2026

Manarat International University

Department: CSE (Computer Science & Engineering)

Thesis Project Report

Project Title: Bank Management System with ATM Integration.

Thesis Approval

The thesis entitled **"Bank Management System with ATM Integration"** submitted to the Department of Computer Science and Engineering, Manarat International University, Dhaka, Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering (B.Sc. in CSE) and approved as to its style and contents.

Md.Tanbir Islam ID: 2219CSE50260

Md.Sohag Hossain ID: 2219CSE50261

Shah Kamal ID: 2219CSE50269

Examination held on 24th January 2026

BOARD OF EXAMINERS

Md. Zahurul Haque Haque
Assistant Professor
Department of CSE .....
Manarat International University (Supervisor)

Prof. Dr. Md. Mijanur Rahman
Professor & Head
Department of CSE .....
Manarat International University (Ex-Officio)

Dr. Ramit Azad
Professor
Department of CSE .....
Manarat intemational Liniversity (Member)

Mohammad Rafiqul Islam
Associate Professor
Department of CSE .....
Manarat International University (Member)

Tanvir Ahmed
Assistant Professor
Department of CSE .....
Manarat International University (Member)

Shreshtha Sayantika Maitra
Lecturer
Department of CSE .....
Manarat International University (Member)

Jannatul Ferdaus
Lecturer
Department of CSE .....
Manarat International University (Member)

Dr. Md. Ezharul Islam
Professor
Department of CSE .....
Jahangirnagar University (Member (External))

Project Name: Bank Management System with ATM Integration.

Submit By

Name: Md.Sohag Hossain
ID: 2219Cse50261
Department:
Batch: 19th
Section: C/Evening
Sing: .....

Name: Md.Tanbir Islam
ID: 2219Cse50260
Department: Computer Science & Engineering
Batch: 19th
Section: C/Evening
Sing: .....

Name: Shah-Kamal
ID: 2219Cse50269
Department: Computer Science & Engineering
Batch: 19th
Section: C/Evening
Sing: .....

Submit To

Md. Jahurul Haque

Assistant Professor

Manarat International University.

Project Supervisor

Sing: .....

Declaration Statement

We hereby declare that the project entitled

“Bank Management System with ATM Integration”

Has been completed by us as part of the requirement for the Bachelor of Science degree in Computer Science and Engineering at Manarat International University.

This project has been carried out under the supervision of

Mr. Jahurul Haque, Assistant Professor, Department of CSE, Manarat International University.

We affirm that the work presented in this report is our own original effort. It has not been submitted, in whole or in part, to any other university or institution for the award of any degree, diploma, or certificate.

Throughout this project, we have developed a complete banking solution using the Java programming language with a graphical user interface (GUI). The system integrates bank management operations, employee control panel, and a fully functional ATM service. All core functionalities—including account creation, authentication, employee management, ATM activation, financial transactions, account blocking/unblocking, password recovery, transaction history search and export, and report generation—have been designed and implemented by our project team.

Any external resources, materials, or references used have been acknowledged properly within this report. We collectively take full responsibility for the design, development, accuracy, and integrity of the system and the documentation submitted.

Md. Sohag Hossain

Md. Tanbir Islam

Md. Shah-Kamal

Department of Computer Science & Engineering

Manarat International University

Acknowledgement

All praise and gratitude to Almighty Allah, the Most Gracious, the Most Merciful, for giving us the strength, patience, and determination to successfully complete this project titled
"Bank Management System with ATM Integration."

We would like to express our deepest appreciation and heartfelt gratitude to our respected supervisor Mr. Jahurul Haque, Assistant Professor, Department of CSE, Manarat International University, for his constant guidance, valuable suggestions, and encouragement throughout the development of this project. His expert insights, constructive feedback, and continuous support helped us overcome challenges and shaped the project into its final form.

We also extend our sincere thanks to the Department of Computer Science and Engineering, Manarat International University, for providing us with the academic platform, laboratory facilities, and learning resources required to complete our work.

Our special appreciation goes to our teachers, mentors, and university staff members who have contributed to our academic journey and shared knowledge that made this work possible.

We are grateful to our parents, families, and friends for their unconditional love, prayers, patience, and emotional support. Their encouragement motivated us to stay committed and complete this project with dedication and confidence.

Finally, we acknowledge the teamwork, collaboration, and collective effort of all group members. Without mutual support, time sharing, and determination, this project would not have been possible.

Md. Sohag Hossain
Md. Tanbir Islam
Md. Shah-Kamal
Department of CSE
Manarat International University

Abstract

This project presents the design and implementation of a comprehensive Bank Management System with ATM Integration, developed using Java and a graphical user interface (GUI). The system combines the functionalities of banking administration, employee workflow management, and customer ATM services within a single secure platform.

The main objective of the project is to provide a digital banking solution capable of supporting multiple users with different roles, including administrators, bank employees, and ATM account holders. Administrators have full control over employee accounts and customer profiles. Employees are able to manage bank accounts, activate ATM cards, perform transactions, and generate reports. Customers can independently access ATM services such as balance inquiries, withdrawals, transfers, mini statements, and password updates.

The system ensures account security through password recovery, access control, automated account blocking, and role-based authentication. It also supports powerful data management features, including account history search, PDF export, and profile storage. By integrating multiple modules into a single system, the application demonstrates how digital platforms can streamline banking activities, reduce manual errors, and increase operational efficiency.

This project emphasizes usability, security, and extendibility within the financial service domain, making it suitable for future development such as online banking, mobile app connectivity, and cloud-based transaction processing.

Table of Contents

Contents

Declaration Statement 4

Acknowledgement 5

Abstract 6

Table of Contents 7

Chapter–1: Introduction 9

1.1 Introduction 9

1.2 Problem Statement 10

1.3 Objectives of the Project 10

1.4 Scope of the Project 11

1.5 Limitations of the Project 12

Chapter–2: Literature Review 14

2.1: Existing banking systems 14

2.2 Related Research Works 15

2.3 Limitations of Existing Banking Systems 16

Chapter–3: system analysis & methodology 18

3.1 About system analysis & methodology 18

3.2 System Analysis 18

3.2.2 Feasibility Study 18

3.3 System Design Approach 19

3.3.1 Top–Down Design 19

3.3.2 Object-Oriented Principles 19

3.4 Development Methodology 19

3.5 Tools and Technologies Used 20

Chapter–4: System Design 20

4.1 System Architecture 20

Layers of Architecture- 21

Database Layer 22

4.3 System Modules 23

4.3.1 Admin Module 24

4.3.2 Employee Module 25

4.3.3 ATM User Module 26
4.3.4 Transaction Module 27
4.3.5 Security Module 27
4.4 Data Flow Diagram (DFD) Overview: 29
4.6 Source Code Design 30
4.7 Relationships: 31
Chapter–5: Implementation 32
5.1 Code explanation 32
5.1.1 Database Connectivity 32
5.1.2 Customer Registration Module 32
5.1.3 Account Creation Module 33
5.1.4 Deposit and Withdrawal Module 33
5.1.5 ATM Authentication Module 33
5.1.6 ATM Withdrawal Module 33
5.1.7 Balance Enquiry Module 34
5.1.8 Mini Statement Module 34
5.1.9 Exception & Error Handling 34
5.1.10 Logout and Session Management 34
5.2 Source Code 34
Chapter–6: Testing & Results 38
Performance analysis: 39
Chapter-7: Future Work 41
Chapter- 8: Conclusion & Recommendation 42
8.1 Conclusion 42
8.2 Recommendations 42
Chapter -9: References 43

