

AI-Powered Lost & Found Management System: A Smart Campus Community Platform

Muthineni Pranay
School of Computer
Applications Lovely Professional
University Punjab, India
12309344

Subramanyam Suresh Sahu
School of Computer Applications
Lovely Professional University Punjab,
India
12307317

Sanny Raj
School of Computer Applications
Lovely Professional University
Punjab, India
12321171

Jatin Verma
School of Computer Applications
Lovely Professional University
Punjab, India
12307050

Sujal Raj
School of Computer Applications
Lovely Professional University Punjab,
India
12325516

Reetik Kumar
School of Computer Applications
Lovely Professional University
Punjab, India
12303618

Supervised By:
Ramandeep Kaur
Assistant Professor
Lovely Professional University Punjab, India - 30438

Abstract—Each semester millions of lost items end up in campus universities. Campus universities are where this happens. These lost personal effects range from important items such as ID cards to textbook to laptops and ear buds. Campus universities has a lost personal effects problem. Existing system where Colleges have been dealing with this issue: a lost items cardboard bin, inquiries at the security desk, physical bulletin boards-are not effective or time consuming and cumbersome for student use. To address this, our team has developed an AI Powered Lost and Found Management System. This paper details the architecture of a mobile application which links the person who has lost items to the person who has found items, along with its technology stack and development process. The frontend was developed in Flutter framework, the NoSQL cloud database we chose was the Firebase NoSQL database. To avoid the use of heavy image recognition models, we have designed an extremely optimized matching algorithm which utilizes NLP and Category filter with Geolocation Data to give us a similarity score of lost and found items. When a match is obtained, the users are automatically notified via Push Notification. Safety is also a top priority: user information is kept confidential until two students have exchanged private messages via the in-app chat application, and we have a QR code validation system so that people can only claim items that they own. Our final system provides a much more efficient, secure, and time effective solution for campus lost and found item management.

Index Terms—AI: Lost & Found, Flutter, Firebase, Smart campus, Mobile Application, QR security, cross-platform, NLP, geolocation.

I. INTRODUCTION

Imagine you lose something of value in a bustling campus/location. There are always thousands of students rushing between lectures, laboratories, and canteens everyday in any given university. Everyone is in a hurry therefore it is quite common for anyone to lose his/her wallet, dropped his/her phone, left his/her jacket behind and so on. Usually it involves the student having to spend valuable hours walking round the various department offices and inquiring the security personnel at the different campus gates, rifling through piles and piles of disordered lost and found items.

People normally do not consider the psychological stresses involved. Imagine you lose a laptop immediately before your midterms or lose your university ID which you require for your examinations. You'd certainly go into a panic state. The current system simply does not work well. You need a bit of luck. Someone would have found your item and returned it to a staff member or pasted it on a notice board. But you would only find it if you actually happened to look at the notice board or approach that staff member on the right day.

We in the team realized that the actual problem here was people do not have the information they need.

The finder of the lost property knows. It is not being transmitted to the owner who has lost it.

We thought if we uploaded it to a website or somewhere on

the internet then it would make life much easier; and also we want to bring in an app called Connected Campus Community and try and change the lost property system at the university.

What we wanted to do was utilize the computer to scan lost property items and get student's, or people who had lost it, to come and claim the lost property that way it would make the task much easier, for it would be done by the computer, not the student: so we decided that it was not really about losing things, and so we came up with Connected Campus Community, so that students can find lost items; the Connected Campus Community's purpose is to assist students to find items they have lost.

A. Objectives of the Project

Our overall goal in this Capstone project was to create an application that helps people quickly find lost items. With this app, we wanted to attain these three things:

- **Centralize the Process:** Clean up physical bulletin boards and clunky Whatsapp groups with a sleek and easy to use mobile reporting interface for items.
- **Build a Matching Engine:** Apply similarity score between the lost and found post using text data, item categories, and the GPS locations automatically.
- **Automate Communication:** If a potential match is found by the system then the system should automatically and instantaneously send a push notification to the student so that the student is not required to be logging in/out of the application and checking continuously.
- **Ensure Student Safety:** Hide user identities until a meeting time is arranged, use a QR code to prevent items being handed over to the wrong person.

