Mini project report on

Lost and Found Management System

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE21CS352B – Object Oriented Analysis and Design using Java

Submitted by:

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

Under the guidance of

Dr. Mannar Mannan J

Associate Professor

PES University

JAN - MAY 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Lost and Found Management System

is a bonafide work carried out by

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

In partial fulfillment for the completion of sixth semester OOADJ Project (UE21CS352B) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period JAN. 2024 – MAY 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 6th semester academic requirements in respect of project work.

Signature
Dr. Mannar Mannan J

Associate professor

DECLARATION

We hereby declare that the OOADJ project entitled **Lost and Found Management System** has been carried out by us under the guidance of **Dr. Mannar Mannan J, Associate Professor** and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester JAN-MAY 2024.

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

ACKNOWLEDGEMENT

We would like to express our gratitude to **Dr. Mannar Mannan J**, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this OOADJ project.

We take this opportunity to thank Dr. Sandesh B J, Professor, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to us various opportunities and enlightenment every step of the way. Finally, this OOADJ Project could not have been completed without the continual support and encouragement we have received from our family and friends.

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

TABLE OF CONTENTS

1. Problem Statement	6
2. Models (Use Case, Class Models, State & Activity)	7
2.1 Overall View of the System	7
2.2 User Case	8
2.3 Class Models	9
2.4 Activity Diagram	10
2.5 State Diagram	11
3. Architecture Patterns	12
4. Design Principles	12
5. Design Patterns	14
6. GitHub Link	15
7. Individual Contributions	15
8. Screenshots	17

Problem Statement

In today's bustling environments, the loss of personal belongings is a common occurrence, often leading to frustration and inconvenience for individuals. Lost and Found Connect emerges as a sophisticated solution to this ubiquitous problem, aiming to transform the way lost items are managed across various settings such as schools, offices, public spaces, and events.

At the core of Lost and Found Connect lies a user-centric approach, beginning with a seamless registration and authentication process. Through secure user accounts, individuals can access the platform with confidence, knowing that their personal information and reported items are safeguarded.

Once registered, users can easily report lost items, providing detailed descriptions including item name along with the reporter details. This reporting mechanism ensures the groundwork for efficient item recovery.

Lost and Found Connect employs an advanced matching algorithm, leveraging sophisticated criteria such as descriptions, categories, and timestamps to pair reported lost items with those found by others. This intelligent system increases the likelihood of successful item reunification, minimizing the time and effort required for both users and administrators.

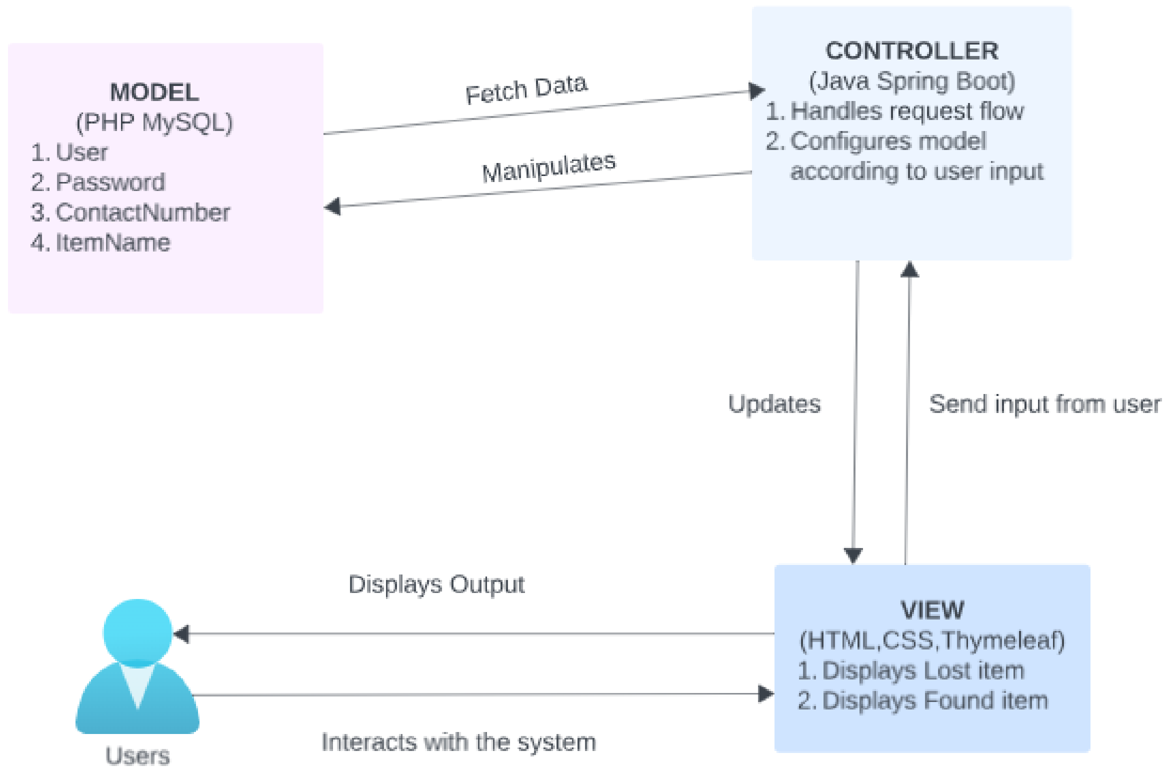
To further enhance accessibility, Lost and Found Connect features comprehensive search and retrieval functionalities. Users can effortlessly navigate through reported lost items, utilizing filters such as where the item reported was lost or found item. This intuitive interface empowers individuals to take control of their search efforts, fostering a sense of empowerment and engagement.

Communication plays a pivotal role in the item recovery process, and Lost and Found Connect excels in this aspect with its robust notification system. Users are promptly alerted when their reported lost items are found or when there are updates regarding the status of their submissions. This real-time communication streamlines the retrieval journey, keeping individuals informed and reassured throughout the process.

