Software Requirements Specification

for

Lost and Found

Version <1.0>

Prepared by

Bhoomi	**22103001**
Kartik Sharma	**22103100**
Prasuk Jain	**22103104**
Hardik Ahuja	**22103106**

Instructor: Dr. Rajesh Bhatia

Course: Software Engineering

Lab Section: *CSE 6*

Date: 27/09/2024

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Bhoomi, Prasuk Jain, Kartik Sharma, Hardik Ahuja	Initial version of the project.	27/09/2024

1 Introduction

The Lost and Found Platform for College is a digital solution aimed at streamlining the process of reporting and recovering lost items within a college campus. With a large campus and diverse activity zones, it becomes difficult for students and staff to locate misplaced belongings. This platform provides a user-friendly interface where users can report lost or found items, including descriptions and images.
In this document, the reader will find detailed information about the platform's purpose, scope, and requirements. It covers both functional and non-functional requirements, including system interfaces, performance, and security considerations. The document also outlines the design constraints, assumptions, dependencies, and use cases to guide the development team in creating a reliable and secure solution for the college environment.

1.1 Document Purpose

This Software Requirements Specification (SRS) outlines the requirements for the **Lost and Found Platform for College**, Version 1.0. The platform is designed to facilitate the reporting, tracking, and recovery of lost and found items within a college campus. By utilizing an AI-powered matching system, it enhances the chances of reuniting users with their lost belongings. The SRS provides a comprehensive description of the software's intended functionality, behaviour, and performance characteristics.

This document defines the entire system, including both the frontend user interfaces and the backend AI-driven matching system. It also specifies the external interfaces, hardware and software requirements, and key assumptions guiding the development. By detailing functional and non-functional requirements, this SRS serves as a blueprint for developers, testers, and stakeholders to ensure the system meets its objectives and provides a smooth user experience.

1.2 Product Scope

The **Lost and Found Platform for College** is a web-based software designed to help students and staff report and track lost and found items within a college campus. The platform allows users to submit detailed reports about lost or found items, including descriptions, images, and relevant details such as the location and time of the incident. Using AI-based natural language processing (NLP), the system matches lost and found reports to notify users when a potential match is found, thus streamlining the process of item recovery.

The primary benefit of this platform is its ability to quickly and efficiently connect lost items with their rightful owners through automated matching. It reduces the manual effort of tracking lost belongings and provides a user-friendly interface that is easy to navigate. The platform also ensures the privacy and security of user data, with authentication required for users to access the system. The objective is to create an intuitive, secure, and efficient tool for improving the lost and found process, ultimately saving users time and reducing the stress associated with recovering lost items.

1.3 Intended Audience and Document Overview

This SRS is intended for various stakeholders, including **developers**, **project managers**, **testers**, **documentation writers**, and the **client (professor and college administration)**. Developers will use this document to understand the system's functional and non-functional requirements, while project managers will ensure the project stays on scope and schedule. Testers will validate that the platform works as expected based on the requirements, and the client will review the document to confirm that the platform aligns with their needs and objectives.

The document is organized into several sections, starting with an introduction to the system's purpose and scope. It proceeds with a detailed description of the platform's functionality, design constraints, and dependencies. The specific requirements section outlines both the external interface and functional requirements, followed by use case models. Lastly, appendices provide additional reference material such as a data dictionary and group log. Readers are advised to begin with the introduction and overall description for context, then dive into the detailed requirements relevant to their roles.

1.4 Definitions, Acronyms and Abbreviations

Below is a list of the abbreviations and acronyms used in this Software Requirements Specification (SRS):
- **AI**: Artificial Intelligence
- **API**: Application Programming Interface
- **DB**: Database
- **NLP**: Natural Language Processing
- **UI**: User Interface
- **SRS**: Software Requirements Specification
- **UX**: User Experience

These terms are specific to this project and are used to describe key technologies, components, and interfaces that form the Lost and Found Platform for College.
4o

1.5 Document Conventions

This Software Requirements Specification (SRS) follows the IEEE formatting standards to maintain consistency and readability. The following conventions were adhered to throughout the document:
- **Font**: The document uses **Arial**, with a font size of **11** or **12** for the body text.
- **Spacing**: The document is formatted with **single spacing** to ensure readability while maintaining a compact layout. Margins are set to **1 inch** on all sides, as per IEEE guidelines.
- **Section and Subsection Titles**: All section and subsection titles follow the template provided, using a clear and hierarchical numbering system (e.g., 1, 1.1, 1.2) for easy navigation.
- **Comments**: Italics are used for comments or placeholder text to distinguish them from the final content.

