International Journal of
Science, Architecture, Technology and Environment

SEE YOU IN CEU: A Digital Lost and Found System for Campus Using Convolutional Neural Network (CNN)

Julianna Marie M. Quitlong[1], Domnic Gabriel E. Escalante[2], Joemel L. Bernardino[3]*, Eliza B. Ayo[4]

[1234]School of Science and Technology, Centro Escolar University, Manila, Philippines

*Corresponding Author: ebayo@ceu.edu.ph

DOI: https://doi.org/10.63680/ijsate0326019.017

Abstract

Traditional lost-and-found systems in university campuses are largely manual, fragmented, and susceptible to administrative inefficiencies. This study presents the design, development, and evaluation of **See You in CEU**, a mobile-based digital lost-and-found system developed for Centro Escolar University (CEU) Manila. The system leverages a Convolutional Neural Network (CNN)-based image recognition module integrated through TensorFlow Lite to automatically extract descriptive keywords from uploaded images of lost items. Rather than relying on computationally intensive matching algorithms, the platform utilizes a lightweight keyword-based matching mechanism to ensure cost-efficiency and accessibility in a resource-constrained academic environment. Developed using the Flutter framework with Firebase backend services, the system incorporates university-based authentication, structured item reporting, administrative validation, and real-time push notifications. The study followed an Agile Software Development Life Cycle (SDLC) framework and evaluated system quality based on ISO/IEC 25010 criteria through functional testing and a Likert-scale survey administered to 48 CEU students, faculty, and staff. Results indicate strong performance across functionality, usability, reliability, security, compatibility, and maintainability dimensions. The findings demonstrate that integrating CNN-assisted image recognition into campus service applications provides a scalable and efficient approach to modernizing lost-and-found operations in higher education institutions.

Keywords: digital lost-and-found; convolutional neural network; mobile application; campus information systems; image recognition; ISO/IEC 25010

1. Introduction

University campuses operate as dynamic ecosystems where thousands of individuals interact daily across classrooms, laboratories, libraries, and open spaces. Within such environments, the loss and recovery of personal belongings is an inevitable occurrence. Traditional lost-and-found systems typically rely on manual logbooks, bulletin boards, or isolated office-based reporting mechanisms. These methods are often inefficient,

time-consuming, and prone to record mismanagement.

At Centro Escolar University (CEU) Manila, lost items are commonly reported through informal channels or surrendered to designated offices. However, the absence of a centralized, technology-driven platform results in delayed matching, miscommunication, and reduced recovery rates. As institutions increasingly adopt digital transformation strategies, the modernization of auxiliary campus services—such as lost-and-found operations—becomes both necessary and strategic.

Advances in mobile computing and artificial intelligence, particularly Convolutional Neural Networks (CNNs), provide new opportunities to enhance object identification and classification tasks. CNNs have demonstrated strong performance in image recognition by automatically extracting visual features without manual feature engineering. Integrating such capabilities into a campus-based mobile application can streamline item reporting and improve matching accuracy.

This study addresses the need for a centralized and intelligent lost-and-found system by developing **See You in CEU**, a CNN-assisted digital platform tailored for the CEU Manila community.

Statement of the Problem

The study was guided by the following research questions:

1. What gaps in the existing literature and comparable applications can the proposed system address?
2. What hardware, operating system, and software technologies are appropriate for its development and deployment?
3. How does the system address those identified gaps?
4. How does the system perform under quality assurance testing across ISO/IEC 25010 criteria?
5. How do end users evaluate the system across the seven dimensions of the same standard?

2. Related Literature

2.1 Digital Lost-and-Found Technologies

The digitization of lost-and-found processes has been an active area of applied research for over a decade. Early efforts focused on mobile reporting interfaces and direct communication channels between users. Salman and Athab (2022) and Kanjalkar et al. (2024) demonstrated that user-friendly reporting platforms significantly reduce the time required for item recovery and lower the administrative load on staff. Tan and Chong (2023) extended this work into a university-specific context, employing a Java-based Android application with local SQLite storage and university-credential authentication. Their findings confirmed that a mobile-first design supports on-the-go reporting and reduces dependence on physical lost-and-found desks.

Subsequent research has explored the integration of artificial intelligence to automate image-based item identification. Prawira and Saputri (2023) combined ResNet-50 with natural language processing to generate descriptive item features, while Zhou et al. (2024) employed MobileNetV2 alongside perceptual hashing to achieve high classification accuracy. These results are impressive, yet the infrastructure requirements—dedicated GPU hardware and large annotated datasets—place such solutions beyond the reach of most academic institutions. The present system addresses this tension by using a pre-trained CNN model deployed via TensorFlow Lite to perform on-device image feature extraction, limiting AI involvement to keyword

generation. This design draws on Si et al. (2022), who argued that lightweight AI-assisted description generation can substantially improve retrieval accuracy without heavy computational costs.

