BANK MANAGEMENT SYSTEM

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by

Arpita Vaish
(University Roll No: 2200290140038)
Arpit Vishwakarma
(University Roll No: 2200290140037)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr Praveen Kumar Gupta
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(JANUARY 2024)

CERTIFICATE

Certified that **Arpita Vaish (2200290140038), Arpit Vishwakarma(2200290140037)** has/ have carried out the project work having "**Bank Management System**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Arpita Vaish (2200290140038)
Arpit Vishwakarma (2200290140037)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr Praveen Kumar Gupta	**Dr. Arun Tripathi**
Assistant Professor	**Head**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of Institutions, Ghaziabad	**KIET Group of Institutions, Ghaziabad**

ABSTRACT

The Bank Management System is a comprehensive software solution designed to streamline and automate various banking processes, enhancing efficiency and customer service within a financial institution. This system encompasses a range of modules, including customer management, account administration, transaction processing, and report generation.

Key features of the Bank Management System include user-friendly interfaces for both customers and bank staff, secure authentication mechanisms, and robust data management capabilities. Customers can access their accounts, perform transactions, and view account statements through an intuitive online portal. Bank administrators benefit from centralized control over customer information, account settings, and financial transactions, leading to improved decision-making.

The system incorporates advanced security measures to protect sensitive financial data, ensuring compliance with regulatory standards. It also supports multi-level user access, allowing different roles within the bank to perform their specific functions securely. Additionally, the system generates detailed reports on transactions, account balances, and other critical metrics, aiding in auditing, analysis, and strategic planning.

In summary, the Bank Management System offers a technologically advanced and secure platform for banks to manage their operations efficiently, enhance customer satisfaction, and adapt to the evolving landscape of modern banking.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Mr. Praveen Kumar Gupta** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Arpita Vaish (2200290140038)

Arpit Vishwakarma(2200290140037)

TABLE OF CONTENT

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Overview

The “Bank cum ATM simulation System” project is a model of ATM system. Automated
Teller Machine enables the clients of a bank to have access to their account without going to the bank. This is achieved only by developing the application using online or programming concepts. This project helps the customers to perform the basic ATM functionalities with ease.

When the project is implemented, the user who uses this product will be able to see all the basic functionalities and services provided by an ATM, when he/she enters the necessary options and arguments. The product provides services like request for pin change, cash deposit, and other requirements of the user. The data is stored in database and is retrieved as and when required. The projects implementation needs hardware to operate or simulated conditions can also be used to successfully use the developed product.

The project is designed in such a way that the user has to enter card number and PIN if already signed up otherwise, sign up first then only will get the card number and PIN number. Once verified, the customer will be provided with a menu and they have to select the options provided in the menu. For example, if the customer wants to withdraw amount from his/her account then they have to select the option and then have to enter the amount which they want to withdraw.

This Bank cum ATM simulation project is entirely based on JAVA programming language for frontend using its Swing and AWT features and My SQL Workbench for connectivity through backend.

The bank management system project is a program that keeps track of a client’s bank account. This project demonstrates the operation of a banking account system and covers the essential functions of bank management software. It develops a project for resolving a customer’s financial applications in a banking environment to meet the needs of an end banking user by providing multiple ways to complete banking chores. Additionally, this project is to provide additional features to the user’s workspace that are not available in a traditional banking project. The project’s bank management system is built on cutting-edge technologies. This project’s main goal is to create software for a bank account management system. This project was designed to make it simple and quick to complete previously impossible processes with manual systems which are now possible with this software. In this project we are using Java programming language for coding purposes and MySQL for database management. The database has account table, customer table and admin table.

1.2 Need

Millions of times per day around the globe people are instantly withdrawing money at automatic teller machines (ATMs). Given the fast pace of the world today, it is not surprising that the demand for access to quick cash is so immense. The power of ATMs would not be possible without secure connections. The final act of ATM dispensing cash is the result of an amazingly fast burst of the customer never sees, but a trust is being done in a confidential manner. Nowadays everyone is busy in their day-to-day life, so they feel that the job or the work of doing transactions must be easier therefore, the system is used to reduce their work which is done in the ATM system. Instead of keeping lots of paper into the record or file and it may be missed somewhere so, this system helps to keep the record the customer it also keeps the details of the customer. It is also easy to access.

1.3 Project Description

The Bank Management System project aims to create a robust and secure software solution for banking institutions. It includes features such as user authentication, customer and account management, transaction processing, and a user-friendly online portal. The system emphasizes security measures, compliance with industry standards, and detailed reporting for informed decision-making. The expected benefits include improved operational efficiency, enhanced customer experience, increased security, and streamlined account and transaction management.

Now a days every one very busy in their work. So they feel that the job must be easier so the system is used to reduce their work which is done in the ATM system. Instead of keeping lots of paper into a record or file and it may be missed somewhere so, this system helps to keep the record of the customer it also keeps the details of the customer. It is also easy to access to access the bank hassle free.

1.4 Proposed System

In this project we are going to explain about **Banking Management System.**
This project has facility to opening account, depositing and withdrawing money. The proposed system is a computerized one. This has greater accuracy and efficiency. This takes only limited time for calculation.
The proposed system can be used to maintain efficiently the BANKING schedule. In larger organizations employees are large. At that time also the proposed system is useful and helpful. The system includes users Administrator (HR) level.
In this project we have admin login feature, we want to fill the admin and password then we enter to home page, The home page facility show all the include features about this banking project. we have a number of options like modify account, add record, modify customer record, deposit money record, withdrawal money record. We can easily choose any option according to our own requirement. We also have feature of validation.

1.5 Features Of This System

User Authentication and Authorization:

- Secure login mechanisms for customers.

- Multi-level access controls to protect sensitive data.

2. Customer Management:

- Comprehensive customer profiles with personal and account details.

- CRUD operations for efficient customer data management.

4. Transaction Processing:

- Real-time processing of deposits, withdrawals, and transfers.

5. Online Banking Portal:

- User-friendly interfaces for seamless online access.

- Transaction history, fund transfer, and bill payment functionalities.

6. Reporting and Analytics:

- Generation of detailed reports for auditing and analysis.

- Customizable reports for regulatory compliance.

7. Security Measures:

- Encryption for secure data transmission and storage.

- Regular security audits and updates.

8. Audit Trail:

- Logging and tracking of all system activities.

- Accessible audit trail for monitoring user actions.

9. Technology Stack:

- Java or Python for backend development.

- Relational database (e.g., MySQL, Oracle).

10. Compliance:

- Adherence to industry security standards.

- Regulatory compliance for banking operations.

12. User Interface (UI):

- Intuitive and responsive UI design.

- Consistent user experience across devices.

In summary, these key features collectively ensure a comprehensive and efficient Bank Management System, meeting the operational and security needs of modern banking institutions.

1.6 Project Scope

The project aims to develop a comprehensive Bank Management System with the following scope:

1. User Management:
- Secure user authentication and authorization.
- Different access levels for customers and staff.

2. Customer and Account Management:
   - Create, update, and delete customer records.
   - Manage various account types with real-time status tracking.

3. Transaction Processing:
   - Real-time processing of financial transactions.
   - Integration for inter-bank transactions.

4. Online Banking Portal:
   - User-friendly interface for customers.
   - Features include transaction history, deposit account, withdraw amount etc.

5. Reporting and Analytics:
   - Generate detailed reports for auditing and analysis.
   - Customizable reports for regulatory compliance.

6. Audit Trail:
   - Logging and tracking of all system activities.
   - Accessible audit trail for monitoring user actions.

7. Technology Stack:
   - Backend development using My SQL.
   - Relational database for data storage.
   - Frontend development using Java.

8. Compliance and Scalability:
   - Adherence to industry security standards and regulatory requirements.
   - Scalable architecture to accommodate future growth.

9. User Interface (UI):
   - Intuitive and responsive UI design.
   - Consistent user experience across devices.

The project scope aims to deliver a robust and secure Bank Management System that enhances operational efficiency, ensures data integrity, and provides a positive user experience for both customers and staff.

CHAPTER 2

Literature Review

An automated or automatic teller machine is a device that provides the clients of financial institutions with access to money transactions in a public space without the need of bank customers. The modern ATMs, the customers are identified by inserting an ATM card with a chip that contains a unique card number and some data such as an expiration date, CVV (Card Value Verification) code and customer name. Customers identification and authentication is done by customer providing a personal identification number (PIN). Using an ATM system, customers can access their bank accounts in order to make money withdrawals, debit card and check their accounts balance.

2.1 Literature Review

Literature review is an expressive study based on the detailed review of earlier pertinent studies related to the various concepts of online shopping to discover the concept of online shopping. It highlights the status of online shopping, importance and problems of online shopping, factors affecting online shopping and a critical review of the privacy and security issues in online shopping.

2.1.1 Working Mechanism

An ATM device is used for getting the cash without the presence of bank clerk or employer. The customer will be identified by a card which contains 16-digit card number. When the customer inserts the card inside the ATM machine card number will be read and it will ask the customer for the secret PIN and the type of transaction the user wants to perform. The card number and secret PIN will be encrypted and send to the server. At the server side the card number and secret PIN will be decrypted and if all the details matched with the particular customer, then the customer is said to be authentic otherwise the transaction will be debited.

2.1.2 Issues in ATM Machine

Currently the triple DES algorithm is used at the client side for the encryption of the PIN which will be transmitted to the server. The algorithm is sluggish in software's and it is very slow. The man-in-the-middle attacks where the criminals can attach the fake keypads for the existing ATM machines. This fake key pad can be used to read the PIN of the customer. Using this PIN, the transaction password which will be used for the online transaction can be stolen by the attacker. The attacker may fix the cameras in the place of the ATM machine. Using this camera, he might be able to read the PIN no of the customer.

