PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Design Document for University Lost and Found Management System

Team 11 Members:

Shreya Joshi - PES2UG21CS501

Shruti C - PES2UG21CS514

Spoorthi Shivaprasad - PES2UG21CS536

Sragvi Anil Shetty - PES2UG21CS537

Assignment Details: Design Document

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:

Develop a Data Flow Diagram (DFD) for your project:

· Level 0 (Context Diagram): This is the top level of the DFD, which provides a bird's eye view of the system. It should include external entities and how they interact with the main system.

· Level 1: This expands the main system from Level 0 and shows its main functions. It should contain processes, data stores, and data flow among them.

· (Optional) Level 2: Further break down the processes from Level 1 for more intricate details.

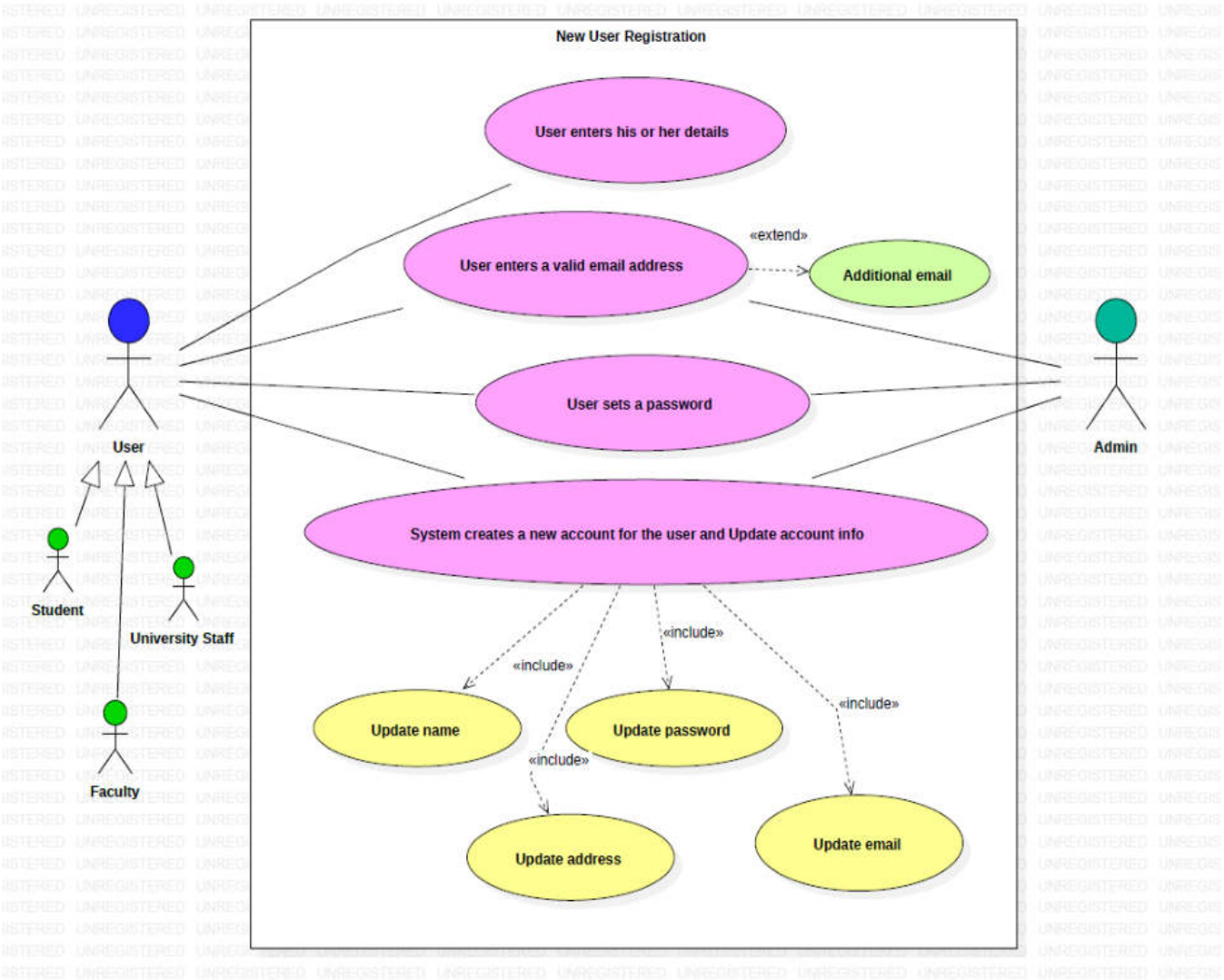
4. Architectural Style Integration:

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

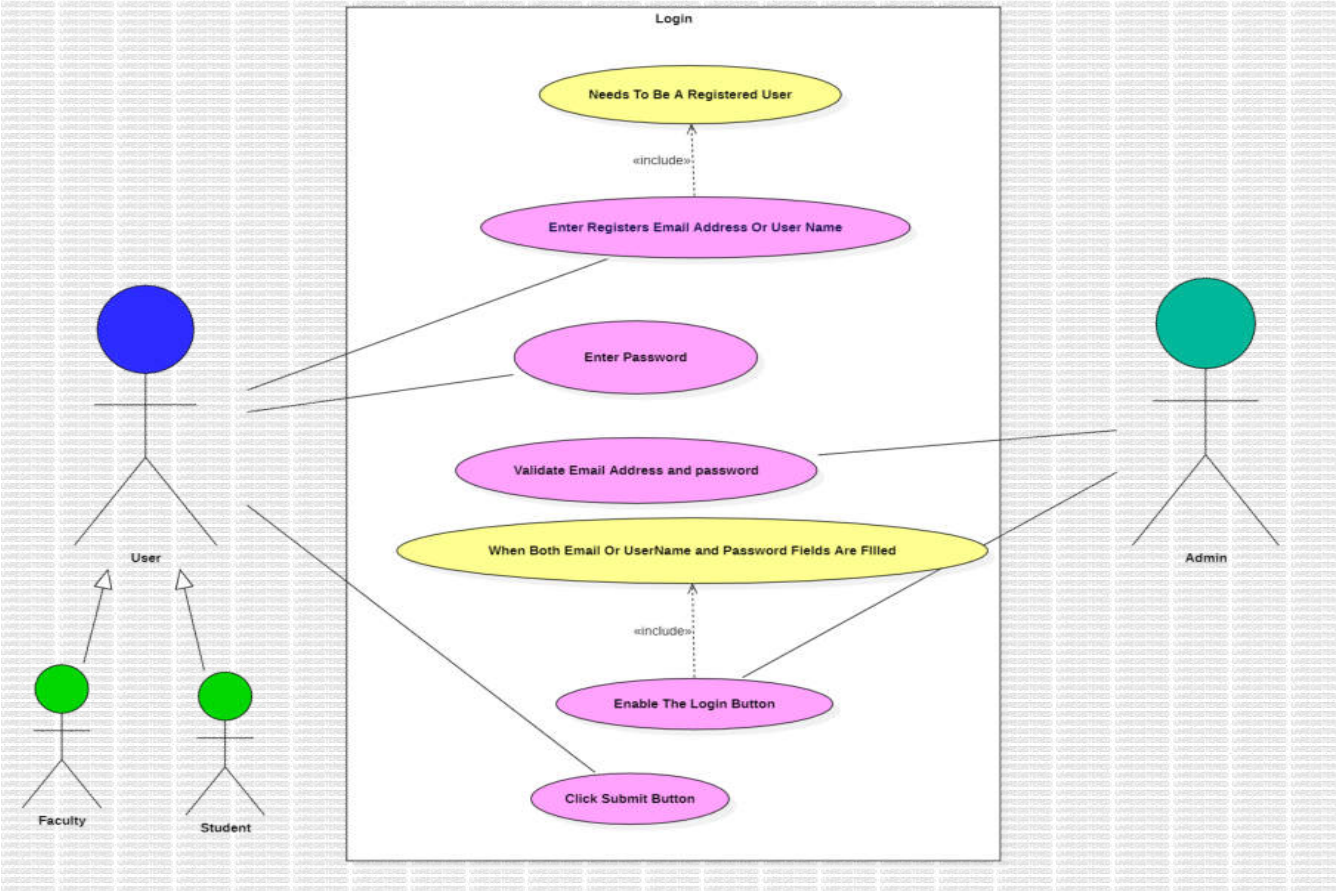
· Describe and justify the chosen architectural style for your system. Explain how this style is suitable for your system's requirements and how it will benefit the project in terms of scalability, maintainability, performance, etc.