2.2 Real-Time Notifications and User Engagement

Real-time notification systems play a central role in improving user responsiveness within digital lost-and-found platforms. Durachman et al. (2020) demonstrated that event-driven push alerts reduce retrieval delays and maintain user engagement without requiring constant manual monitoring. Applied to a lost-and-found context, this ensures that a user who submits a report receives an automated alert as soon as a matching item is logged. Community engagement strategies are equally important. Ar Razi and Putra (2020) applied the Hooked Model to lost-and-found mobile applications, showing that habit-forming design elements increase interaction frequency. Jiang et al. (2020) found that gamification features further strengthen participation. Although the present system does not implement formal gamification, its push notification pipeline provides timely feedback that reinforces reporting behavior.

2.3 Security, Privacy, and Verification

Security and privacy are significant concerns in community-facing applications. Liu et al. (2021) developed a passive tracking system linking surveillance to smartphones, raising ethical questions about consent. Yagi et al. (2021) addressed similar concerns in GO-Finder by giving users control over stored data. Almilby and Alkhawaji (2024) identified privacy risks in GPS-based tracking systems deployed in crowded environments. University-based systems face distinct security requirements, particularly unauthorized item claims. Mohammed et al. (2023) and Tan and Chong (2023) emphasized the necessity of institutional identity verification. Mahadik et al. (2021) highlighted the role of administrative oversight. The present system addresses these concerns by restricting access to verified CEU account holders, routing reports through administrative approval, and excluding GPS tracking in favor of office-based custody reporting.

2.4 Gaps in Existing Systems

A synthesis of the reviewed literature reveals recurring limitations. First, AI-based matching systems often impose excessive computational requirements (Zhou et al., 2024). Second, many applications lack integration with university-specific verification protocols (Tan & Chong, 2023). Third, real-time GPS tracking generates privacy risks difficult to mitigate in student-facing apps (Almilby & Alkhawaji, 2024). Fourth, cross-platform compatibility remains underexplored (Zohud & Zein, 2021). Finally, many platforms lack structured administrative validation (Soumya et al., 2024). The present study targets these gaps through specific technical and procedural design decisions.

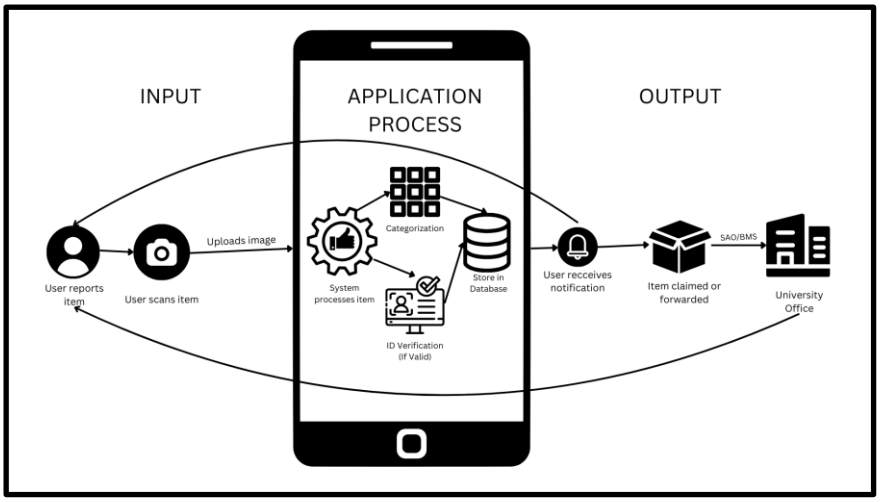


Figure 1. *Conceptual Framework*

3. Methodology

3.1 Research Design and Approach

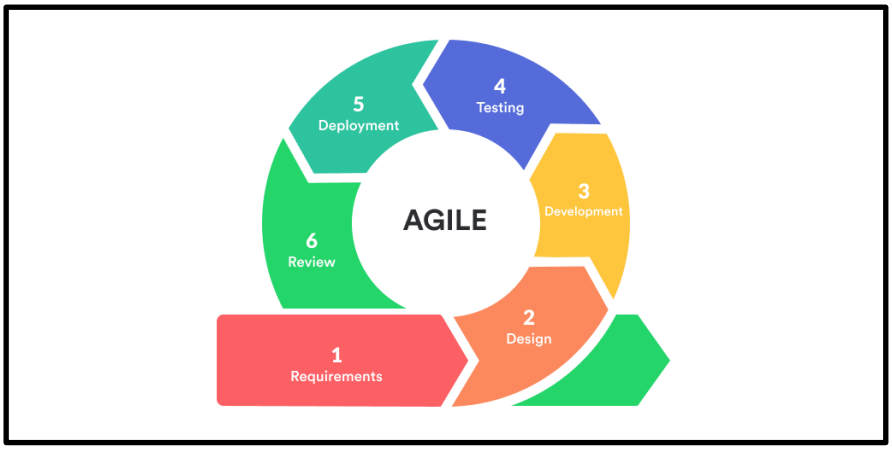


Figure 2. *Software Development Process*

The study employed a developmental research design, appropriate for producing and evaluating a functional artifact (Richey & Klein, 2014). Both quantitative and qualitative data were collected: quantitative metrics included item-matching accuracy and ISO 25010 survey scores, while qualitative feedback was gathered through focus groups. A descriptive-evaluative approach was applied to interpret performance data against quality criteria.

