Object Oriented Design

Name : Anita sultana asamony

ID:**0242220005341045**

Section :39 (A)

Submitted to:

Course Instructor : Akash Ghosh

Lecturer, Department of Software Engineering, Daffodil International

Project name: Mobilebanking system

Project Proposal:

- The project aims to develop a mobile banking system where users can perform various transactions such as sending money, cashing out, and paying bills using a Java-based application.
- Key features include user authentication, menu-driven interface, transaction processing, and account management.

Implemented Features:

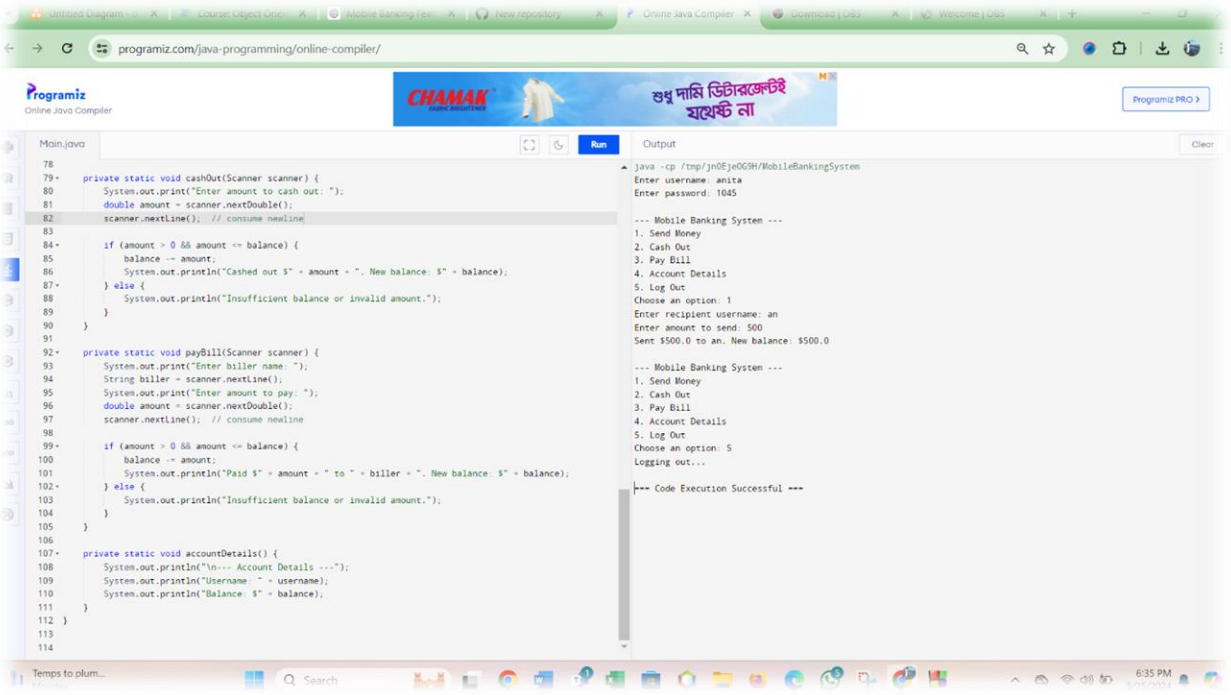
- User authentication using username and password.
- Menu-driven interface for users to select transaction options.
- Transaction functionalities including sending money, cashing out, paying bills, and viewing account details.

