**<Bank Management System>**

****

Prepared by: Souhail Madani

Supervised by: Jorge Martinez

Language Used: Java

**Abstract**

This project aims to fulfill the requirements of customers that use online banking for payment. The design and development of this project provides a secured approach in managing bank customer’s information. Any organization needs to manage customer requirements smoothly in order to be successful. Such acts help to strengthen the relationships between banks and their customers. Bank Management System has been designed to perform the tasks smoothly and easily which cannot be accomplished through any manual system efficiently. This project has interesting features like Creating New Account, Depositing and Withdrawing money, Displaying and managing account holders data, Transferring cash, Deleting all records and the Help section. The programming language used to develop this project is Java.

**Table of Contents**

1. [Introduction: 4](#_TOC_250014)
2. [Problem Statement (Existing System) 4](#_TOC_250013)
3. [Solutions of problems by Proposed System 5](#_TOC_250012)
4. [Functional and Non-Fonctional Requirements 6](#_TOC_250011)
5. [Modules: 7](#_TOC_250010)
   1. [Module 1: Create an account 7](#_TOC_250009)
   2. [Module 2: Deposit money 7](#_TOC_250008)
   3. [Module 3: Withdraw money 8](#_TOC_250007)
   4. [Module 4: View all accounts 8](#_TOC_250006)
6. [System Limitations: 9](#_TOC_250001)
7. [Design 10](#_TOC_250000)
8. Coding 18
9. Implementation 30

# Introduction

Banking activities are in fact considered to be of great importance to the national economy. Without these services, trading or business activities etc. cannot be carried on smoothly and efficiently. Banks are the distributors and main protectors of liquid capital which is of vital significance to a developing country and even to a developed country. Efficient administration of the banking system helps in the economic growth of the nation as it is also useful to trade and commerce. During the last couple decades, bank operations are being performed electronically, instead of manually and this is online banking.

Online banking allows the customers of a financial institution to conduct financial transactions on a secured website operated by the institution. It is an umbrella term for the process by which a customer may perform banking transactions electronically without visiting the institution personally. It is an activity that is now not new to banks or their customers. Banks have been providing their services to the customers electronically for years through software programs to make their work much more efficient. Such software programs allow the users to access the banks directly.

In the past, banks were reluctant to provide their customers with online banking due to security concerns but now, such concerns are no more. A computer based management system is designed to make operations much more efficient and to allow the users to access the features of a bank without making any personal trips. We also aspire to ease the banking operations through our management system. Our project performs all essential functions of a bank quite efficiently and organizes the user’s data with great care.

# Problem Statement (Existing System)

The Bank Management System is an application that is used for maintaining a person’s account in a bank. The system should provide access to the costumer to features such as creating an account, depositing/withdrawing the cash from his account etc. Systems that we stumbled upon had quite a lot of problems such as:

* Cannot create multiple accounts at the same time.

*No help section to provide contact information of head office of the organization.

* Some systems could create multiple accounts at the same time but the number was fixed.

* Some identified users on basis of their names. If two users have the same name, there were confusions and concerns about which account belonged to which user.

** Customer’s information could not be saved in a master file. No option available to delete all records at the same time.

* No option available to view all records saved in the system.

*Interface was not user friendly.

The above mentioned problems could prove be quite deadly to the organization and customer relationship. The motivation to create a banking management system came from the desire to counter these problems in order to make online banking more efficient for the customers.

# Solutions of problems by Proposed System

The purpose of our proposed system is to provide a platform which can help the users to access and perform the banking operations easily. This system will be extremely beneficial for the customers who have the intention to make all their transactions online. They will get various benefits in the field of management of accounts on a clean and user-friendly platform. Our system has the following benefits:

*It provides flexibility for secure and safe transaction.

*Using our system will reduce manual power needed to manage records in organizations.

**A computer system works way more accurately than manual labor. It has a faster performance as compared to manual power.

*It is User ID protected and is a relatively safe system.

*The interface is user friendly and clear enough so users can easily understand the system to make various transactions.

*The record entered is saved in a master file and can be viewed inside the application or in the form of a txt file.

**Complex banking transactions are efficiently handled. It is cost effective.

*Complex Banking operations and Transaction operations are efficiently handled by our system.

The above mentioned points are implemented in our application by the help of the programming language .