- **Naming Conventions**: All system components, interfaces, and functions are named descriptively to avoid ambiguity, and acronyms are defined upon first use and listed in Section 1.4.

These standards and conventions help ensure the document is clear, structured, and easy to follow for all stakeholders.

1.6 References and Acknowledgments

The following documents and resources are referenced throughout this Software Requirements Specification (SRS):

1. ***IEEE Standards for Software Requirements Specifications****: This document provides guidelines and best practices for creating SRS documents.*
2. ***CS1503 Software Engineering Course Syllabus****: Contains course objectives and requirements relevant to the project.*
3. ***Natural Language Processing Resources****: Documentation for NLP models such as GPT-3 and BERT, which will be used for the AI matching system.*
4. ***User Interface Style Guide****: A style guide detailing the design principles and standards to be followed for the platform's user interface (specific link to be provided if available).*
5. ***Project Vision and Scope Document****: Outlines the overall goals and deliverables of the Lost and Found Platform project.*

These references provide foundational knowledge and context that supports the requirements outlined in this SRS.

2 Overall Description

2.1 Product Overview

The **Lost and Found Platform for College** is a new, self-contained product designed to address the common problem of misplaced items in a college environment. Unlike existing systems, which often rely on manual reporting methods or physical bulletin boards, this platform leverages modern technology to streamline the process of reporting and recovering lost and found items. By utilizing an AI-powered matching system, it automates the identification of potential matches between lost and found items, significantly improving the chances of recovery.

The platform is intended to function independently but can be integrated with existing college systems, such as student databases for user authentication and notifications. This integration will ensure that only authorized users can access the platform, maintaining a secure environment for sharing personal information. The requirements outlined in this SRS are designed to support a user-friendly interface, robust backend processing, and efficient data management, making it a valuable tool for both students and staff.

2.2 Product Functionality

The **Lost and Found Platform for College** is designed to perform several key functions that enhance the user experience and streamline the process of reporting and recovering lost items. The major functions of the system include:
- **Lost Item Reporting**: Users can submit detailed reports of items they have lost, including descriptions and optional images.
- **Found Item Posting**: Users can register found items by providing descriptions and images, ensuring accurate records for potential matches.
- **AI-Based Matching**: The system utilizes natural language processing (NLP) algorithms to automatically match lost items with found items based on descriptions, alerting users when a match is identified.
- **User Authentication**: Secure login using college credentials ensures that only authorized users can access and utilize the platform.
- **Notifications**: Users receive alerts via email or in-app notifications when potential matches are detected, facilitating timely communication.
- **User-Friendly Interface**: The platform offers an intuitive interface for easy navigation and interaction, making it accessible for all users.
- **Data Management**: Efficient storage and management of user reports and item data in a secure database, ensuring data integrity and privacy.

2.3 Design and Implementation Constraints

The development of the **Lost and Found Platform for College** will be subject to several design and implementation constraints that may limit the options available to developers:

- **Technology Stack**: The frontend will be developed using **React.js** or **Angular**, while the backend will use **Node.js** and **Express**. The database will be either **MongoDB** or **PostgreSQL**. Developers must adhere to these technologies to ensure compatibility and maintainability.
- **COMET Method**: The project will follow the **COMET** (Conceptual, Operational, Mechanistic, Evaluation, and Technical) method for software design, which emphasizes a structured approach to system design and development. This method ensures clarity in requirements and design specifications(SRS-template).
- **UML Modelling Language**: Unified Modelling Language (UML) will be utilized for creating system diagrams, including use case diagrams and class diagrams. This modelling language helps in visualizing system architecture and interactions, which is essential for communicating design choices effectively(SRS-template).
- **User Authentication**: The system will require integration with the college's existing authentication mechanisms to ensure secure access. This may limit the choice of authentication methods to those that are already in use at the institution.
- **Performance Limitations**: The system must handle multiple simultaneous user requests efficiently, with an expected response time for user actions (e.g., submitting a report or receiving a notification) not exceeding 5 seconds. This may limit the complexity of the operations that can be performed concurrently.
- **Data Privacy Regulations**: Compliance with relevant data privacy regulations (e.g., GDPR, FERPA) will restrict how user data is collected, stored, and processed. Developers must ensure that all data handling practices adhere to these regulations.
- **Network Protocols**: The system will operate over standard web protocols (HTTP/HTTPS). Developers must ensure that all data transmissions are encrypted to protect user information and maintain data integrity.
- **Security Considerations**: Given that the platform will handle sensitive user information, developers must implement robust security measures, including data encryption, secure authentication, and regular security assessments to identify and mitigate vulnerabilities.

2.4 Assumptions and Dependencies

*The development of the **Lost and Found Platform for College** is based on several key assumptions that could significantly affect the project requirements and design if they prove to be incorrect or change:*

1. ***User Base Access***: *It is assumed that all potential users (students and staff) will have access to internet-enabled devices and a stable internet connection to use the platform effectively. If this assumption changes, it may impact user engagement and the overall success of the platform.*
2. ***Integration with College Systems***: *The project assumes seamless integration with the college's existing authentication and user management systems. Any issues with these integrations could hinder the functionality of user login and data access.*
3. ***AI Matching Accuracy***: *It is assumed that the natural language processing (NLP) algorithms employed in the AI matching system will be effective in accurately identifying matches between lost and found items. If the algorithms underperform, it may lead to user dissatisfaction.*

4. **Data Privacy Compliance**: *The project assumes that all data handling practices will comply with relevant data privacy regulations (e.g., GDPR, FERPA) as defined by the college. Changes in these regulations or non-compliance could pose legal risks.*
5. **Third-Party Services**: *The platform assumes the availability and reliability of third-party services, such as cloud hosting or email notification systems, which are essential for operational functionality. If these services experience downtime or discontinuation, it may impact the platform's performance.*
6. **User Engagement**: *It is assumed that users will actively engage with the platform by reporting lost and found items. Low engagement could affect the effectiveness of the AI matching system and the overall utility of the platform.*
7. **Technical Support and Maintenance**: *The project assumes that the college will provide ongoing technical support and maintenance for the platform after deployment. Any changes in support availability could affect the long-term sustainability of the system*

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The **Lost and Found Platform for College** features a user-friendly web interface designed to facilitate easy interaction for both students and staff. The main components of the user interface include:

1. **Home Page**: The landing page provides a clear overview of the platform's purpose and quick access to the primary functionalities, such as reporting lost items or posting found items.
2. **Lost Item Reporting Form**:
   o Users can fill out a simple form to report lost items, which includes fields for item description, location of loss, time of loss, and an option to upload an image of the item.
   o The form is designed to be intuitive, with dropdown menus for location and a character counter for description length to guide users.
3. **Found Item Posting Form**:
   o Similar to the lost item form, this section allows users to submit details about items they have found, including descriptions, locations, and images.
   o It provides a straightforward interface to ensure all necessary information is collected for effective matching.
4. **Matching Alerts**:
   o Users will receive notifications via email or within the platform when a potential match is identified.
   o Notifications will include a summary of the matched item, a link to view details, and contact options for the parties involved.
5. **User Authentication**:
   o Users will log in using their college credentials, ensuring secure access to the platform.
   o The authentication process is straightforward, with options for password recovery.

Interaction Method:
- The user interface will be web-based and accessible on desktops and mobile devices.
- Users will interact with the platform primarily through **clickable buttons** and **forms**.
- Touchscreen devices will support tap interactions for form submission and navigation, while keyboard input will be available for text fields.

