Software Requirements Specification

for

Lost & Found

Version 1.0 approved

Prepared by

Bernard Chiang
Cheong Jia Rong
Guo Yuan Lin
Wang Jie Rui,
Jerome
Rodmond Tan

Nanyang Technological University, A43 Team Meta-4

09-04-2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Everyone	18/1/2023	Initial documentation after specifying requirements	1.0
Everyone	9/4/2023	Update non-functional requirements, use case models and user interface	1.0.1

1. Introduction

1.1 Purpose

The purpose of this document is to build a mobile application for people to exchange information about lost items.

This SRS will cover the system which runs this product.

1.2 Intended Audience and Reading Suggestions

This project is a prototype of a lost and found system and it's intended to be used within NTU's premises. This has been implemented under the tutelage of Lab assistants and professors. This project will be useful for all users who uses the system, particularly people to tend to lose things.

1.3 Product Scope

The Lost & Found mobile application is created and designed to be a one stop place for people who have lost items or found items can go to exchange information, and acts as a social media platform for the facilitation of lost and found items.

Lost items can be posted for users to help with the search or update people who have found said lost item. Found items can be posted for owners to come and identify their ownership.

2. Overall Description

2.1 Product Perspective

Lost&Found is a novel app that helps users find their beloved lost items. The app is designed for anyone who has lost or found an item, and who wants to report the item or connect with others who may have information about the item. It also uses interactive map display with precise pins to indicate the location of the reported lost/found items to ease the lost and found process. Users are also able to search for users/posts/category based on a filter system. The introduction of reputation scores and reports will increase the credibility of the users, enhancing the credibility of Lost&Found. Lost&Found aims to simplify the process of finding lost items and to increase the chances of reuniting lost items with their owners.

2.2 Product Functions

- *Users can create account and log into their accounts.*
- *Users can create posts for either lost/found items.*
- *Users can view all the unresolved posts on the main page.*
- *Users can view all the unresolved posts on the map page.*
- *Users can search for posts via user/category/items.*
- *Users can view other users profile.*
- *Users can make a request to a post (either I found it/ I lost it).*
- *Post owners can receive the requests.*
- *Post owners can verify the request via an approve/reject button.*
- *Users can report fake posts.*
- *Reported users will have reputation decreased.*
- *Verified lost/found requests will increase the finder's reputation score.*

2.3 User Classes and Characteristics

User
Owners of Lost Items:
These are individuals who have lost an item and are using the app to report the item as lost and to search for it among items that have been found and reported by others.

Finders of Lost Items:
These are individuals who have found a lost item and are using the app to report the item as found and to search for the rightful owner of the item among items that have been reported as lost by others.

Regular Users:
These are individuals who use the app to browse lost and found items, search for specific items, and communicate with owners or finders of items.

2.4 Operating Environment

Development Environment	Description
Front-end: React Native for Mobile Application	React Native is a User Interface(UI) Software Development Library that is created by Facebook. It allows the development of native apps. Hence, the same codebase could be used to run in both Android and iOS phones
Backend: Node.js,Express.js, MongoDB	1. Authentication is handled by JsonWebToken 2. All data is stored in MongoDB

2.5 Design and Implementation Constraints

1. This application uses local storage for file storage which may be limited.
2. This application uses Mongo Atlas for database storage, which is limited at 512MB.
3. Security Constraints: Data privacy needs to be protected, hence passwords must be stored as hashes.
4. Authentication and Authorization Constraints: The app must be designed to ensure that only authorized users can access sensitive information about lost and found items, such as contact information for the owners.
5. Reporting Constraints: The app must be designed to allow users to report lost or found items, including the ability to attach photos and provide detailed descriptions of the items.
6. User Interface Constraints: The app must be designed with a user-friendly interface that allows users to easily report lost or found items, search for items, and view the status of their reports.

2.6 User Documentation

No user documentation needed as our app is interactive and intuitive to use, with good labelling to guide users around our app.

2.7 Assumptions and Dependencies

Dependencies:

1. OneMap API

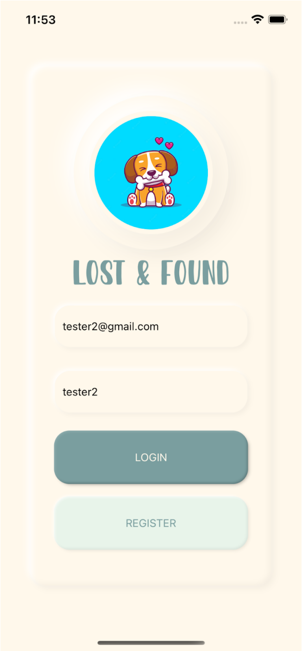
Assumptions:

1. Data obtained from OneMap API are accurate.
2. Users have strong internet connection as most data are delivered by the server.

