SC2006: Software Engineering

Lab 1 Deliverables

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

Group members	**Matriculation Number**
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: 07/02/2024

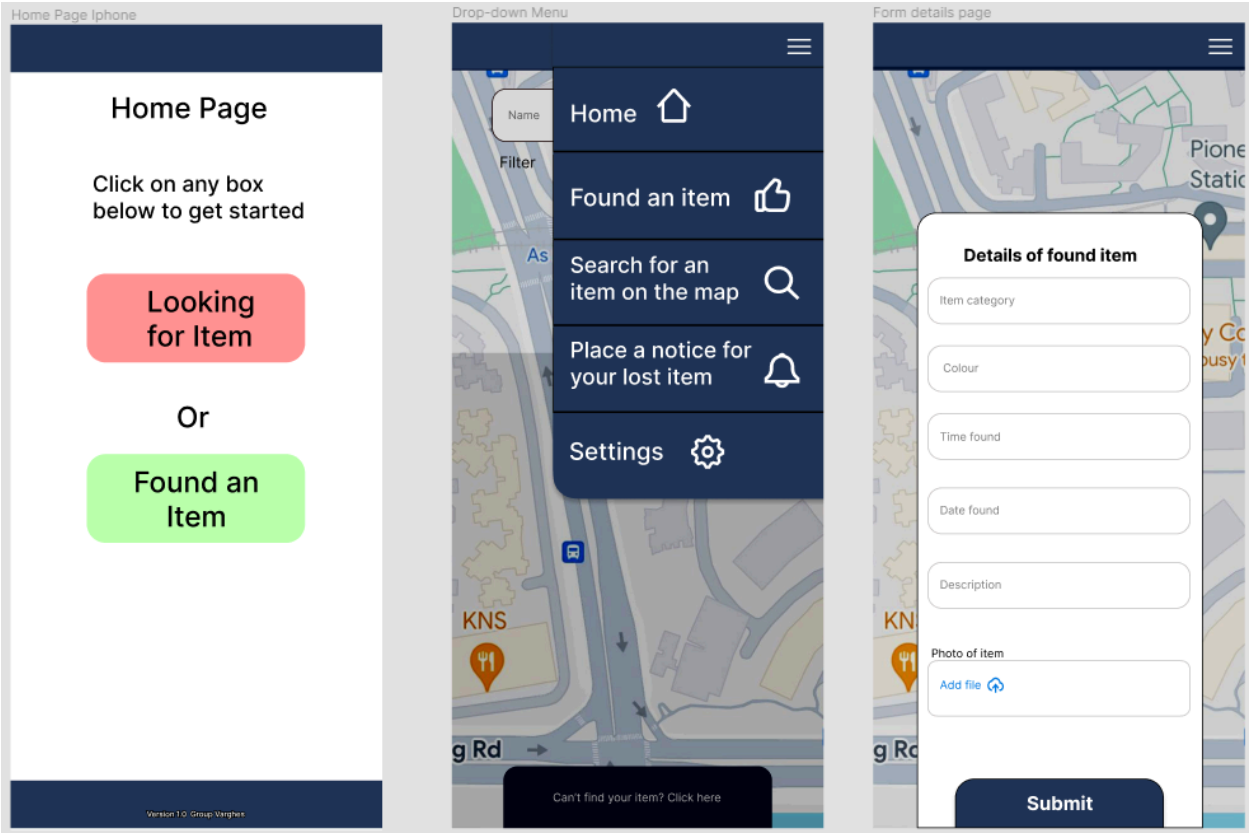
Document Outline

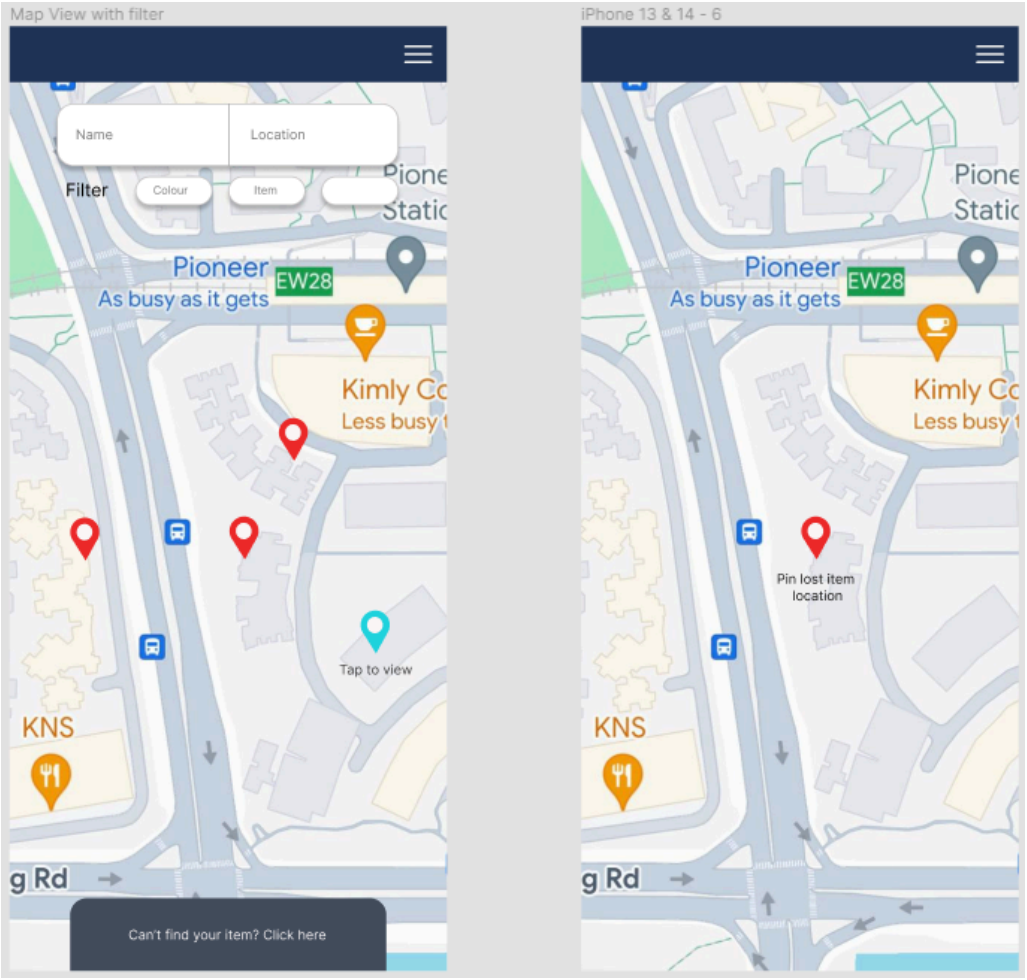
Project Description

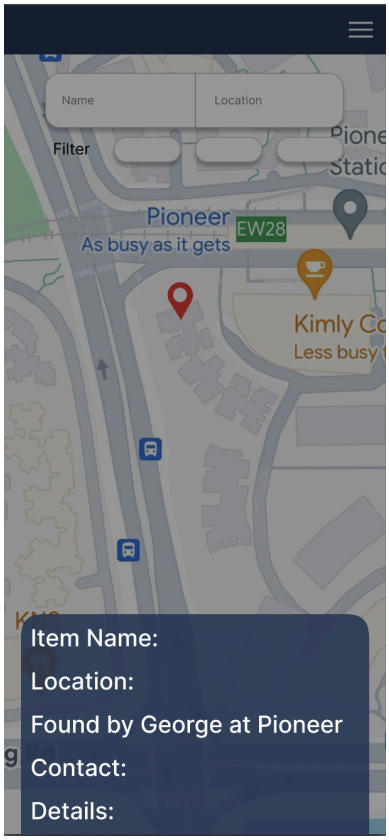
Our project is a lost and found system designed primarily for mobile web usage but also supports desktop web usage. The application allows users to upload lost items into the system and to find lost items that were uploaded into the system. The application utilises Google Maps' API to present users with a map view to provide an intuitive interface for users to view and interact with lost items in the system. Users are split into two usage groups, Finders and Losers. The application supports Finders and Losers contacting one another to aid in the retrieval of lost items.