# Project Scope

# 4.2: Functional Requirements:

1. Add Account: The user must be able to add an account using his personal informations
2. Type of Account: Our bank will afford three types of accounts. Current Acount, Save Account, and Student Account.
3. Save Account must have a maximum withdraw limit
4. Current Account: must have a Trade Liscence Number and the balance must be greater than 5000
5. Student Account: must have institution name
6. Deposit Account: The user must be able to add money to his account by affording the generated id given after opening of account.The condition of save account must be respected.
7. Withdraw Accoint: The user can withdraw with his id. The balance must be checked simultaneously.

# 4.1. Non-functional requirements:

1. Computer with an i5 with 8G RAM running under Windows 10 with a hard disk of 200GB Windows 7 compatible software
2. Simplified and practical user interface
3. Main language: French

Summary:

This project intends to introduce more user friendliness in all the banking transactions. All modules of the system have been made quite simple but only authorized personnel can access the records, the authorized personnel being the customers and administrators of the system. The records are up to date and are saved in master file. Accounts can be created per user demand. Users can also withdraw money, deposit money and apply for a loan. A help section has also been designed so that if users want, they can have the necessary information to contact the head office for more information on tricky matters.

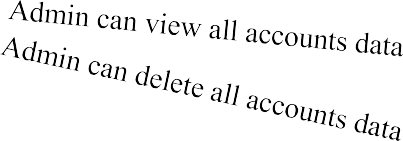
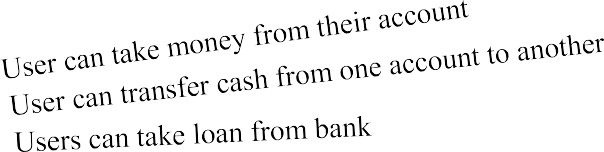
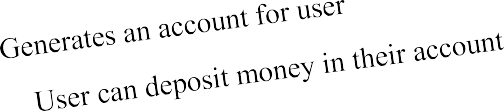
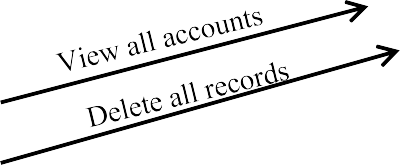
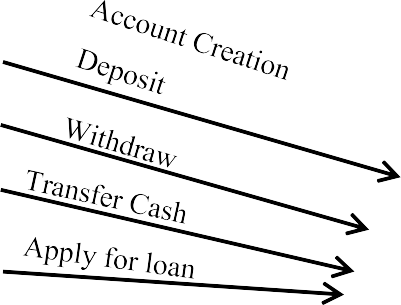
FEASIBILITY ANALYSIS:

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

Development of this application is highly economically feasible. The organization needed not spend much m one for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so , we can attain the maximum usability of the corresponding resources .Even after the development , the organization will not be in a condition to invest more in the organization .Therefore , the system is economically feasible.

* + **evel 0 dfd**



Help

Bank Management

System

Contact info of main branch

A D M I N

C U S T O M E R

# Modules

## Module 1: Create an account

### Features:

* + - In this module, features regarding account creation will be handled.
    - New users can easily register for our system by filling in the form.
    - After filling in, they can also see their information being displayed.
    - Multiple accounts can be created at one time, depending on the user input.

## Module 2: Deposit money

### Features:

* + - Users will be asked to input their account number.
    - Only if the account exists can they deposit their money.
    - No limit on depositing cash.

## Module 3: Withdraw money

### Features:

* + - Users will be asked to input their account number.
    - Only if the account exists can they withdraw their money.
    - There is a limit on withdrawing money.
    - Users can withdraw only more than 500Rs and less than 25000Rs at one time.

## Module 4: View all accounts

### Features:

* + - A pin has to be entered in order to access this module.
    - Pin is set so that only the system administrator is able to access this feature.
    - All data of users will be stored in a master file (txt file).
    - By help of this module, that information can be viewed inside the system in an organized format.

## Module 8: Help

In this module, the contact number, email and address of the head office of the organization is given.

