



Wentworth Institute of Technology

COMP4960 – Software Engineering

Instructor: Dr. Koorosh Firouzbakht

Software Design Document

for

Campus Lost and Found Portal

Version 1.4

Prepared By

Team 2:

Craig Bakke, bakkec@wit.edu

Bjordi Ismailati, ismailatib@wit.edu

Aliyan Hidayatallah, hidayatallaha@wit.edu

Sammy Liao, liaos@wit.edu

GitHub Link

<https://github.com/casbakke/comp-4960-team-2>

11/03/2025

Contents

Revision History.....	3
1. Introduction.....	4
1.1. Document purpose.....	4
1.2. Product overview.....	4
1.3. Product functionality.....	4
1.4. Definitions.....	4
1.5. Acronyms and abbreviations.....	4
2. System requirements.....	4
2.1. Functional requirements.....	4
2.2. Non-functional requirements.....	4
2.3. Other requirements.....	4
3. System architecture.....	5
3.1. Overall architecture.....	5
3.2. Components mapping.....	5
3.3. Technology stack selection.....	5
4. System Design.....	5
4.1. UI.....	5
4.2. Class diagram.....	5
4.3. Sequence/activity diagram.....	5
4.4. Database.....	5
5. Others.....	5
6. Test plan.....	6
6.1. Function #1.....	6
6.2. Function #2.....	6
7. References.....	6

Revision History

Date	Version	Description	Author(s)
10/02/2025	1.0	<i>Initial changes completed in the document.</i>	Craig Bakke Bjordi Ismailati
10/22/2025	1.1	Edited 1.3, 1.5, 3.3, 4.1, 6.	Sammy Liao Bjordi Ismailati Craig Bakke
10/23/2025	1.2	Edited 4.1, 4.3, 4.4, 1.1, 6.	Sammy Liao Bjordi Ismailati Craig Bakke
10/30/2025	1.3	Edited 6, 4.2.	Bjordi Ismailati Aliyan Hidayatallah Sammy Liao Craig Bakke
11/06/2025	1.4	Updated document according to professor feedback from v1.3. Added diagrams to 3.1, 3.2.1, and 3.2.2.	Bjordi Ismailati Craig Bakke

1. Introduction

1.1. Document purpose

The purpose of this document is to list all the details on how the Web and IOS application will be constructed. It will describe the architecture and design of each part of the application. It will include a walk through each step and function of the application as well as display all test cases that will be used for all functions.

1.2. Product overview

1.2.1. Problem statements

Around the Wentworth Institute of Technology campus, students and faculty frequently lose personal items (e.g., keys, IDs, electronics, textbooks, clothing, bags) across various campus buildings, labs, and common areas. Currently, there is no centralized, easily accessible system for reporting and tracking these lost and found items.

1.2.2. Proposed solution

Our Solution is an iOS and Web application, the Campus Lost & Found Portal. This app lets students and staff members report lost items around campus and uses location access to make it easy for users to find lost items.

1.2.3. Novelty

N/A

1.3. Product functionality

The app must allow users to:

- 1.3.1. Register and log in using their Wentworth Microsoft account.
- 1.3.2. Report a lost item with category, description, image, and location.
- 1.3.3. Report a found item with category, description, image, and location.
- 1.3.4. Have access to a list of approved lost and found item reports.
- 1.3.5. View and change their reports.
- 1.3.6. Get notified when a lost item is found.

The app must give administrators the ability to:

- 1.3.7. View, approve, and remove item reports.

1.4. Definitions

Administrator: Members of the project group shall be system administrators and have access to more app functionalities, until the program is completed and handed off to the client. The client will provide a list of email addresses to be designated as administrators.

User: A student or staff member of Wentworth Institute of Technology.

1.5. Acronyms and abbreviations

IDE: Integrated Development Environment

UI: User Interface

2. System requirements

2.1. Functional requirements

2.1.1 The system shall allow users to register an account using their Wentworth ID.

2.1.2 The system shall allow all registered users access to the app's services.

2.1.3 The system shall ask if the user wants to report a found or lost item.

2.1.4 The system shall allow the user to report a lost item. The user must complete the following subsections in order to report the item as lost:

2.1.4.1 Category (e.g., backpack, coat, laptop)

2.1.4.2 Description

2.1.4.2.1 The system shall require users to write a basic description of the item.

2.1.4.2.2 The system shall allow users to optionally upload an image of their lost item.

2.1.4.3 Last known location

2.1.4.3.1 The system shall allow users to mark the last known location with a pin, using their device's GPS.

2.1.4.3.1 Users shall be able to select the building in which their lost item was last seen.

2.1.4.4 Owner's Contact Information

2.1.4.4.1 Name

2.1.4.4.2 Phone Number

2.1.5 The UI shall display a search bar, for users to search through the list of found items.

2.2. Non-functional requirements

2.2.1 The system shall restrict access to users with a valid Wentworth ID.

2.2.2 The report form shall include labeled fields for category, description and location.

2.2.3 Lost and found item reports that are approved by an admin shall be visible to users in under 1 minute.

2.2.4 Previously found items shall be saved into a database.

2.2.5 User data and contact information shall only be visible to the item reporter and system administrators.

2.2.6 The database shall have the capacity to store a minimum of 5000 lost and found item reports.

2.2.7 The system shall allow users to view and delete their own reports.

2.2.8 The system shall send an email notification to a user when their reported lost item gets reported as found by another user.

2.3. Other requirements

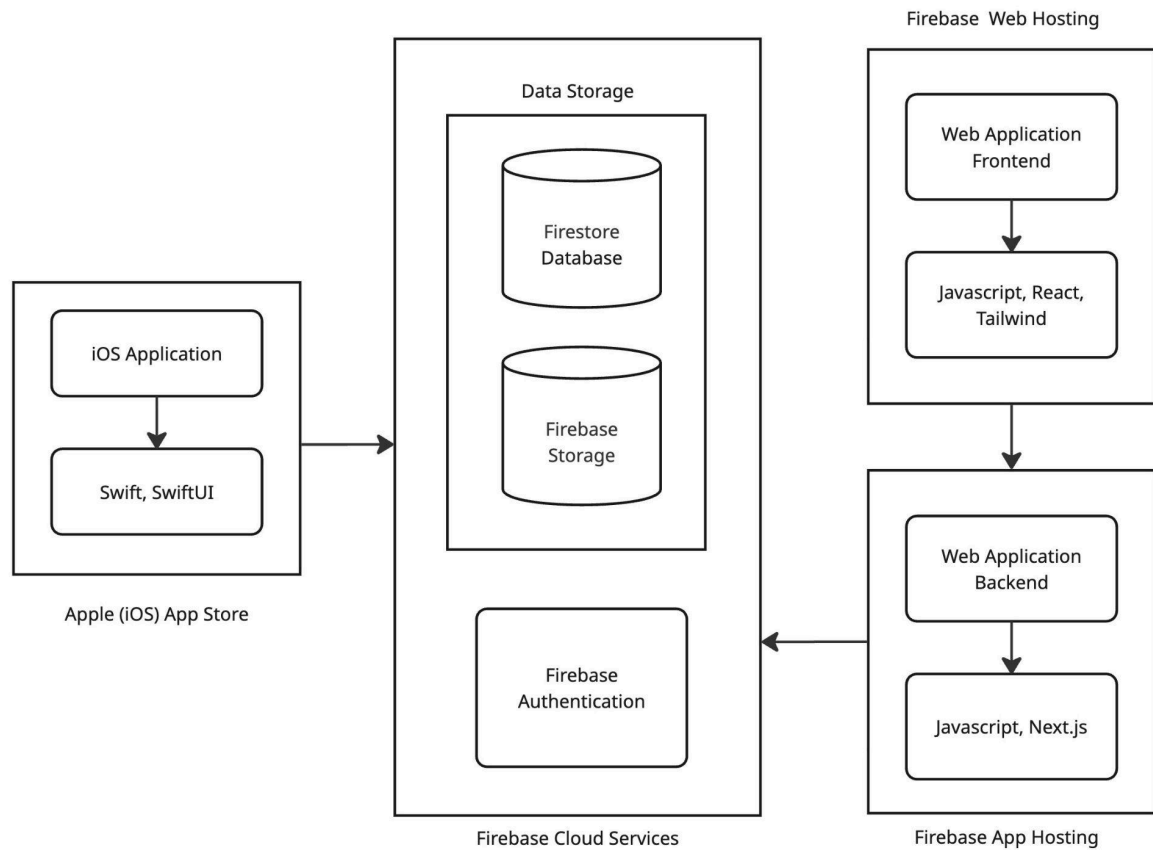
2.3.1 The system shall be built using Swift and Swift UI for the mobile application.

