PES UNIVERSITY
Department of Computer Science & Engineering

Object Oriented Analysis and Design using Java (UE20CS352)

Title of Project: Library ManagementSystem

Team Members:

1. Vinti Agrawal -PES2UG20CS385

2. Vishwa Mehta -PES2UG20CS389

3. Vismaya R -PES2UG20CS391

Submitted to,

DR. Kamatchi Priya LAssociate Professor

Problem Statement:

Our application involves designing and implementing a system that can simulate various ATM transactions using a web interface. The web application should allow users to access an ATM machine through a web interface and perform various transactions such as cash withdrawal, balance inquiry, and fund transfer. It is built using spring boot MVC architecture.

Synopsis:

The application has the following features:

- Create, read, and update user details from a database.
- Display page to display list of users associated with that bank to admin.
- Users can withdraw money using their account number and pin and the same is updated on a database.
- A user interface which acts as a home page containing various option like generate pin, change pin , display profile and Make Transactions which includes Transfer and withdraw.

major features:

login using card number and pin

withdrawal

transfer funds

minor features:

pin generation/change

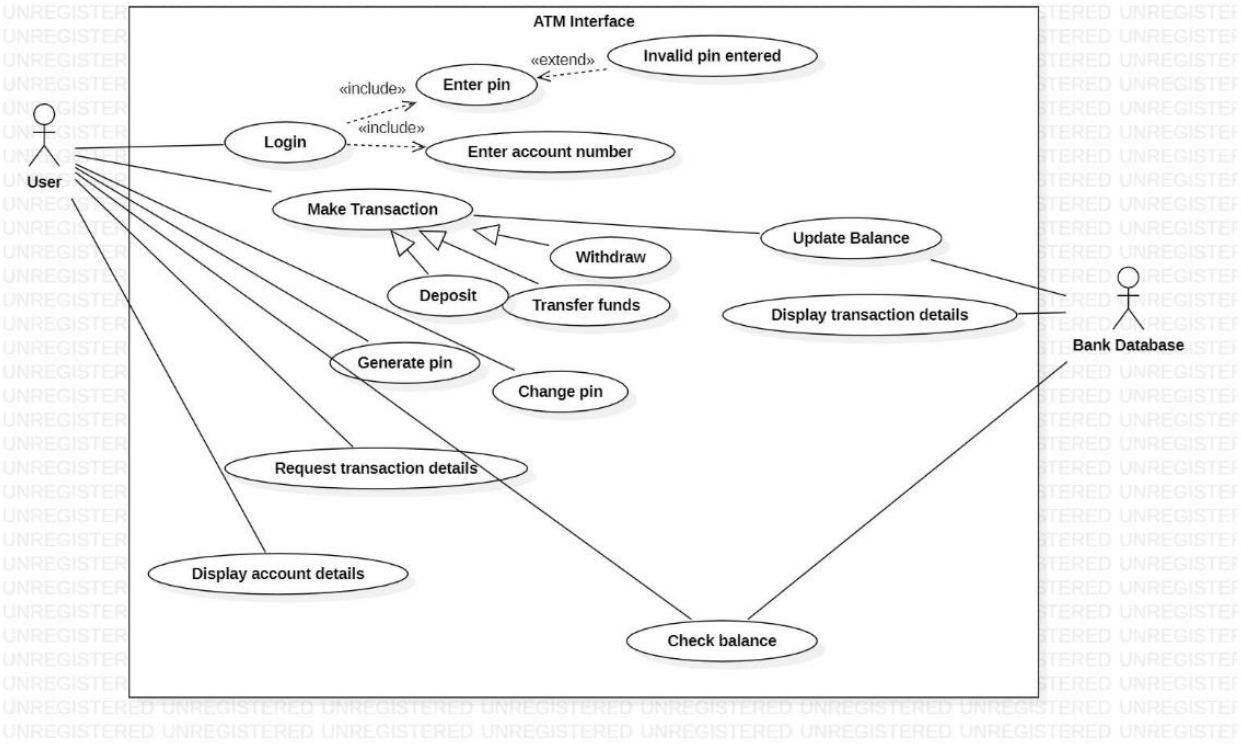
transaction records

account details

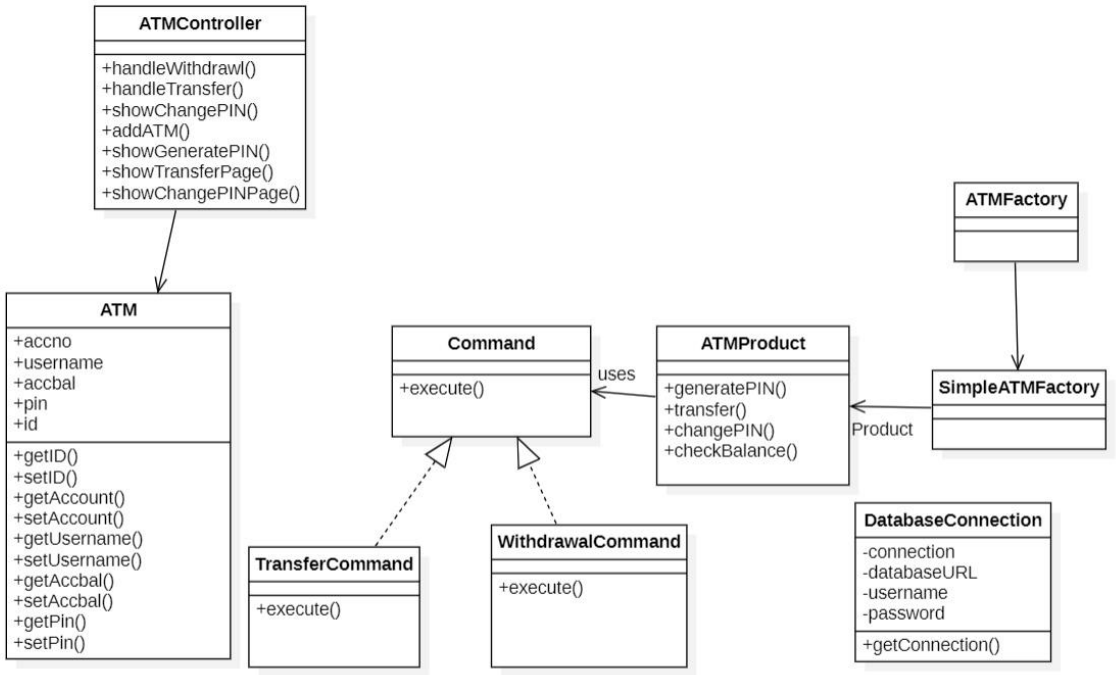
Overall, the goal of this project is to build a reliable, user-friendly, and secure web application that simulates an ATM machine and provides a seamless user experience.

Models:

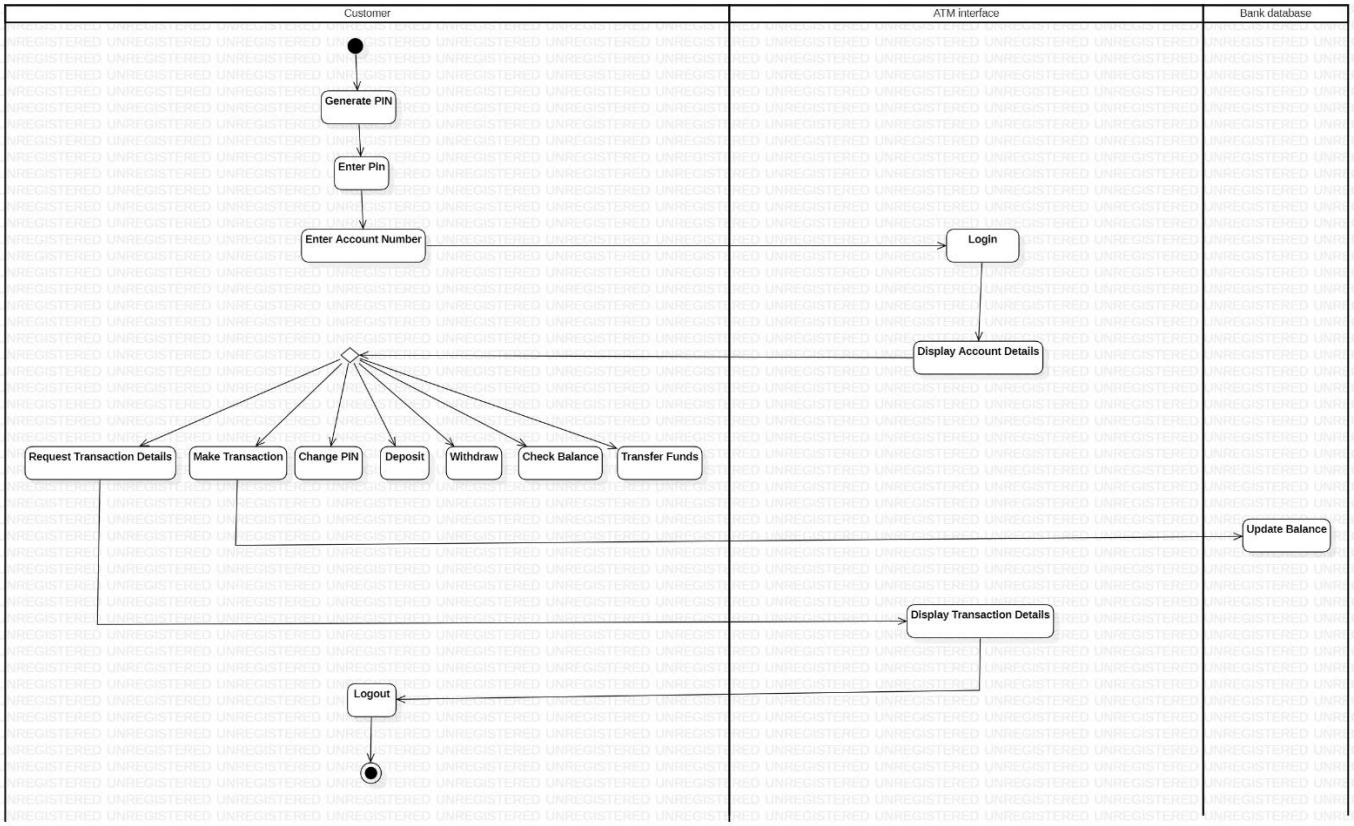
(i) Use Case:



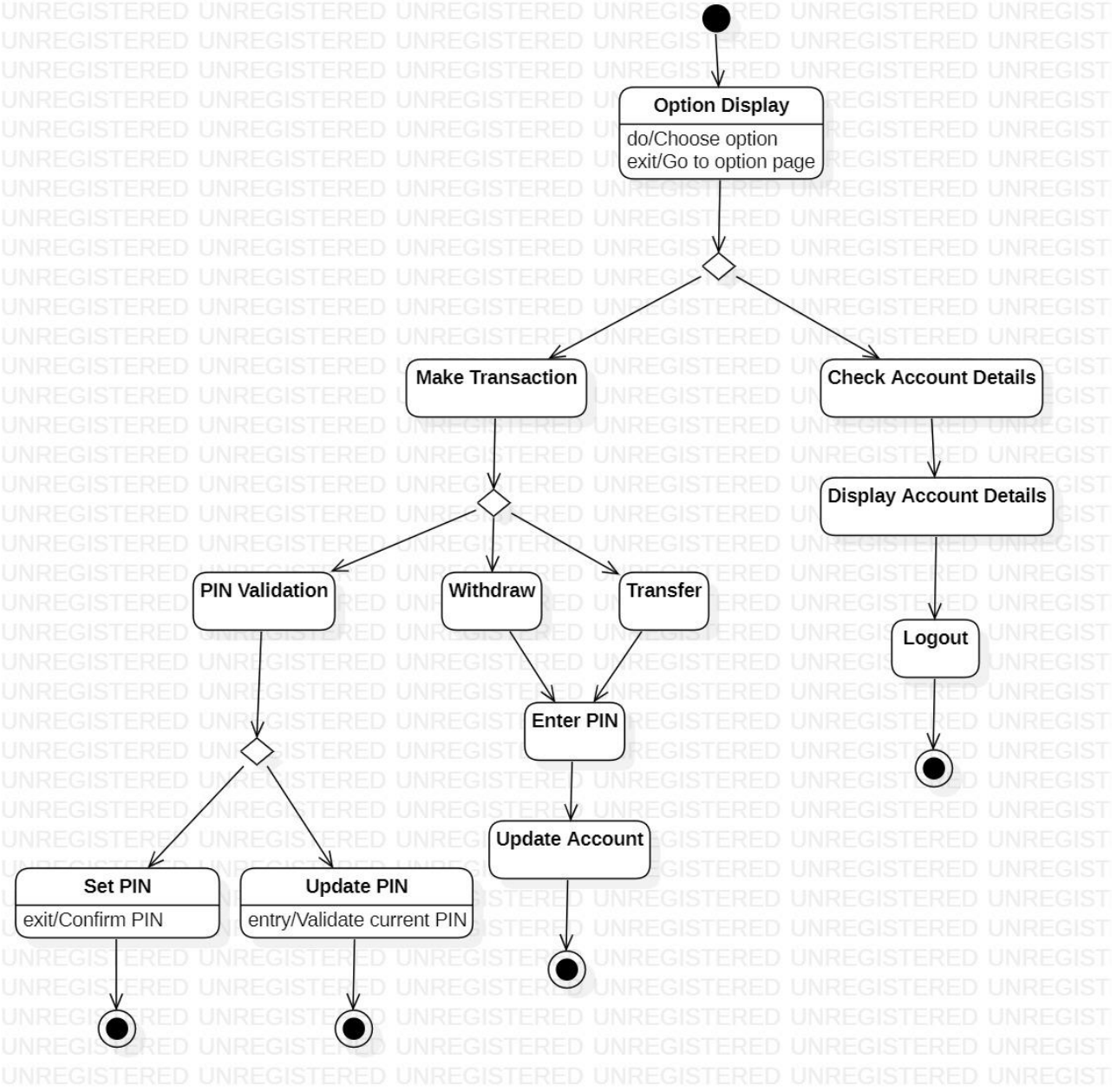
(ii) Class Diagram:



(iii) Activity Diagram



(iv)State Diagram



Architecture Pattern:

The application uses the MVC pattern. In MVC pattern, the Model represents the business logic and data, the View represents the presentation layer and the Controller handles the user requests and communicates with both Model and View to complete the request-response cycle.

In the application, the Model classes are the entries for user details like getting and setting account number, pin ,balance and username where the business logic is implemented. The View layer consists of Thyme leaf templates that handle the presentation of data which include home page, withdrawal page , display page and all other functionalities . The Controller is the Spring Controller that receives user requests, communicates with the Model layer to retrieve, and modify data, and then passes that data to the View layer for rendering.

Design Patterns:

Factory Method Design pattern: It is a Creational Design pattern. First, we define an interface InventoryFactory that declares a factory method createInventory().Next, we can implement this interface with a concrete factory SimpleInventoryFactory that creates different types of inventory objects based on the user's input. Finally, we can modify the InventoryController class to use this factory to create new inventory objects when the user adds inventory

Command Design pattern: It is a behavioral Design Pattern. In this implementation, we define an interface Command that defines a execute () method. We then define a concrete implementation of the Command interface called Withdrawal Command, which takes an Inventory object, an amount to withdraw, and a PIN as constructor arguments. The Withdrawal Command class implements the execute ()

method, which performs the withdrawal logic. If the provided PIN exists in the database and the withdrawal amount is less than or equal to the account balance, the account balance is updated and saved to the database.

