BANKING APPLICATION IN JAVA

Project Report

Mini Project(IDS353)

Degree

B.Tech CS&E DATA SCIENCE
(In Collabration with Inurture)

PROJECT GUIDE:

MR. SUDHANSHU KUMAR

SUBMITTED BY:

Ali Faiz (TCA2166003)
Priyanshu (TCA2166015)
Sahil Jain (TCA2166016)

DEC. , 2022

College Of Computing Science And Information Technology

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

DECLARATION

We hereby declare that this Project Report titled **BANKING APPLICATION IN JAVA** submitted by us and approved by our project guide, Faculty of Engineering & Computing Sciences. Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Signature

Student Name: Ali faiz

Student Name: Priyanshu

Student Name: Sahil Jain

Project Guide : MR. Sudhanshu Kumar

Table of Contents

Appendix
A: Data Flow Diagram (DFD)
B: Entity Relationship Diagram (ERD)
C: Use Case Diagram (UCD)
D: Input Code for your progrm
E: Screen Shots

1 Project Title

" BANKING APPLICATION IN JAVA"

The banking management system is an application for maintaining a person's account in a Bank.

This project have facility to opening account, depositing and withdrawing money.

2 Problem Statement

In any Bank Transaction, there are several parties involved to process transaction like a merchant, bank, receiver, etc. so there are several numbers reasons that transaction may get failed, declined, so to handle a transaction in Java, there is a JDBC (Java Database Connectivity) which provides us an API to connect, execute, fetch data from any databases. It provides the language Java database connectivity standards. It is used to write programs required to access databases

3 Project Description

The Bank management system is an application for maintaining a person’s account in a bank. The system provides the access to thecustomer to create an account, deposit/withdraw the cash fromhis account, also to view reports of all accounts present. Thefollowing presentation provides the specification for the system.

3.1 Scope of the Work

This Bank Application System has been designed to overcome all the disadvantages of the traditional banking system. A user has the option of creating an account, cash transactions all at their own discretion. There is a separate module for each and every action

3.2 Project Modules

Bank Application System Modules

1.New Account Module:– This module lets users add a new account by providing name, address and age.

2.Deposit module:– This module has been designed to assist the users in depositing money. It takes as input account number, type, date and amount.

3.Display module:– This module provides account information, balance, withdrawal and deposit information. It has four sub-modules:

4.Account Info module:– This displays the information of the account whose number has been entered into the given field

5.Balance Info module:– This displays the current balance in the queried account

6.Deposit Info module:– This shows the logs of the deposits which have been made into the account.

7.Withdrawal Info module:– This module shows the logs of the withdrawals that have been made from the account.

8.Withdrawal module:– This module has been designed to assist the users in withdrawing money. It takes as input account number, type, date and amount.

9.Delete Account module:– This lets user to delete an existing account.

4 Implementation Methodology

In this section, we will learn how to create a mini-application for a banking system in Java. In this program, we will add some basic functionalities of a bank account like a deposit of amount, withdrawal of amount, etc. Initially, the program accepts the number of customers we need to add and adds the customer and account details accordingly. Further, it displays the series of menus to operate over the accounts.

The series of menus displayed are as follows:

1.Display all account details

2.Search by account number

3.Deposit the amount

4.Withdraw the amount

5.Exit

5 Technologies to be used

5.1 Software Platform

Front-end – JAVA

5.2 Hardware Platform

RAM, Hard Disk

6 Advantages of this Project

The manual Bank Application System that carry out the whole process in a bank are more prone to errors. It takes a vast amount of time to take an action be it the creation of an account or depositing money. It is especially negative when there are time constraints for the customers to get a work done. This makes it a lot inconvenient for the customer as well as it is time consuming. Banks are open only in limited hours in a day and any transaction to be made after those times in cases of emergencies become impossible. This causes a lot of inconvenience

7 Future Scope and further enhancement of the Project

Today's online banking customers enjoy instant access from their computer or mobile device to a full range of services, allowing you to: Check balances on accounts and view records of your transactions. Pay bills automatically each month with easy-to-set-up auto payment. Transfer funds between accounts.