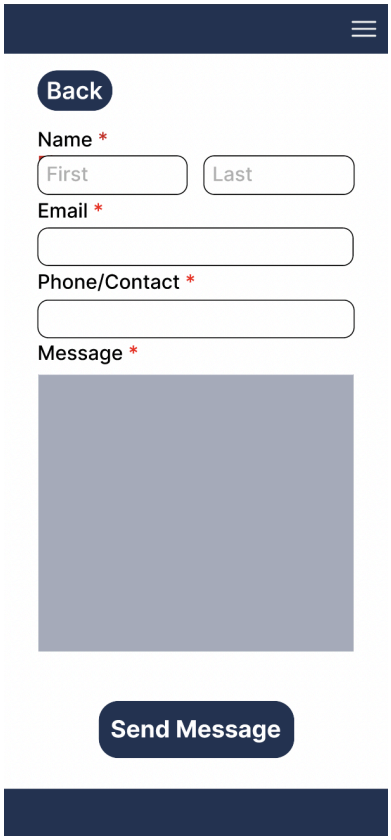
UI Mockups

Mobile View UI Mockup



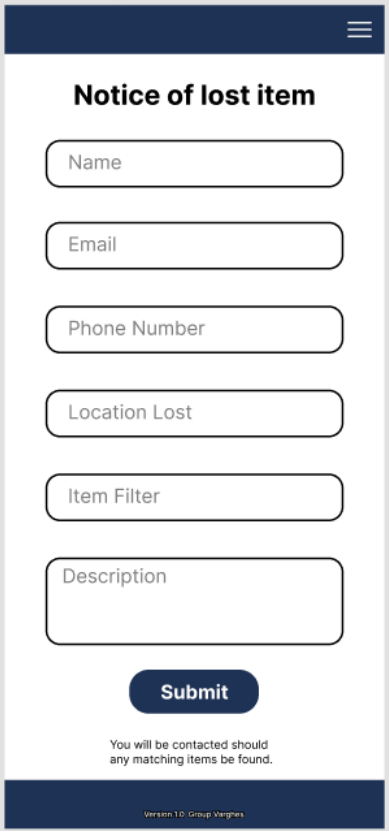

Map View with filter
Name
Location
Filter
Colour
Item
Pioneer
EW28
As busy as it gets
Kimly Cc
Less busy
Tap to view
KNS
g Rd
Can't find your item? Click here
iPhone 13 & 14 - 6
Pioneer
EW28
As busy as it gets
Kimly Cc
Less busy
Pin lost item location
KNS
g Rd