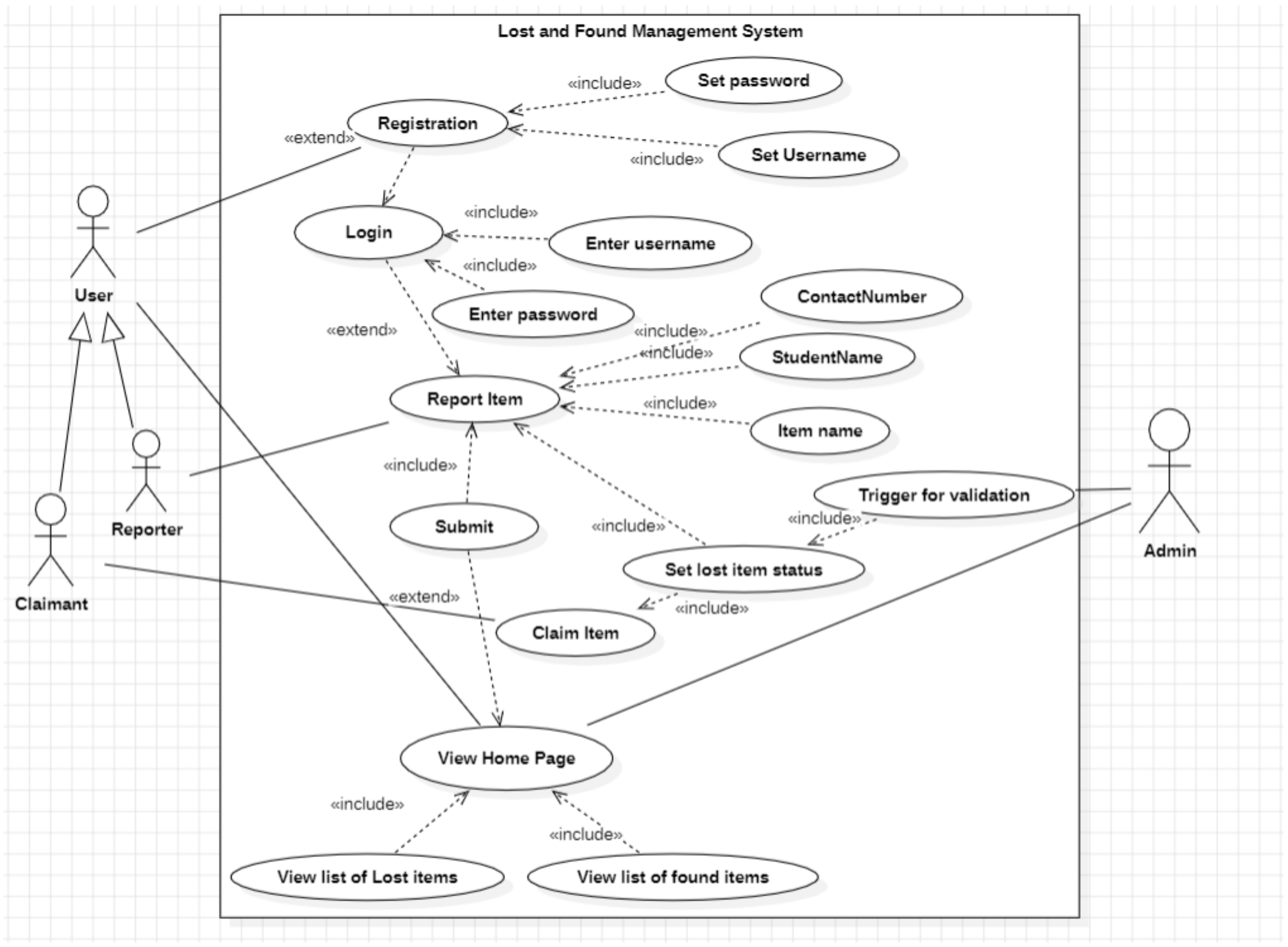
In conclusion, Lost and Found Connect represents a paradigm shift in lost item management, offering a comprehensive and user-friendly platform that simplifies the reporting, matching, and retrieval processes.

Models

OVERVIEW OF THE SYSTEM (MVC ARCHITECTURE)