8 Team Details

Project Name & ID	Course Name	Student ID	Student Name	Role	Signature
Banking Application in JAVA	Project (IDS353)	TCA2166003	Ali Faiz	Implementation	
		TCA2166015	Priyanshu	Document work	
		TCA2166016	Sahil Jain	Implementation	

9 Conclusion

The application illustrated or demonstrates the way to develop an online banking system by using interactive web client by using JSP, Servlet with safer way to access &encapsulate database by EJB component. This suggests the application server simply deployable and accessible.

This project developed, incorporated all the activities involved in the browsing centre. It provides all necessary information to the management as well as the customer with the use of this system; the user can simply sit in front of the system and monitor all the activities without any physical movement of the file. Management can service the customers request best in time. The system provides quickly and valuable information. These modules have been integrated for effective use of the management for future forecasting and for the current need.

10 Refrences

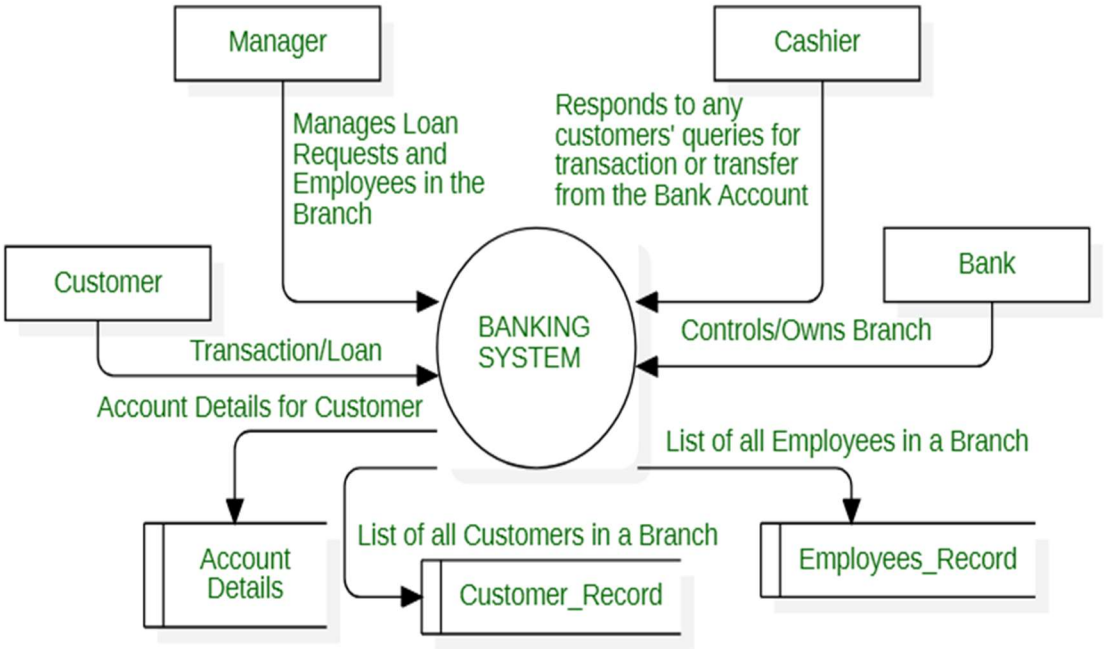
Books Used for study – Programming with JAVA
(~ E Balagurusamy)