2.3.2 The system shall be built using Node.js/Express backend with a Firebase database for the web app.

2.3.3 The system shall have a separate system view for admins.

3. System architecture

3.1. Overall architecture



3.2. Components mapping

3.2.1. Functional requirements

Functional requirements	Mapped components
User registration via Wentworth Microsoft account.	Firebase Authentication
Search bar actions for users to search through the list of found items.	React/SwiftUI, Firestore Database
View, edit or delete options for user reports.	React/SwiftUI, Firestore Database
Report a lost or found item(category, description, location, contact information, and optional image).	React/SwiftUI, Firestore Database, Firebase Storage

3.2.2. Non-functional requirements

Non-functional requirements	Mapped components
Access restricted to only @wit.edu accounts.	Firebase Authentication
All reports shall be saved in the database.	Firestore Database
All reports shall require item description, category, and location.	React/SwiftUI
Lost and found reports shall be approved by an administrator, before they are displayed in the app.	Firestore Database, Admin Dashboard

3.3. Technology stack selection.

- Visual Studio Code - main editor used to design the web application front and back end.
- Tailwind - a third party framework to help with the design of the application.
- React - a javascript library for building UI.
- Firestore Database - for storing text-based application data
- XCode - Apple's IDE for building apps on all Apple platforms.
- GitHub - allows version control for collaboration between team members.

4. System Design

4.1. UI

4.1.1. User Creation

4.1.1.1. A button that prompts the user to create an account with firebase

4.1.2. Report an Item Lost or Found

4.1.2.1. A found button that brings the user to a page with found items

4.1.2.1.1. Clicking on an image/item brings user to a claim form

Description: user types in description of when and where they lost their item

Submit Button: sends the form to the user that found said item

4.1.2.2. Search bar

Filter Dropdowns: allows the user to filter through category, location, date, possibly time

Key Word Match: input of search bar would show possible keyword matches of items

Search Button: performs the search action

4.1.2.3. A lost button that brings the user to a page with a form:

Location: Place Lost

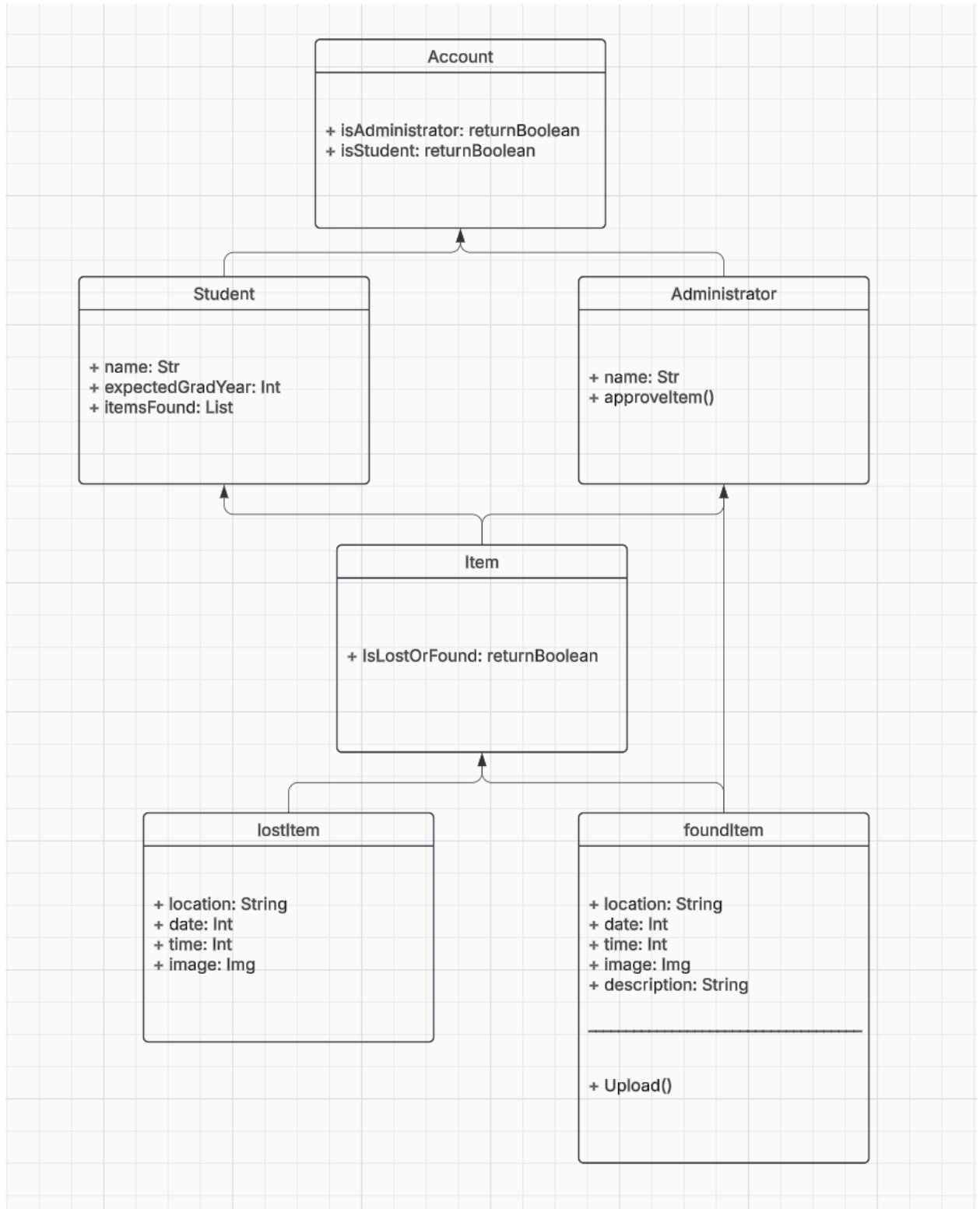
Date/Time: Approximate Time and Date Lost

