Java Banking Application Profiling and Report

Abhishek Kanade

October 29, 2023

1 Introduction

This report analyzes a Java code snippet for a simple banking application. The code provides basic functionality for managing customer accounts, including account creation, account details display, deposit, withdrawal, and transaction history. It uses MySQL as the underlying database for data storage and retrieval.

2 Aim

The purpose of the provided program is to create a basic command-line banking application in Java. The aim is to demonstrate the implementation of core banking functionalities, such as creating customer accounts, managing account details, making deposits and withdrawals, and tracking transaction history. This program serves as a simplified example of how a banking system can be developed using Java and a relational database (MySQL). It is intended to showcase fundamental concepts of database connectivity, SQL query execution.

3 Code Overview

3.1 Customer Class

The 'Customer' class is the core of the application and provides methods for various banking operations. It interacts with the database and allows customers to manage their accounts.

3.1.1 Create New Account (`get_account` method)

This method allows customers to create a new bank account. It collects essential information, such as name, phone number, PAN card number, PIN, account type, and initial deposit. A new row is inserted into the 'Bank_cus' table, and a transaction history table is created for the account.

3.1.2 Show Customer Details (`Show_details` method)

Customers can view their account details by providing the account number. This method retrieves and displays information such as the customer's name, account type, phone number, PAN card number, and current balance.

3.1.3 Deposit Money (`deposit` method)

Customers can deposit money into their accounts using this method. It requires the account number and PIN for security verification. The amount is added to the balance, and a transaction entry is recorded in the respective transaction history table.

3.1.4 Withdraw Money (`withdraw` method)

Similar to the deposit method, this function allows customers to withdraw money from their accounts. It performs security verification, subtracts the withdrawal amount from the balance, and records the transaction.

3.1.5 Show Transaction History (`Show_transition` method)

Customers can view their transaction history by providing the account number and PIN. This method retrieves transaction data from the corresponding transaction history table and displays it in a tabular format.

3.2 Bank Class (Main)

The 'Bank' class serves as the entry point of the program. It establishes a database connection and creates an instance of the 'Customer' class to interact with the banking system's functionality. Users are presented with a menu to choose from the available banking operations.

3.3 Database Connectivity

The code connects to a MySQL database using JDBC, specifying the database URL, username, and password.

3.4 Menu and User Interaction

The program provides a menu where users can select operations by entering a corresponding number.

4 Conclusion

The provided Java code offers a simplified implementation of a banking system. It allows customers to create accounts, view account details, deposit and withdraw money, and check their transaction history. However, in a real-world

scenario, it is essential to implement more robust security and error-handling mechanisms. Additionally, improvements can be made in areas such as using constants for database table names and optimizing SQL queries.

5 Profile

This java application code is Profiled by Jprofiler for graphical overview of code

Call Graph