II. FEASIBILITY STUDY AND USER SURVEY

Prior to actually starting any coding or configuring databases we wanted to make sure the problem was indeed a deal. We performed a study to evaluate if we were investing our time appropriately. Furthermore we conducted a survey distributed among university departments to determine concrete figures relating to the problem in question.

A. Survey Methodology

We then developed a google form, distributed to over 350 students, all studying a variety of programs (BCA, B.Tech and MBA). We sought to discover the rate at which individuals lose items.

Individuals lose their items. They then attempt to retrieve them.

We then also sought to discover the extent at which individuals are successful at retrieving lost items.

We sought to determine the extent at which an application would be utilized to aid lost item recovery.

We needed to ensure that our application was an idea that individuals would not only value but use.

B. Survey Results

It was quite evident from all answers that supported the idea that we had proposed.

- 1) **High Rate of Loss:** About 82% of the 350 surveyed (287 students) have lost at least one item of personal property on campus since enrolling.
- 2) **Poor Recovery Rate:** Only 24% of those people who lost something did manage to retrieve it with security guards/lost and found. Everyone else was out of luck and had to repurchase.
- 3) **High Interest:** It also appears that 95% of participants would download and use the campus lost and found application, provided that it was extremely quick to use to file a report.

Figure 1 shows a breakdown of the specific types of items students reported losing most often.

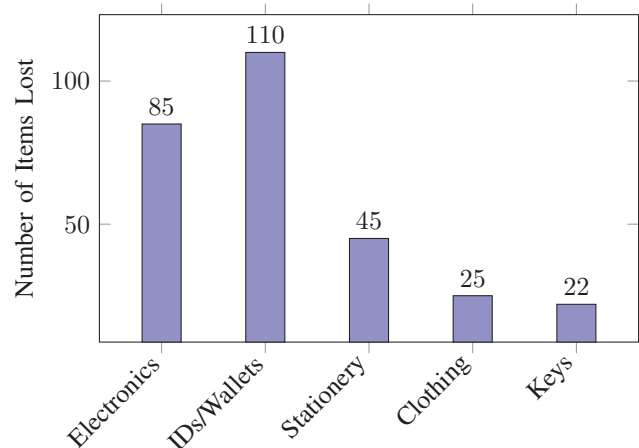


Fig. 1. Survey Results: Types of items most frequently lost by students. IDs, wallets, and electronics made up the vast majority of the data.

We saw from the results that the main lost items were important items (IDs and Wallets) and valuable items (Electronics such as chargers and earbuds). That only 24% of items are recovered indicated a failure in the current system. The survey gave us an important clue to our UI design. If it takes 5 minutes and twenty text fields to report a lost item, students will never use it.

III. BACKGROUND AND LITERATURE REVIEW

To ensure our strategy was. We tried to learn from the ways which universities and developers have dealt with the issue. This study of existing has led us to understand good and bad aspects. We found what good and bad things about the and approaches respectively.

A. Old-School Ledger Systems

Still a lot of colleges have kept the old tradition with pen and paper. A security guard at the university that you found a lost item will register it on the logbook and that will be it. That's the way of doing it. The only thing is that the research

suggests that using paper in the documentation and registration process of things will always involve considerable lost data and human error for the paper tracking itself. Paper tracking is not good [11]. When the student asks for a laptop one week later, there is a great difficulty for the staff. The staff must read pages after pages of a handwriting which is very complicated. This will take a lot of time. And when the book is lost (which can arrive very easily) the data, relative to the laptop is lost. The staff must restart from zero in order to look for the black laptop data.

B. Web Portals and Social Media

Some of the universities have switched to websites or use Facebook and Whatsapp groups. This is better than using paper, however there is a rapid accumulation of information that soon become disordered. For example a message about a lost USB will not be accessible, after a very short space of time, due to other information that had been broadcasted in the group. The Universities use Facebook and Whatsapp groups to communicate [4]. Additionally, you cannot search for something, like "Electronics" or "Lost on Monday", in a WhatsApp chat. People have no other way than scanning every message hoping to find the desired content in a WhatsApp chat concerning Electronics or in a WhatsApp chat concerning something lost on Monday.

