B.TECH. (CSE)

V Semester

UE21CS341A –Software Engineering

PROJECT REPORT

on

University Lost and Found Management System

Submitted by :

SHREYA JOSHI	**PES2UG21CS501**	**SHRUTI C**	**PES2UG21CS514**
SPOORTHI SHIVAPRASAD	**PES2UG21CS536**	**SRAGVI ANIL SHETTY**	**PES2UG21CS537**

August – Dec 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BENGALURU – 560100, KARNATAKA, INDIA

Table of Contents

1. Project Proposal

1.1 Proposed project description

Our project, Lost and Found Management System has 2 modules: Management and Public sites. The Management site is accessible to authorized personnel, enabling administrators to manage all system data while granting only limited permissions to staff.

It also allows updates to take place in the Public site. The Public site allows visitors to explore published and unclaimed items, browse site content and filter items by category. All these features make our project user-friendly and efficient.

1.1.1 Tools and Software used:

- HTML
- PHP
- MySQL
- CSS
- Javascript
- Jquery
- Bootstrap Frameworks
- XAMPP

1.1.2 Entities

1. Category
2. Item
3. User
4. Messages

1.2 System Description

Functionalities of this Project include the following:

1. Add-Add new category, add new item, add new user (implemented by: Shreya Joshi)
2. View- View categories, view items, view users (implemented by: Shruti C)
3. Update- Update category details (implemented by: Spoorthi Shivaprasad)
4. Delete-Delete category after has been found (implemented by: Sragvi Anil Shetty)

1.**ADD**:

This functionality allows users to add new lost and found items to the system.

The add functionality typically requires users to provide the following information:

Unique identifier for the item

Description of the item

Category the item belongs to

Location where it was found

Contact information of the person who reported the item lost.

New users will also be added to the database.

2.**VIEW**:

View Categories: This functionality allows users to view all the categories of lost and found items that have been reported.

View Item: All the details entered with respect to the lost and found item that have been reported will be displayed here.

View Users: This functionality allows users to view all the users who have reported lost and found items.

3**.UPDATE**:

The update functionality typically allows users to change the following information about a lost and found item:

The item's description

The item's location

The contact information of the person who reported the lost item

4.**DELETE**:

The delete functionality typically requires users to confirm that they want to delete the item which can be done after it has been found by the respective owner. Once the item is deleted, it cannot be restored.

2. Project Planning

2.1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model. (Shreya Joshi - PES2UG21CS501)

Answer:

We plan to follow the Agile methodology for the execution of our project. Agile is a very realistic approach to software development, which promotes teamwork and cross-training.

Our project will span over a period of 2-3 months, involving 4 team members. In agile, the resource requirements are minimum and the functionalities can be developed rapidly and demonstrated. Agile enables concurrent development and delivery within an overall planned context. This aligns with our resource availability and team environment. We have hence chosen the Agile methodology for our project.

Listed below are the reasons why we preferred Agile over the SDLC Legacy models:

1. The Waterfall and V models assume that the requirements are frozen. They are difficult to change and are sequential in nature.

2. The Prototype model requires that the entire system prototype be built to understand the requirements. This may increase the complexity.

3. The Incremental model involves successive release of functionalities which are continually integrated until the entire system is achieved. This seems like an ideal model for our project too. However, the drawback is that it is hard to identify common functionalities across increments. Also, once an increment is released, it cannot be reverted back for any changes.

4. The Iterative model is rigid, involves continual rework that may cause the project to get extended.

2.2: Identify the tools which you want to use throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

(Shreya Joshi - PES2UG21CS501)

Answer:

Planning tool – Microsoft Excel

Design tool – Microsoft Word, Lucid Chart

Version Control – GitHub

Development tool – XAMPP Server, PHP, MySQL

Bug tracking – JIRA

Testing tool – Selenium

2.3: Determine all the deliverables and categorize them as reuse/build components and justify the same. (Sragvi Anil Shetty - PES2UG21CS537)

2.3.1 Reusable Components

1. User Interface Design (Wireframes)- The reusable components in this include design templates, style guides, UI components.

2. Source code and Executables - Though the source code for University Lost and Found Management System is specific to this project, some libraries and frameworks that can be reused in future projects may exist.

3. User Documentation - User manuals can be used as templates for similar documentation for other projects.

4. Training Materials - These are the materials designed to facilitate the learning and understanding of how to use the software (in this case the Lost and Found System). They can be reused for:
    a. Continuous training - As the system will evolve with new updates and features these materials can be updated.
    b. Scaling to other Campuses

5. Legal and Compliance Documentation - Can be adapted for other lost and found projects/ applications.

6. Test Plan - The overall structure and format of a test plan document can serve as a template for creating test plans for other similar projects.
   The Risk assessment framework in the test plan can be reused with adjustments to assess the risks for new projects, especially if the risk categories and criteria are similar.

7. Maintenance and support plan - While the specific content of this plan will vary based on the characteristics of the project, the underlying principles, frameworks and procedures are usually consistent.

2.3.2 Built Components

1. Software Requirements Specification (SRS) - It is specific to the project and outlines the requirements for University level Lost and Found Management system. It cannot be directly used for other projects.

2. System Design Documentation - system architecture and the database design documents are tailored to be specific for this project. While the design principles may be adopted, it's not directly reusable.

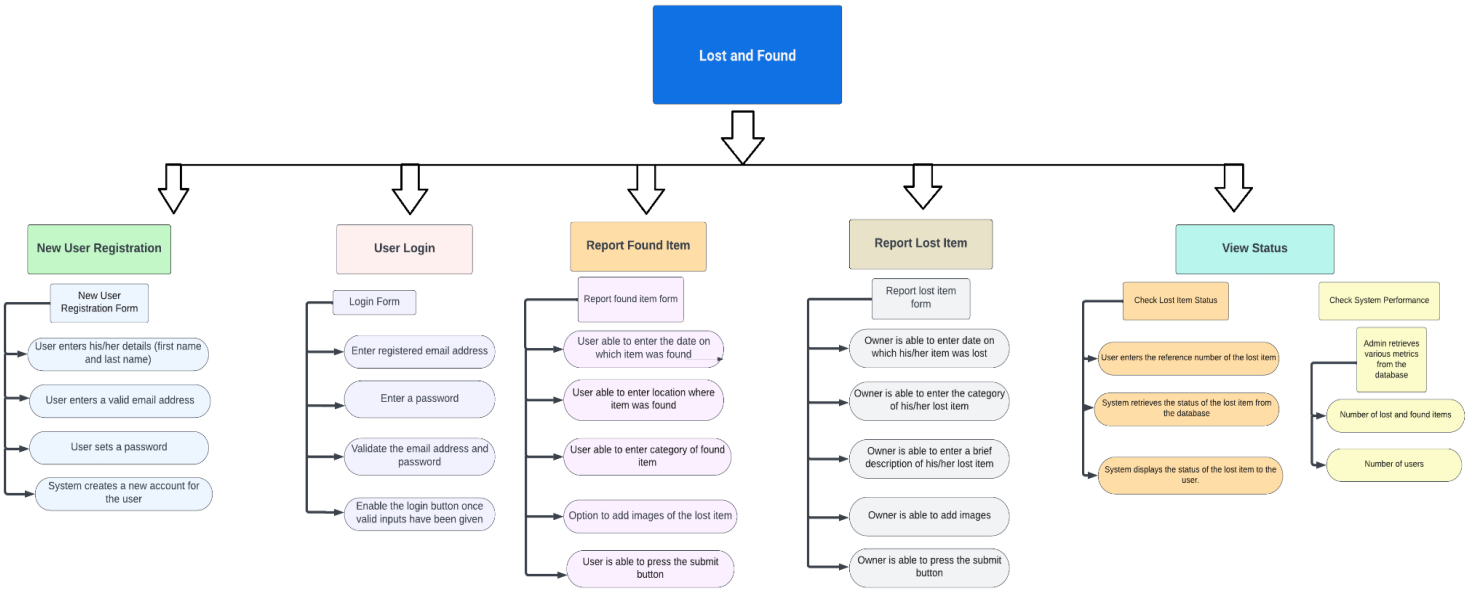
3. Source Code and executables - The primary/core source code and executables are project-specific.

4. Post-implementation review (PIR) - is conducted after the completion of the project and evaluates the overall performance. This outcome is specific to the project.

5. Data-Intensive Components - parts that handle large amounts of data like database schema, data models or data processing. These components are built specifically for the project's requirements. Hence, it is not directly reusable by other projects.

6. Custom Components - These are software modules or functionalities which are developed to address the unique requirements of the user. In our project this could include: user authentication, matching lost and found items, item reporting form etc.

2.4: Create a WBS for the entire functionalities in detail. (Spoorthi Shivaprasad - PES2UG21CS536)



2.5: Do a rough estimate of effort required to accomplish each task in terms of months. (Sragvi Anil Shetty - PES2UG21CS537)

Since we have a small team size, we are using the Organic CoCoMo model for calculating the rough estimate of the effort and time required to carry out the tasks in the WBS.

The project has a maximum duration of 3 months (September to November).

Parameters in Organic CoCoMo: a=24 b=1.05 c=2.5 d=0.38

Assuming the size of each component to be:

- New user registration: 2 KLOC
- User login: 1 KLOC
- Report found items: 2 KLOC
- Report lost items: 1 KLOC
- View status: 2 KLOC

Calculating the effort and time for each of the task:

1. New user registration:
   - Effort (E) = 2.4 * (2^1.05) = 4.36 person-months
   - Time (T) = 2.5 * (Effort^0.38) = 2.12 months

2. User login:
   - Effort (E) = 2.4 * (1^1.05) = 2.4 person-months
   - Time (T) = 2.5 * (Effort^0.38) = 1.17 months

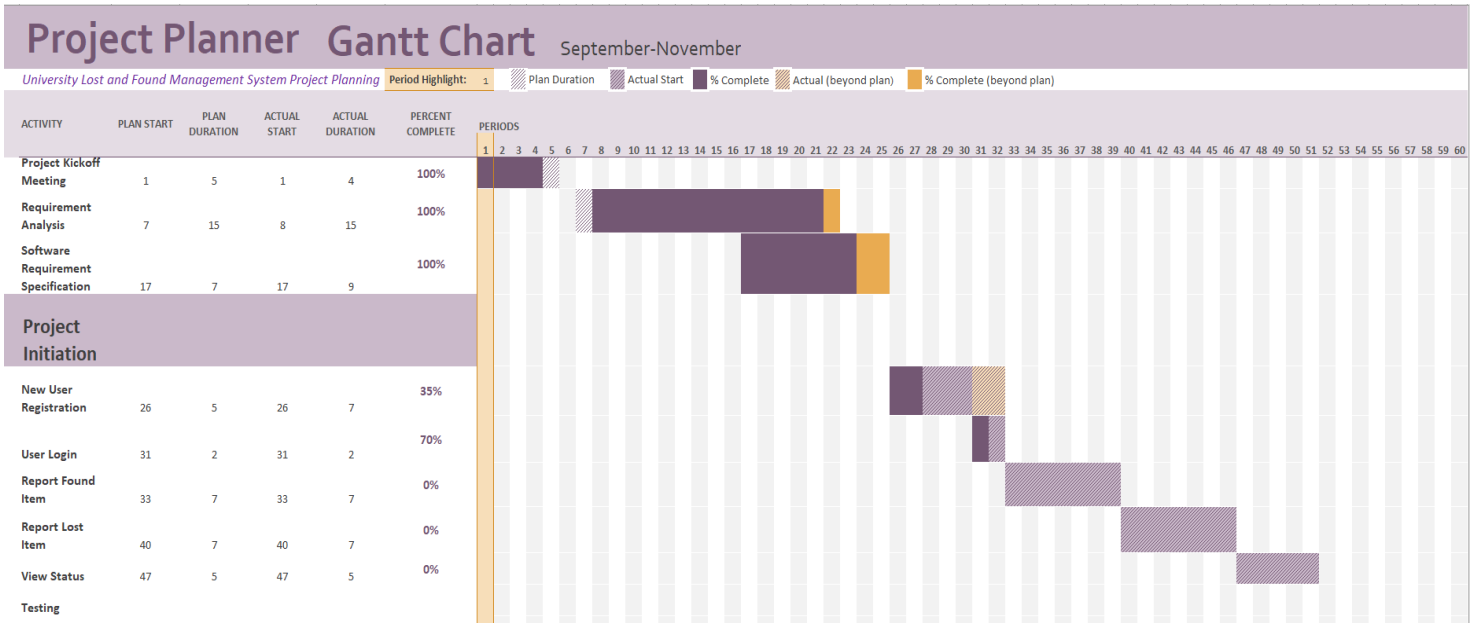
3. Report found items:
   - Effort (E) = 2.4 * (2^1.05) = 4.36 person-months
   - Time (T) = 2.5 * (Effort^0.38) = 2.12 months

4. Report lost items:
   - Effort (E) = 2.4 * (1^1.05) = 2.4 person-months
   - Time (T) = 2.5 * (Effort^0.38) = 1.17 months

5. View status:
   - Effort (E) = 2.4 * (2^1.05) = 4.36 person-months
   - Time (T) = 2.5 * (Effort^0.38) = 2.12 months

These tasks can be run parallelly. Since there are 4 people responsible for the development of the project, each task will be assigned to 1 member. The 5th task will be initiated a little later so the total time taken to complete the project will be around 3 months.

2.6: Create the Gantt Chart for scheduling using any tool. (Shruti C - PES2UG21CS514)



- MS Excel is the tool used to create the Gantt Chart.
- The Project Kickoff Meeting , Requirement Analysis and Software Requirement Specification are the tasks that are 100% completed.
- The remaining software components are in the stage of progress , which will be accomplished in the given duration of time.

The purpose of a Gantt chart in scheduling is to help project managers plan, track, and communicate the project schedule. Gantt charts are visual representations of project schedules, and they can be used to:

- **Identify and manage task dependencies:** Gantt charts show how tasks are related to each other, which can help project managers to identify and manage dependencies. This is important because some tasks cannot start until other tasks are finished.
- **Estimate the project duration:** Gantt charts can be used to estimate the total duration of a project by adding up the duration of all of the tasks. This can help project managers to set realistic deadlines and track progress towards those deadlines.

Communicate the project schedule to stakeholders: Gantt charts are a clear and concise way to communicate the project schedule to stakeholders. Stakeholders can use Gantt charts to see what tasks are being worked on, when those tasks are scheduled to be completed, and how the project is progressing overall.

3. Software Requirements Specification (SRS)

3.1. Introduction

3.1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive overview of the University-Level Lost and Found Management System. This document outlines the objectives, features, functionalities, and constraints of the system. It serves as a guide for the development team, stakeholders, and any parties involved in the project.

3.1.2 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- Development Team: Developers, designers, and testers who will be involved in the design, development, and testing of the University-Level Lost and Found Management System.

- Stakeholders: University administrators, department heads, and other stakeholders with an interest in the successful implementation and operation of the system.

- Quality Assurance Teams: Those responsible for ensuring that the system conforms to the specified requirements and quality standards.

- Users: Students, faculty, staff, and other members of the university community who will interact with the system.

- Administrators: University personnel responsible for managing the system, handling reported items, and resolving matches.

3.1.2.1 Reading Suggestions:

- Developers should focus on the technical details and functional specifications of the system.

- Stakeholders should review the document to gain a clear understanding of the system's scope, objectives, and features.

- Users should refer to sections that describe user roles, functionalities, and guidelines for interacting with the system.

- Administrators should pay particular attention to sections that pertain to system administration, item management, and resolution processes.

3.1.3 Product Scope

The University-Level Lost and Found Management System is a specialized web-based application designed to simplify and enhance the process of reporting lost items, finding lost items, and facilitating communication between the university community members. The system allows users, including students, faculty, and staff, to report lost or found items, search for items, and receive notifications about potential matches within the university campus.

3.1.4 References

There are no specific external references for this university-level lost and found management system. However, the following internal references and documents may be relevant during the development and operation of the system:

- University policies and regulations related to lost and found items.

- Campus security guidelines and procedures.

- Any internal standards or guidelines for data privacy and information security within the university.

These internal references provide essential context and guidelines for developing and operating the University-Level Lost and Found Management System within the university's specific environment and policies.

3.2. Overall Description

3.2.1 Product Perspective

The Lost and Found Management System is a standalone software application designed to streamline the process of reporting and retrieving lost and found items. It operates independently and does not rely on external systems. However, it may integrate with external services for user authentication (e.g., OAuth with social media platforms) and email notifications.

3.2.2 Product Functions

3.2.2.1 The primary functions of the Lost and Found Management System include:

- User registration and authentication.
- Reporting lost items with detailed information.
- Reporting found items with descriptions and locations.
- Searching for lost and found items based on various criteria.
- Matching lost and found items and facilitating contact between owners and finders.
- Admin panel for user and item management.

3.2.3 User Classes and Characteristics

3.2.3.1 The system supports the following user classes:

1. Registered Users: Individuals who have created an account and can report, search for, and manage lost and found items.

2. Guest Users: Individuals who can perform basic searches for lost and found items without registering.

3. Admins: Administrators responsible for overseeing the system, moderating user accounts, and managing reported items.

3.2.4 User Characteristics:

- Registered users may vary in age, technical expertise, and familiarity with the system.
- Guest users are typically looking to search for specific lost or found items.
- Admins have privileged access to the system's admin panel for moderation and resolution of items.