3.1.2 Hardware Interfaces

The **Lost and Found Platform for College** primarily operates as a web-based application, meaning it interacts with various hardware devices through standard web interfaces. The following are the primary hardware interfaces supported by the system:

1. **User Devices**:
   o **Desktops/Laptops**: Users can access the platform via standard web browsers on personal computers. The platform is optimized for various screen sizes and resolutions to ensure usability.

o **Tablets and Smartphones**: The platform will also be accessible on mobile devices, with a responsive design that adapts to smaller screens. Users can interact via touch input or virtual keyboards.
2. **Web Servers**:
    o The application will be hosted on web servers that handle HTTP/HTTPS requests from user devices. The servers will manage the application's backend logic, including data processing, user authentication, and database interactions.
3. **Database Servers**:
    o The system will communicate with database servers (e.g., MongoDB or PostgreSQL) to store and retrieve user data, lost and found item reports, and notifications. This interaction will involve data queries, updates, and storage operations.
4. **Network Infrastructure**:
    o The platform will rely on standard network hardware, such as routers and switches, to facilitate data transmission between user devices, web servers, and database servers. Reliable network connectivity is essential for the platform's performance and user experience.

3.1.3 Software Interfaces

1. **APIs**: Integration with email APIs (e.g., SendGrid) for notifications, and possibly REST APIs for future mobile app integration.
2. **Database**: Relational or NoSQL database (e.g., MySQL, PostgreSQL, MongoDB) for storing reports.
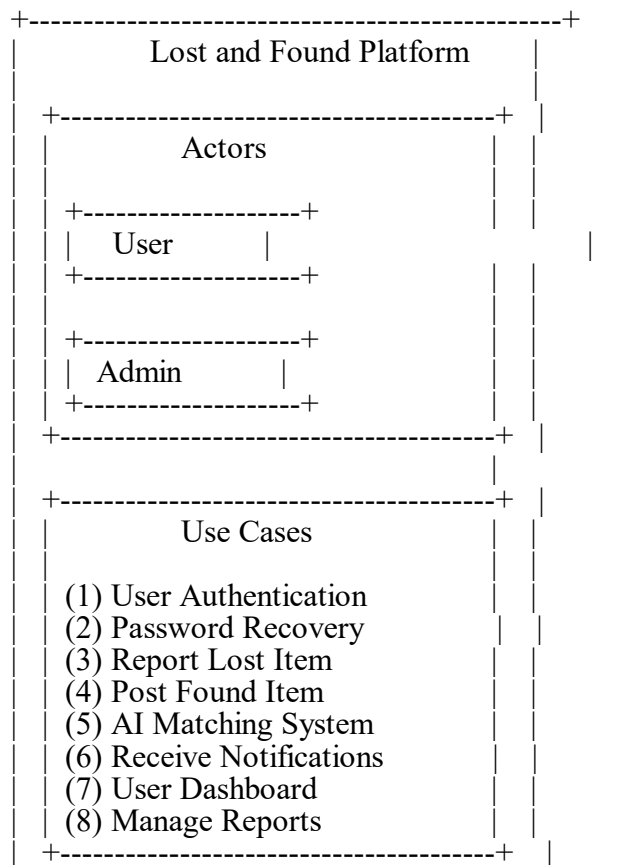
3.2 Functional Requirements

