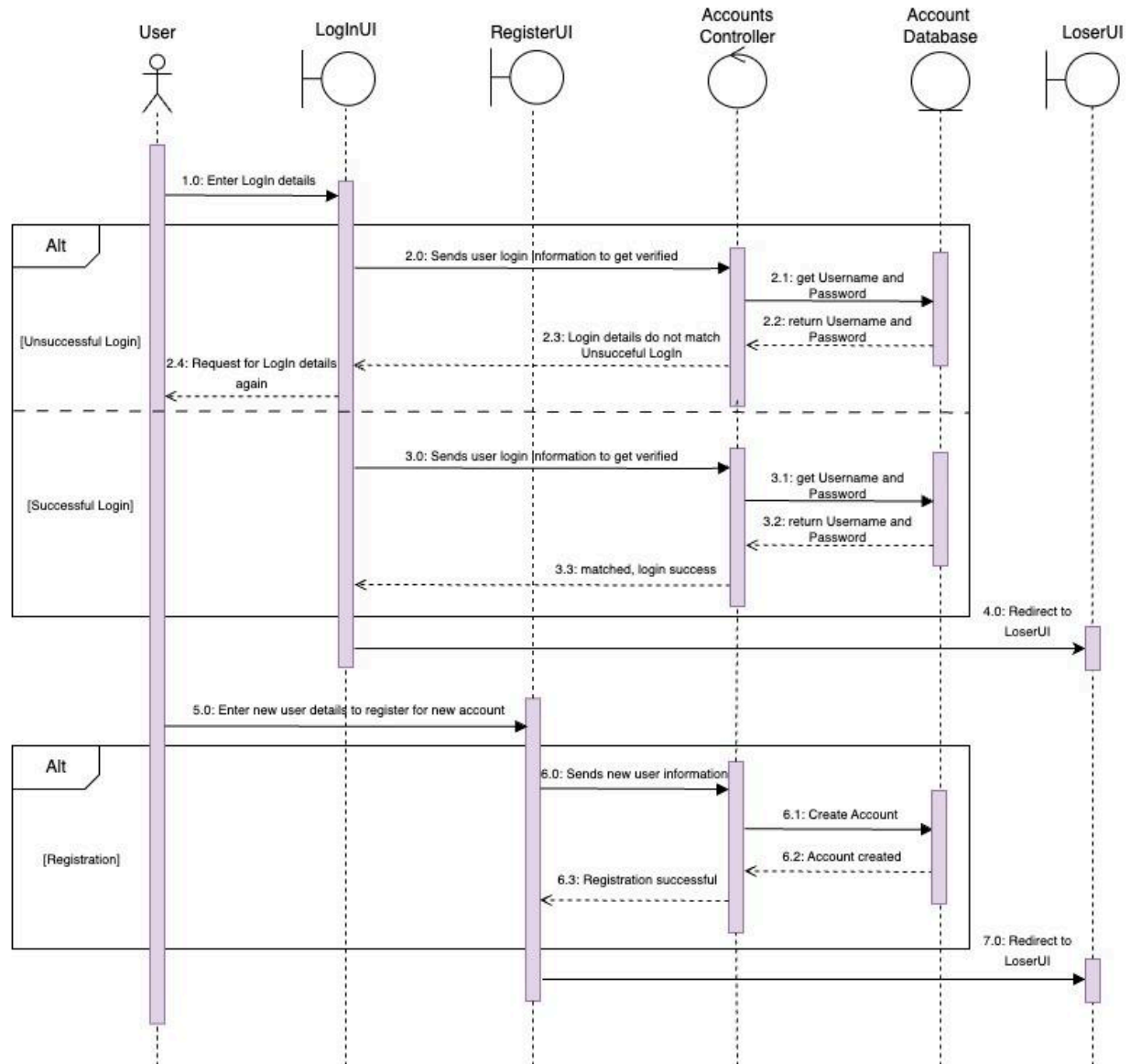
[illegible]