AES (Advanced Encryption Standard) is the one of the strong and secure algorithms used for the encryption and as well as decryption. AES is a block cipher with a block length of 128 bits. AES allows for three different key lengths: 128, 192, or 256 bits. Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical.

It uses a massive combination of keys therefore the encrypted code is not easy to crack. The algorithm performs well in almost all the software's as well as hardware devices. Instead of Triple DES (Data Encryption Standard) which is currently use the AES algorithm can be used.

2.1.3 Security Requirements in ATM System

A. Authorization: A transaction must always be authorized by the payer and needs payer authentication (physical, PIN, or digital signature). A payment may also need to be authorized by the bank.

B. Data confidentiality and authenticity: The transaction data should be authentic and external parties should not have access to data. And some data need to be hidden even from participants of the transaction in ATM place.

C. Availability and reliability: A transaction infrastructure should always be available and centralized systems should be designed with care.

D. Privacy: The customers should be able to control how their personal data is used by the other parties. The ATM transaction should provide the privacy of the data

2.1.4 Future Work

The security of the ATM system can be further increased by the encrypting the 3D secure password and with the introduction of the OTP (One Time Password) feature. The various methods such as visual cryptography and steganography can be used to hide the customer data. An algorithm called Advanced

Encryption Algorithm (AES) can be used since it provides the strong encryption of the PIN number.

2.4 Literature Survey

The purpose of this study is to empirically investigate the dimensions and drivers of entrepreneurial perceptions in the pursuit of emerging e-business opportunities for traditional (or offline) firms. This study introduces the subjectivist theory of entrepreneurship into the IS research context, and identifies three dimensions that make up entrepreneurial perceptions: collaboration perception, planning perception, and operation perception. The authors tested the proposed research model using structural equation modelling (SEM) with survey data collected from 203 firms in China. Results reveal that external pressures and IT infrastructure maturity are positively and significantly related to entrepreneurial perceptions. The results also suggest that IT infrastructure maturity has stronger effects on collaboration perception and planning perception than external pressures. This paper provides clear guidance for entrepreneurs to understand the three entrepreneurial perceptions for emerging e-business opportunity discovery and the driving forces to the entrepreneurial perceptions.

In, Information and communication technology (ICT) has helped to drive increasingly intense global Competition. In the world history the most of the countries are most developed because of they are financially very clear for how to use the high amount of money in the developing process in own country. We also use the SOA architecture for providing the scalable and reliable service therefor we studied related to the SOA architecture to know how we use to implementation process in our project using Service Oriented Architectures (SOA). We also refer the paper who give the case study information about Scandinavian bank and a Swiss bank these two banks are working on the basis of service-oriented architecture for providing the service for the customer. SOA provides potential for greater organizational agility (and thereby competitiveness). In , in the second paper we learn which type of problems are created in banking system during the different types of transactions. Here discuss about if any region the transaction may be fail then how to avoid it and fixed it. We also studied about Firms in Italy defaulted more against banks with high levels of past losses. This `selective' default increases where legal enforcement is weak. Poor enforcement thus can create a systematic transaction risk by encouraging banking users to defaulted masse once the continuation value of their bank relationships comes into doubt. In banking-sector the security also must and when we talk about money or property this case is more sensational then we found the security is the major thing to do in banking system.

CHAPTER 3

FEASIBILITY STUDY

The feasibility study for the proposed Bank Management System reveals promising outcomes across technical, economic, operational, and scheduling dimensions. From a technical standpoint, the required software, hardware, and expertise are readily available, facilitating seamless development and integration. Economic feasibility is supported by a favorable cost-benefit analysis, showcasing a reasonable return on investment over time. Operationally, the study indicates a positive outlook, with users displaying openness to the system, and its alignment with organizational goals is evident. The scheduling feasibility is reinforced by a well-structured project timeline, accounting for dependencies and potential risks. The project's regulatory and legal aspects are deemed feasible, with a commitment to compliance and data privacy measures. The system exhibits scalability and flexibility, ensuring adaptability to future demands. Overall, the feasibility study indicates a strong foundation for the successful development and implementation of the Bank Management System.

3.1 Technical Feasibility

Technical Feasibility (TF) analysis in software development can be carried out to verify this hypothesis and learn more about the potential outcomes of the proposed project. This applies across all sectors, providing a brighter future for software development.

The technical feasibility study in software engineering is conducted in various ways depending on the company. Some people may do it in a precise and organized method, while others may do it as needed. However, you must have the following resources -

- Hardware

- Software
- Technology
- Skills and knowledge
- Time and budget for development
- Specialists
- Software development tools

3.1.1 Hardware Feasibility

- Processor must be i3 or more
- RAM must be 4GB or more
- Windows must be of latest version i.e. Windows 11 or 10.

3.1.2 Software Feasibility

- For frontend development, Eclipse IDE or IntelliJ Idea platform.
- For backend development, My SQL.

Since the above hardware requirements are available to me and my team and the software, we have used for frontend development, which has been developed using JAVA programming language, we have used IntelliJ Idea because me and my team member were comfortable in using that software tool for developing the project.

JAVA being the platform independent language to generate the user-friendly software system and My SQL as a backend database system, this will facilitate user in operating the system successfully.

A platform is the hardware or software environment in which a program runs. The java platform differs from most other platforms in that it's an only software platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

3.2 Economic Feasibility

The economic viability of a project considers both the costs and potential returns. Therefore, making a ROM (Rough Order of Magnitude) estimate is normal practice to ascertain financial viability. In the Economic Feasibility study cost and benefit of the project are analysed. This means under this feasibility study a detailed analysis is carried out will be the cost of the project for development which includes all required costs final development hardware and software resources required, design and development costs and operational costs, and so on. After that, it is analysed whether the project will be beneficial in terms of finance for the organization or not. we find the total cost and benefit of the proposed system over the current system. For this project, the main cost is the documentation cost.

3.3 Legal Feasibility

Considering the product's legal viability ensures it won't get you in trouble. For instance, HIPAA compliance is required for any medical software that handles PHI (Protected Health Information). In addition, you must investigate the potential legal threats to your project and how best to mitigate them.

3.4 Operational Feasibility

Implementing a project within the current business environment might impact daily operations. Operational feasibility involves analysing the practicality of implementing the project within the current business environment and determining how it will impact daily operations. This assessment involves undertaking a study to analyse and determine whether and how well the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development. It is the ease and simplicity of operation of the proposed system. The system does not require any special skill set for users to operate it. This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to be carried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

3.5 Behavioural Feasibility

It evaluates and estimates the user attitude or behaviour toward the development of the new system. It helps in determining if the system requires special effort to educate, retrain, transfer, and change an employee's job status on new ways of conducting business. Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need for online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

3.6 Risk Analysis

Identifies potential risks associated with the project and assesses the impact of those risks on project success. In this a risk mitigation plan is also developed to cope up with the upcoming risks.

CHAPTER 4

DESIGN

A properly designed database provides you with access to up-to-date, accurate information. Because a correct design is essential to achieving your goals in working with a database, investing the time required to learn the principles of good design makes sense. In the end, you are much more likely to end up with a database that meets your needs and can easily accommodate change. This article provides guidelines for planning a desktop database. You will learn how to decide what information you need, how to divide that information into the appropriate tables and columns, and how those tables relate to each other. You should read this article before you create your first desktop database. Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of an enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept for designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored. The main objectives behind database designing are to produce physical and logical design models of the proposed database system. To elaborate on this, the logical model is primarily concentrated on the requirements of data and the considerations must be made in terms of monolithic considerations hence the stored physical data must be stored independent of the physical conditions. On the other hand, the physical database design model includes a translation of the logical design model of the database by keeping control of physical media using hardware resources and software systems such as Database Management System (DBMS).

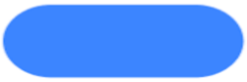
4.1 Flowchart

A flowchart is a graphical representation of an algorithm. Programmers often use it as a program- planning tool to solve a problem. It makes use of symbols that are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as "flowcharting".

4.1.1 Basic Symbols used in Flowchart Designs

4.1.1.1 Terminal:

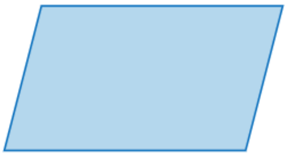
The oval symbol indicates Start, Stop, and Halt in a program's logic flow. A pause/halt is generally used in a program logic under some error conditions. The terminal is the first and last symbol in the flowchart.



Terminal

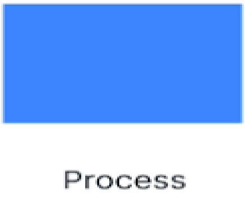
4.1.1.2 Input/Output:

A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelograms in a flowchart.



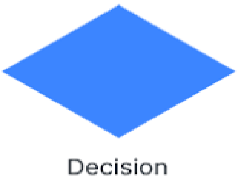
4.1.1.3 Processing:

A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication, and division are indicated by action or process symbol.



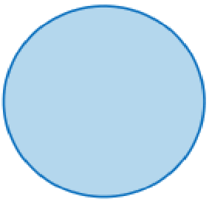
4.1.1.4 Decision:

Diamond symbol represents a decision point. Decision-based operations such as yes/no questions or true/false are indicated by diamonds in the flowchart.



4.1.1.5 Connectors:

Whenever the flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.



4.1.1.6 Flow lines:

Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of the flow of control and the relationship among different symbols of the flowchart.