Online Websites Used - www.javatpoint.com

www.geeksforgeeks.org

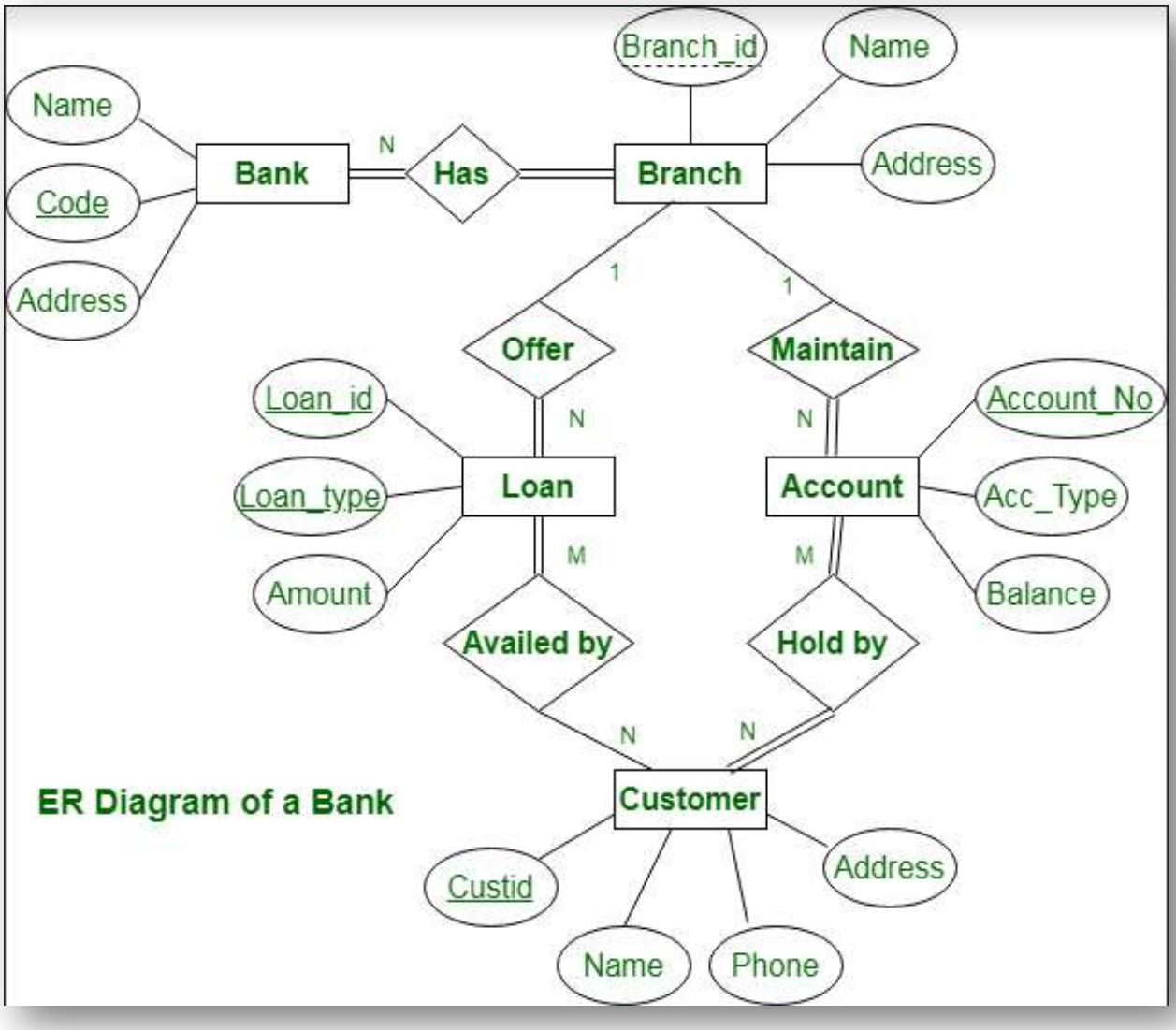
www.academia.edu

Annexure A
Data Flow Diagram (DFD)
(Mandatory)



LEVEL - 0

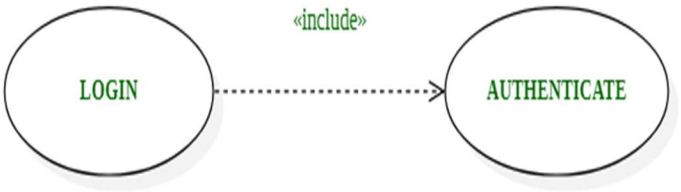
Annexure B
Entity-Relationship Diagram (ERD)
(Mandatory)



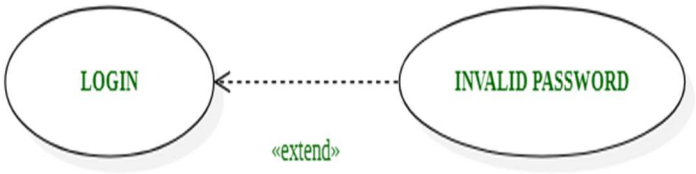
Annexure C

Use-Case Diagram (UCD)

(Optional)



LOGIN is the BASE USE CASE and AUTHENTICATE is the INCLUDED USE CASE.