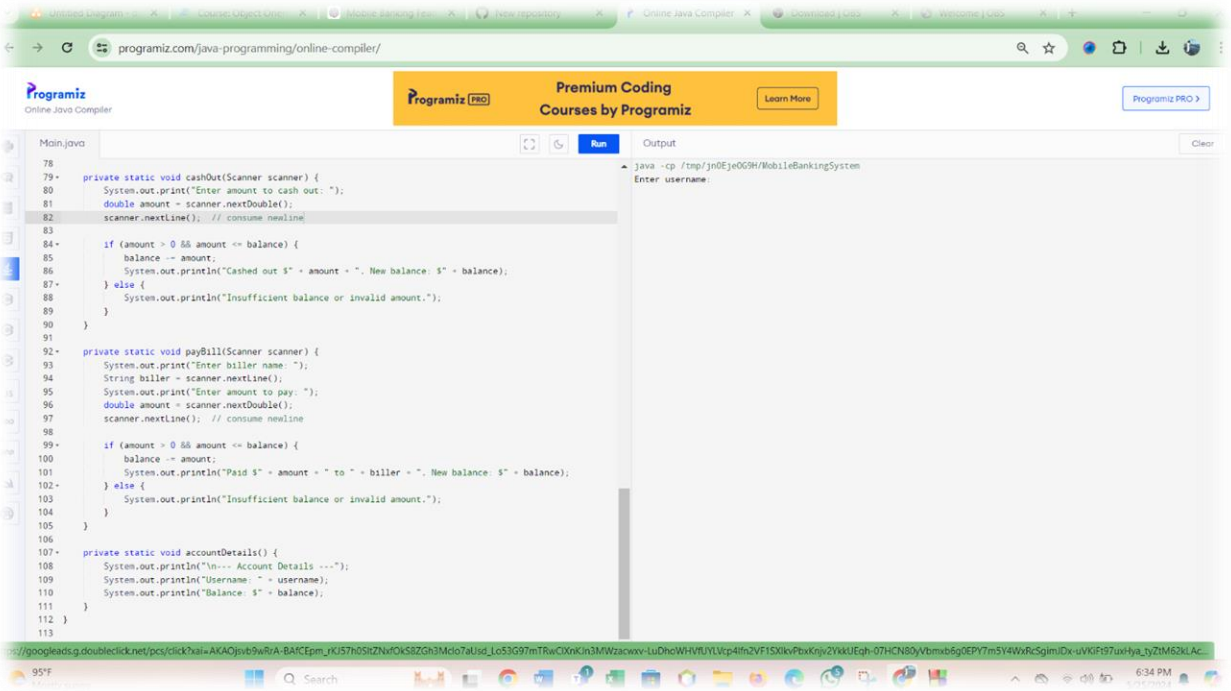
. Tools Used:

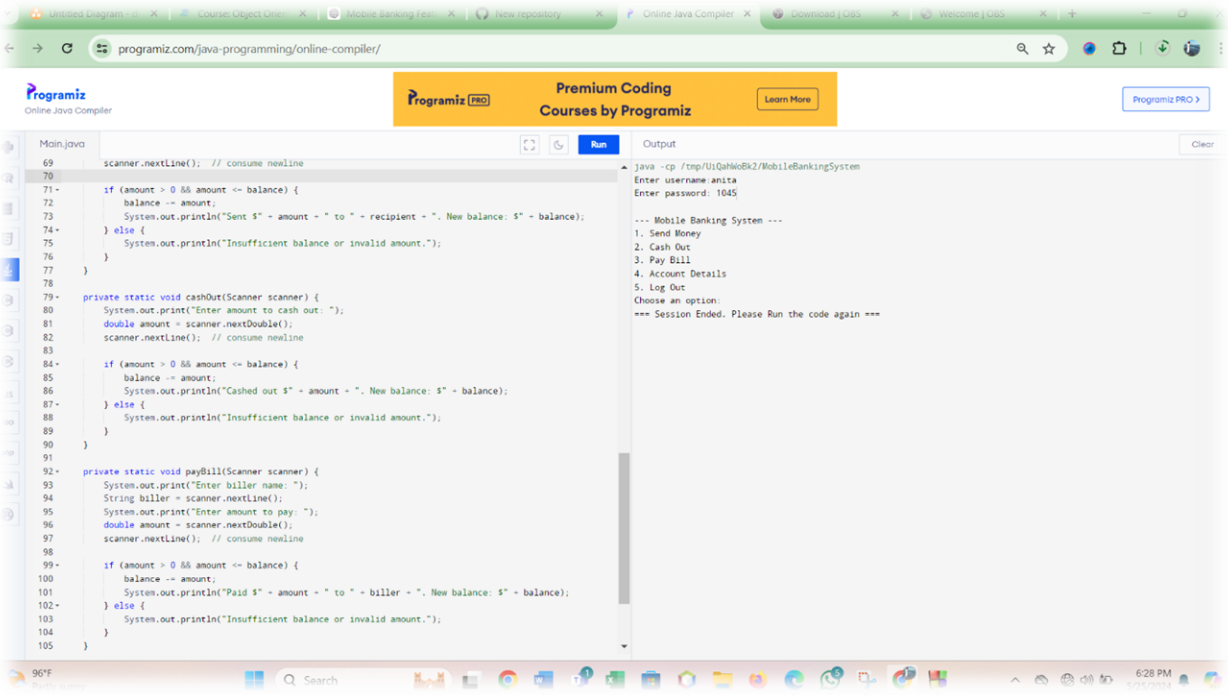
- Java programming language for application development.
- Integrated Development Environment (IDE) such as IntelliJ IDEA or Eclipse for coding and debugging.
- Version control system like Git for managing project revisions.
- Scanner class for user input handling.
- Command-line interface for running and testing the application.

