|  |  |
| --- | --- |
| |  | | --- | | INTRAINZ MAJOR PROJECT | |

##### Java Internship

Project Title : Bank Management System software

using JAVA

Aim of the project: -

Building a basic Bank Management System software using JAVA to deal with all the bank accounts

and transactions in a bank. The proposed system is a web-based project that digitally simulates how

a bank works.

A basic Bank Management System application needs to have the following features :

 Three login portals - for Bank customers, Bank employees and the Admin. The customers

will enter all their personal details while registering themselves into the system. The Bank

employees can login themselves by entering their details like, employee id, name and

position. The admin login should be able to view the profiles of both customers and

employees.

 The Bank customers should be able to deposit money and withdraw money from a particular

account as they desire.

 There should be a validation to allow only a particular amount of cash inflows at any time, as

well as to allow withdrawals if the balance is sufficient. There should also be the calculation

of interest and its addition to the balance every month.

import java.util.\*;

class Transaction {

private String transactionId;

private String accountNumber;

private String transactionType;

private double amount;

private String transactionDate;

public Transaction(String transactionId, String accountNumber, String transactionType, double amount, String transactionDate) {

this.transactionId = transactionId;

this.accountNumber = accountNumber;

this.transactionType = transactionType;

this.amount = amount;

this.transactionDate = transactionDate;

}

public String getAccountNumber() {

return accountNumber;

}

public String getTransactionType() {

return transactionType;

}

public double getAmount() {

return amount;

}

public String getTransactionDate() {

return transactionDate;

}

}

class Customer {

private String customerId;

private String name;

private String address;

private String phoneNumber;

public Customer(String customerId, String name, String address, String phoneNumber) {

this.customerId = customerId;

this.name = name;

this.address = address;

this.phoneNumber = phoneNumber;

}

public String getCustomerId() {

return customerId;

}

public String getName() {

return name;

}

}

class Account {

private String accountNumber;

private String customerId;

private double balance;

private String accountType;

public Account(String accountNumber, String customerId, String accountType) {

this.accountNumber = accountNumber;

this.customerId = customerId;

this.accountType = accountType;

this.balance = 0.0;

}

public String getAccountNumber() {

return accountNumber;

}

public String getCustomerId() {

return customerId;

}

public double getBalance() {

return balance;

}

public void deposit(double amount) {

this.balance += amount;

}

public void withdraw(double amount) {

if (balance >= amount) {

this.balance -= amount;

} else {

System.out.println("Insufficient balance!");

}

}

public String getAccountType() {

return accountType;

}

}

class BankService {

private Map<String, Customer> customers = new HashMap<>();

private Map<String, Account> accounts = new HashMap<>();

private List<Transaction> transactions = new ArrayList<>();

public void createCustomer(String customerId, String name, String address, String phoneNumber) {

Customer customer = new Customer(customerId, name, address, phoneNumber);

customers.put(customerId, customer);

System.out.println("Customer created successfully!");

}

public void createAccount(String accountNumber, String customerId, String accountType) {

if (customers.containsKey(customerId)) {

Account account = new Account(accountNumber, customerId, accountType);

accounts.put(accountNumber, account);

System.out.println("Account created successfully!");

} else {

System.out.println("Customer not found!");

}

}

public void deposit(String accountNumber, double amount) {

Account account = accounts.get(accountNumber);

if (account != null) {

account.deposit(amount);

transactions.add(new Transaction("T" + (transactions.size() + 1), accountNumber, "Deposit", amount, "2024-08-18"));

System.out.println("Deposit successful!");

} else {

System.out.println("Account not found!");

}

}

public void withdraw(String accountNumber, double amount) {

Account account = accounts.get(accountNumber);

if (account != null) {

account.withdraw(amount);

transactions.add(new Transaction("T" + (transactions.size() + 1), accountNumber, "Withdrawal", amount, "2024-08-18"));

System.out.println("Withdrawal successful!");

} else {

System.out.println("Account not found!");

}

}

public double getBalance(String accountNumber) {

Account account = accounts.get(accountNumber);

if (account != null) {

return account.getBalance();

} else {

System.out.println("Account not found!");

return 0.0;

}

}

public void viewTransactionHistory(String accountNumber) {

System.out.println("Transaction History for Account: " + accountNumber);

for (Transaction transaction : transactions) {

if (transaction.getAccountNumber().equals(accountNumber)) {

System.out.println(transaction.getTransactionType() + ": " + transaction.getAmount() + " on " + transaction.getTransactionDate());

}

}

}

}

public class BankManagementSystem {

public static void main(String[] args) {

BankService bankService = new BankService();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\n--- Bank Management System ---");

System.out.println("1. Create Customer");

System.out.println("2. Create Account");

System.out.println("3. Deposit");

System.out.println("4. Withdraw");

System.out.println("5. View Balance");

System.out.println("6. View Transaction History");

System.out.println("7. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.print("Enter Customer ID: ");

String customerId = scanner.next();

System.out.print("Enter Name: ");

String name = scanner.next();

System.out.print("Enter Address: ");

String address = scanner.next();

System.out.print("Enter Phone Number: ");

String phoneNumber = scanner.next();

bankService.createCustomer(customerId, name, address, phoneNumber);

break;

case 2:

System.out.print("Enter Account Number: ");

String accountNumber = scanner.next();

System.out.print("Enter Customer ID: ");

customerId = scanner.next();

System.out.print("Enter Account Type (Savings/Checking): ");

String accountType = scanner.next();

bankService.createAccount(accountNumber, customerId, accountType);

break;

case 3:

System.out.print("Enter Account Number: ");

accountNumber = scanner.next();

System.out.print("Enter Deposit Amount: ");

double depositAmount = scanner.nextDouble();

bankService.deposit(accountNumber, depositAmount);

break;

case 4:

System.out.print("Enter Account Number: ");

accountNumber = scanner.next();

System.out.print("Enter Withdrawal Amount: ");

double withdrawalAmount = scanner.nextDouble();

bankService.withdraw(accountNumber, withdrawalAmount);

break;

case 5:

System.out.print("Enter Account Number: ");

accountNumber = scanner.next();

double balance = bankService.getBalance(accountNumber);

System.out.println("Current Balance: " + balance);

break;

case 6:

System.out.print("Enter Account Number: ");

accountNumber = scanner.next();

bankService.viewTransactionHistory(accountNumber);

break;

case 7:

System.out.println("Exiting the system.");

scanner.close();

System.exit(0);

default:

System.out.println("Invalid option. Please try again.");

}

}

}

}