ILOGIN is the BASE USE CASE and INVALID PASSWORD is the EXTENDED USE CASE.

Annexure D
Input Code for JAVA

```
import java.util.Scanner;
class BankDetails {
   private String accno;
   private String name;
   private String acc_type;
   private long balance;
   Scanner sc = new Scanner(System.in);
   //method to open new account
   public void openAccount() {
      System.out.print("Enter Account No: ");
      accno = sc.next();
      System.out.print("Enter Account type: ");
      acc_type = sc.next();
      System.out.print("Enter Name: ");
      name = sc.next();
      System.out.print("Enter Balance: ");
      balance = sc.nextLong();
   }
   //method to display account details
   public void showAccount() {
      System.out.println("Name of account holder: " + name);
      System.out.println("Account no.: " + accno);
      System.out.println("Account type: " + acc_type);
      System.out.println("Balance: " + balance);
   }
   //method to deposit money
   public void deposit() {
      long amt;
      System.out.println("Enter the amount you want to deposit: ");
      amt = sc.nextLong();
      balance = balance + amt;
   }
   //method to withdraw money
   public void withdrawal() {
      long amt;
      System.out.println("Enter the amount you want to withdraw: ");
      amt = sc.nextLong();
      if (balance >= amt) {
         balance = balance - amt;
         System.out.println("Balance after withdrawal: " + balance);
      } else {
         System.out.println("Your balance is less than " + amt + "\tTransaction failed...!!" );
      }
   }
```

```
   //method to search an account number
   public boolean search(String ac_no) {
      if (accno.equals(ac_no)) {
         showAccount();
         return (true);
      }
      return (false);
   }
}
public class BankingApp {
   public static void main(String arg[]) {
      Scanner sc = new Scanner(System.in);
      //create initial accounts
      System.out.print("How many number of customers do you want to input? ");
      int n = sc.nextInt();
      BankDetails C[] = new BankDetails[n];
      for (int i = 0; i < C.length; i++) {
         C[i] = new BankDetails();
         C[i].openAccount();
      }
      // loop runs until number 5 is not pressed to exit
      int ch;
      do {
         System.out.println("\n ***Banking System Application***");
         System.out.println("1. Display all account details \n 2. Search by Account number\n 3. Deposit the
amount \n 4. Withdraw the amount \n 5.Exit ");
         System.out.println("Enter your choice: ");
         ch = sc.nextInt();
            switch (ch) {
               case 1:
                  for (int i = 0; i < C.length; i++) {
                     C[i].showAccount();
                  }
                  break;
               case 2:
                  System.out.print("Enter account no. you want to search: ");
                  String ac_no = sc.next();
                  boolean found = false;
                  for (int i = 0; i < C.length; i++) {
                     found = C[i].search(ac_no);
                     if (found) {
                        break;
                     }
                  }
                  if (!found) {
                     System.out.println("Search failed! Account doesn't exist..!!");
                  }
                  break;
```

```
            case 3:
               System.out.print("Enter Account no. : ");
               ac_no = sc.next();
               found = false;
               for (int i = 0; i < C.length; i++) {
                  found = C[i].search(ac_no);
                  if (found) {
                     C[i].deposit();
                     break;
                  }
               }
               if (!found) {
                  System.out.println("Search failed! Account doesn't exist..!!");
               }
               break;
            case 4:
               System.out.print("Enter Account No : ");
               ac_no = sc.next();
               found = false;
               for (int i = 0; i < C.length; i++) {
                  found = C[i].search(ac_no);
                  if (found) {
                     C[i].withdrawal();
                     break;
                  }
               }
               if (!found) {
                  System.out.println("Search failed! Account doesn't exist..!!");
               }
               break;
            case 5:
               System.out.println("See you soon...");
               break;
         }
      }
      while (ch != 5);
   }
}
```

Annexure E
Screenshots

(Output Results Attached below)