LOSTNFOUND KMUTT: KEEP A CLOSE EYE ON YOUR ITEMS.

MR.PASSAKORN PUTTAMA

MR.VATCHARAMAI RODRING

MR.PATTHADOL RAKSAPRAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

SCHOOL OF INFORMATION TECHNOLOGY

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI

2024

LOSTNFOUND KMUTT: KEEP A CLOSE EYE ON YOUR ITEMS.

MR.PASSAKORN PUTTAMA

MR.VATCHARAMAI RODRING

MR.PATTHADOL RAKSAPRAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

SCHOOL OF INFORMATION TECHNOLOGY

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI

2024

LOSTNFOUND KMUTT: KEEP A CLOSE EYE ON YOUR ITEMS.

MR.PASSAKORN PUTTAMA

MR.VATCHARAMAI RODRING

MR.PATTHADOL RAKSAPRAM

A Thesis Submitted in Partial Fulfillment of the Requirements for

The Degree of Bachelor of Science (Computer Science)

School of Information Technology

King Mongkut's University of Technology Thonburi

Academic Year 2024

Thesis Committee

..... Advisor

(Asst.Prof. Worarat Krathu Ph.D)

..... Member

(Dr. Watanyoo Suksa-ngaim Ph.D)

..... Member

(Dr. Vithida Chongsuphajaisiddhi Ph.D)

Thesis Title LOSTNFOUND KMUTT: KEEP A CLOSE EYE ON YOUR ITEMS.
Thesis Credits 6
Candidate MR.PASSAKORN PUTTAMA
MR.VATCHARAMAI RODRING
MR.PATTHADOL RAKSAPRAM
Thesis Advisors Asst.Prof. Worarat Krathu Ph.D
Program Bachelor of Science
Field of Study Computer Science
Faculty School of Information Technology
Academic Year 2024

Abstract

LostNFound KMUTT is a web-based platform designed to assist KMUTT students and university personnel in efficiently managing lost and found items within the university. The platform allows users to report lost or found items through dedicated forms, providing item descriptions, locations, and timestamps for precise matching. It utilizes advanced technologies such as Natural Language Processing (NLP) for text similarity matching and secure authentication via university email to ensure rightful ownership. Additionally, it features a history of claimed items for accountability and a notification system to remind users to retrieve their belongings promptly.

The development process demonstrates the feasibility of creating an efficient lost-and-found management system using existing university resources and modern web technologies. By streamlining the process, the platform minimizes user frustration and enhances operational productivity for KMUTT staff.

Keywords : Lost and Found, KMUTT, Web Platform, Item Matching, Secure Claims

Acknowledgement

We would like to express our heartfelt gratitude to the individuals whose guidance and support made this project possible. Our deepest appreciation goes to our generous advisor Asst. Prof. Dr. Worarat Krathu, provided invaluable insights and mentorship throughout the development of the LostNFound KMUTT platform.

We also extend our special thanks to our special supporters, including Mr.Ukrit Ruckcharti, Mr.Sukanya Chinwicha, and Mr. Thanatat Wongabut for sacrificing their time for additional consulting. Lastly, we appreciate the support of the KMUTT community, whose feedback and encouragement have been instrumental in shaping this project.

Contents

Contents(CONT.)

List of Tables

Table **PAGE**

List of Figures

Chapter 1

Introduction

1.1 Background

Losing personal belongings within a university environment is a common challenge, and KMUTT is no exception. Currently, the recovery process involves contacting security personnel in different areas, which can be inefficient and inconvenient for both students and staff. This lack of a centralized system often leads to prolonged delays in retrieving lost items and potential frustrations for the involved parties.

To address these issues, the LostNFound KMUTT platform aims to streamline and enhance the lost-and-found process through a centralized web-based solution. By utilizing modern web technologies and intelligent text-matching models, this platform simplifies reporting, cataloging, and locating lost items.

1.2 Objective

To create a user-friendly and efficient platform that facilitates the reporting and retrieval of lost items. By integrating modern technology into the existing process, we aim to reduce the complexities associated with lost and found items, making it easier for both users and officers to manage and locate belongings promptly.

