Wentworth Institute of Technology

COMP4960 – Software Engineering

Instructor: Dr. Koorosh Firouzbakht

Project Final Report

for

Campus Lost and Found Portal

Version 2.0

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

12/05/2025

This page was intentionally left blank.

Executive Summary

The Campus Lost and Found Portal was developed to address the ongoing challenge of lost personal items across the Wentworth Institute of Technology campus. We have created a unified, intuitive, and efficient platform that simplifies this process through both an iOS application and a Web application, ensuring accessibility for a wide range of users.

The completed system allows users to authenticate securely using their @wit.edu Microsoft accounts, ensuring that only members of the Wentworth community can access the platform. Once logged in, users are able to create detailed lost or found item reports, including item descriptions, category selection, contact information, and optional image uploads. These submissions are stored and synchronized through Firebase Firestore and Firebase Storage, providing real-time data updates across platforms. Users can also view recently submitted reports, search through existing entries using simple filtering and browsing tools, and manage their own submissions within the interface. On the Web platform, users have the added ability to mark their reports as resolved once an item has been recovered.

In addition to user functionality, the system incorporates administrative controls designed to maintain the accuracy and integrity of reported information. Administrators have the ability to review newly submitted reports and choose to approve or deny them before they appear in the public-facing list of lost and found items. This layer of oversight ensures quality control and creates a trustworthy environment in which users can confidently rely on the information presented.

The project was built using a modern and industry-relevant technology stack, including SwiftUI for iOS, React for the Web front end, Firebase Authentication for secure login, Firestore for scalable cloud database storage, and Firebase Storage for image handling. These technologies enabled the team to design a solution that is both responsive and maintainable, while also offering a strong foundation for future development. The platform operates as a cohesive minimum viable product (MVP) that successfully demonstrates the core functionality required for a campus-wide lost and found solution.

Overall, the Campus Lost and Found Portal represents a meaningful step toward improving item recovery and communication across the Wentworth community. The completed work can be established as a solid base that future student teams, campus IT staff, or administrative departments can expand upon to include additional features, enhanced automation, and deeper integration with campus systems.

1. Contents

2. Revision History

Date	Version	Description	Author(s)
11/13/2025	1.0	Initial version - (Parts 9 and on left for later versions when product is complete)	Bjordi Ismailati Craig Bakke
11/21/2025	1.1	1. Removed all remaining placeholder text. 2. Marked the backside of the cover page as intentionally blank. 3. Edited section 3.1. 4. Corrected the page numbers listed in section 1.	Craig Bakke
12/01/2025	1.2	1. Re-phrased functional requirements (Section 4.1) to better conform with the RTM format.	Craig Bakke
12/05/2025	2.0	1. Added Executive Summary 2. Added Sections 9, 10, 11, 12, 13, 14, 15, and 16. 3. Updated the page numbers listed in section 1.	Aliyan Hidayatallah Craig Bakke

3. Introduction

3.1. Document purpose

The purpose of this document is to provide a final overview of the process that was undertaken in order to create the Campus Lost and Found Portal application. It provides an overview of the problem and solution, outlines the software requirements and design, and describes the technology stack and final testing steps taken before the project's completion and handoff to the client.

3.2. Product overview

3.2.1. 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.

3.2.2. 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.

3.2.3. Novelty

N/A.

3.3. Product functionality

The app shall allow users to:

- 3.3.1. Register and log in using their Wentworth Microsoft account.
- 3.3.2. Report a lost item with category, description, image, and location.
- 3.3.3. Report a found item with category, description, image, and location.
- 3.3.4. Have access to a list of approved lost and found item reports.
- 3.3.5. View and change their reports.
- 3.3.6. Get notified when a lost item is found.

The app shall give administrators the ability to:

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

3.4. Definitions

3.4.1. 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.

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

3.5. Acronyms and abbreviations

3.5.1. IDE: Integrated Development Environment

3.5.2. UI: User Interface

3.5.3. ID: Identification Number or Card

3.5.4. GPS: Global Positioning System

3.5.5. GCP: Google Cloud Platform

3.5.6. JS: JavaScript

4. System requirements

4.1. Functional requirements

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

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

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

4.1.4. The system shall allow the user to report a lost item. The user must report the item category, description, last known location, and their contact information.