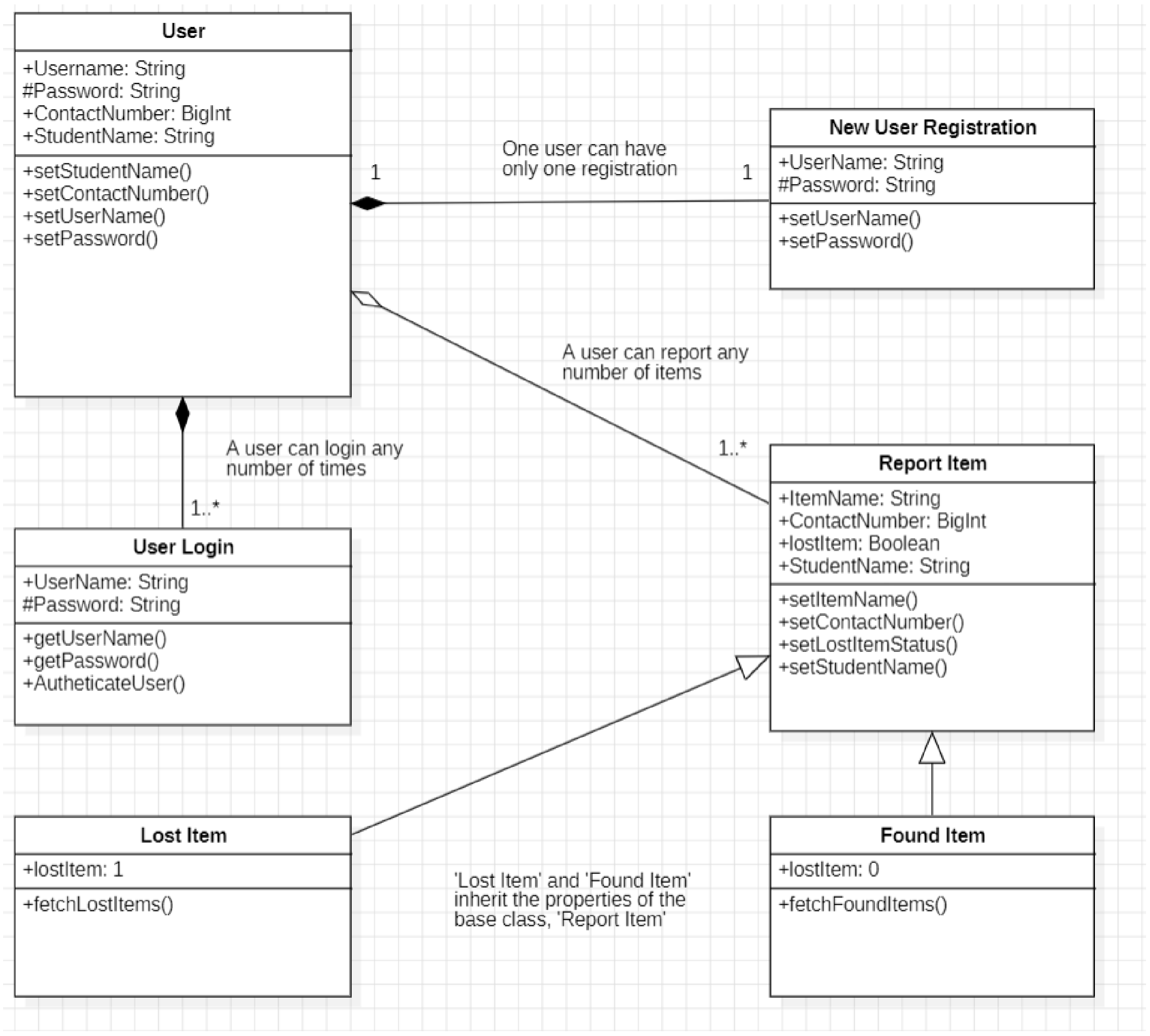


1. Use Case

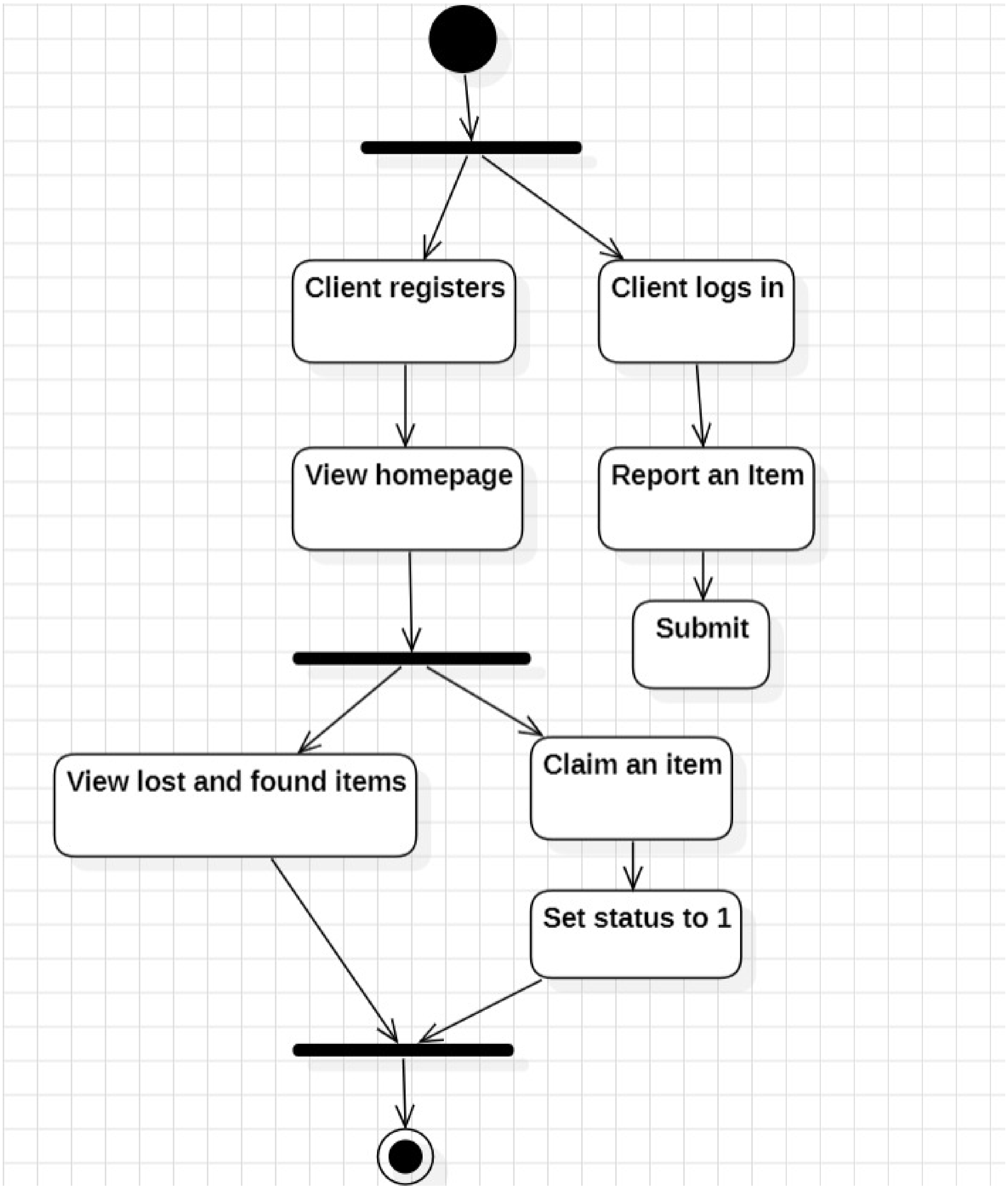


Lost and Found Management System
User
Reporter
Claimant
Admin
Registration
Set password
Set Username
Login
Enter username
Enter password
Report Item
ContactNumber
StudentName
Item name
Submit
Trigger for validation
Set lost item status
Claim Item
View Home Page
View list of Lost items
View list of found items
«include»
«extend»

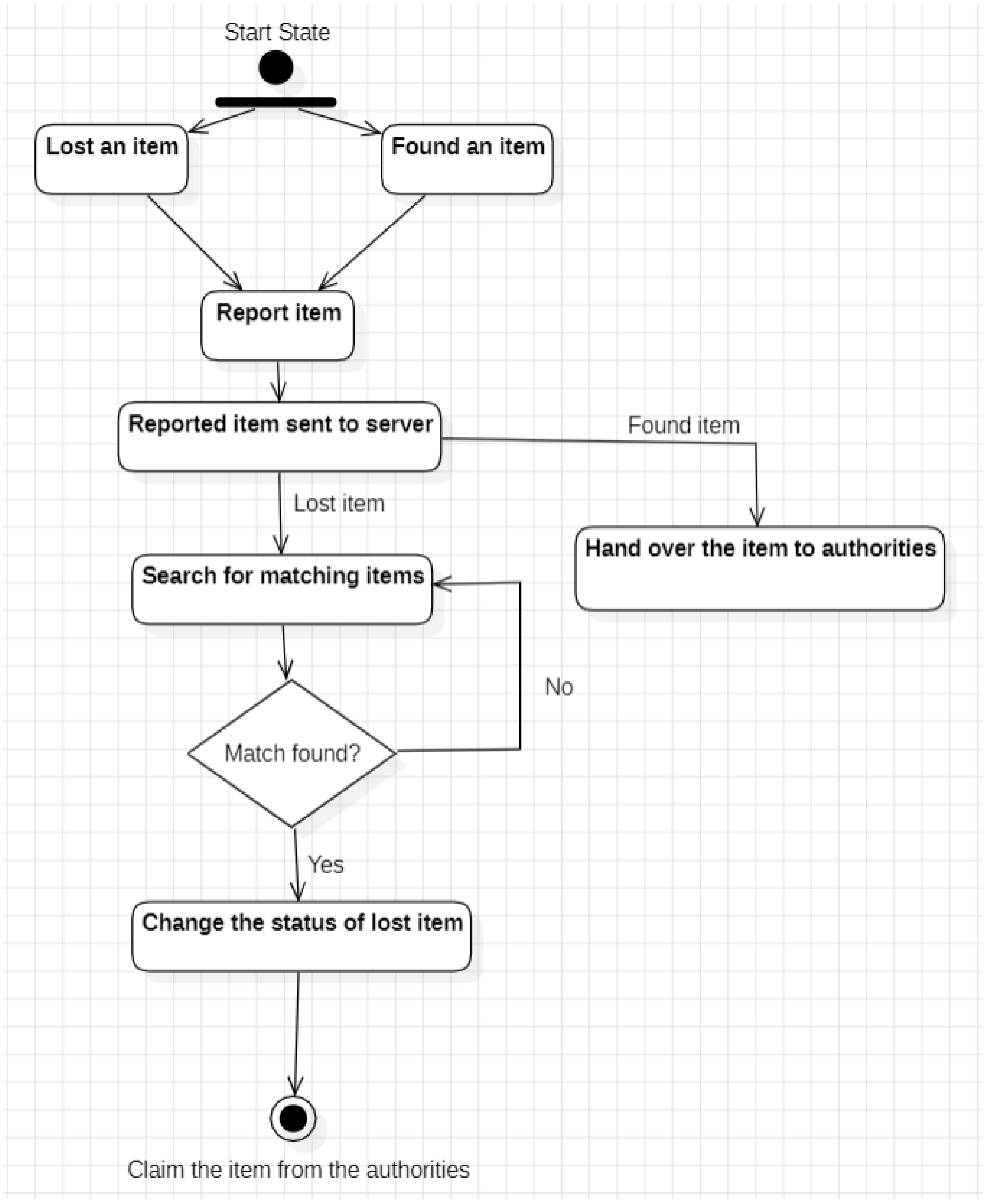
2. Class Models



3. Activity Diagram



4. State Diagram



Start State
Lost an item
Found an item
Report item
Reported item sent to server
Found item
Lost item
Hand over the item to authorities
Search for matching items
No
Match found?
Yes
Change the status of lost item
Claim the item from the authorities

Architecture Pattern:

Client-Server Architecture

Web browser acts as a Client :Where User interacts with the system through web interface(HTML and CSS files). Web browser sends request to server for functionalities .
SpringBoot Application acts as Server: Handles the user requests and interacts with the database. Controller (StudentController) handles incoming requests, processes and interacts with the repository to access and update data. Repositories (StudentRepository) provide an abstraction layer for interacting with the database (MySQL) for their functionalities.