Settings page iphone

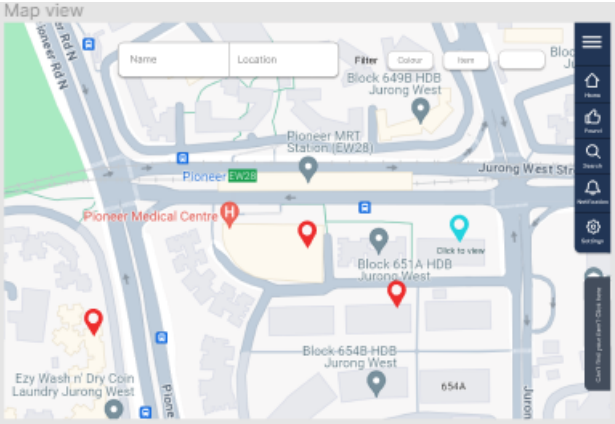
Notice Of Lost Item



Desktop View UI Mockup

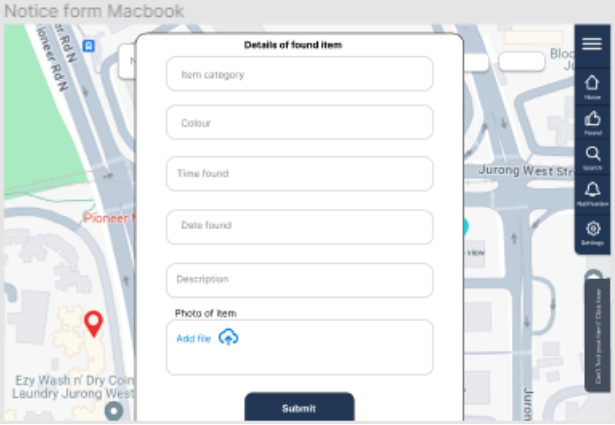
Home Page

Map view



Drop down menu macbook

Notice form Macbook



Settings page macbook

Notice Of Lost Item

Functional Requirements

1. Users who wish to upload a lost item they have found to the system are not required to sign in or create an account.
2. Users who wish to search for a lost item in the system are required to sign in or create an account.
    2.1. Users can sign in.
    2.2. Users who do not have an account can create one.
        2.2.1. Accounts created will be recorded in the “Accounts” Database.
3. The system includes a dropdown menu for users to access different functionalities.
    3.1. The dropdown menu appears upon a tap or a mouse click.
    3.2. The dropdown menu will display a list of selectable options.
        3.2.1. The dropdown menu will display a “Home” option
            3.2.1.1. The system redirects users to the “Home” page upon clicking.
        3.2.2. The dropdown menu will display a “Found an item” option.
            3.2.2.1. The system redirects users to the “Found an item” page (9.1) upon clicking.
        3.2.3. The dropdown menu will display a “Search for an item” option.
            3.2.3.1. The system redirects users to the “Search for item” page (4) upon clicking.
        3.2.4. The dropdown menu will display a “Place a notice for your lost item” option.
            3.2.4.1. The system redirects users to the “Place a notice for your lost item” page upon clicking.
        3.2.5. The dropdown menu will display a “Settings” option.
            3.2.5.1. The system redirects users to the “Settings” page upon clicking.
    3.3. The dropdown menu is present in all pages.
4. Users who are looking for a lost item are able to search for the item in the system.
    4.1. The system generates a filtered search page for a user who is looking for a lost item.
        4.1.1. The filtered search page takes in optional filters to select.

4.1.1.1. The filtered search page offers the text input of the item name.

4.1.1.2. The filtered search page offers the dropdown menu to select the item category.

4.1.1.3. The filtered search page offers the dropdown menu to select the item colour.

4.1.1.4. The filtered search page offers the text input of the item's found location.

4.1.1.5. The filtered search page is exited and performs the search upon the user clicking the search button.

4.2. The system directs the user to a results page containing the suitable lost items.

4.2.1. The results page contains a list view of all suitable lost items.

4.2.1.1. Each item displays an image if an image was uploaded.

4.2.1.1.1. The item displays a no-image message if no image was uploaded.

4.2.1.2. Each item displays the item name.

4.2.1.3. Each item displays the location it was found at.

4.2.1.4. Each item offers a button to view the item.

4.2.2. The results page offers the live amending of all filters mentioned in (3.1.1).

4.2.2.1. The lost items list view changes live on page when the filters are amended.

4.2.3. The results page offers the user a button to view and search on the map view instead.

4.2.4. The results page offers the user a button to click if they are unable to find their lost item.

4.3. The system outputs a "no items found" message on the lost items list view if no such item is found.

4.3.1. The results page offers all other functionalities mentioned in (4.2).

5. Users who are looking for a lost item are able to search for the item on the map view.

5.1. The map view offers optional filters for the user to narrow their search.

5.1.1. The map view offers a location filter.

5.1.1.1. The map view zooms into the local region of the selected location if the location filter is filled.

5.1.2. The map view offers all filters mentioned in (4.1.1).

5.1.3. The pin annotations on the map view display changes live on page when the filters are changed.

5.2. The map view displays a map containing pin annotations of lost items.

5.2.1. Each pin annotation is placed on the coordinates that their respective lost item was found.

5.2.2. The map view displays a small in-line popup above the pin annotation when the pin annotation is tapped.

5.2.2.1. The popup contains the item name.

5.2.2.2. The popup contains a button to view the item.

5.2.3. The map view page offers the user a button to click if they are unable to find their lost item.

5.3. Exceptions

6. The system is able to display a pop-up containing a lost item's information.

6.1. The user clicks the "view item" button of a chosen item to view the selected item's information pop-up.

6.2. The pop-up contains all of the lost item's information.

6.2.1. The pop-up contains the "Item Name", "Location Found", "Date Found", "Time Found" and "Location Placed" texts.

6.2.2. The pop-up may contain the finder's "First Name", "Last Name", "Contact Number" and "Email Address" texts, the item "Description" text, and a photograph of the item.

6.2.3. The pop-up contains a button to contact the finder by email if the finder has filled up the "Email Address" field.

6.2.3.1. The button directs the user to a secondary pop-up to send an email to the finder using the application.

6.2.3.1.1. The email pop-up prompts the user to enter their contact.

6.2.3.1.2. The email pop-up prompts the user to enter their message.

6.2.3.1.3. The email pop-up contains a send button.

6.2.3.1.3.1. The send button sends an email to the finder's email address with the filled up user's details.

6.2.4. The pop-up contains a "I have collected this item" button.

6.2.4.1. The button tags the item as found by the user's email address and removes the item from the viewable lost items.