Image Upload: Image of item

Item Details: Extra Item Details

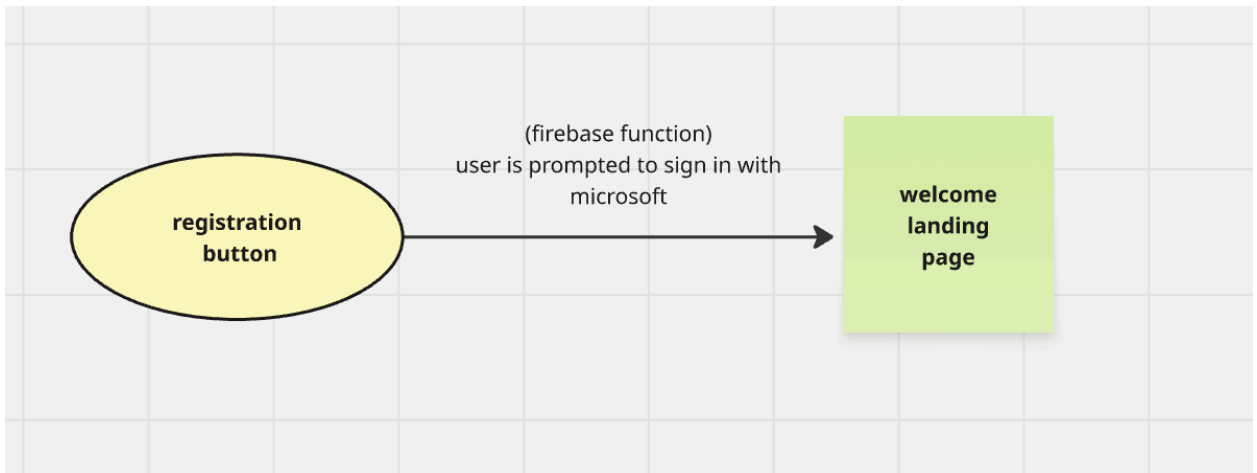
Submit Button: Submits the form.