Key Boundary Classes and Control Classes

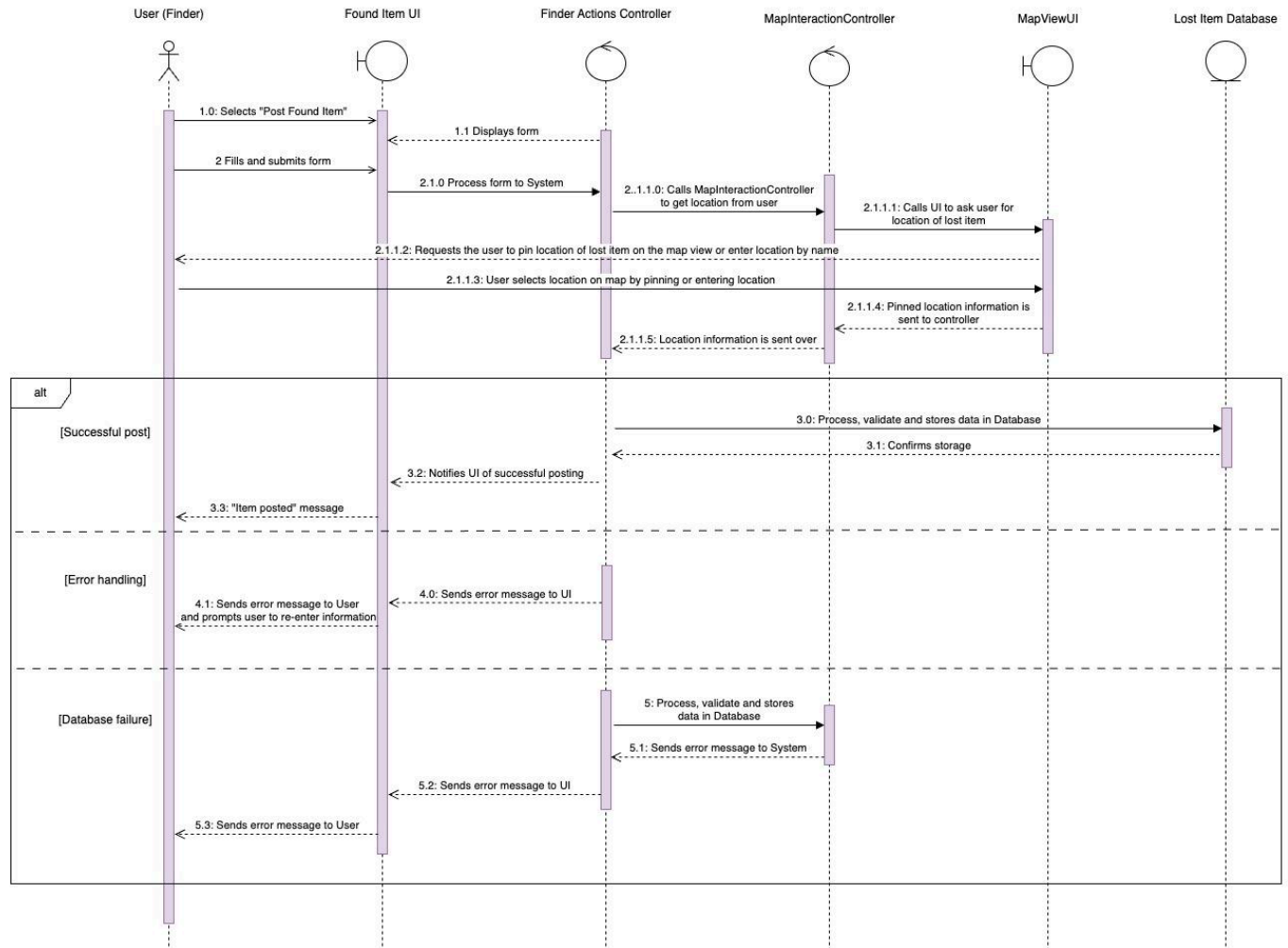


Sequence Diagrams of some use cases