6.2.4.2. The system thanks the user and wishes the user a good day.

6.3. Exceptions.

7. Users who click the “unable to find your item” button are redirected to a page to place a notice for the item.
    7.1. The page prompts the user to enter details about the item they are looking for as well as their contact details.
        7.1.1. The user is required to enter their name, email, phone number and item description and the location they last saw the item.
        7.1.2. The user is offered options to select tags for the item.
            7.1.2.1. The tags will include “Item colour” and “Item category”.
        7.1.3. The page contains a submit button for the user to submit their lost item notice.
    7.2. The user’s lost item notice is logged into the system database.
    7.3. Exceptions.
8. Users who have placed a lost item will be sent an email notification if any similar matching items are added into the system.
    8.1. The system is notified when any found lost items that are added to the system have a high enough similarity index with any lost item notices that have been placed.
    8.2. The system automatically writes an email to the lost item notice’s tagged email address that a similar item has been found.
        8.2.1. The email contains the found item’s information.
            8.2.1.1. The email contains the found item’s “Item Name”, “Location Found”, “Date Found”, “Time Found” and “Location Placed”.
            8.2.1.2. The email may contain the item finder’s finder’s “First Name”, “Last Name”, “Contact Number” and “Email Address”, the item “Description” text, and a photograph of the item.
            8.2.1.3. The email contains a URL link to view the item in the webpage.
9. Users who have found a lost item are able to input the lost item into the system.
    9.1. The system redirects users to a form page to fill in details about the item found.
        9.1.1. The form has mandatory fields that must be filled in.
            9.1.1.1. The form mandates users to complete the "Item Name" text field.
            9.1.1.2. The form mandates users to complete the “Location Found” text field.
            9.1.1.3. The form mandates users to complete the “Date Found” field.

9.1.1.4. The form mandates users to complete the “Time Found” field.

9.1.1.5. The form mandates users to complete the “Location Placed” text field.

9.1.2. The form page takes in optional fields to be filled in

9.1.2.1. The form page supports the optional input of a “First Name” text field.

9.1.2.2. The form page supports the optional input of a “Last Name” text field.

9.1.2.3. The form page supports the optional input of a “Contact Number” numerical field.

9.1.2.4. The form page supports the optional input of an “Email Address” text field.

9.1.2.5. The form page supports the optional input of a “Description” text field.

9.1.2.6. The form page supports the optional input of a photograph that captures the item found.

9.2. The system records the entry into the “Found Items” database.

9.2.1. The system generates a pop-out window that displays a thank you message to the user.

10. Users can adjust page settings in the “Settings” page.

10.1. The “Settings” page will have a list of selectable options.

10.1.1. The “Settings” page will have a toggle button to switch between “Light-mode” & “Dark-mode”.

10.1.2. The “Settings” page will have a “Report a bug” option.

10.1.3. The “Settings” page will have a “Help” option.

10.1.4. The “Settings” page will have a “Language” dropdown menu.

10.1.4.1. Users can select between the four options, “English”, “Chinese”, “Malay” & “Tamil”.

10.1.4.2. The system reloads the web application, & changes every text line in the web application to the selected language option upon selection.

10.1.5. The “Settings” page will have a “Feedback” option.

10.1.6. The “Settings” page will have a “FAQ” option.

10.1.7. The “Settings” page displays the web applications version number.

10.1.8. The “Settings” page displays the project team name.

Non-Functional Requirements

Usability:

1. User Interface Design:
    1.1. Requirement: The user interface shall be intuitive, with clear and consistent navigation throughout the application.
    1.2. Requirement: The design shall follow established usability principles, incorporating a user-friendly layout and visual hierarchy.
2. Consistency:
    2.1. Requirement: The user interface elements, terminology, and interactions shall be consistent across all pages and modules of the web application.
    2.2. Requirement: Feedback messages, alerts, and notifications shall follow a standardised format for clarity.
3. Efficiency:
    3.1. Requirement: Common user tasks, such as account updates or password changes, shall be accomplished with at most 4 steps.
    3.2. Requirement: The web application shall provide shortcuts and keyboard commands such as drop down menus to enhance efficiency.

Maintainability:

1. Version Control:

1.1. Requirement: The source code and configuration files shall be maintained in a version control system such as Git, with clear commit messages.