Singleton Design Pattern : The DatabaseConnection class is an example of the Singleton design pattern. The Singleton pattern ensures that only one instance of a class is created throughout the application and provides a global point of access to that instance.In this implementation, the DatabaseConnection class has a private static volatile Connection object, which is the single instance of the class. The getConnection() method is the factory method that returns the single instance of the Connection. It ensures that no two instance of same database is created. It will create it for the first time and later it will just use the same existing instance.

Design Principles:

Open/Closed Principle (OCP): We have interface that implements the functions which makes it open for implementation and closed for modifications thus following OCP Principle.

Single Responsibility Principle (SRP): Each class and method should have only one responsibility and reason to change. For example, the Inventory (Model Class) should only handle business logic related to products.

Liskov Substitution Principle (LSP): objects of a superclass should be replaceable with objects of its subclasses without breaking the application. In our application The jpaRepository is inherited by sub-classes without breaking the application for CRUD operation.

GitHub Link:

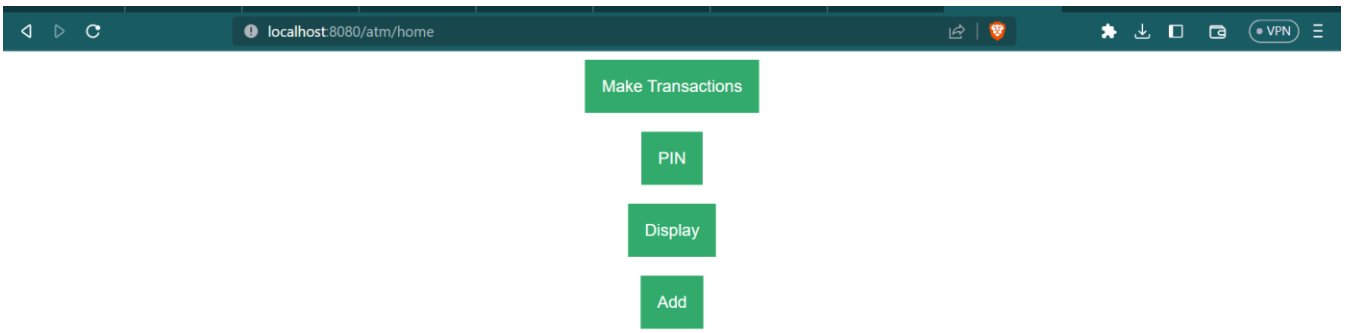
https://github.com/vintiAGRAWAL/OOAD_PROJECT.git

Individual Contributions of the Team Members:

1. Vinti Agrawal – Command Design Pattern, Update, Read, Display User
2. Vishwa Mehta – Factory Design Pattern, Create, Update Add User
3. Vismaya R – Singleton Design Pattern, Create, Update Generate Pin

Screenshot with populated values and Output shown:

Home Page:



Add User:

Add New User

Account Number:

156789903354

User Name:

Adam

Account Balance:

80000

PIN:

6767

Submit

Display User:

List of Users

ID	Account Number	User Name	Account Balance	PIN
1	156789903354	Adam	80000	6767

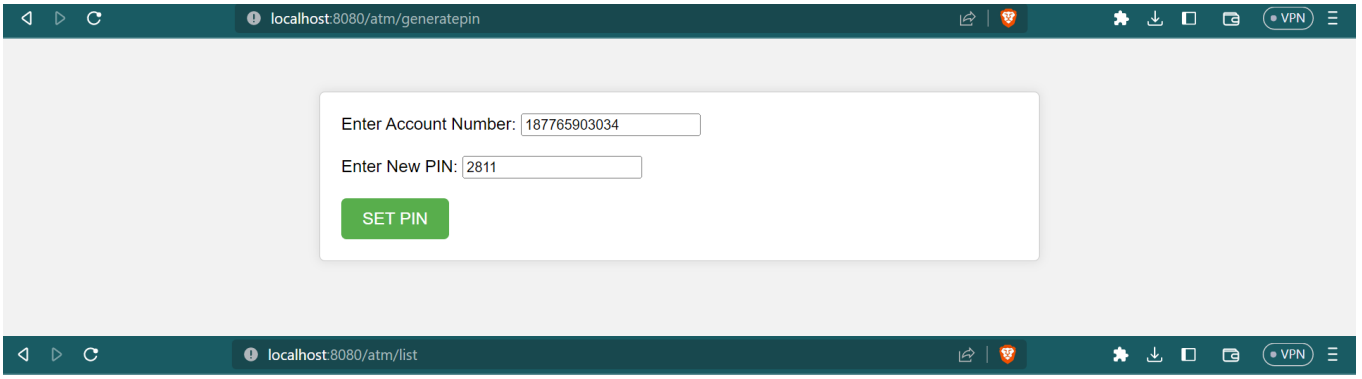
Generate PIN :

