

Campushive: A Cloud-Based Platform for Campus Grievance and Lost-And-Found Services

Mrs. Shilpa C¹, Jeevan M², Harshith K³, Aditya A⁴, Chiranth Gowda S⁵

Assistant Professor, Dept of CSE, K.S. Institute of Technology, Bengaluru¹

Student, Dept of CSE, K.S. Institute of Technology, Bengaluru²⁻⁵

ABSTRACT

Modern educational establishments need efficient administration systems to ensure student wellness and hold all parties involved responsible. Classical methods of addressing grievances and managing lost items prove to be less than effective due to various barriers including delays and misunderstandings. In this research, a detailed examination of 12 different grievance management and lost-and-found systems was performed to identify common functional failures. The proposed solution, namely CampusHive, comprises of a cloud mobile application which applies binary logistic regression to prioritize complaints, a prototype approach to managing lost items, and unique mechanism of giveaways.

Keywords: Grievance Management, Logistic Regression, Property Lifecycle, CampusHive, Educational Accountability, Cloud Computing.

1. INTRODUCTION

University students are often viewed as a disadvantaged group unable to address their issues related to educational, psychological, and infrastructural aspects, as they lack proper communication channels with administrative officials. Problems range from building degradation including water leakage and electricity outages to much more serious cases such as harassment or disputes during the studying process. Creating an inclusive environment where every student can participate actively avoid any form of discrimination becomes essential.

Currently, lost-property management systems are based mostly on popular social networking services, such as Telegram or WhatsApp public accounts. These tools provide wide coverage but create numerous privacy-related risks by distributing personal data widely regardless of identification of claims' legitimacy. CampusHive system integrates several components into a single mobile application to solve these problems.

2. LITERATURE REVIEW

A. A Survey on Student Grievance Redressal System

The evaluation of the technologies of 12 available systems for handling students' grievances showed the presence of certain developments and restrictions in the field.

1. Prajapat et al.: Comparing manual and computerized processes, inadequate network connectivity was recognized to cause delays in processing complaints.

2. Viral Patel et al.: An app based on the technologies of JavaScript and NoSQL databases was built; nevertheless, the algorithms applied were impractical for implementation into real software.

3. Venkatesan et al.: A statewide grievance resolution platform was suggested by the authors; yet, the technology of the platform was vulnerable to manipulations from outside.

4. Amir Shareghi Najar et al. (Çokluk, 2010): The main restriction in a service-oriented technology was the possibility of submitting false complaints.

5. Buldak et al.: To reduce manual data entry, a website was designed; still, the system did not provide users with access to their complaint history.

6. Radhakrishnan et al.: Designing an intelligent management system with MongoDB database, inefficiency and cumbersome user interface were identified.

B. UTHM Lost and Found Management System

Current technologies for managing lost property are impractical for use in universities.

- Commercial Services: Crowdfind, Chargerback, iLost for Business offer tools of account verification and item registration for easy recovery of lost property. However, the majority do not allow for "giving away" property held for 30–90 days.

• Shahzan & Arbai (2025): The paper suggests designing a web-based management information system that would allow automating identification process and donating items to charities.

3. OBJECTIVES

1. To develop a centralized cloud-based platform that integrates grievance redressal, lost-and-found management, and campus communication into a single digital ecosystem.
2. To implement a machine learning-based prioritization mechanism using Binary Logistic Regression for identifying and ranking complaints based on urgency and contextual factors.
3. To provide a transparent grievance management workflow that enables students to submit, track, and receive real-time updates regarding complaint status.
4. To incorporate a community-driven voting mechanism that dynamically influences complaint prioritization and improves decision-making efficiency.
5. To design a structured lost-and-found management system that supports item reporting, user interaction, verification, notification handling, and time-based escalation.
6. To ensure secure and scalable communication between users, administrators, and system modules through a multi-tiered cloud-based architecture.
7. To improve institutional accountability, response efficiency, and overall student engagement through an intelligent and user-centric campus management system.

4. MATHEMATICAL BASIS: LOGISTIC REGRESSION

Binary Logistic Regression helps distinguish the intelligent approach of CampusHive for prioritizing incidents.

A. Concept and Estimation

This mathematical tool, or logistic regression, is used for classification purposes by dividing objects according to dependent dichotomous variable values into two groups (High Priority and Low Priority). Linear regression attempts to estimate the value of a continuous variable, whereas logistic regression attempts to predict the probability of occurrence of an event within the range [0,1].