1.3 Expected Benefits

1. **Efficiency Improvement:** Reduced processing time for reporting, cataloging, and retrieving lost items.
2. **User Convenience:** Accessible information through a user-friendly interface.
3. **Enhanced Security:** Robust identity verification to ensure rightful ownership.
4. **Reduced Frustration:** Simplified processes for users and staff.
5. **Resource Optimization:** Improved productivity for staff by automating processes.
6. **Timely Notifications:** Notifications to remind users to pick up their items.
7. **Documentation and Accountability:** Comprehensive records of claimed items for auditing.

1.4 Scope

1.4.1 Target

King Mongkut's University of Technology Thonburi (KMUTT).

1.4.2 Scope of Work

1. Define Project Features:

    - **Distinct Features:**
        (a) Item Found & Lost Form: Allow users who found or lost items to report them, providing the time and location where it was found and describing the item to enable matching.
            – Found Item Form.
            – Lost Item Form.
        (b) Claim Process: Allow users to claim found items after the form is matched.
            – Text comparison using Natural Language Processing (NLP).
            – Notification system.
        (c) Secure Items Ownership: Prevent unauthorized claims using university email authentication.
    - **Support Features:**
        (a) Authentication using OAuth with Microsoft authentication.
        (b) History of Claimed Items.

2. Architecture and Technology Stack:

    - **Frontend:** Next.js 14.
    - **Backend:** Next.js 14 and FastAPI.
    - **Database:** PostgreSQL.
    - **Authentication:** NextAuth.js with Microsoft as the provider.
    - **Infrastructure:** Google Cloud (GCP), Docker.
    - **Model:** Public text similarity and NER model from Hugging Face.
    - **Testing:** Jest (Unit testing), Postman (API testing).

Chapter 2

Feasibility Study

2.1 Introduction

The LostNFound platform is a centralized web application designed to address the challenges associated with managing lost-and-found items at King Mongkut's University of Technology Thonburi (KMUTT). The platform aims to provide students, staff, and university personnel with an efficient, accessible, and secure solution for reporting and reclaiming lost items.

Unlike the current manual and on-site processes, the LostNFound platform digitizes the entire lost-and-found workflow. Users can report lost items, search for them, and claim ownership through the platform, reducing time and effort. It also introduces features such as real-time item status updates, secure ownership verification, and centralized database management. This initiative aligns with KMUTT's commitment to modernizing student services and fostering a more convenient university experience.

2.2 Problem Statement

Currently, KMUTT's lost-and-found system relies on physical reporting to either nearby security personnel or the Student Affairs Division. This approach is inefficient, time-consuming, and prone to security risks such as unauthorized claims. The lack of a unified system makes tracking and managing lost items cumbersome for both users and staff. Furthermore, the absence of a centralized database means there is no effective way to ensure transparency or accessibility for all stakeholders.

The LostNFound platform aims to resolve these issues by providing a digital solution. The platform will centralize all lost-and-found operations, improve communication between stakeholders, and reduce the time required to locate and reclaim lost items. By introducing advanced features such as user profiles, secure claims verification, and automated notifications, the system will create a seamless experience for KMUTT's community.

2.3 Related Research and Projects

2.3.1 Lifeguard Lost & Found

A system designed to handle lost and found items in public spaces like airports and transport hubs, which shares similarities with the university setting. It allows users to report, search for, and claim items, ensuring secure ownership verification.

2.3.2 MissingX

A centralized platform for lost and found items, focusing on automatic matching of lost and found property. The platform also supports user notifications when a match is made, aligning

with this system's goal of real-time updates and seamless matching.

2.3.3 ItemFinder

Designed specifically for institutions like universities, ItemFinder handles lost and found items by providing a centralized platform for reporting, cataloging, and verifying items.

2.3.4 Comparison of Existing Functions

Applications	Item Reporting	Matching (NLP)	Notifications	History of Claims	User Profiles
Lifeguard Lost & Found	Yes	No	No	No	No
MissingX	Yes	Yes	Yes	No	No
ItemFinder	Yes	No	No	No	Yes
LostNFound KMUTT	Yes	Yes	Yes	Yes	Yes

Table 2.1: Existing Functions of Related Applications

2.4 Requirement Specifications

2.4.1 Functional Requirements

1. Report lost and found items by filling out a form with details such as item description, location, and time.
2. Provide real-time updates regarding the status of items.
3. Use Natural Language Processing (NLP) to match descriptions of lost and found items.
4. Notify users when a match is found for their lost item or when their claimed item is ready for pickup.
5. Maintain a history of claimed items, including photos and verification details.
6. Integrate with the university's email system (OAuth with Microsoft) for authentication and ownership verification.