3.2 System Architecture and Technology Stack

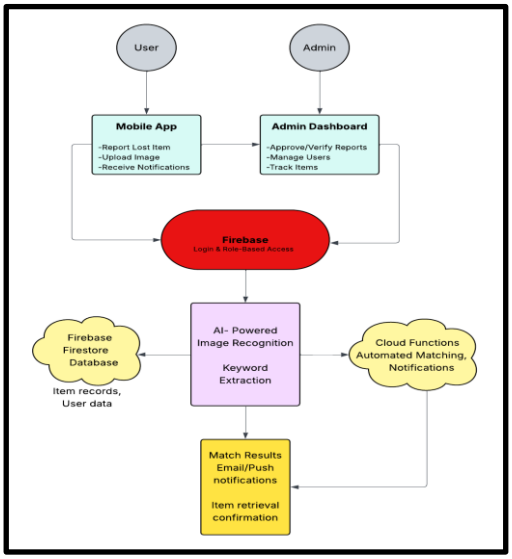


Figure 3: *Convolutional Neural Network–Based Object Detection Algorithm*

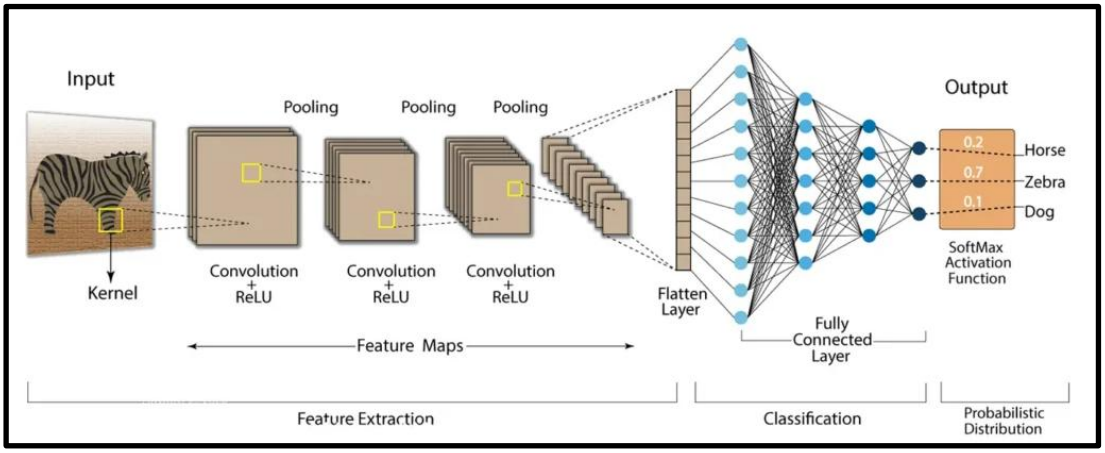


Figure 4: *Convolutional Neural Network*

The system integrates a Convolutional Neural Network (CNN)–based object detection algorithm as a core component for identifying and classifying lost and found items within the "See You in CEU" Digital Lost and Found System. Images submitted by users are processed through a pre-trained CNN model deployed using TensorFlow Lite and supported by Google AI technologies, which automatically extract visual features such as edges, shapes, textures, and object patterns through successive convolutional and pooling layers. These extracted features are analyzed by detection layers to generate object labels, confidence scores, and bounding box information, enabling the system to recognize item categories commonly found within the university environment. By leveraging a CNN-based approach powered by Google AI, the system transforms raw image data into structured and meaningful information that supports accurate item reporting, verification, and matching. This method reduces reliance on manual description alone and enhances consistency and reliability

in item identification across varying lighting conditions, camera qualities, and submission scenarios. The detected object information is subsequently utilized by higher-level system modules for item validation, administrative review, and user notification, thereby improving the overall efficiency and trustworthiness of the campus-wide lost and found process.

The application was built using Flutter, an open-source cross-platform framework producing native binaries for Android and iOS. Firebase provided backend services: Authentication managed university login; Cloud Firestore served as the database; Cloud Storage held images; and Cloud Functions automated logic. Image-based keyword generation utilized a pre-trained CNN model via TensorFlow Lite. The CNN extracts spatial features from uploaded photos to generate descriptive keywords, which are stored with the item record. Downstream matching compares these descriptors textually, avoiding heavy neural inference during the search process.

3.3 System Features

The application supports regular users (students, faculty, staff) and administrators. Users report items via a structured form including photos and location. Reports require administrative approval before becoming searchable. The search interface supports keyword queries and filtering. When a potential match is detected, push notifications alert both parties. Administrators use a dashboard to manage reports, users, and item custody status.