4.2. Class diagram

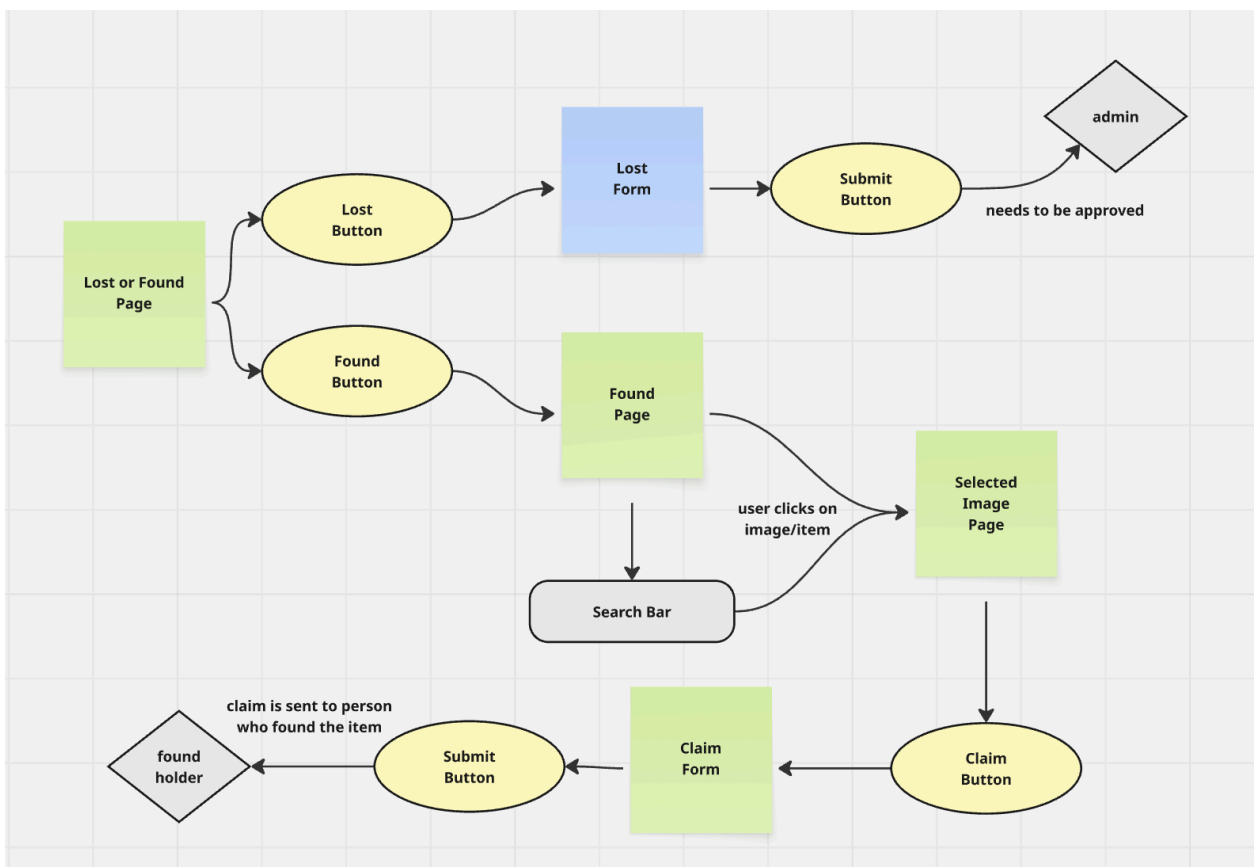


4.3. Sequence/activity diagram

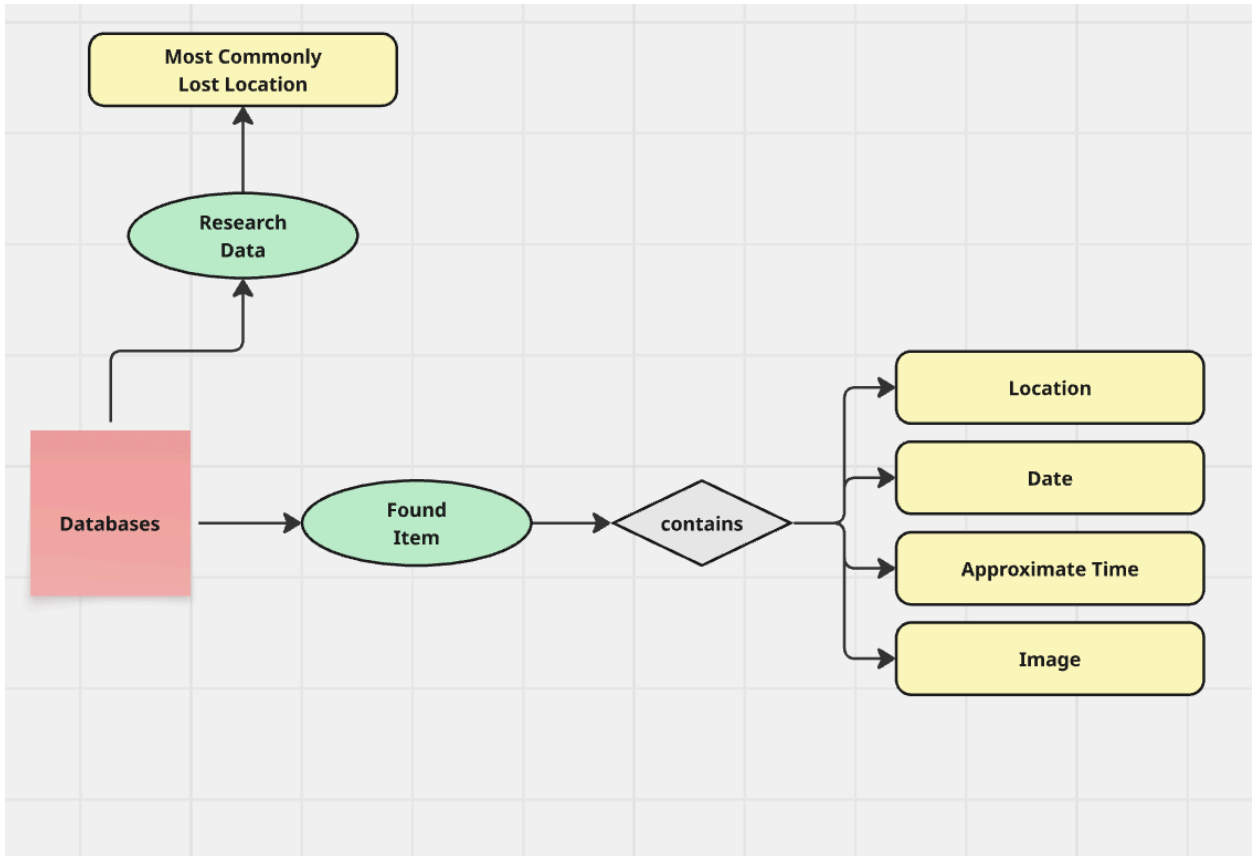
4.3.1. User Creation



4.3.2. Report an Item Lost or Found



4.4. Database



5. Others

Design Applications

- Miro
- Figma
- Lucid

6. Test plan

No.	Test case	User input	Pass criteria
1	Successful login	Valid Wentworth Microsoft account and passes Microsoft Security check.	The user is sent to the main view of the app.
2	Failed login	Invalid Wentworth Microsoft account or Security check incomplete.	An error is displayed and the user gets sent back to the app login view.

3	Item report request accepted	Item report that completes all required details for a report.	Report displays in user account reports and needs to be approved by an admin to show up on app reports page.
4	Item report request rejected	Item report that has incomplete details for a report.	The user screen shows error and sends the user back to their report page, where they are shown where their report is missing information.
5	Search Report by keyword	The user enters the search menu and types out a keyword or chooses one of the category options.	The user is sent to a new page that displays items related to the keyword or category chosen. If no changes to search are submitted, then the user is displayed a view of the most recent reports.
6	Image upload	In the item report the user tries uploading an image for their report.	If the image file has the right format (.jpg) then the image is approved and saved to the item report. If the image is not of the right format then the image is rejected and so is not saved in the user report.