4.1.2 Flowcharts for different modules

4.1.2.1 Signup Flowchart

In this, the flow goes from clicking on sign up button after which a form of details will be displayed and on clicking next a new form will again be displayed and on clicking again a form of additional details will be displayed and after filling all the forms a unique card number and a PIN will be generated by the system. The flowchart is as below:

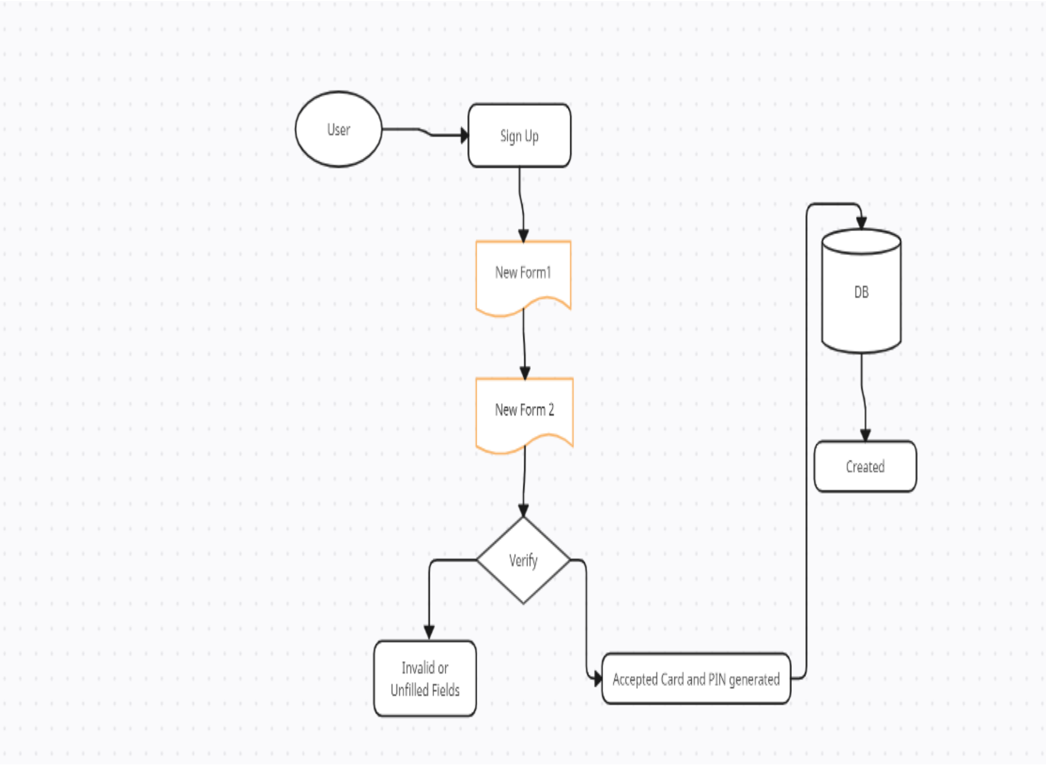


Fig 4.1

4.1.2.2 Deposit and withdraw Flowchart

In this, the flowchart shows the flow of deposit module and withdraw module. The flowchart is shown below:

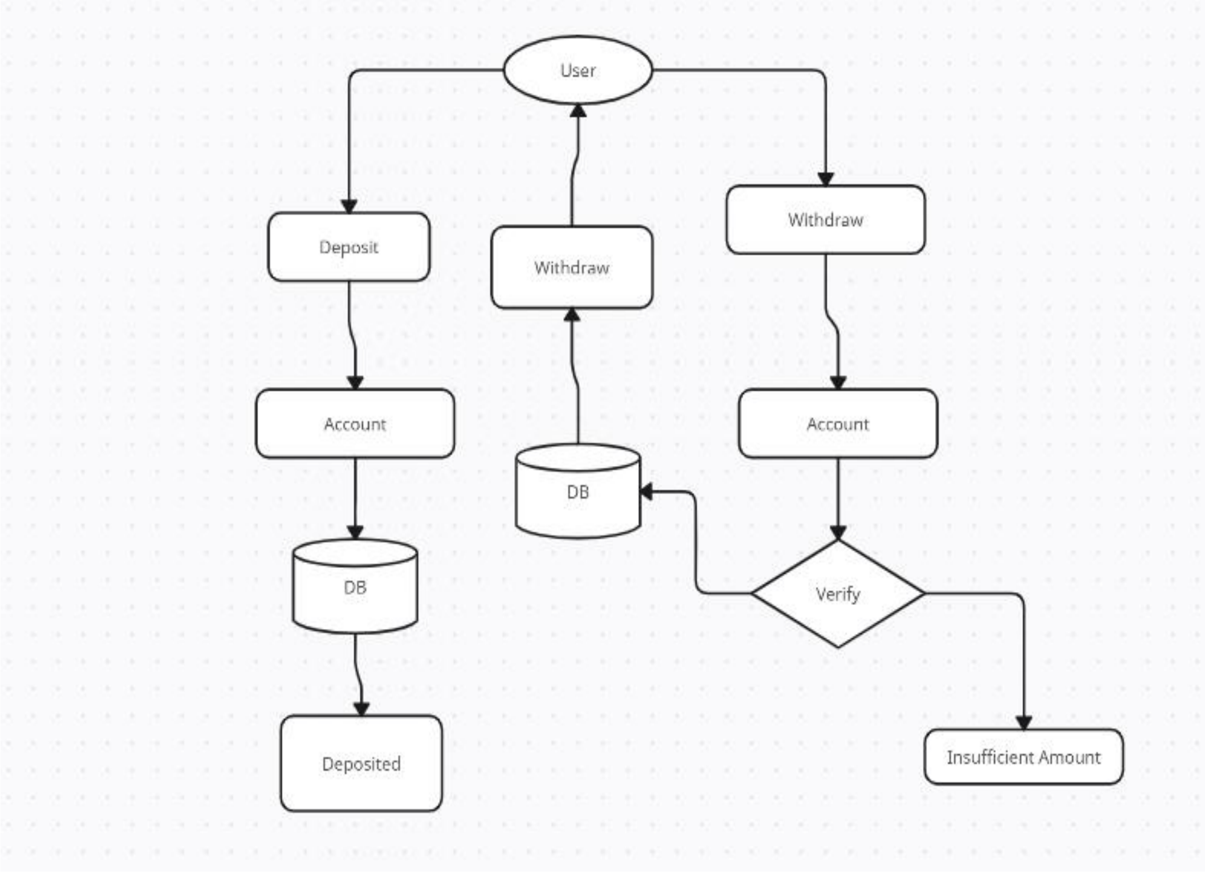


Fig 4.2

The above flowchart shows the deposition and withdrawal of amount to and from the customer's account. First, the user/customer selects the option for withdraw or deposit amount, after selection of the option from option menu in main page.

For depositing the amount the system will go to the account of the user and then deposits the amount into the database of the user's account.

For withdrawal of any amount firstly the system will go to the user's account database to check whether the entered amount is present in the user's account or not. If the entered amount is less or equal to the current amount then the amount will be withdrawn from the account otherwise an error will be displayed that insufficient amount.

4.2 Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. Use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

The purpose of a use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as the other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from the other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

Used to gather the requirements of a system. Used to get an outside view of a system. Identify the external and internal factors influencing the system. Show the interaction among the requirements actors.

4.2.1 Use case diagram components.

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

Actors: The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

System: A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

Goals: The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

4.2.2 Use case diagram symbols and notation.

The notation for a use case diagram is pretty straight forward and doesn't involve as many types of symbols as other UML diagrams.

Use cases: Horizontally shaped ovals that represent the different uses that a user might have.

Actors: Stick figures that represent the people actually employing the use cases. · Associations: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

System boundary boxes: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.

Packages: A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

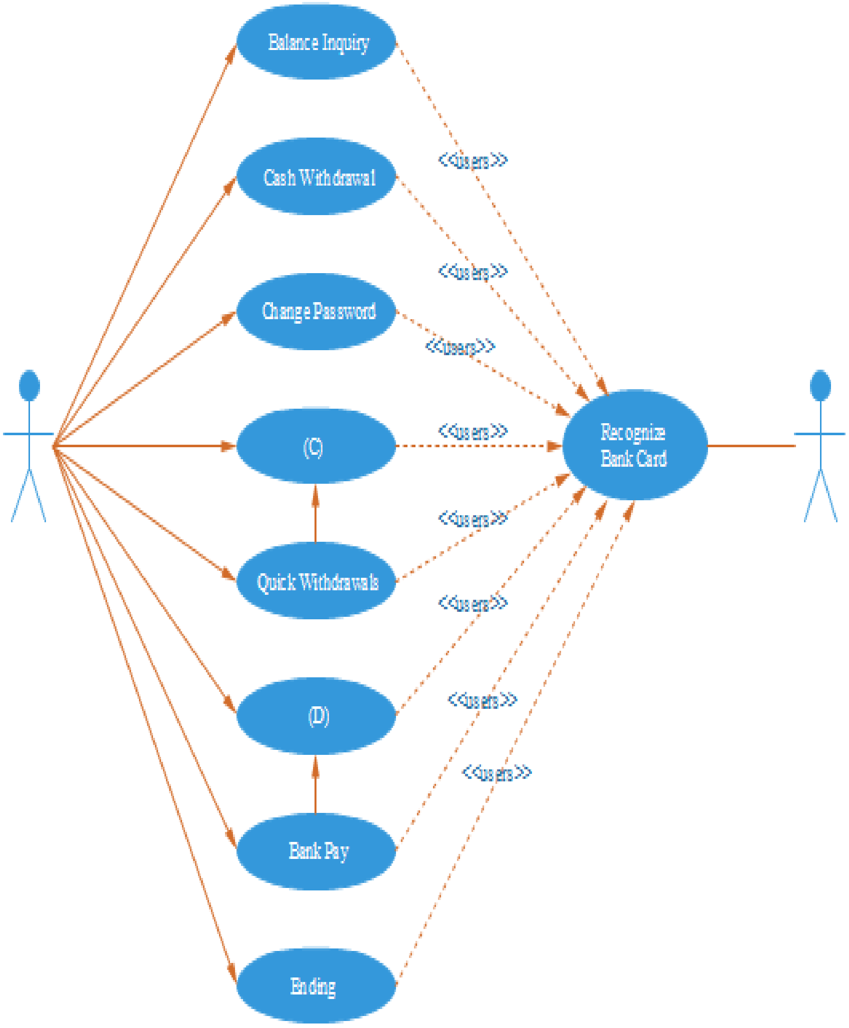


Fig 4.3

Above is the use case diagram for bank management system. The above diagram shows that the user can login to the system using their unique pin and card number. After logging in the user can select the options such as balance enquiry, mini statement, pin change, deposit and withdraw. The admin on the other hand can login and see the bank user's details.