The communication between the client (web browser) and server (Spring Boot application) happens through HTTP requests and responses.

The controllers uses Spring MVC annotations to map URLs to handler methods, process user input, and generate appropriate responses

Design Principles:

1. **Controller (GRASP)**
   One of the classes make use of this design principle in:
   UserRegistration: It acts as a controller by handling user interactions and coordinating with the database**.**

2. **Single Responsibility Principle SRP (GRASP)**:
   One of the classes make use of this design principle is:
   NotNullItemQuery: Class is only responsible for defining the query to retrieve not null items from the database.

2. **Dependency Inversion Principle (DIP) (SOLID):**
   One of the classes making use of this design principle
   The ListItemsQuery class depends on the DatabaseQuery abstract class and adhering to DIP. Flexibility to make changes to the behavior of the query is provided without modifying ListItemsQuery class itself.

3. **High Cohesion (GRASP):**
   Under LostItemZeroStudentFactory we see that there are two methods getStudents and displayStudents that are related to fetching students from the database and displaying them and they work together to serve this purpose and operate on a common theme.

4. **Low Coupling (GRASP):**
   Abstractions provided by JDBC are used rather than going for the concrete implementation and this supports ease of maintenance and promotes modularity.

SOLID Principles:

1. Singleton Design Pattern:
   a. UserRegistration
2. Template Method Design Pattern:
   a. DatabaseQuery
   b. ListItemsQuery
3. Adapter Design Pattern:
   a. NotNullItemsAdapter
   b. DataQuery
   c. NotNullItemsQuery
4. Factory Method Design Pattern:
   a. LostItemZeroStudentFactory
   b. LostItemOneStudentFactory

GRASP Principles:

1. Creator:
   a. LostItemZeroStudentFactory
   b. LostItemOneStudentFactory
2. High Cohesion:
   a. UserRegistration
   b. DatabaseQuery
   c. ListItemsQuery

    d. LostItemZeroStudentFactory
    e. LostItemOneStudentFactory
3. Low Coupling:
    a. UserRegistration
    b. DatabaseQuery
    c. ListItemsQuery
    d. LostItemZeroStudentFactory
    e. LostItemOneStudentFactory
4. Controller:
    a. UserRegistration
    b. LostItemZeroStudentFactory
    c. LostItemOneStudentFactory
    d.
5. Indirection:
    a. UserRegistration
    b. DatabaseQuery
    c. ListItemsQuery
    d. LostItemZeroStudentFactory
    e. LostItemOneStudentFactory

Design Patterns

The design patterns being used in this project are:

1. **Singleton pattern (Creational):**
   The Singleton method Design Pattern is used where there must be exactly one instance of a class and it must be accessible to clients from a well-known access point. We have used the singleton design pattern for user registration functionality. Everytime the user registration class is called, the same instance is returned through the getInstance() method instead of the default constructor returning a new instance at a different memory location every time.

2. **Template pattern (Behavioural):**

   The template pattern defines the skeleton of an algorithm or operations in a superclass (often abstract) and leaves the details to be implemented by the child classes. We have used it to print the list of found items. It defines a skeleton algorithm in a superclass ‘DatabaseQuery’ but allows subclasses ListItemsQuery to provide specific implementations for certain steps ‘getQuery’ and ‘processRow'. This way, the overall structure of the algorithm is maintained in the superclass while customization happens in subclasses.

3. **Adapter pattern (Structural):**

   The Adapter design pattern is a structural pattern that allows the interface of an existing class to be used as another interface. It acts as a bridge between two incompatible interfaces, making them work together. It adapts the behavior of a database query class 'Database Query' into a more generic interface 'ItemQuery' using an adapter class 'NotNullItemsAdapter'. We have used it to print the list of lost items. This allows the adapter to execute the original query through the adapted interface.

4. **Factory pattern (creational):**

   The Factory Method Design Pattern is a creational design pattern that provides an interface for creating objects in a superclass, allowing subclasses to alter the type of objects that will be created. Both factory classes perform database operations to fetch students based on the lost_item value. They utilize the Student class to represent student data and JDBC for database interactions. Each factory class contains a main method for testing and printing the retrieved student information.

Github Link:
https://github.com/ShruC24/OOADJ_LostAndFoundManagementSystem_501_514_536_537