4.1.4.1. The system shall require the user to input the item's category (e.g., backpack, coat, laptop).

4.1.4.2. The system shall require the user to report a description of the item.

4.1.4.2.1. The system shall require the user to to write a basic text description of the item.

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

4.1.4.3. The system shall require the user to report the last known location of their item.

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

4.1.4.3.2. The system shall allow users to optionally select the building in which their lost item was last seen.

4.1.4.4. The system shall require the user's contact information.

4.1.4.4.1. The system shall require the user's name.

4.1.4.4.2. The system shall require the user's phone number.

4.1.5. The system's user interface shall display a search bar, for users to search through the list of found items.

4.2. Non-functional requirements

4.2.1. The system shall restrict access to users with a valid Wentworth account.

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

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

4.2.4. Previously found items shall be saved into a database.

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

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

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

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

4.3. Other requirements

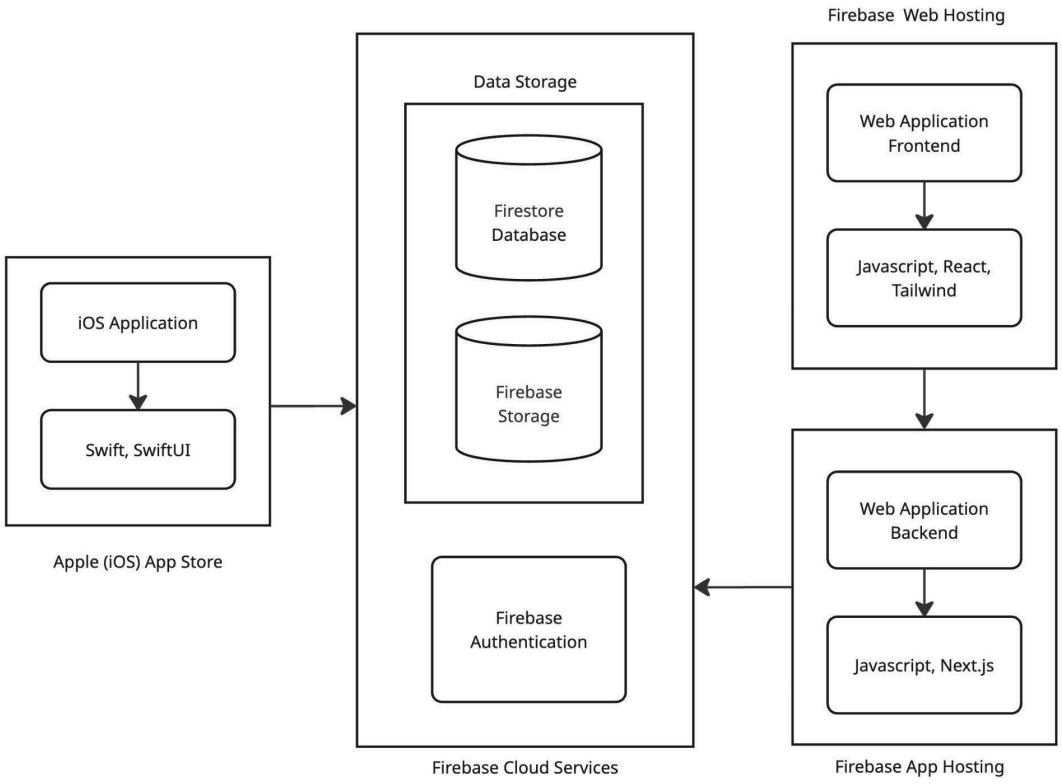
4.3.1. The system shall be built using Swift and Swift UI for the mobile application.