3. External Interface Requirements

3.1 User Interfaces

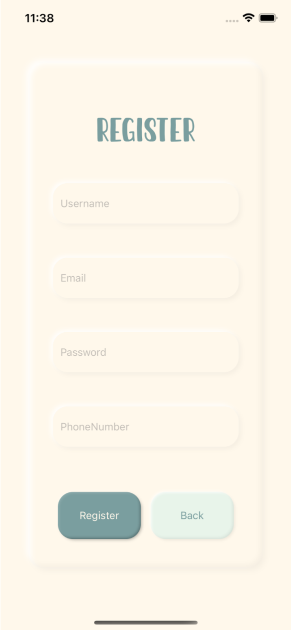
Login/Sign up Page



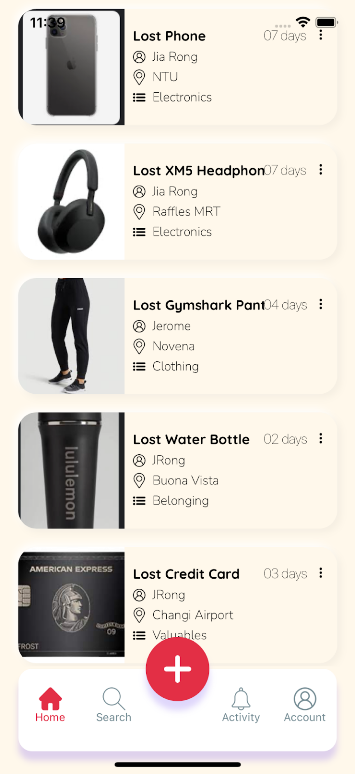
tester2@gmail.com

tester2

Registration Page



HomePage



11:39

Lost Phone 07 days
Jia Rong
NTU
Electronics

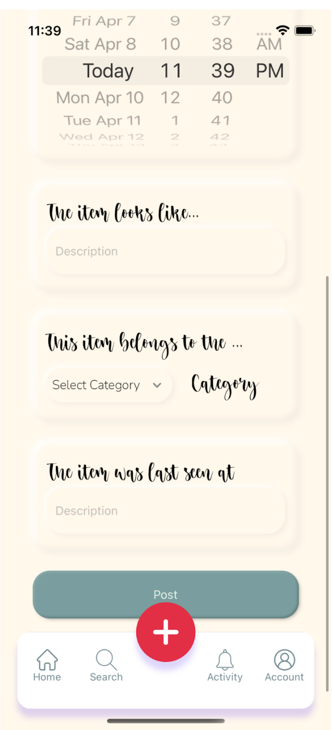
Lost XM5 Headphon 07 days
Jia Rong
Raffles MRT
Electronics

Lost Gymshark Pant 04 days
Jerome
Novena
Clothing

Lost Water Bottle 02 days
JRong
Buona Vista
Belonging

Lost Credit Card 03 days
JRong
Changi Airport
Valuables

Create post page



11:39

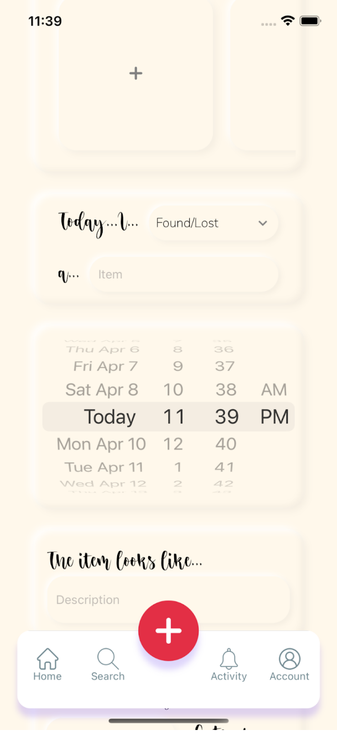
Fri Apr 7 9 37
Sat Apr 8 10 38 AM
Today 11 39 PM
Mon Apr 10 12 40
Tue Apr 11 1 41
Wed Apr 12 2 42