3.4 Participants and Sampling

The study population comprised enrolled students, faculty, and staff at CEU Manila. Sample size was determined using Slovin's formula (95% confidence, e=0.05), yielding a minimum of 48 participants. Stratified random sampling ensured proportional representation. Participation was voluntary and preceded by informed consent.

3.5 Data Collection Instruments

Three instruments were used: (1) A Likert-scale questionnaire (1=Poor, 4=Excellent) aligned with ISO/IEC 25010 criteria; (2) System logs from Firebase Performance Monitoring and Crashlytics; (3) Semi-structured focus group interviews with eight participants to elicit qualitative feedback.

3.6 Quality Assurance Testing

Software quality was assessed using: Firebase Test Lab (functional/compatibility), Postman (API validation), Figma and Flutter DevTools (usability), Firebase Performance Monitoring (latency), OWASP ZAP (security), Firebase Rules Simulator (access control), Firebase Crashlytics (reliability), and SonarQube (code quality).

3.7 Data Analysis

Quantitative survey responses were analyzed using descriptive statistics (means, standard deviations). QA scores were reported on a 10-point scale. Qualitative data were analyzed thematically. Survey means were interpreted as: 3.50–4.00 (Excellent), 2.50–3.49 (Satisfactory), 1.50–2.49 (Fair), 1.00–1.49 (Poor).

3.8 Ethical Considerations

The study adhered to CEU Manila's ethical guidelines. Informed consent was obtained, data were anonymized and encrypted, and access was restricted to verified university accounts. Data retention followed a strict one-year policy before secure deletion.

4. Results

4.1 Research Gaps

Table 1
Combined Gaps Identified in Literature and Comparable Applications

Category	Source/Application	Identified Gap	Citation
Literature	LostNet (A Smart Way for Lost and Find)	High computational cost of AI-based item matching	Zhou et al., 2024
Literature	Spatio-temporal Modeling for Lost Item Tracking	Absence of cost-effective solutions for university settings	Hassan et al., 2023
Literature	Campus Lost-and-Found Platform (Digital Map)	Limited integration with university-specific verification systems	Jiang et al., 2020
Literature	Lost and Found Assistance App for the Holy Mosque	Privacy risks associated with real-time GPS tracking	Almilby & Alkhawaji, 2024
Literature	Smart Lost and Found Solutions: A Review	Cross-platform testing and long-term maintenance challenges	Zohud & Zein, 2021
Application	AUFound	Limited customization for campus-specific workflows	Castro et al., 2023
Application	LostNet	Heavy dependency on computationally intensive AI image recognition	Zhou et al., 2024
Application	FOUNDIT	Absence of university identity verification for item claiming	Patil et al., 2024
Application	Digital Lost and Found Portal	No structured category-based search filtering	Mahadik et al., 2021
Application	Campus Lost	No integration of secure administrative validation workflow	Soumya et al., 2024

4.2 Technologies and Implementation

Table 2
Hardware and Operating System Requirements

Category	Component	Development Configuration	Implementation Requirement
Hardware	CPU/Laptop	ASUS TUF DASH F15 (Intel Core i5, 8GB RAM)	Intel Core i3 minimum; i5 recommended
Hardware	Mobile Device	Android 13 / iOS 17	Android 10 / iOS 14 or higher

Category	Component	Development Configuration	Implementation Requirement
Hardware	Display	1080p monitor	5.5-inch 720p minimum
Hardware	Internet	25 Mbps broadband	5 Mbps stable minimum
OS	Development	Windows 10 Home 64-bit	Primary build environment
OS	Emulator	Android Studio (Android 13)	UI testing
OS	User Devices	Android 10+ / iOS 14+	Minimum supported versions

Table 3

Software Stack for Development and Implementation

Component	Development	Implementation
Framework	Flutter SDK (Dart)	CEU Digital Lost & Found App
Backend	Firebase (Auth, Firestore, Storage)	Firebase Cloud Services
AI Processing	TensorFlow Lite	On-device keyword extraction
Design	Figma	Material UI components

4.3 System Responses to Identified Gaps

Table 4

System Responses to Identified Gaps

Identified Gap	System Response
High AI computational cost / AI overreliance	Replaced end-to-end AI matching with CNN-based keyword generation and lightweight textual comparison.
Lack of cost-effective solutions	Built using open-source technologies (Flutter, Firebase free tier, TensorFlow Lite).
Insufficient university integration	Mandatory CEU credential authentication and admin approval workflow.
Privacy risks from GPS tracking	GPS tracking excluded; location stored as office designation only.
Cross-platform maintenance issues	Single Flutter codebase compiled for Android and iOS.
Limited customization	Tailored campus-specific categories and office-based custody management.
No university ID verification	Login restricted to @ceu.edu.ph domain.
No structured search filtering	Category dropdown filters and comma-separated keyword search.
No administrative validation workflow	Two-tier admin approval for posting and releasing items.