7	Email notification	Users either complete an item report, claim an existing item report, or their item report is claimed by someone else.	The user is sent an email to their @wit.edu account that lets them know of the events that have happened related to their item reports.
8	Location request	The user tries to submit	Users are displayed a

		their location for an item report.	pop-up that if they accept will take their location from their device, if declined, the user can then type a location or building where the item is found.
--	--	------------------------------------	--

6.1. App Functionality

- 6.1.1. Microsoft authentication with @wit.edu.
- 6.1.2. Create an item report for either a lost or found item and list the following item applicable details (category, description, image and location).
- 6.1.3. Search approved reports by category or keyword. (maybe building if location integration makes it possible.)
- 6.1.4. Display items from most recent in the item search view.
- 6.1.5. Email notification when an item's status changes.

6.2. Administrator Functionality

- 6.2.1. View, approve, and remove item reports.
- 6.2.2. User view access and Admin view access.

6.3. User Functionality

- 6.3.1. Check, edit, or delete personal reports.
- 6.3.2. Allow to mark reports as resolved when the item is claimed.
- 6.3.3. Update contact information - none required.

6.4. Report Functionality

- 6.4.1. Location request or location typed out.
- 6.4.2. Require description, location and title for item report.
- 6.4.3. Image upload function.

7. References

Not Applicable.