2.4.2 Data Requirements

1. **User Information:** Collect and store user details (e.g., phone number, university email, profile information, claim history) for authentication and personalized user experience.
2. **Item Information:** Store details of each lost and found item, including item type, description, location, date, and images.

2.5 Implementation Techniques

2.5.1 Frontend

- Programming Language: TypeScript.
- Framework: Next.js.

2.5.2 Backend

- Programming Language: TypeScript, Python.
- Frameworks: Next.js, FastAPI.

2.5.3 Infrastructure

- OS: Debian Linux.
- Cloud Provider: Google Cloud Platform (GCP).
- Containerization: Docker.
- CI/CD: Google Cloud Build.
- Deployment Platform: Google Cloud Run.

2.5.4 Database

- Database Type: Relational (PostgreSQL).

2.5.5 Testing

- Unit Testing: Jest.
- API Testing: Postman.

2.6 Implementation Plan

No.	Task Name	Duration	Start Date	End Date
Phase 1: Project Initiation				
1	Conduct feasibility study	10 Days	15 July 2024	24 July 2024
2	Define user requirements	10 Days	15 July 2024	24 July 2024
3	Identify technical solutions	10 Days	15 July 2024	24 July 2024
4	Develop project scope and timeline	10 Days	15 July 2024	24 July 2024
Phase 2: System Architecture Design				
1	Define system architecture (Frontend & Backend)	10 Days	25 July 2024	3 August 2024
2	Design database schema (PostgreSQL)	10 Days	25 July 2024	3 August 2024
3	Design user authentication flow (NextAuth.js)	10 Days	25 July 2024	3 August 2024
4	Define text similarity and matching logic (FastAPI)	1 Week	4 August 2024	10 August 2024
Phase 3: Development				
1	Setup project repository & development environment (Docker, GCP)	1 Week	11 August 2024	17 August 2024
2	Develop frontend (Next.js + Tailwind CSS)	2 Weeks	18 August 2024	31 August 2024
3	Develop backend API (FastAPI for matching & item processing)	2 Weeks	18 August 2024	31 August 2024
4	Integrate PostgreSQL database with backend	2 Weeks	4 September 2024	18 September 2024
5	Implement user authentication (OAuth/Microsoft ID)	2 Weeks	4 September 2024	18 September 2024
6	Implement matching logic (NLP, Text Similarity)	2 Weeks	4 September 2024	18 September 2024
7	Setup notifications	3 Days	19 September 2024	21 September 2024
Phase 4: Testing				
1	Conduct unit tests (Frontend and Backend)	1 Week	22 September 2024	28 September 2024
2	Perform integration testing (Frontend + Backend)	4 Weeks	1 October 2024	30 October 2024
Phase 5: Deployment Preparation				
1	Prepare deployment environment on Google Cloud	2 Days	30 October 2024	31 October 2024
2	Prepare deployment environment on Google Cloud	2 Days	1 November 2024	2 November 2024
Phase 6: Deployment				
1	Deploy the application to Google Cloud using Cloud Run	4 Days	4 November 2024	8 November 2024

Table 2.2: Implementation Plan

Chapter 3

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

This chapter provides an explanation of our project design in the following sections: analysis of the existing system, diagram, system architecture, system design, and user interface design.

3.2 Analysis of the existing system

The scope of this project is centered on King Mongkut's University of Technology Thonburi (KMUTT) as the designated target area. The primary user groups are the personnel within the organization.

3.3 User Requirement Analysis

3.3.1 User Requirements

The following are the functional specifications that users expect from the LostNFound KMUTT platform:

1. **Lost and Found Item Reporting:** The system should allow users (students and university staff) to report lost and found items. Users should be able to submit detailed descriptions for the items they are reporting. This information is essential for accurate matching.

2. **Item Matching:** The platform should use advanced text-matching techniques, such as Natural Language Processing (NLP), to compare lost and found item descriptions and ensure the correct matching between them. This allows users to easily identify and claim their belongings.