Individual contributions

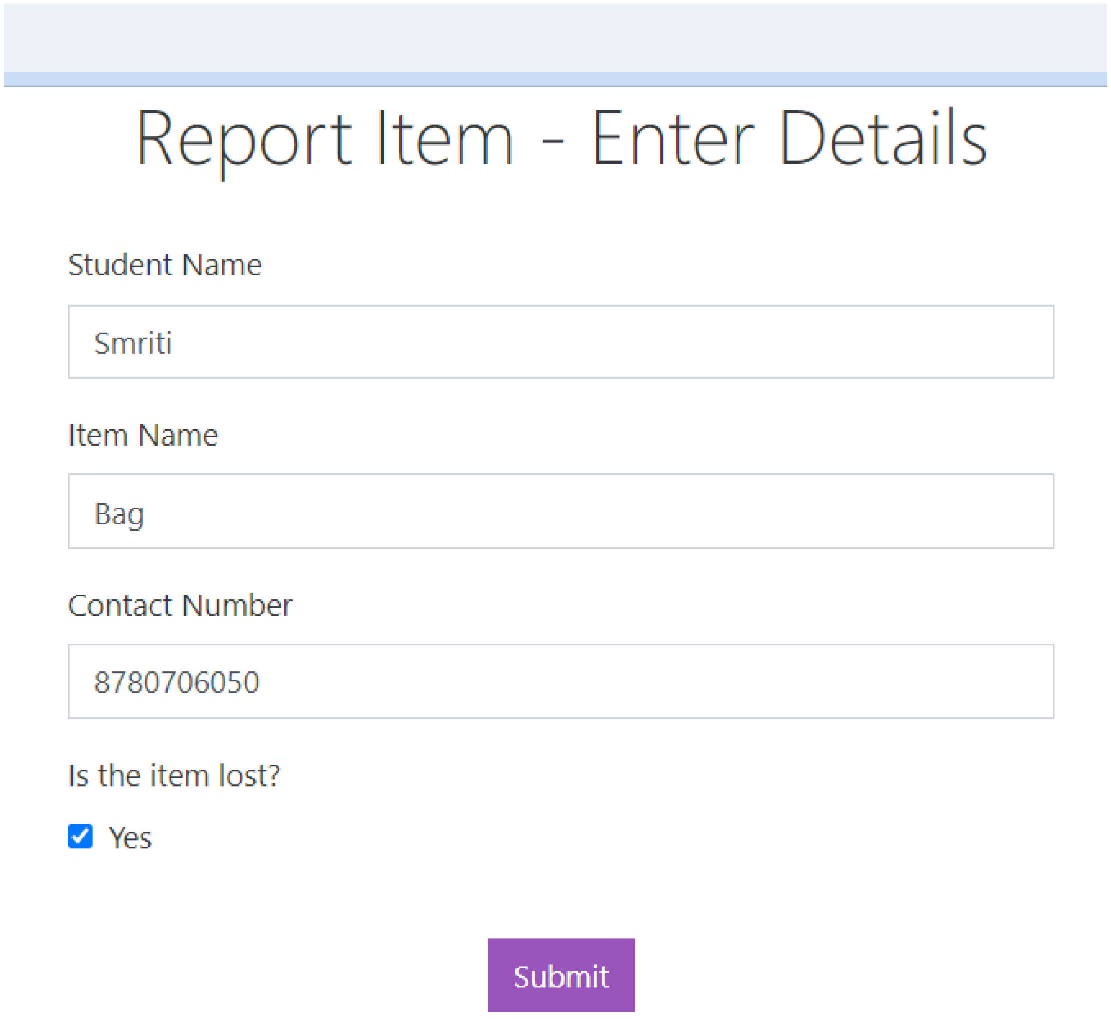
1. **SHREYA JOSHI** (PES2UG21CS501):
   - Design Patterns
   - Design Principles
   - Crud Operations
   - Documentation and Models
   - Testing
2. **SHRUTI C** (PES2UG21CS514):
   - Crud Operations
   - Documentation
   - Design Principles
   - Models
   - Testing
3. **SPOORTHI SHIVAPRASAD** (PES2UG21CS536):
   - Design Patterns

- Design Principles
- Crud Operation
- Documentation and Models
- Testing

4. **SRAGVI ANIL SHETTY** (PES2UG21CS537):
    - Design Patterns
    - Documentation
    - Crud Operation
    - Models
    - Testing

Screenshots

1. Report an item



Report Item - Enter Details

Student Name

Smriti

Item Name

Bag

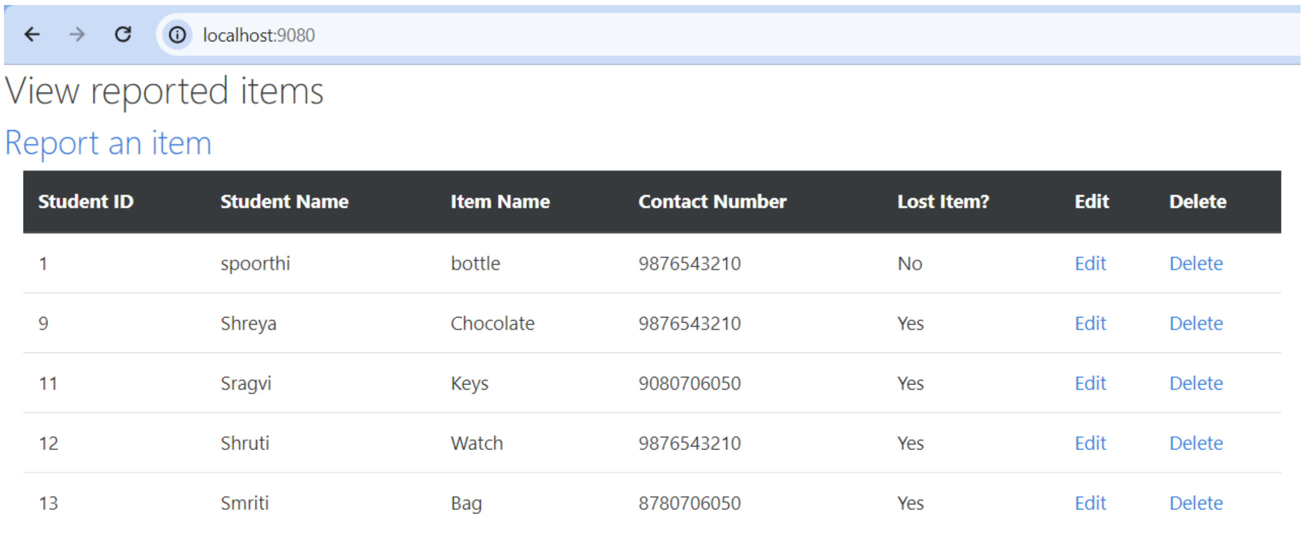
Contact Number

8780706050

Is the item lost?

[✓ Yes]

