Rishita Gupta

Rishitagupota790@gmail.com

Bank application

In this application I would work on the basic step which is required for bank just like (transaction , withdraw , deposit , transfer and if the work ending they the account get log off).

[High level design (hld)]

[Banking Application]

Contents

Table of Contents:

1. [Bank Management System](https://iq.opengenus.org/system-design-of-bank-management-system/#bankmanagementsystem)
2. [System Requirements](https://iq.opengenus.org/system-design-of-bank-management-system/#systemrequirements)
   * [Functional Requirements](https://iq.opengenus.org/system-design-of-bank-management-system/#functionalrequirements)
   * [Non-Functional Requirements](https://iq.opengenus.org/system-design-of-bank-management-system/#nonfunctionalrequirements)
   * [Software Requirements](https://iq.opengenus.org/system-design-of-bank-management-system/#softwarerequirements)
3. [High-Level Design](https://iq.opengenus.org/system-design-of-bank-management-system/#highleveldesign)
   * [Monolithic Architecture](https://iq.opengenus.org/system-design-of-bank-management-system/#monolithicarchitecture)
   * [Event-Driven Architecture](https://iq.opengenus.org/system-design-of-bank-management-system/#eventdrivenarchitecture)
4. [Low-Level Design](https://iq.opengenus.org/system-design-of-bank-management-system/#lowleveldesign)
5. [Leading Bank Management Systems](https://iq.opengenus.org/system-design-of-bank-management-system/#leadingbankmanagementsystems)
6. Code

Abstract

This application basically work on banking application from user side if user enter their id and password and once it get verify then user want to do operation accordingly . once their work get completed then the option come to log off . this is quite basic intro of application.

Introduction

**INTRODUCTION**

The main objective of the project is to develop online Banking system for banks. In present system all banking work is done manually. User have to visit bank to Withdrawal or Deposit amount. In present bank system it is also difficult to find account information of account holder. In this bank management system we will automate all the banking process. In our bank management system user can check his balance online and he can also transfer money to other account online. In this Software you can keep record for daily Banking transactions. The main purpose of developing bank management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

This bank management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also provide security check to reduce fraud. The system will check the user’s existence in the database and provide the set of services with respect to the role of the user.

The objective of this bank management system are:-

The main object of this system is to provide a secure system. Our system is password protected and it only allows authorized user to access various functions available in the system.

Our system will help the user to Locate any A/C wanted by the user. It will Reduced manual work as most of the work done by computer. As all the manual work will be done automatically so it will increase work speed and reduce time consumption to complete any bank related work. It will also increase the work efficiency as few employees can handle more customers. This will reduced the manual workload and give information instantly.

The Project Banking system has been made to automate the Banking system. Through this bank management system user can manage all bank account activity like deposit money, withdraw money, transfer money from one account to another account, online payment etc. Using this bank management system user can check his account detail online like balance in account, bank statement etc. The Administrator can check bank account with a login can work out with A/C holders of the bank can withdraw/ deposit cash / cheque /DD to/from their accounts. This system is also help bank user to create New account easily. The project makes a sincere effort to provide all the below-mentioned features to meet the requirements of the bank.

In this project we have automate the bank process like Account Opening, Daily Transactions, Loan Sanctions, Account Maintenance. In this bank management system use can also search record of a particular Account Holder.

Using this system user can manage following account type

Deposit money

Withdraw money

Transfer money

Exit

### FEATURES PROPOSED IN THE CURRENT BANK MANAGEMENT SYSTEM

The main objective of the system is to automate all the banking process with improved performance an realize the vision of paperless banking. Salient features of the proposed bank management system is given below.

Using this bank management system any information can be easily searched. User can view all the details of the customer.

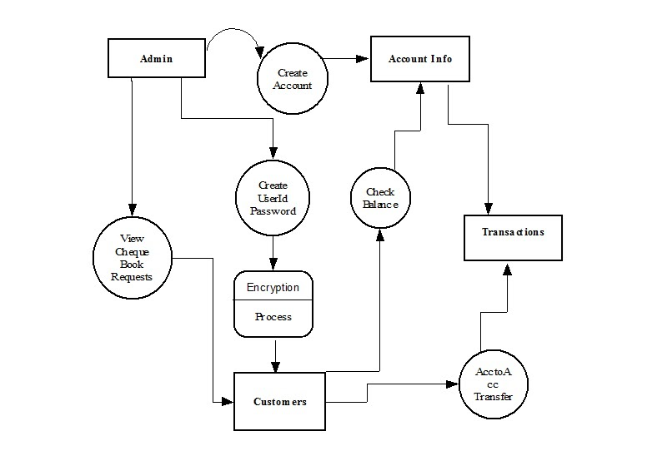
Using this system user can create new customer account and maintain its data efficiently and effectively. All records of account and customer are stored in separate files. Which are maintained constantly update by system.