3.2.5 Operating Environment

The Lost and Found Management System is designed to operate in the following environments:

- **Web-based:** The system is accessible via standard web browsers on desktop and mobile devices.

- **Cross-platform:** It is compatible with various operating systems, including Windows, macOS, and mobile OS (iOS, Android).

3.2.6 Design and Implementation Constraints

The system design and implementation are subject to the following constraints:

-**Technological Stack:** The system will be developed using specific programming languages, frameworks, and libraries, as determined during the development phase.

- **Database:** The choice of a relational database system will influence data storage and retrieval.

- **Data Security:** Compliance with data protection laws and standards is mandatory, leading to specific security constraints.

- **Scalability:** The system should be designed with scalability in mind to accommodate future growth in user and item databases.

3.2.7 Assumptions and Dependencies

The development and operation of the Lost and Found Management System are based on several assumptions and dependencies:

- **User Authentication:** The system assumes that users will provide accurate and verifiable information during registration.

- **Internet Connectivity:** Users and admins are assumed to have a stable internet connection to access the system.

- **External Services:** If integrated, the system depends on external services (e.g., email servers, third-party authentication providers) to function correctly.

- **Legal Compliance:** The system assumes that users and administrators will adhere to legal requirements related to lost and found items and data privacy.

- **Maintenance:** Ongoing system maintenance, bug fixes, and updates are assumed to be carried out regularly to ensure its optimal operation.

These assumptions and dependencies are critical for the successful deployment and use of the Lost and Found Management System.

3.3. External Interface Requirements:

We are planning to develop this project using technologies that allow you to create a dynamic and interactive web application.

- HTML and CSS are used to create the structure and style of the web pages.
- PHP and MySQL are used to create and manage the database, and to generate the dynamic content of the web pages.
- JavaScript and jQuery are used to add interactivity to the web pages, such as allowing users to click on buttons and submit forms.
- Ajax requests are used to communicate with the server without reloading the web page, which makes the web application more responsive.
- The Bootstrap framework and icons are used to create a consistent and modern look and feel for the web application.
- The NiceAdmin template is used to provide a starting point for the design of the web application.

3.3.1. User Interface:

In this project we will develop a website. This website will have the following features which will make it user-friendly.
The homepage of our website will have clear and concise labels for the links to the lost and found page, about the website page, contact us page, and post and item page. The labels will be easy to understand and will accurately reflect the content of the linked pages, we will also provide a button labeled login which on clicking will lead to the login page which different **users** can use to login through the website.All of these labels will be present at the header of the website.

The post an item page on your website should have a form that allows users to enter the details of the lost object and upload images. The form will consist of the following sections:

Category: This section should allow users to mention the category of the lost object.
Founder name: This section should allow users to enter their name.

- **Title:** This section should allow users to enter a title for the lost item. The title should be a brief description of the item, such as "Lost iPhone" or "Lost black wallet".
- **Found contact:** This section should allow users to enter their contact information, such as their phone number.
- **Description:** This section should allow users to enter a detailed description of the lost item.

We will include appropriate validation to check the contents filled in the form.

Client-side validation: Client-side validation is performed by the web browser before the form is submitted to the server. This type of validation can be used to check for basic things like required fields and valid email addresses.

Server-side validation: Server-side validation is performed by the server after the form is submitted.We use a scripting language like PHP or Python to check the form data after it is submitted. For example, you could use PHP to check that the username exists in the database or not.

The lost and found items page will be used to display the contents filled in the form.This page will be used by the user to view the lost and found items.

3.3.2. Software Interface:

The website would need to be connected to a database to store the data for the lost and found items. Here we will make use of the MySql database.

Application programming interface protocols

The website would use the following application programming interface (API) protocols:

- **MySQL API:** The website would use the MySQL API to communicate with the MySQL database.
- **PHP Mailer API:** The website would use the PHP Mailer API to send email notifications.

The website would need to be connected to the following software components:

- **XAMPP:** XAMPP is a software stack that includes a web server, a database server, and a programming language interpreter. The website would use the Apache web server to serve its files to users, the MySQL database server to store its data, and the PHP programming language interpreter to process its code.
- **HTML:** HTML is a markup language that is used to create the structure and content of the website's pages.

- **PHP:** PHP is a programming language that is used to develop the website's functionality.
- **MySQL Database:** MySQL Database is a relational database management system (RDBMS) that is used to store the website's data.
- **jQuery:** jQuery is a JavaScript library that simplifies the process of adding interactivity to web pages.
- **Ajax Requests:** Ajax requests are used to communicate with the server without reloading the web page. This allows the website to be more responsive and user-friendly.

Data items or messages coming into the system:

- User registration information (name, email address, password)
- Lost item information (category, title, description, images)
- Found item information (category, title, description, contact information)

3.3.3. Communication Interface:

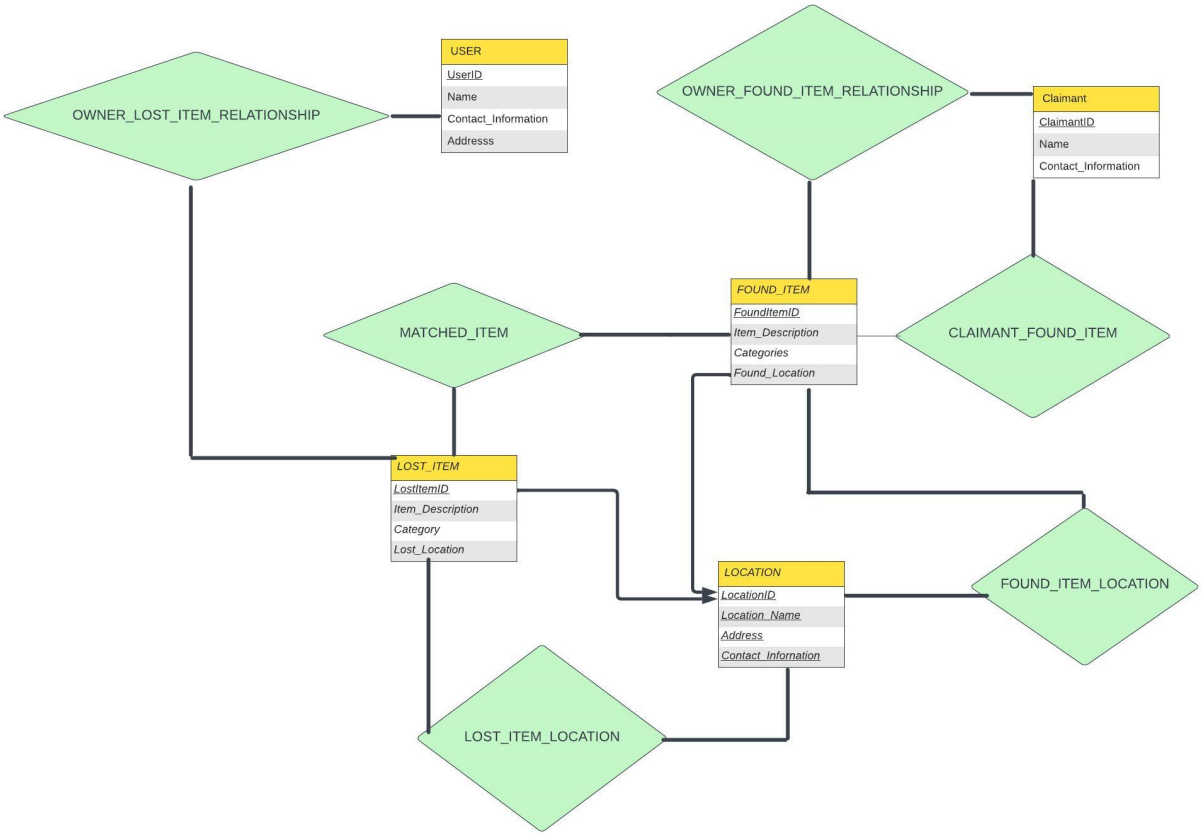
The contact information page on your website can be used to contact the admin user in a number of ways:

- **Email:** The user can send an email to the email address provided on the contact page.
- **Telephone:** The user can call the telephone number provided on the contact page.
- **Mobile phone:** The user can call the mobile phone number provided on the contact page**.**
- **Walk-in:** The user can visit the office address provided on the contact page.
- **Contact form:** The user can fill out the contact form on the contact page and submit it.

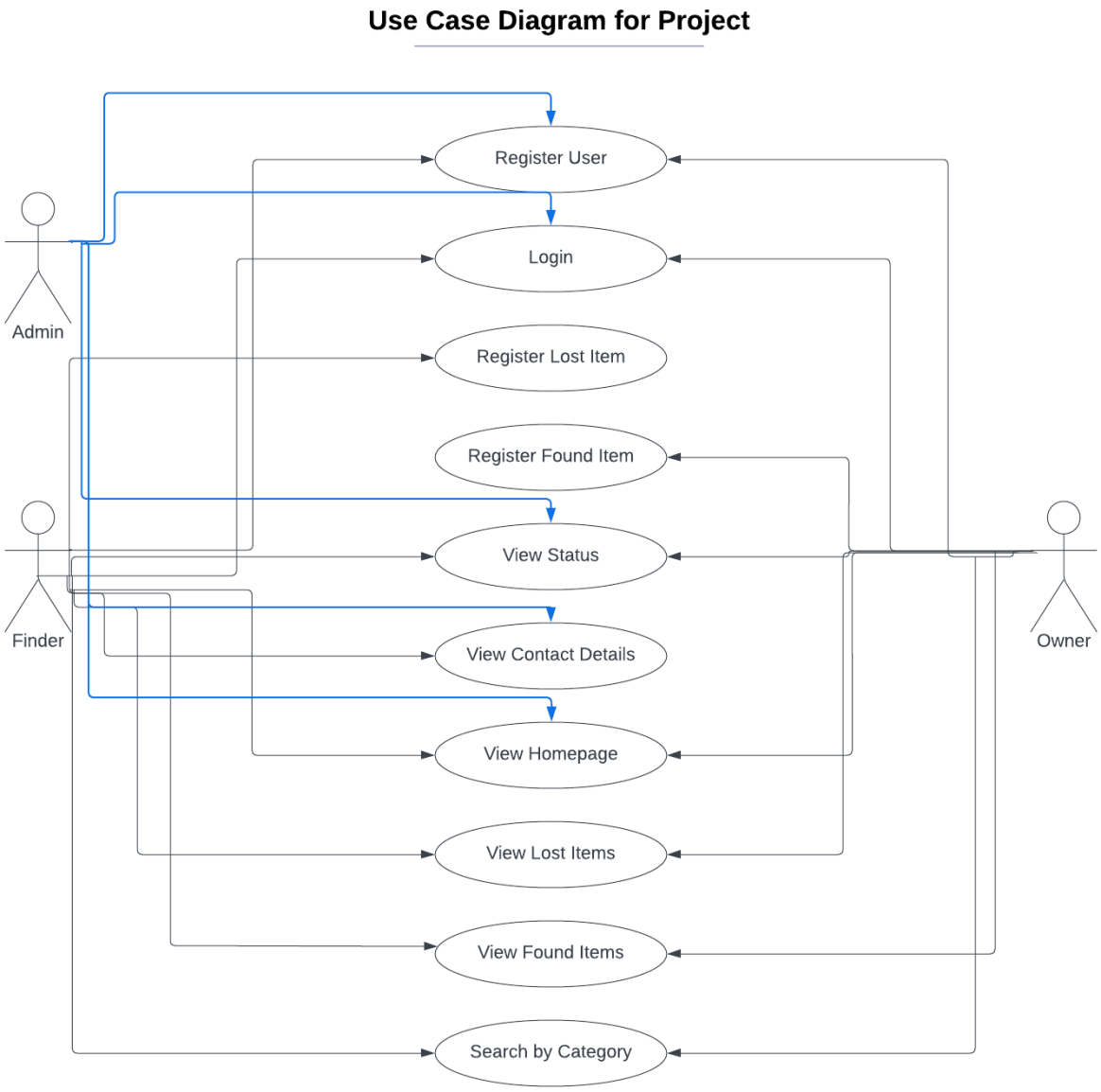
The contact form is a convenient way for users to contact the admin user. The form should collect the user's name, email address, contact number, and message. The admin user can then respond to the user's message using the contact information provided in the form.

3.4. Analysis Model

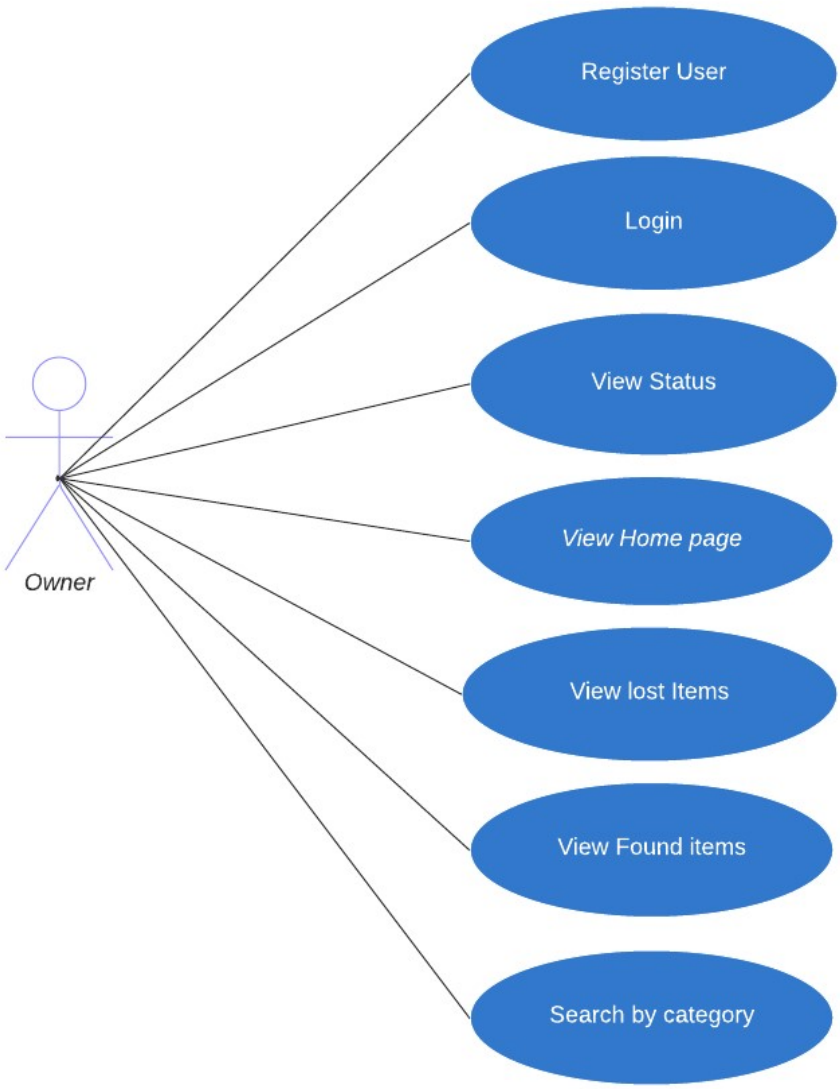
3.4.1 ER DIAGRAM



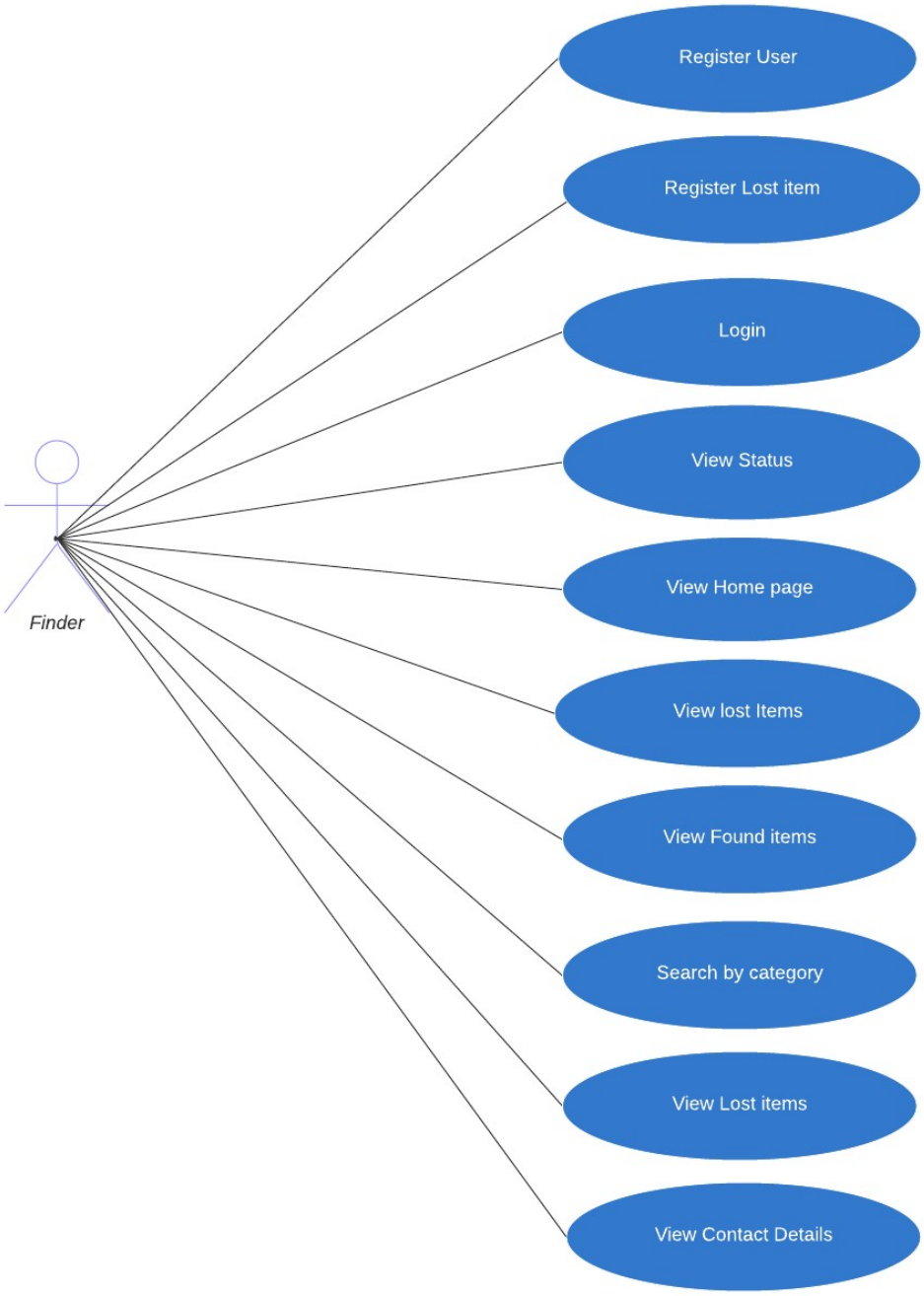
3.4.2 Use Case Diagram for Project:



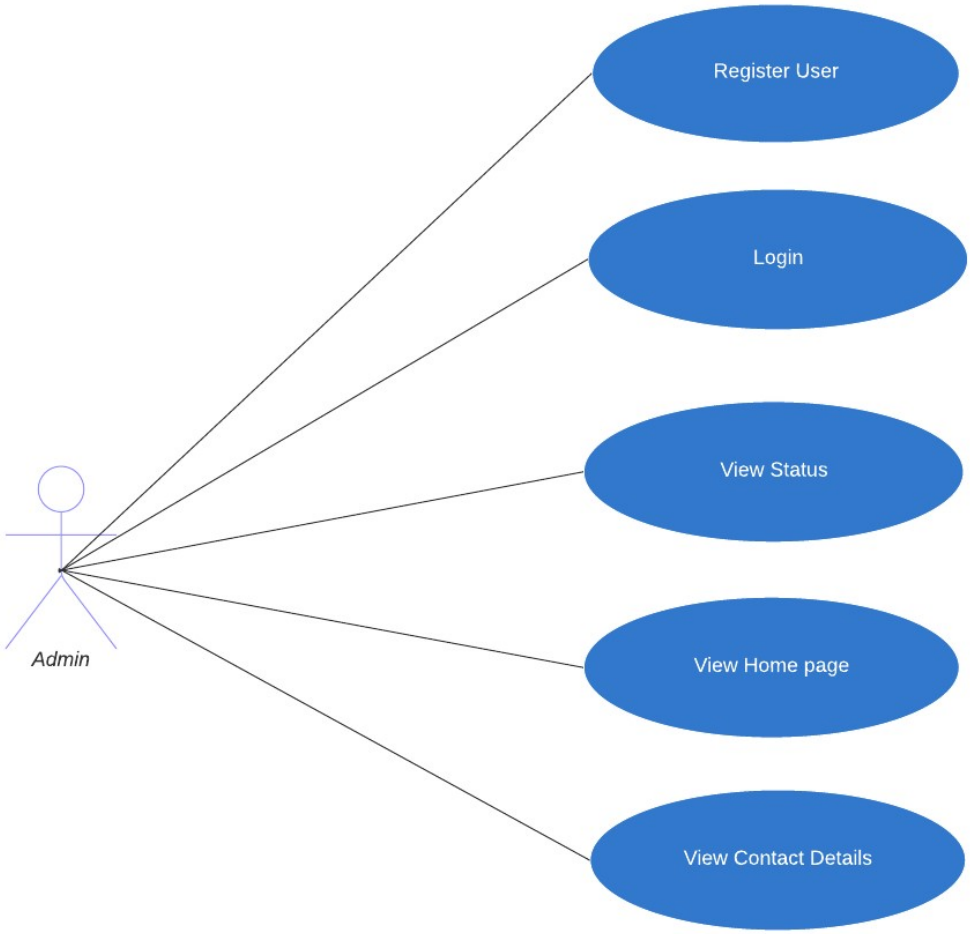
3.4.3 Use case diagram for Owner:



3.4.4 Use case diagram for Finder:



3.4.5 Use case diagram for Admin:



3.5. System Features

3.5.1 System Feature-1: User login

3.5.1.1 Description:

User login allows registered users to login to their accounts securely. This feature is of high priority since it ensures user access control and data privacy.

3.5.1.2 Stimulus/Response sequences:

-User enters his/her registered email and password

-System validates the entered credentials

-If correct, system logs in to the user's account

3.5.1.3 Functional Requirements:

-REQ-1: The owner/finder can login with the system if he/she is registered with the system

-REQ-2: The owner/finder can enter a registered username.

-REQ-3: The owner/finder can enter a password

-REQ-4: If owner/finder has given valid username and password, they are able to press the login button

-REQ5: User can login into the system

3.5.2 System Feature-2: Report found item

3.5.2.1 Description:

Users can report a found item on the system. This is of high priority found items need to be returned and their status updated accordingly.

3.5.2.2 Stimulus/Response sequences:

-User enters details of a found item

-System updates the status of the lost item to 'found.'

3.5.2.3 Functional requirements:

-REQ-1: The user can report his/her lost item
-REQ-2: The user can enter the date on which item was found
-REQ-3: The user can enter location where the item was found
-REQ-4: The user can enter category of found item
-REQ-5: After entering the details, user can press the submit button

3.5.3 System Feature-3: Report lost item

3.5.3.1 Description:

Users are able to report a lost item on the system.

3.5.3.2 Stimulus/Response sequences:

-User enters the details of the lost item
-User clicks on the ‘submit’ button
-System records the lost item into the database

3.5.3.3 Functional requirements:

-REQ-1: The owner can report his/her lost item
-REQ-2: The owner can enter date on which his/her item was lost
-REQ-3: The owner can enter the category of his/her lost item
-REQ-4: They can enter a brief description of his/her lost item
-REQ-5: After entering the details, owner can press the submit button

3.5.4 System Feature-4: New user registration

3.5.4.1 Description:

New user registration allows an owner/finder to register himself/herself on the lost and found management system.

3.5.4.2 Stimulus/Response sequences:

-User enters his/her details first name and last name
-User enters a valid email address
-User sets a password
-System creates a new account for the user

3.5.4.3 Functional Requirements:

-REQ-1: The owner or finder can register himself/herself onto the system
-REQ-2: Users are able to enter their first and last name
-REQ-3: The user is able to enter a valid email address
-REQ-4: The user is able to set a password

3.5.5 System feature-5: View status

3.5.5.1 Functional requirements:

-REQ-1: The owner/finder can check the status of the lost item by entering the reference number of that item

-REQ-2: The admin can check the performance of the system

-REQ-3: Admin can block the user, who is trolling, posting abusive content, etc

3.5.6 System feature-6: View homepage

3.5.6.1 Functional requirements:

-REQ-1: The lost items highlights are displayed on the main page
-REQ-2: The user can search the lost item from the search page
-REQ-3: The user can report a lost item
-REQ-4: The user can report a found item

3.6. Other Non-functional Requirements

3.6.1 Performance Requirements:

The system should be able to handle up to 100 lost and found item reports daily and with a response time of under 5 seconds per request.

3.6.2 Safety Requirements:

The system should not disclose any sensitive information and should comply with all applicable safety regulations within the university.

3.6.3 Security Requirements:

- The system should implement authentication.
- All sensitive data in the system's database should be encrypted using strong encryption algorithms.
- Secure data transmission provided by using HTTPS for communication between client and server.
- Security patches and updates should be added regularly to the system.

3.6.4 Software Quality Attributes:

The system should emphasize user-friendliness, comprehensibility and ease of maintenance.

3.6.5 Compatibility Requirements:

- The system should be compatible with the latest versions of web browsers.
- It should also be compatible with mobile devices (such as smartphones and tablets)

3.6.6 Logging and Monitoring Requirements:

The system should maintain logs of user interactions, events and errors.

3.6.7 Business Rules:

- Found items can be categorized only by the admins.
- Only members of the university (students and faculty) can report lost items.
- When reporting lost items, reporters must provide a detailed description of the item.
- User consent - users using the Lost and Found system for reporting and claiming lost items should abide by the system's policy.

4. Design Diagrams

1. Existing Project: Use your ongoing or proposed software project as a foundation for this assignment.

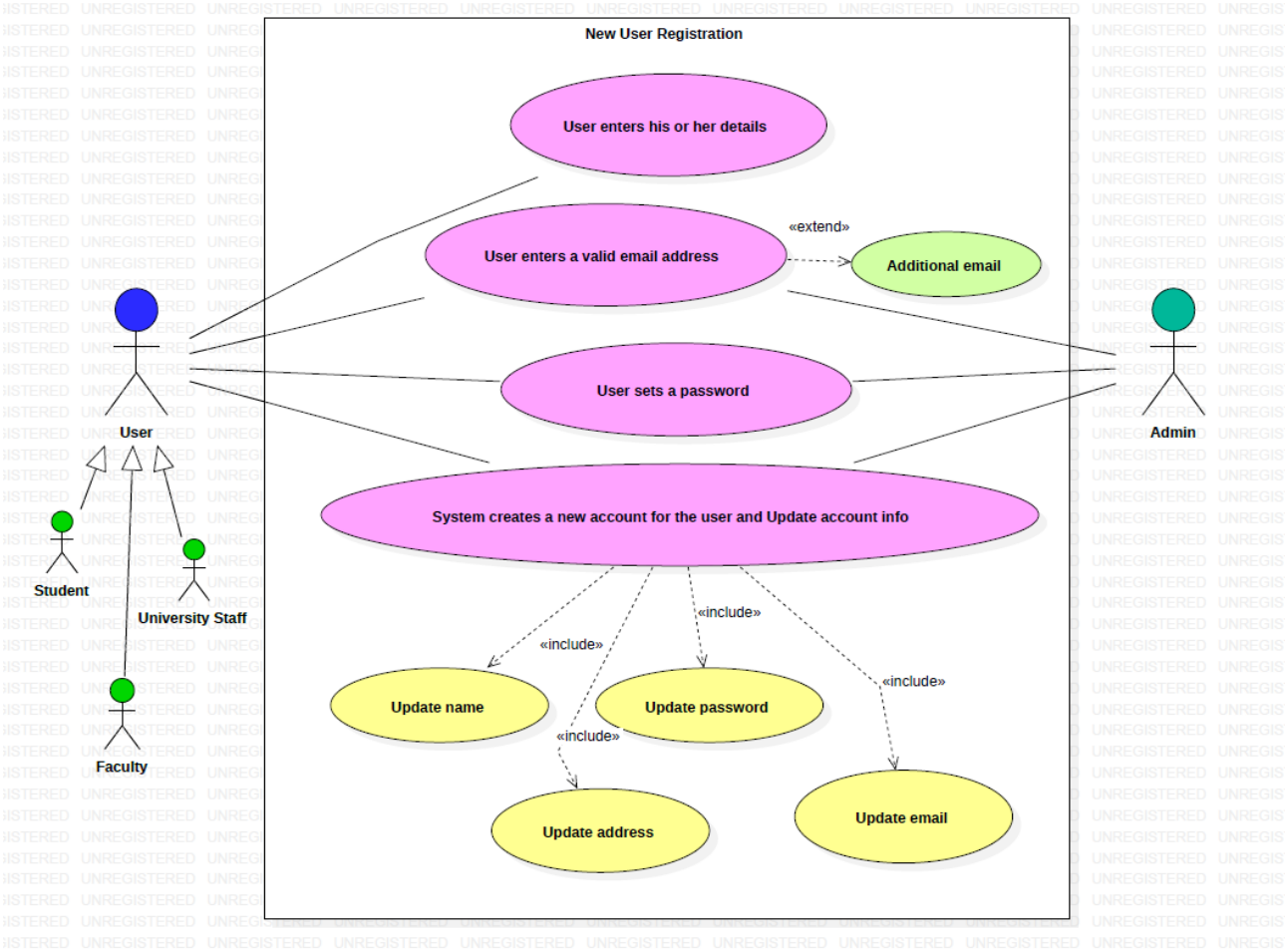
2. Create a Use case and Class Diagram

3. Incorporate DFDs:

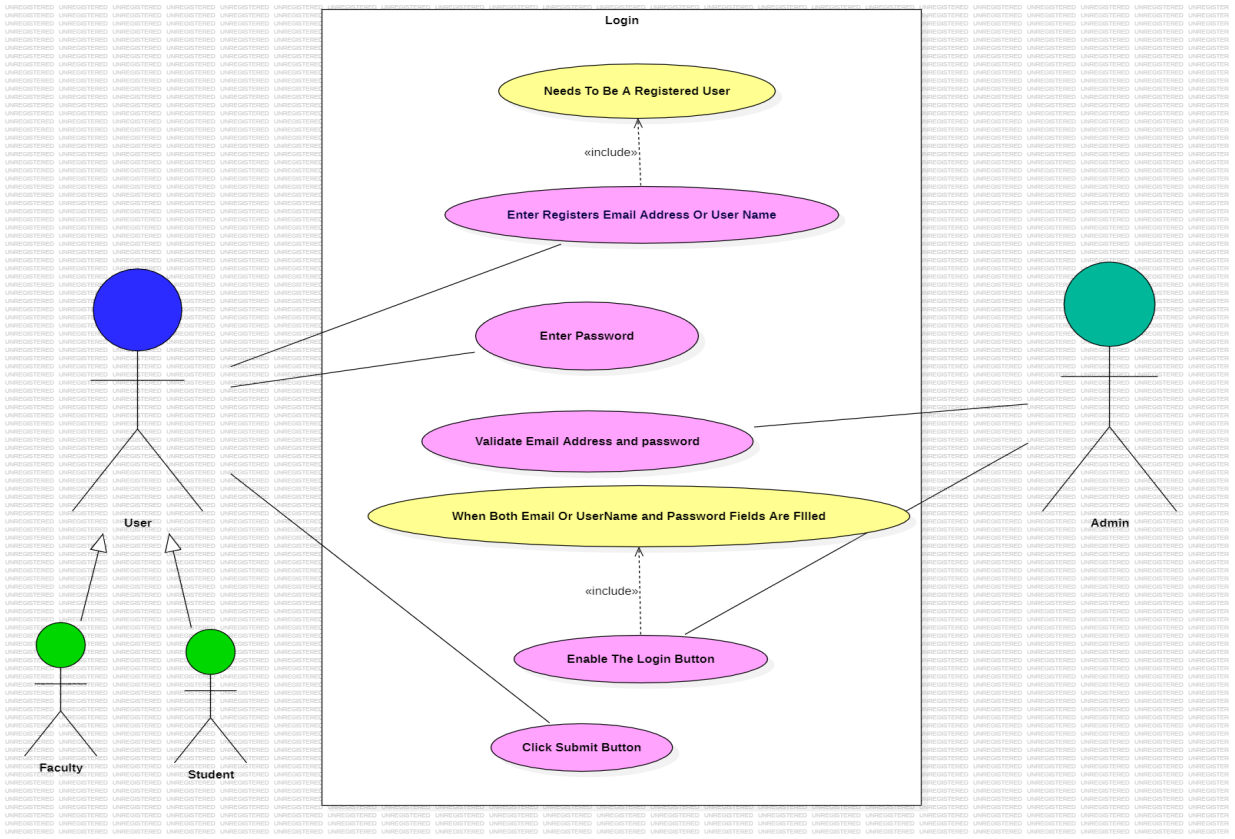
4. Architectural Style Integration:
· Choose and design an architectural style for your project (e.g., Layered, Client-Server, Microservices, Peer-to-Peer, Service-Oriented).