Chapter–1: Introduction

1.1 Introduction

In the modern digital era, banking services are rapidly transitioning from traditional manual processes to fully automated and software-driven platforms. Banks today depend heavily on advanced information systems to manage customer accounts, handle transactions, and ensure secure financial operations. With increasing customer expectations, data security requirements, and the demand for 24/7 service availability, developing an efficient computerized banking system has become a crucial focus of the information technology industry.

This project titled “Bank Management System with ATM Integration” has been developed to provide a complete, secure, and user-friendly solution that supports the daily operations of a banking environment. The system is designed using Java and GUI technologies, enabling easy navigation, fast execution, and efficient handling of data. The application integrates three major roles: Administrator, Employee, and ATM User, each equipped with tailored functionalities and access privileges.

The Administrator maintains full control over the system, including employee management, account activation, data monitoring, and security enforcement. Employees are responsible for creating and maintaining customer accounts, handling deposits, withdrawals, and inter-account transfers while also generating reports and account history. ATM Users are provided with essential self-service features such as balance inquiry, password update, fund transfer, and cash withdrawal.

The purpose of this project is to minimize paperwork, reduce human error, and provide customers with fast, secure, and reliable access to financial services. With features like password recovery, account blocking, transaction history search, and export options, the system offers enhanced flexibility and operational efficiency compared to traditional banking systems.

By combining banking operations with ATM functionality in a single platform, the project demonstrates how automation can transform financial institutions, improve customer service, and strengthen data security. This implementation also lays a foundation for future enhancements such as online banking modules, mobile integration, real-time notifications, and database scalability.

1.2 Problem Statement

Traditional banking processes rely heavily on manual record keeping, face-to-face interaction, and human-driven procedures for account management, transaction recording, and customer service delivery. These manual operations are time-consuming, prone to errors, difficult to track, and inconvenient for both bank staff and customers. As financial activities continue to expand and customer expectations for faster services increase, most existing systems fail to provide an integrated platform that supports secure management of accounts, employee operations, and self-service facilities.

Moreover, many local banks do not offer a unified system that connects internal bank operations with customer ATM services. Customers often depend on separate ATM networks, while administrators and employees lack tools to monitor all accounts, track transaction history, or control user activity from a centralized platform. This results in delays, inefficient workflows, poor security enforcement, and limited service availability outside banking hours.

Therefore, there is a critical need for a computerized banking solution that can automate account management, ensure secure transactions, control employee access, support ATM-based services, and allow customers to manage routine activities independently. A fully integrated Bank Management System with ATM functionality can address these limitations by minimizing manual tasks, reducing processing errors, improving data accessibility, and enhancing service reliability.

This project aims to solve these challenges by developing a complete digital banking platform that supports administrators, employees, and customers within a single, secure system.

1.3 Objectives of the Project

The primary objective of this project is to design and develop a secure and user-friendly Bank Management System with ATM Integration capable of supporting core banking activities through automation. To achieve this goal, the project focuses on the following specific objectives:

1. **Automate Banking Operations**

Replace manual banking processes with a computerized system to manage accounts, transactions, employee activities, and ATM services efficiently.

2. **Provide Role-Based Access Control**

Develop separate login systems for administrators, employees, and ATM users, ensuring each user has access only to authorized features.

3. **Enable Secure Account Management**

Allow the creation, activation, blocking, and modification of customer and employee accounts with proper authentication and password recovery mechanisms.

4. **Support Financial Transactions**

Implement reliable functionalities for deposits, withdrawals, transfers, and transaction history tracking.

5. **Integrate ATM Services**

Provide ATM card users with essential features including balance inquiry, fund transfer, mini statement, and password change outside the bank interface.

6. **Enhance Data Monitoring and Reporting**

Allow users to search transaction history, view account details, and export data as PDF or image files for recordkeeping and analysis.

7. **Improve Security and Reliability**

Ensure confidentiality of user data through password security, access restrictions, and account control (activation/blocking).

8. **Increase Efficiency and Reduce Human Error**

Minimize paper-based record handling, reduce processing time, and eliminate errors caused by manual data entry.

9. **Support Future System Expansion**

Develop a modular framework that can later incorporate features such as online/mobile banking, cloud storage, or machine learning–based fraud detection.

1.4 Scope of the Project

The scope of this project covers the design and implementation of a complete Bank Management System with ATM Integration, aimed at digitizing key financial operations and providing secure access to banking services for multiple user types. The system is developed using Java with a graphical user interface, making it suitable for deployment in small to medium-sized banking environments.

This project supports three primary user roles—Administrator, Employee, and ATM User—each with unique features and responsibilities. The administrator is able to manage employees, customer accounts, ATM card activation, account blocking or deletion, password resets, transaction monitoring, and report generation. Employees are granted capabilities to create and update customer accounts, conduct deposits, withdrawals, and fund transfers, and manage ATM activation while generating transaction histories. ATM users can independently perform basic banking operations such as cash withdrawal, balance inquiries, password changes, and money transfers without interacting directly with bank staff.

The project includes secure login systems, password recovery, transaction validation, account tracking, and the ability to search and export transaction records between specified dates. The system maintains accuracy through data validation and reduces human interaction by allowing customers to access services 24/7 through ATM modules.

However, the scope of this project focuses on desktop-based banking automation and does not currently include online banking, mobile application connectivity, or real-time interbank transaction settlement. These features can be introduced as future enhancements.

Overall, this project aims to replace manual financial workflows with a reliable software solution—improving operational efficiency, reducing errors, enhancing security, and offering a foundation for expansion into more advanced digital banking services.

1.5 Limitations of the Project

Although the Bank Management System with ATM Integration provides valuable automation and improves banking efficiency, the project has certain limitations, which are listed below:

1. **Limited to Desktop Application**

The system is designed as a standalone desktop-based platform and does not support web-based or mobile banking access.

2. **Single Branch Operation**

The project focuses on managing accounts within one bank branch and does not include inter-branch or interbank transaction support.

3. **No Real-Time Online Connectivity**

ATM features operate within the system database and are not linked to a live bank network or ATM switch service.

4. **Basic Security Implementation**

While password protection and account control exist, advanced encryption, biometric authentication, and fraud detection techniques are not implemented.