3. **Secure Claim Process:** The system must verify users' identities using their **university email** to ensure only the rightful owner can claim an item. The platform should reject any claims from users who cannot prove ownership of the lost item.

4. **History of Claimed Items:** The system needs to maintain a detailed record of all claimed items, including the name and ID of the user who claimed the item. This helps provide accountability and keeps a history of item claims for future reference.

5. **Notification System:** Users should receive timely notifications when a match is found for their lost item or when they are reminded to pick up an item they have claimed.

6. **Authentication and Security:** The system must ensure secure authentication using **Microsoft Extra ID** to authenticate users. Only users with valid **KMUTT email accounts** should be able to access the platform and participate in reporting or claiming items.

3.4 System design

3.4.1 Proposed Architecture Diagram

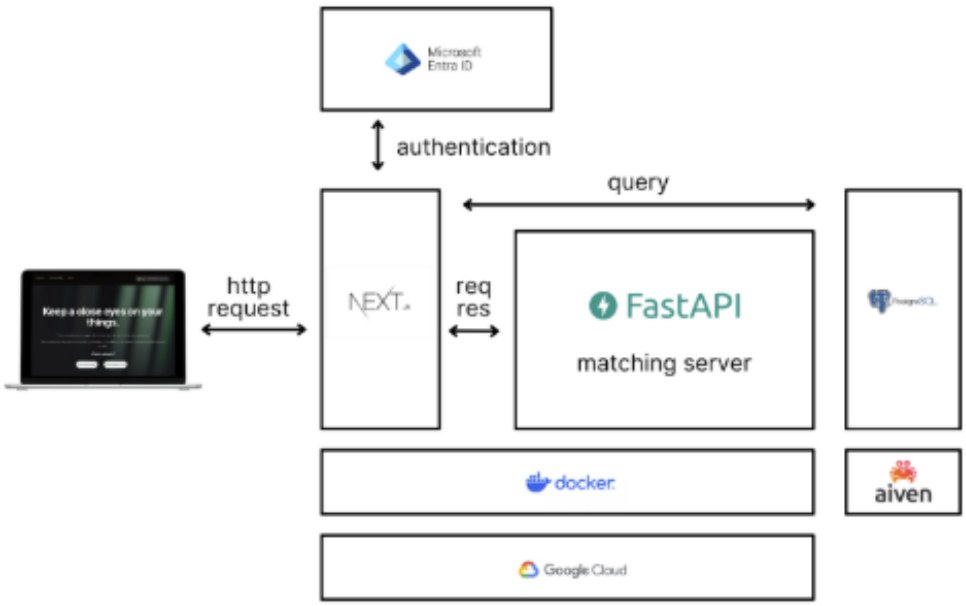

Figure 3.1: Architecture Diagram

Figure 3.1 represents our system architecture. The platform will leverage Google Cloud Platform (GCP) for hosting, with Cloud Registry used for storing Docker images, and Cloud Build for automating the build process. The application will run on Cloud Run, providing a serverless environment for deploying containerized applications. For frontend development, we will use Next.js with Tailwind CSS for responsive and stylish web pages. The backend will be developed with FastAPI, which will expose NER and Text Similarity APIs for matching lost and found items. PostgreSQL will be used for storing data related to users and items, hosted on Aiven for managed database services. Docker will be utilized for containerizing the entire application, enabling efficient deployment and scalability within the cloud environment.

3.4.2 Context Diagram

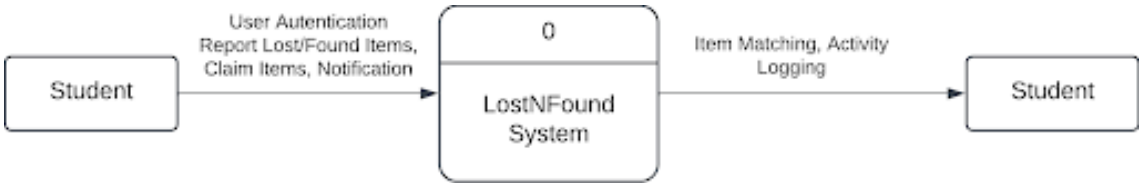

Figure 3.2: Context Diagram of LostNFound System

This figure shows the context diagram for the LostNFound KMUTT system. The diagram illustrates the overall interactions between the users and the system. Users (students and staff) are required to authenticate via their KMUTT email to access the platform. Once authenticated, students can report lost and found items. The core interactions of the system include the submission of item reports, matching items using **Text Similarity and NER APIs**, and the ability for users to track and claim their lost belongings.