4.4 Quality Assurance Results

Table 5
Quality Assurance Testing Summary

Quality Attribute	Tool Used	Test Conducted	Score	Interpretation
Functionality	Firebase Test Lab	Reporting, searching, claiming modules	8/10	Good
Functionality	Postman	API validation	9/10	Excellent
Usability	Figma	Heuristic evaluation	9/10	Excellent
Usability	Flutter DevTools	Widget tree review	8/10	Excellent
Performance	Flutter DevTools	Frame rate and memory usage	8/10	Good
Performance	Firebase Monitoring	Network latency	9/10	Excellent
Security	OWASP ZAP	Vulnerability scanning	9/10	Excellent
Security	Firebase Rules	Access rule verification	9/10	Excellent
Compatibility	Firebase Test Lab	Cross-version behavior	8/10	Good
Compatibility	Emulator	Responsiveness testing	8/10	Good
Reliability	Crashlytics	20 test cycles	9/10	Excellent
Reliability	Manual Testing	Repeated reporting cycles	9/10	Excellent
Maintainability	SonarQube	Code quality analysis	8/10	Good
Maintainability	GitHub	Version control efficiency	9/10	Excellent

4.5 User Evaluation Based on ISO 25010 (n = 48)

Table 6
User Evaluation Results by ISO 25010 Quality Characteristics

Quality Characteristic	Criterion	M	SD	Interpretation
Functional Suitability	Covers specified functions	3.77	0.47	Excellent
	Produces correct results	3.62	0.66	Excellent
	Supports goal achievement	3.83	0.42	Excellent
Performance Efficiency	Responds quickly	3.70	0.53	Excellent
	Runs smoothly	3.66	0.65	Excellent
	Performs under load	3.72	0.48	Excellent
Compatibility	Works on different devices	3.22	0.94	Satisfactory
	Compatible with OS versions	3.06	1.08	Satisfactory
	Works with concurrent apps	3.58	0.73	Excellent
Usability	Easy to understand/use	3.81	0.48	Excellent
	Easy to navigate	3.81	0.44	Excellent
	Clear interface	3.79	0.45	Excellent
Reliability	Operates without crashing	3.66	0.55	Excellent
	Consistently accessible	3.70	0.57	Excellent

Quality Characteristic	Criterion	M	SD	Interpretation
Security	Protects personal information	3.70	0.57	Excellent
	Prevents unauthorized alteration	3.77	0.50	Excellent
	Verifies membership	3.79	0.45	Excellent
Portability	Installation on other devices	3.47	0.79	Satisfactory
	Setup/transfer speed	3.58	0.67	Excellent
	Reliable across locations	3.70	0.53	Excellent

Note. M = Mean; SD = Standard Deviation; n = 48 respondents.

```
await Firebase.initializeApp();

FirebaseFirestore.instance.settings = const Settings(

  persistenceEnabled: true,

  cacheSizeBytes: Settings.CACHE_SIZE_UNLIMITED,

);
```

The code above initializes Firebase services and enables offline persistence for Firestore to improve performance and reliability.

```
void _setupNotificationActionHandlers() {
AwesomeNotifications().setListeners(
onActionReceivedMethod: (receivedAction) async {
if (receivedAction.payload != null) {
final payload = receivedAction.payload!;
navigationService.navigateToItemDetails(payload);
}
},
);
}
```

The code above registers a listener for push notification interactions using the Awesome Notifications package. When a user taps a notification, the system retrieves the attached payload data (e.g., item ID and type) and automatically navigates to the corresponding item details page. This enables real-time user engagement and seamless redirection to relevant system content.

```
class HomePage extends StatelessWidget {
@override
Widget build(BuildContext context) {
return StreamBuilder<User?>(
stream: FirebaseAuth.instance.authStateChanges(),
builder: (context, snapshot) {

if (snapshot.hasData && snapshot.data != null) {
```

```
final user = snapshot.data!;

if (user.email != null &&
user.email!.toLowerCase().endsWith('@ceu.edu.ph')) {
return UserHomePage(user: user);
}
}

return LoginSelectionPage();
},
);
}
}
```

The code above listens to Firebase Authentication state changes using a StreamBuilder. If a user is already authenticated and the email domain matches @ceu.edu.ph, the system automatically redirects the user to the main user dashboard. Otherwise, the login selection interface is displayed. This ensures that only verified institutional accounts can access the system, enforcing university-based identity validation.