4.3 Data Flow Diagram:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

4.3.1 DFD Levels:

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond.

- Level 0 DFD:

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analysed or modelled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

- Level 1 DFD:

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.

- Level 2 DFD:

  DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system’s functioning. It can be used to plan or record the specific/necessary detail about the system’s functioning.

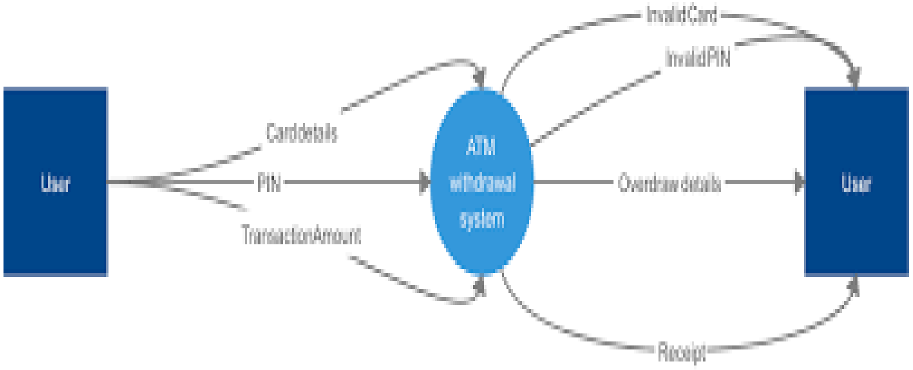


Fig 4.3.1 Context Diagram for bank cum ATM System

The above is the context-level or level-0 DFD for bank management system. This shows that the user will login to the system using their unique card number and PIN number. After logging in to the system the user will get into the main interface of the ATM system and can use the entire services of the system.

Level-1 DFD:

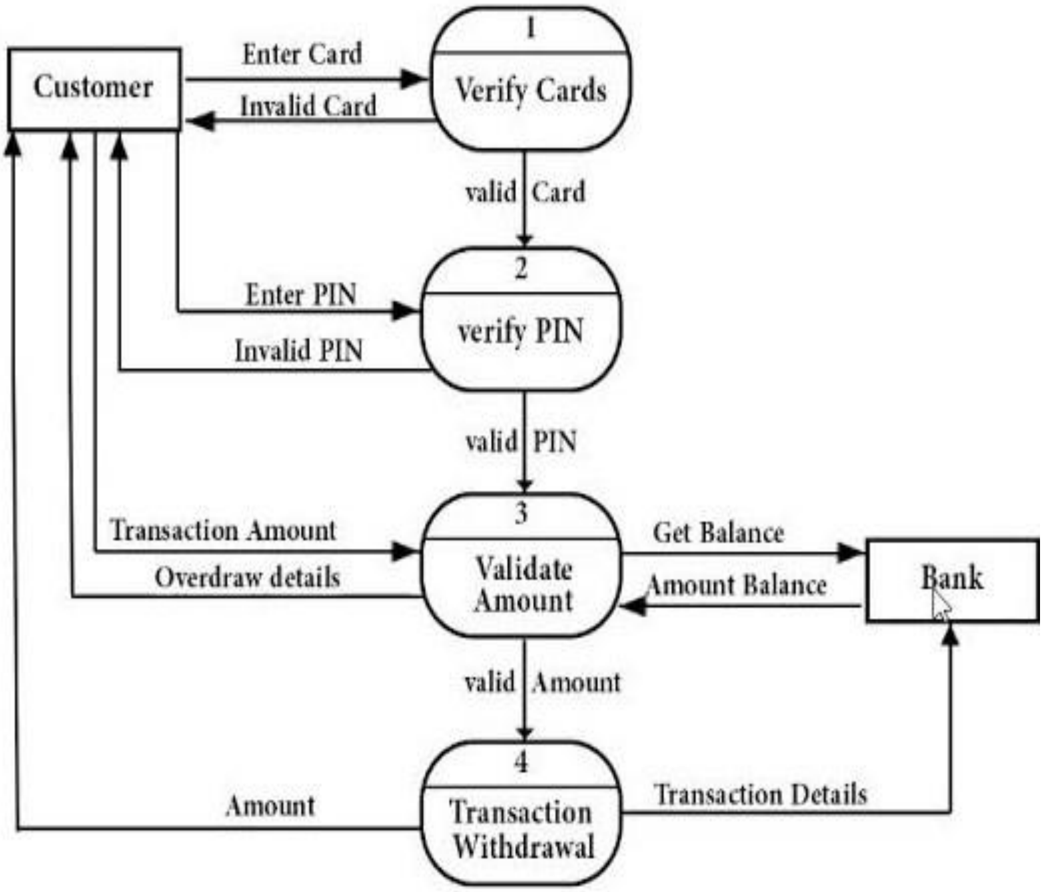


Fig 4.3.2 Level 1 DFD for bank management System

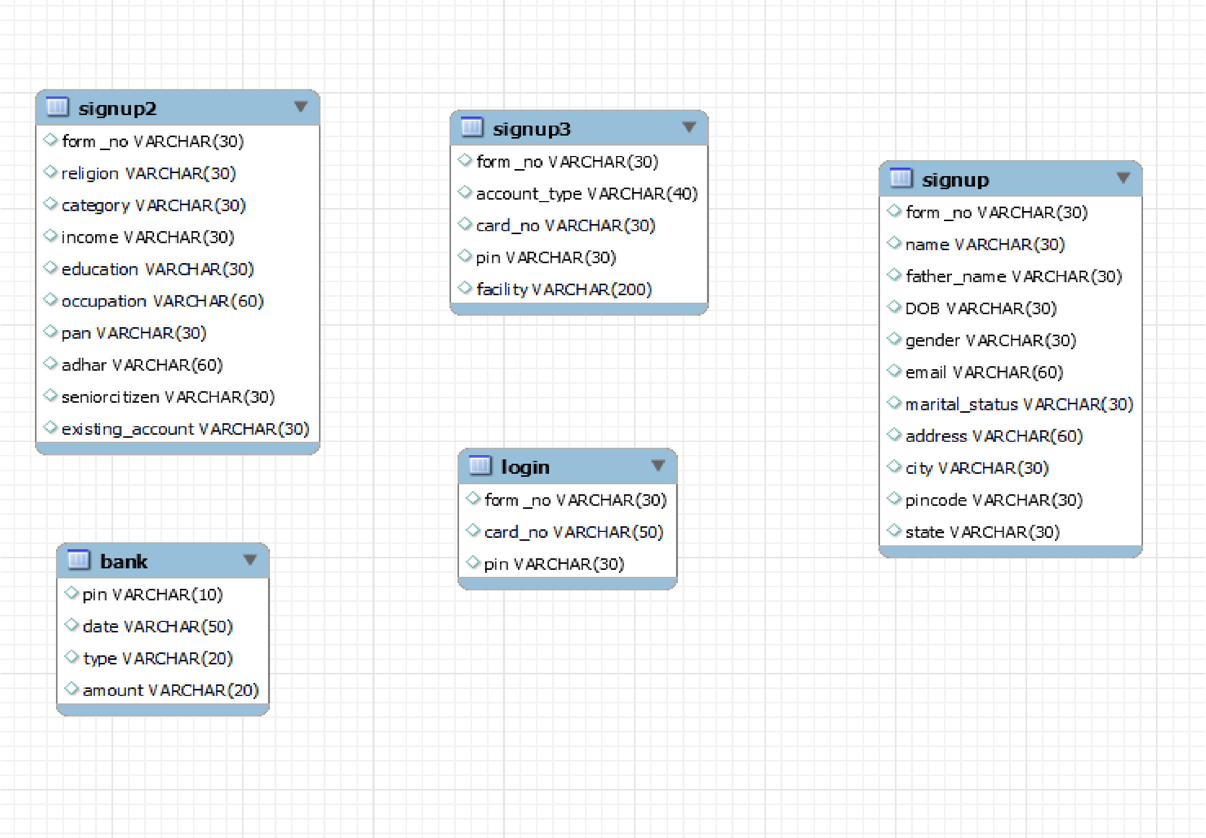
The above is the level-1 DFD for bank management system. In this the user enters the card number which is verified from the database if verified the user will be entered to the system else displayed invalid card or PIN.

The user will select withdraw option then the amount entered will be checked and withdrawn after verification.