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

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

5. System architecture

5.1. Overall architecture



5.2. Components mapping

5.2.1. 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

5.2.2. 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

5.3. Technology stack selection

Technology	**Purpose**
Firestore Database	Stores user reports and other general text-based app data.
Firebase Storage	Stores user-submitted images attached to lost/found reports.
Firebase Auth	Manages user sign-in, restricted to @wit.edu Microsoft accounts.
Firebase Hosting	Hosts the frontend and backend of the web application.
Javascript	Programming language that easily integrates required libraries for both frontend and backend.
Tailwind	CSS framework for rapidly styling the web application's frontend user interface.
React	JavaScript library for building the web app’s interactive UI components and dynamic pages.
Next.js	Framework used for server-side logic, routing, and acting as the API middleman
Visual Studio Code	Development environment used to build the web application’s frontend and backend.
Xcode	Development environment for developing, debugging, and compiling the iOS

	application.
Swift/SwiftUI	Programming language and native UI kit for building the iOS app.
Github and Jira	Development and collaboration tools used for tracking project progress.
Figma	Design tools used for UI mockups and visual diagrams like flowcharts and architecture.

6. System Design

6.1. UI

6.1.1. User Creation

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

6.1.2. Report an Item Lost or Found

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

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

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

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

6.1.2.2. Search bar

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

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

6.1.2.2.3. Search Button: performs the search action

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

6.1.2.3.1. Location: Place Lost

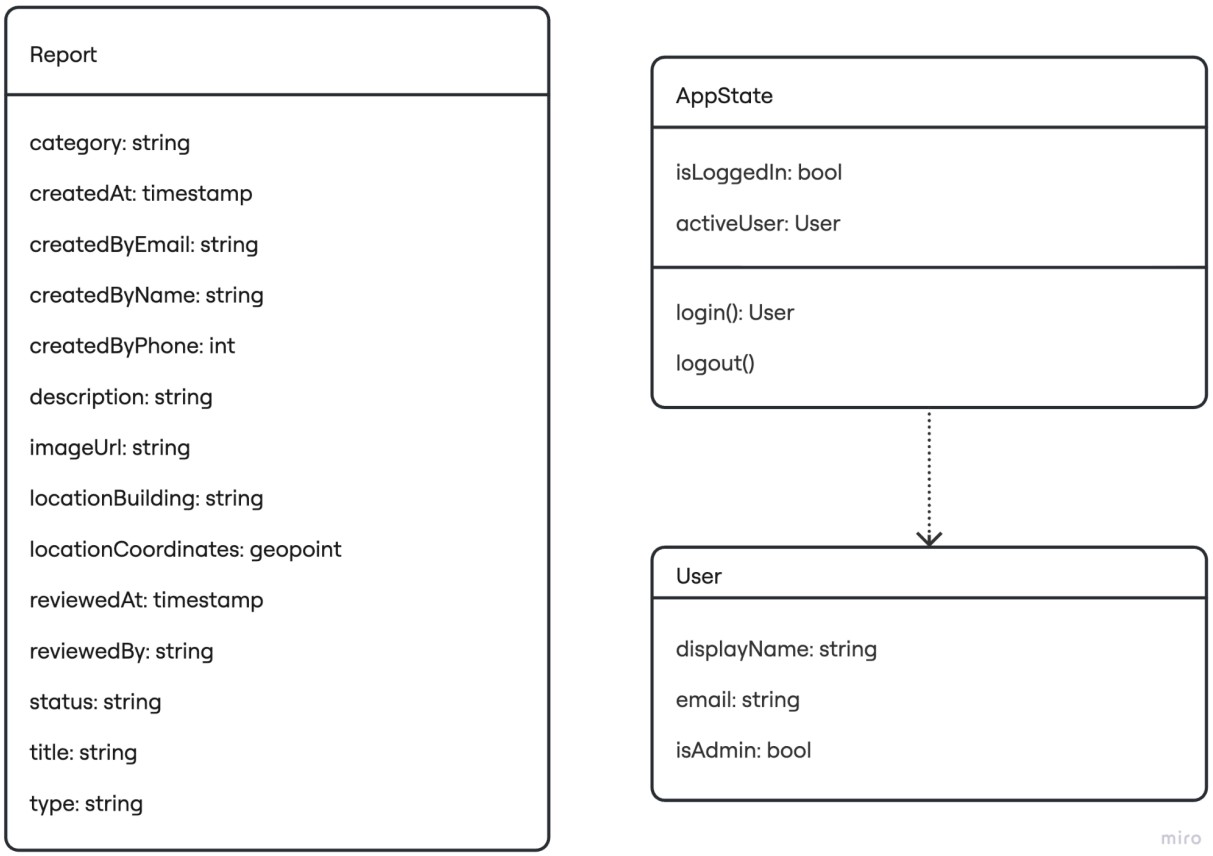
6.1.2.3.2. Date/Time: Approximate Time and Date Lost