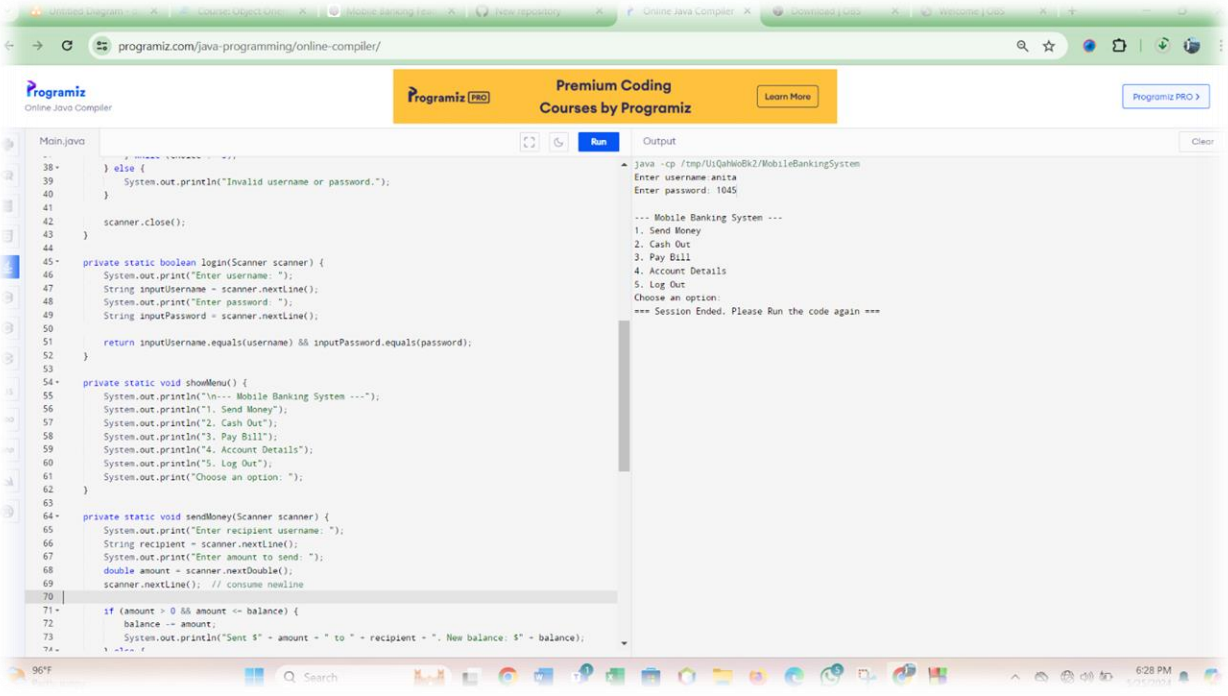
OOP Concepts:

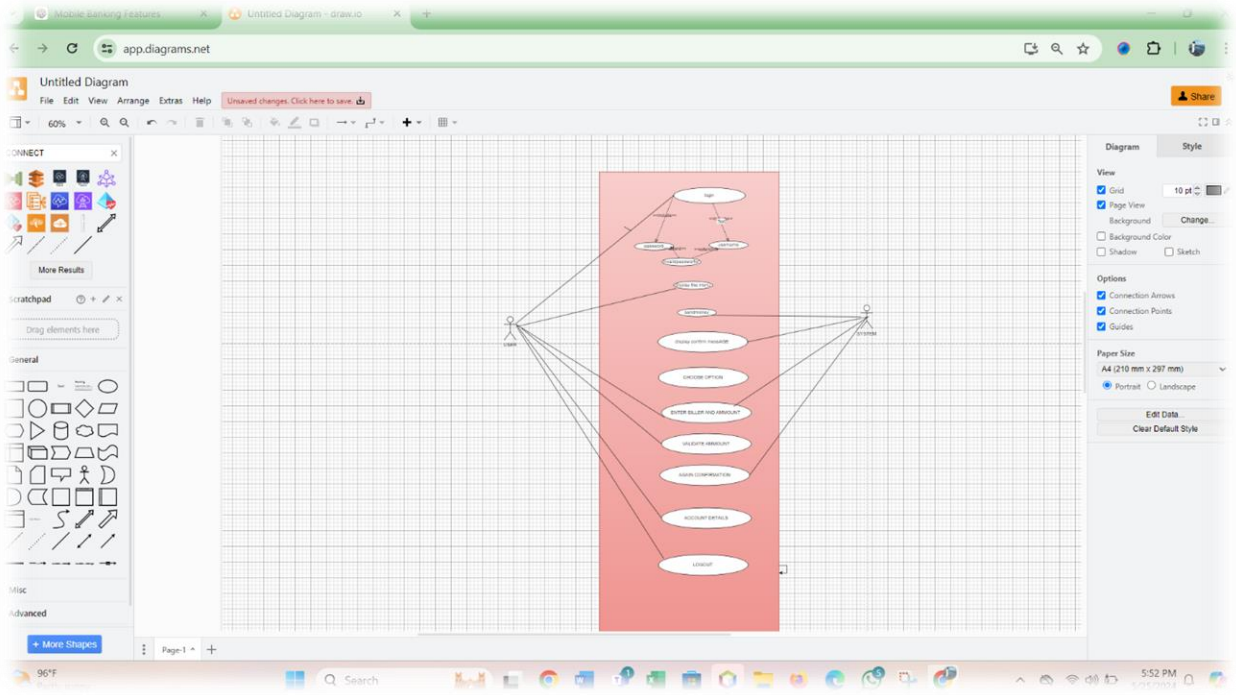
- Yes, the project follows Object-Oriented Programming (OOP) principles such as encapsulation, inheritance, and polymorphism.
- Encapsulation: Data hiding through private access modifiers for class fields and methods.
- Inheritance: The main class inherits functionalities from other classes (e.g., Scanner for input handling).
- Polymorphism: Different methods (e.g., sendMoney, cashOut) perform similar actions but may have different implementations.