C. Hardware Tracking

It is commonly thought that tracking is to be achieved by using special tags and devices that emit signal. These specially designed tags are known as RFID tags or Bluetooth beacons that could be used to monitor the campus assets. Tracking of the campus assets by RFID tags or Bluetooth beacons is possible. It can be a particular thing [9]. This is useful for items that are owned by the university, such as library books or projectors used in the lab. It is hardly feasible to expect students to run out and buy an RFID tag, and affix it to their own personal notebook or water bottle. So this system seems to work for university items such as library books or lab projectors. But for items such as a students own notebook or water bottle, having the student go out and buy an RFID tag seems to be the problem.

D. The Gap in the Market

This is useful for items that are owned by the university, such as library books or projectors used in the lab. It is hardly feasible to expect students to run out and buy an RFID tag, and affix it to their own personal notebook or water bottle. So this system seems to work for university items such as library books or lab projectors. But for items such as a students own notebook or water bottle, having the student go out and buy an RFID tag seems to be the problem.

IV. SYSTEM REQUIREMENTS AND TECH STACK

We wanted to create an application that will work correctly if it was used simultaneously by hundreds of students. When planning for it we decided upon some rules. The rules were of software and computers that would be supporting it.

A. Choosing the Software Stack

We chose the stack so we could really quickly build features that run across multiple platforms, and save money on backend servers. Using our modern stack allows us to rapidly develop cross platform features.

- **Frontend Framework:** We chose Flutter. We prefer to use Flutter over React Native as Flutter compiles down to machine code and therefore can achieve smoother scrolling of images on long lists. We are also big fans of Flutter for speed and performance, and image lists are very quick to display.
- **Language:** Dart strongly typed language that was helpful in finding many bugs.
- **Database:** Firebase Cloud Firestore Since there are many pieces of information about lost things which is constantly changing, the NoSQL database would be ideal for our usage. The information about a wallet is quite different to the information about a laptop.
- **Authentication:** Through Firebase Auth we have also been able to limit the logins to only allow university emails.
- **Storage:** All the images the users uploaded were stored in Firebase Cloud Storage.
- **Cloud Logic:** The Firebase Cloud Functions were also written in Node.js. Our Firebase Cloud Functions have a really important task of running our matching algorithm on the background. That is our matching algorithm which did everything. Our matching algorithm did work without the assistance of humans and our Firebase Cloud Functions run our matching algorithm on Node.js.

B. Hardware Requirements

The minimum requirements for the students who would use the application are:

- Any normal Android or iOs phone capable of accessing the internet.
- Working camera to capture images of items and scan QR codes.
- GPS compatible so the app can drop a pin at where the item was lost or found.

V. SYSTEM ARCHITECTURE AND FLOW DESIGN

So to make our code neat and organized we planned out the entire journey that the user would take as well as the whole system architecture using diagrams.

A. User Journey Flowchart

We aimed to be sure that a stressed student wouldn't get themselves confused and report the item.

As shown in **Figure 2**, a user can take three main paths from the home screen:

- 1) **Report Lost:** Fill in the input information (Name, Category, Description), choose a photo, add an photo, and Save As (Lost).
- 2) **Report Found:** Follow the form but "Found" as item. The easier the form, easier to learn.

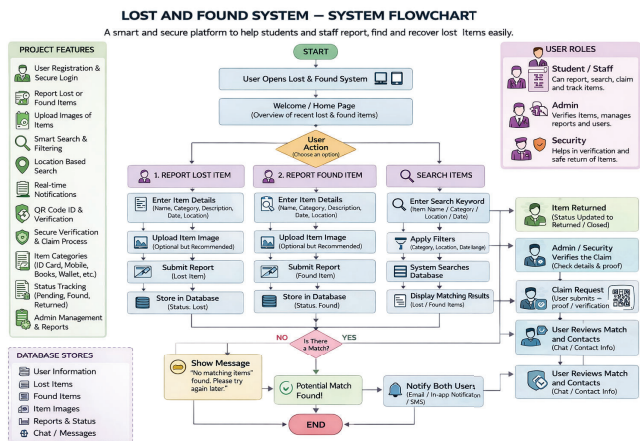


Fig. 2. The System Flowchart illustrating the complete user journey, from opening the app to the final QR Verification process.