6.1.2.3.3. Image Upload: Image of item

6.1.2.3.4. Item Details: Extra Item Details

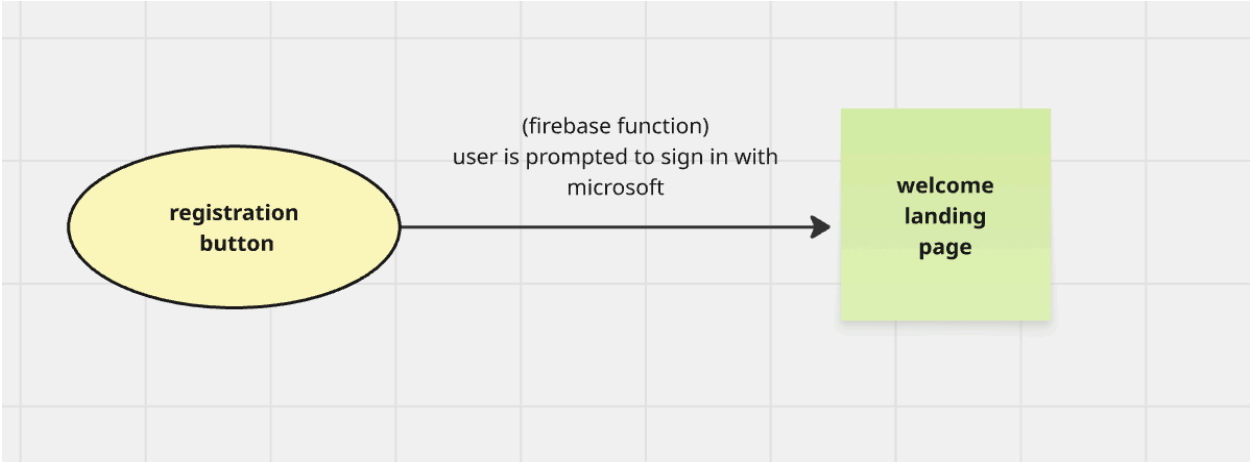
6.1.2.3.5. Submit Button: Submits the form.

