**Command-Line Payment Application Documentation**

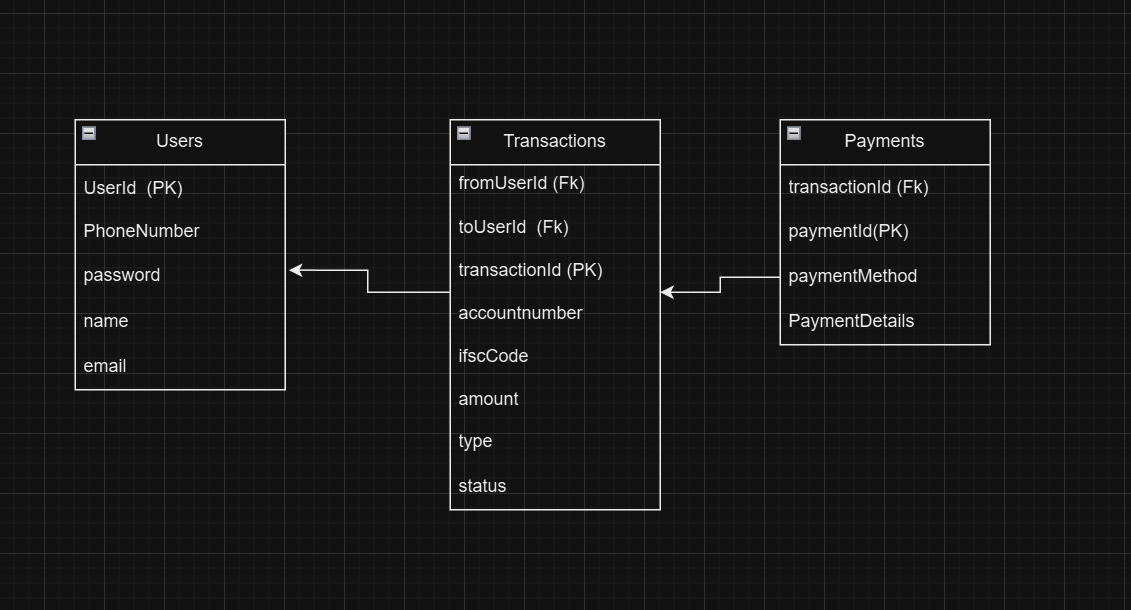
**I)Introduction:**

The Command-Line Payment Application is a Java-based application that facilitates financial transactions through a command-line interface. It allows users to register, log in, update their profiles, send money to other users or bank accounts, make payments, refund transactions, and view their transaction history. The application uses MySQL as the backend database and JDBC for database connectivity.

**Schema Diagram:**



**Class Diagram:**



**II)Features:**

The application supports the following features:

**1)User Management:**

a) Register a new user with a phone number and password.

b) Log in with a registered phone number and password.

c) Update the user profile (name, email, and phone number).

**2)Transactions:**

a) Send money to another user (via phone number).

b)Send money to a bank account (via account number and IFSC code).

c)Make payments for transactions using a card, UPI, or Netbanking.

**3)Refund Transaction:**

a)Refund a transaction (the refund amount goes back to the source).

**4)Transaction History:**

a)View transaction history for the logged-in user.

**III)Technologies Used:**

**Programming Language:** Java

**Database:** MySQL

**Database Connectivity:** JDBC (MySQL Connector/J 9.2.0)

**IDE:** VS Code

**IV)Setup Instructions:**

**Prerequisites:**

Before setting up the project, ensure the following are installed:

Java Development Kit (JDK): Version 8 or later.

MySQL Server: installed and running.

MySQL Connector/J: Download the MySQL JDBC driver (mysql-connector-j-9.2.0.jar).

**V)Database Setup:**

**Create the database:** Open MySQL Command Line or MySQL Workbench and run the following commands:

**SQL Commands:**

CREATE DATABASE PaymentApp;

USE PaymentApp;

**Create Tables:**

Run the following SQL commands to create the required tables:

CREATE TABLE Users (

userId INT AUTO\_INCREMENT PRIMARY KEY,

phoneNumber VARCHAR(15) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

name VARCHAR(255),

email VARCHAR(255)

);

CREATE TABLE Transactions (

transactionId INT AUTO\_INCREMENT PRIMARY KEY,

fromUserId INT NOT NULL,

toUserId INT,

accountNumber VARCHAR(20),

ifscCode VARCHAR(20),

amount DOUBLE NOT NULL,

type ENUM('PAYTM', 'BANK') NOT NULL,

status ENUM('PENDING', 'COMPLETED', 'REFUNDED') NOT NULL,

FOREIGN KEY (fromUserId) REFERENCES Users(userId),

FOREIGN KEY (toUserId) REFERENCES Users(userId)

);