Use case diagrams and Class diagrams

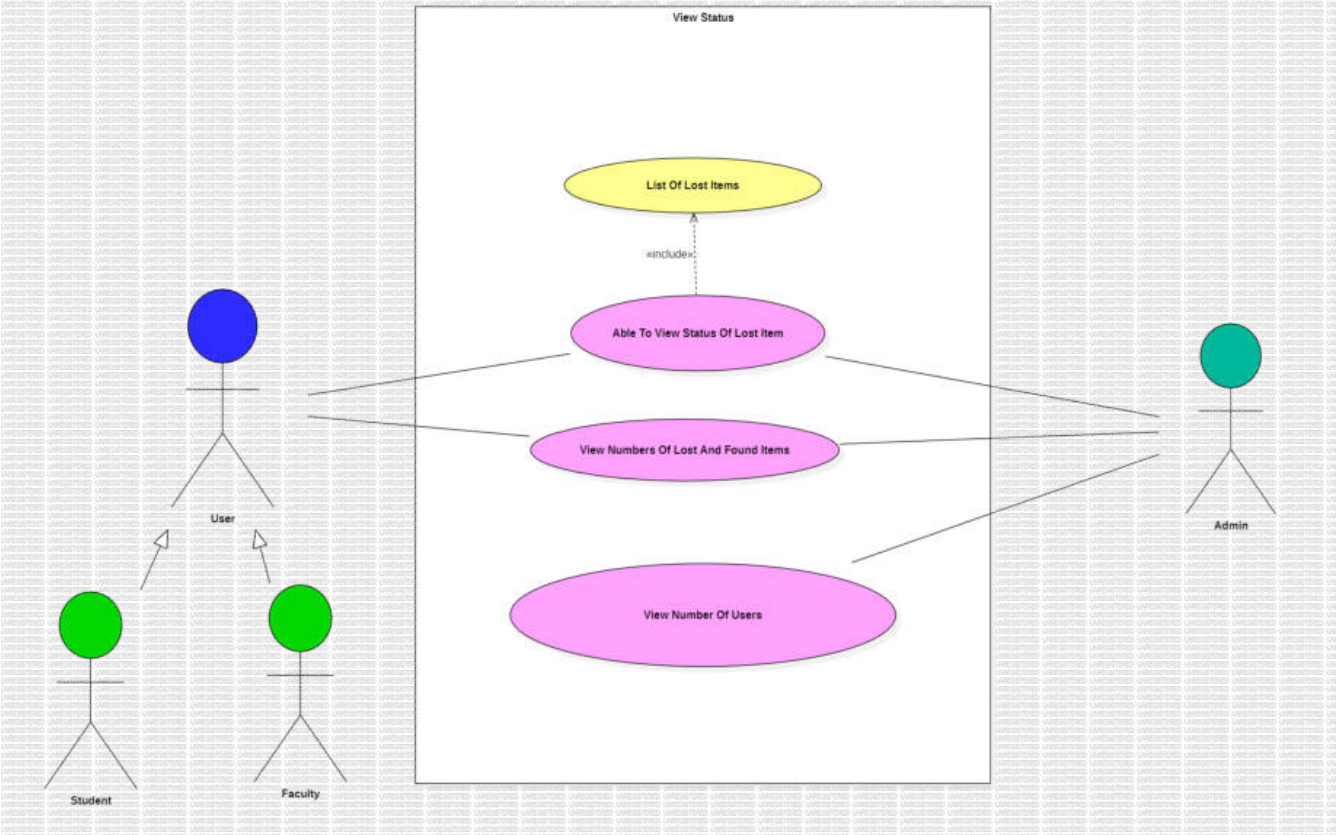
2. New User Registration (SRN:PES2UG21CS514):