3.4.3 Data Flow Diagram

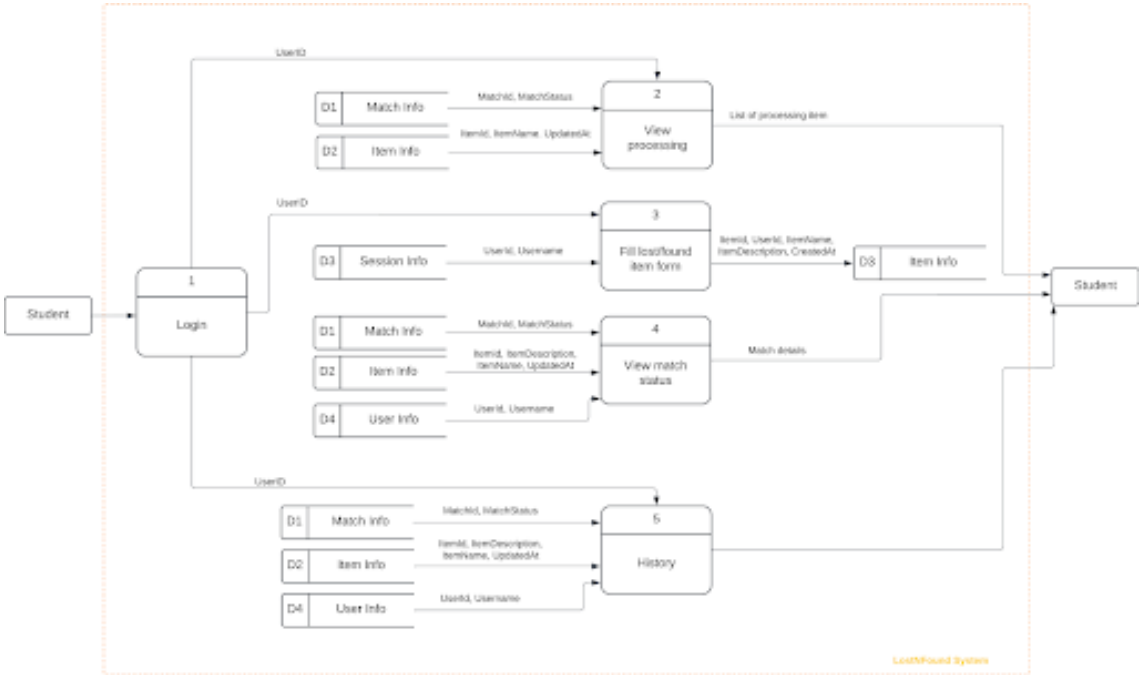

Figure 3.3: Diagram 0

Figure 3.3 represents the data flow diagram of the LostNFound KMUTT system. The process starts when the user inputs their KMUTT email and password to authenticate. Once authenticated, the user can report a lost or found item by providing item details, such as description, location, and timestamp. The system processes and stores the data in the PostgreSQL database. Users can then view item matches using NLP for text similarity, claim found items, and receive notifications when a match is found or when they need to pick up their claimed item.