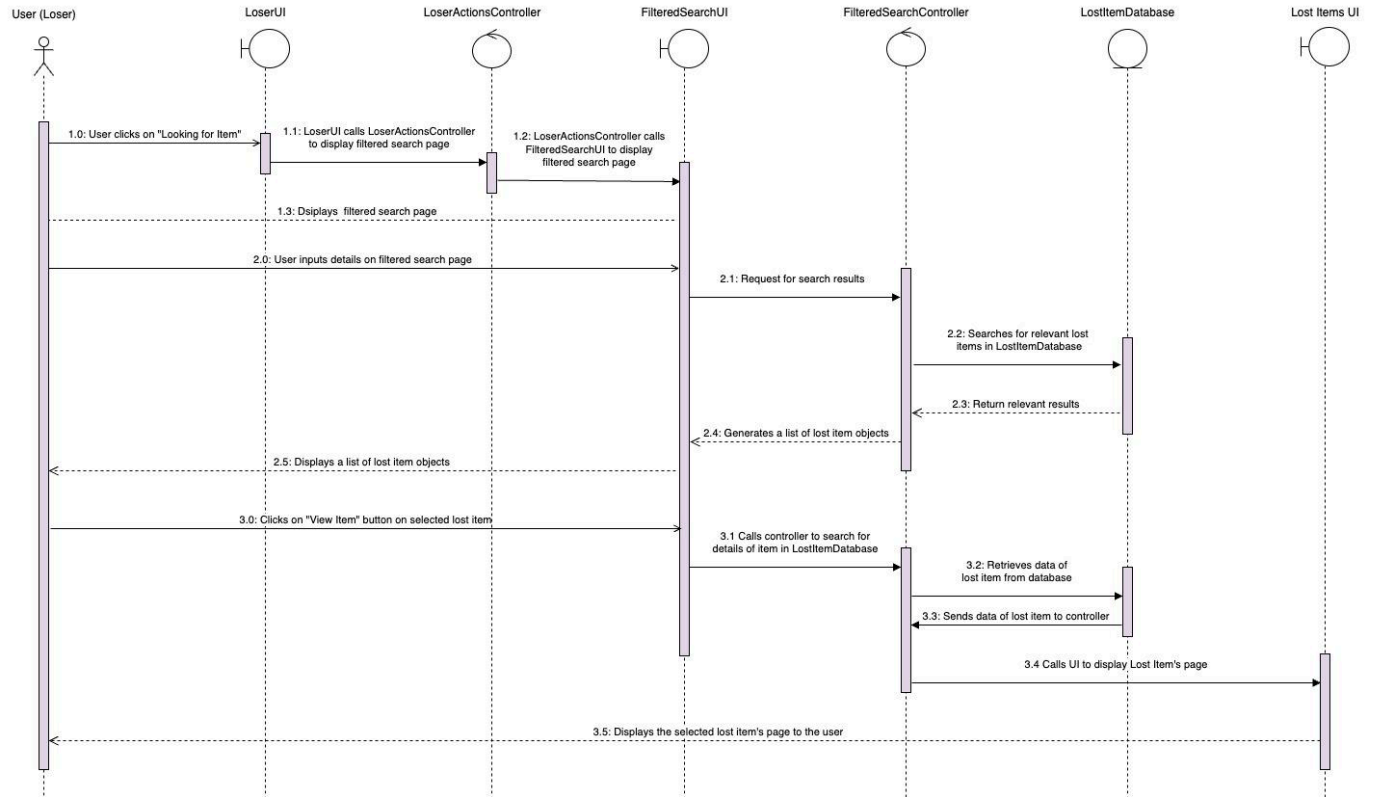
User Login



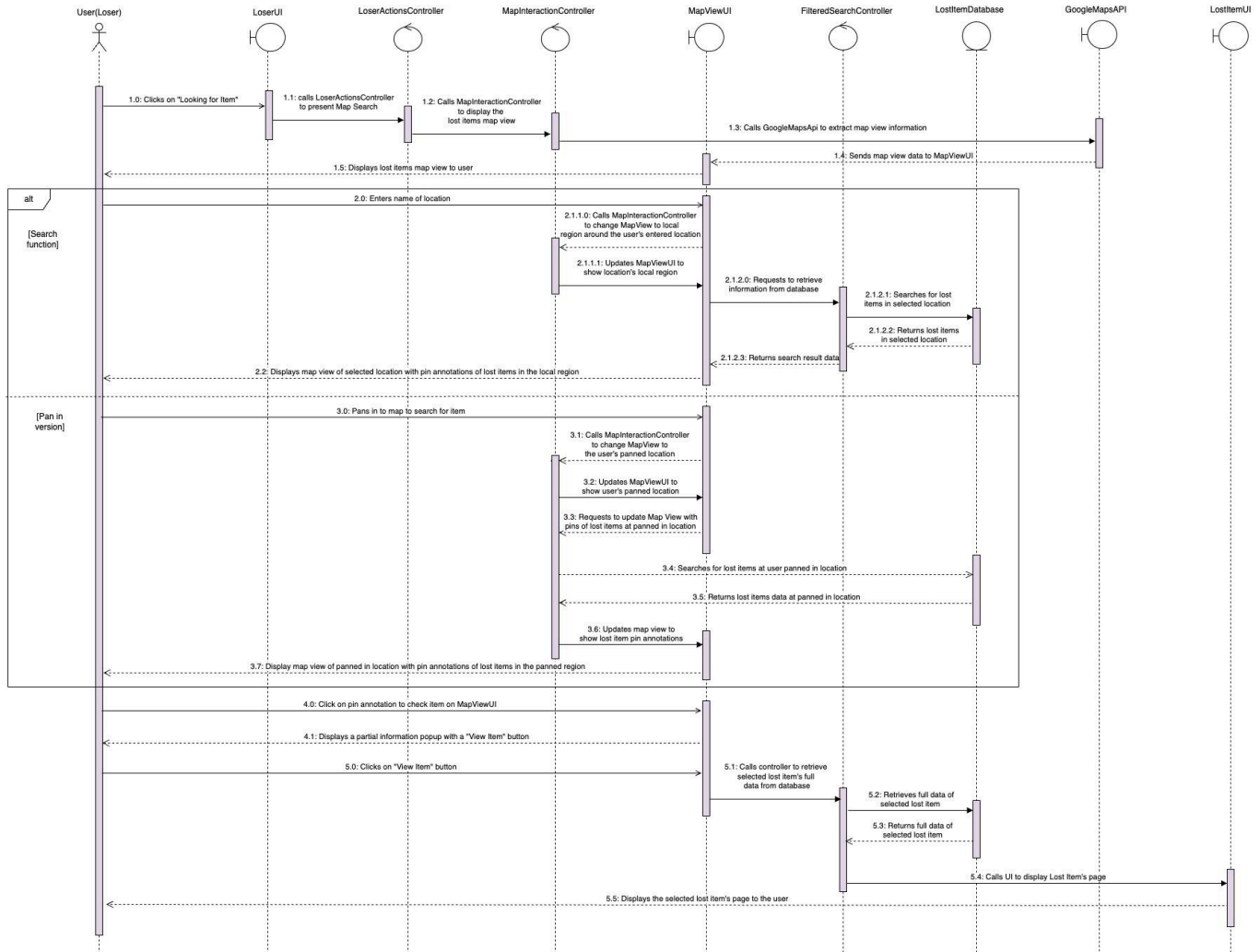