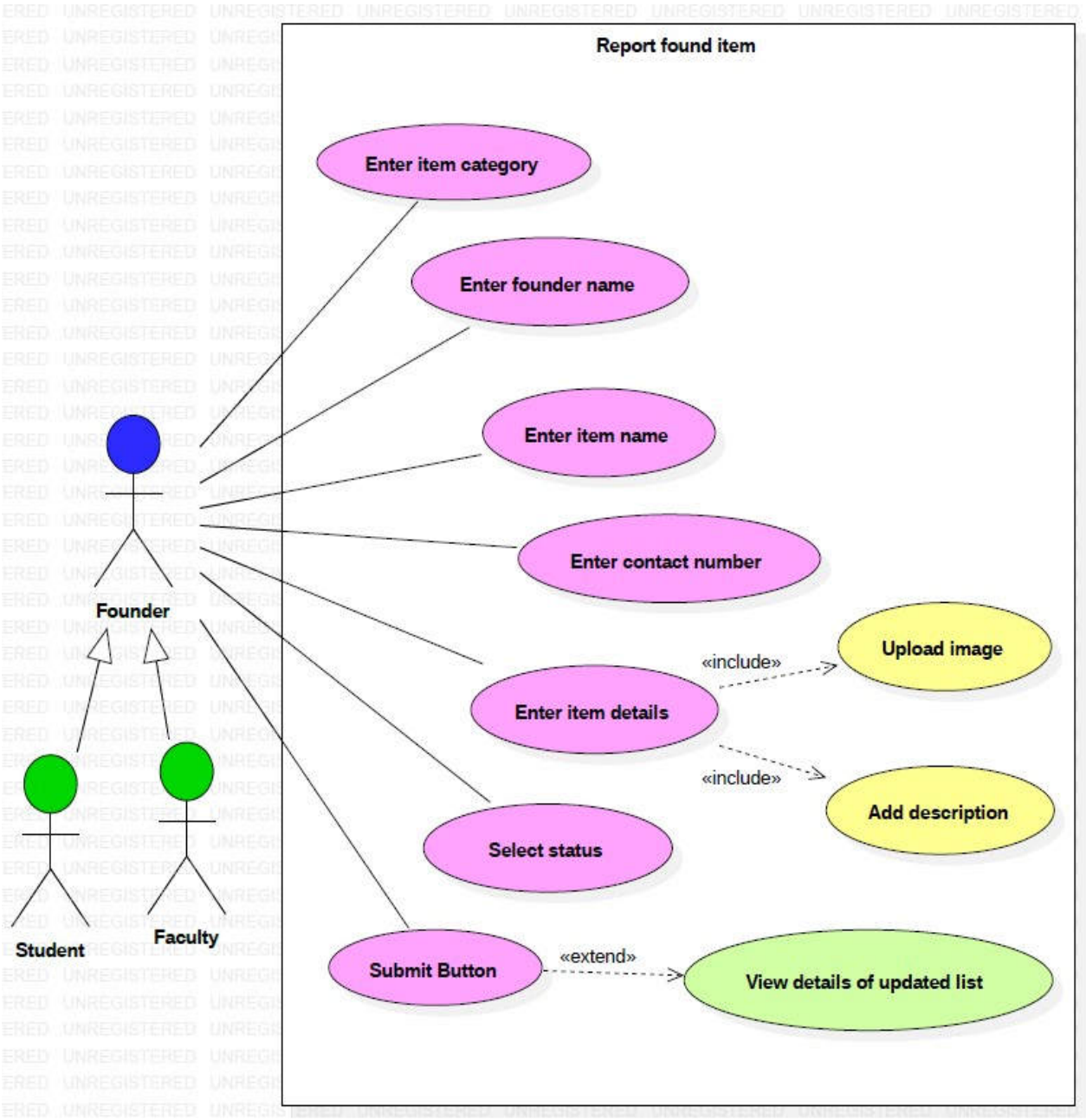
Login (SRN: PES2UG21CS536):