The item looks like...
Description

This item belongs to the ...
Select Category
Category

The item was last seen at
Description

Post



11:39

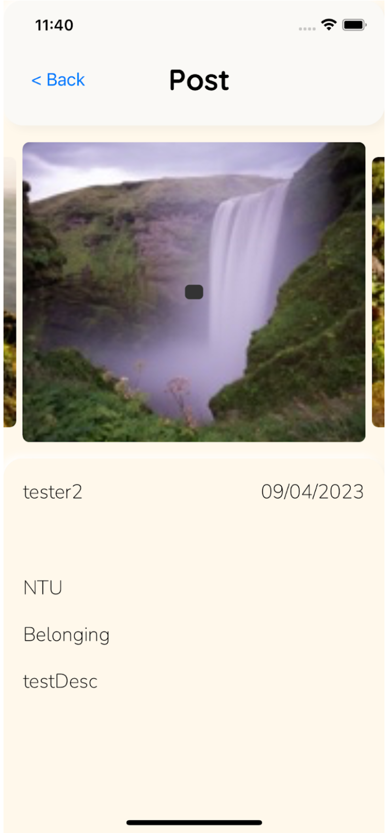
+

Today... I... Found/Lost
a... Item

Thu Apr 6 8 36
Fri Apr 7 9 37
Sat Apr 8 10 38 AM
Today 11 39 PM
Mon Apr 10 12 40
Tue Apr 11 1 41
Wed Apr 12 2 42

The item looks like...
Description

Post view page



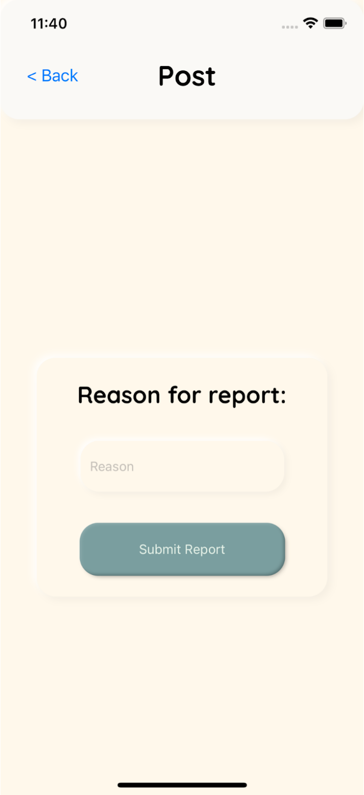
tester2 09/04/2023

NTU

Belonging

testDesc

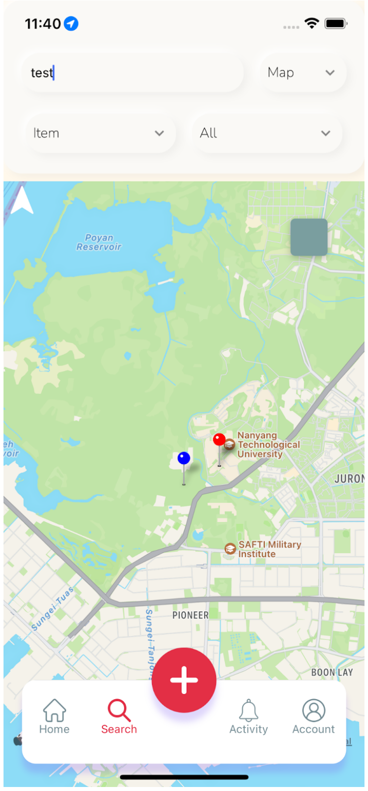
Report page



Reason for report:

Reason

map page



11:40

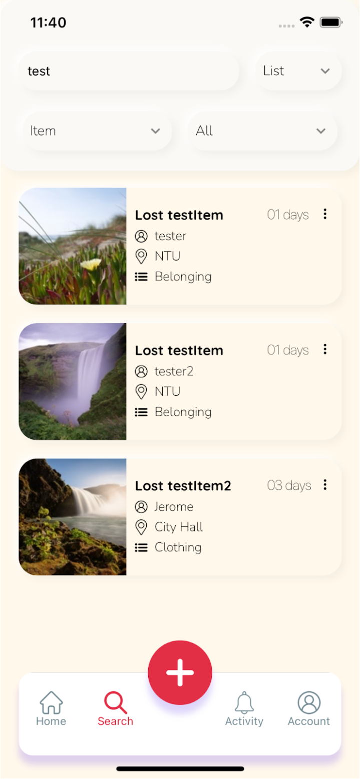
test

Map

Item

All

search page



11:40

test

List

Item

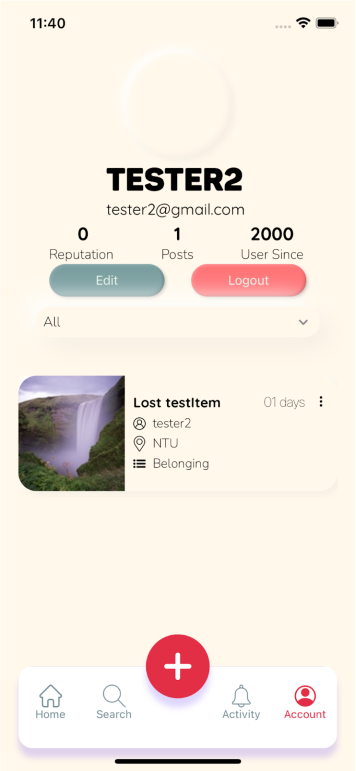
All

Lost testItem 01 days
tester
NTU
Belonging

Lost testItem 01 days
tester2
NTU
Belonging

Lost testItem2 03 days
Jerome
City Hall
Clothing

accounts page



11:40

TESTER2

tester2@gmail.com

0 Reputation 1 Posts 2000 User Since

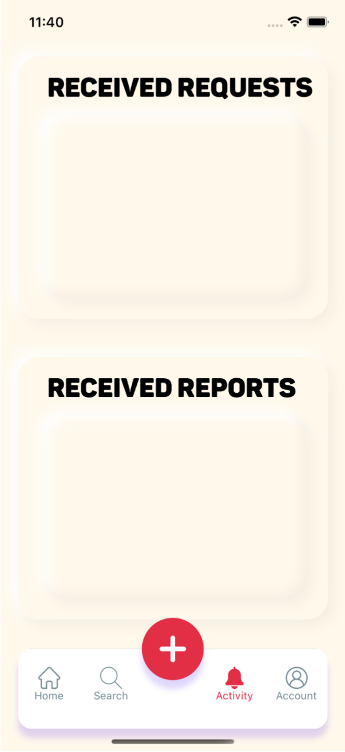
Lost testItem 01 days

tester2

NTU

Belonging

activity page



11:40

RECEIVED REQUESTS

RECEIVED REPORTS

3.2 Hardware Interfaces

Mobile Device Interface

Logical Characteristics: Lost & Found is designed to run on both Android and iOS devices.
The app will interact with the device's camera and GPS functions to allow users to report and search for lost and found items based on location and visual identification.

Physical Characteristics: Lost & Found will be installed and run on the user's mobile device.
The app will utilize the device's camera and GPS hardware to capture images and location data.

OneMap API Interface

Logical Characteristics: Lost & Found utilizes OneMap APIs to display the item locations for a pictorial image of the locations hence increasing the chances of finding lost items.

Physical Characteristics: The app will communicate with the OneMapAPI through REST APIs, using JSON data.

3.3 Software Interfaces

1. MongoDB - Version 4.0 or higher
   MongoDB is a NoSQL document database used to store and manage data about lost and found items. The app will communicate with the MongoDB database using the Mongoose ODM library.

2. Node.js - Version 12 or higher
   Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to build scalable, high-performance web applications. The app will use Node.js as the server-side runtime environment.

3. Express.js - Version 4.17 or higher
   Express.js is a fast, minimalist web framework for Node.js that provides a set of robust features for web and mobile applications. The app will use Express.js to build the RESTful API endpoints for data communication between the server and client.

4. React Native - Version 0.64 or higher
   React Native is an open-source mobile application framework created by Facebook that allows developers to build native mobile apps for iOS and Android using JavaScript and React. The app will use React Native to build the mobile user interface and logic for the app.

Data items and messages coming into the system and going out include:

1. User authentication and authorization data
2. Lost and found item data, including item descriptions, location data, and contact information for owners and finders

3. Media files related to lost and found items, such as images
4. The purpose of each of these data items and requests is to provide users with the ability to report and search for lost and found items, as well as to manage the ownership and recovery process of these items.

Services needed include:

1. User authentication and authorization services
2. Database management services for storing and retrieving lost and found item data
3. Media storage and retrieval services for storing and retrieving media files related to lost and found items
4. Location-based services for determining the location of lost and found items and users

   The nature of communications between these services is RESTful API communication over HTTP or HTTPS protocols.