The functional requirements for the **Lost and Found Platform for College** detail the specific behaviours and capabilities that the system must provide. These requirements capture the intended services and tasks that users will perform within the platform.
1. **User Authentication**
    o **F1**: The system shall allow users to log in using their college credentials.
    o **F2**: The system shall provide a password recovery option for users who forget their passwords.
    o **F3**: The system shall ensure that only authenticated users can access the reporting and notification features.
2. **Lost Item Reporting**
    o **F4**: The system shall allow users to submit reports of lost items.
    o **F5**: The system shall require users to provide a description of the lost item, including fields for item type, colour, size, and unique identifiers.
    o **F6**: The system shall provide an option for users to upload images of the lost item.
    o **F7**: The system shall prompt users to specify the location and time of the loss.
3. **Found Item Posting**
    o **F8**: The system shall allow users to submit reports of found items.
    o **F9**: The system shall require users to provide a description of the found item, including fields for item type, color, size, and unique identifiers.
    o **F10**: The system shall provide an option for users to upload images of the found item.
    o **F11**: The system shall prompt users to specify the location where the item was found.
4. **AI-Based Matching**
    o **F12**: The system shall utilize AI algorithms to compare lost and found item reports based on descriptions and images.
    o **F13**: The system shall notify users of potential matches via email or in-app notifications when a match is found.

          o   **F14**: The system shall provide a summary of the matched items, including links to view details.
5. **User Notifications**
          o   **F15**: The system shall send notifications to users when their reported items match found items.
          o   **F16**: The system shall allow users to view their notification history within their user dashboard.
6. **User Dashboard**
          o   **F17**: The system shall provide a user dashboard that allows users to view their submitted lost and found reports.
          o   **F18**: The system shall allow users to edit or delete their submitted reports.
          o   **F19**: The system shall display notifications related to matches and other relevant alerts.
7. **Data Management**
          o   **F20**: The system shall securely store user data and reports in the database.
          o   **F21**: The system shall ensure data integrity and protect user privacy in compliance with applicable data protection regulations.

3.3 Use Case Model

```
+-------------------------------------------------+
|              Lost and Found Platform            |
|                                                 |
|   +-----------------------------------------+   |
|   |                Actors                   |   |
|   |                                         |   |
|   |   +--------------------+                |   |
|   |   |    User            |                |   |
|   |   +--------------------+                |   |
|   |                                         |   |
|   |   +--------------------+                |   |
|   |   |   Admin            |                |   |
|   |   +--------------------+                |   |
|   +-----------------------------------------+   |
|                                                 |
|   +-----------------------------------------+   |
|   |               Use Cases                 |   |
|   |                                         |   |
|   |  (1) User Authentication                |   |
|   |  (2) Password Recovery                  |   |
|   |  (3) Report Lost Item                   |   |
|   |  (4) Post Found Item                    |   |
|   |  (5) AI Matching System                 |   |
|   |  (6) Receive Notifications              |   |
|   |  (7) User Dashboard                     |   |
|   |  (8) Manage Reports                     |   |
|   +-----------------------------------------+   |
```

Use Case Descriptions

1. **User Authentication**: Users log in using their college credentials to access the platform.
2. **Password Recovery**: Users can recover their password if forgotten through email verification.
3. **Report Lost Item**: Users submit a report detailing their lost items.
4. **Post Found Item**: Users submit a report for items they have found.
5. **AI Matching System**: The system matches lost items with found items using AI algorithms.
6. **Receive Notifications**: Users are notified via email or within the platform when potential matches are identified.
7. **User Dashboard**: Users can view their submitted reports, notifications, and manage their account settings.
8. **Manage Reports**: Users can edit or delete their submitted reports from their dashboard.

This use case model visually represents the interactions between the actors (Users and Admins) and the various functionalities of the Lost and Found Platform, providing a comprehensive overview of the system's capabilities.

3.3.1 Use Case #1 Submit Lost Item Report

- **Purpose**: To allow users to submit reports for lost items by providing necessary details and images.
- **Requirements Traceability**: Linked to functional requirement FR2.
- **Priority**: High – Reporting lost items is essential to the system's core functionality.
- **Preconditions**: The user must be logged into the system.
- **Postconditions**: The system stores the lost item report in the database, pending approval by an admin.
- **Actors**: Student/Staff (user).
- **Flow of Events**:

    1. The user logs into the system.

    2. The user navigates to the "Report Lost Item" section.

    3. The user fills out the required fields (description, location, date, etc.).

    4. The user uploads an image of the lost item.

    5. The system stores the report and notifies the user.

- **Notes/Issues**: Ensure that image uploads are validated for size and format (e.g., JPEG, PNG).