3.4.4 Activity Diagram

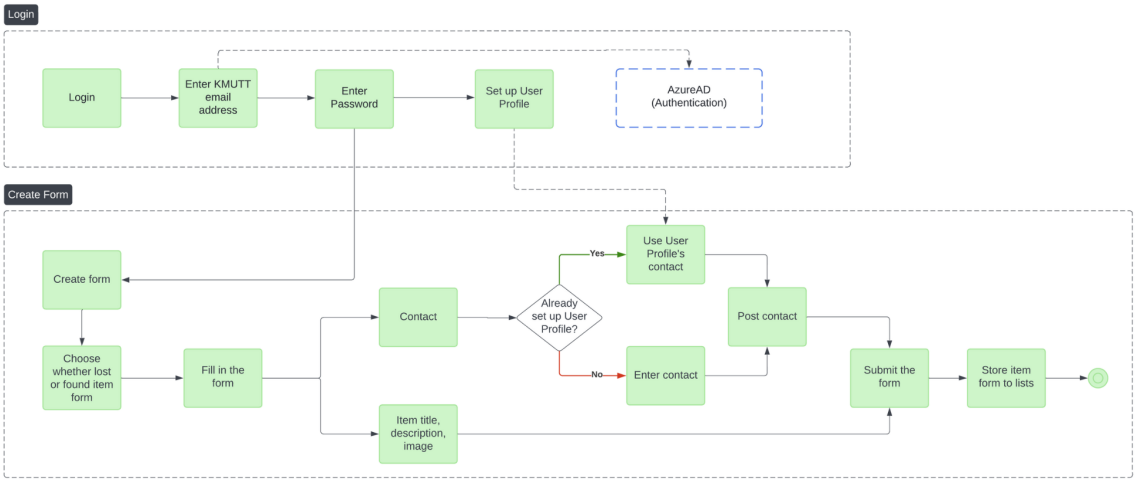

Figure 3.4: Activity Diagram for student logins and creating lost or found item form of LostNFound System

Figure 3-4 represents the flow when the student uses a web application by logging in with Microsoft Extra ID authentication. Once authenticated, the student is directed to the form page where they can choose to create a lost item or found item form. The student fills in the required details, including the item description. After submission, the form data is processed and stored in the system, where it can be used for matching with other lost or found items.

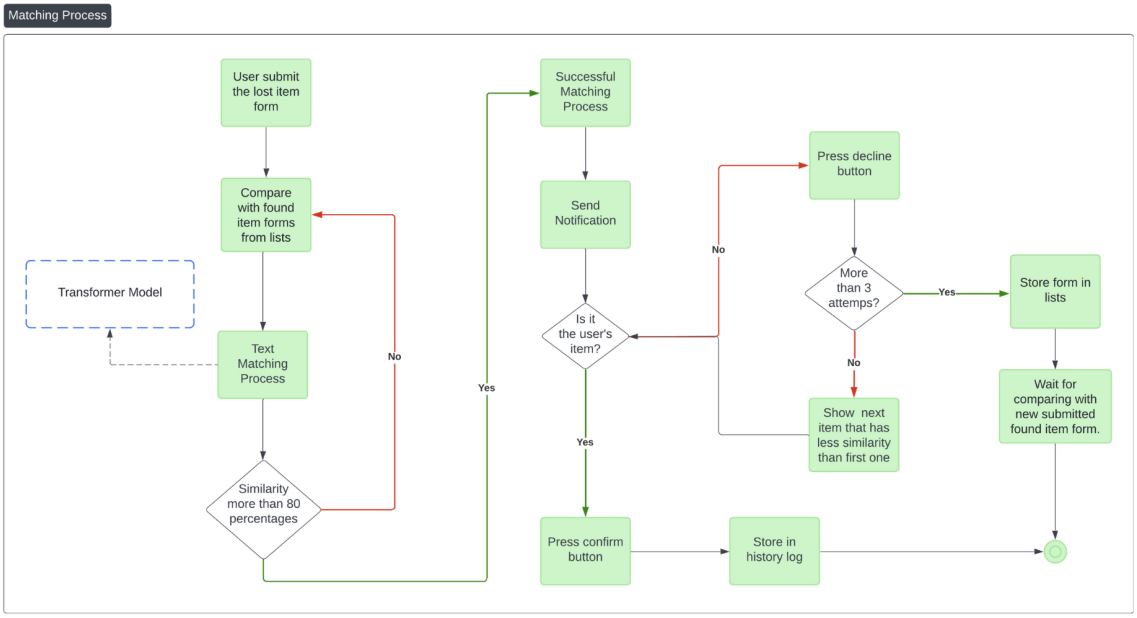

Figure 3.5: Activity Diagram for a matching process of LostNFound System

Figure 3.5 represents the matching process in the LostNFound KMUTT system. The process begins when a lost item report or a found item report is submitted by a student or staff member. Once the report is submitted, the system uses Natural Language Processing (NLP) and Text Similarity algorithms to compare the item descriptions. The system processes key details such as description, product name, and location to identify potential matches.