   Data that will be shared across software components includes user authentication data, lost and found item data, and image files related to lost and found items. The data sharing mechanism will be implemented using MongoDB as the database management system and REST APIs for data communication.

3.4 Communications Interfaces

The communication between frontend and backend will be following HTTPS protocol and TCP/IP network protocol.
User authentication and authorization will be implemented using token authentication, which will be securely transmitted in the HTTP request header.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Create Account

4.1.1 Description and Priority

Register an account for new users. The system will prompt the user for their username, email, password and phone, and register the user once entered.
Priority level: High

4.1.2 Stimulus Response Sequences

1. User clicks on register button in the log in page

2. System display text box for username, email, password and phone
3. User enters all four boxes with valid inputs and clicks on the register button
4. System will check that all fields have been entered and email is not registered in the current database
5. Once inputs are verified, system will register the account into the database and bring the user back to the log in page

4.1.3 Functional Requirements

REQ_1 The system shall allow a user to register for a new account
REQ_2 The system shall only register fully accounts with all details fulfilled

4.2 Login

4.2.1 Description and Priority

Log in to the app using an existing account.
Priority level: High

4.2.2 Stimulus Response Sequences

1. The system displays the login page.
2. The user supplies username and password and clicks on the "login" button.
3. The system authenticates the user
4. The system displays the home page upon successful authentication

4.2.3 Functional Requirements

REQ_03 User shall only be able to log in with valid login credentials.
REQ_04 The system shall authenticate login credentials (username and password) using hashed password encryption mechanisms.

4.3 Create Post

4.3.1 Description and Priority

Users will be able to create post on their lost items or found items.
Priority level: High

4.3.2 Stimulus Response Sequences

1. The user will click on the create post icon indicated by the red plus
2. The system will display the create post page with the post details to be filled up

3. The user will fill in the item type (lost/found), item name, category, location and time
4. Optionally user will upload image and fill in the item description
5. Once all fields are filled the users will click on the post button
6. The system will display a success or error message, as appropriate
7. For successful post the system will add the post to the user's profile page as well as the home page

4.3.3 Functional Requirements

REQ_05 The system will only allow logged in users to upload posts
REQ_06 The system will only allow completed post to be posted

4.4 Search Post

4.4.1 Description and Priority

Users will be able to search post by typing in the search box.
Priority level: Medium

4.4.2 Stimulus Response Sequences

1. The user will click on the search icon in the navigation bar
2. The user will then type in the item or user name in the search bar and filter based on search type and category of post
3. The system will then display the posts that fulfill the conditions in step 2

4.4.3 Functional Requirements

REQ_01 The system will only allow logged in users to search
REQ_03 The system will display posts to users

4.5 View Post

4.5.1 Description and Priority

Users will be able to view the post by clicking on a specific post.
Priority level: High

4.5.2 Stimulus Response Sequences

1. The user will click on the specified post to be viewed
2. The system will then show the post details to the user

4.5.3 Functional Requirements

REQ_01 The system will only allow logged in users to view post
REQ_02 The system will display post to users

4.6 View Map

4.6.1 Description and Priority

Users will be able to view the posts as pins on a interactive map.
Priority: High

4.6.2 Stimulus Response Sequences

1. User will click on the view map button
2. System will display interactive map with pins, blue for current location, green for found items and red for lost items.
3. Users will click on a pin
4. System will display the item name and item description and location of the clicked pin

4.6.3 Functional Requirements

REQ_01 The system will be able to show post items in a map view

4.7 Edit Post

4.7.1 Description and Priority

User can edit posts that were previously created by themselves
Priority: High

4.7.2 Stimulus Response Sequences

1. The user will click on the create i.e. "+" button on the navigation bar
2. The system will bring the user to a post creation page
3. The user will fill in the post details such as item name, whether it's lost, item category, location, date, time, item description and images of item
4. Once created, the user will click on submit button to finalize creation of post

4.7.3 Functional Requirements

REQ_01 Only authorized users are allowed to create posts
REQ_04 The system shall only create posts with all details fulfilled

4.8 Remove Post

4.8.1 Description and Priority

Users can delete posts previously created by themselves
Priority: High

4.8.2 Stimulus Response Sequences

1. The user will click on their profile page
2. The system will display the user profile including all their previous posts
3. The user will find the post he wants to delete and click on the three dots at the top right of the post
4. The system will show a dropdown to edit or delete the post
5. The user will click on the delete button
6. The system will prompt the user are if he is sure to delete the post
7. The user will click the yes button
8. The system will delete the post from the user's profile page and the shared home page

4.8.3 Functional Requirements

REQ_04 The user can only delete their own posts.
REQ_08 Once a post is removed, it will not be shown to any users or in the system.