4.1 Use case diagrams and Class diagrams

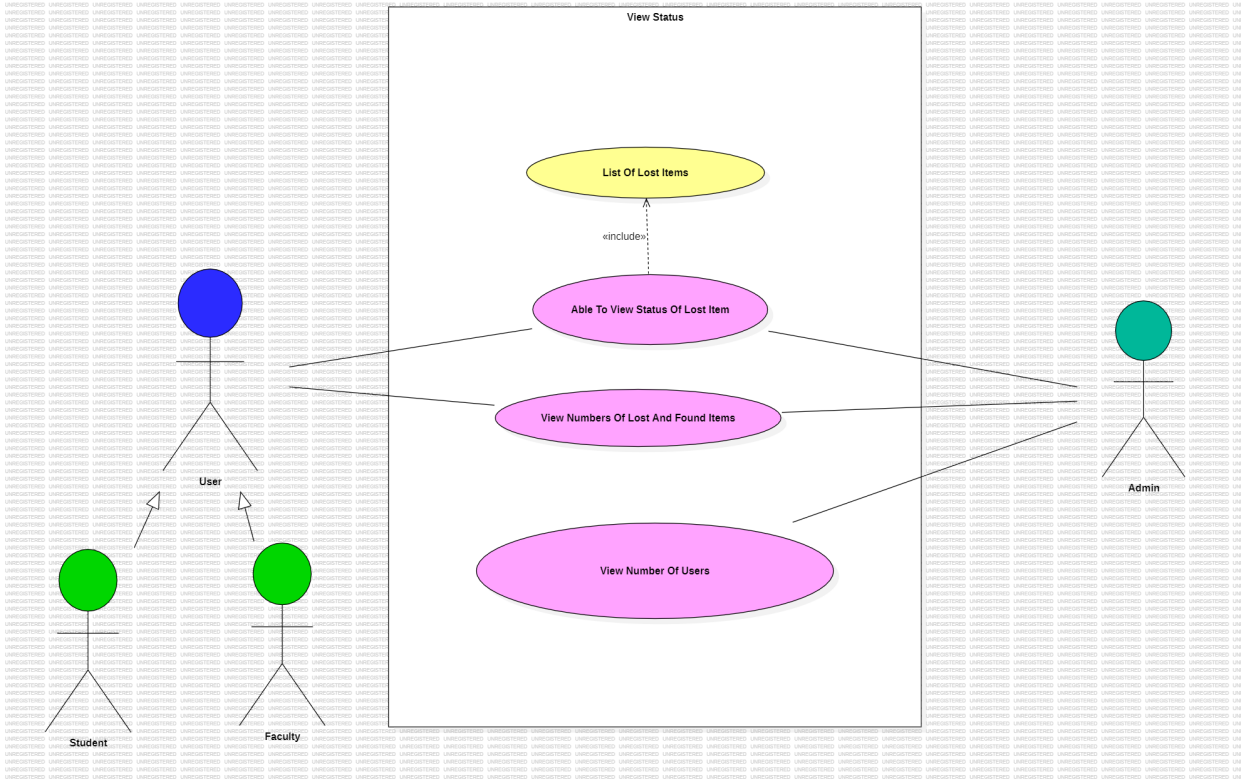
4.1.1 New User Registration (SRN:PES2UG21CS514):



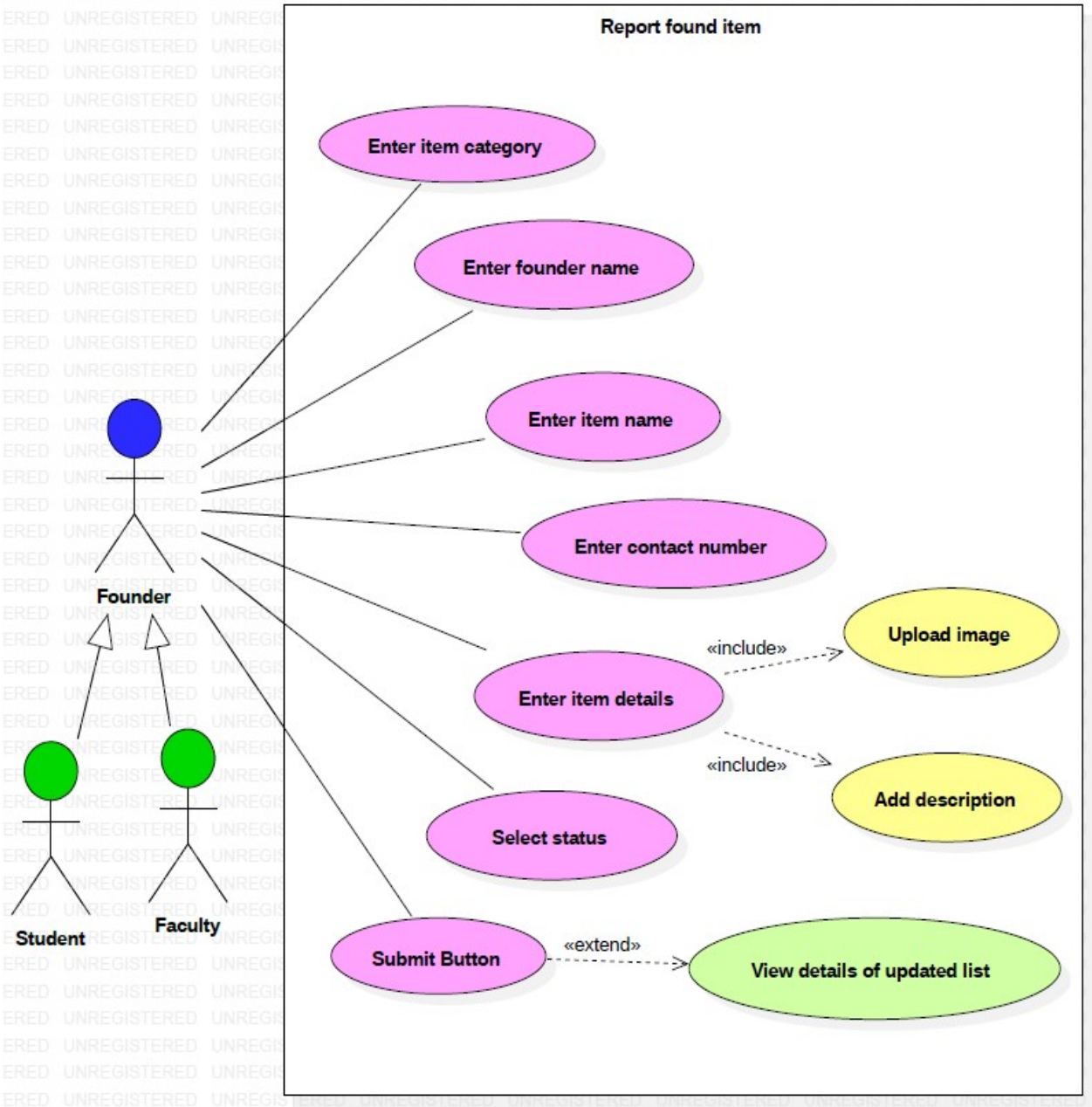
4.1.2 Login (SRN: PES2UG21CS536):



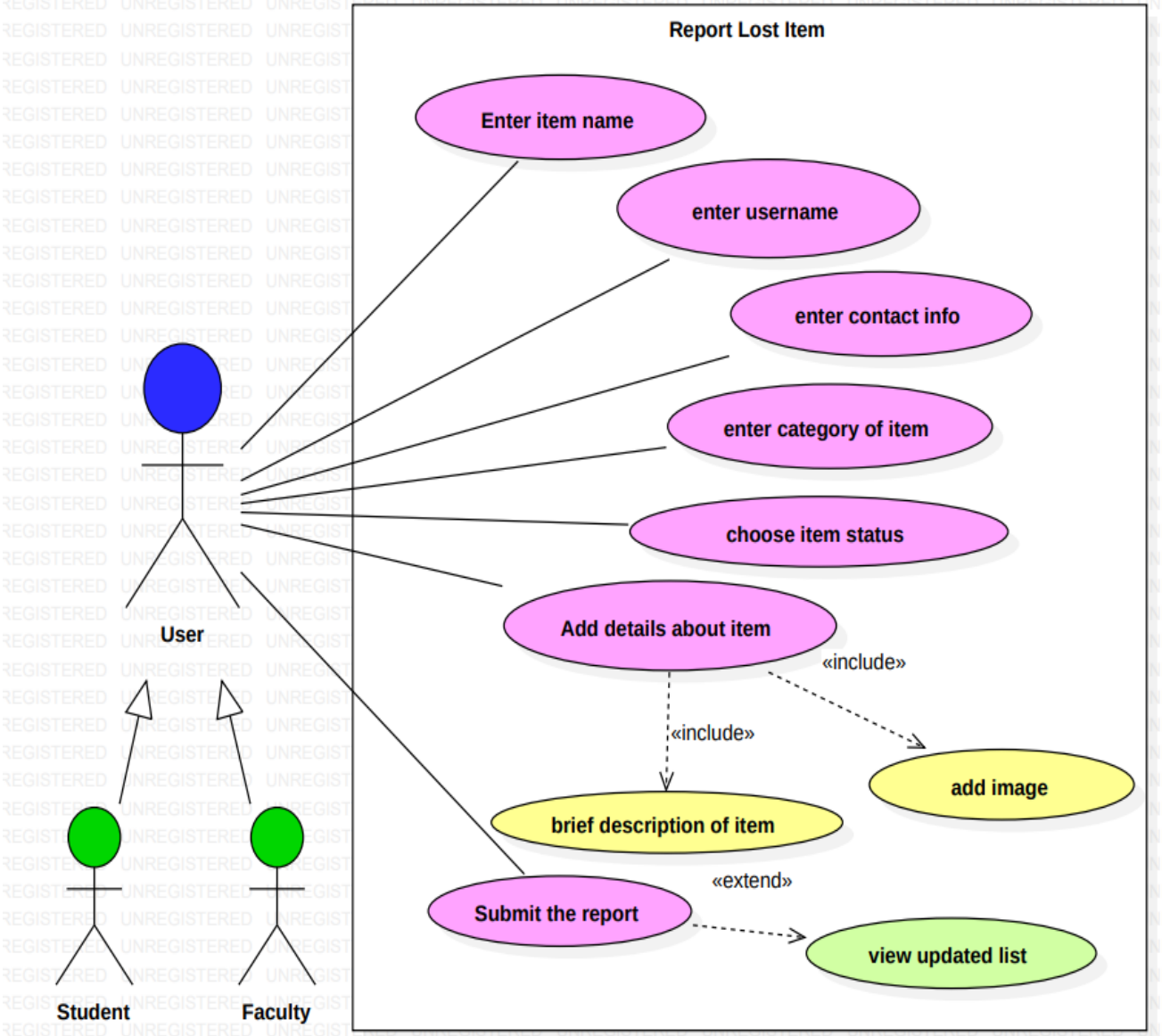
4.1.3 View Status:



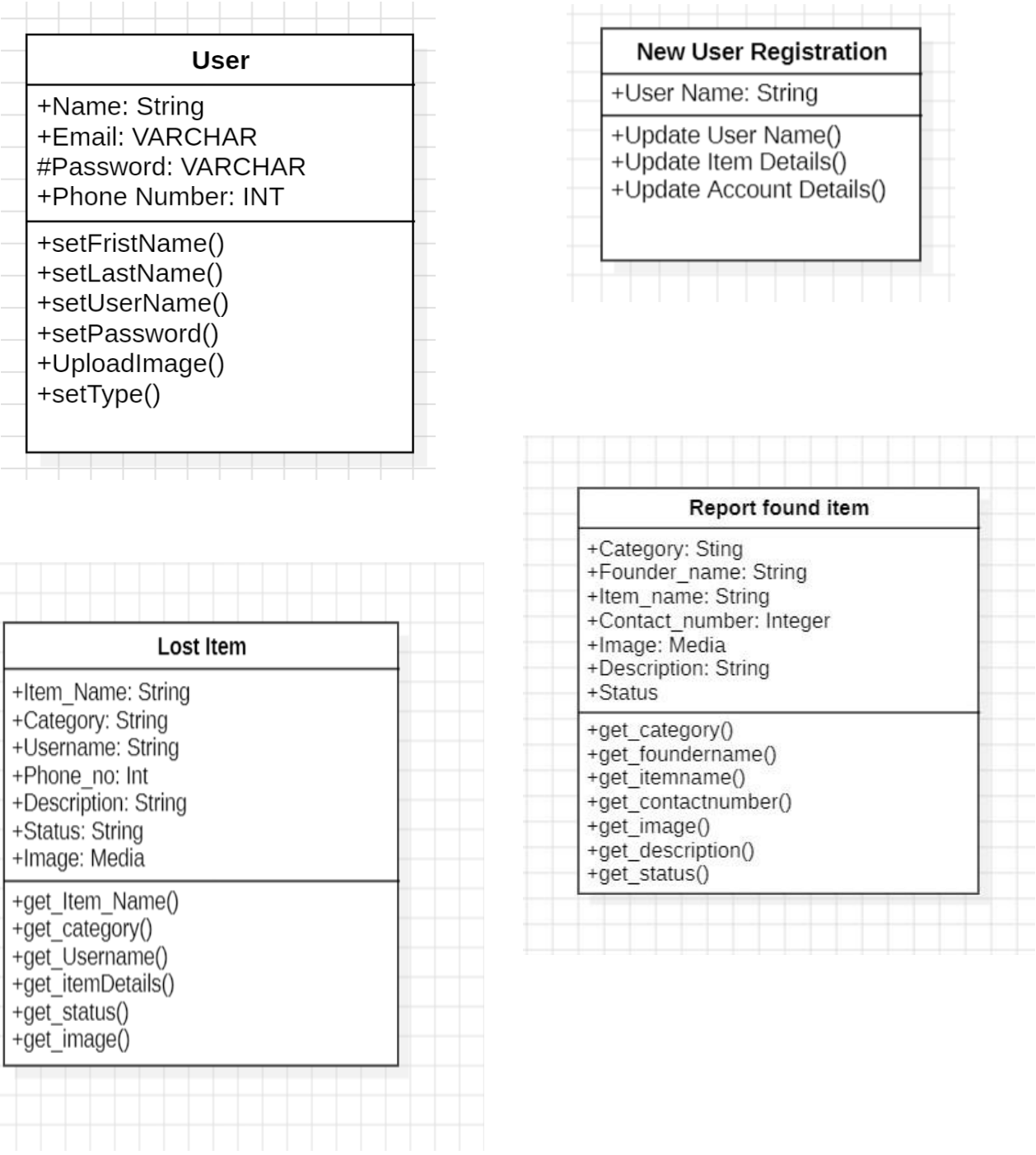
4.1.4 Report found item (SRN: PES2UG21CS501):

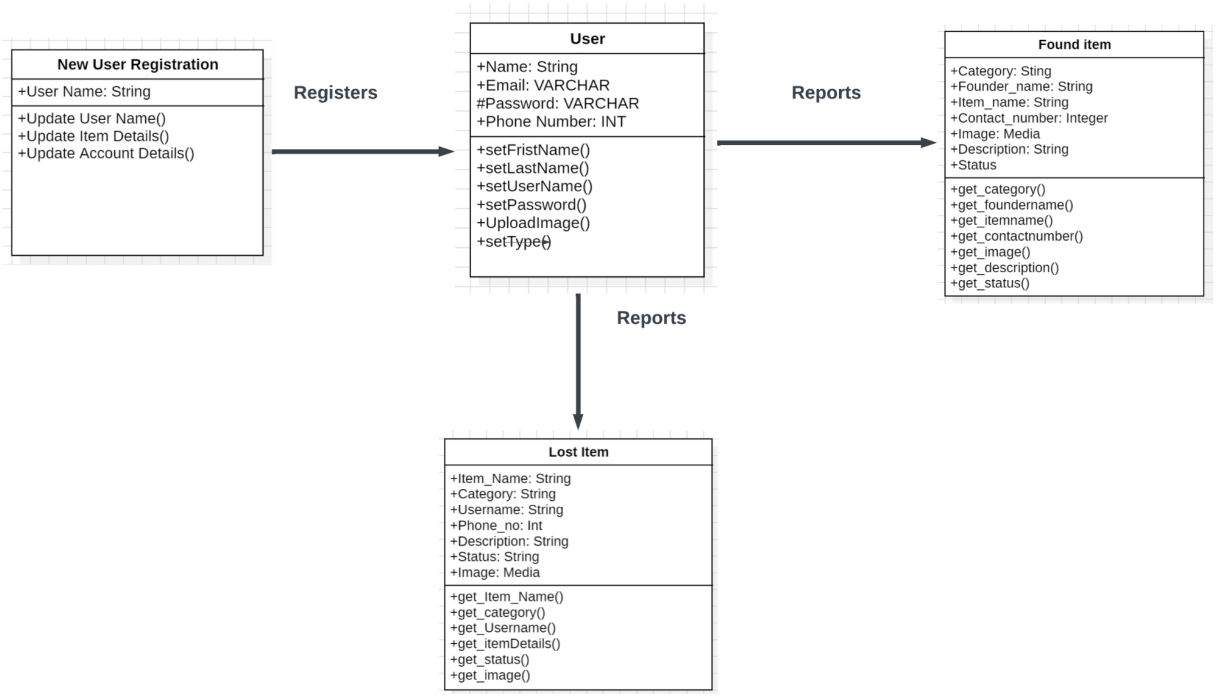


4.1.5 Report Lost Item (SRN: PES2UG21CS537):



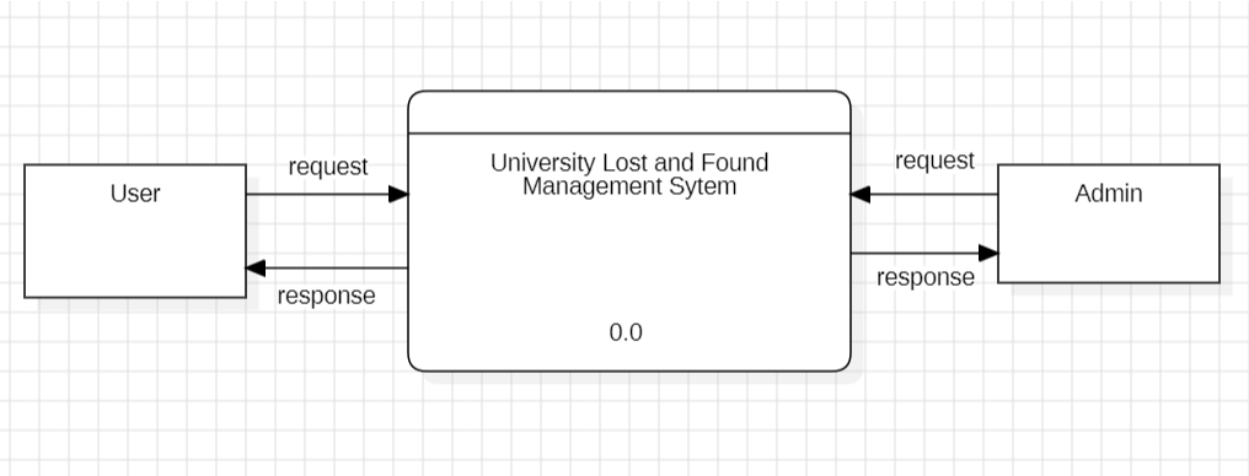
4.2 Class Diagram:



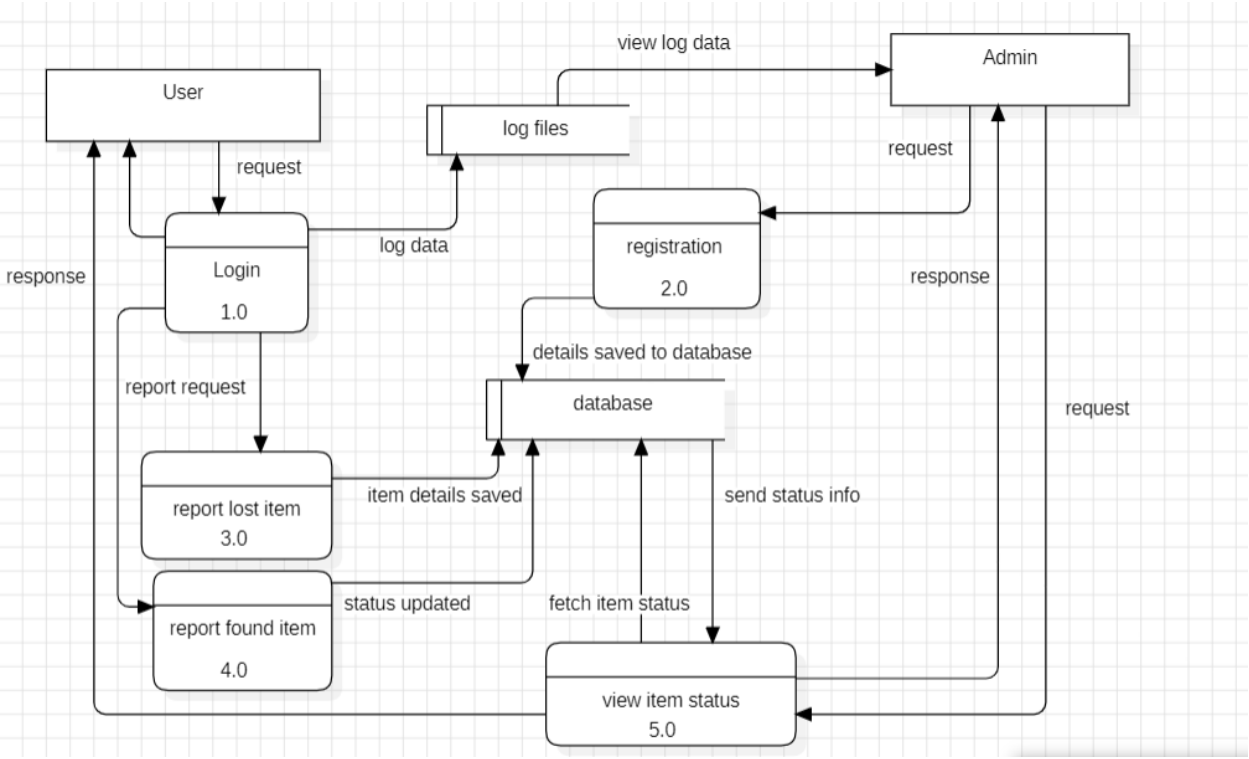


4.3 DFD

4.3.1 DFD - Level 0



4.3.2 DFD - Level 1



4.4. Architecture Design:

We will make use of three-tier architecture design.
The presentation layer, which is responsible for the user interface
The business logic layer, which is responsible for the core functionality of the application.
The data access layer, which is responsible for interacting with the database.

- The presentation layer would include the HTML and CSS code that is used to render the website's pages. We have made use of bootstrap for this purpose.
- The business logic layer would be responsible for the core functionality of the website which would include the code that is used to process user input, validate data, and send queries to MySQL the database we have used.We will make use of PHP here.
- The data access layer would include MySql queries that are used to read, write, and update data in the database.

We have chosen the **three-tier architectural style** for our project because it provides a clear separation between the presentation layer, the business logic layer, and the data access layer. This separation makes the code more modular and **easier to maintain.**

Additionally, the three-tier architecture makes it **easier to scale** the system as **each layer can be scaled independently**. For example, if the system becomes more popular, the presentation layer can be scaled by adding more web servers. The data access layer can also be scaled by adding more database servers.

In terms of **performance**, the three-tier architecture can help to improve performance by **reducing the amount of coupling between the layers**. This is because each layer is responsible for a specific task.The separation of tasks can help to improve performance because it reduces the amount of communication that is required between the layers.

5. Testing Document

5.1 Testing tools Critique

5.1.1 Selenium

Selenium is a free, open-source testing tool that is compatible with a wide range of platforms and supports multiple programming languages. It is suitable for a variety of testing methods, including functional and regression testing, and can be easily integrated with various testing frameworks.

5.1.2 Advantages of Selenium:

- Free and open source
- Platform-agnostic
- Supports multiple programming languages.
- Suitable for a range of testing methods
- Easily integrates with testing frameworks.

5.1.3 Disadvantages of Selenium:

- Limited applicability for non-web applications
- Steep learning curve
- Requires frequent script updates.

5.1.4 Applications of Selenium:

- Automated web application testing.
- Ensures consistent performance across browsers and platforms through cross-browser testing.

Facilitates compatibility testing across operating systems and devices

Selenium is a free and powerful testing tool that can be used to test web applications on a variety of platforms. It is easy to learn and use, but it can be challenging to master. Selenium is not suitable for testing non-web applications, and it requires frequent script updates. However,

its open-source nature, wide platform compatibility, and integration capabilities make it a popular choice for web application testing.

Overall, Selenium is a good choice for software testing teams that are looking for a free, open-source, and versatile testing tool.

5.2 Junit

5.2.1 Introduction:

JUnit, a widely used testing framework for Java, plays a crucial role in ensuring the reliability and stability of Java applications. This report critically analyzes the strengths and weaknesses of JUnit, along with its practical applications in the context of software development and testing.

5.2.2 Advantages of JUnit:

Streamlines the testing process, making it easier for developers to identify and rectify potential issues in the code.

Offers comprehensive support for annotations, assertions, and test runners, facilitating organized and efficient test suite creation.

Integrates seamlessly with popular build tools like Maven and Ant, simplifying the automation of testing procedures within the development workflow.

Enables the seamless incorporation of testing into the continuous integration and deployment process, ensuring the consistent quality of the software application.

5.2.3 Disadvantages of JUnit:

Limited functionality in handling complex tests, potentially posing challenges when dealing with intricate integration or system-level testing scenarios.

Reliance on external libraries, leading to possible compatibility issues and increased dependencies within the project.

Inability to inherently support asynchronous testing, making it cumbersome to test asynchronous code without additional custom implementations or workarounds.

5.2.4 Where to Use JUnit:

Ideal for conducting unit testing, ensuring the individual components of the code function as intended.

Widely employed in Test-Driven Development (TDD) practices, where tests are written before the code, fostering an iterative and systematic development process.

Integrated into continuous integration and deployment pipelines to automate testing and guarantee the stability and reliability of the software application before deployment.

5.2.5 Conclusion:

In conclusion, JUnit serves as a valuable tool in the realm of Java software testing, aiding developers in identifying and addressing potential issues within the code. While it excels in unit testing and seamlessly integrates with the development workflow, its limitations in handling complex tests and asynchronous code necessitate careful consideration during its implementation. Nonetheless, JUnit remains an essential component for ensuring the overall quality and robustness of Java applications.

5.3 Apache JMeter

5.3.1 Introduction:

Apache JMeter, an advanced version of the popular testing tool, offers enhanced collaborative features that enable multiple users to work simultaneously through a server-client setup, leveraging workstations and JVMs. This report critically assesses the strengths and limitations of Apache JMeter's advanced version, emphasizing its collaborative functionalities and their implications for software testing and development teams.

5.3.2 Advantages of Apache JMeter's Advanced Version:

Facilitates collaborative work, allowing multiple team members to contribute to the testing process simultaneously, thereby expediting the testing workflow.

Enables the setup of a server-client architecture, providing a centralized platform for test management and execution across different workstations.

Offers compatibility with JVMs, ensuring seamless integration with existing Java-based systems and environments, thereby streamlining the testing process within Java development projects.

5.3.2 Disadvantages of Apache JMeter's Advanced Version:

Complexity in setting up and maintaining the server-client architecture may pose challenges for teams with limited technical expertise, potentially leading to delays in the testing process.

The increased reliance on JVMs may result in resource constraints and performance issues, particularly when dealing with large-scale or resource-intensive testing scenarios, thereby affecting the overall testing efficiency and accuracy.

5.3.4 Applications of Apache JMeter's Advanced Version:

Ideal for collaborative testing projects that involve multiple team members working simultaneously on various testing components within the same testing environment.

Suitable for the centralized management and execution of complex testing scenarios, ensuring a coordinated and efficient testing process across different workstations.

Widely utilized in Java-based software development projects that require comprehensive testing solutions integrated with existing Java Virtual Machine environments.

5.3.5 Conclusion:

In conclusion, the advanced version of Apache JMeter introduces valuable collaborative features and server-client capabilities, enhancing the testing process for teams working on Java-based projects. Despite its advantages, the complexities associated with setting up and maintaining the server-client architecture, along with potential resource constraints linked to JVM usage, should be carefully considered to ensure a seamless and efficient testing experience for development teams.

We have selected Selenium as our testing framework for our agile project, which consists of small, integrated components. Selenium is a versatile tool that can be used to automate end-to-end, unit, and integration tests, while JUnit and Apache JMeter are not as well-suited for this type of testing.

5.2 Test Cases

5.2.1 New user registration :

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-001	User Registration	Successful New User Registration	The user is on the registration page	1. Enter valid user details (name, email, password). 2. Click the "Register" button	Valid user details (e.g., Name: John Doe, Email: john@example.com, Password: securepwd123)	The user is registered successfully, and they are directed to their profile page.	The user is registered successfully and directed to their profile page.	Pass
UT-002	User Registration	User Registration with Invalid Email	The user is on the registration page.	1. Enter invalid user details with an invalid email format. 2. Click the "Register" button	Invalid email format (e.g., Name: Jane Doe, Email: invalid-email, Password: pwd123)	User registration should fail, and an error message is displayed.	User registration fails, and an error message is displayed.	Pass
UT-003	User Registration	User Registration with Weak Password	The user is on the registration page.	1. Enter valid user details with a weak password. 2. Click the	Weak password (e.g., Name: Alice Smith,	User registration should fail due to a weak password,	User registration fails due to a weak passwor	Pass

				"Register" button.	Email: alice@example.com, Password: weak).	and an error message is displayed.	d, and an error message is displaye d.	
UT-004	User Registration	User Registration with Existing Email	The user is on the registration page.	1. Enter valid user details with an email that is already registered. 2. Click the "Register" button.	Existing email (e.g., Name: Mark Johnson, Email: existing @email.com, Password: newpwd 123).	User registration should fail due to the existing email, and an error message is displayed.	User registration fails due to the existing email, and an error message is displaye d.	Pass
UT-005	User Registration	User Registration with Special Characters in the Name	The user is on the registration page.	1. Enter user details with special characters in the name field. 2. Click the "Register" button.	Name with special characters (e.g., Name: Robert #Smith, Email: robert@example.com, Password: strongpwd123).	User registration should fail due to special characters in the name, and an error message is displayed.	User registration fails due to special characters in the name, and an error message is displaye d.	Pass

Black-box testing, also known as behavioral or functional testing, was employed to evaluate the user registration process from the user's perspective. This approach focuses on the system's external behavior without delving into the internal code structure. By executing test cases UT-001, UT-002, UT-003, and UT-004, we assessed the system's response to various user inputs, including valid and invalid email formats, weak passwords, and existing email

addresses. These test cases were designed to ensure that the registration process functions as intended from the user's standpoint, regardless of the underlying implementation details.

White-box testing, also known as structural or code-based testing, was employed in test case UT-005 to verify the validation rules for the name field. This test case specifically targeted the code logic responsible for rejecting inputs containing special characters. By entering a name with special characters, the tester confirmed that the code adhered to the defined validation rules and prevented invalid name registrations. This demonstrates the effectiveness of white-box testing in ensuring the proper functioning of specific code segments and validation mechanisms.

5.2.2 User login:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	User Login	Verify that a user can log in with a valid username and password.	User must be a registered user.	1. Navigate to the login page. 2. Enter a valid username in the username field. 3. Enter a valid password in the password field. 4. Click the Login button.	Valid username and password Username: John Doe Password: securepwd123	User is successfully logged in.	User is successfully logged in.	Pass

UT-02	User Login	Verify that a user cannot log in with an empty username.	User must be a registered user.	1. Navigate to the login page. 2. Leave the username field blank. 3. Enter a valid password in the password field. 4. Click the Login button.	Empty username. Username: Password:123	An error message is displayed indicating that the username is required.	An error message is displayed indicating that the username is required.	Pass
UT-03	User Login	Verify that a user cannot log in without entering a username and password.	User must be a registered user.	1. Navigate to the login page. 2. Click the Login button.	No username and password entered. Username: Password:	An error message is displayed indicating that the username and password are required.	An error message is displayed indicating that the username and password are required.	Pass
UT-04	User Login	Verify that a user cannot log in with an empty password.	User must be a registered user.	1. Navigate to the login page. 2. Enter a valid username in the username field. 3. Leave the password field blank. 4. Click the Login button.	Empty password Username: John Doe Password:	An error message is displayed indicating that the password is required.	An error message is displayed indicating that the password is required.	Pass

UT-05	User Login	Verify that a user cannot log in with a valid username and password that do not match.	User must be a registered user.	1. Navigate to the login page. 2. Enter a valid username in the username field. 3. Enter an incorrect password in the password field. 4. Click the Login button.	Valid username and incorrect password. Username;John Doe Password:123	An error message is displayed indicating that the username and password do not match.	An error message is displayed indicating that the username and password do not match.	Pass

The given test cases are all classified as black-box testing because they focus on the system's external behavior from the user's perspective. They do not require knowledge of the internal code structure to execute. For example, test case UT-01 verifies that a user can log in with a valid username and password. This test case can be executed without knowing how the login functionality is implemented in the code. The tester simply needs to provide valid input data and observe the system's response.

Black-box testing is a valuable approach for validating the functionality of a system from the user's perspective.

5.2.3 Report found item:

Test Case ID	**Name of Module**	**Test Case Description**	**Pre-conditions**	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Test Result**
UT-01	Report Found Item	Verify the ability to report a found item with valid inputs.	User is logged in and navigated to the "Report Found Item" form.	1. Enter valid item category. 2. Enter valid founder name.	- Item Category: Electronics - Founder Name: John Doe	Found item should be successfully reported, and a confirmation message	Found item reported successfully. Confirm	Pass

				3. Enter valid item name. 4. Enter valid contact number. 5. Upload a valid image of the found item. 6. Add a valid description of the found item. 7. Click on the "Submit" button.	- Item Name: Laptop - Contact Number: 123-456-7890 - Description: Laptop found in the library.	should be displayed.	ation message displayed.	
UT-02	Report Found Item Integration	Verify the integration of the "Report Found Item" form with the database.	User has submitted the found item form.	Check the database for the reported item	None	Found item data should be stored in the database	Found item data is stored in the database.	Pass
UT-03	Report found item front-end	Verify the user interface and user experience of the "Report Found Item" form.	User is on the "Report Found Item" form page. .	1. Check the layout and alignment of the form fields. 2. Verify the upload functionality for images 3. Verify error messages for	None	Form fields should be well-organized, image upload should work, and appropriate error messages should be displayed for	Form fields are organized, image upload works, and correct error messages are	Pass

				missing or invalid inputs.		invalid inputs.	displayed for invalid inputs.	
UT-04	Image Validation	Verify the validation of uploaded images in the "Report Found Item" form.	User is on the "Report Found Item" form page.	Attempt to upload an image with an unsupported format (e.g., .txt file).	Invalid Image: test.txt	Form should reject the invalid image file and display an error message.	Form rejects the invalid image file and displays an appropriate error message.	Pass
UT-05	Field Length Validation	Verify the maximum length validation for various fields in the "Report Found Item" form.	User is on the "Report Found Item" form page.	Attempt to input data exceeding the maximum allowed length for each field.	- Founder Name: A name with more than 50 characters. - Item Name: A name with more than 50 characters. - Contact Number: A number with more than 15 digits.	Form should reject input data exceeding the maximum allowed length and display an error message.	Form rejects input data exceeding the maximum allowed length and displays appropriate error messages.	Pass
UT-06	Duplicate Submission	Report found	Verify the functionality	User has entered data	None	Form fields should be	Form fields	Pass

		item form reset	of the "Reset" button in the "Report Found Item" form.	in the form fields.		cleared/reset to the default state.	are cleared/reset to the default state.	