3.3.2 Use Case #2 Submit Found Item Report

- **Purpose**: To allow users to report found items and provide relevant details for identification.

- **Requirements Traceability**: Linked to functional requirement FR3.

- **Priority**: High – Reporting found items is a core part of the platform's goal.

- **Preconditions**: The user must be logged into the system.

- **Postconditions**: The system stores the found item report in the database, pending approval by an admin.

- **Actors**: Student/Staff (user).

- **Flow of Events**:

    1. The user logs into the system.

    2. The user navigates to the "Report Found Item" section.

    3. The user fills out the required fields (description, location, date, etc.).

    4. The user uploads an image of the found item.

    5. The system stores the report and notifies the user.

- **Notes/Issues**: The system should prevent duplicate reports for the same found item by checking keywords and images.

3.3.3 Use Case #3 Search for Lost/Found Items

- **Purpose**: To allow users to search the platform for reported lost or found items using various filters.

- **Requirements Traceability**: Linked to functional requirement FR4.

- **Priority**: High – The search function is crucial for users to locate their items.

- **Preconditions**: The user must be logged into the system.

- **Postconditions**: The system displays matching results based on the user's search criteria.

- **Actors**: Student/Staff (user).

- **Flow of Events**:

    1. The user logs into the system.

    2. The user navigates to the search bar or filter section.

    3. The user inputs keywords or uses filters like location or date.

    4. The system retrieves and displays matching reports.

    5. The user can view details of each report and request admin assistance if necessary.

- **Notes/Issues**: The system should implement autocomplete for common keywords and locations.

3.3.4 Use Case #4 Item Match Notification

- **Purpose**: To notify users when a lost item matches a found item based on descriptions and filters.

- **Requirements Traceability**: Linked to functional requirement FR6.

- **Priority**: Medium – Matching helps streamline item recovery but is secondary to item reporting.

- **Preconditions**: There must be matching reports in the system.

- **Postconditions**: Both the user who lost an item and the user who found an item are notified of a potential match.

- **Actors**: Student/Staff (user), System.

- **Flow of Events**:

    1. The system regularly checks for matching reports.

    2. When a match is found, the system sends a notification to both the lost and found report submitters.

    3. The users can request admin assistance to verify the match.

    4. The admin facilitates contact between the users.

- **Notes/Issues**: Matching algorithms should be optimized to prevent false matches based on vague descriptions.

4 Other Non-functional Requirements

4.1 Performance Requirements

The performance requirements for the **Lost and Found Platform for College** are designed to ensure that the system operates efficiently and meets user expectations under various conditions. Below are the specific performance requirements:

1. **Login Response Time**:
   o **P1**: The system shall authenticate user credentials and respond to login requests within **3 seconds** under normal operating conditions. This ensures a smooth user experience and reduces frustration during login.
2. **Item Reporting and Posting**:
   o **P2**: The system shall allow users to submit reports for lost and found items, with confirmation of successful submission displayed within **2 seconds** after the user clicks the submit button. Quick feedback encourages user engagement and confidence in the system.
3. **AI Matching Processing Time**:
   o **P3**: The system shall process item matching requests using the AI algorithms within **5 seconds** after a user submits a lost or found item report. This rapid processing is critical to provide timely notifications and maintain user satisfaction.
4. **Notification Delivery**:
   o **P4**: The system shall send email notifications for potential matches to users within **10 seconds** of identifying a match. Prompt notifications increase the chances of successful item recovery and enhance user trust in the platform.
5. **Dashboard Load Time**:
   o **P5**: The user dashboard shall load within **4 seconds** after authentication. Fast loading times are essential for user retention and satisfaction, as users expect quick access to their information.
6. **Concurrent Users**:
   o **P6**: The system shall support a minimum of **100 concurrent users** without degradation in performance. This requirement is vital to accommodate peak usage times, such as during the beginning of the semester or following campus events.
7. **Data Retrieval Time**:
   o **P7**: The system shall retrieve and display lost and found item data from the database within **2 seconds** for any user query. Efficient data retrieval ensures that users can quickly find relevant information.