1.2. Requirement: Releases and versioning shall follow a standardised approach, making it easy to roll back to previous versions if necessary.

2. Documentation:

2.1. Requirement: Comprehensive documentation shall be provided for the system architecture, APIs, databases, and configuration settings.

2.2. Requirement: The documentation shall be updated every time there is a slight change to the system.

Performance:

1. Response Time:

1.1. Requirement: The web application shall provide a response time of under 2 seconds for any queries or page redirects.

1.2. Requirement: The system shall maintain an average response time of 3 seconds during peak usage

2. Throughput:

2.1. Requirement: The application shall support a minimum of 200 concurrent users without a more than 15% decrease in throughput.

2.2. Requirement: The system shall handle 300 transactions per minute during peak load.

Scalability:

1. Concurrent Users:

1.1. Requirement: The web application shall support a minimum of 200 concurrent users without a significant decrease in performance.

2. Scalable APIs:

2.1. Requirement: The chosen web application's APIs should be designed to scale independently, being able to sustain the increase in user traffic without a more than 15% decrease in response time & link availability.

Portability:

4. Mobile Responsiveness:

4.1. Requirement: The application shall be responsive and optimised for mobile devices, such as reframing the web application to have a mobile friendly display

4.2. Requirement: Touch-friendly interactions and gestures shall be supported for mobile users.

Security:

1. Authentication:

1.1. Requirement: The web application shall implement secure authentication mechanisms, such as multi-factor authentication, to verify the identity of users.

1.2. Requirement: User passwords shall be stored securely using industry-standard encryption algorithms and practices.

2. Data Protection:

2.1. Requirement: Personally Identifiable Information (PII) and other sensitive user data shall be encrypted during transmission over the network (using HTTPS) and storage.

Reliability:

1. Availability:

1.1. Requirement: The web application shall have an availability target of 99.9%, allowing for scheduled maintenance periods.

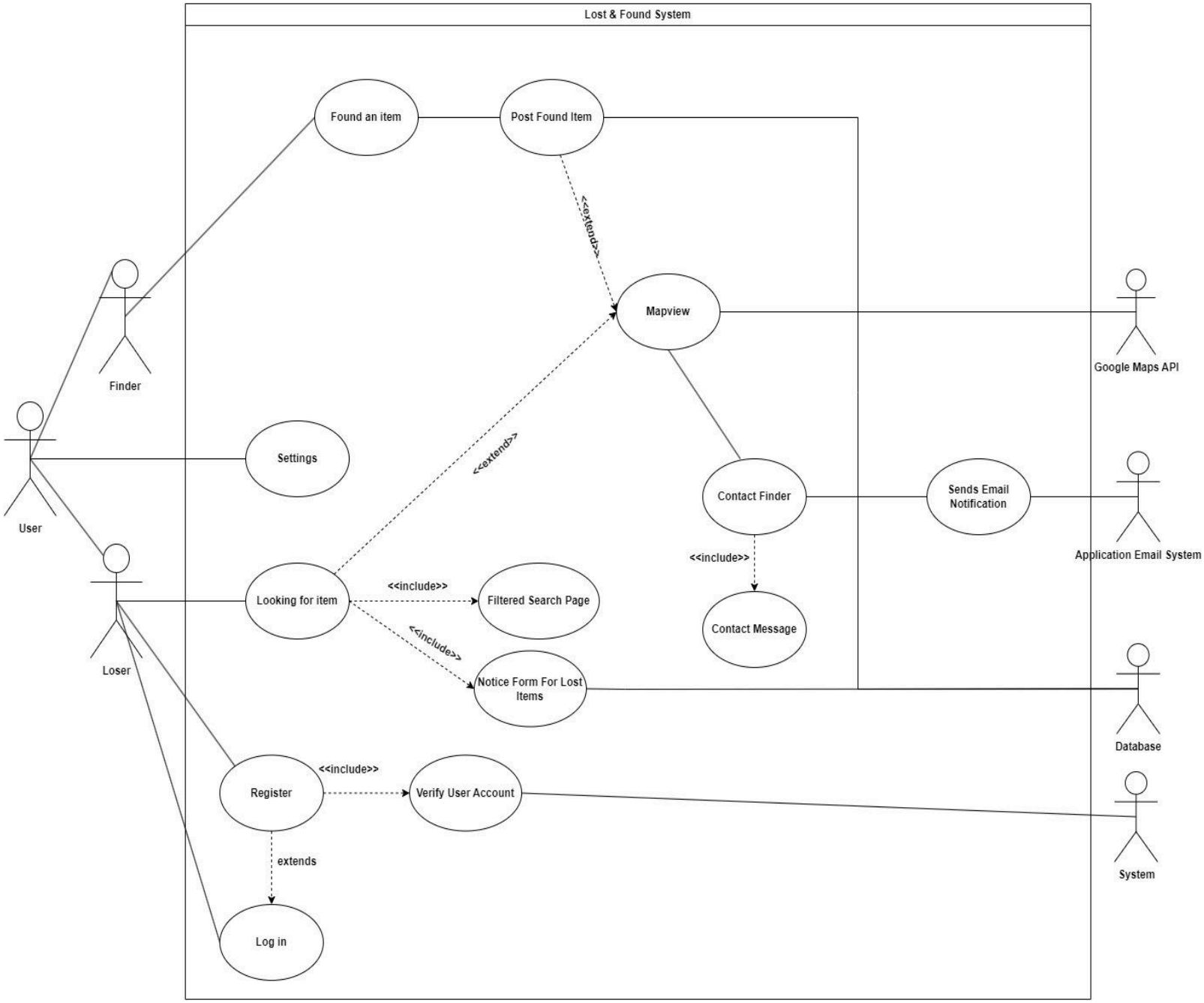
1.2. Requirement: Planned downtime for maintenance shall be communicated to users 14 days in advance.