The user will select the Pin change option from menu then the system will update the new password in the database if and only if both entered and re-entered password are same else error will be displayed.

4.4 Logical Database:

The logical database design is meant to describe the representation of the database in terms of its entities in form of tables and the existing relationships. Below is an illustration of the systems logical design as generated by the MYSQL workbench design tool.



signup2
form _no VARCHAR(30)
religion VARCHAR(30)
category VARCHAR(30)
income VARCHAR(30)
education VARCHAR(30)
occupation VARCHAR(60)
pan VARCHAR(30)
adhar VARCHAR(60)
seniorcitizen VARCHAR(30)
existing_account VARCHAR(30)

signup3
form _no VARCHAR(30)
account_type VARCHAR(40)
card_no VARCHAR(30)
pin VARCHAR(30)
facility VARCHAR(200)

signup
form _no VARCHAR(30)
name VARCHAR(30)
father_name VARCHAR(30)
DOB VARCHAR(30)
gender VARCHAR(30)
email VARCHAR(60)
marital_status VARCHAR(30)
address VARCHAR(60)
city VARCHAR(30)
pincode VARCHAR(30)
state VARCHAR(30)

login
form _no VARCHAR(30)
card_no VARCHAR(50)
pin VARCHAR(30)

bank
pin VARCHAR(10)
date VARCHAR(50)
type VARCHAR(20)
amount VARCHAR(20)

Fig 4.4.1 Logical Database Design

The above figure shows the logical database of the designed system in which all the tables with their attributes are displayed with their datatype and size.

4.5 Physical Database:

As one of the core elements of a Health record management system, the database had to be designed in a meticulous systematic manner. This process started at the analysis phase of the project. From the analysis, the researcher was able to identify the necessary tables required for the database and the associated field names, format and length of each table. Below is a list of these tables.

Attribute	**Field Type**	**Length/Size**	**Description**
Form no	varchar	30	Primary key, stores form no generated randomly
Card no	varchar	50	To store card no of the user
pin	varchar	30	Stores Pin no of user

Table 4.5.1 Login

The above table is the table for login page which holds the attributes card number, PIN number and Form number where Form number is the Primary key which stores the unique form number for each user/customer.

This table is used to store the login details for each user/customer and this table is used by the system to check the card number and pin number i.e. for verification and validation purpose of the customer.

Attribute	Field Type	Length/Size	Description
pin	varchar	10	Primary key
date	varchar	50	Stores date of transaction
type	varchar	20	Stores type of account
amount	varchar	20	Stores amount in account

Table 4.5.2 Bank

The above table is of bank where all the details of the user's bank account is stored such as pin number, date, type and amount. The date attributes stores the date of any sort of transaction and type attribute defines the type of account i.e. saving, current, recurring etc. The amount attribute stores the account's balance.

Attribute	Field Type	Length/Size	Description
Form no	varchar	30	Primary key, stores form no
name	varchar	30	Stores name of the user
Father name	varchar	30	Stores father's name of user
DOB	varchar	30	Stores date of birth
gender	varchar	30	Stores gender of customer
email	varchar	60	Stores email id of user
Marital status	varchar	30	Stores marital status of user
address	varchar	60	Stores address of user
city	varchar	30	Stores residing city of user
pin code	varchar	30	Store pin code of the city
state	varchar	30	Stores State of the user

Table 4.5.3 Signup

The above table contains all the personal details of the customer and is filled when the customer signs up i.e. when the customer is new to the system and does not has his/her card number and pin number.

Attribute	**Field Type**	**Length/Size**	**Description**
Form no	varchar	30	Primary key
religion	varchar	30	Stores religion of user
category	varchar	30	Store category of user
income	varchar	30	Stores annual income of user
education	varchar	30	Stores educational details
occupation	varchar	60	Stores occupational details
Pan	varchar	30	Stores pan number of user
Aadhar	varchar	60	Stores Aadhar number of user
senior citizen	varchar	30	Stores seniority status
Existing acc	varchar	30	Stores account details as new or existing.

Table 4.5.4 Signup 2

The above table has all the additional details of the customer such as religion, category, annual income, educational knowledge, occupation, pan number, Adhar number etc. The pan and Adhar number are validated using regular expression. This table is filled after the first signup table is filled and this table contains all the essential additional details of the customer.

Attributes	Field Type	Length/Size	Description
Form no	varchar	30	Primary key
Account type	varchar	40	Stores type of account
Card no	varchar	30	Stores card no of user
Pin	varchar	30	Stores pin no of user
facility	varchar	200	Stores the facilities to be taken

Table 4.5.5 Signup 3

This table contains the details about the customer's account which are used to log the user in to the system using the stored and entered card number and pin number.

4.6 Data Inputs

The entire bank management system itself contains number of interfaces. However, for main components, below are the snapshots of key interfaces.

4.6.1 Login Interface

The login interface below is the first thing which the user sees when access the system. The user has to enter the card no and pin no in order to gain access to the system resources. It is used to gain access and determine that the user can access the resources or not.

Input: The input in this interface are individuals unique card number and PIN number.

Output: The output from this interface will be the user/customer getting into the main interface and utilizing the resources of the system.

Process: The process of this interface is that when the user/customer enters the card number and PIN number the user will be logged in after getting verified from the database data stored in login table such as card number and PIN number.