Before:

List of Users

ID	Account Number	User Name	Account Balance	PIN
1	156789903354	Adam	80000	6767
2	187765903034	Eve	14000	

After:



List of Users

ID	Account Number	User Name	Account Balance	PIN
1	156789903354	Adam	80000	6767
2	187765903034	Eve	14000	2811

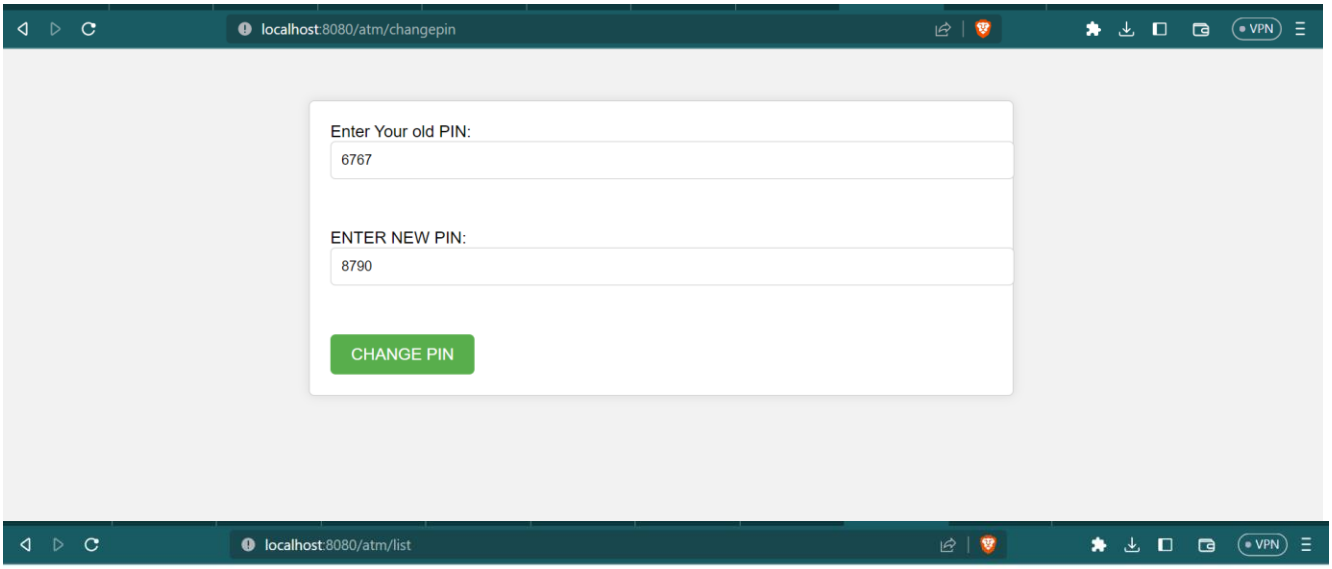
Change PIN:

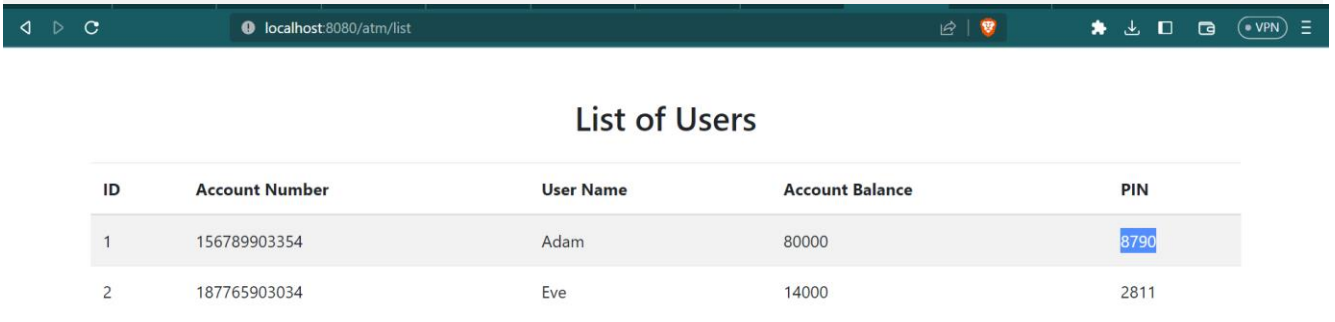
Before:

List of Users

ID	Account Number	User Name	Account Balance	PIN
1	156789903354	Adam	80000	6767
2	187765903034	Eve	14000	2811

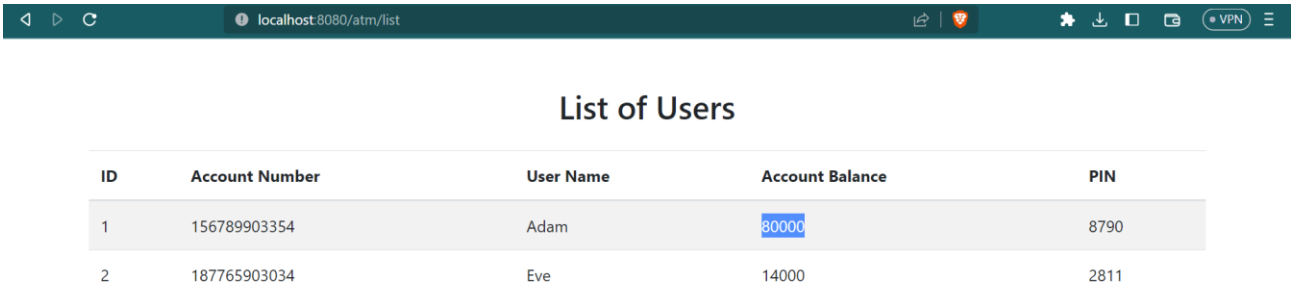
After :