5. **Manual Database Backup Required**

The system does not include automated data backup, recovery, or cloud synchronization features.

6. **Limited Error Handling**

The system may not handle unexpected failures such as power outages, system crashes, or network interruptions with full recovery support.

7. **No Multi-Currency or International Transaction Support**

Only one currency type is handled, and international transfers or exchange rates are beyond the project scope.

8. **Limited Reporting Tools**

Although users can search and export history, advanced analytics, auditing dashboards, and trend forecasting are not included.

9. **Scalability Constraints**

The current system architecture may need modifications before being expanded to support large-scale deployments or thousands of simultaneous users.

10. **No Integration with External Banking APIs**

The system operates independently and does not interact with national banking systems, SMS gateways, mobile payments, or credit card networks.

Chapter–2: Literature Review

2.1: Existing banking systems
Existing Banking Systems

Traditional and modern banking rely on a variety of software solutions to manage financial services, customer accounts, and transaction records. Existing banking systems used today can generally be categorized into manual systems, core banking systems, and ATM/online banking platforms, each serving different levels of automation and service delivery.

1. Manual Banking Systems

Historically, banks operated using paper-based record keeping. Customer information, account balances, and transaction logs were stored in ledgers and maintained manually by staff.

However, manual systems are:

- Time-consuming
- Prone to human errors
- Difficult to update and track
- Not suitable for handling large-scale data Due to these disadvantages, most banks have moved toward automation.

2. Core Banking Systems (CBS)

Modern banks adopt Core Banking Systems, which allow all major activities—account creation, deposits, withdrawals, loan management, and customer service—to be processed through centralized computer networks.

CBS offers:

- Real-time data synchronization
- Account access from any bank branch
- Improved data security and reliability
- Faster processing and reporting

Still, these systems are expensive to implement and require trained staff, continuous maintenance, and secure infrastructure.

3. Online and Mobile Banking Systems

With the growth of digital technology, banks now provide services through:

- Internet banking portals
- Mobile applications
- Digital wallets

These systems allow customers to:

- Check balances
- Transfer funds

- Pay bills
- Receive notifications

But they require stable internet connectivity and advanced cyber security protections to prevent threats such as hacking and fraud.

4. ATM and Card-Based Banking

Automated Teller Machines (ATM) extend banking access beyond working hours and allow users to:

- Withdraw cash
- Check balances
- Request mini statements

Many ATMs are integrated with national or international networks such as VISA or Mastercard, enabling global money access.

However, ATM operations require secure system integration, maintenance, and constant monitoring.

Conclusion on Existing Systems

While modern banking systems offer convenience, security, and fast access, many smaller institutions still rely on basic or limited-function systems. The need remains for affordable, easy-to-use banking software that connects administrative tasks, employee workflows, and customer-facing ATM functions—especially for banks without large IT infrastructure.

Your Bank Management System with ATM Integration aims to bridge this gap by providing an all-in-one platform that:

- Automates internal operations
- Supports secure ATM usage
- Reduces errors
- Improves service access

2.2 Related Research Works

Several research studies and software development projects have focused on enhancing banking operations through automation, secure transaction processing, and user-centric design. These works provide insight into both the design principles and implementation challenges associated with computerized banking systems.

Researchers have emphasized the importance of computerized banking systems to replace manual record-keeping and improve service delivery. In many studies, core banking solutions have been proposed to manage customer accounts, process transactions, and provide administrative controls within a secure

digital environment. These systems typically include features such as multi-user access, role-based permissions, account reconciliation, and automated reporting. The research highlights how automation reduces processing time, eliminates redundant paperwork, and ensures better data integrity.

Another area of focus in the literature is ATM integration and self-service banking. Several projects investigate how banks can extend services beyond branch counters through ATM interfaces, allowing customers to withdraw cash, check balances, and perform fund transfers independently. Researchers point out that ATM modules must interact seamlessly with the central banking system and maintain synchronization of data in real time to avoid inconsistencies. Security considerations, such as password protection and transaction validation, are also underscored in these works.

Some researchers have developed prototype bank management systems using object-oriented programming languages including Java, C++, and Python. These projects demonstrate how graphical user interfaces improve usability for different user roles—administrators, employees, and customers—making banking operations more intuitive. Works in this area often include modules for account creation, updating customer data, transaction logging, and reporting utilities such as generating PDF statements.

A number of studies also explore security mechanisms in banking applications. These include password recovery procedures, access control models, data encryption techniques, and auditing capabilities. Research in this field stresses that financial applications must prevent unauthorized access, ensure confidentiality of customer information, and maintain transaction integrity to build user trust.

Moreover, researchers have examined the limitations of existing banking systems, such as dependency on internet connectivity for online banking, lack of integration between internal systems and ATMs, and challenges in scalability. These findings inform the design of more flexible and modular systems that can be expanded in the future—for example, adding mobile banking or cloud-based backup.

Overall, existing research supports the need for robust, integrated banking solutions that unify administrative controls with customer-facing services. The present project builds upon these ideas by implementing a complete Bank Management System with ATM Integration that automates core banking tasks, facilitates secure transactions, and improves operational efficiency through user-friendly GUI features.

2.3 Limitations of Existing Banking Systems

Despite the advancements in banking software and automated systems, existing banking solutions still face several limitations that affect efficiency, security, and user experience. These limitations provide motivation for developing an improved, integrated system like the one presented in this project:

1. Fragmented Systems

Many banks operate multiple independent systems for core banking, employee management, and ATM services. This fragmentation leads to inconsistent data, delayed updates, and difficulty in monitoring transactions across all platforms.

2. Limited Role-Based Access Control

Existing systems often lack proper separation of responsibilities. Administrators, employees, and customers may have overlapping access rights, increasing the risk of unauthorized changes or misuse of sensitive data.

3. Manual and Semi-Automated Processes

Some banks still rely on semi-automated processes, where transactions or account updates require manual intervention. This slows down operations, increases the likelihood of human error, and reduces overall productivity.

4. Insufficient Security Measures

Many current systems have basic password protection but lack advanced security features like automated account blocking, secure password recovery, and proper transaction validation. This exposes both banks and customers to potential fraud and data breaches.

5. Limited ATM Integration

Existing banking systems often operate ATM networks separately from the core bank system. This limits the ability to synchronize transactions in real-time, track account histories accurately, or provide a seamless user experience across branches and ATMs.

6. Poor Reporting and Data Analysis Tools

While transaction histories exist, most systems offer minimal options for generating detailed reports, exporting data, or analyzing patterns over time. Administrators and employees face difficulty in auditing accounts efficiently.

7. Accessibility Constraints

Many systems do not provide a 24/7 solution for customers. ATM and online banking functionalities are either limited or require additional infrastructure, reducing customer convenience.

8. Scalability Issues

Existing solutions are often designed for small-scale operations and cannot efficiently handle a growing number of users, accounts, or simultaneous transactions without performance degradation.