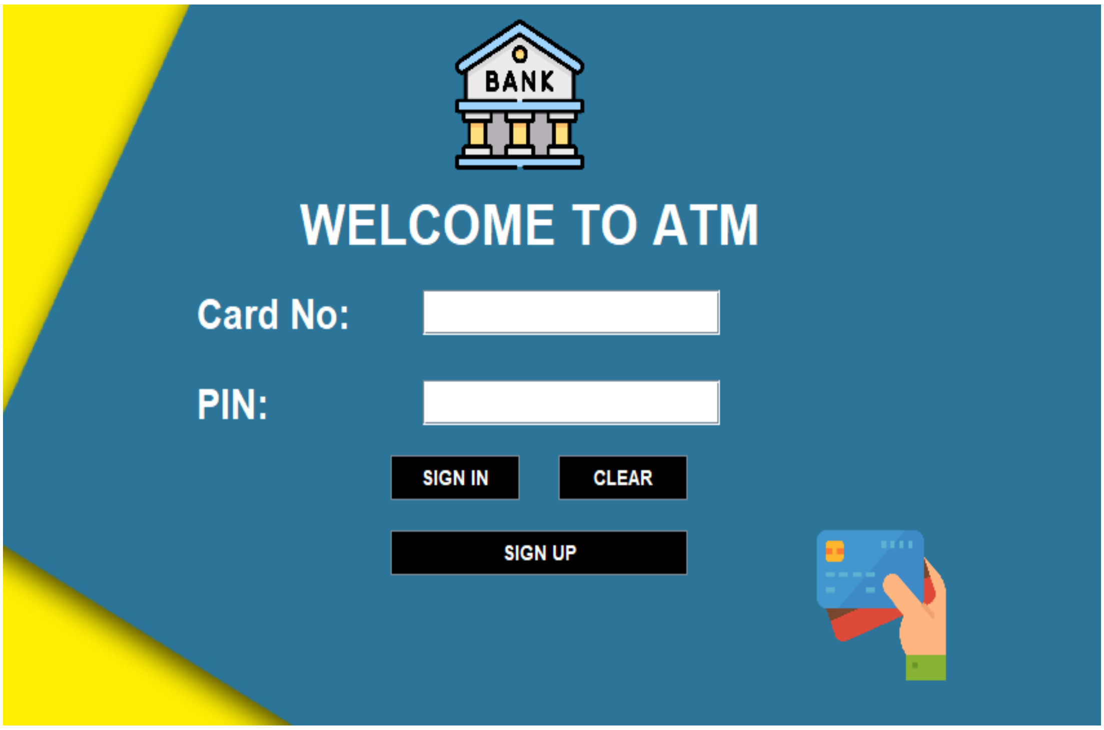


Fig 4.6.1.1 Login interface

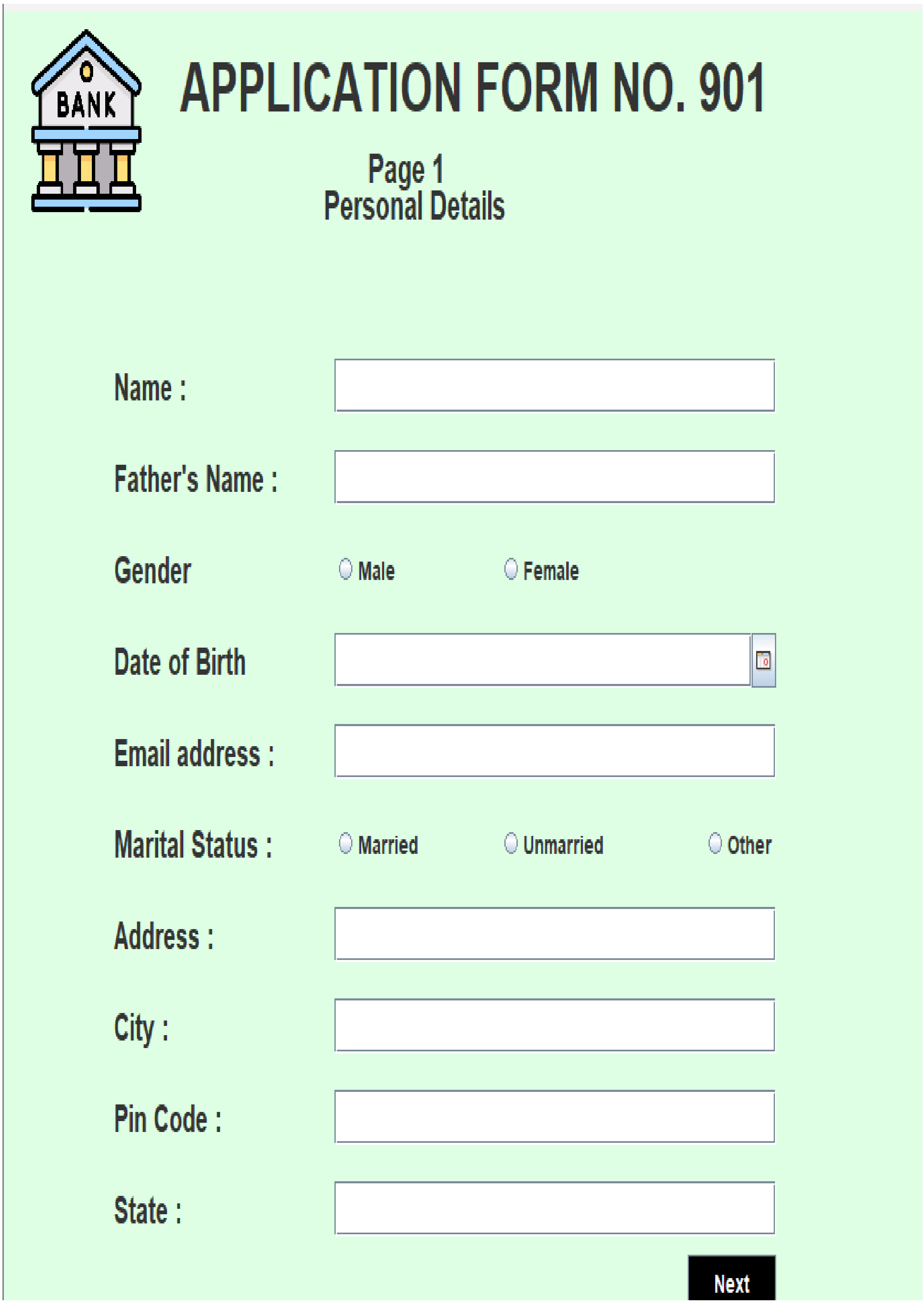
4.6.2 Signup interface

The signup interface below shows the account creation page 1 for the user. It accesses all the basic personal information of the user in order to provide unique card and pin number. This account creation interface helps the users to get access to the resources of the system at their own pace with ease.

Input: The input in this interface are the personal details of the user /customer who is signing in the system for the first time. The details include name, gender, date of birth, city, state etc.

Output: The output received from this interface will be the next signup page where additional details need to be filled.

Process: The processing starts from the login page where a button for signup is present when the user clicks the signup button the signup interface is opened where all the personal details must be filled by the user/customer after filling up the entire form the user will click on the next button and will be lead to the next signup page i.e. signup 2 page.



APPLICATION FORM NO. 901

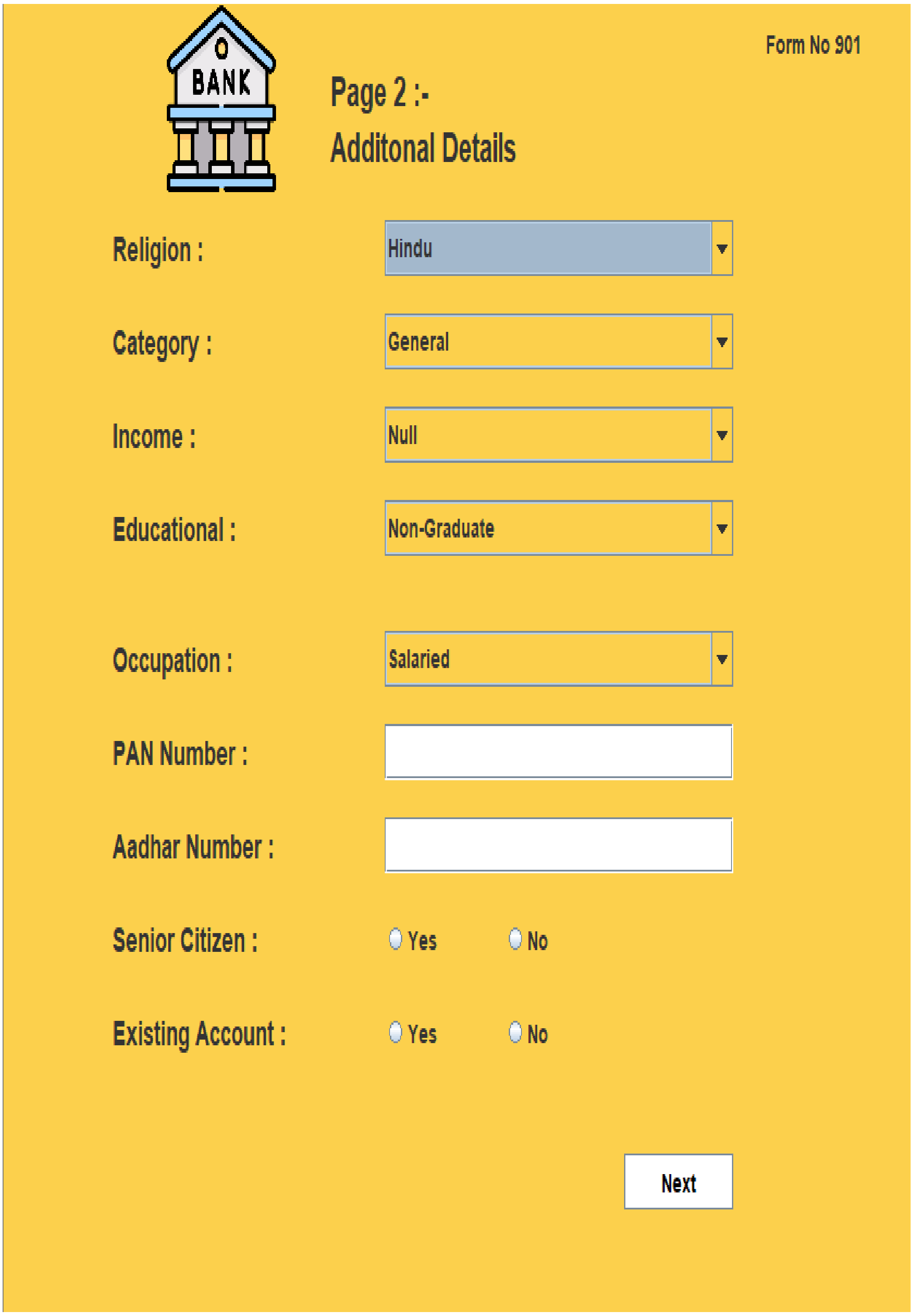
Page 1
Personal Details

Name :			
Father's Name :			
Gender	○ Male	○ Female	
Date of Birth			
Email address :			
Marital Status :	○ Married	○ Unmarried	○ Other
Address :			
City :			
Pin Code :			
State :			

Fig 4.6.2.1 Signup Page

4.6.3 Signup 2 interface

The signup 2 interface is the second page of account creation for the user. It includes additional details such as educational details, occupational details, Pan and Aadhar details etc. In this the user has to enter all the mandatory fields before moving to next page else will be displayed an error message.



Form No 901

Page 2 :-

Additonal Details

Religion :	Hindu ▾	
Category :	General ▾	
Income :	Null ▾	
Educational :	Non-Graduate ▾	
Occupation :	Salaried ▾	
PAN Number :		
Aadhar Number :		
Senior Citizen :	○ Yes	○ No
Existing Account :	○ Yes	○ No

Fig 4.6.3.1 Signup 2 Page

Input: The input in this interface will be the additional details of the user/customer. The details include religion, Education, income, Adhar, Pan details etc.

Output: The output generated here will be the customer/user entering into the next signup page i.e. signup 3 page.

Process: The processing here starts from the signup page when the user clicks on the next button present in signup page which leads the user to the signup 2 page where the user must fill the additional details.