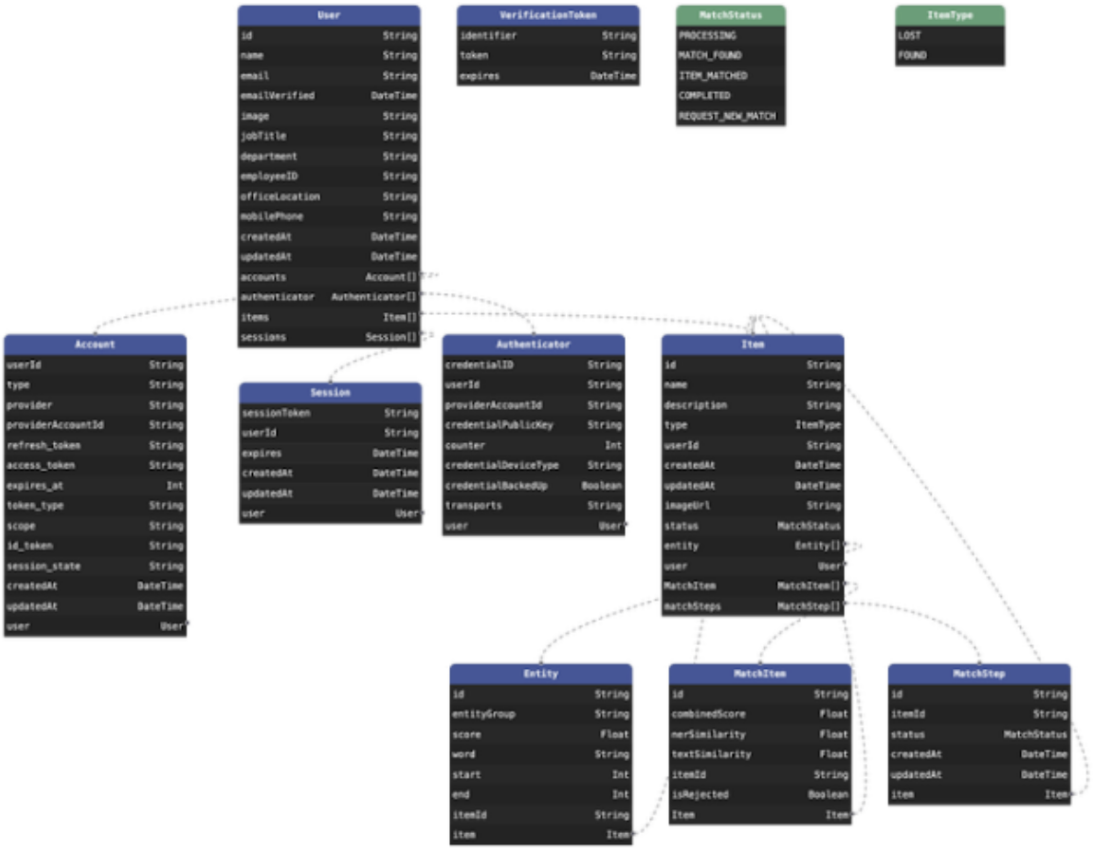

Figure 3.6: ER Diagram of LostNFound System

Figure 3.6 represents the conceptual diagram of the LostNFound System which we will use to store information about users and match details.

Chapter 4

System Functionality

4.1 Introduction

This chapter describes the core aspects of the system's functionality, covering its architecture, primary functions, planning, and testing results that define the application's capabilities. The first part explains the system's architecture, and the second part details the main functionalities.