These performance requirements provide a framework for developers to ensure that the Lost and Found Platform functions efficiently and meets user expectations, contributing to a positive overall experience.

4.2 Safety and Security Requirements

The Lost and Found Platform for College must adhere to several safety and security requirements to protect users' data and ensure the platform's responsible use. The following requirements outline the necessary safeguards, actions to be prevented, and compliance with regulations:

User Authentication Security:

S1: The system shall require strong password policies, enforcing a minimum password length of 8 characters with a combination of upper-case letters, lower-case letters, numbers, and special characters.
S2: The system shall implement multi-factor authentication (MFA) to enhance security during the login process, requiring users to verify their identity through a secondary method (e.g., SMS or email verification).
Data Encryption:

S3: All user data, including passwords and personal information, shall be encrypted during transmission using TLS (Transport Layer Security) to prevent interception by unauthorized parties.
S4: Sensitive data stored in the database shall be encrypted to protect it from unauthorized access or data breaches.
User Privacy Protection:

S5: The platform shall comply with applicable data protection regulations, such as GDPR and FERPA, ensuring that user data is handled with care and transparency.
S6: The system shall include privacy settings that allow users to control who can view their reports and personal information.
Incident Response and Reporting:

S7: The system shall have a clear protocol for responding to data breaches or security incidents, including immediate notification to affected users and relevant authorities as required by law.
S8: The platform shall implement logging and monitoring mechanisms to detect and respond to suspicious activities or unauthorized access attempts.
Access Control:

S9: The system shall ensure that only authorized users (students and staff) can access the platform's functionalities. Unauthorized access attempts shall be logged and restricted.
S10: User roles and permissions must be clearly defined, ensuring that users can only perform actions appropriate to their role (e.g., students cannot access admin functionalities).
Mobile Connection Security:

S11: The system shall ensure that all mobile connections to the platform are secured using HTTPS to prevent data exposure over unsecured networks.
S12: The mobile application (if developed) shall implement secure storage practices for sensitive user data, avoiding local storage of unencrypted information.
User Education and Awareness:

S13: The platform shall provide users with guidance on creating strong passwords and recognizing phishing attempts to promote safe usage practices.
By implementing these safety and security requirements, the Lost and Found Platform for College can effectively protect user data, comply with relevant regulations, and provide a secure environment for users to report and recover lost items.

4.3 Software Quality Attributes

*The quality attributes of the **Lost and Found Platform for College** are essential to ensure that the system meets user expectations and performs reliably in various conditions. The following subsections outline specific quality characteristics that will be prioritized in the development process.*

- *4.3.1 Reliability*
- ***Requirement***: *The system shall maintain an uptime of **99.5%** over a one-month period.*

- **Rationale**: *High reliability is crucial for user trust and satisfaction. The platform should be accessible at all times, especially during peak usage periods (e.g., start of the semester).*
- **Implementation**: *The system will employ redundant servers and load balancing to manage user traffic and minimize downtime. Regular maintenance schedules will also be established to ensure optimal performance.*

- **4.3.2 Usability**
- **Requirement**: *The system shall achieve a user satisfaction score of **85% or higher** in usability testing conducted with a representative sample of students and staff.*
- **Rationale**: *Usability directly affects user adoption and engagement. A user-friendly interface will encourage more frequent use of the platform for reporting lost and found items.*
- **Implementation**: *User interface design will focus on intuitive navigation, clear labeling, and accessibility features. User testing will be conducted throughout the development process to gather feedback and make necessary adjustments.*

- **4.3.3 Maintainability**
- **Requirement**: *The system shall allow for updates and modifications to be implemented within a time frame of **24 hours** for critical changes and **48 hours** for non-critical updates.*
- **Rationale**: *Maintainability ensures that the system can adapt to new requirements, bug fixes, and enhancements without significant downtime or resource expenditure.*
- **Implementation**: *The codebase will be organized according to best practices, with thorough documentation provided for all components. Version control systems will be utilized to manage changes effectively.*