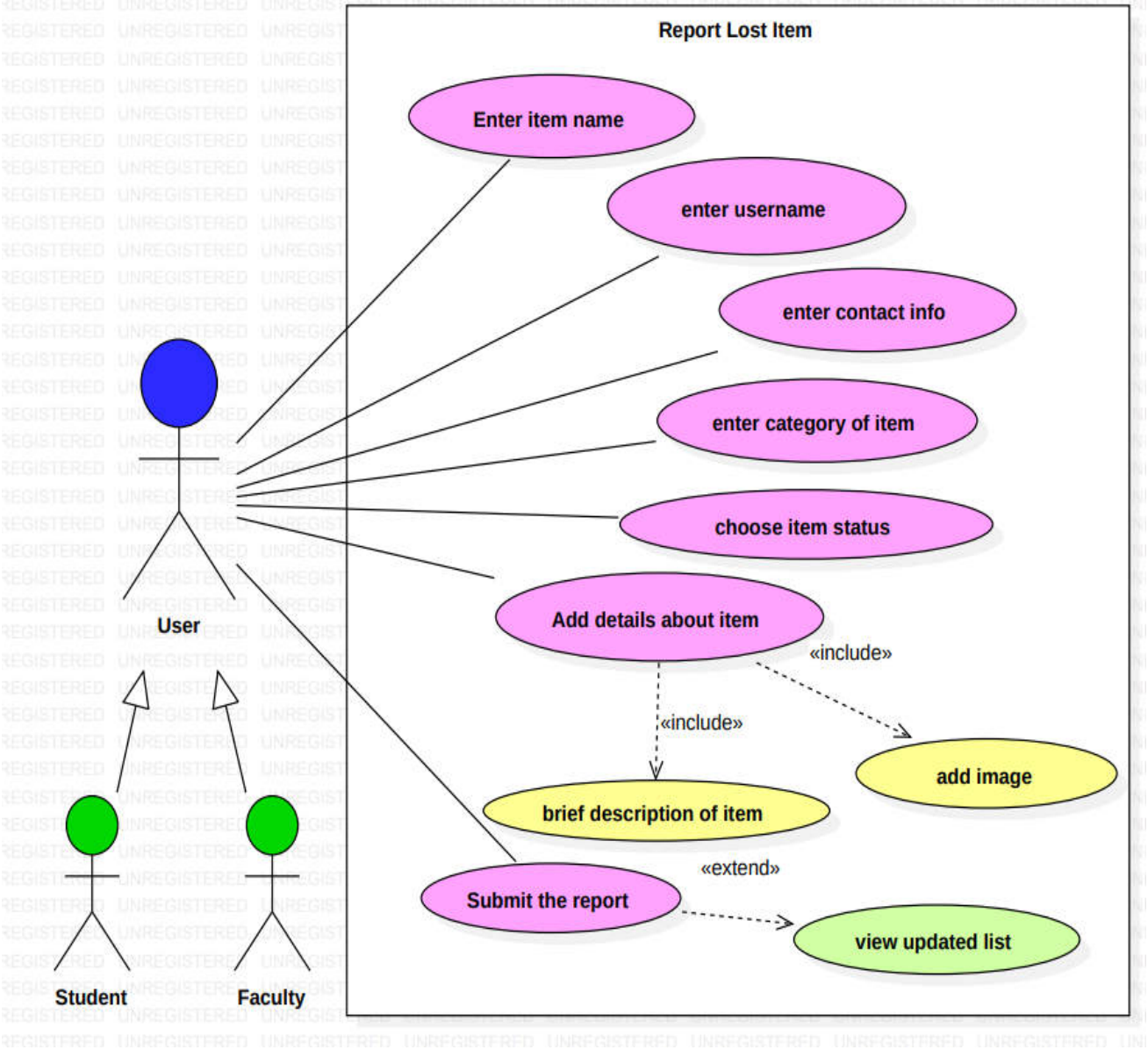
View Status:



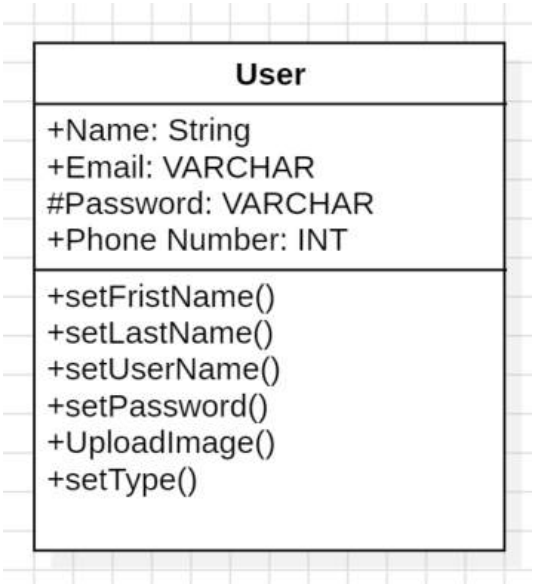
Report found item (SRN: PES2UG21CS501):