4.2 System Architecture

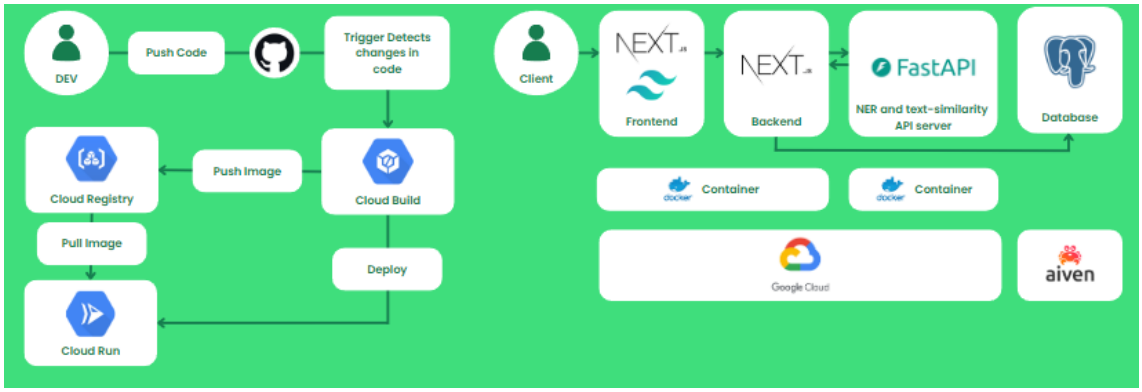

Figure 4.1: System Architecture

The system architecture consists of two main components designed to deliver an efficient lost-and-found platform. The front end, developed using Next.js, serves as the user interface for tasks like reporting lost items and browsing found items. It communicates with the backend via REST APIs and is containerized using Docker, deployed on Google Cloud Run for scalability. The backend, also built with Next.js and integrated with FastAPI, handles advanced functionalities like Natural Entity Recognition (NER) and text similarity algorithms to automate item matching. Data is managed using a PostgreSQL database hosted on Aiven, ensuring reliable and secure storage. The deployment process is streamlined with a CI/CD pipeline using Google Cloud Build, which automatically builds and deploys container images to Cloud Run, enabling an agile and scalable system.

4.3 Test Plan and Results

Module	Test Description	Expectation	Result
Authentication	Login with Microsoft Azure	Can log in with the KMUTT account	Success
Home Page	Navigate to Lost Report and Found Report forms	Can navigate to both forms	Success
Home Page	Navigate to the user profile page	Correct navigation to profile page	Success
Home Page	Log out from the profile	User logged out successfully	Success
Create Report Form	Fill in the form	Form is successfully filled and submitted	Success
Create Report Form	Validate required fields	Shows error for missing mandatory fields	Success
Create Report Form	Submit the report form	Report saved and reflected in the database	Success
Matching Process	Match lost items to found items	Matches displayed correctly with similarity score	Success
Matching Process	Display contact information of matched user	Contact information displayed correctly	Success
History Log	View previous reports filed by the user	Complete list of past reports displayed	Success
User Profile	Update user information (e.g., phone number)	User information updated successfully	Success

Table 4.1: Test Plan and Results

Chapter 5

Summary and Suggestions

5.1 Introduction

This chapter summarizes the project's outcomes, identifies the obstacles faced during development, and proposes strategies to address them. Additionally, it provides recommendations for improving the system's functionality and scalability to influence future revisions of the LostNFound platform.

5.2 Project Summary

The LostNFound platform introduced a centralized lost-and-found management system for King Mongkut's University of Technology Thonburi (KMUTT). The system streamlines reporting and recovering lost items, integrating a Next.js full-stack framework for user interaction and a FastAPI-powered backend. Advanced features such as Natural Entity Recognition (NER) and text similarity algorithms are employed for accurate item matching. PostgreSQL is used for reliable data management, and Docker with Google Cloud Run ensures seamless deployment and scalability. By addressing the inefficiencies of existing manual processes, this digital solution reduces response time and enhances user experience.

5.3 Problems Encountered and Solutions

5.3.1 Matching Algorithm Development

A significant challenge was finding an appropriate matching algorithm to achieve accurate and efficient recommendations. Extensive research and testing were conducted throughout development to fine-tune the algorithm, which ultimately met the project's requirements.

5.3.2 Domain Mapping to Server

Initially, mapping the domain name to the server posed difficulties due to inexperience with Google Cloud Platform (GCP). This issue was resolved by consulting documentation, properly configuring DNS settings, and successfully linking the domain to the server.

5.4 Suggestions for Further Development

5.4.1 Mobile Application

Developing a mobile version of the platform will increase accessibility and usability for KMUTT students.

5.4.2 Scalability

Expanding the system to support other universities or institutions facing similar lost-and-found challenges will broaden its impact.

5.4.3 AI Enhancement

Enhancing the item-matching algorithms with machine learning techniques can handle more complex scenarios, such as multi-language descriptions or vague item details.