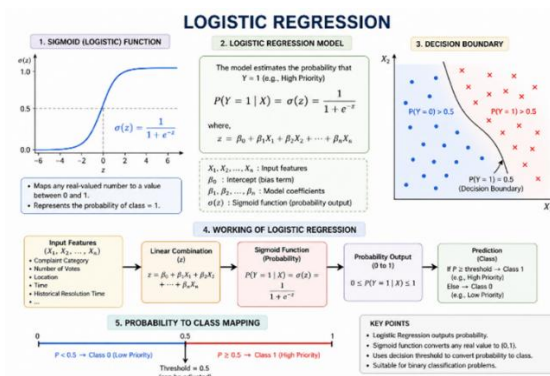


Fig 4.1: Binary Logistic Regression Probability Curve

B. Selection of Variables for CampusHive

Based on the study conducted by Çokluk 2010), the algorithm strives to provide adequate predictions with the help of fewer predictor variables. These include:

- Complaint Category & Description: Dichotomous attributes (Academic and Infrastructure)
- User Votes: Urgency attributes offered by users.
- Contextual Variables: Urgency-related context (Location & Time)
- Historical Resolution Time: Historical data about the resolution time of similar complaints.

The goodness of fit between the model and observed data is analyzed with the help of the Hosmer-Lemeshow test; thereby allowing detecting problems at the campus.

5. DESIGN AND ARCHITECTURE OF CAMPUSHIVE SYSTEM

A. Multi-Tiered Architecture

Campushive utilizes a multi-layer architecture to attain flexibility, scalability, efficiency in real-time communication, and modularity. The architecture consists of five related layers/modules, each with a designated role while coordinating with the rest via Application Programming Interfaces (APIs).

1. Client Layer: The client layer is implemented in the form of a cross-platform application using Flutter. The client layer acts as the interface between the users and the system and allows users to perform various actions in the system, such as submitting complaints and reporting lost and found items. Updates from the system can be seen through the client layer. Any activities within the client layer are sent to the back-end server in the form of API requests.

2. Notification System: Notification system is a cloud-based module tasked with sending updates to users about complaints and other related information, such as comments. The module helps in sustaining user involvement in the application process through providing relevant updates.

3. Application Layer: The application layer is developed with the use of Python Flask framework and acts as the logic layer in the system. This layer includes a RESTful API layer, authentication module, and business logic module, which manages the lifecycle of complaints and lost-and-found items.

4. Machine Learning Module: Machine learning module involves independent development and is aimed at prioritizing the complaint prediction. Priority predictions are determined using an algorithm that uses data provided by the application layer.

5. Database Layer: Database layer constitutes a MySQL database that includes structured data on users, complaints, votes, lost-and-found items, comments, and notifications.

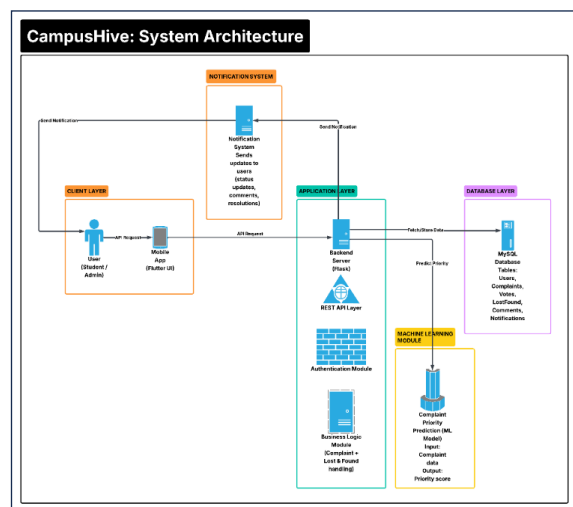


Fig 5.1: CampusHive Multi-Tier System Architecture

B. Entity Relationship Diagram (ERD)

The ER diagram for CampusHive is developed using Chen notation that includes all the main entities, relationships, and attributes of the system.

1. User Hierarchy: The basic USER entity is related to STUDENT and ADMIN in an ISA relationship. Attributes of the STUDENT entity include USN, department, semester, section, and contact number.

2. Core Entities: Two entities considered as essential for the system include the COMPLAINT (attributes:

complaint_id, title, description, status, image) and LOSTFOUND (attributes: item_id, title, type, status, location).

3. Interaction Entities:

- VOTE (attributes: vote_id) is related to CASTS (User-Vote) and FOR (Vote-Complaint).
- COMMENTS (attributes: comment_id, message, timestamp) are related to WRITES (User-Comments) and GETS (LostFound-Comments).

4. Priority Entity: The PRIORITY entity (attributes: priority, vote_count, ml_score) is related to the COMPLAINT through HAS relationship.

5. System Entities:

- NOTIFICATION (attributes: notification_id, message, type, status) is related to the USER through the RECEIVES relationship.
- ACTIONS (attributes: action_id, action_type, timestamp) is related to the ADMIN through the PERFORMS relationship.

6. Main Relationships: POSTS (User-Complaint), CASTS (User-Vote), FOR (Vote-Complaint), WRITES (User-Comments), GETS (LostFound-Comments), HAS (Complaint-Priority), RECEIVES (User-Notification), PERFORMS (Admin-ACTIONS).