9. Lack of Modular Design for Future Upgrades

Many current systems are rigid and cannot easily incorporate new features like online banking, mobile integration, or advanced security mechanisms, limiting their long-term usability.

Conclusion:

The above limitations highlight the need for a fully integrated, secure, and user-friendly banking system that combines account management, employee workflow, and ATM services within a single platform. This motivates the development of the Bank Management System with ATM Integration, which addresses these gaps by offering automation, role-based control, transaction security, and flexible reporting features.

Chapter–3: system analysis & methodology

3.1 About system analysis & methodology

This chapter presents the analysis of the existing problems in banking operations and outlines the methodology applied to design, develop, and implement the proposed Bank Management System with ATM Integration. The goal of the analysis is to identify key operational challenges and provide a structured approach to solving them through a fully automated and secure software solution.

3.2 System Analysis

3.2.1 Requirement Identification Based on observation and project goals, the following requirements were identified:

Functional Requirements

- Separate login modules for Admin, Employees, and ATM users
- Account creation, modification, activation, and blocking
- Fund transfer, deposit, and withdrawal
- Automated ATM access linked with main bank records
- Password recovery and change functionality
- Search and export transaction histories
- Role-based user privileges
- Generate PDF statements and save customer profile images
- Maintain transaction logs for all activities
- Non-Functional Requirements
- Security: User authentication, restricted access, password safety
- Reliability: Accurate and consistent transaction processing
- Usability: Simple, user-friendly GUI interface
- Scalability: System designed to accommodate future features
- Performance: Fast execution with minimal processing delay

3.2.2 Feasibility Study

The project was evaluated from several feasibility perspectives:

1. Technical Feasibility:

Java programming language, GUI libraries, and a relational database were used, all of which support robust system development and are widely available.

2. Economic Feasibility:

No commercial licensing is required for development tools, making the system cost-effective and suitable for small or medium financial institutions.

3. Operational Feasibility:

The system provides an intuitive interface and modular workflow that simplifies daily bank operations, making it easy for employees to adopt.

4. Schedule Feasibility:

The project was planned and implemented within the academic timeframe, following an organized development plan.

3.3 System Design Approach

3.3.1 Top–Down Design

The project follows a top–down design methodology, breaking the system into small, manageable modules:

- Admin module
- Employee module
- ATM user module
- Transaction processing module
- Data management module
- Security and validation module

Each component was designed individually and later integrated to form a complete system.

3.3.2 Object-Oriented Principles

- The system was built using object-oriented design concepts:
- Encapsulation ensures secure handling of account data and transactions
- Inheritance allows code reuse across similar components
- Polymorphism enables flexible handling of different user actions
- Abstraction hides internal logic from the user interface

3.4 Development Methodology

3.4.1 Software Development Life Cycle (SDLC) This project followed the Waterfall Model, consisting of sequential phases:

- **Requirement Analysis –** Understanding system needs
- **System Design –** Structuring user modules and workflows
- **Implementation –** Coding modules in Java
- **Integration & Testing –** Combining modules and validating output

- **Deployment –** Running functional system on a desktop environment
- **Maintenance –** Debugging errors and refining features

The Waterfall method was selected because project requirements were clearly defined early and development progressed in predictable stages.

3.5 Tools and Technologies Used

- **Programming Language:** Java (JDK 25)
- **GUI Framework:** Java Swing / JavaFX (as applicable)
- **Database**: MySQL / SQLite
- **IDE:** NetBeans / Eclipse / IntelliJ IDEA
- **Database Connectivity:** JDBC and MySQL Connector
- **Output Formats:** PDF generation and PNG image creation
- **Libraries & APIs:**
    - JavaMail API for email notifications and communication
    - iText library for PDF generation
    - Java Activation Framework for data handling
    - JCalendar for date selection and calendar management
- **Security:** User authentication and input validation checks

Chapter–4: System Design

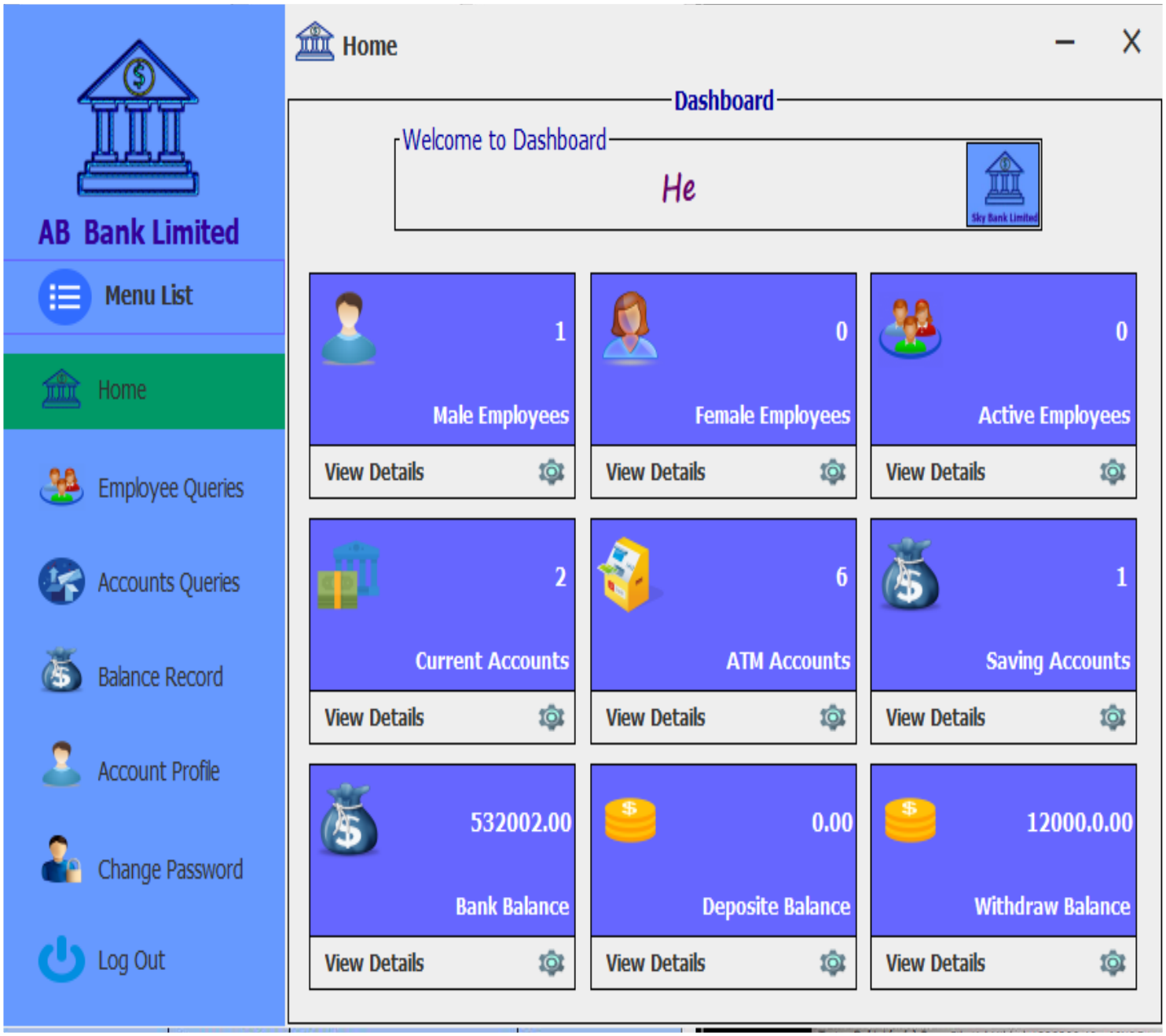
System design is the process of translating requirements and analysis into a structured technical solution. This chapter describes how the Bank Management System with ATM Integration is organized, including architecture, modules, workflows, data flow, and database structure. The design ensures that the system is secure, scalable, and efficient, while allowing easy maintenance and future enhancements.