3) **Search:** Navigate to the database and check using the keywords and category filters.

When our backend algorithm finds a good match, it completely skips the manual search and notifies both users for them to start the chat.

B. System Architecture

We have designed it such that it can be modular, just like Lego blocks to ensure the client app is really fast.

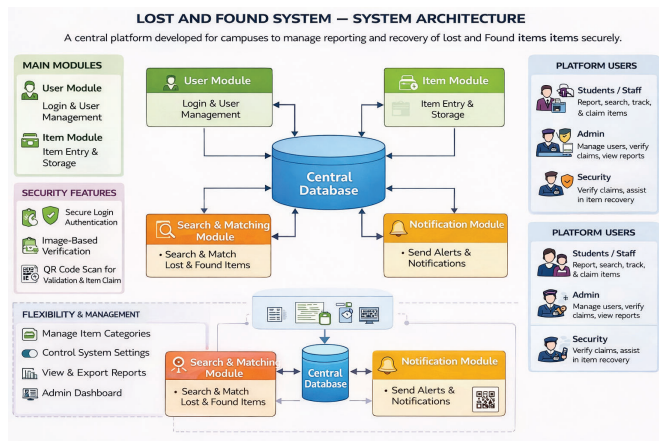


Fig. 3. The Modular System Architecture showing how the user app connects to the Central Database and background services.

Figure 3 illustrates our main architecture. Firebase Database is at center. The flutter mobile app manage User and Item modules (data to the database). And Search & Matching module is executed on separate Node.js server for background checking of matching, to prevent charging the mobile phone battery.

Note on "Image-Based Verification": On the Architecture diagram there is a block for Image-Based Verification. As

we had far too little expertise for our deep-learning AI for analyzing images it will actually be a **Human Visual Verification**. High quality pictures will be displayed in the app's chat view for safe and reliable use and for the user to be able to analyze particular scratches and details to confirm that the item is really the user's, before they meet up.

C. Simplified Operational Summary

We also developed a linear simple workflow that presents the app to a non-technical audience:

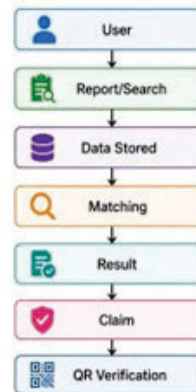


Fig. 4. A simple linear workflow representing the core operational steps.

As seen in **Figure 4**, the system moves from Report → Data Storage → Matching → Result → Claim → QR Verification in a strict order.

VI. UI/UX DESIGN BREAKDOWN

We had a segment in our project dedicated solely to the aesthetic of the app, and how user friendly the app could be. We believed the app had to have a nice enough interface to not infuriate someone who just had their phone lost. With this we concluded, from the testing we performed, to split the app into a few primary screens:

1. Onboarding Screen: you would like to know about the matching. Smart matching is another thing that could help locate things. When something has gone wrong, you must tell those responsible about the situation. They examine what has been wrong and repair what ever has broken. Smart matching is helpful, it assist individuals to locate whatever they are looking for. When something is out of place, you should let the responsible people know about it to enable smart matching to run correctly.

2. Login Screen: The system includes a gateway which asks for a university email address. This helps remove the fields which are not required to allow users a quicker login. With just the university email address one is able to sign in and gain access to the app.

3. Dashboard: The landing page displays a horizontally scrollable display of newly discovered items and active alerts. I lost something and I found something buttons dominate the screen.

4. Reporting Form: Instead of providing one giant text box for the user to fill, we split the form into steps. They can select an icon, like the one on the left representing Electronics, or they could select an icon for Keys. They then type in a description.