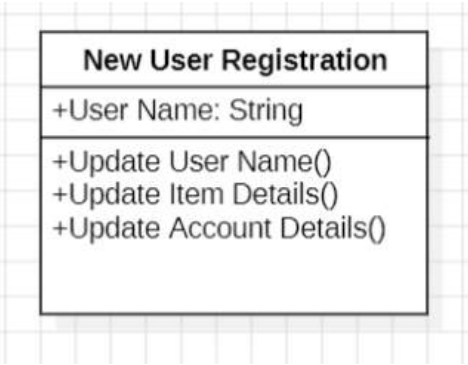


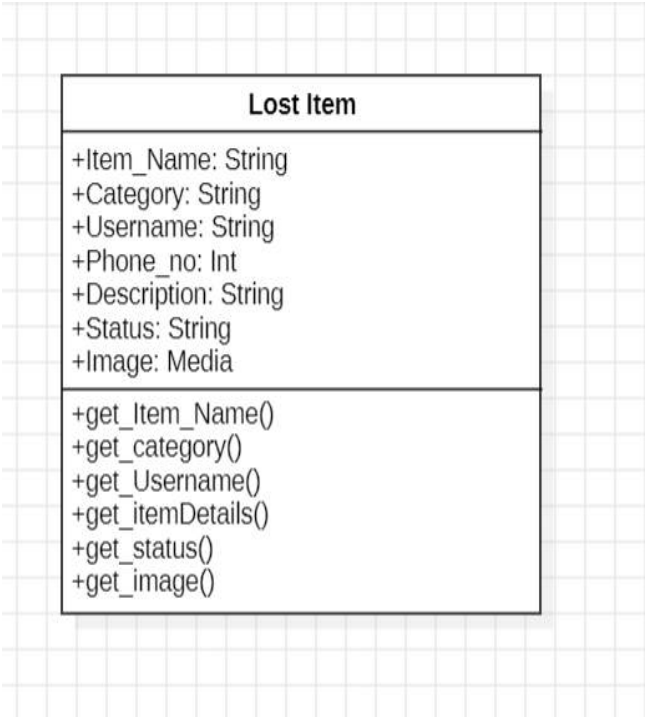
Report Lost Item (SRN: PES2UG21CS537):