2. Backup and Recovery:

2.1. Requirement: Regular backups of critical data shall be performed, at least once every 7 days.

Use Case Model

Use Case Diagram


Lost & Found System
Found an item
Post Found Item
<<extend>>
Mapview
Google Maps API
Finder
Settings
<<extend>>
Contact Finder
Sends Email Notification
Application Email System
User
<<include>>
Looking for item
<<include>>
Filtered Search Page
Contact Message
Loser
<<include>>
Notice Form For Lost Items
Database
Register
<<include>>
Verify User Account
System
extends
Log in

Use Case Description

Use Case ID:	UC02		
Use Case Name:	User login		
Created By:	Phoebe Lee	Last Updated By:	
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:	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 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:	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:	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:	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:	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:	- 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:	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:	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:	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:	1. User click on the Settings option in homepage 2. User is on Settings page
Postconditions:	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:	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 Mapvew 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:	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:	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:	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. 9. The user inputs specific parameters for each selected filter criterion. 10. The user clicks on the "Apply Filters" button or similar action to activate the selected filter criteria. 11. The system processes the filter parameters and retrieves relevant search results based on the user's criteria. 12. Results may be displayed in a list format, grid view, or on a map depending on the design of the search page.

Alternative Flows:	1. If there are no search results that matches the item description, User will be prompted to post a notice for lost item
Exceptions:	NIL
Includes:	NIL
Special Requirements:	Requires a large enough database to handle all the tags attached to a posted item.
Assumptions:	1. Losers can correctly remember the characteristics of their lost item. 2. Finders correctly attach the tags that correspond with the item they found.
Notes and Issues:	NIL

Data Dictionary

Term	Definition
User	An individual using the web application. The user is a general term describing any user of the application, for any purpose.
System	An entity describing the web application as a whole, consisting of all of its different domains including its front-end display and back-end logic. System may be used to describe the application entity's output action and internal logic.
Finder	A user who has found a lost item and whose purpose of using the application is to upload the lost item into the application system.
Loser	A user who has lost an item and whose purpose of using the application is to discover the lost item on the application and retrieve it.
Lost Item	An item in the system that describes an object that has been lost that is either uploaded into the system by a finder or is being searched for by a loser.
Similarity_index	An index for how similar a lost item uploaded by a finder matches the information data of a lost item put on notice by a loser.
Map	The map view displayed and used by the system, generated using the Google Maps API.
Local Region	A 500-metre radius circular region on the map centred around

	a single selected coordinate.
Pin Annotation	A viewable icon designated on a specified coordinate on the map that users can click/tap on to interact with.
Text Filter	A filtered search that takes in a text input from the user.
Dropdown Filter	A dropdown menu that users can select from to choose a tag to filter their search options.
Notice	An item in the system that is notified to perform an action when a specified event occurs.
Photograph	A .jpg file containing an image in the system.