4.1 System Architecture

- The system architecture follows a simple client–server model, where:
- The Client (User Interface) interacts with Admins, Employees, and ATM Users.
- The Application Layer processes logic such as authentication, transactions, and account management.
- The Database Layer stores all account details, transaction records, user profiles, and login credentials.

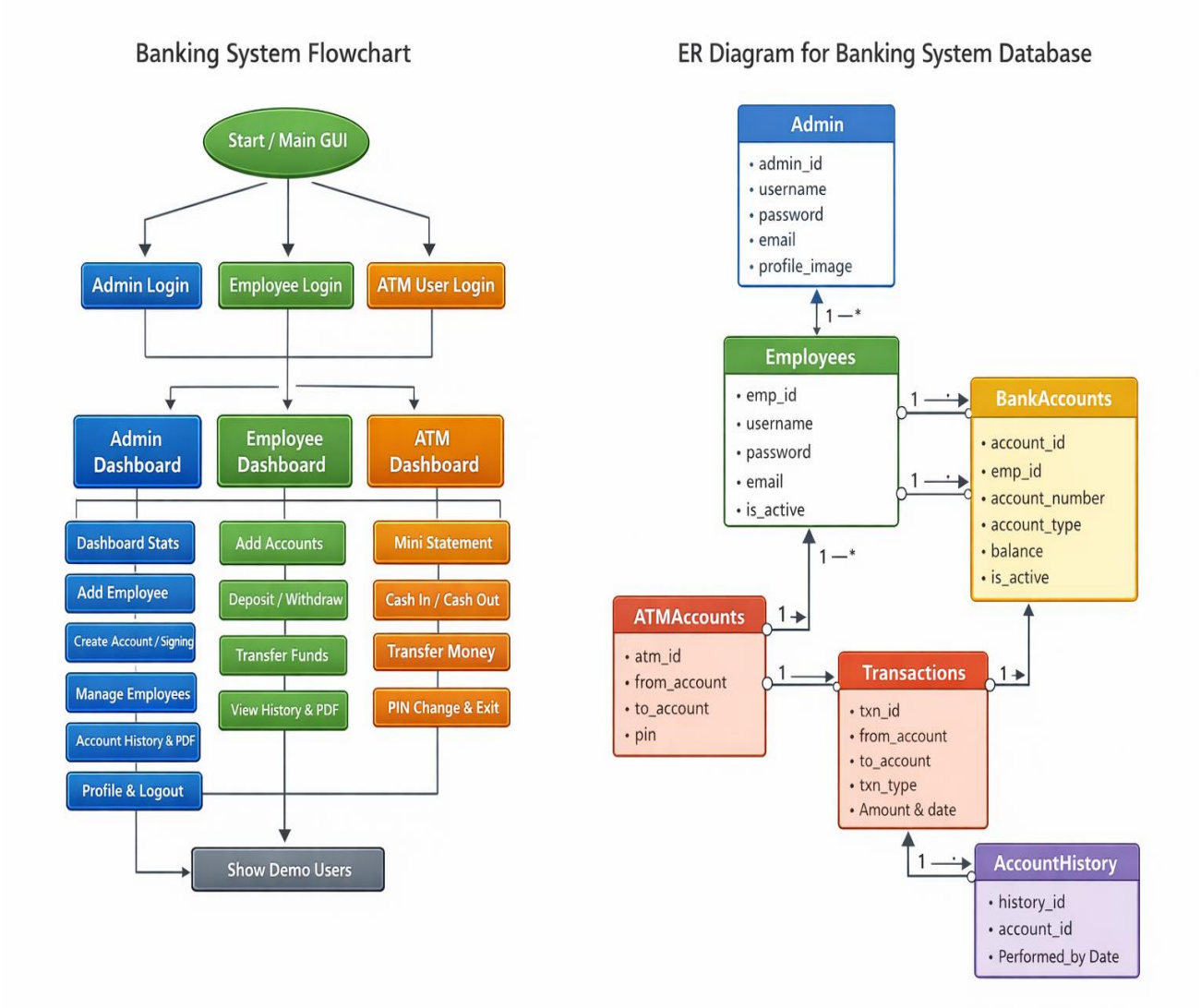
Layers of Architecture-

Presentation Layer (GUI): Bank Management System Admin Page



Picture: (Admin page View)