Post Found Item



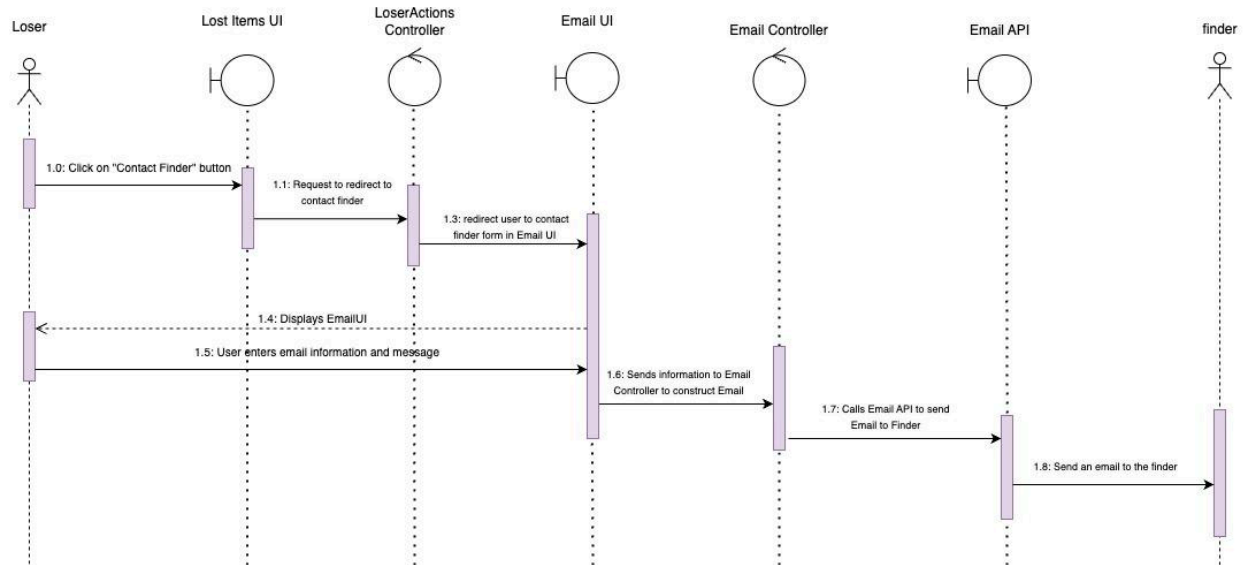
Finding for Lost Item (Filtered Search Method)