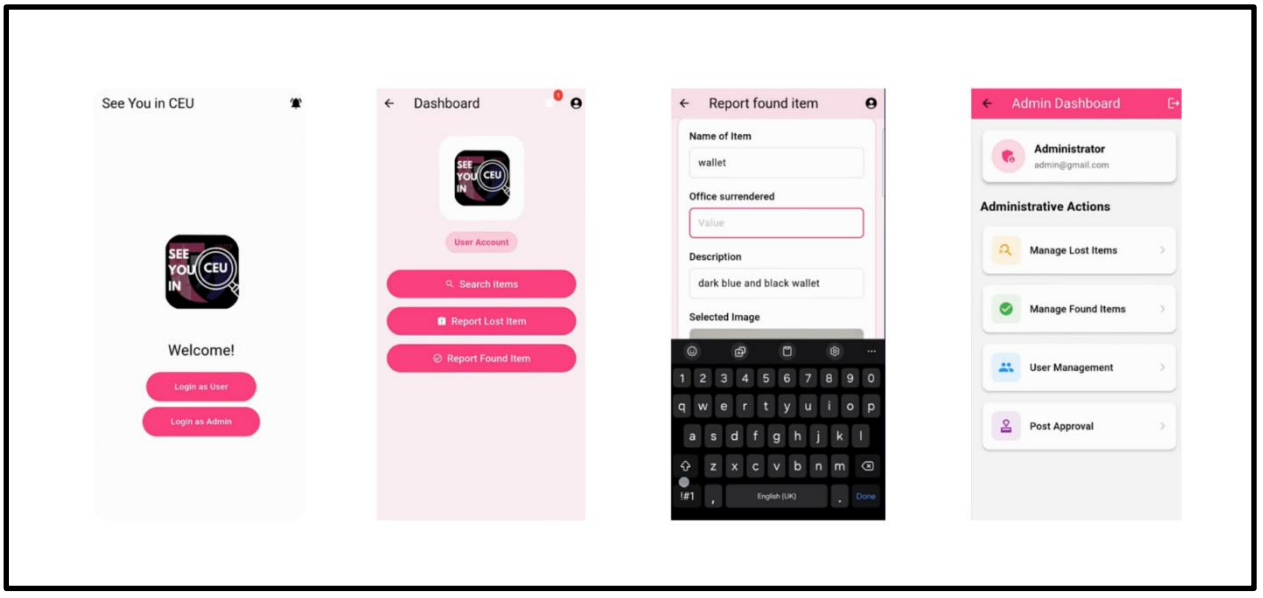


Figure 5. *Software Interface*

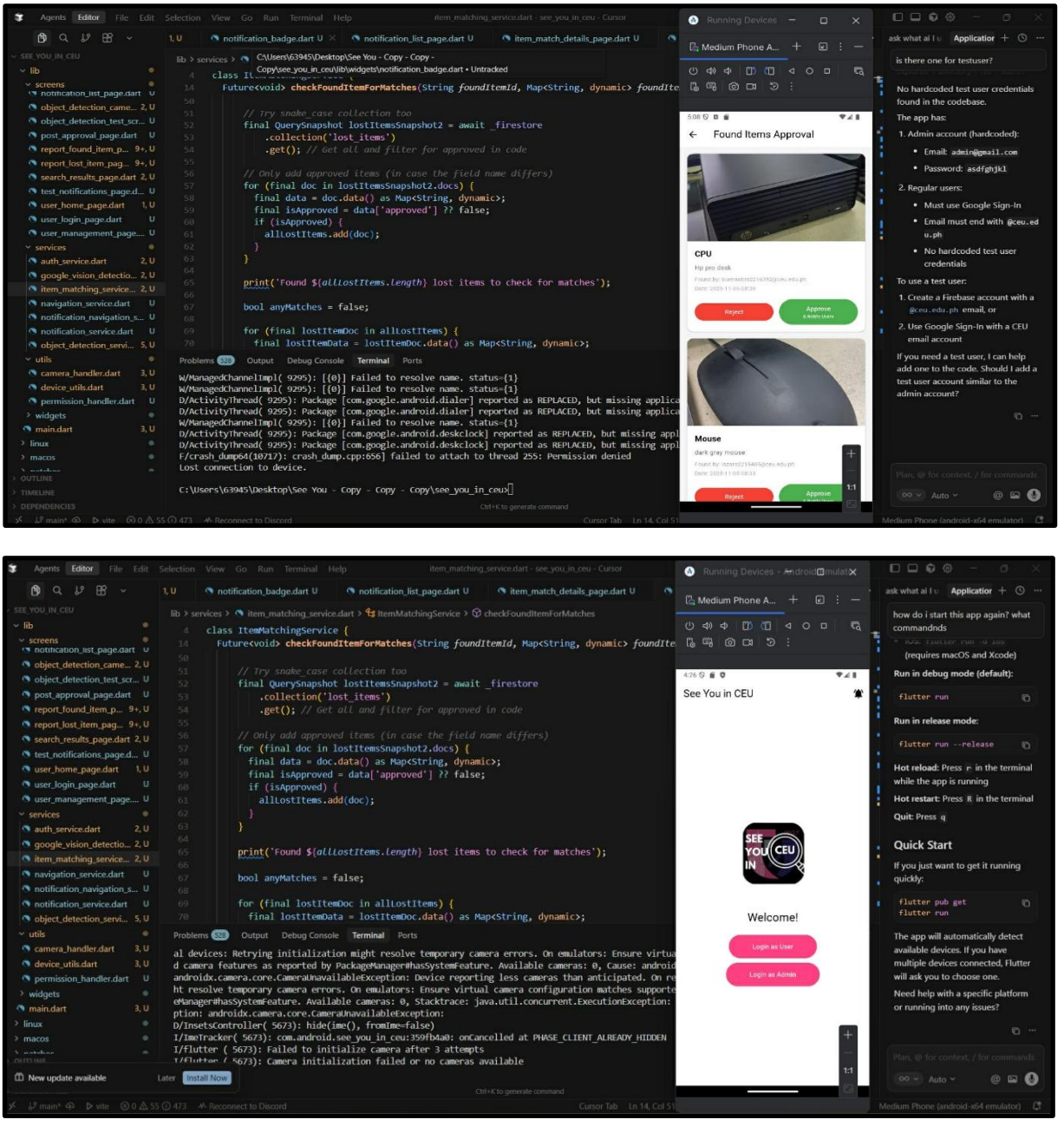


Figure 6. *See You in CEU codes*

5. Discussion

5.1 System Performance and Quality Assurance

Quality assurance testing results were consistently strong across six of the seven ISO/IEC 25010 criteria. Functionality testing yielded scores of 8/10 and 9/10, confirming that core reporting, search, and claiming operations behave as intended. Minor UI latency during report submission was attributed to network round-trip time, consistent with similar Firebase-backed applications (Tan & Chong, 2023). Security testing detected no significant vulnerabilities, and role-based access controls were verified as effective. Reliability scores were excellent, with zero crashes reported across 20 simulated cycles, suggesting the system is ready for sustained deployment.

5.2 User Evaluation Findings

User evaluation scores were overwhelmingly positive. Functional Suitability (3.62–3.83), Usability (3.79–3.81), and Security (3.70–3.79) received the highest ratings. The high Usability score indicates that the interface successfully accommodates users with varying digital literacy. However, Compatibility produced the lowest mean scores (3.06–3.22), particularly regarding older Android versions where specific Material Design components displayed rendering inconsistencies. Portability ratings revealed some friction in installation (3.47), likely due to the need for sideloading the application during the pilot phase rather than downloading it from a public app store.

5.3 Addressing the Research Gaps

The study demonstrates that a practical, institutionally aligned lost-and-found system can be built without the heavy computational overhead of end-to-end AI matching. The keyword-based approach reduces inference requirements to a single on-device pass, making the system viable on consumer hardware. Furthermore, integrating CEU-specific verification and office-based custody workflows addresses the gap regarding institutional accountability. By excluding real-time GPS tracking, the system also navigates privacy risks effectively, offering a secure compromise between utility and user protection.