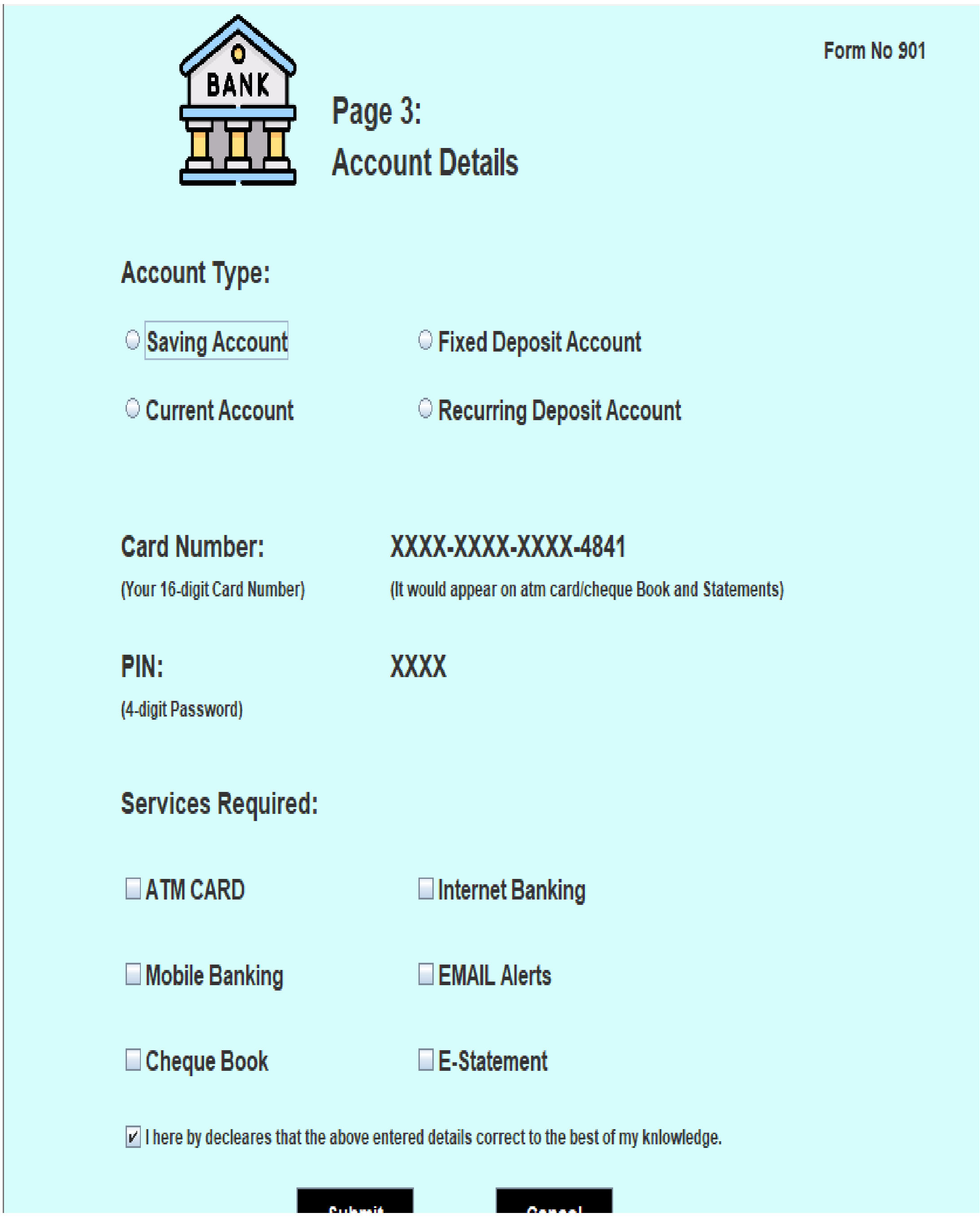
4.6.4 Signup 3 interface

The signup 3 interface below has account details and facilities information which a user can opt. This page is the last application form with a card number and pin number encrypted in it. The user can select all the facilities or can select some of them. In this the user also has to select the account type to finish sign up process. After finishing this step, the user will be redirected to main page.

Input: The input in this page is the user selection of type of account and the services that the user want with their card number and pin number present in the interface in encrypted form.

Output: The output from this page will be the user getting their unique card and pin number and after the whole above process entering into the main interface of the system.

Process: The processing of this starts as soon as the user/customer clicks the next button in the signup page. After clicking on next button in previous page the user will be directed to the signup 3 page where all the mandatory details must be filled by the user. At the end there is a declaration checkbox where the user need to check to get the card and pin number.



Form No 901

Page 3:
Account Details

Account Type:

○ Saving Account ○ Fixed Deposit Account

○ Current Account ○ Recurring Deposit Account

Card Number: XXXX-XXXX-XXXX-4841

(Your 16-digit Card Number) (It would appear on atm card/cheque Book and Statements)

PIN: XXXX

(4-digit Password)

Services Required:

□ ATM CARD □ Internet Banking

□ Mobile Banking □ EMAIL Alerts

□ Cheque Book □ E-Statement

☑ I here by decleares that the above entered details correct to the best of my knlowledge.

Fig 4.6.4.1 Signup 3 page

4.6.5 Transaction interface

This interface below is the main interface for doing transactions. In this, the user has access to all the resources of the main page and can select any option from the interface and can perform the transaction.

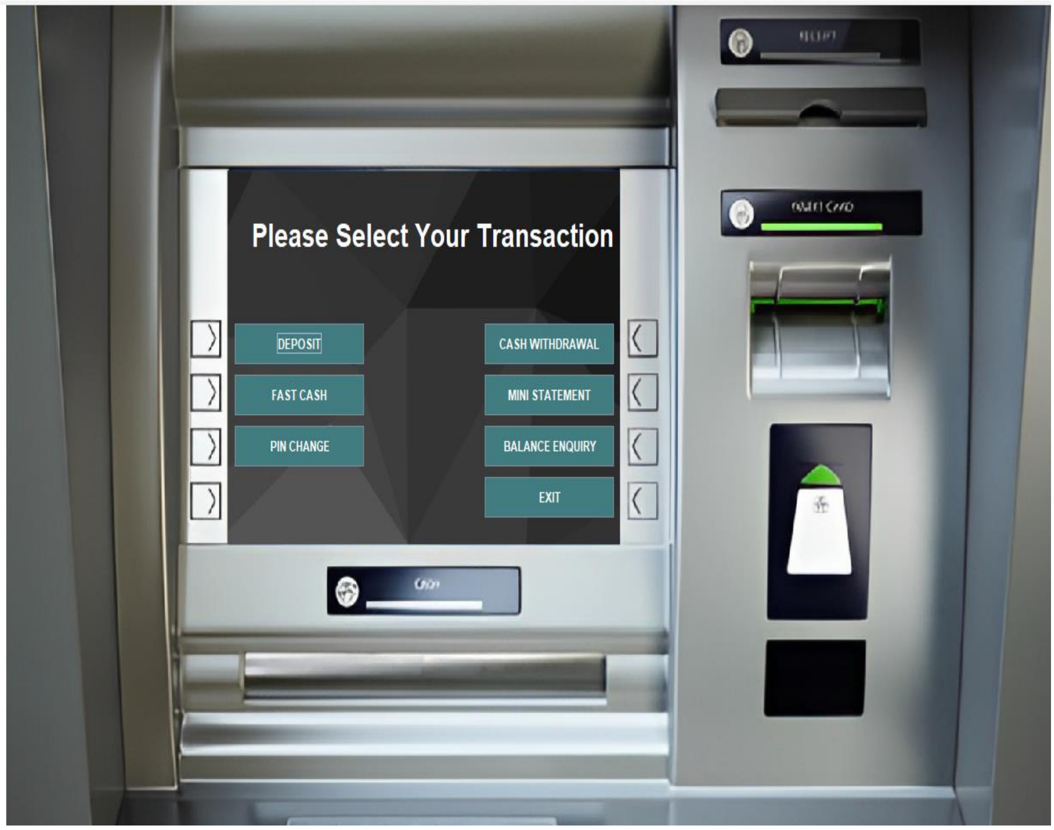


Fig 4.6.5.1 Main Page

Input: The input in above interface is the selection of options from the available options by the user/customer after logging in or signing up.

Output: The output from this interface will be the utilization of the selected service such as deposit, withdraw, mini statement etc.

Process: The processing starts the moment the user/customer logs in to their account and selects the service. Suppose the user selects to deposit the amount into their account so they need to enter the amount they want to deposit and the amount will be deposited into their account and will be saved in database.

4.6.6 Mini Statement interface

In the interface shown below, the user after logging in or signing up can access this feature of the system. I this feature, the user will get the statement of their account's past transactions with date and time.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

5.1.1 Overview of Technologies Used

The Bank Management System was developed using JAVA programming language in front-end and MySQL in back-end. Both of them played very crucial role in building the project.

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. While building this project, we have used the features of JAVA Swing and AWT for icons, buttons, textboxes etc.

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. For the project’s back-end, we have used My SQL Workbench which has the database tables where the data will be stored.

5.1.2 Development Environment Setup

The initial phase started with setting up the environment in IntelliJ Idea by creating a package. This setup provided the server environment facilitating seamless testing and development.

5.1.3 Database Design

The backbone of the bank management system relied on a well-structured database. The design incorporated multiple tables enabling efficient data storage and retrieval.

5.1.4 Frontend Development

The frontend development was entirely based on Java. The features of Swing and AWT were used to build the GUI based interface for the user. Key aspects of this are:

- User-friendly environment: Java's Swing and AWT features provided a GUI based interface which made it easily understandable to users.
- Interactive Elements: The swing features created an interactive element of main interface which saves the real-time data in database.

5.1.5 Backend Development

The backend was developed using My SQL Workbench which helped in connectivity of frontend with backend using JDBC drivers.

5.2 Testing

Testing is critical for a newly developed system as a prerequisite for it being put into an environment where the end users can use it. Exhaustive testing is conducted to ensure accuracy and reliability and to ensure that bugs are detected as early as possible. In the process of designing the RMS, three levels of testing were conducted, namely, unit testing, integration testing and system testing.

5.2.1 Unit Testing

Unit test is where the system is tested partially and independently, component by component, to ensure that particular portion or module is workable within it. In the development of the records management system, each component was tested independently before finally integrating each of them into one system. This test was used by the researcher to verify that every input of data was assigned to the appropriate tables and fields.

5.2.1.1 Login Functionality

Objective: Ensure accurate login functionalities using card and pin number.

Test Cases:

- Test successful login with valid card number and PIN.
- Test failed login with invalid card number.
- Test failed login with incorrect PIN (multiple attempts).
- Test login with expired card.
- Test login with blocked card.

5.2.1.2 Pin Generation

Objective: To ensure the accuracy and non-redundancy of PIN generation.

Test Cases:

- Test PIN generation for new cards (length, format, randomness).
- Test PIN storage and retrieval (encryption, security).
- Test PIN change functionality (validation, update).

5.2.1.3 Deposit Module

Objective: To ensure the accurate deposition of amount to individual's account.