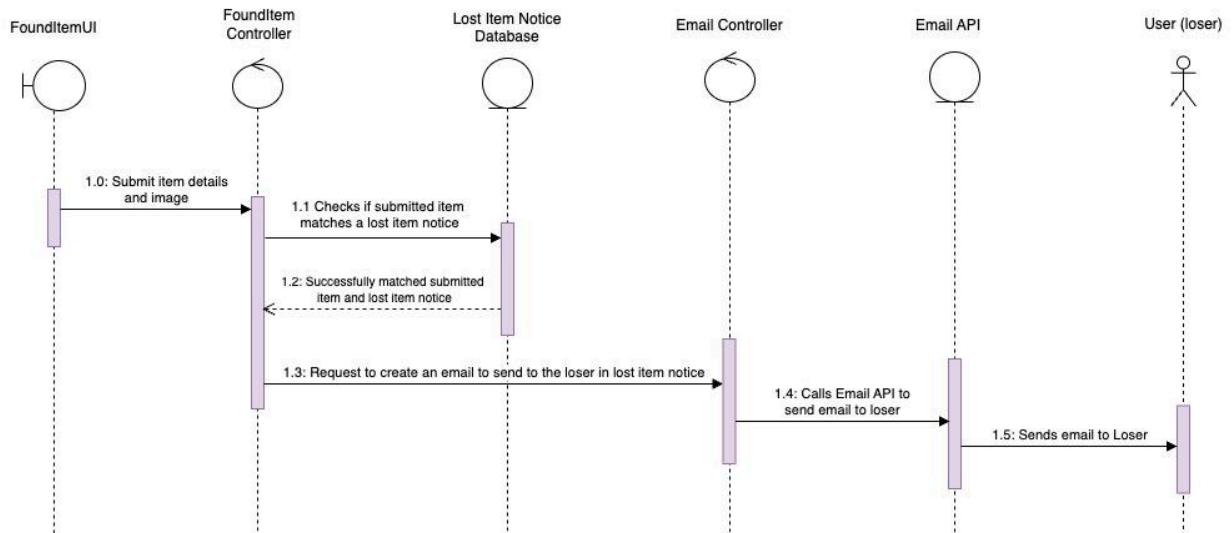
Finding for Lost Item (Map Search Method)