4.9 Make Request

4.9.1 Description and Priority

When the original owner of the lost items / finder of the lost items, saw the corresponding post, they can make a request to the post owner, indicating that they lost it or found it respectively.
Priority: High

4.9.2 Stimulus Response Sequences

1. The user click on the post that they want to make a request, from the home page.
2. The user click on the "I found it"/ "I lost it" button depending on whether its a lost item post or found item post.

4.9.3 Functional Requirements

1. The system must allow user to make a request

4.10 Update Request

4.10.1 Description and Priority

When the post owner received a request from potential owner/finders of the posted lost item, the post owner can approve/reject the request after verification.
Priority: High

4.10.2 Stimulus Response Sequences

1. User clicks on the activity page to view received requests.

4.10.3 Functional Requirements

REQ_01 The system must reject all other requests tagged to the post after one request tagged to the post has been accepted.
REQ_02 The system must show the list of requests received by the post owner.

4.11 Report User

4.11.1 Description and Priority

The report function is used when a user spot fake posts or received fake requests from other users.
Priority: Medium as we do not have a dedicated admin.

4.11.2 Stimulus Response Sequences

1. The User will click on the profile of the user he/she wants to report, and click on the report button.

4.11.3 Functional Requirements

REQ_01 The system must allow users to fill in details of the report
REQ_02 The system must update the reputation score of the reported users after the report has been verified to be true.

4.12 View User Profile Page

4.12.1 Description and Priority

When user click on the posts, it will direct them to the post owner profile page. User can also access their own profile page via the bottom navigation bar - Account.
Priority: High

4.12.2 Stimulus/Response

1. User clicks on the Account button in the bottom navigation bar.
2. The system redirects the user to their profile page.

4.12.3 Functional Requirements

REQ_01 The system must check the User is logged in to use this feature
REQ_02 The system must allow the User to view their own posts
REQ_03 The system must allow User to filter their posts based on All/Resolved/Unresolved

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software application should be able to handle 500 concurrent users without a drop in performance. It should also be able to handle at least 5000 user information including their account, posts, chats and reports made. 90% of users should be able to complete signup within < 5 min. The whole documentation process should take < 5 min. Ideally, 90% of the cases should take <= 2 min to finish documenting.

5.2 Safety Requirements

90% of users should be able to identify the trustworthiness of users in < 1 min by checking the other users' reputation score.

5.3 Security Requirements

Ensure that 100% of users are authenticated via login page before they are able to post, view posts, view other user information, report and make any form of requests within the app. Ensure that while logged in, the users for their login session have a token to authenticate their access to the app functions during that particular login session.

5.4 Software Quality Attributes

The app software should be easily maintainable, so that changes or updates can be made without disrupting the existing functionality. The app should also be able to have new features created on top of the existing ones without disrupting existing modules.

5.5 Business Rules

Only authorized users can access all software features of the app with the exception of login and registration. Only authorized users who created a specific post are allowed to edit or delete it, while the other authorized users can only view or report posts that are not theirs. Display error messages when users encounter an error such as logging failure.

6. Other Requirements

Use of an online cloud database to ensure that entries created while using the app are not stored on the users' local storage but a centralized database.

Appendix A: Glossary

Term	Definition
Report	When a dispute arises, an incident is automatically created for admin to process, investigate and resolve.
Item Finders	Users of the app that found missing items (listed/non-listed)
Item Owners	Users of the app that report their missing items.
Reputation score	A score calculated based on both positive actions that help the community, and negative actions that harmed the community.
Item Post	Post (in the forum) that documents details of lost or found items.
Lost Item	Post that shows an item that is of “lost” status
Found Item	Post that shows an item that is of “found” status
Request	A request made by sender to notify the post owner that the sender is the owner/finder of the post items.
Home Page	A page whereby users of the app can report missing items/ found items.
Login Page	A page whereby users of the app can log in to the app from.
Activity Page	A page that shows all the requests and reports received for the user
Register Page	A page whereby users of the app can register from.
Map	Map Interface that shows Singapore and various pins denoting the lost or found posts.

Resolved cases	Cases whereby lost items are found and the posts are thereafter closed, will be labelled as “RESOLVED”.

Appendix B: Analysis Models

In the lab 5 folder