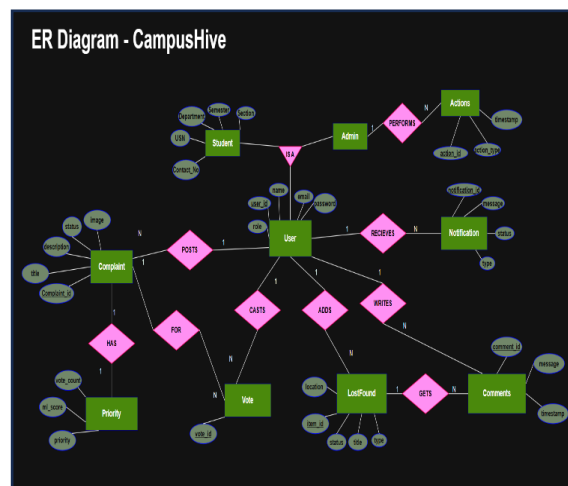


Fig 5.2: Entity-Relationship Diagram (ERD) of CampusHive

6.SYSTEM WORKFLOWS AND METHODOLOGY

A. Integrated Development Methodology

CampusHive uses Structured Prototype Development Methodology with six stages of development: Planning,

Requirements Analysis, Design, Prototyping, Testing, and Implementation.

B. Logic of the complaint module

The logic of the complaint module follows a predefined workflow, moving through steps from authentication to processing complaints, involving system-based, community-based, and administrative operations.

1. Authentication and submission of complaints: At first, the user gets authenticated in the application; then, he/she files a complaint. In case of failed user authentication, the user is redirected to re-authenticate in the application. Once the user is authenticated, he/she submits the complaint with additional information, such as pictures, category, description, and location of the complaint.

2. Storing data and Initial processing: After submitting the complaint, it gets stored in the database using the backend; the processing of the complaint starts automatically.

3. Priority prediction based on machine learning: Using machine learning techniques, the priority of the complaint is estimated first; then, the estimated priority gets used to determine the position of the complaint among other submitted complaints.

4. Voting by community members and dynamic change of priority: The complaint is shared with all the users, who get the opportunity to either support or reject the complaint by voting. Voting is considered a positive signal that together with machine learning estimates can be used to update the priority.

5. Processing complaints by administrators: All complaints are listed in the administrator's panel, sorted by priority. Administrators examine the complaints and decide whether the problem should be solved or kept in the queue of pending complaints.

6. Complaint status change: When it becomes necessary to investigate the complaint or solve the described problem, the administrator assigns a new status to the complaint, such as "In Progress" or "Resolved."

7. Notification to complainant: When there is a change in complaint status, the notification is sent to the owner of the complaint.

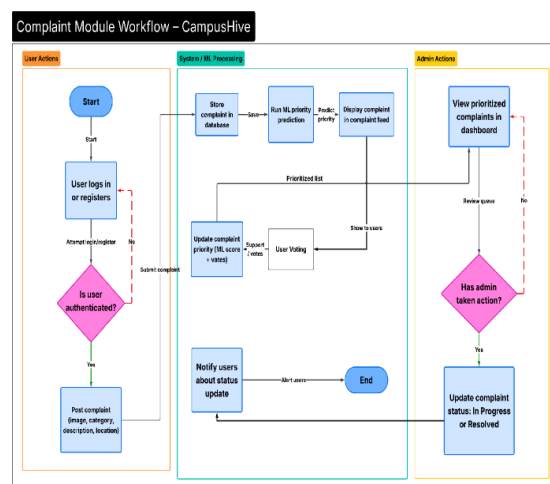


Fig 6.1: Complaint Management Workflow in CampusHive

C. Lost and Found Management Workflow

There is a particular logic associated with the management of properties in CampusHive, depending on user participation, verification processes, notifications, and time-based escalation.

1. Identification and Posting: Firstly, in terms of the workflow, its starting point is that the user undergoes an authentication process followed by his or her decision whether to post something as lost or found. Once the decision is made, the user provides information about the object in question and posts it. At this point, the user waits for other participants to respond to their message.

2. User Interaction: Next, the participants start exchanging messages and comments with the purpose of identifying a match between the found/lost item and other properties. Users exchange messages multiple times until they get ready to proceed to the next step.

3. Verification and Resolution of the Case: Once the verification process is over, the system makes sure whether the found property was verified successfully or not. If yes, the item becomes resolved and appropriate notifications are sent to the participants.

4. Notification Handling: All the participants receive notifications throughout the process.

5. Time-Based Escalation: If within seven days of the posting there are no results from the discussion, the matter gets escalated to the administrator.