2. View reported items

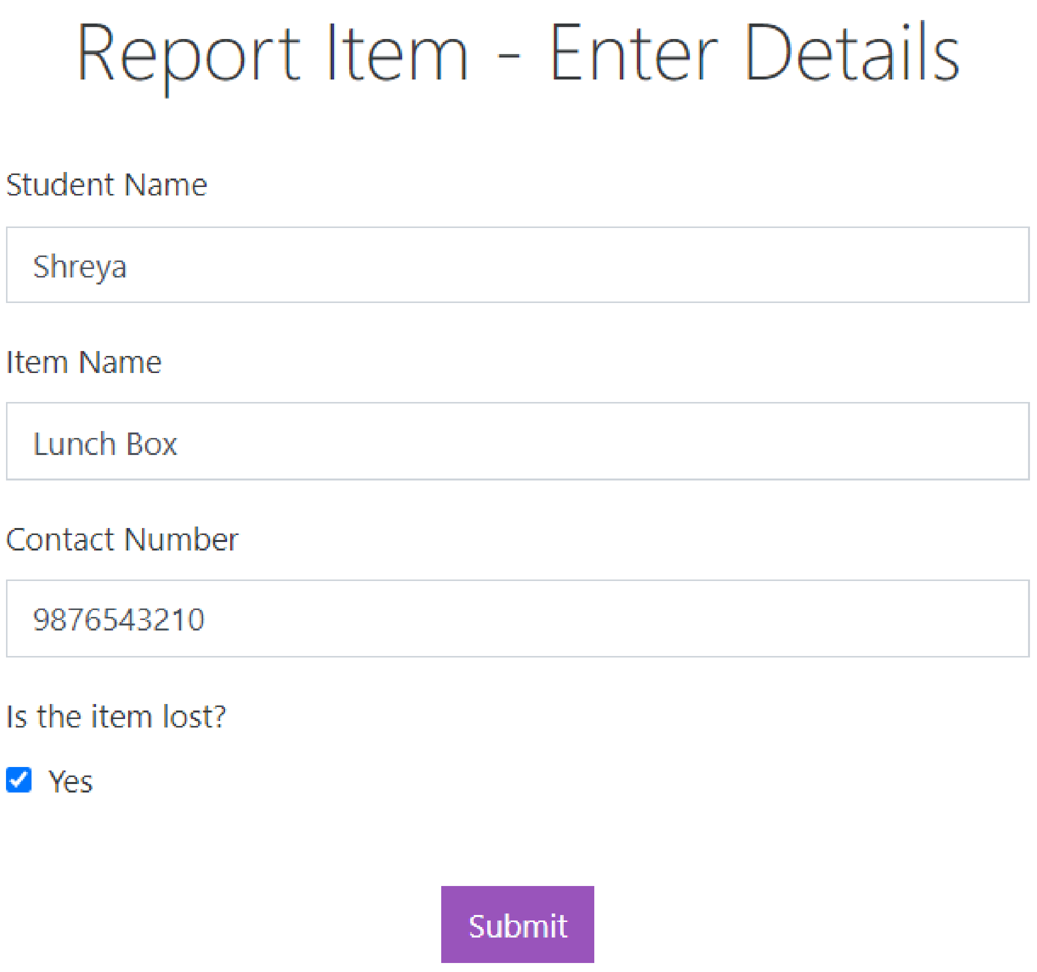

localhost:9080

View reported items

Report an item

Student ID	Student Name	Item Name	Contact Number	Lost Item?	Edit	Delete
1	spoorthi	bottle	9876543210	No	Edit	Delete
9	Shreya	Chocolate	9876543210	Yes	Edit	Delete
11	Sragvi	Keys	9080706050	Yes	Edit	Delete
12	Shruti	Watch	9876543210	Yes	Edit	Delete
13	Smriti	Bag	8780706050	Yes	Edit	Delete

3. Update reported item details



Report Item - Enter Details

Student Name

Shreya

Item Name

Lunch Box

Contact Number

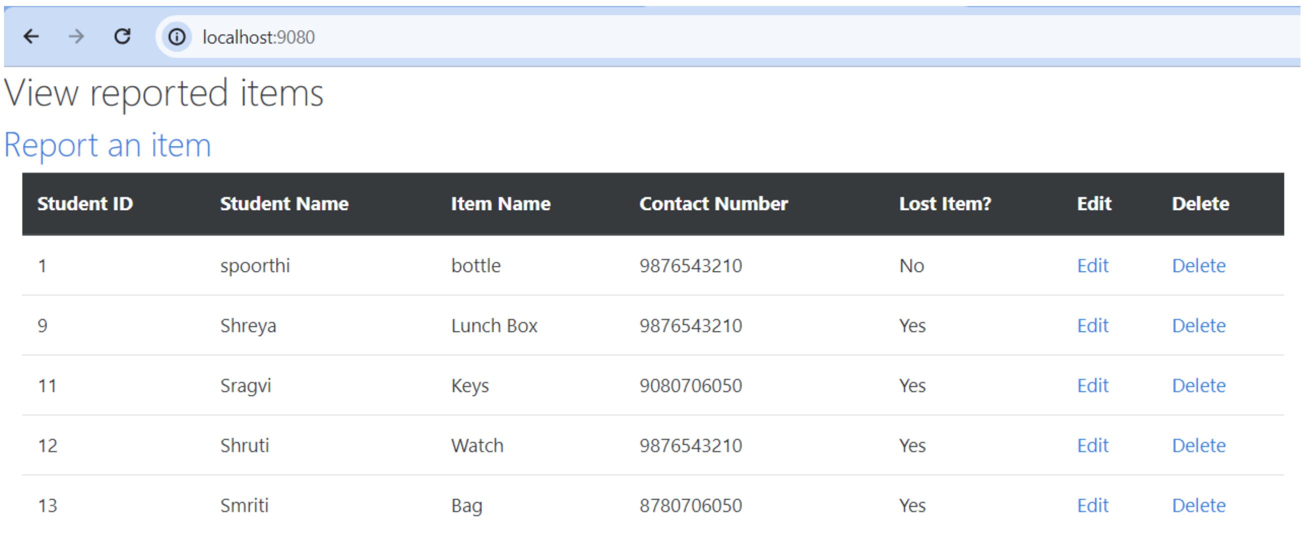
9876543210

Is the item lost?

[✓ Yes]

Submit

4. View updated details



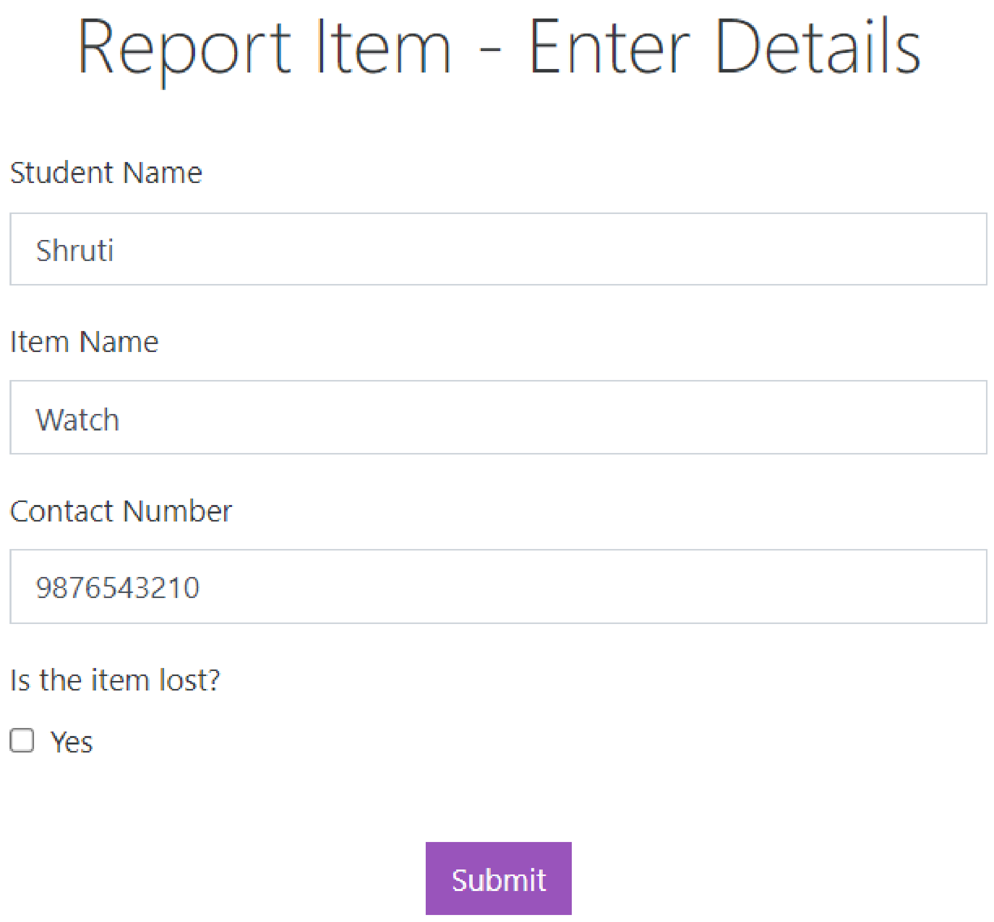
localhost:9080

View reported items

Report an item

Student ID	Student Name	Item Name	Contact Number	Lost Item?	Edit	Delete
1	spoorthi	bottle	9876543210	No	Edit	Delete
9	Shreya	Lunch Box	9876543210	Yes	Edit	Delete
11	Sragvi	Keys	9080706050	Yes	Edit	Delete
12	Shruti	Watch	9876543210	Yes	Edit	Delete
13	Smriti	Bag	8780706050	Yes	Edit	Delete