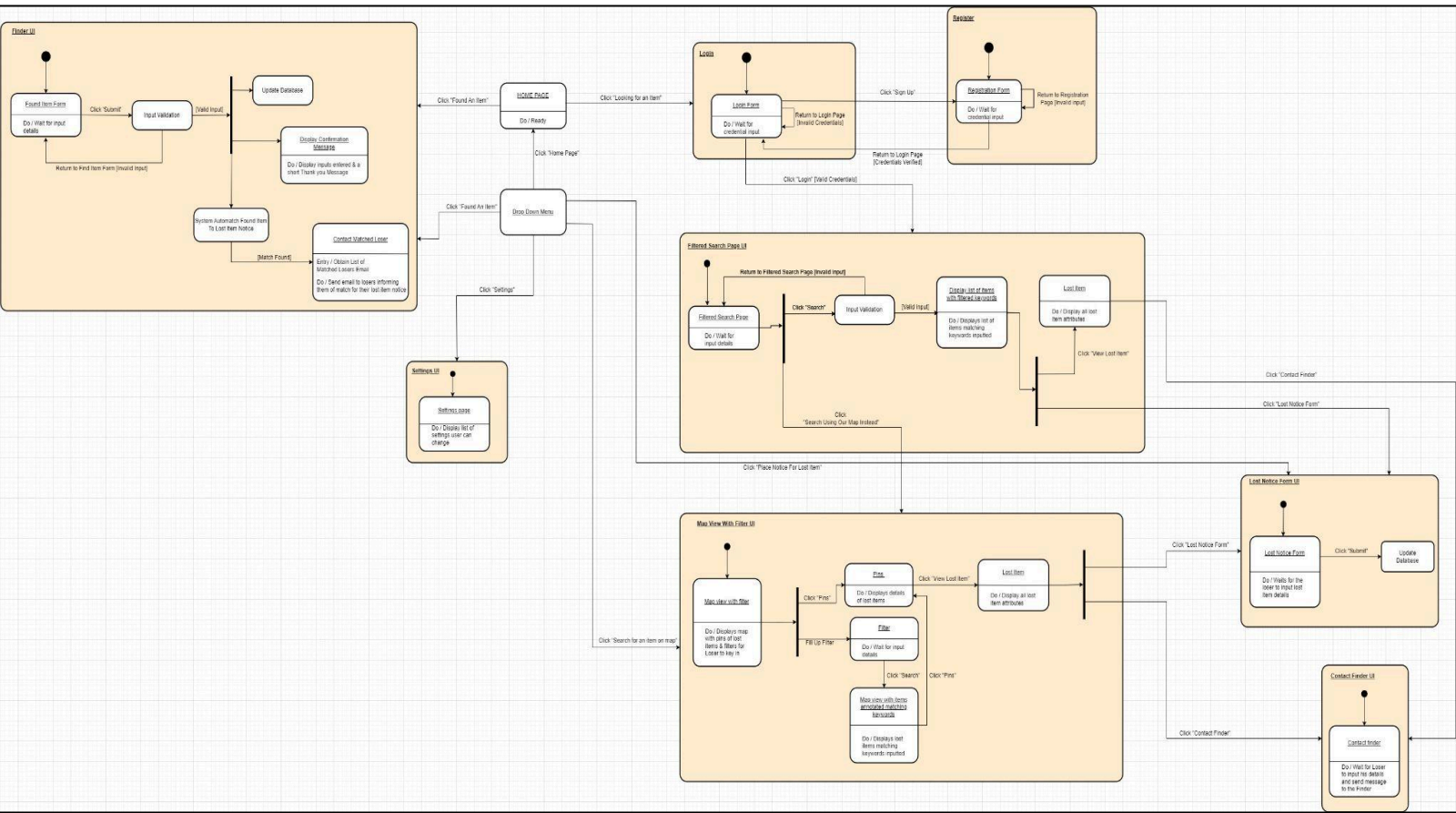
Contact Finder



Contact Loser



Initial Dialog Map



Use Case Descriptions

| | | | |
|----------------|------------|------------------|--|
| Use Case ID: | UC02 | | |
| Use Case Name: | User login | | |
| Created By: | Phoebe Lee | Last Updated By: | |

| | | | |
|---------------|--------|--------------------|--|
| Use Case ID: | UC02 | | |
| Date Created: | 7/2/24 | Date Last Updated: | |

| | |
|-------------------|--|
| Actor: | User, Loser, Finder and System |
| Description: | Loser needs to register and login to the website to post a notice or find the item they have lost |
| Preconditions: | The user must register for an account before logging in |
| Postconditions: | The user is successfully logged into their account and can access system functionalities |
| Priority: | High |
| Frequency of Use: | Regular |
| Flow of Events: | <ol style="list-style-type: none"> 1. The user accesses the system login page 2. If the user has an account, the user enters their username and password. 3. The user submits the login credentials |

| | |
|-----------------------|---|
| | <ol style="list-style-type: none"> 4. System verifies the credentials 5. If the credentials are valid, the system grants access to the user's account 6. If the credentials are invalid, the system displays an error message and prompts the user to re-enter their credentials |
| Alternative Flows: | If the user does not have an account, they can click on the "Create Account" link to register for a new account |
| Exceptions: | If the system is down or experiencing technical issues, the user would be unable to login |
| Includes: | n/a |
| Special Requirements: | Only losers are required to login to the system, finder is not required to login. |
| Assumptions: | Users have a valid username and password to log in to the system |
| Notes and Issues: | n/a |

| | | | |
|----------------|-----------------|--------------------|--|
| Use Case ID: | UC07 | | |
| Use Case Name: | Post Found Item | | |
| Created By: | Marcus Ong | Last Updated By: | |
| Date Created: | 7/02/2024 | Date Last Updated: | |