Lastly is the Google Map, the user can drop a pin in the location where their item was lost.

The user utilizes this map to pin point the location...

5. Match Results View: If you search for an item that our app may already have found for you, we can tell you how good the match is. Say you had lost your silver watch, the app could pull up a found silver watch and put a badge of "85 percent Match" on it.

This 85 percent match was decided by where the watch was found, and the descriptive words that you used when you had reported it lost.

This utilizes keywords along with location to tell us how good of a match the item you are looking for is to the one we may have found. So if you were looking for a silver watch we can show you other silver watches, using the keywords to find and display other watches of that kind. We use "silver" and "watch" to describe these items and match them to the ones we have found.

- **Category** ($W_1 = 40\%$): A lost ID will never be a found jacket. By searching first of the category, the system can immediately discard 80% of the database, reducing a tremendous amount of work time.
- **Time** ($W_2 = 20\%$): We check the timestamps. If keys are found on a Monday, they cannot be compared to the keys that have been lost on Wednesday.
- **Location** ($W_3 = 20\%$): GPS co-ordinates are used to work out how far "lost" is from "found".
- **Text Similarity** ($W_4 = 20\%$): We have simple Natural Language Processing to match some keywords in titles and description [8].

A. The Math Behind the Code

1. Calculating Distance: It isn't just basic geometry to work out distance between two GPS points because the Earth is round, we use the Haversine formula [5]:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1)$$

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \quad (2)$$

$$d = R \cdot c \quad (3)$$

Where R is the Earth's radius, φ is latitude, and λ is longitude. If the distance d is less than 200 meters (a short walk on campus), it gets maximum points.

2. Text Matching: If somebody has lost a particular thing such as a Dell laptop which has a sticker we delete words that don't mean anything for example a and with.

We consider the words Dell, laptop and sticker. We compare these to the words describing the item found.

The Dell laptop is what it looks like so that is important to us. We have a method by which we can compare the Dell laptop to the found item.

This comparison method is called the Jaccard Similarity Coefficient. By using this method it measures how similar the Dell laptop is to the item found.

With the Dell laptop words that matter to us are; black, Dell, laptop and sticker.

We consider the Dell laptop, sticker and laptop words and compare it to the words that describe the item found.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4)$$

This is the extent of the similarity between the two descriptions. If the sum of the scores calculated will be over 75%, then a notification will be pushed to both the students. This notification is sent to the students to inform them of the close proximity between their descriptions, of each other. The students receive this notification as the system seeks to keep the students informed of this information.

VII. DATABASE STRUCTURE

Firestore being a NoSQL database, we have to pay extra attention on the database structuring. A wrong structure will increase the number of reads which are going to cost us money and make the app slow. We have three top-level collections:

- 1) **Users Collection:** Stores user's email address, name and device token that is necessary to send him/her notifications.
- 2) **Items Collection:** This is where all the lost and found posts are stored. Each post (document) is represented with a category field, description field, a timestamp and a location map. It also has a imageUrl reference, and the actual large image file itself is kept in a separate storage bucket to ensure that the main text database runs incredibly quickly.
- 3) **Chats Collection:** It stores private conversation. It records the two user ids and the list of the messages, in the order of their time.

VIII. SECURITY AND PRIVACY

When we were creating the app that the entire campus will use, protecting students and their private information was an issue we all had to keep in mind.

A. Protecting User Data

Connection from mobile app to Firebase is secured with encryption for all data transmitted. We wrote strong rules in the database that nobody is able to read a chatlog if User ID is not explicitly designated Participant for that chatlog [2].

B. Anonymous chat

When using the public feed, the full name and phone number of students cannot be seen. Students who have matched up will speak via the anonymous chat function and will only be able to see the validated names when both users have agreed to meet to prevent abuse.

C. QR Code Verification

We have also developed the system whereby to stop a pupil fraudulently stating an iPad or similar item of value which doesn't belong to them is his [1].