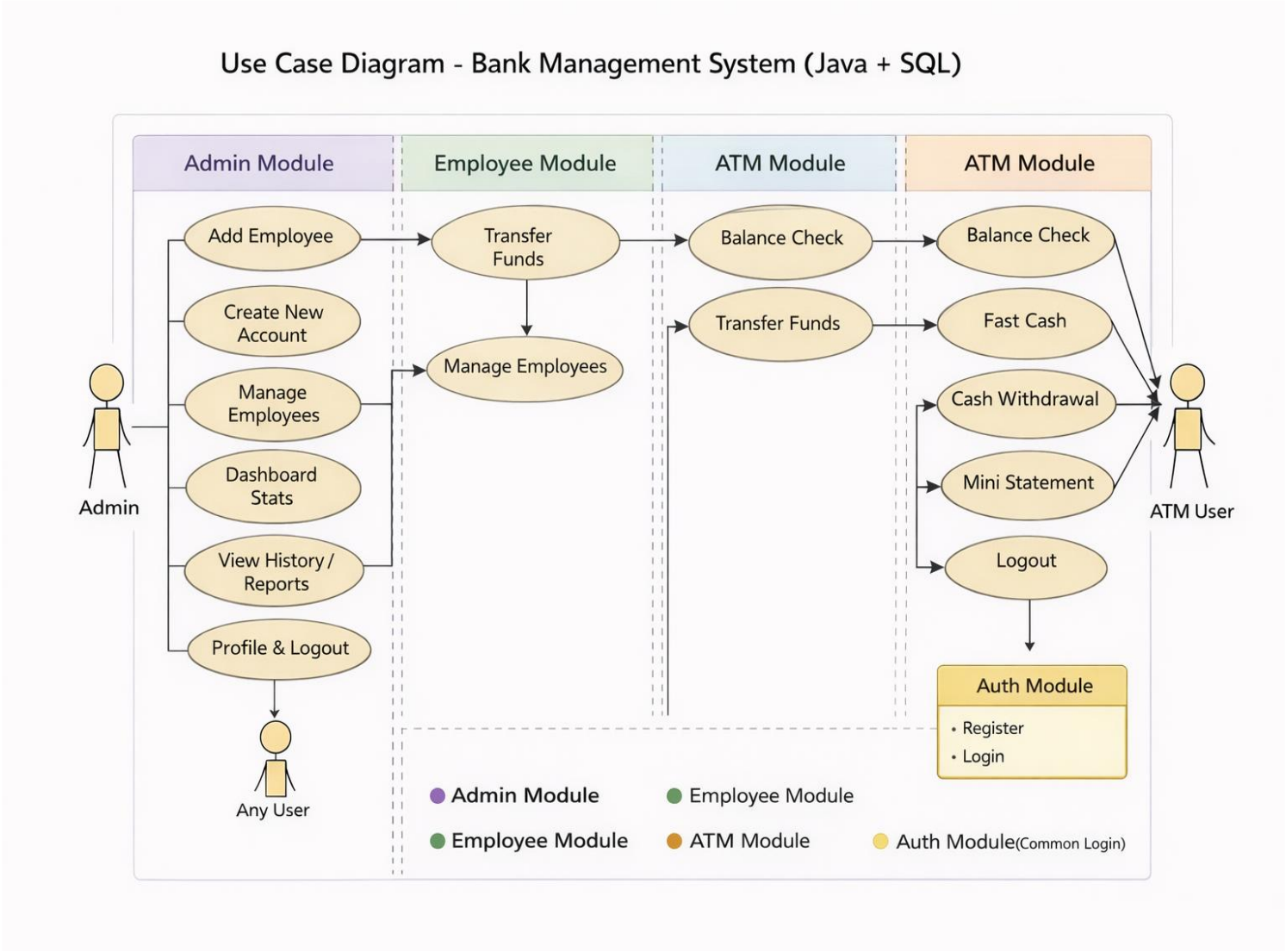
Database Layer: Stores customer accounts, employee records, ATM credentials, and transaction history.



Picture: (Database System and Bank Flow Chart).

4.3 System Modules

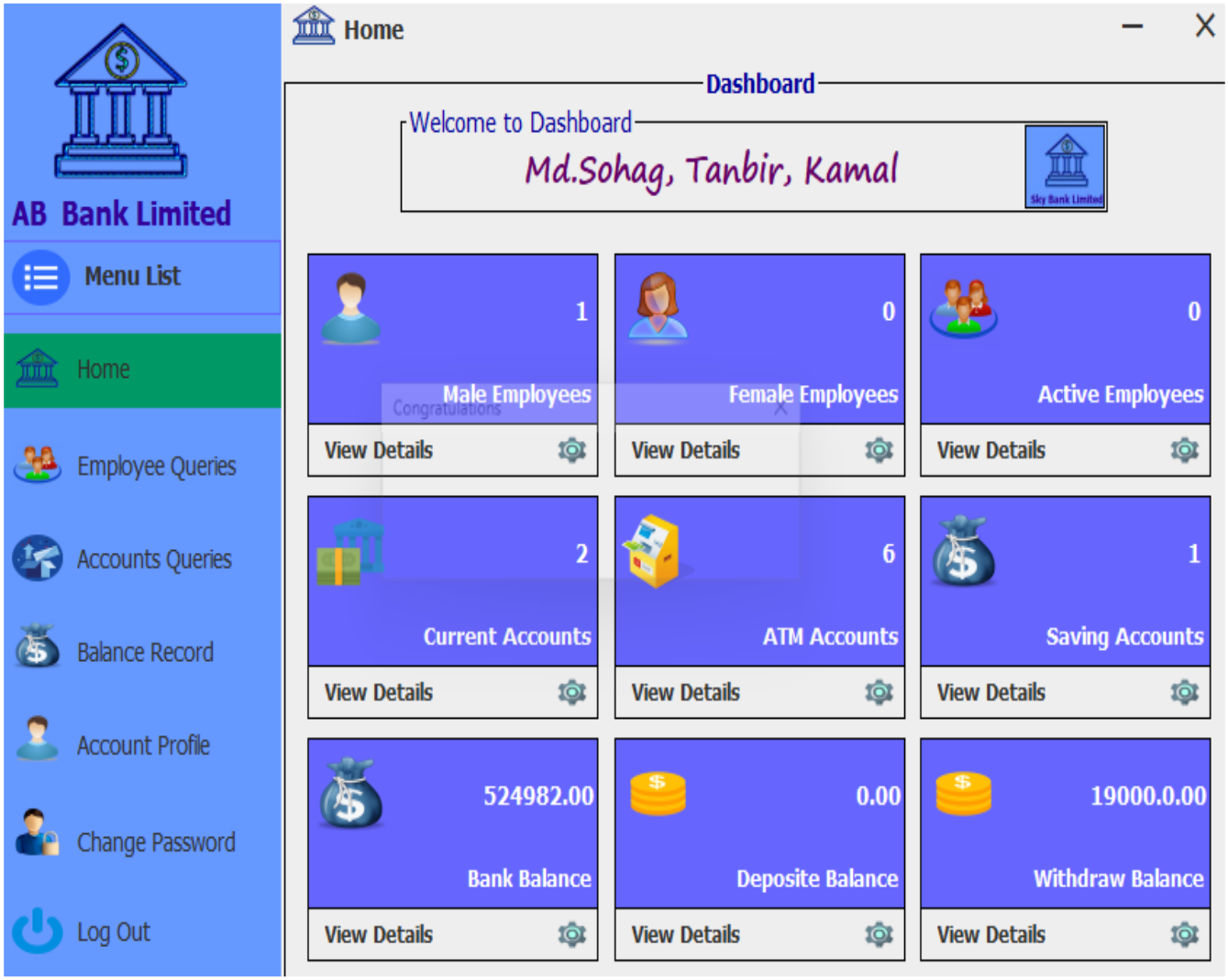
The system is divided into major functional modules, each handling specific responsibilities.