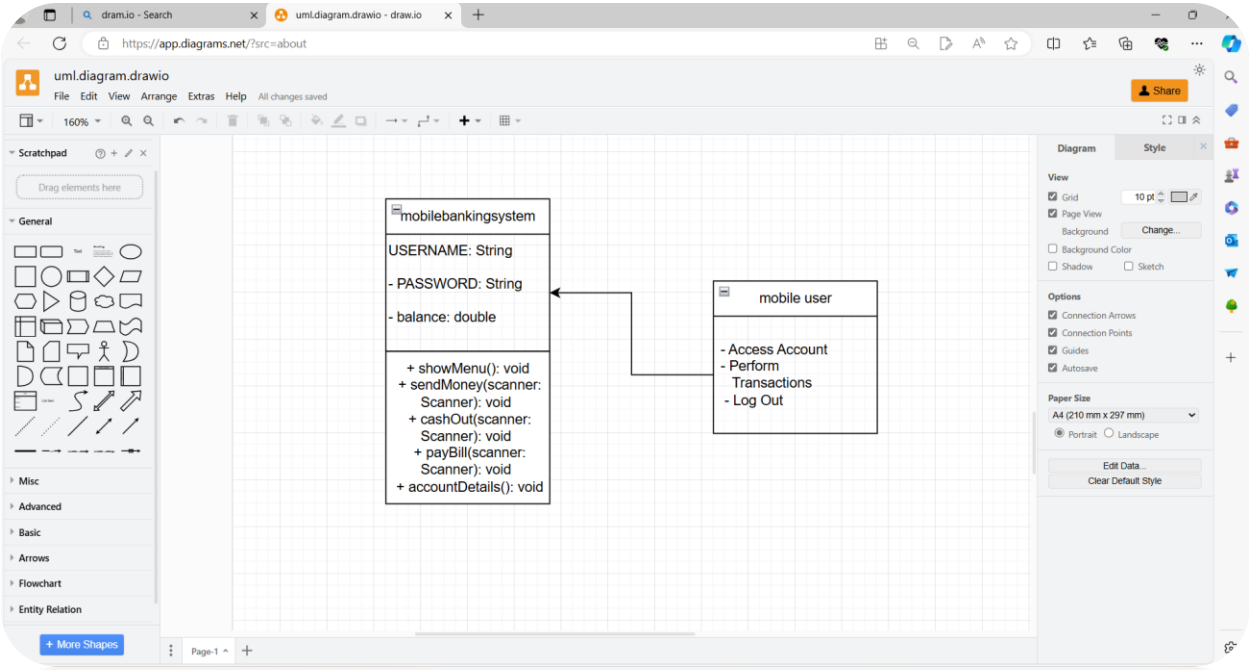

programiz.com/java-programming/online-compiler/
Programiz
Online Java Compiler
CHAMAK
শুধু দামি ডিটারজেন্টই যথেষ্ট না
Programiz PRO
Main.java
Run
Output
Clear
78
79 private static void cashOut(Scanner scanner) {
80 System.out.print("Enter amount to cash out: ");
81 double amount = scanner.nextDouble();
82 scanner.nextLine(); // consume newline
83
84 if (amount > 0 && amount <= balance) {
85 balance -= amount;
86 System.out.println("Cashed out $" + amount + ". New balance: $" + balance);
87 } else {
88 System.out.println("Insufficient balance or invalid amount.");
89 }
90 }
91
92 private static void payBill(Scanner scanner) {
93 System.out.print("Enter biller name: ");
94 String biller = scanner.nextLine();
95 System.out.print("Enter amount to pay: ");
96 double amount = scanner.nextDouble();
97 scanner.nextLine(); // consume newline
98
99 if (amount > 0 && amount <= balance) {
100 balance -= amount;
101 System.out.println("Paid $" + amount + " to " + biller + ". New balance: $" + balance);
102 } else {
103 System.out.println("Insufficient balance or invalid amount.");
104 }
105 }
106
107 private static void accountDetails() {
108 System.out.println("\n--- Account Details ---");
109 System.out.println("Username: " + username);
110 System.out.println("Balance: $" + balance);
111 }
112 }
113
114
java -cp /tmp/jn0EjeOG9H/MobileBankingSystem
Enter username: anita
Enter password: 1045
--- Mobile Banking System ---
1. Send Money
2. Cash Out
3. Pay Bill
4. Account Details
5. Log Out
Choose an option: 1
Enter recipient username: an
Enter amount to send: 500
Sent $500.0 to an. New balance: $500.0
--- Mobile Banking System ---
1. Send Money
2. Cash Out
3. Pay Bill
4. Account Details
5. Log Out
Choose an option: 5
Logging out...
=== Code Execution Successful ===
Temps to plum...
Search
6:35 PM