1. **System Design:**

6.1. ER Diagram:

Diagram

Description automatically generated

6.2: Use Case Diagram:

Diagram

Description automatically generated

6.3: Sequence Diagram:  
Table

Description automatically generated

Assumptions

- In this bank it is allowed for loans, accounts and cards to have multiple customers associated with them

- An account, card, loan, can have many transactions, but a transaction can only be associated with one of these each o It is the same idea as before for vendor, bank and teller

- Total participation is expected for a lot of specializations in some relationship sets

- There are no weak entity sets that could identified as you cannot really delete financial information without destroying a financial trail that government agencies may need

- Some attributes such as name in customer have not been broken down into more complex attributes but could be in the future if identified as a need. It was not seen a necessary for the iteration based on the interface demands

- Customers are expected to know their customer ID, their account ID, loan ID if they have a loan, and all their card information. If they forget one of these, they will have to contact a bank personnel and confirm their identity to gain access back. For cards they can cancel and then create a new card through the interface.

- If a customer sees that false transactions are being made on their behalf, they will have to contact bank personnel, verify their identity and identify the false transactions made.

- Since most transactions happen through the Banks created interface, we do not expect a customer to create a password, instead they are expected to not share their ANY of their ID information to prevent fraud. Once a system is made to operate on a website, then that field will be necessary, and you will have to encrypt and store them for protection.

- Currently we have the amount fields, balance fields and other fields regarding monetary value set as numbers with a decimal limit of two. This was decided for simplicity and the fact that Nickel Bank will have a quite small number of customers at the start, meaning that any floating-point errors will cause minimal issues in the bank. However, as the bank size increases, it could be a good idea to separate the dollars and cents amounts to prevent floating point errors.

- Payment was used for loan\_payment and credit\_payment as they fundamentally have the exact same behavior, just paying off different items

6.4. Design Patterns:

I used two design patterrns for solving existing problems in this project. These design patterns are Observer Design Pattern and Singleton Design Pattern.

Firstly, a user has only one bank account, and only performs operations of money through this bank account. Therefore, it is sufficient to produce only 1 object from the BankAccount class for 1 user. For this reason, the BankAccount class is defined as Singleton Class.

Text

Description automatically generated

Finally, when the user makes a withdrawal or investment, the user is informed via SMS and e-mail. The amount of money the user trades and the amount of money in the bank make up this information. Because of this, SMS and e-mail systems that send information of transactions are considered as Observers. The BankAccount class is also considered Subject. Every transaction made in BankAccount is notified to SMSNotifier and EmailNotifier classes and these classes do the necessary processing.

# System Limitations

Following are some of the constraints and limitations of the system:

*The system is a console based application and does not have a graphical user interface.

*A java IDE is required to run this system.

# Design

## Case 1: Create an account

**Start**

Select an option

(1-5)

**Note:**



Display Case 1:

Create Account

Input n

(Number of accounts to create)

For

i<=n

Enter Customer Details

Display Customer Details

**Stop**

i++

In all the flowcharts, stop means that a particular task has been performed and the user will return to the main menu, with the

exception being case 9, where user will totally exit the system.

* + **Case 2: Deposit**

**Start**

Select an option (1-5)



Display Case 2:

Deposit

Input Account Number

False

If user

exists

True

Display User has been

successfully found!

Input amount

Display Amount has

been deposited

Display Error! User

does not exist!

**Stop**

* + **Case 3: Withdraw**

**Start**

Select an option (1-5)

False



Display Case 3:

Withdraw

Input Account Number

False

If user

exists

True

Display User has been

successfully found!

Input amount (z)

Display transaction

not successful

If z<25000

&& z>500

Display Error! User

does not exist!

True

Display withdraw

successful

**Stop**

* + - **Case 4: View all accounts**

**Start**

Select an option (1-5)

Display Case 4:

View all accounts

Input pin

False

If

pin==1234

True

Displaying

Records

Display Error!

**Stop**

* **Case 9: Exit**



**Start**

Select an option(1-5)

Display Case 9: Exit



Display Thanks for using our Bank Management System

### End

### This is where the program will truly stop

**Implementation**

* + **Main Menu**
  + Graphical user interface

    Description automatically generated
* **Case 1: Create an account**
* Graphical user interface, application

  Description automatically generated

Graphical user interface, application

Description automatically generated

* **Case 2: Deposit**

Graphical user interface, application

Description automatically generated

* **Case 3: Withdraw**

Graphical user interface

Description automatically generated

* **Case 4: View all accounts**

Graphical user interface, text, application

Description automatically generated