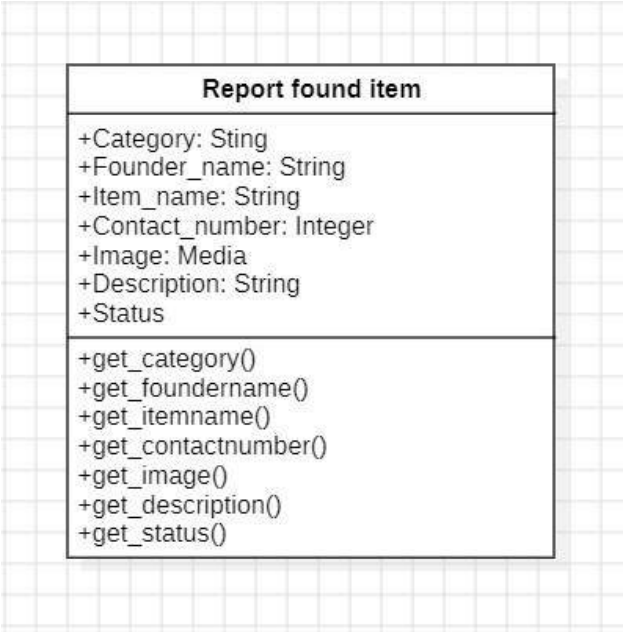


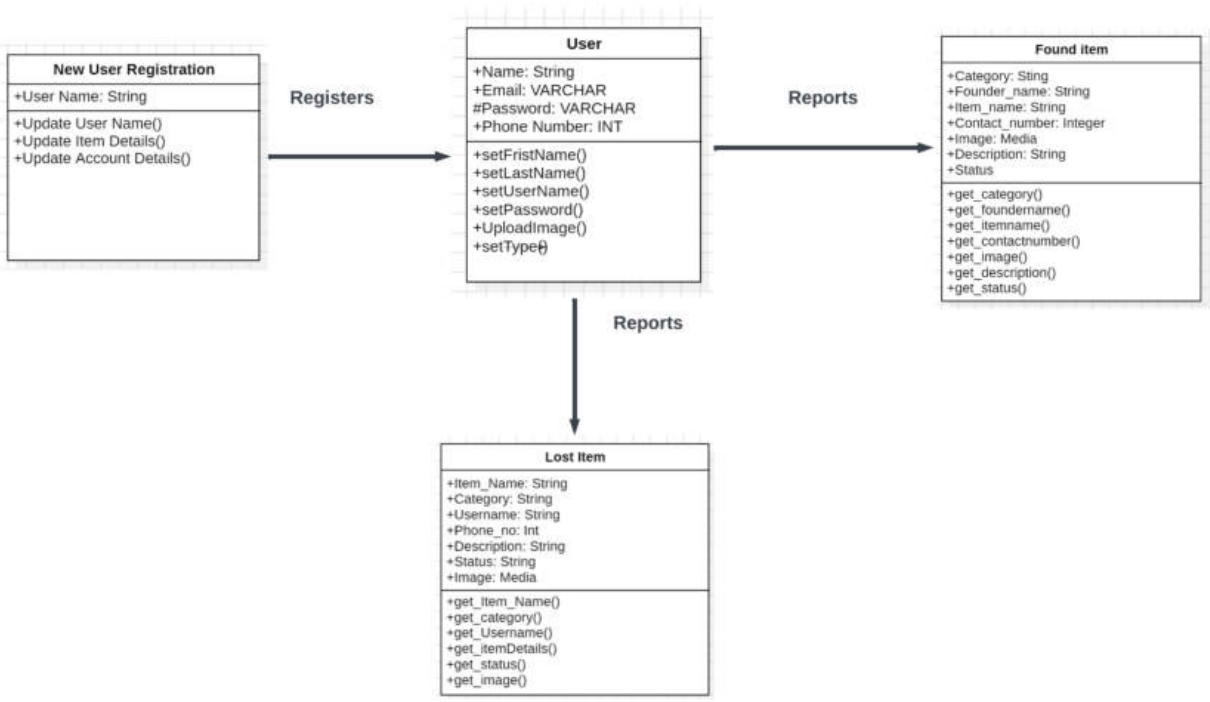
Class Diagram:



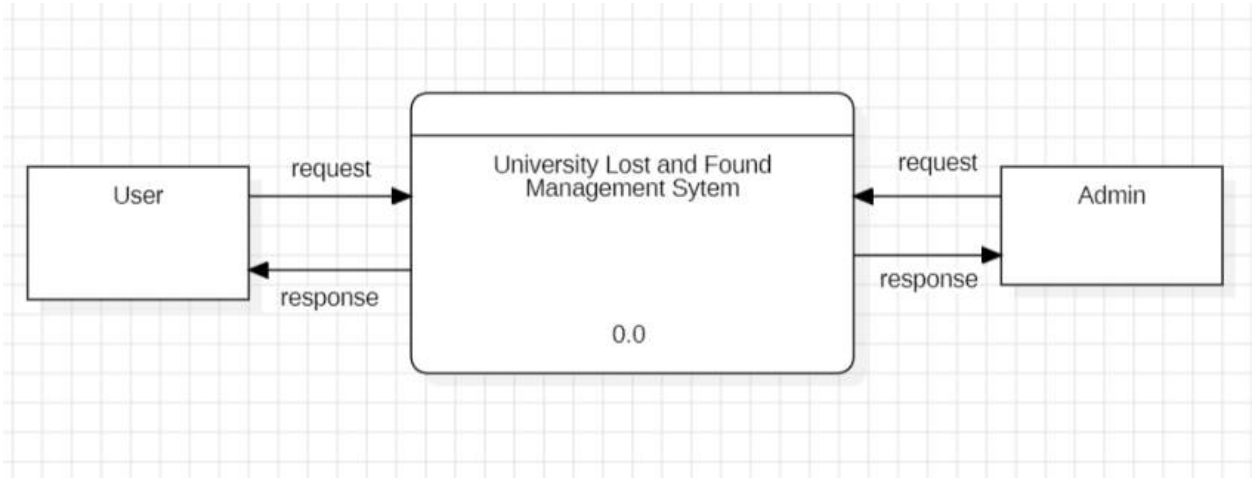




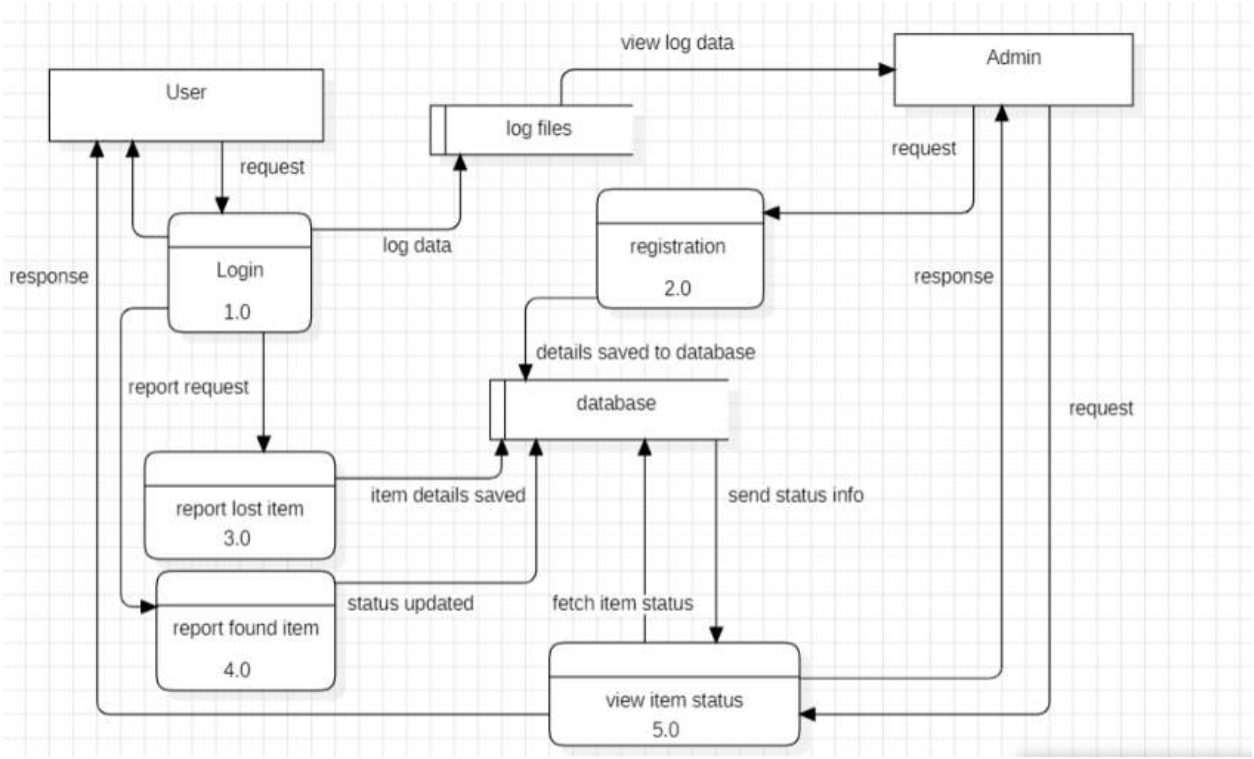



New User Registration
+User Name: String
+Update User Name()
+Update Item Details()
+Update Account Details()
Registers
User
+Name: String
+Email: VARCHAR
#Password: VARCHAR
+Phone Number: INT
+setFristName()
+setLastName()
+setUserName()
+setPassword()
+UploadImage()
+setType()
Reports
Found item
+Category: Sting
+Founder_name: String
+Item_name: String
+Contact_number: Integer
+Image: Media
+Description: String
+Status
+get_category()
+get_foundername()
+get_itemname()
+get_contactnumber()
+get_image()
+get_description()
+get_status()
Reports
Lost Item
+Item_Name: String
+Category: String
+Username: String
+Phone_no: Int
+Description: String
+Status: String
+Image: Media
+get_Item_Name()
+get_category()
+get_Username()
+get_itemDetails()
+get_status()
+get_image()

3. DFD - Level 0


User
request
response
University Lost and Found Management Sytem
0.0
request
response
Admin

DFD - Level 1



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.