- 1) The student who claims the item will generate a QR code on the students screen.
- 2) The student will then go to a safe, public meeting place to meet the item finder.
- 3) The item finder will then scan the students QR code using the app.
- 4) The secret token for that student in the database will be checked. If the secret token is correct then the item will be updated to "Returned" in the database and a log of the action will be created.

IX. TESTING AND PERFORMANCE

We knew the app had to be rock solid during finals, the time students are under the most stress and likely to misplace most. We implemented several integration tests to check our application.

A. Test Cases

We used a populated dummy test database and automated our tests. We executed the tests listed in Table I:

TABLE I
SYSTEM INTEGRATION TEST RESULTS

Module	Test Scenario	Expected	Status
Login	Try non-university email	Reject	PASS
Database	Submit item with no description	Reject	PASS
Algorithm	Run match on 500 items at once	< 2 sec	PASS
Security	Scan an expired QR code	Reject	PASS

B. Matching Accuracy

We also measured how our algorithm for text-and-location performed by providing it with 200 fake posts.

TABLE II
ALGORITHM ACCURACY BY CATEGORY

Category	Tests	Correct Matches	Accuracy
Electronics	50	47	94.0%
Documents/IDs	50	49	98.0%
Clothing/Bags	50	36	72.0%
Keys	50	34	68.0%

Algorithm works perfectly with IDs and Electronics as student names, brand names are extremely distinct. But they seem to have a little problem with Clothing and Keys, as students sometimes write "blue jacket". In this case, they need to examine by pictures themselves.

C. Handling Heavy Traffic

We then ran a test, to see how a large number of students running the app at the precise same time would effect the server.

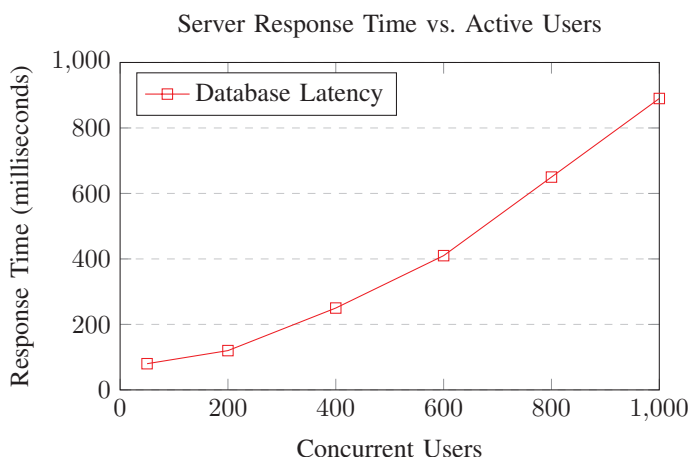


Fig. 5. Even with 1000 students searching at once, the database responds in under 1 second, meaning the app won't freeze.

X. DEPLOYMENT AND CHALLENGES

A. Cost Analysis

Since we used a serverless Firebase framework, it's very affordable to run this on the Spark free tier on our first launch. This will provide us with 50,000 free reads per day, which will be more than enough for our campus. In addition, if we branch out to other universities, it will be extremely affordable with the pay as you go tier.

B. Challenges We Faced

Developing an entire app made students encounter a few tough issues:

- **UI Lag:** Initially our app redraw the whole of the chat screen on every incoming message which caused stuttering on the phone and killed the battery, so we rewrote our state management code using the flutter `Provider` package to only redraw the incoming text bubble.
- **Bad User Input:** Our matching algorithm broke in the face of really poorly worded descriptions. We addressed this by requiring users to select a 'Color' and a 'Brand' from dropdowns in order to post.

XI. CONCLUSION AND FUTURE SCOPE

Our AI Powered Lost & Found Management System is an evidence that basic cloud technology combined with smart algorithm can solve a common issue faced by many in their campus. By substituting the useless cardboards and complicated Whatsapp groups, our app greatly decreased the time to retrieve the lost item by making it as fast as scanning it in the phone through our rapid mobile application. Flutter helped in creating an enjoyable UI and Firebase in providing the speed to the whole system.