programiz.com/java-programming/online-compiler/
Programiz
Online Java Compiler
Programiz PRO
Premium Coding Courses by Programiz
Learn More
Programiz PRO
Main.java
Run
Output
Clear
78
79 private static void cashOut(Scanner scanner) {
80 System.out.print("Enter amount to cash out: ");
81 double amount = scanner.nextDouble();
82 scanner.nextLine(); // consume newline
83
84 if (amount > 0 && amount <= balance) {
85 balance -= amount;
86 System.out.println("Cashed out $" + amount + ". New balance: $" + balance);
87 } else {
88 System.out.println("Insufficient balance or invalid amount.");
89 }
90 }
91
92 private static void payBill(Scanner scanner) {
93 System.out.print("Enter biller name: ");
94 String biller = scanner.nextLine();
95 System.out.print("Enter amount to pay: ");
96 double amount = scanner.nextDouble();
97 scanner.nextLine(); // consume newline
98
99 if (amount > 0 && amount <= balance) {
100 balance -= amount;
101 System.out.println("Paid $" + amount + " to " + biller + ". New balance: $" + balance);
102 } else {
103 System.out.println("Insufficient balance or invalid amount.");
104 }
105 }
106
107 private static void accountDetails() {
108 System.out.println("\n--- Account Details ---");
109 System.out.println("Username: " + username);
110 System.out.println("Balance: $" + balance);
111 }
112 }
113
java -cp /tmp/jn0EjeOG9H/MobileBankingSystem
Enter username:
://googleads.g.doubleclick.net/pcs/click?xai=AKAOjsvb9wRrA-BAfCEpm_rKJ57h0StZNxfOkS8ZGh3Mclo7aUsd_Lo53G97mTRwCiXnKIn3MWzacwxv-LuDhoWHVfUYLVcp4ifn2VF1SXlkvPbxKnjv2YkkUEqh-07HCN80yVbmxb6g0EPY7m5Y4WxRcSgimJDx-uVKiFt97uxHya_tyZtM62kLAc...
95°F
Search
6:34 PM


programiz.com/java-programming/online-compiler/
Programiz
Online Java Compiler
Programiz PRO
Premium Coding
Courses by Programiz
Learn More
Programiz PRO
Main.java
Run
Output
Clear
scanner.nextLine(); // consume newline
if (amount > 0 && amount <= balance) {
balance -= amount;
System.out.println("Sent $" + amount + " to " + recipient + ". New balance: $" + balance);
} else {
System.out.println("Insufficient balance or invalid amount.");
}
}
private static void cashOut(Scanner scanner) {
System.out.print("Enter amount to cash out: ");
double amount = scanner.nextDouble();
scanner.nextLine(); // consume newline
if (amount > 0 && amount <= balance) {
balance -= amount;
System.out.println("Cashed out $" + amount + ". New balance: $" + balance);
} else {
System.out.println("Insufficient balance or invalid amount.");
}
}
private static void payBill(Scanner scanner) {
System.out.print("Enter biller name: ");
String biller = scanner.nextLine();
System.out.print("Enter amount to pay: ");
double amount = scanner.nextDouble();
scanner.nextLine(); // consume newline
if (amount > 0 && amount <= balance) {
balance -= amount;
System.out.println("Paid $" + amount + " to " + biller + ". New balance: $" + balance);
} else {
System.out.println("Insufficient balance or invalid amount.");
}
}
java -cp /tmp/UiQahWoBk2/MobileBankingSystem
Enter username:anita
Enter password: 1045
--- Mobile Banking System ---
1. Send Money
2. Cash Out
3. Pay Bill
4. Account Details
5. Log Out
Choose an option:
=== Session Ended. Please Run the code again ===
96°F
Search
6:28 PM