| | |
|-------------------|--|
| Actor: | User, System and Database |
| Description: | User uploads a lost item into the system. |
| Preconditions: | <ol style="list-style-type: none"> 1. The user can click/tap on the “Found an item” 2. The user has found a lost item and intends to upload it into the system |
| Postconditions: | <ol style="list-style-type: none"> 1. System has logged a new lost item into the database 2. The user receives a “Your lost item has been successfully uploaded into the database” message 3. The system displays the lost item as an annotation on the map |
| Priority: | High |
| Frequency of Use: | High |

| | |
|-----------------------|--|
| Flow of Events: | <ol style="list-style-type: none"> 1. User(Finder) clicks/taps “found an item”. 2. System generates a popup view for submitting a found lost item. 3. User fills in the information fields of the form. 4. User submits the form. 5. System removes the popup view and gives the displays to the user a thank you message. 6. System creates a new lost item object based on the uploaded details and adds the new lost item object into the database. |
| Alternative Flows: | If the user does not fill up a required field, they will be unable to submit the lost item form, and the system will prompt them to fill up the form before submitting again. |
| Exceptions: | n/a |
| Includes: | n/a |
| Special Requirements: | n/a |
| Assumptions: | <ol style="list-style-type: none"> 1. Assume that the user has a real lost item that he has found and wishes to upload it in the application |
| Notes and Issues: | <ul style="list-style-type: none"> - The user may communicate in a language that the system does not support, and thus be unable to read/write while using the application |

| | |
|--------------|------|
| Use Case ID: | UC09 |
|--------------|------|

| | | | |
|----------------|-------------------|--------------------|--|
| Use Case Name: | Finding Lost Item | | |
| Created By: | Xavier Tan | Last Updated By: | |
| Date Created: | 7/2/24 | Date Last Updated: | |

| | |
|-------------------|---|
| Actor: | User, System and Database |
| Description: | Finding lost item on map view |
| Preconditions: | The user lost an item and is trying to find it |
| Postconditions: | The user has successfully identified their lost item |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | <ol style="list-style-type: none"> 1. The user can use the filter to further aid in finding their lost item. 2. The user can click on various pins on the map view to see a pop up that details the specifications of the lost items. 3. Once the user has successfully identified the location of their pin (lost item) on the map view. They can click on the pin once again, leading to the pop up description of the item 4. After they verified the description of the item matches their lost item, they will proceed to contact the finder |

| | |
|-----------------------|---|
| Alternative Flows: | The user can't find item on map view and will have to submit a lost notice instead |
| Exceptions: | The system might not be updated instantly when the finder uploads a pin in the map view and the user can't find the item. |
| Includes: | n/a |
| Special Requirements: | The user must be logged in |
| Assumptions: | The user can remember characteristics of the lost item |
| Notes and Issues: | n/a |

| | | | |
|----------------|----------------|--------------------|--|
| Use Case ID: | UC0623 | | |
| Use Case Name: | Contact Finder | | |
| Created By: | Rhea | Last Updated By: | |
| Date Created: | 7/2/24 | Date Last Updated: | |

| | |
|-----------------------|--|
| Actor: | User, System and Database |
| Description: | User contacting Finder |
| Preconditions: | The user found item on lost view |
| Postconditions: | The item that has been collected and indicated will be removed from the application view. |
| Priority: | High |
| Frequency of Use: | Medium |
| Flow of Events: | <ol style="list-style-type: none"> 1. Clicking on the pop up which then leads to a page to contact the user who has found the item. 2. The user seeking their lost item should fill the form up. 3. This sends a follow up message to the finder to ask for more information 4. Communication regarding how to retrieve lost items is then handled outside of the application between the two parties. 5. The user (Loser) proceeds to the location to claim his lost item. 6. Once the user has collected their lost item they should click the button whereby it indicates they have collected the item. |
| Alternative Flows: | n/a |
| Exceptions: | Requires user activeness on the platform in order for orderly communication regarding lost items. |
| Includes: | Changing the message to an email to the finder |
| Special Requirements: | Requires some form of contact information |

| | |
|-------------------|--|
| Assumptions: | Assume that finder have email addresses to be contacted |
| Notes and Issues: | <ol style="list-style-type: none"> 1. User could possibly not have any contact information 2. Requires user to be active on the application in order to interact accordingly |

| | | | |
|----------------|------------|--------------------|--|
| Use Case ID: | UC010 | | |
| Use Case Name: | Settings | | |
| Created By: | Poh Qi Bin | Last Updated By: | |
| Date Created: | 7/2/24 | Date Last Updated: | |