In future when we have a much better grasp of machine learning, we intend to integrate an actual TensorFlow image recognition model so that the app automatically tags image without any user intervention, thereby the user does not has to

type anything to tag the image, and develop a web application which can be viewed and operated by the campus security.

REFERENCES

- [1] S. Sinha et al., "A Novel Approach to Enhance Campus Lost and Found Services through Integration of QR Code with Personalized Item Registration," *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, Pune, India, 2024, pp. 1-7, doi: 10.1109/TQCEBT59414.2024.10545109.
- [2] H. Zhao and S. Peng, "Design and Implementation of the Lost-and-Found System Based on Amap API," *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2018, pp. 1-4, doi: 10.1109/ICSESS.2018.8663776.
- [3] P. Choudhary, A. K. Choudhary, A. P. Srivastava and A. Singh, "Find Mine: Find the Lost Items via Mobile App," *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, London, United Kingdom, 2021, pp. 491-495, doi: 10.1109/ICIEM51511.2021.9445379.
- [4] M. J. Hossain, M. A. Bari, M. T. Mahmud and M. M. Khan, "Web and Mobile Application Based Missing Query Platform (Lost and Found BD)," *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, 2021, pp. 0074-0079, doi: 10.1109/IEMCON53756.2021.9623081.
- [5] S. Konishi, Y. Kawakita and H. Ichikawa, "Method for estimation of distance between objects and its application for finding lost objects," *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, USA, 2012, pp. 708-710, doi: 10.1109/CCNC.2012.6181152.
- [6] G. K. Prashanth, U. Uday Kumar, B. G. Premasudha, N. L. Rajesh and N. Vinothini, "Mobile Technology for Efficient Lost and Found Item Retrieval Using GIS Based Approach," *2025 3rd International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, Tumakuru, India, 2025, pp. 1-5, doi: 10.1109/ICSSES64899.2025.11009353.
- [7] S. Chutichudet, T. Kanthathasiri, I. Ritsakunchai and D. Wongsawang, "LFD: Lost and found dog application on mobile," *2014 Third ICT International Student Project Conference (ICT-ISPC)*, Nakhonpathom, Thailand, 2014, pp. 147-150, doi: 10.1109/ICT-ISPC.2014.6923238.
- [8] S. Khruahong, X. Kong, K. Sandrasegaran and L. Liu, "Develop An Indoor Space Ontology For Finding Lost Properties for Location-Based Service of Smart City," *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, Bangkok, Thailand, 2018, pp. 54-59, doi: 10.1109/ISCIT.2018.8588014.
- [9] N. A. Hamidi, A. A. Aziz, A. Ismail and A. M. Lokman, "Found It! Object Tracker Mobile Application," *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, Kuala Lumpur, Malaysia, 2023, pp. 1-6, doi: 10.1109/ICRAIE59459.2023.10468161.
- [10] M. Ghazal, F. Haneefa, S. Ali, Y. Alkhalil and E. Rashed, "Mobile-Based Archival and Retrieval of Missing Objects Using Image Matching," *2015 3rd International Conference on Future Internet of Things and Cloud*, Rome, Italy, 2015, pp. 627-632, doi: 10.1109/FiCloud.2015.80.
- [11] N. M. David, G. M. Catapang, K. P. V. Pabilando, L. T. Dala, R. F. Marcos and A. D. Lacasandile, "A Comparative Study on Lost and Found Management Systems in Academic Institutions: Assessing Reliability Across Universiti Tunku Abdul Rahman, California State Polytechnic University, and Nazareth School of National University," *2024 6th International Workshop on Artificial Intelligence and Education (WAIE)*, Tokyo, Japan, 2024, pp. 230-236, doi: 10.1109/WAIE63876.2024.00049.
- [12] G. K. Prashanth, U. Uday Kumar, B. G. Premasudha, N. L. Rajesh and N. Vinothini, "Mobile Technology for Efficient Lost and Found Item Retrieval Using GIS Based Approach," *2025 3rd International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, Tumakuru, India, 2025, pp. 1-5, doi: 10.1109/ICSSES64899.2025.11009353.