programiz.com/java-programming/online-compiler/
Programiz
Online Java Compiler
Programiz PRO
Premium Coding
Courses by Programiz
Learn More
Programiz PRO
Main.java
Run
Output
Clear
} else {
System.out.println("Invalid username or password.");
}
scanner.close();
}
private static boolean login(Scanner scanner) {
System.out.print("Enter username: ");
String inputUsername = scanner.nextLine();
System.out.print("Enter password: ");
String inputPassword = scanner.nextLine();
return inputUsername.equals(username) && inputPassword.equals(password);
}
private static void showMenu() {
System.out.println("\n--- Mobile Banking System ---");
System.out.println("1. Send Money");
System.out.println("2. Cash Out");
System.out.println("3. Pay Bill");
System.out.println("4. Account Details");
System.out.println("5. Log Out");
System.out.print("Choose an option: ");
}
private static void sendMoney(Scanner scanner) {
System.out.print("Enter recipient username: ");
String recipient = scanner.nextLine();
System.out.print("Enter amount to send: ");
double amount = scanner.nextDouble();
scanner.nextLine(); // consume newline
if (amount > 0 && amount <= balance) {
balance -= amount;
System.out.println("Sent $" + amount + " to " + recipient + ". New balance: $" + balance);
java -cp /tmp/UiQahWoBk2/MobileBankingSystem
Enter username:anita
Enter password: 1045
--- Mobile Banking System ---
1. Send Money
2. Cash Out
3. Pay Bill
4. Account Details
5. Log Out
Choose an option:
=== Session Ended. Please Run the code again ===
96°F
Search
6:28 PM


app.diagrams.net
Untitled Diagram
File Edit View Arrange Extras Help
Share
Diagram
Style
More Shapes


uml.diagram.drawio
File Edit View Arrange Extras Help All changes saved
Share
mobilebankingsystem
USERNAME: String
- PASSWORD: String
- balance: double
+ showMenu(): void
+ sendMoney(scanner: Scanner): void
+ cashOut(scanner: Scanner): void
+ payBill(scanner: Scanner): void
+ accountDetails(): void
mobile user
- Access Account
- Perform Transactions
- Log Out
Diagram
Style
More Shapes

•

Code:

```
import java.util.Scanner;

public class MobileBankingSystem {

    // User credentials
    private static final String USERNAME = "anita";
    private static final String PASSWORD = "1045";
    private static double balance = 1000.00;

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        if (login(scanner)) {
            int choice;
            do {
                showMenu();
                choice = scanner.nextInt();
                scanner.nextLine(); // consume newline
                switch (choice) {
                    case 1:
                        sendMoney(scanner);
                        break;
                    case 2:
                        cashOut(scanner);
                        break;
                    case 3:
                        payBill(scanner);
                        break;
                    case 4:
                        accountDetails();
                        break;
                    case 5:
                        System.out.println("Logging out...");
                        break;
                    default:
                        System.out.println("Invalid choice. Please try again.");
```

```
            }
        } while (choice != 5);
    } else {
        System.out.println("Invalid username or password.");
    }

    scanner.close();
  }

  // Method to handle user login
  private static boolean login(Scanner scanner) {
    System.out.print("Enter username: ");
    String inputUsername = scanner.nextLine();
    System.out.print("Enter password: ");
    String inputPassword = scanner.nextLine();

    return inputUsername.equals(USERNAME) && inputPassword.equals(PASSWORD);
  }

  // Method to display the menu options
  private static void showMenu() {
    System.out.println("\n--- Mobile Banking System ---");
    System.out.println("1. Send Money");
    System.out.println("2. Cash Out");
    System.out.println("3. Pay Bill");
    System.out.println("4. Account Details");
    System.out.println("5. Log Out");
    System.out.print("Choose an option: ");
  }

  // Method to handle sending money
  private static void sendMoney(Scanner scanner) {
    System.out.print("Enter recipient username: ");
    String recipient = scanner.nextLine();
    System.out.print("Enter amount to send: ");
    double amount = scanner.nextDouble();
    scanner.nextLine(); // consume newline

    if (amount > 0 && amount <= balance) {
        balance -= amount;
```

```
            System.out.println("Sent $" + amount + " to " + recipient + ". New balance: $" +
balance);
        } else {
            System.out.println("Insufficient balance or invalid amount.");
        }
    }

    // Method to handle cash out
    private static void cashOut(Scanner scanner) {
        System.out.print("Enter amount to cash out: ");
        double amount = scanner.nextDouble();
        scanner.nextLine(); // consume newline

        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Cashed out $" + amount + ". New balance: $" + balance);
        } else {
            System.out.println("Insufficient balance or invalid amount.");
        }
    }

    // Method to handle bill payment
    private static void payBill(Scanner scanner) {
        System.out.print("Enter biller name: ");
        String biller = scanner.nextLine();
        System.out.print("Enter amount to pay: ");
        double amount = scanner.nextDouble();
        scanner.nextLine(); // consume newline

        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Paid $" + amount + " to " + biller + ". New balance: $" + balance);
        } else {
            System.out.println("Insufficient balance or invalid amount.");
        }
    }

    // Method to display account details
    private static void accountDetails() {
        System.out.println("\n--- Account Details ---");
```

```
        System.out.println("Username: " + USERNAME);
        System.out.println("Balance: $" + balance);
    }
}
```