| | |
|----------------|---|
| Actor: | User |
| Description: | User accesses and configures the settings page |
| Preconditions: | <ol style="list-style-type: none"> 1. User click on the Settings option in homepage 2. User is on Settings page |

| | |
|--------------------|---|
| Postconditions: | <ol style="list-style-type: none"> 1. The user's preferences are updated and reflected throughout the application. 2. Any changes made by the user are stored permanently. 3. User submits a description of the bug or issue |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | <ol style="list-style-type: none"> 1. User enters the application and clicks on settings 2. Settings page shows up and user have options "Change theme", "Report a bug", "Language", "Help", "Feedback", "Frequently Asked Questions(FAQ)" to choose from 3. User can report a bug or issue 4. User can request for help 5. User can switch language 6. User can give feedback 7. User can click on frequently asked questions (FAQ) section, which drops down a tab to give user more details on FAQs |
| Alternative Flows: | n/a |

| | |
|-----------------------|---|
| Exceptions: | n/a |
| Includes: | n/a |
| Special Requirements: | The UI must have both dark and light mode |
| Assumptions: | User understanding the default language before coming to the settings section |
| Notes and Issues: | n/a |

| | | | |
|----------------|------------|--------------------|--|
| Use Case ID: | UC05 | | |
| Use Case Name: | Mapview | | |
| Created By: | Phoebe Lee | Last Updated By: | |
| Date Created: | 7/2/24 | Date Last Updated: | |

| | |
|--------|-------------------------|
| Actor: | User and GoogleMaps API |
|--------|-------------------------|

| | |
|-------------------|--|
| Description: | The user can access the Mapview to view lost items based on their locations |
| Preconditions: | NIL |
| Postconditions: | The user is presented with the Mapview displaying pinned locations of items based on where they were found |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | <ol style="list-style-type: none"> 1. The user can access the system's map view feature 2. The user can select any optional filters they wish to apply to narrow down the search 3. The user may input specific location criterias to filter lost items on the map 4. System will update the map view based on the selected filters 5. The map will zoom into the region if location filter is filled 6. User can click on any pin annotations on the map to view the item's details in a pop-up 7. The pop-up will display the the item's descriptions |

| | |
|-----------------------|---|
| | 8. The user can interact with the map to explore different pinned annotations |
| Alternative Flows: | If the user wants to change or clear the filters, they can do it from the Mapview interface |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | The Mapview must integrate with Google Maps API to display lost items |
| Assumptions: | The users are familiar with interacting with map interfaces and know how to use filters |
| Notes and Issues: | NIL |

| | | | |
|----------------|----------------------|------------------|--|
| Use Case ID: | UC04 | | |
| Use Case Name: | Filtered Search Page | | |
| Created By: | Luo Maoyuan | Last Updated By: | |

| | | | |
|---------------|------------|--------------------|--|
| Date Created: | 07/02/2024 | Date Last Updated: | |
|---------------|------------|--------------------|--|

| | |
|-------------------|---|
| Actor: | User and Google Maps API |
| Description: | User utilises the filtered search page to more specifically identify lost items. |
| Preconditions: | <ol style="list-style-type: none"> 1. The user identifies as a Loser 2. The user attaches tags that is descriptive of the item he is trying to find |
| Postconditions: | The user will generate a list of items that matches the object he wishes to locate |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | <ol style="list-style-type: none"> 1. The user Navigates to Search Page: 8. Initiate Filtered Search & identifies the criteria they want to filter by, such as colour, location, category, etc. |

| | |
|-----------------------|---|
| | <p>9. The user inputs specific parameters for each selected filter criterion.</p> <p>10. The user clicks on the "Apply Filters" button or similar action to activate the selected filter criteria.</p> <p>11. The system processes the filter parameters and retrieves relevant search results based on the user's criteria.</p> <p>12. Results may be displayed in a list format, grid view, or on a map depending on the design of the search page.</p> |
| Alternative Flows: | <p>1. If there are no search results that matches the item description, User will be prompted to post a notice for lost item</p> |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | Requires a large enough database to handle all the tags attached to a posted item. |
| Assumptions: | <p>1. Losers can correctly remember the characteristics of their lost item.</p> |

| | |
|-------------------|--|
| | 2. Finders correctly attach the tags that correspond with the item they found. |
| Notes and Issues: | NIL |