Manage large number of customer details with ease. Particular A/c information can be modified A particular customer record can be modified for one or more field’s customer name, address by providing A/c number.

Create a statistical report to facilitate the finance department work. Activities like updating, modification, deletion of records should be easier. A customer record can be easily deleted by authorize user by providing A/c number.

The proposed system provides faster data access, data entry and retrieval.

The proposed system is more efficient, fast, reliable, user friendly. Over and above the proposed system does not have any possibility of data loss during processing.

Data flow diagram of bank application . The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user’s work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using PHP, HTML language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyses the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP and HTML. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

Take control

Online banking helps you become more of a banker, running your accounts like a small

business that you control every day. Once you get started, you'll be hooked. Soon enough

you'll be checking your bank account as often as your e-mail.

Features of BAMS

• User registration for online banking if not register.

• Adding Beneficiary account by customer.

• Transferring amount to the local customer account number.

• Admin must approve the user account activation before it can be used and transferring

funds, view statement history.

Take control

Online banking helps you become more of a banker, running your accounts like a small

business that you control every day. Once you get started, you'll be hooked. Soon enough

you'll be checking your bank account as often as your e-mail.

Features of BAMS

• User registration for online banking if not register.

• Adding Beneficiary account by customer.

• Transferring amount to the local customer account number.

• Admin must approve the user account activation before it can be used and transferring

funds, view statement history.

• Customer gets to know his last login date and time each time he logs in.

• Customer can check all transactions made with their account.

• Customer can check their account statement within a date range.

• Customer can request for ATM and Cheque Book.

• Admin can add/edit/delete customer account’s

• All two of them (customer & admin) can change their password.

• Admin Login pages are hidden from customer for security purpose.

• Passwords are stored as encrypted hashes with an additional random salt for added

security.

Goals and Objectives

1. Main Goals:

o Our motto is to develop a software program for managing the entire bank

process related to Administration accounts customer accounts and to keep

each every track about their property and their various transaction processes

efficiently.

o Hereby, our main objective is the customer’s satisfaction considering today’s

faster in the world.

2. Customer Satisfaction:

o Client can do his operations comfortably without any risk or losing of his

privacy.

o Our software will perform and fulfill all the tasks that any customer would

desire.

3. Saving Customer Time:

o Client doesn't need to go to the bank to do small operation.

4. Protecting The Customer:

o It helps the customer to be satisfied and comfortable in his choices, this

protection contains customer’s account, money and his privacy.

o Help client transferring money to/or another bank or country.

5. Transferring Money:

.System Design

Design is the first step into the development phase for any engineered product or system.

Design is a creative process. A good design is the key to effective system. The term “design”

is defined as “the process of applying various techniques and principles for the purpose of

defining a process or a system in sufficient detail to permit its physical realization”. It may be

defined as a process of applying various techniques and principles for the purpose of defining

a device, a process or a system in sufficient detail to permit its physical realization. Software

design sits at the technical kernel of the software engineering process and is applied

regardless of the development paradigm that is used. The system design develops the

architectural detail required to build a system or product. As in the case of any systematic

approach, this software too has undergone the best possible design phase fine tuning all

efficiency, performance and accuracy levels. The design phase is a transition from a user

oriented document to a document to the programmers or database personnel.

System design goes through two phases of development:

⎫ Logical Design and

⎫ Physical Design.

3.2.Logical Design

The logical flow of a system and define the boundaries of a system. It includes the following

steps:

⎫ Reviews the current physical system – its data flows, file content, volumes,

frequencies etc.

⎫ Prepares output specifications – that is, determines the format, content and frequency

of reports.

⎫ Prepares input specifications – format, content and most of the input functions.

⎫ Prepares edit, security and control specifications.

⎫ Specifies the implementation plan.

⎫ Prepares a logical design walk through of the information flow, output, input, controls

and implementation plan.

⎫ Reviews benefits, costs, target dates and system constraints.

Physical Design

Physical system produces the working systems by define the design specifications that tell the

programmers exactly what the candidate system must do. It includes the following steps.

⎫ Design the physical system.

⎫ Specify input and output media.

⎫ Design the database and specify backup procedures.

⎫ Design physical information flow through the system and a physical design Walk

through.

⎫ Plan system implementation.

⎫ Prepare a conversion schedule and target date.

⎫ Determine training procedures, courses and timetable.

⎫ Devise a test and implementation plan and specify any new hardware/software.

⎫ Update benefits, costs, and conversion date and system constraints.