Test Cases:

- Test successful deposit with valid amount and cash acceptance.
- Test failed deposit with invalid amount (negative, zero, exceeding limits).
- Test deposit exceedingly daily/monthly deposit limits.
- Test deposit with insufficient cash in ATM (if applicable).

5.2.1.4 Cash withdrawal Module

Objective: To ensure the accuracy in cash withdrawal and easy transaction.

Test Cases:

- Test successful withdrawal with sufficient balance and cash dispensing.
- Test failed withdrawal with insufficient balance.
- Test withdrawal exceedingly daily/monthly withdrawal limits.
- Test withdrawal exceeding ATM's cash availability.

5.2.1.5 Mini Statement Module

Objective: To ensure the accuracy of transactional history with accurate date and time.

Test Cases:

- Test display of recent transactions with correct details (date, time, amount, type).
- Test display of correct account balance.
- Test handling of empty transaction history.

5.2.1.6 Pin Change Module

Objective: To ensure the accuracy while changing the PIN of the account.

Test Cases:

- Test successful PIN change with valid new PIN.
- Test failed PIN change with invalid new PIN (length, format, similarity to old PIN).
- Test PIN change limit (if applicable, e.g., maximum changes per day).

5.2.2 Integration Test

Integration testing for the BMS (Bank Management System) involved verifying the interaction and interoperability of different modules and functionalities to ensure they work together seamlessly.

Login and PIN Handling Integration:

- Test successful login followed by each module (deposit, withdrawal, mini statement, balance inquiry, PIN change).
- Test failed login attempts and subsequent module access restrictions.
- Test PIN change and its effect on subsequent logins.

Transaction Flow Integration:

- Test balance updates after successful deposits and withdrawals.
- Test transaction history updates after deposits, withdrawals, and PIN changes (reflected in mini statement).
- Test daily/monthly transaction limit enforcement across modules.

Error Handling Integration:

- Test error handling for invalid inputs across modules (e.g., negative amounts, incorrect PIN retries).
- Test error handling for hardware failures (e.g., card reader issues, cash dispenser jams).
- Test communication failures between ATM and bank systems (if applicable).

Security Integration:

- Test authentication and authorization for all modules (ensure access only for valid cardholders).
- Test encryption of sensitive data during communication and storage.
- Test intrusion detection and prevention mechanisms (if applicable).

5.2.3 System Test

System testing for the BMS (Bank Management System) involved comprehensive testing of the entire system as a whole to ensure its functionality, performance, and reliability.

Functional Testing:

- **Login and PIN Handling:**
  - Test successful logins with valid card numbers and PINs.
  - Test failed logins with invalid card numbers, incorrect PINs (multiple attempts), expired cards, and blocked cards.
  - Test PIN change functionality and its effect on subsequent logins.
- **Transactions:**
  - Test balance inquiries for accurate balance display.
  - Test cash withdrawals with sufficient balance, cash dispensing, and balance updates.
  - Test deposits with valid amounts, cash acceptance, and balance updates.
  - Test mini statement for correct display of recent transactions and balance.
  - Test enforcement of daily/monthly transaction limits.

Non-Functional Testing:

- **Performance:**
  - Measure transaction response times under various load conditions (e.g., multiple concurrent users).
  - Test system behaviour under peak load and stress conditions.
- **Security:**
  - Conduct penetration testing to identify vulnerabilities in authentication, authorization, encryption, and other security measures.
  - Test intrusion detection and prevention mechanisms.
- **Usability:**
  - Evaluate user interface design for clarity, ease of use, and accessibility.
  - Conduct user testing to gather feedback on system usability.
- **Reliability:**
  - Test system uptime and availability over extended periods.
  - Conduct recovery testing to verify system resilience after failures.

CHAPTER 6

EVALUATION AND CONCLUSION

6.1 Evaluation

In the attempt to evaluate the designed system, it is imperative that the researcher look back at the predefined functionalities, goals and objectives and analyse those in relation to the expectations met by the system. The Bank Management System was evaluated based on the set of predefined objectives and expected functionalities it was able to fulfil. The Bank Management System was designed to facilitate efficient transaction management in banking system by providing an efficient, reliable computerized banking management system and after a careful evaluation process; it met a considerable portion of those expectations.

The main objective was to design a system that enables faster and more efficient storage, retrieval and updating of customer's account's records. As far as this is concerned, the system met this expectation by giving direct benefit to the customer such as fast cash deposition, withdrawal etc. Analysis was successfully completed. This evaluation is based on the fact that data requirements were collected that successfully enabled the design and development of the system.

The design objectives of creating an efficient bank management system was further accomplished with the creation of add, delete, search and edit functionalities in the system that not only enable computerized but rather efficient, reliable and fast data entry. All these functionalities possess a relatively high level of accuracy. In evaluating this objective in relation to the system's performance, it would therefore be accurate to state that it was achieved to a large extent.

6.2 Limitations of the System

1. Functional Limitations:

- Limited transaction types: The system doesn't support common transactions like fund transfers, bill payments, or mobile top-ups, limiting its usefulness for customers.
- No account opening or management: Users cannot open new accounts or manage existing ones (e.g., change personal details, update addresses) through the system.
- No integration with other banking services: The system operates in isolation, not linking to other online banking features or mobile apps, which hinders a seamless user experience.

2. Security Limitations:

- Random PIN generation: While convenient, it might be less secure than user-defined PINs, as users might forget randomly generated ones more easily, leading to potential card blocking or fraud.
- No biometric authentication: The system lacks modern authentication methods like fingerprint or facial recognition, which offer stronger security and convenience.
- Potential vulnerabilities: As a simulation, it might not have undergone rigorous security testing and could have vulnerabilities that real-world ATMs would address.

3. Usability Limitations:

- Simulated interface: The user experience might not fully replicate real-world ATM interactions, potentially leading to confusion or errors during testing.
- Limited accessibility features: The system might not cater to users with disabilities, such as visual impairments or those requiring assistive technologies.

4. Reliability Limitations:

- Simulated environment: The system's reliability in a real-world deployment with hardware components, network connectivity, and external dependencies isn't tested.

- Error handling: The system's ability to handle unexpected errors or failures in a real-world setting might not be fully evaluated.

5. Data Limitations:

- Simulated data: The system uses artificial data, which might not accurately reflect real-world transaction patterns, account balances, or customer behaviour.
- No historical data: The system lacks historical transaction data, limiting its usefulness for analysis, reporting, and fraud detection.

6.3 Problems Encountered

Functional Issues:

- Incorrect transaction processing:
    - Errors in calculating balances, dispensing cash, or logging transactions.
    - Inaccurate account updates after deposits or withdrawals.
- Invalid input handling:
    - System crashes or unexpected behaviour due to invalid inputs (e.g., non-numeric amounts, incorrect card numbers).
- Data inconsistency:
    - Discrepancies between displayed information and actual account data.
- Communication errors:
    - Issues with network connectivity or data exchange with external systems.

Security Vulnerabilities:

- Authentication bypass:
    - Unauthorized access due to weak authentication mechanisms or loopholes.
- PIN theft:
    - Exposure of PINs during transmission or storage, leading to potential fraud.
- SQL injection and Cross-site scripting (XSS):

    - Vulnerabilities in database queries or user input allowing data manipulation or unauthorized actions.

Usability Problems:

- Confusing interface:
    - Unclear instructions or navigation, leading to user errors.
- Accessibility barriers:
    - Lack of accommodations for users with disabilities.
- Slow response times:
    - Unacceptable delays in transaction processing, frustrating users.

Hardware or Software Faults:

- Card reader malfunctions:
    - Failure to read cards accurately.
- Cash dispenser errors:
    - Incorrect dispensing or jamming of cash.
- Software bugs:
    - Crashes, freezes, or unexpected behaviour due to programming errors.

Data Integrity Issues:

- Data corruption:
    - Loss or damage to account data due to hardware failures, software bugs, or malicious attacks.
- Data inconsistency:
    - Discrepancies between different parts of the system's database.

Integration Challenges:

- Communication failures:
    - Issues with data exchange between the ATM system and other banking systems.
- Incompatibility:

    - Problems with integration due to different technologies or protocols.

6.4 Recommendation/Future Search

Recommendations:

- Expand functionality:
    - Add fund transfers, bill payments, mobile top-ups, and account opening/management features.
    - Integrate with online banking and mobile apps for a seamless user experience.
- Strengthen security:
    - Implement biometric authentication (fingerprint, facial recognition).
    - Conduct rigorous security testing and address vulnerabilities.
    - Encrypt sensitive data during transmission and storage.
- Improve usability:
    - Design an intuitive, user-friendly interface.
    - Incorporate accessibility features for users with disabilities.
    - Conduct user testing to evaluate usability and identify areas for improvement.
- Test in a real-world environment:
    - Deploy the system with real hardware and data to assess reliability and performance.

Future Search:

- Advanced authentication methods:
    - Explore emerging technologies like behavioural biometrics and multi-factor authentication.
- Personalization and customization:
    - Tailor the user experience based on individual preferences and transaction patterns.

6.5 Conclusion

In Conclusion, from a proper analysis and assessment of the designed system, it can be safely concluded that the system is an efficient, usable and reliable records management system. It is working properly and adequately meets the minimum expectations that were set for it initially.

BIBLIOGRAPHY AND REFERENCES

- http://www.sun.com
- http://www.coreservlets.com
- http://www.serverside.com
- http://www.w3schools.com
- https://netbeans.org/kb/index.html
- https://www.javatpoint.com/online-banking-project
- https://www.apachefriends.org%2Findex.html