Picture: (Full System Module Layout)

4.3.1 Admin Module

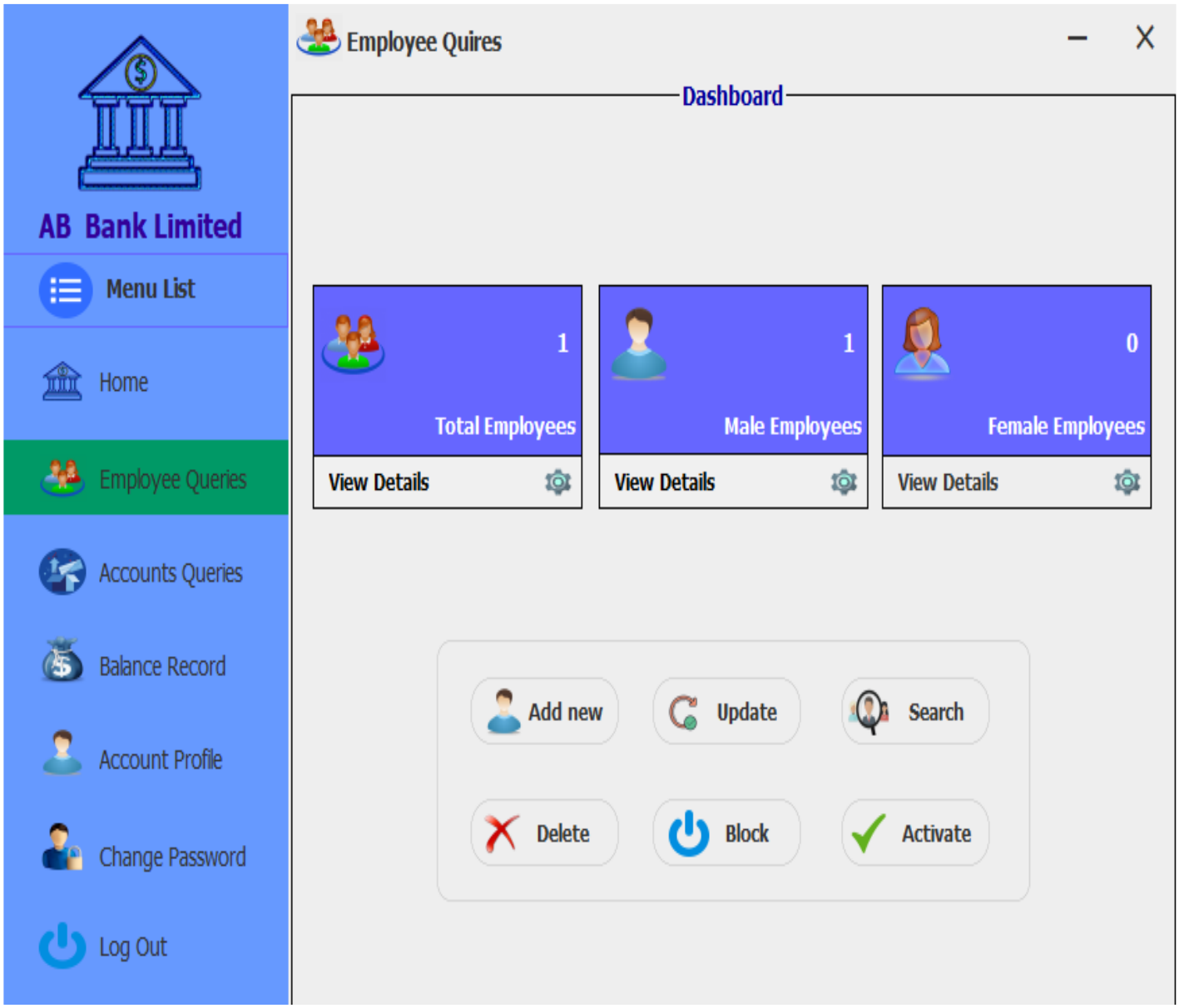
- Login and authentication
- Manage employee accounts (add, block, activate, delete)
- Manage customer accounts and ATM activation
- View and search transaction history
- Export reports to PDF and save profile images
- Reset password and update profile



Picture: (Admin Module Page).

4.3.2 Employee Module

- Add and manage customer bank accounts
- Activate and block ATM accounts
- Perform deposit, withdrawal, and fund transfers
- Search account history by date range
- Generate reports and export files
- Update employee profile and password



Picture: (Employee Module Page)

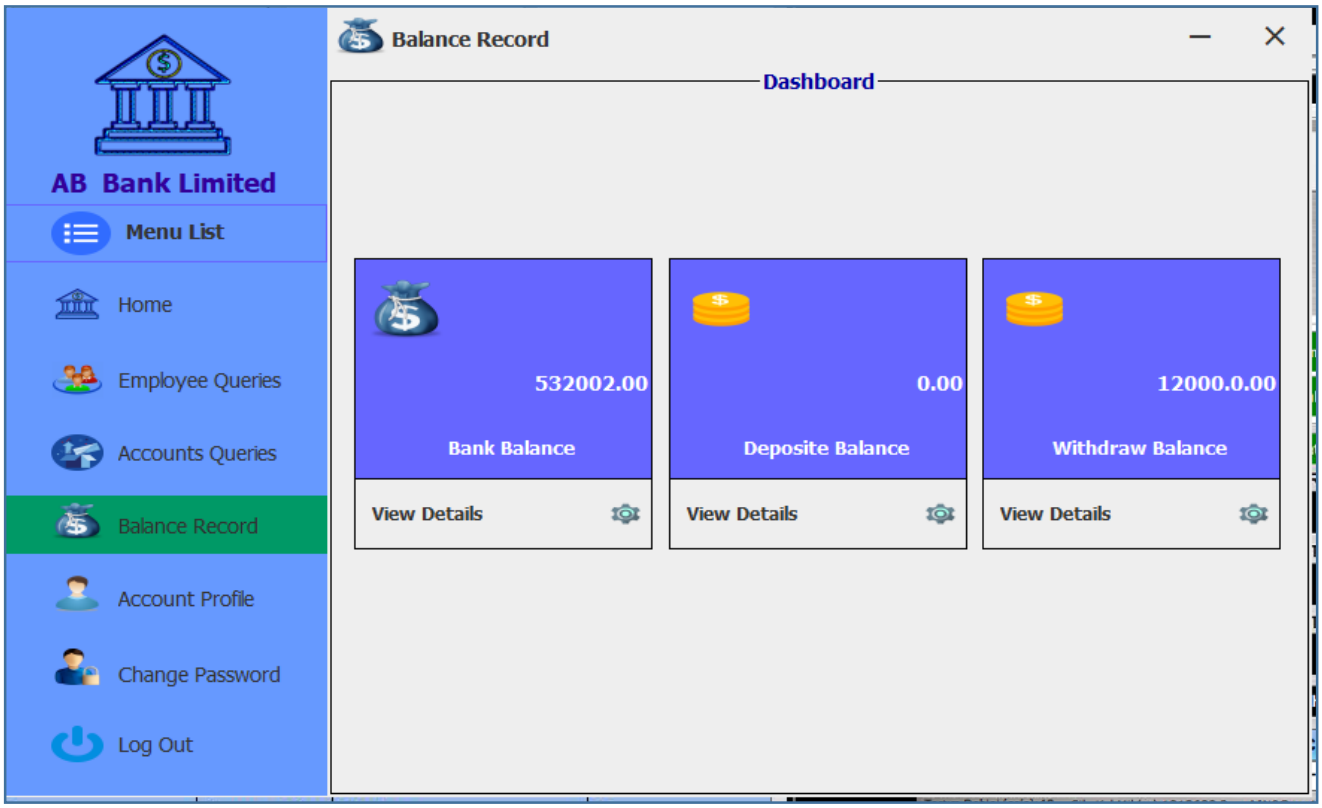
4.3.3 ATM User Module

- Login using ATM credentials
- Withdraw cash
- Transfer funds between accounts
- View mini statement
- Change ATM password
- Exit ATM session securely