5.4 Limitations

Several limitations exist. The sample size of 48 respondents limits generalizability. The keyword-based matching relies on the quality of generated descriptions; vague terms may lead to missed matches. Additionally, the pilot did not test high-volume concurrent access, leaving questions about performance during peak academic periods.

6. Conclusion

This study presented the design and evaluation of See You in CEU, a digital lost-and-found system tailored for Centro Escolar University. The system effectively bridges the gap between manual processes and high-cost AI solutions by utilizing lightweight CNN-based keyword generation. Quality assurance and user evaluation results confirm the system is robust, secure, and user-friendly, with "Excellent" ratings across most criteria. It

is ready for campus-wide deployment, offering a scalable model for other resource-constrained academic institutions.

RECOMMENDATIONS

1. Enhancement of AI matching: Explore lightweight semantic similarity measures (e.g., cosine similarity) to improve retrieval recall for vaguely described items.

2. Web-based administrative dashboard: Develop a browser-based interface to simplify bulk report management for campus offices.

3. App Store deployment: List the application on Google Play and the Apple App Store to eliminate installation friction.

4. Campus awareness campaigns: Conduct orientation sessions to establish reporting norms among new students.

5. Longitudinal evaluation: Perform a full-semester study to track actual retrieval rates and validate long-term effectiveness.

Video Demonstration : https://youtu.be/GaPjjcuJNhM

Declaration of Conflicting Interests

The authors declare no potential conflicts of interest with respect to the research, authorship and publication of this article.

Funding

The author received no financial support for the research, authorship and publication of this article.