6.2. Class diagram

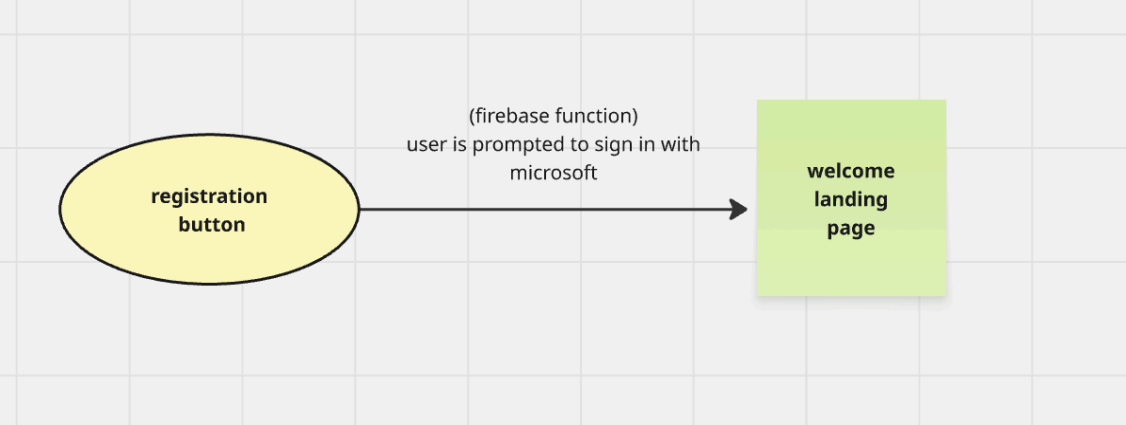


6.3. Sequence/activity diagram

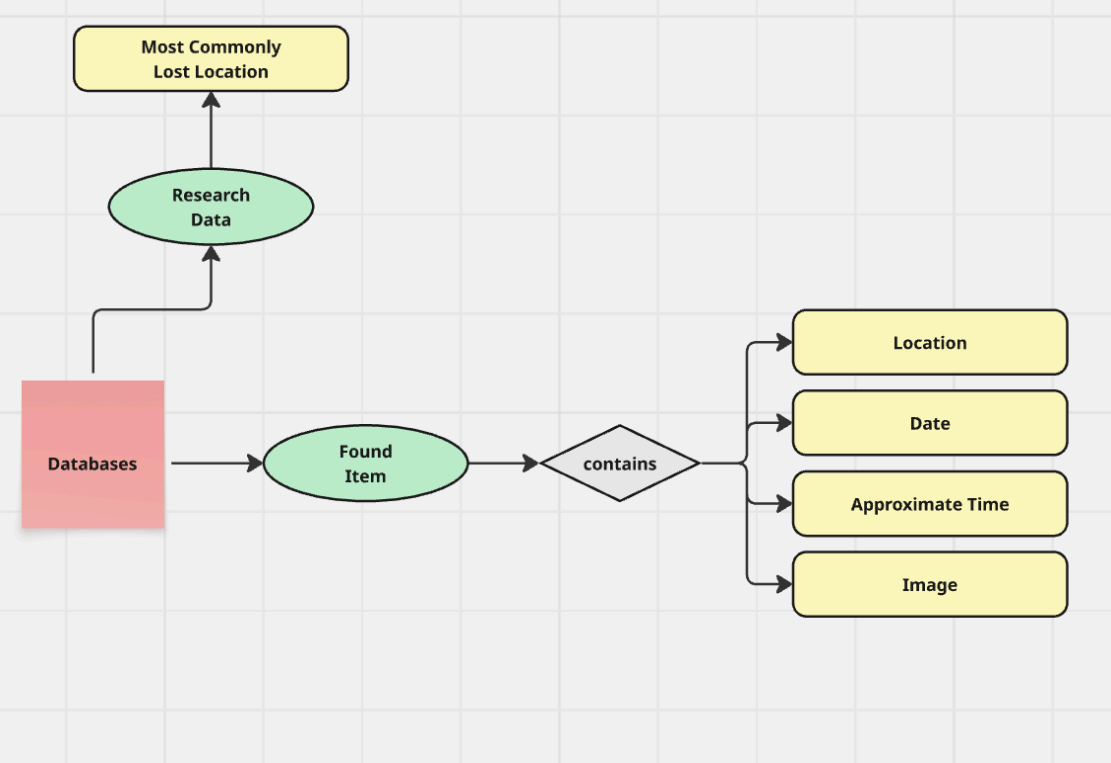
6.3.1. User Creation



6.3.2. Report an Item Lost or Found



6.4. Database



7. Others

Design Applications

- Miro
- Figma
- Lucid

8. 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 the 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 their location for an item report.	Users are displayed a 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.

8.1. App Functionality

8.1.1. Microsoft authentication with @wit.edu.

8.1.2. Create an item report for either a lost or found item and list the following item applicable details (category, description, image and location).

8.1.3. Search approved reports by category or keyword. (maybe building if location integration makes it possible.)

8.1.4. Display items from most recent in the item search view.

8.1.5. Email notification when an item's status changes.

8.2. Administrator Functionality

8.2.1. View, approve, and remove item reports.

8.2.2. User view access and Admin view access.

8.3. User Functionality

8.3.1. Check, edit, or delete personal reports.

8.3.2. Allow to mark reports as resolved when the item is claimed.

8.3.3. Update contact information - none required.

8.4. Report Functionality

8.4.1. Location request or location typed out.

8.4.2. Require description, location and title for item report.

8.4.3. Image upload function.

9. Delivery Metrics

Out of the 26 stated requirements, 24 have been fulfilled. Defect density is very low, as is test coverage. All 24 of the fulfilled requirements were delivered during the third sprint, because the first two sprints were entirely used for planning and preparation, not implementation. It took the team 4 weeks to deliver 24 features, resulting in an average of 6 new features per week.