CREATE TABLE Payments (

paymentId INT AUTO\_INCREMENT PRIMARY KEY,

transactionId INT NOT NULL,

paymentMethod ENUM('CARD', 'UPI', 'NETBANKING') NOT NULL,

paymentDetails VARCHAR(255),

FOREIGN KEY (transactionId) REFERENCES Transactions(transactionId)

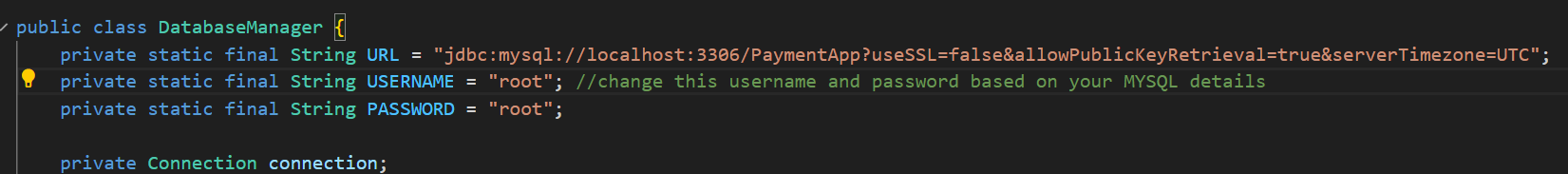
);

**VI)Project Setup:**

**Download the Project:** Clone or download the project to your local machine using the Github url(<https://github.com/kamalhemanth/projects>)

**Add MySQL Connector/J:** Place the mysql-connector-j-9.2.0.jar file in the lib folder of the project.

**Update Database Credentials:** Open the DatabaseManager.java file and update the USERNAME and PASSWORD constants with your MySQL credentials as show in below Example



**Compile the Project:**

Open a terminal or command prompt in the project's root directory and run:

javac -cp lib/mysql-connector-j-9.2.0.jar -d bin src/models/User.java src/models/Transaction.java src/models/Payment.java src/database/DatabaseManager.java src/services/UserService.java src/services/TransactionService.java src/services/PaymentService.java src/Main.java

**Run the Application:**

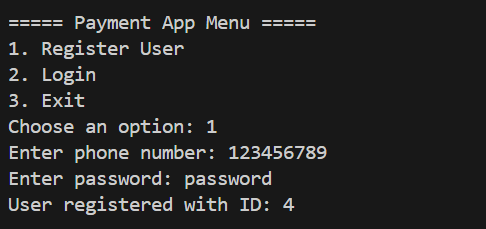
Run the following command to start the application:

java -cp bin;lib/mysql-connector-j-9.2.0.jar Main (in case the command didn't mention the complete path for the jar file as mentioned below)

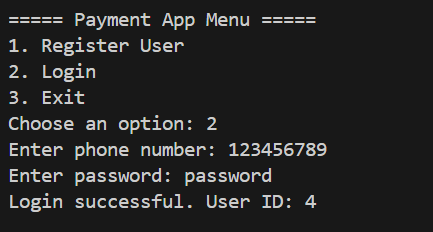
java -cp "bin;C:\Users\kamal\OneDrive\Desktop\PaymentApplication\lib\mysql-connector-j-9.2.0.jar" Main

**VII)Usage Instructions:**

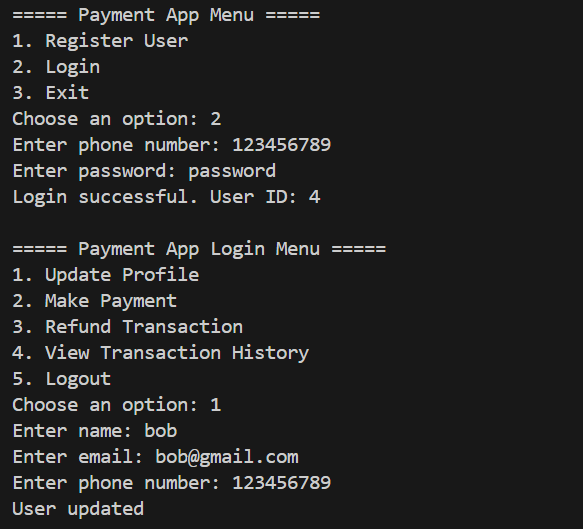
**1. Register a User:** Choose option 1 from the main menu and enter a phone number and password for user registration



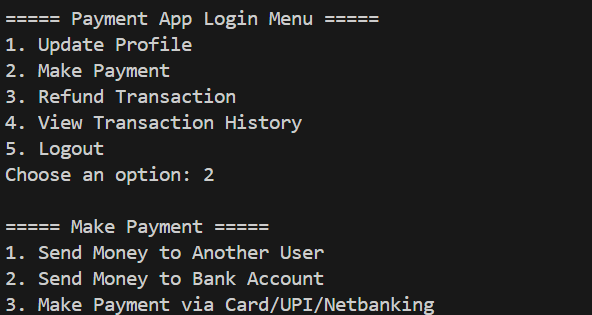
**2. Log In:** Choose option 2 from the main menu and enter the registered phone number and password.