All six test cases are classified as black-box testing because they focus on the system's external behavior from the user's perspective and do not require knowledge of the internal code structure to execute. For example, test case UT-01 verifies that the system can successfully report a found item with valid inputs. This test case can be executed without knowing how the found item reporting functionality is implemented in the code. The tester simply needs to provide valid input data and observe the system's response.

5.2.4 Reporting Lost Item:

Test Case ID	**Name of Module**	**Test Case Description**	**Pre-conditions**	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Test Result**
UT-01	Reporting Lost Item	To test the functioning of posting a lost item report	User is logged into the system and is on the 'Report Lost Items' page	1. Enter the details of the items (category, title, description). 2. Click on the save/ upload button	Category : Electronics Title: Ear pods Description : Blue color pTron buds	Lost item report should be successfully posted	The Lost item report should be successfully posted	Pass

UT-02	Reporting lost item	To test system's validation for mandatory fields when filling report	User is logged in and is on the 'Report Lost Item' page	1. Leave one or more mandatory fields empty 2. Click on Save/upload button	Title: lost Phone Description: Black Samsung A55. Left in cafeteria	The system should display error message indicating that mandatory fields are empty	An error message is displayed and report not posted	Pass
UT-03	Reporting lost item	To test character limit for title in lost item report	User is logged in and is on the 'Report Lost Item' page	1. Enter the title exceeding the character limit 2. Click on save/upload	Category: Books Title: And to my Nephew Alber I leave the island what I won off fatty hagan in a Poker Game Description: Lost in classroom no 304	The system should prevent posting a report with such a long title	An error message is displayed indicating the exceeded limit of title	Pass

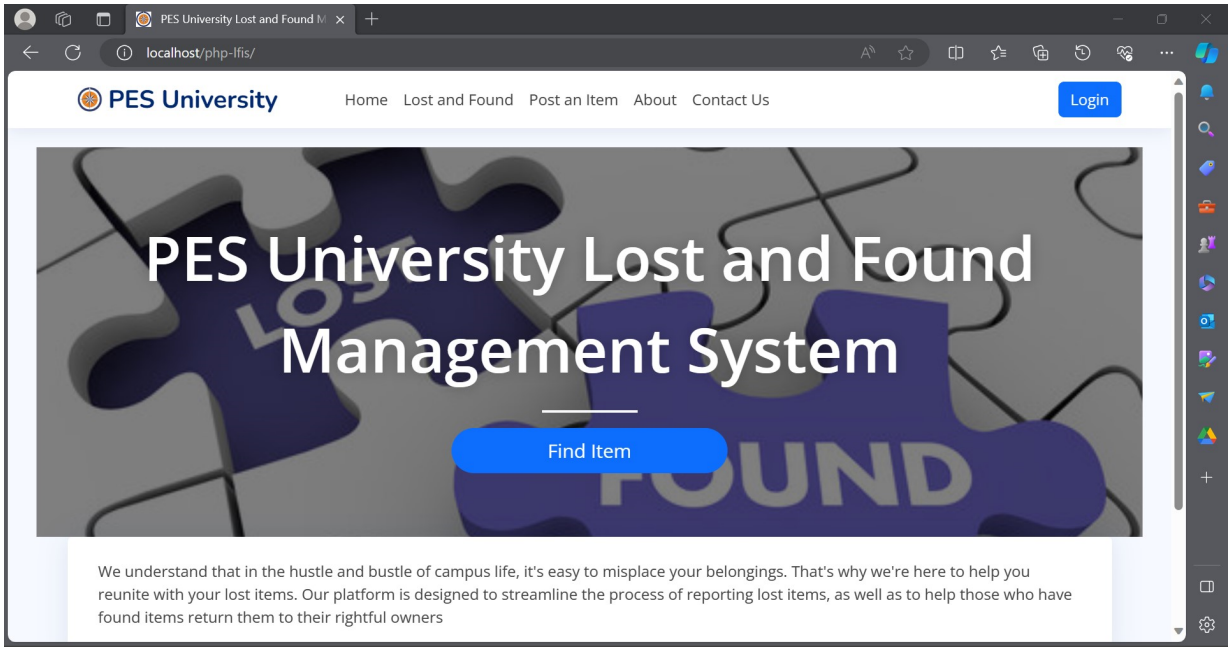
UT-04	Reporting Lost item	To test the system's behavior when user attempts to post using invalid data	User is logged in and is on the 'Report Lost Item' page	1. Enter invalid data in one or more fields 2. Click on save/submit button	Category: Jewelry Title: @#$%#ring Description: Left in the gym	System should prevent posting a report	An error message is displayed and report is not posted	Pass
UT-05	Reporting Lost item	To test item's response when user successfully posts multiple lost item	User is logged in and is on the 'Report Lost Item' page	1. Enter valid data for each report (at least 2 reports) 2. Click on save/submit button	Report 1: Category: Clothing Title: Lost Jacket Description: Left in the gym Report 2: Category: Accessories Title:Lost sunglasses Description: Left on the bus	Multiple lost item reports should be successfully posted and user should receive confirmation message for each report	Multiple lost item reports are successfully posted and user receives confirmation message for each report	Pass

UT-06	Reporting Lost item	Testing the integration with user authentication	User is logged in and is on the 'Report Lost Item' page	1. Log out of the system 2. Attempt to report lost item without logging in	User's authentication status: Unauthorized User account details: N/A	System should prevent unauthenticated users,	Unauthenticated users are redirected to login page	Pass

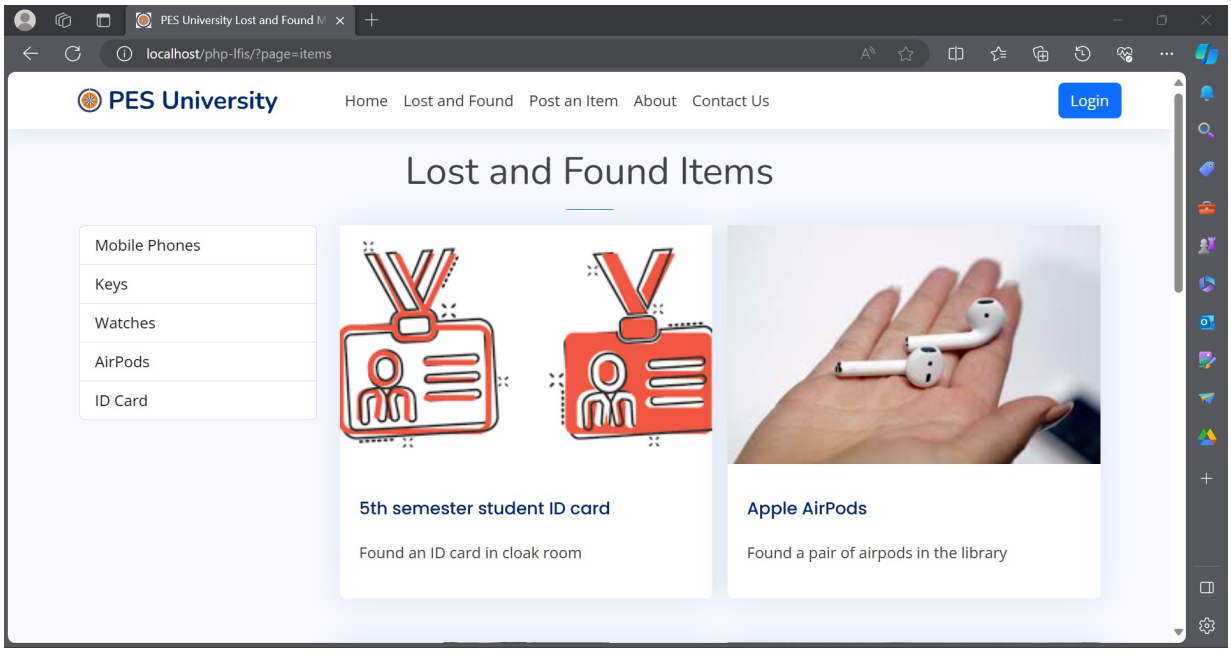
All six test cases are classified as black-box testing because they focus on the system's external behavior from the user's perspective and do not require knowledge of the internal code structure to execute. For example, test case UT-01 verifies that the system can successfully post a lost item report with valid inputs. This test case can be executed without knowing how the lost item reporting functionality is implemented in the code. The tester simply needs to provide valid input data and observe the system's response.

6. Project Outlook:

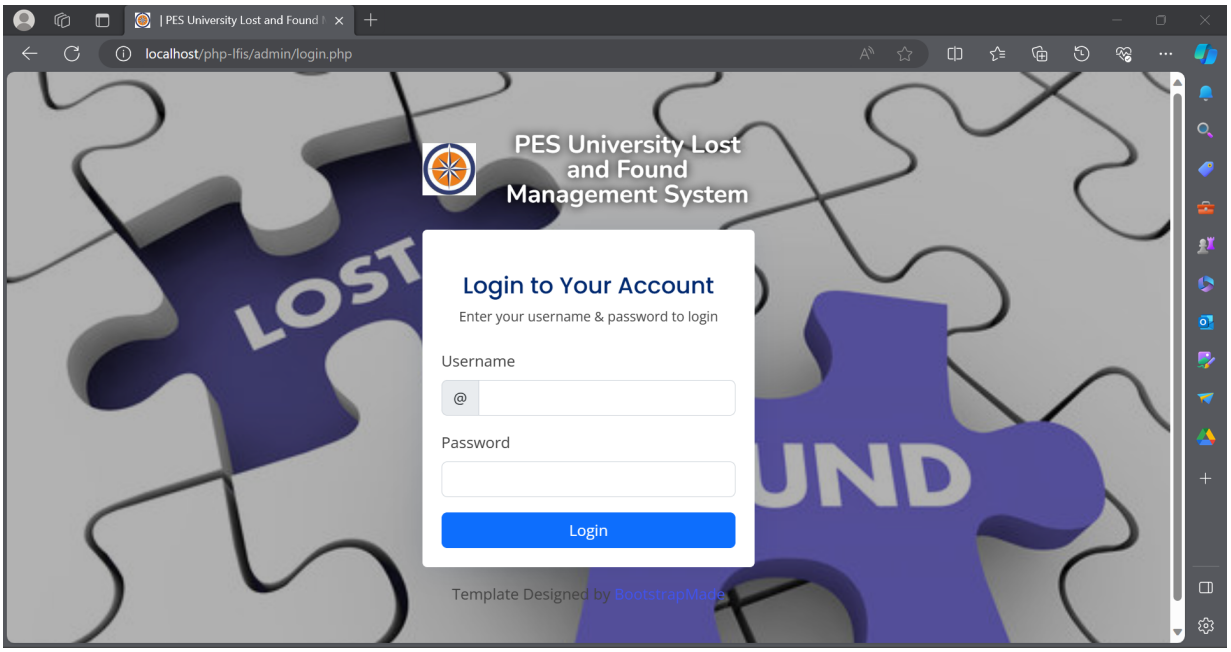
6.1 Home Page:



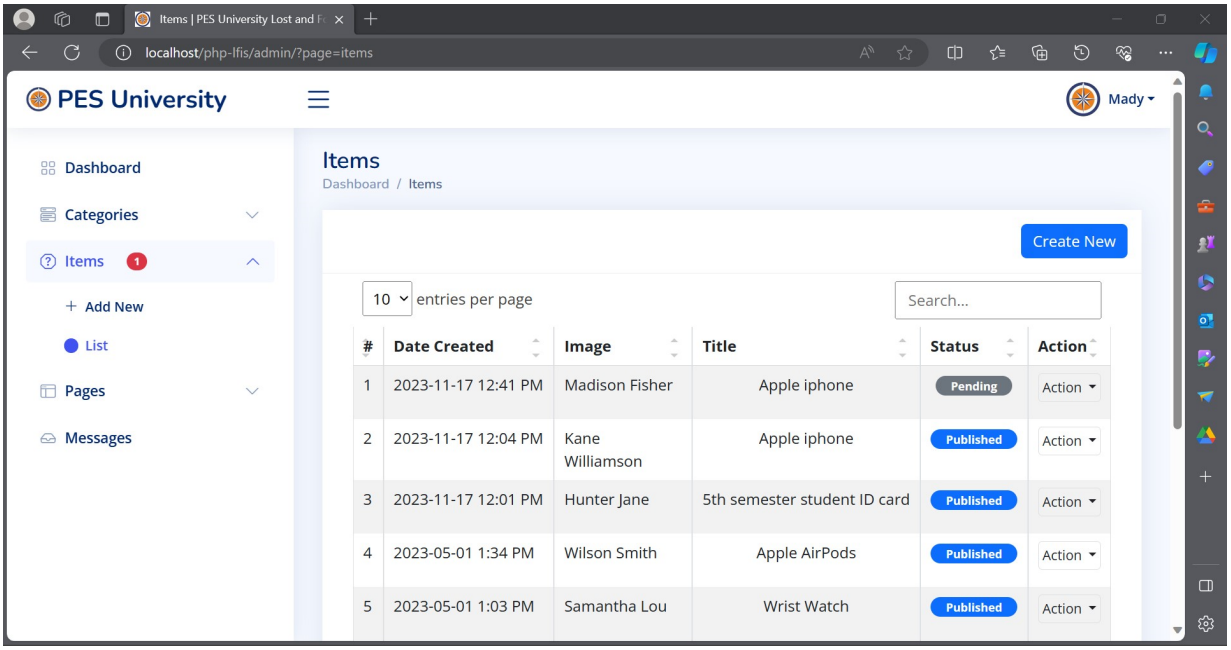
6.2 Lost and Found Items Page:



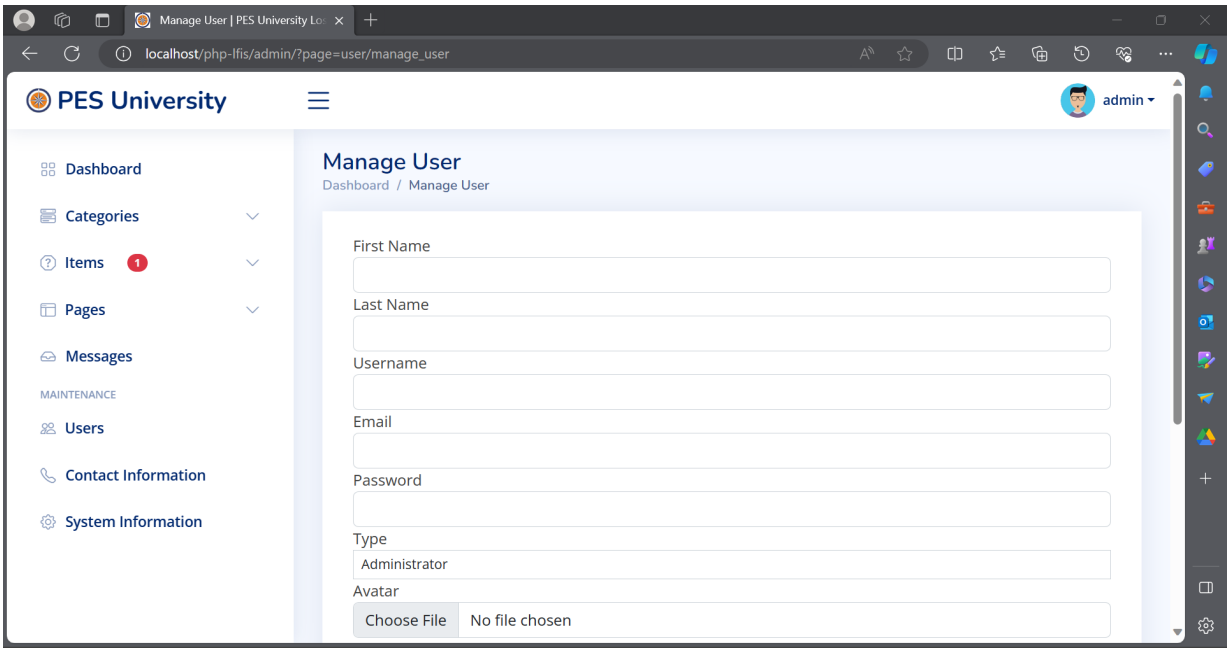
6.3 User Login:



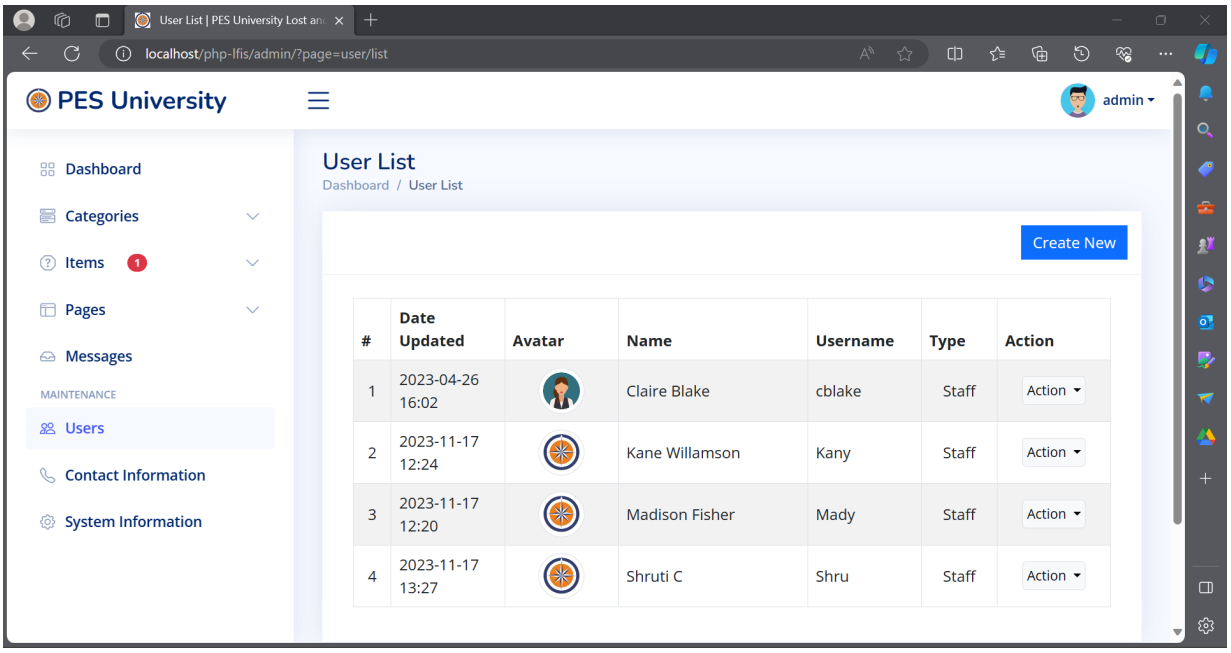
6.4 User Logged in as Staff:



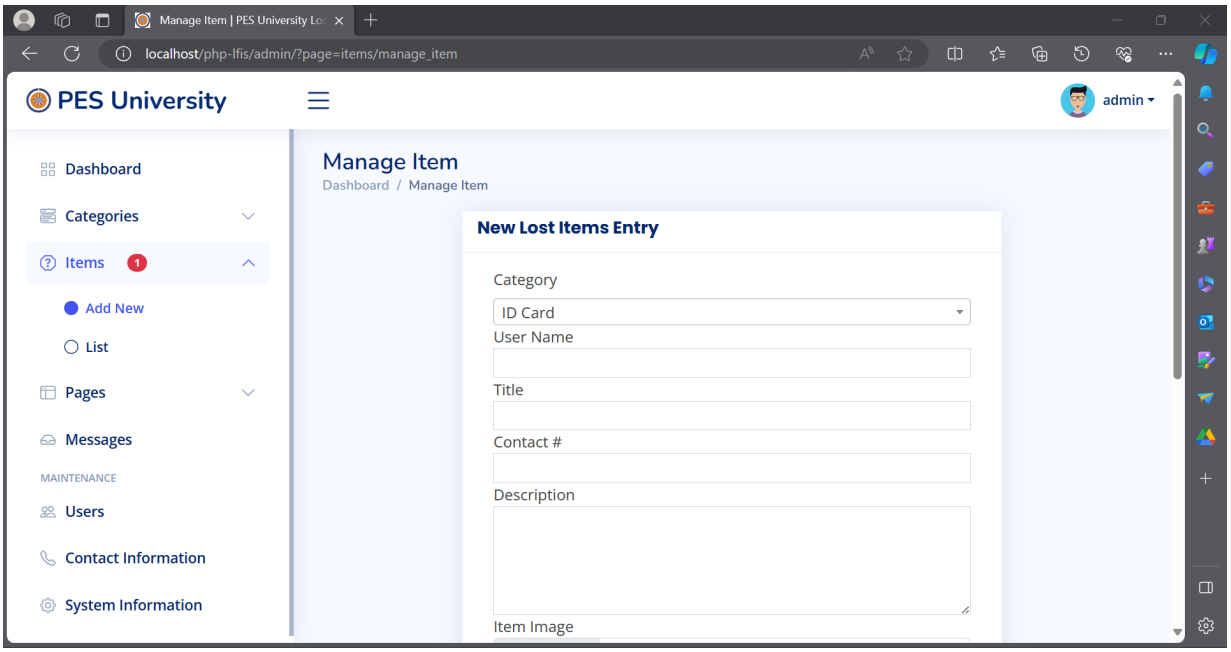
6.5 User Logged in as Admin and Registering a New User:



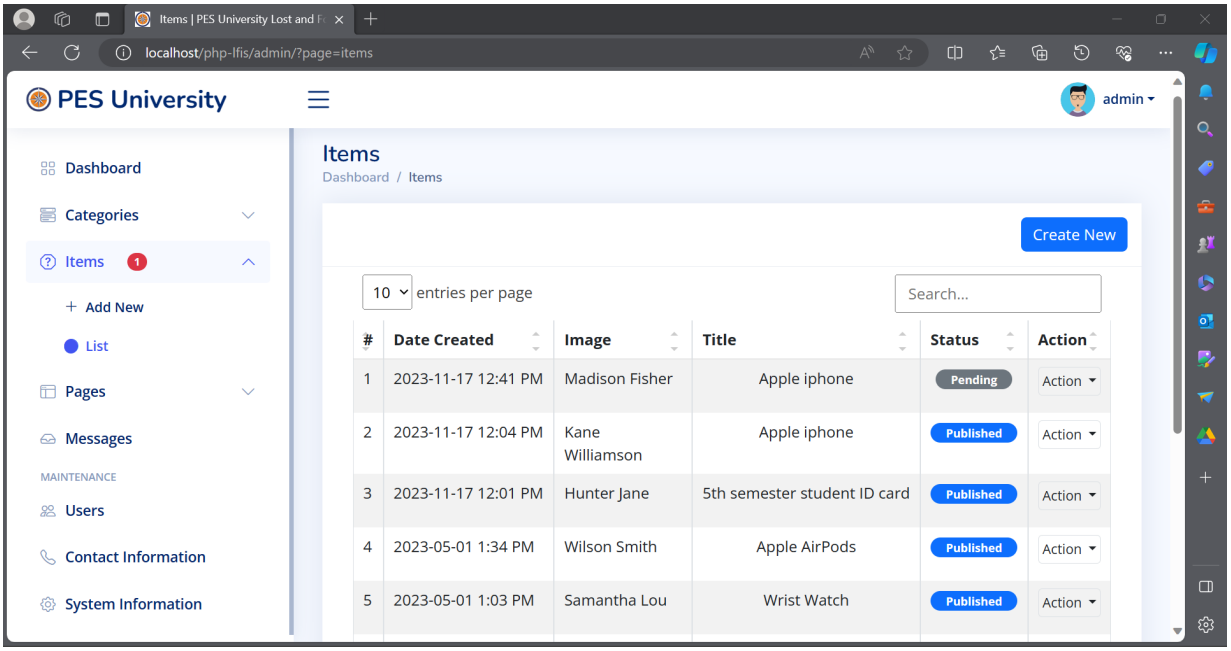
6.6 View of Registered Users:



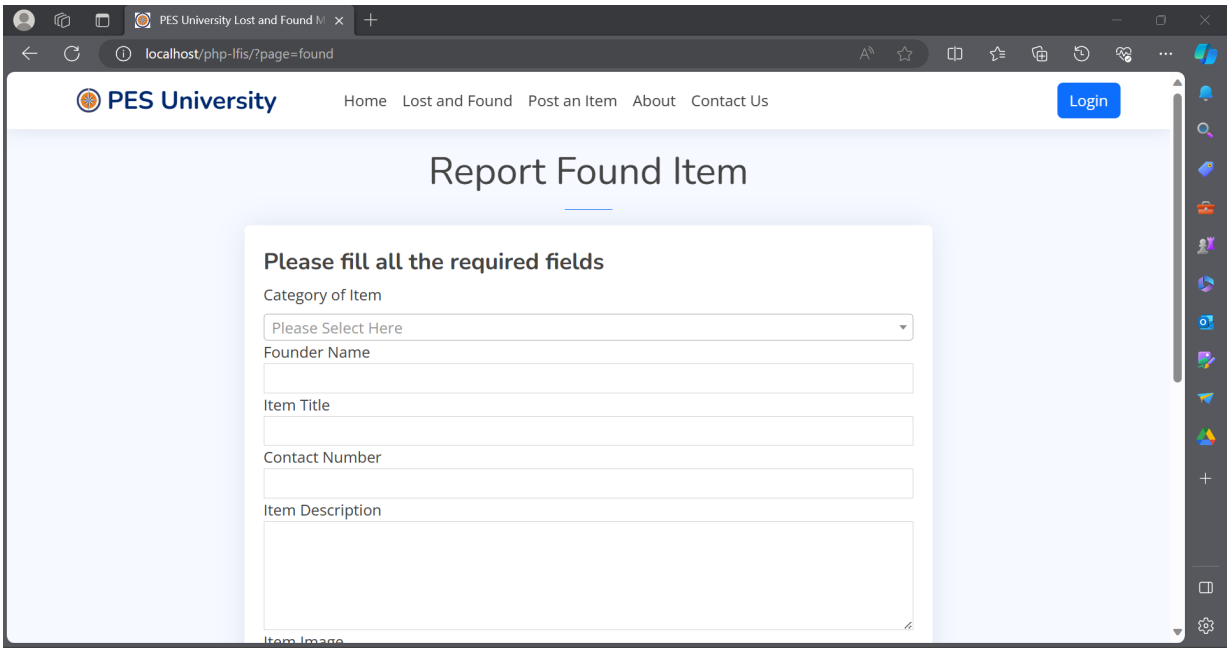
6.7 Report Lost Items:



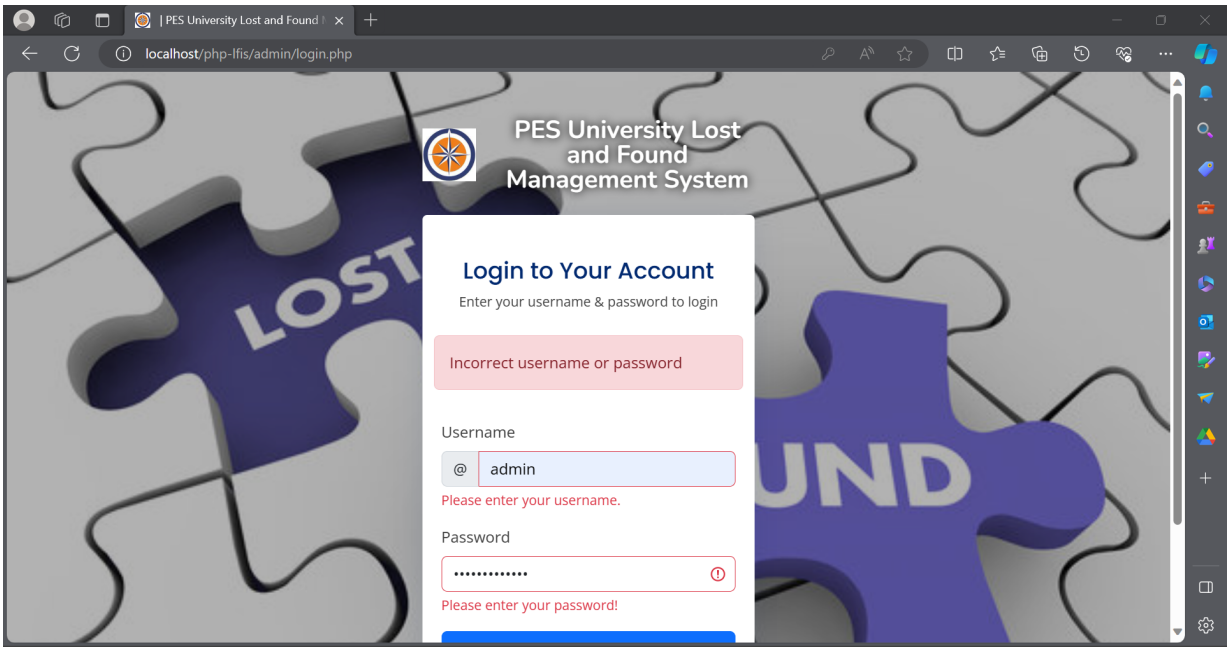
6.8 View of Lost Items:

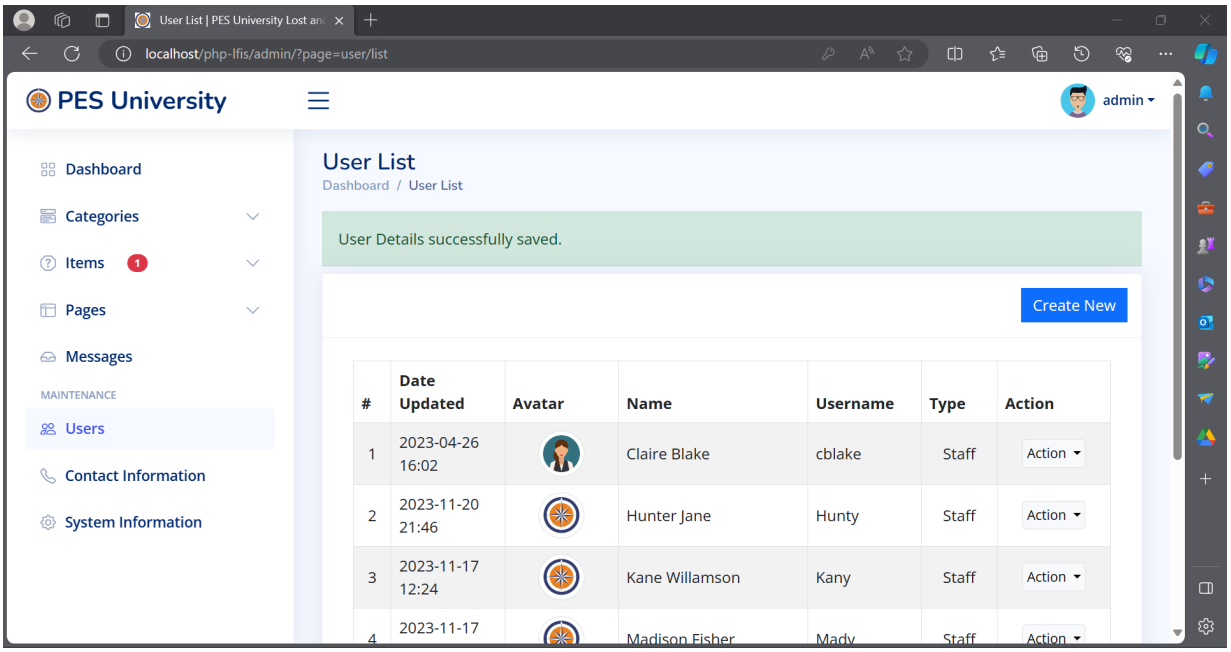


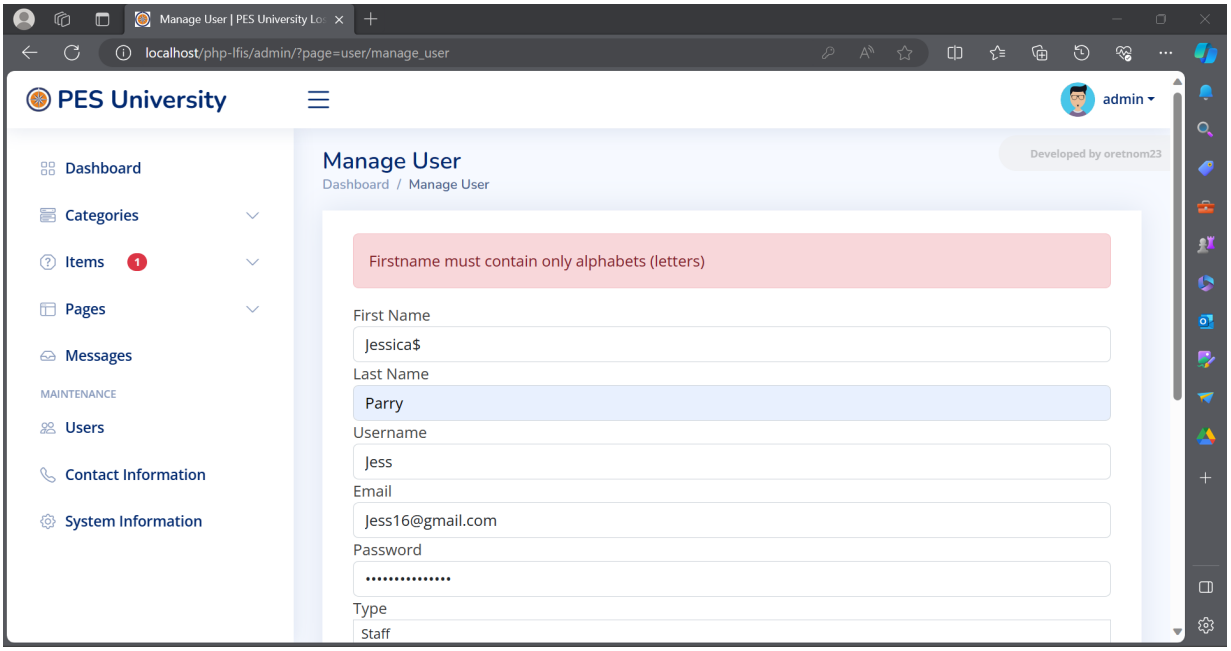
6.9 Report Found Items:

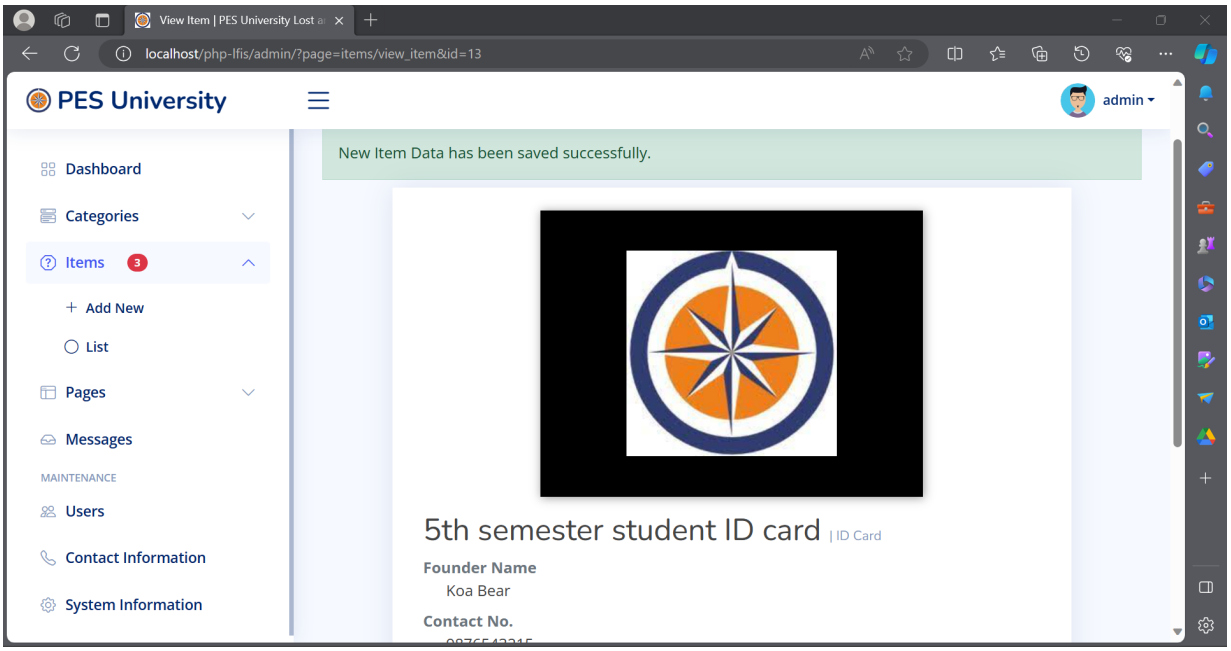


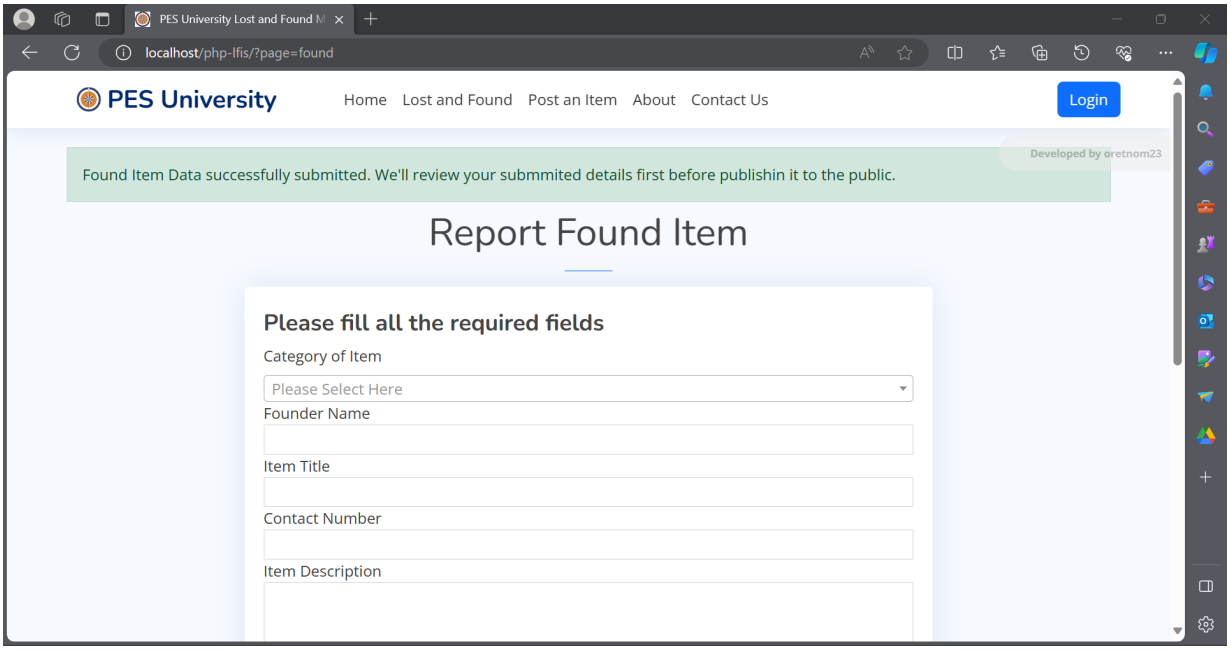
Checking Test Cases:

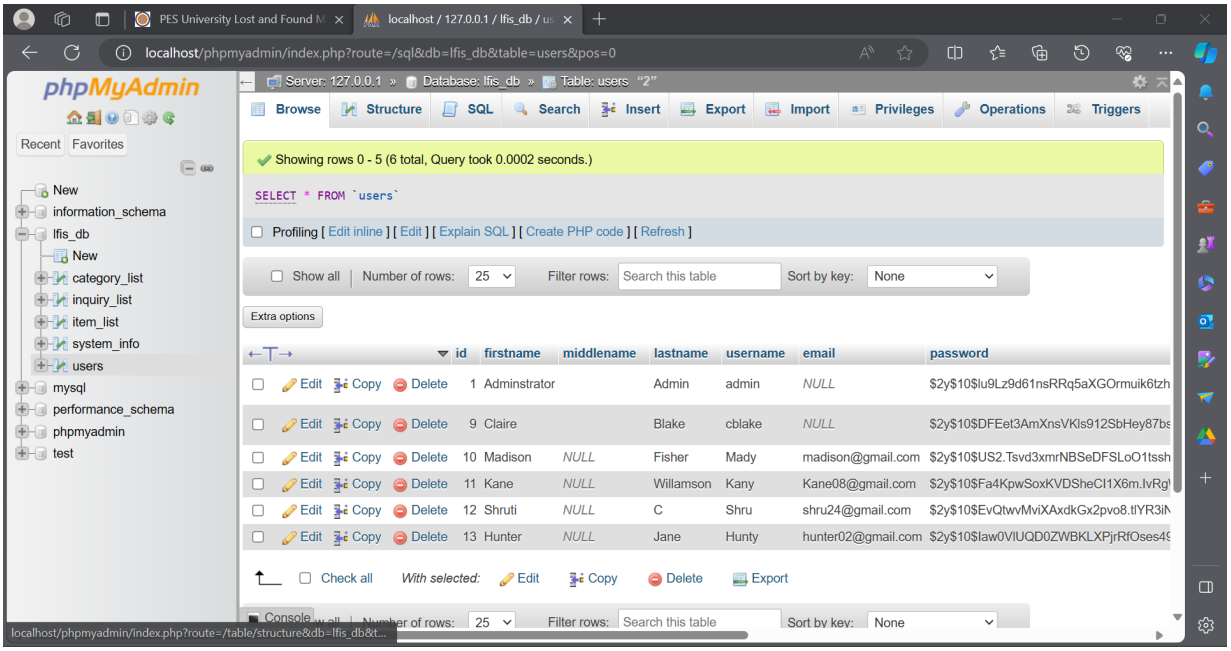


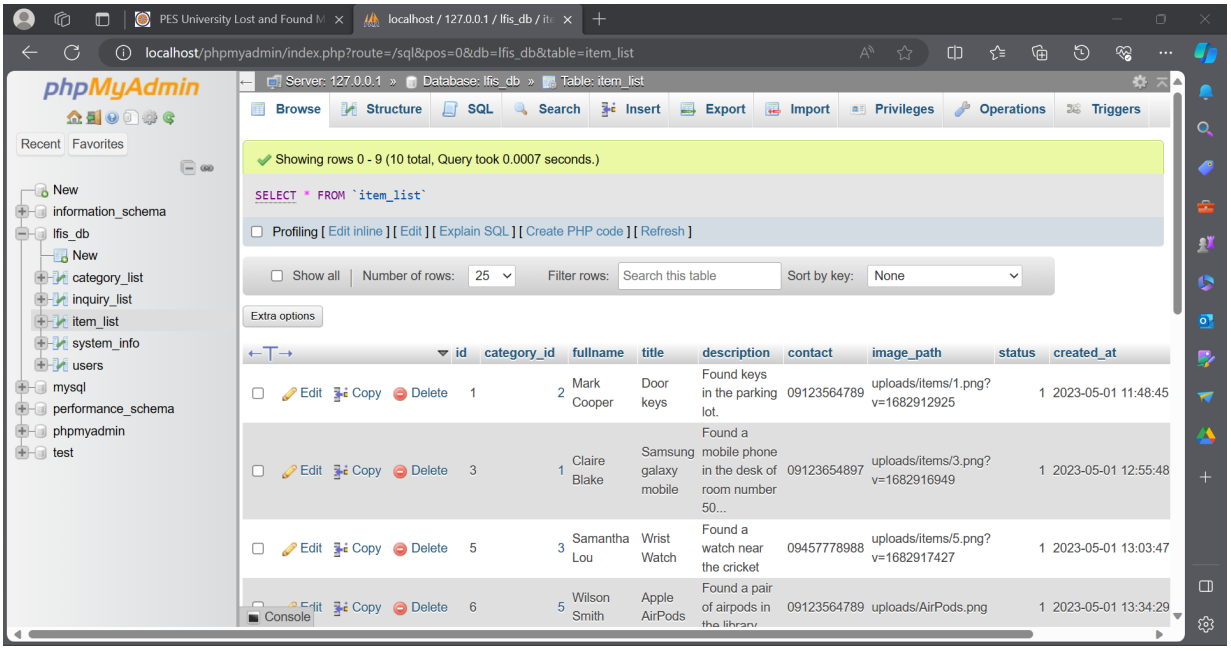



Manage User | PES University Lo
localhost/php-lfis/admin/?page=user/manage_user
PES University
admin
Dashboard
Categories
Items 1
Pages
Messages
MAINTENANCE
Users
Contact Information
System Information
Manage User
Dashboard / Manage User
Developed by oretnom23
Firstname must contain only alphabets (letters)
First Name
Jessica$
Last Name
Parry
Username
Jess
Email
Jess16@gmail.com
Password
Type
Staff


View Item | PES University Lost a
localhost/php-lfis/admin/?page=items/view_item&id=13
PES University
admin
Dashboard
Categories
Items 3
Add New
List
Pages
Messages
MAINTENANCE
Users
Contact Information
System Information
New Item Data has been saved successfully.
5th semester student ID card
ID Card
Founder Name
Koa Bear
Contact No.


PES University Lost and Found M
localhost/php-lfis/?page=found
PES University
Home
Lost and Found
Post an Item
About
Contact Us
Login
Developed by oretnom23
Found Item Data successfully submitted. We'll review your submmited details first before publishin it to the public.
Report Found Item
Please fill all the required fields
Category of Item
Please Select Here
Founder Name
Item Title
Contact Number
Item Description


PES University Lost and Found M
localhost / 127.0.0.1 / lfis_db /
localhost/phpmyadmin/index.php?route=/sql&db=lfis_db&table=users&pos=0
phpMyAdmin
Recent
Favorites
New
information_schema
lfis_db
New
category_list
inquiry_list
item_list
system_info
users
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: lfis_db » Table: users "2"
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers
Showing rows 0 - 5 (6 total, Query took 0.0002 seconds.)
SELECT * FROM `users`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options
id
firstname
middlename
lastname
username
email
password
Edit Copy Delete 1 Adminstrator Admin admin NULL $2y$10$lu9Lz9d61nsRRq5aXGOrmuik6tzh
Edit Copy Delete 9 Claire Blake cblake NULL $2y$10$DFEet3AmXnsVKls912SbHey87bs
Edit Copy Delete 10 Madison NULL Fisher Mady madison@gmail.com $2y$10$US2.Tsvd3xmrNBSeDFSLoO1tssh
Edit Copy Delete 11 Kane NULL Willamson Kany Kane08@gmail.com $2y$10$Fa4KpwSoxKVDSheCI1X6m.IvRg
Edit Copy Delete 12 Shruti NULL C Shru shru24@gmail.com $2y$10$EvQtwvMviXAxdkGx2pvo8.tlYR3iN
Edit Copy Delete 13 Hunter NULL Jane Hunty hunter02@gmail.com $2y$10$law0VIUQD0ZWBKLXPjrRfOses4
Check all
With selected:
Edit
Copy
Delete
Export
Console
Number of rows: 25
Filter rows: Search this table
Sort by key: None
localhost/phpmyadmin/index.php?route=/table/structure&db=lfis_db&t...



Appendix A: Glossary

Term	Definition
SRS (System Requirements Specification)	Document is to provide a comprehensive overview of the University-Level Lost and Found Management System
REQ (Requirements)	Requirements in an SRS (Software Requirements Specification) are descriptions of what the software system must do and how it must perform.
OAuth	OAuth is an open standard for access delegation, commonly used as a way for internet users to grant websites or applications access to their information on other websites but without giving them the passwords.
PHP Mailer API	The PHPMailer API is a PHP library that provides a simple and flexible way to send emails from a PHP script.
MySql API	The MySQL API is a set of functions and classes that developers can use to interact with MySQL databases.

Security Patches	A security patch is a software update that fixes a security vulnerability. Security vulnerabilities are weaknesses in software that can be exploited by attackers to gain access to systems or data.

Appendix B:

Field Layout:

Field	Length	Data Type	Description	Is Mandatory
Item ID	5	Alphanumeric		Y
Item Name	20	String		Y
Category	8	String	The category to which the item belongs	Y
Color	8	String	Color of the lost item	Y
Size	25	Alphanumeric	Size of the lost item	Y
User Name	60	String		Y
Brand	4	String	Reject Reason code in case mandate is rejected	N
Item Description	50	String	Additional details about item appearance	

Lost location	25	String	Location where the item was lost	Y
Finder's Name	20	String	Name of the finder	Y
Owner's Name	20	String	Name of person claiming ownership	Y
Contact Information	10	Numeric	Contact information of the owner	Y

Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1.	REQ-01	Users can report lost items	UseCase Diagram	Use Case Specification	Implementation File	TC-01	STC-01
2.	REQ-02	Users can report found items.	UseCase Diagram	Use Case Specification	Implementation File	TC-02	STC-02
3.	REQ-03	Record lost item details: description, category, and location.	Entity-Relationship Diagram	Entity-Relationship Diagram	Data Model File	TC-03	STC-03
4.	REQ-04	Record found item details: description, category, and location.	Entity-Relationship Diagram	Entity-Relationship Diagram	Data Model File	TC-04	STC-04

5.	REQ-05	Users can claim ownership of found items.	UseCase Diagram	Use Case Specification	Implementation File	TC-05	STC-05
6.	REQ-06	Associate found items with claimants.	Use Case Diagram	Use Case Specification	Implementation File	TC-06	STC-06
7.	REQ-07	Track the location where items were lost.	Entity-Relationship Diagram	Entity-Relationship Diagram	Data Model File	TC-07	STC-07
8.	REQ-08	Track the location where items were found.	Entity-Relationship Diagram	Entity-Relationship Diagram	Data Model File	TC-08	STC-08
9.	REQ-09	Users can search for lost items.	Use Case Diagram	Use Case Specification	Implementation File	TC-09	STC-09
10.	REQ-10	Users can search for found items.	Use Case Diagram	Use Case Specification	Implementation File	TC-10	STC-10

GitHub Link: https://github.com/ShruC24/Team-11