Video link:

https://drive.google.com/file/d/1h4syfpKChhW5XKEVv_SPLKLmW_RBVKmU/view?usp=sharing

conclusion:

provided Java code implements a basic mobile banking system that allows a user to log in, send money, cash out, pay bills, and check account details. Here is a summary of the functionalities and future plans for the project:

Summary of Implemented Functionalities

1. **User Authentication**:
    - Users can log in using a predefined username and password.
    - If the login credentials are incorrect, access is denied.
2. **Main Menu**:
    - A menu is displayed after a successful login, offering several options: send money, cash out, pay bills, check account details, and log out.
3. **Send Money**:
    - Users can send money to another user by specifying the recipient's username and the amount to send.
    - The system checks if the amount is valid and sufficient before processing the transaction.
4. **Cash Out**:
    - Users can cash out a specified amount.
    - The system verifies if the amount is within the available balance before proceeding.
5. **Pay Bill**:
    - Users can pay bills by entering the biller’s name and the amount to be paid.
    - The system ensures the amount is valid and within the available balance.
6. **Account Details**:
    - Users can view their account details, including the username and current balance.
7. **Log Out**:
    - Users can log out from the system, ending the session.

Future Plans for the Project

To enhance and expand the functionality of this mobile banking system, the following future plans are proposed:

1. **User Registration**:
    - Implement a registration system allowing users to create new accounts with unique usernames and passwords.
2. **Enhanced Security**:
    - Add password encryption and more robust authentication mechanisms.
    - Implement session management to handle multiple users concurrently and securely.
3. **Database Integration**:
    - Store user data and transaction records in a database for better data management and persistence.
    - Implement data retrieval and update mechanisms for real-time data access.
4. **Transaction History**:
    - Provide users with the ability to view their transaction history, including details of all sent money, cash-out transactions, and bill payments.
5. **Improved User Interface**:

    - Develop a graphical user interface (GUI) to make the system more user-friendly and visually appealing.
    - Consider mobile app development for more accessible mobile banking.
6. **Additional Features**:
    - Implement more financial operations such as deposits, withdrawals, and transfers between different accounts.
    - Introduce notifications and alerts for transactions and low balance warnings.
7. **Error Handling and Validation**:
    - Improve error handling to provide more informative messages and prevent invalid inputs.
    - Implement validation checks to ensure data integrity and prevent fraudulent activities.

By implementing these future enhancements, the mobile banking system will become more robust, user-friendly, and secure, providing a better experience for us