CODE

In first package-

package com.iNeuron;

import java.sql.SQLException;

import java.util.Scanner;

class DbConnector {

int id = 1;

String pass = "Password";

public int getId() {

return id;

}

public String getPass() {

return pass;

}

}

class Access {

Integer userId;

String userPass;

Scanner scan;

void access() throws SQLException {

DbConnector db = new DbConnector();

scan = new Scanner(System.in);

DbConnector ba = new DbConnector();

System.out.println("Enter your id :: ");

userId = scan.nextInt();

System.out.println("Enter your Password :: ");

userPass = scan.next();

int id = ba.getId();

System.out.println(id +" you detail "+ userId );// for checking

String pass = ba.getPass();

if (id == userId && pass.equals(userPass)) {

System.out.println("Welcome \nStart your Further Procedure");

ShowView sv = new ShowView();

sv.display();

} else {

System.err.println("Wrong Password");

System.err.println("\nTRY AGAIN");

}

}

}

public class BankApp {

public static void main(String[] args) throws ClassNotFoundException, SQLException {

Access ac = new Access();

ac.access();

}

}

In Second package-

package com.iNeuron;

import java.sql.SQLException;

import java.util.Scanner;

public class ShowView {

Integer select;

Scanner sc;

void display() throws SQLException {

ShowView svv = new ShowView();

System.out.println("\nATM Interface");

System.out.println(

"Enter 1 to see Transaction History \nEnter 2 for withdraw money \nEnter 3 for Depositing\nEnter 4 For Transfer\nEnter 5 for Exit");

sc = new Scanner(System.in);

select = sc.nextInt();

switch (select) {

case 1:

new Transaction().history();

break;

case 2:

new Withdraw().withdraw();

break;

case 3:

Depositing dep = new Depositing();

dep.deposit();

break;

case 4:

new Transfer().transfer();

break;

case 5:

System.out.println("Exit");

break;

default: {

System.err.println("Choose Valid number");

svv.display();

break;

}

}

}

}

In third package-

package com.iNeuron;

import java.sql.SQLException;

import java.util.Scanner;

public class ActionOnAccount {

static int depoIn = 0;

static int transIn = 0;

static int withIn = 0;

public static int data = 5000;

public Scanner scan = new Scanner(System.in);

public int money = 0;

public String decide = null;

void furtherWork() throws SQLException {

System.out.println("YOu have other work to do :: yes/no");

decide = scan.next();

if (decide.equalsIgnoreCase("yes")) {

ShowView svv = new ShowView();

svv.display();

} else {

System.out.println("Account get log off ");

}

}

void viewData() {

System.out.println("Are you want to see data:: [yes/no]");

decide = scan.next();

if (decide.equalsIgnoreCase("yes")) {

System.out.println(data);

}

}

}

class Transaction {

void history() {

Depositing d= new Depositing();

int val1=d.depoIn;

System.out.println("The amount deposite in last few times "+val1);

Transfer t= new Transfer();

int val2=t.transIn;

System.out.println("The amount Transfer in last few times "+val2);

Withdraw w= new Withdraw();

int val = w.withIn;

System.out.println("The amount Withdraw in last few times "+val);

System.out.println("\n");

ActionOnAccount aa= new ActionOnAccount();

try {

aa.furtherWork();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

class Withdraw extends ActionOnAccount {

void withdraw() throws SQLException {

System.out.println("\nWITHDRAW SECTION ");

System.out.println("Enter amount for withdrawing :: ");

money = scan.nextInt();

data = data - money;

System.out.println("Collect Money.....");

withIn = withIn+money;

viewData();

System.out.println("\n");

furtherWork();

}

}

class Depositing extends ActionOnAccount {

void deposit() throws SQLException {

System.out.println("MONEY DEPOSITE SECTION");

DbConnector dbb = new DbConnector();

System.out.println("Enter Amount to deposite ");

money = scan.nextInt();

data = data + money;

System.out.println("Your Money Succesfully Deposite ");

depoIn = depoIn + money;

viewData();

System.out.println("\n");

furtherWork();

}

}

class Transfer extends ActionOnAccount {

String accountName = null;

void transfer() throws SQLException {

System.out.println("TRANSFER SECTION \nEnter Money for Transfering");

money = scan.nextInt();

System.out.println("Enter Receiver Account Number");

accountName = scan.next();

data = data - money;

System.out.println("Data tranfer successfully in " + accountName + " account");

transIn = transIn + money;

viewData();

System.out.println("\n");

furtherWork();

}

}

What is Java used for in banking?

Java's components are designed to **restrict data access and provide a number of memory safety features that help mitigate vulnerabilities caused by common programming mistakes**. Java banking applications are reliable and that's the first and foremost reason why banks choose Java.

THANK YOU