10. Release Summary

The final release of the Campus Lost and Found Portal includes the completed minimum viable product (MVP) for both the iOS and Web applications. This release delivers the core functionality required for students and staff to submit, browse, and manage lost and found item reports. Major features included in this release are Microsoft authentication, report submission with image upload, Firestore database integration, admin approval tools, search functionality, display of recent reports, and user-controlled resolution of items on the Web platform.

This release reflects the combined development efforts across the semester and represents a stable, functional system that demonstrates all essential workflows of the platform. It serves as the baseline version for future enhancements and potential campus-wide adoption.

11. Performance Metrics

11.1. Reports that are approved by administrators consistently become visible to other users within 1 second. This exceeds the expectations set by requirement 4.2.3, which states that reports will be visible within 1 minute of approval.

11.2. The database is capable of storing in excess of 50,000 item reports. This exceeds the expectations set by requirement 4.2.6, which states that the database must be capable of storing at least 5,000 item reports.

11.3. Users consistently see database API response times of less than 200 ms.

11.4. Due to GCP's ability to automatically assign resources, the maximum throughput (requests per second) scales as more is needed, to a maximum of 50,000 requests per day.

12. Defect Management and Known Issues

No defects were reported nor were any resolved during the reporting period. There are no currently known defects with the program.

13. Test Summary

Testing primarily consisted of iterative manual testing performed by the development team across both platforms. Tests included verification of login authentication, report submission, image upload, admin approval, browsing of recent reports, and search functionality.

Each major feature was validated to ensure correct database write/read operations, proper synchronization across platforms, and smooth visual rendering in the UI. Testing confirmed that the core workflows of the MVP function as expected and that the system remains stable across supported environments.

14. Customer/Stakeholder Feedback

14.1. One customer suggested we work more closely with the Wentworth Police to better integrate with the various physical “lost and found” locations around campus.

15. Lessons Learned, Recommendations and Future Work

During the development of the Campus Lost and Found Portal, we gained valuable experience in cross-platform development, database design, and working with cloud services such as Firebase Authentication, Firestore, and Storage. We also learned how important communication and coordination are when building parallel features across iOS and Web, as well as the need to plan ahead for external dependencies, such as obtaining approval for Microsoft OAuth authentication through Wentworth’s NSS department.

Based on our experience, we recommend that future teams define the data model and user workflows early, maintain consistent naming and structure across platforms, and allocate additional time for testing and UI refinement. Clear documentation and regular check-ins helped our team stay aligned, and future teams would benefit from continuing these practices.

For future work, several enhancements could significantly expand the system’s capabilities, including adding AI-based image matching, implementing user messaging, improving claim verification, integrating location-based reporting, and expanding the admin dashboard. These additions would build on the strong foundation established by the current MVP and help create a more complete and seamless lost and found experience for the Wentworth community.

16. Conclusions

The Campus Lost and Found Portal project successfully demonstrates the design and implementation of a functional cross-platform system built for the Wentworth community. Through the development of both an iOS and a Web application, we delivered a cohesive and intuitive solution that streamlines the reporting and recovery of lost items across campus. The system integrates secure Microsoft authentication, cloud-based data storage, image handling, user reporting features, and administrative review tools, all of which contribute to a reliable and user-friendly experience.

This project highlights the ability to apply software engineering principles, collaborate effectively, and utilize modern technologies such as SwiftUI, React, Firebase Authentication, Firestore, and Firebase Storage. The resulting minimum viable product provides a strong foundation for future enhancements and demonstrates the potential for real-world adoption within the university.

Overall, the project achieved its primary goal of creating a centralized, accessible, and scalable lost and found platform. The work completed serves as a meaningful step toward improving campus communication and item recovery, while also offering a roadmap for future development and continued improvement by subsequent teams or university departments.

17. References

N/A.