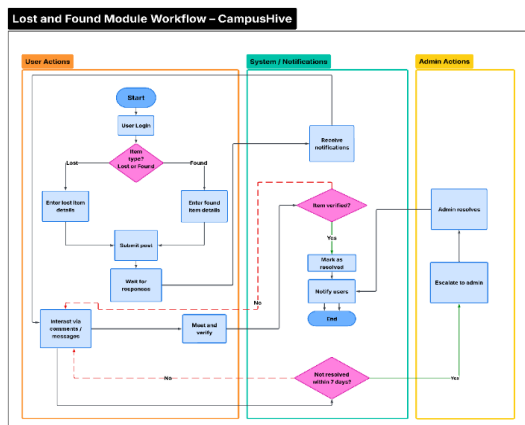


Fig 6.2: Lost and Found Management Workflow

7. COMPARATIVE ANALYSIS

Table 1: Comparison analysis

Features	Traditional Web Portals	Informal Social Media	Campus Hive (Proposed)
Technology	PHP, MySQL	Messaging Applications	Flutter, Python Flask
Prioritisation	Manual	None	ML-Powered (Logistic Regression)
User Tracking	Often Lacking	Manual Search	Real-time Tracking
Verification	Basic Verification	Unreliable	Admin-Verified Claimants
Storage Solution	Bulk Record Storage	No Structured Storage	Automated Giveaway Module

8. CONCLUSION

Transitioning from isolated and manual campus system management to central "hive" management, utilizing cloud computing, is critical to increase efficiency and transparency in current times. Conventional systems of managing grievances, lost and found items face problems

with delays, lack of transparency, and minimal interaction of users. However, when the mechanisms for dealing with grievances, lost-and-found items, and holding events are combined into a single technological product, CampusHive becomes an intelligent solution to these challenges. Also, the introduction of ML in the task prioritization helps to identify and address important issues effectively. Moreover, the users' participation in voting and other interactive functions adds more transparency and responsibility.

ACKNOWLEDGEMENT

Thus, we would like to express our sincere appreciation to Mrs. Shilpa C for her significant contribution to the realization of the idea and implementation of this project. We are grateful to her for devoting her precious time. Next, we would like to thank the honorable faculty members of KSIT for their whole-hearted cooperation and support.

REFERENCES

[1] Er. A. Meshram et al., "A Survey on Student Grievance Redressal System," *IJRASET*, Feb 2023.

[2] N. F. Shahzan and N. Arbaiy, "UTHM Lost and Found Management System," *AITCS*, 2025.

[3] Ö. Çokluk, "Logistic Regression: Concept and Application," *Educational Sciences*, 2010.

[4] V. Bhutada and P. Deshmukh, "Students' Grievance System for Educational Institutions," *JETIR*, 2023.

[5] Shaligram Prajapat, Vaibhav Sabharwal, and Varun Wadhvani, "A Prototype for Grievance Redressal System," 2018.

[6] Mukesh Buldak et al., "An Online Grievance Redressal System," *IJRTE*, 2019.

[7] Crowdfind, "Lost and Found Management Software," Available: <https://crowdfind.com>

[8] Chargerback, "Lost and Found Recovery System," Available: <https://www.chargerback.com>

[9] iLost for Business, "Lost Property Management Platform," Available: <https://ilost.co/business>

[10] "Web Personalization using Amalgamation of Web Navigational Patterns & User Profiles", Sowbhagya M. P., Yogish H. K., G. T. Raju, published in

INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING, IJISAE, 2024, 12(1), 377–384.

[11] “AI-Augmented Metagenomic Diagnostic for Early Detection of Emerging Microbial Pathogens “, published in International Journal of Engineering Technology and Management Sciences, V.Manjula, Divya C P, Shilpa C, G. Satheesh Kumar, A. Lakshmannath,, ijetms.in Issue:3 Volume No.10 May - June – 2026 DOI:10.46647/ijetms.2026.v10i03.024 ISSN: 2581-4621

[12] “User Profiling for Web Personalization”, Sowbhagya M. P., Yogish H. K., G. T. Raju, published in 2022 IEEE International Conference on Data Science and Information System.

[13] “Perception based User Profiles for Web Personalization”, Sowbhagya M. P., Yogish H. K., G. T. Raju, published in International Journal on Recent and Innovation Trends in Computing and Communication, IJISAE, 2024, 12(1), 377–384. ISSN: 2321-8169 Volume: 11 Issue: 7s DOI: <https://doi.org/10.17762/ijritcc.v11i7s.6986>.

[14] “Extraction of users' behavior patterns for web personalization”, Sowbhagya M. P., Yogish H. K., G. T.

Raju, published in European chemical bulletin, 2023,12(10), 6995-7006.

[15] “Knowledge Discovery from Web Data for Web Personalization”, Sowbhagya M. P., Yogish H. K., G. T. Raju, published in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-2S, December 2019.