- **4.3.4 Adaptability**
- **Requirement**: *The system shall be designed to integrate with potential future systems (e.g., mobile applications or third-party services) within **one month** of development initiation for each new integration.*
- **Rationale**: *As technology evolves, the platform must adapt to new tools and services that may enhance its functionality.*
- **Implementation**: *The system architecture will employ modular design principles, allowing for easy addition or removal of components without affecting overall functionality.*

- **4.3.5 Security**
- **Requirement**: *The system shall implement security measures that ensure **100%** of user passwords are stored using strong encryption algorithms (e.g., bcrypt).*
- **Rationale**: *Security is paramount to protect user data and maintain trust in the platform.*
- **Implementation**: *Regular security audits will be conducted, and the latest best practices for data protection will be integrated into the development process.*

By focusing on these quality attributes, the Lost and Found Platform for College will not only meet functional requirements but also ensure a positive user experience, adaptability to future needs, and a secure environment for all users.

5 Other Requirements

5.1.1 Database Requirements:

- The Lost and Found Platform will store user data, lost item reports, found item reports, and notifications in a relational database (e.g., MySQL, PostgreSQL). The database must be designed to handle concurrent access and be scalable to accommodate a growing number of reports and users.

5.1.2 Internationalization Requirements:

- The system must support multiple languages to cater to a diverse college community. User interface elements, notifications, and help documentation should be available in at least English and Hindi, with the ability to dynamically switch languages based on user preferences.

5.1.3 Legal Requirements:

- The platform must comply with data privacy laws, including but not limited to GDPR for users in the European Union and any relevant data protection regulations in India. All user data should be processed and stored in compliance with these regulations, including obtaining user consent for data collection.

5.1.4 Reuse Objectives:

- The system should be designed with modularity in mind, allowing components such as the reporting engine or notification system to be reused in other applications or projects, facilitating future enhancements and integration with other systems.

Appendix A – Data Dictionary

A data dictionary is used to track all the different variables, states, and functional requirements described in this document. Below is a list of constants, state variables (and their possible states), inputs, and outputs used in the this system. The table includes the description of these items, as well as related operations and requirements.

Field	Description	Data Type	Constraints
User ID	Unique identifier for each user	String	Must be unique
Email	User's college email address	String	Must be valid email format
Password	User's account password	String	Must be encrypted
First Name	User's first name	String	Max length: 50 characters
Last Name	User's last name	String	Max length: 50 characters
Role	Role of the user (e.g., Student, Staff)	String	Must be either 'Student' or 'Staff'
Item ID	Unique identifier for each lost/found item	String	Must be unique
Item Description	Description of the lost or found item	String	Max length: 255 characters
Location	Location where the item was lost or found	String	Max length: 100 characters
Time of Loss/Found	Date and time the item was lost or found	DateTime	Required
Image URL	Link to the uploaded image of the item	String	Must be a valid URL
Match Status	Status of the matching process (e.g., Matched, Not Matched)	String	Default: 'Not Matched'
Notification Sent	Boolean indicating if a notification has been sent to the user about a match	Boolean	Default: false
Timestamp	Date and time of the report submission	DateTime	Automatically generated

Appendix B – Group Log

*The group log documents the activities and meetings held by the development team during the project lifecycle of the **Lost and Found Platform for College**.*

Date	Group Members Present	Discussion Points and Decisions Made	Action Items Assigned
01/09/2024	Bhoomi, Prasuk, Kartik, Hardik	Discussed SRS structure and project scope	Bhoomi: Document Structure; Prasuk: Functional Requirements; Hardik: Non-functional Requirements; Kartik: Use Case Model
10/09/2024	Bhoomi, Prasuk, Kartik, Hardik	Reviewed functional requirements and use cases	Finalized functional requirements
15/09/2024	Bhoomi, Prasuk, Kartik, Hardik	Finalized non-functional requirements and constraints	Bhoomi: Review Document; Prasuk: Prepare for final submission