**3. Update Profile:** After logging in, choose option 1 from the payment app Login menu.

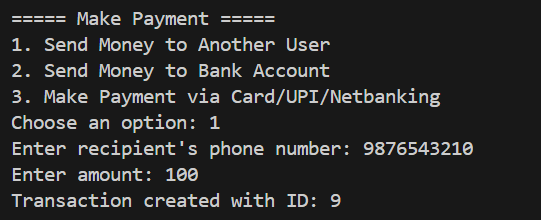


4. **Make a Payment:** Choose any options to make a payment as your wish



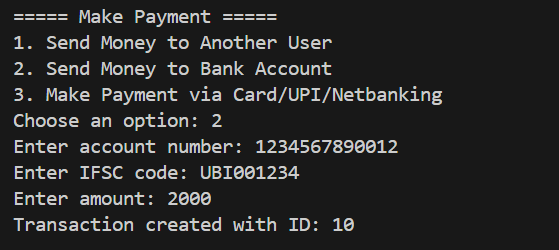
1)Send Money to Another User

* Enter the recipient's phone number and the amount.



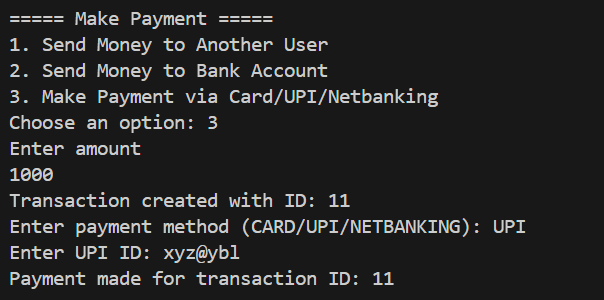
2)Send Money to a Bank Account

* Enter the account number, IFSC code, and the amount.



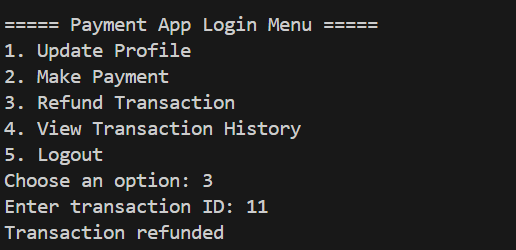
3)Make Payment via Card/UPI/Netbanking

* Enter the amount, payment method (CARD/UPI/NETBANKING), and payment details.



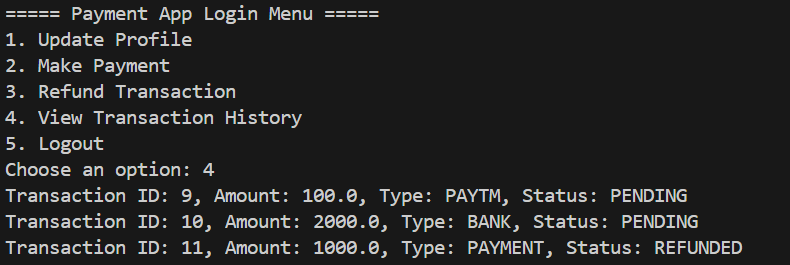
5. Refund a Transaction

Choose option 3 from the payment app Login menu and Enter the transaction ID to refund.



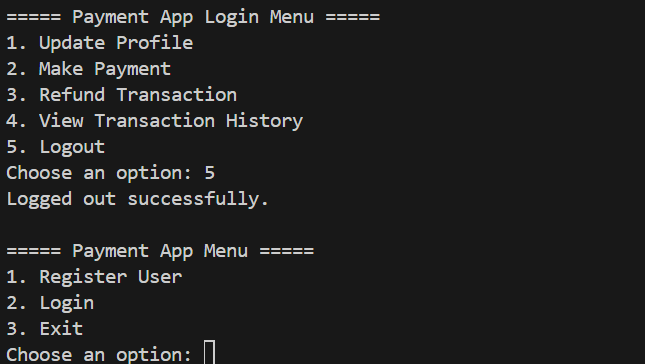
6. View Transaction History

Choose option 4 from the payment app Login menu to view all transactions for the logged-in user.



7. Logout

Choose option 5 from the payment app Login menu to log out and return to the main menu.



**VIII)Future Enhancements:**

**Password Encryption:** Encrypt user passwords before storing them in the database.

**Transaction Notifications:** Send notifications to users for successful transactions.

**GUI:** Develop a graphical user interface (GUI) for better user experience.

**IX)Conclusion:**

The Command-Line Payment Application is a robust and scalable solution for managing financial transactions. It demonstrates the use of Java, MySQL, and JDBC to build a functional command-line application. With proper setup and usage, it can be extended to support more advanced features.