REFERENCES

Almilby, Y. A., & Alkhawaji, R. N. (2024). Haram lost and found. British Journal of Computer Networking and Information Technology, 7(1), 93–107. https://doi.org/10.52589/BJCNIT-4ZV3EY9S

Ar Razi, A. A., & Putra, R. P. (2020). The Hooked Model as communication strategy of "Kembaliin" app as an information media for handling lost and found. Proceedings of the 2nd Social and Humaniora Research Symposium (SoRes 2020). https://doi.org/10.2991/assehr.k.200225.044

Castro, E., David, K., De Silva, K., Roxas, L., & Macaspac, J. (2023). AUFound: Retrieval of misplaced personal belongings through mobile application and web-based management system designed for Angeles University Foundation. https://doi.org/10.46254/an12.20220383

Durachman, Y., Andarmoyo, S., Phong, T., Shankar, K., & Rahim, R. (2020). Real-time alerting for medical science and healthcare. International Journal of Recent Technology and Engineering, 8(2S11), 4157–4161. https://doi.org/10.35940/ijrte.b1600.0982s1119

Gottlieb, O., & Schreiber, I. (2020). Acts of meaning, resource diagrams, and essential learning behaviors: The design evolution of Lost & Found. International Journal of Designs for Learning, 11(1), 151–164. https://doi.org/10.14434/ijdl.v11i1.24100

Hassan, A. N., Rizwan, K., Mehmood, A., & Syed, T. A. (2023). Spatio-temporal modeling and application for efficient online reporting and tracking of lost items during huge crowd gatherings. International Journal of Emerging Technologies, 12(3), 98–106.

Jiang, L., Mao, H., & Kang, Z. (2020). The design of campus lost and found platform based on digital map data. In 2020 International Conference on Computer Information Analytics and Intelligent Systems (CIAIS 2020). Francis Academic Press. https://doi.org/10.25236/ciais.2020.01363

Kanjalkar, P., Pingle, J., Sagar, A., Lohe, R., Patil, S., & Kanjalkar, J. (2024). Lost and found web application. International Journal for Research in Applied Science and Engineering Technology, 12(11), 2098–2104. https://doi.org/10.22214/ijraset.2024.65573

Liu, H., Alali, A., Ibrahim, M., Li, H., Gruteser, M., Jain, S., Dana, K., Ashok, A., Cheng, B., & Lu, H. (2021). Lost and Found! Proceedings of the 23rd International Conference on Human-Computer Interaction with Mobile Devices and Services (pp. 499–500). https://doi.org/10.1145/3458864.3466904

Mahadik, T., Renapurkar, R., & Ganorkar, N. (2021). Digital lost and found item portal. International Journal of Scientific Research in Engineering and Management, 8(3), 44–51.

Mohammed, A., et al. (2023). Institutional identity verification in academic systems.

Ocansey, S. A., Sackitey, G. L., & Agboyi, M. (2024). Investigating the moral behaviour of public transport users in returning lost but found items.

Urban Planning and Transport Research, 12(1). https://doi.org/10.1080/21650020.2024.2305277

Patil, M. R., Patil, R. S., & Kushare, R. P. (2024). A novel approach "FOUNDIT" for lost items. International Journal for Multidisciplinary Research, 6(1). https://doi.org/10.36948/ijfmr.2024.v06i01.12121

Prawira, J., & Saputri, T. R. D. (2023). Lost item identification model development using similarity prediction method with CNN ResNet algorithm. Journal of Autonomous Intelligence, 7(2). https://doi.org/10.32629/jai.v7i2.1381

Republic of the Philippines. (2012). Republic Act No. 10173: Data Privacy Act of 2012. Official Gazette.

Richey, R. C., & Klein, J. D. (2014). Design and development research: Methods, strategies, and issues. Routledge.

Salman, Z. A., & Athab, O. (2022). Smartphone application for managing missed and found belongings. MEST Journal, 10(1), 66–71. https://doi.org/10.12709/mest.10.10.01.08

Si, L., Liu, B., & Fu, Y. (2022). Image recognition algorithm based on artificial intelligence and machine learning. Mobile Information Systems, 2022, 1–5. https://doi.org/10.1155/2022/8125969

Soumya, N. M., Mehak, N. S. Z., Chandra, N. V. V., Saamrat, N. P., & Bandi, N. S. (2024). Campus lost: Connect and recover. International Journal of Engineering Technology and Management Sciences, 8(3), 152–160. https://doi.org/10.46647/ijetms.2024.v08i03.018

Tan, S. Y., & Chong, C. R. (2023). An effective lost and found system in university campus. Journal of Information System and Technology Management, 8(32), 99–112. https://doi.org/10.35631/jistm.832007

Yagi, T., Nishiyasu, T., Kawasaki, K., Matsuki, M., & Sato, Y. (2021). GO-Finder: A registration-free wearable system for assisting users in finding lost objects via hand-held object discovery. In Proceedings of the 26th International Conference on Intelligent User Interfaces (pp. 139–149). https://doi.org/10.1145/3397481.3450664

Zhou, M., Fung, I., Yang, L., Wan, N., Di, K., & Wang, T. (2024). LostNet: A smart way for lost and find. PLOS ONE, 19(10), e0310998. https://doi.org/10.1371/journal.pone.0310998

Zohud, T., & Zein, S. (2021). Cross-platform mobile app development in industry: A multiple case-study. International Journal of Computing, 20(1), 46–54. https://doi.org/10.47839/ijc.20.1.209