5. **Change item status to found**



Report Item - Enter Details

Student Name

Shruti

Item Name

Watch

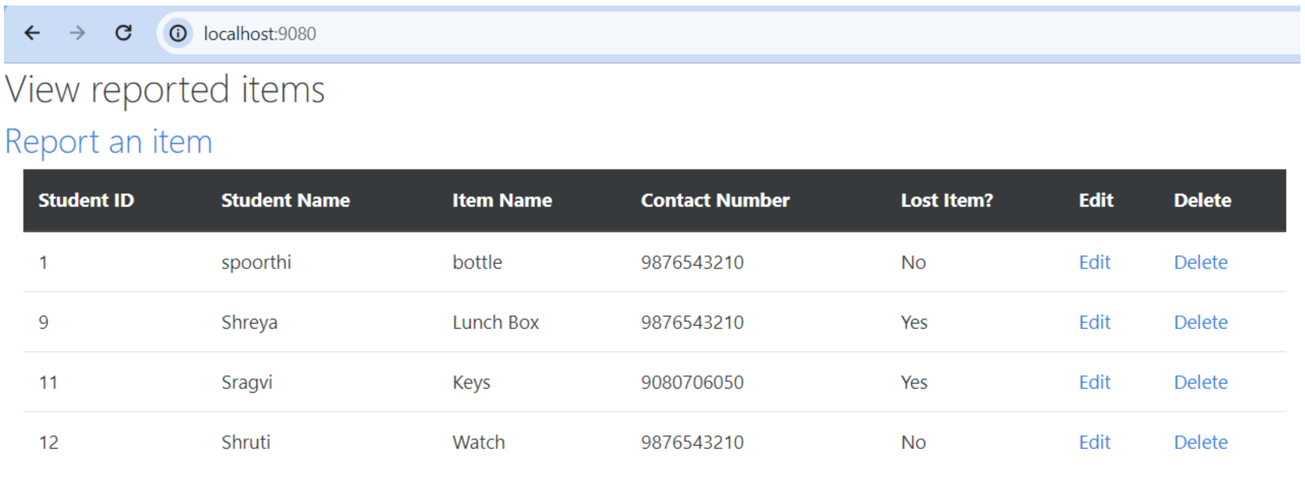
Contact Number

9876543210

Is the item lost?

☐ Yes

6. **Delete an item record**



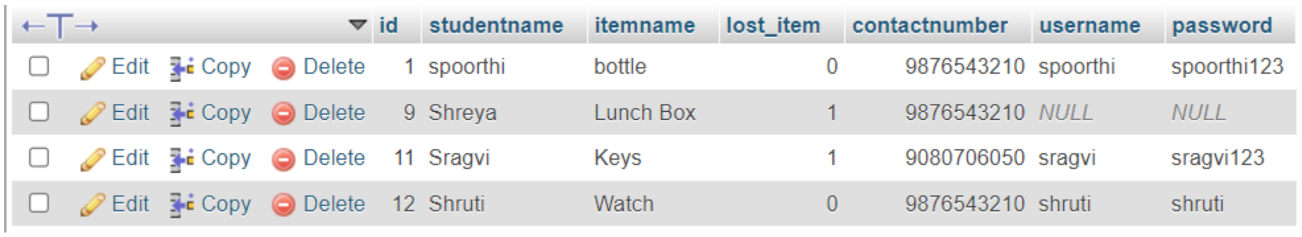
localhost:9080

View reported items

Report an item

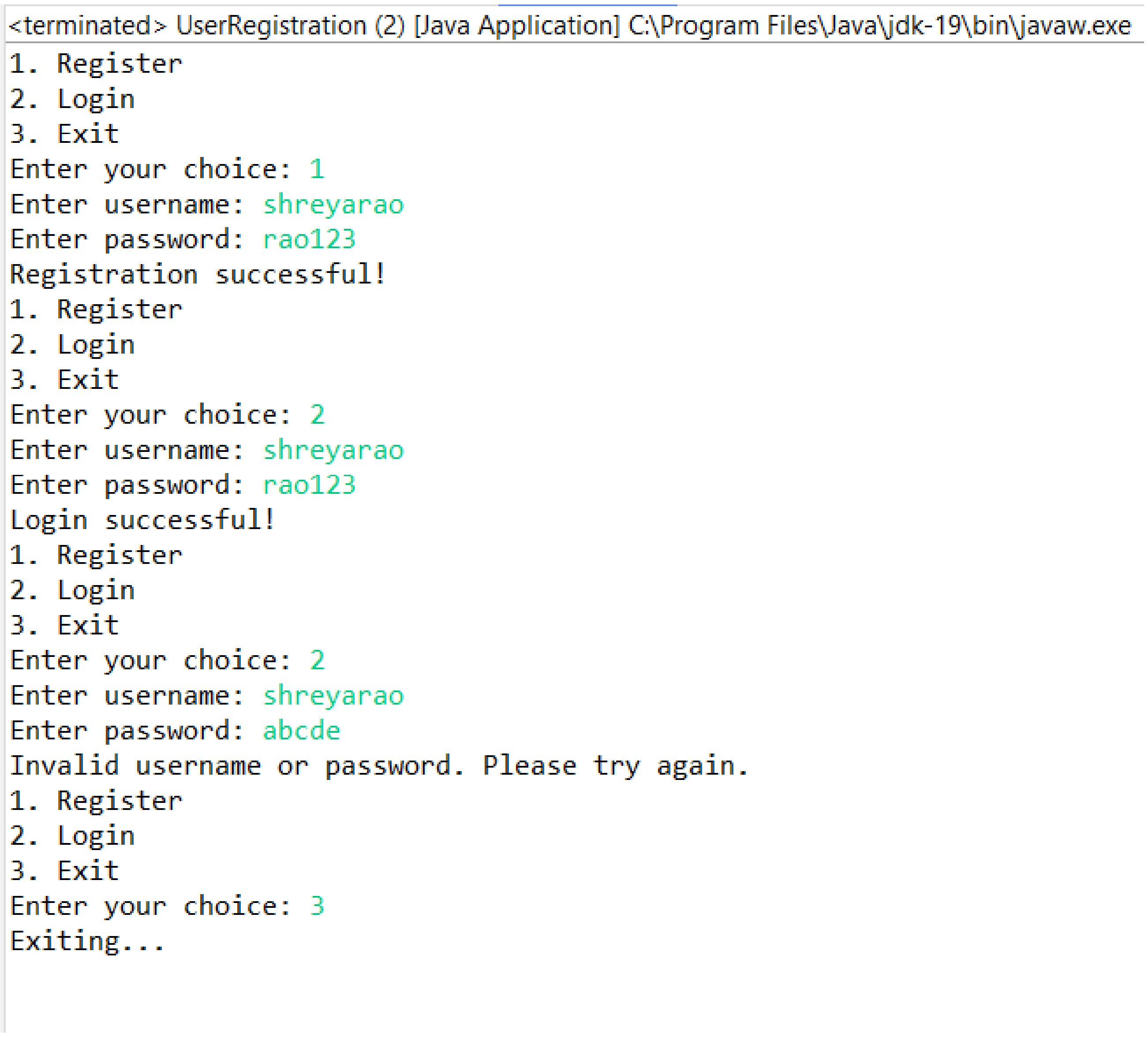
Student ID	Student Name	Item Name	Contact Number	Lost Item?	Edit	Delete
1	spoorthi	bottle	9876543210	No	Edit	Delete
9	Shreya	Lunch Box	9876543210	Yes	Edit	Delete
11	Sragvi	Keys	9080706050	Yes	Edit	Delete
12	Shruti	Watch	9876543210	No	Edit	Delete