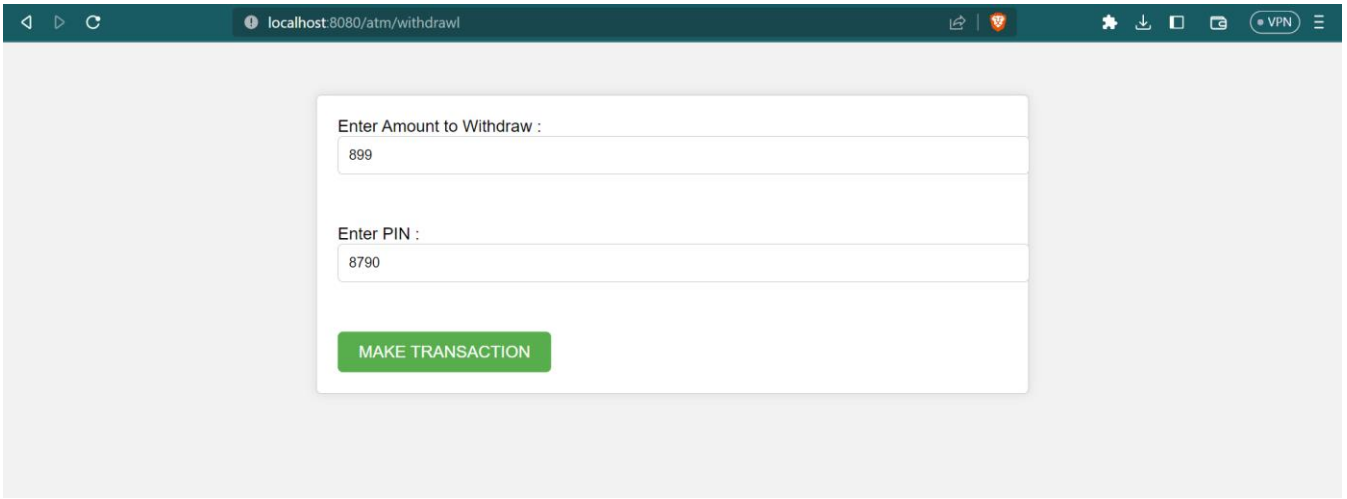


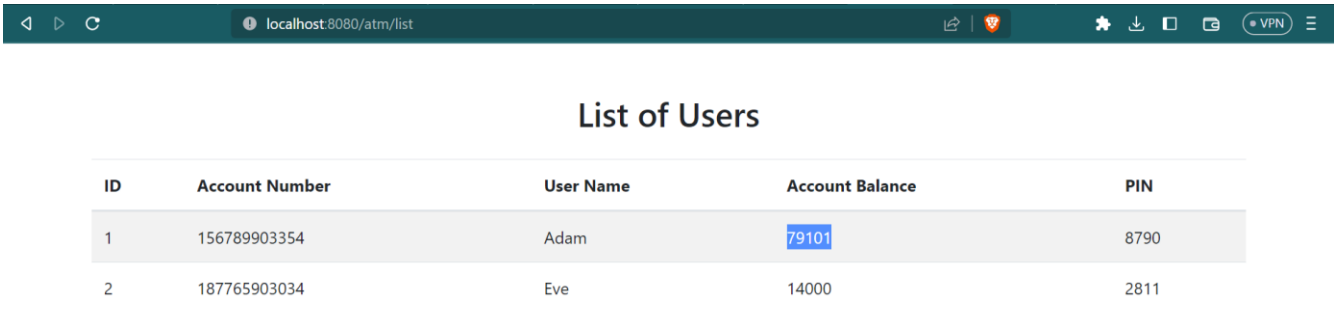
Withdrawl:

Before:

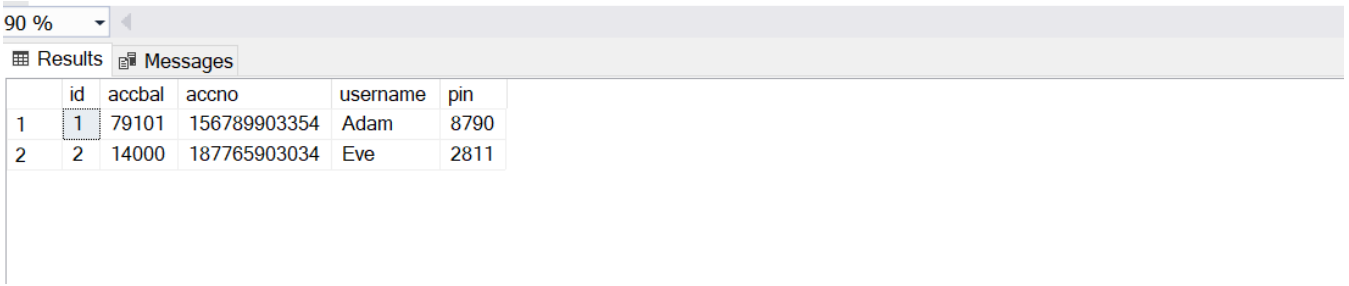


After:


localhost:8080/atm/withdrawl
VPN
Enter Amount to Withdraw :
899
Enter PIN :
8790
MAKE TRANSACTION


localhost:8080/atm/list
VPN
List of Users
ID
Account Number
User Name
Account Balance
PIN
1
156789903354
Adam
79101
8790
2
187765903034
Eve
14000
2811

Database:


90 %
Results
Messages
id
accbal
accno
username
pin
1
1
79101
156789903354
Adam
8790
2
2
14000
187765903034
Eve
2811