7. Database



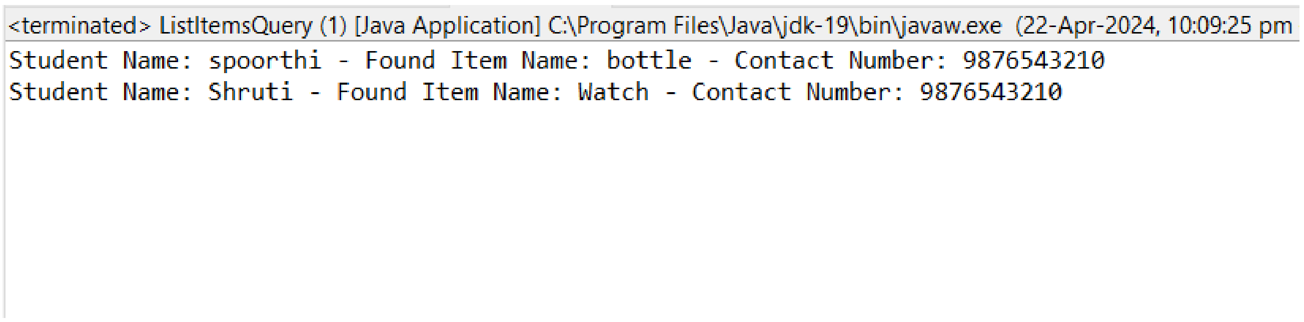
				id	studentname	itemname	lost_item	contactnumber	username	password
☐	Edit	Copy	Delete	1	spoorthi	bottle	0	9876543210	spoorthi	spoorthi123
☐	Edit	Copy	Delete	9	Shreya	Lunch Box	1	9876543210	NULL	NULL
☐	Edit	Copy	Delete	11	Sragvi	Keys	1	9080706050	sragvi	sragvi123
☐	Edit	Copy	Delete	12	Shruti	Watch	0	9876543210	shruti	shruti

8. Singleton design pattern



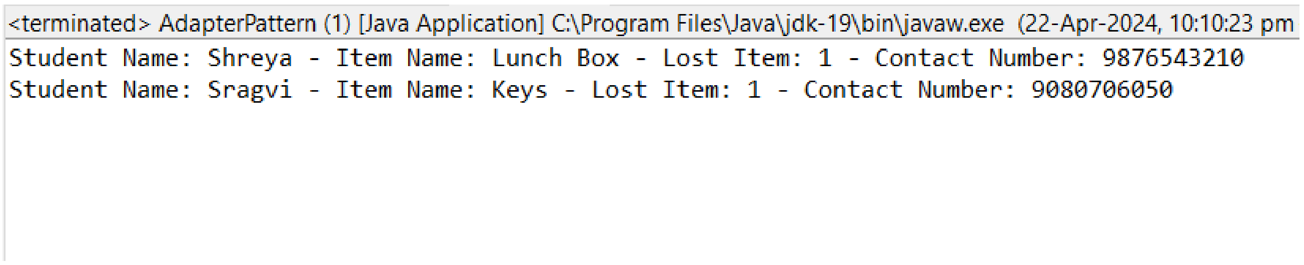
```
<terminated> UserRegistration (2) [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe
1. Register
2. Login
3. Exit
Enter your choice: 1
Enter username: shreyarao
Enter password: rao123
Registration successful!
1. Register
2. Login
3. Exit
Enter your choice: 2
Enter username: shreyarao
Enter password: rao123
Login successful!
1. Register
2. Login
3. Exit
Enter your choice: 2
Enter username: shreyarao
Enter password: abcde
Invalid username or password. Please try again.
1. Register
2. Login
3. Exit
Enter your choice: 3
Exiting...
```

9. Template design pattern



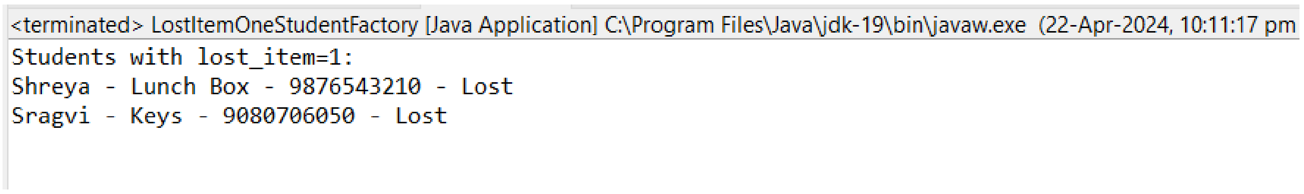
```
<terminated> ListItemsQuery (1) [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (22-Apr-2024, 10:09:25 pm
Student Name: spoorthi - Found Item Name: bottle - Contact Number: 9876543210
Student Name: Shruti - Found Item Name: Watch - Contact Number: 9876543210
```

10. Adapter design pattern

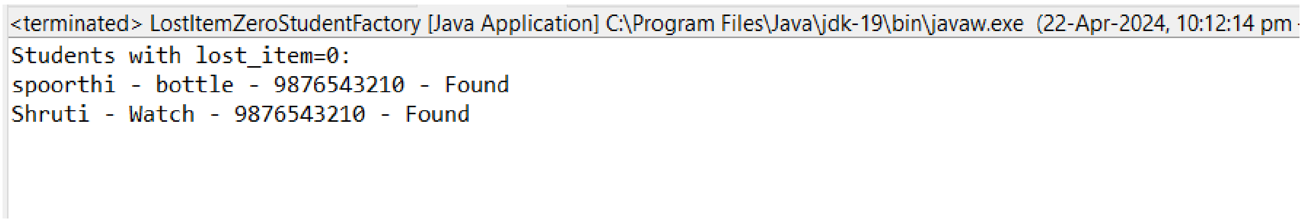


```
<terminated> AdapterPattern (1) [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (22-Apr-2024, 10:10:23 pm
Student Name: Shreya - Item Name: Lunch Box - Lost Item: 1 - Contact Number: 9876543210
Student Name: Sragvi - Item Name: Keys - Lost Item: 1 - Contact Number: 9080706050
```

11. Factory design pattern



```
<terminated> LostItemOneStudentFactory [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (22-Apr-2024, 10:11:17 pm
Students with lost_item=1:
Shreya - Lunch Box - 9876543210 - Lost
Sragvi - Keys - 9080706050 - Lost
```



```
<terminated> LostItemZeroStudentFactory [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (22-Apr-2024, 10:12:14 pm
Students with lost_item=0:
spoorthi - bottle - 9876543210 - Found
Shruti - Watch - 9876543210 - Found
```