Picture :(ATM User Module Page)

4.3.4 Transaction Module

- Validates customer balance
- Logs every deposit, transfer, and withdrawal
- Ensures secure real-time updates to the database

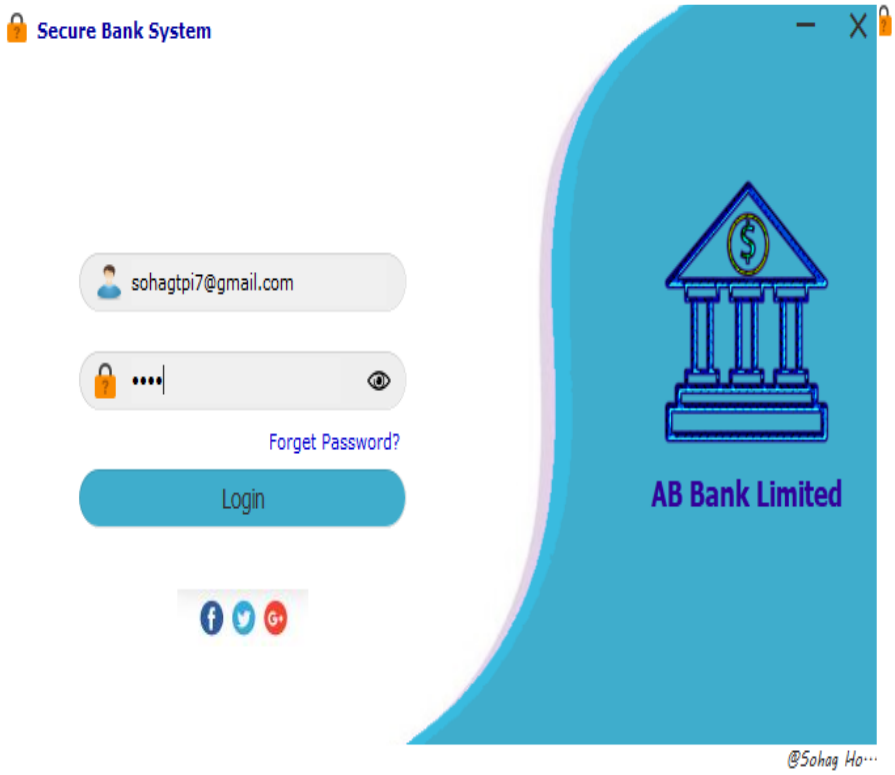


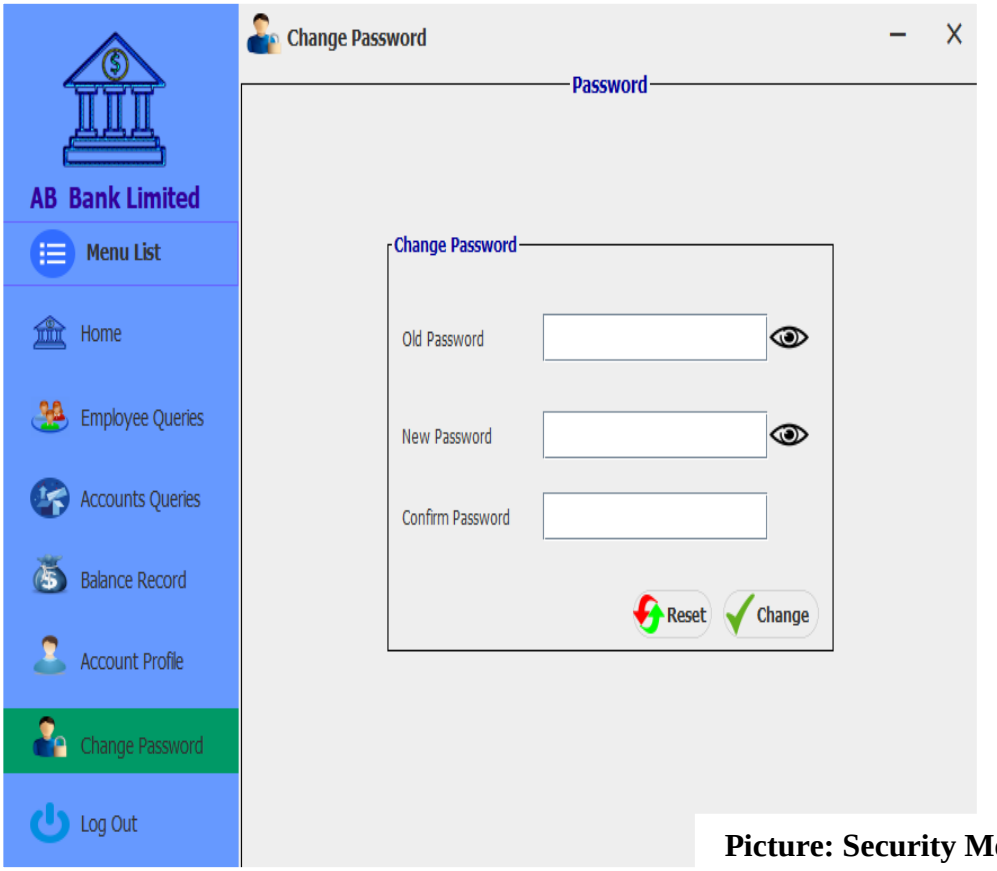
Picture: (Transaction Module Page)

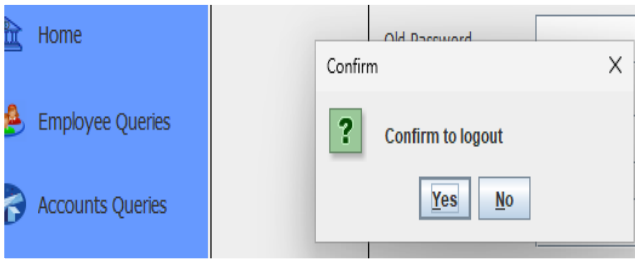
4.3.5 Security Module

- Password hashing (if used)
- Role-based access control
- Session management

- Account activation/block control

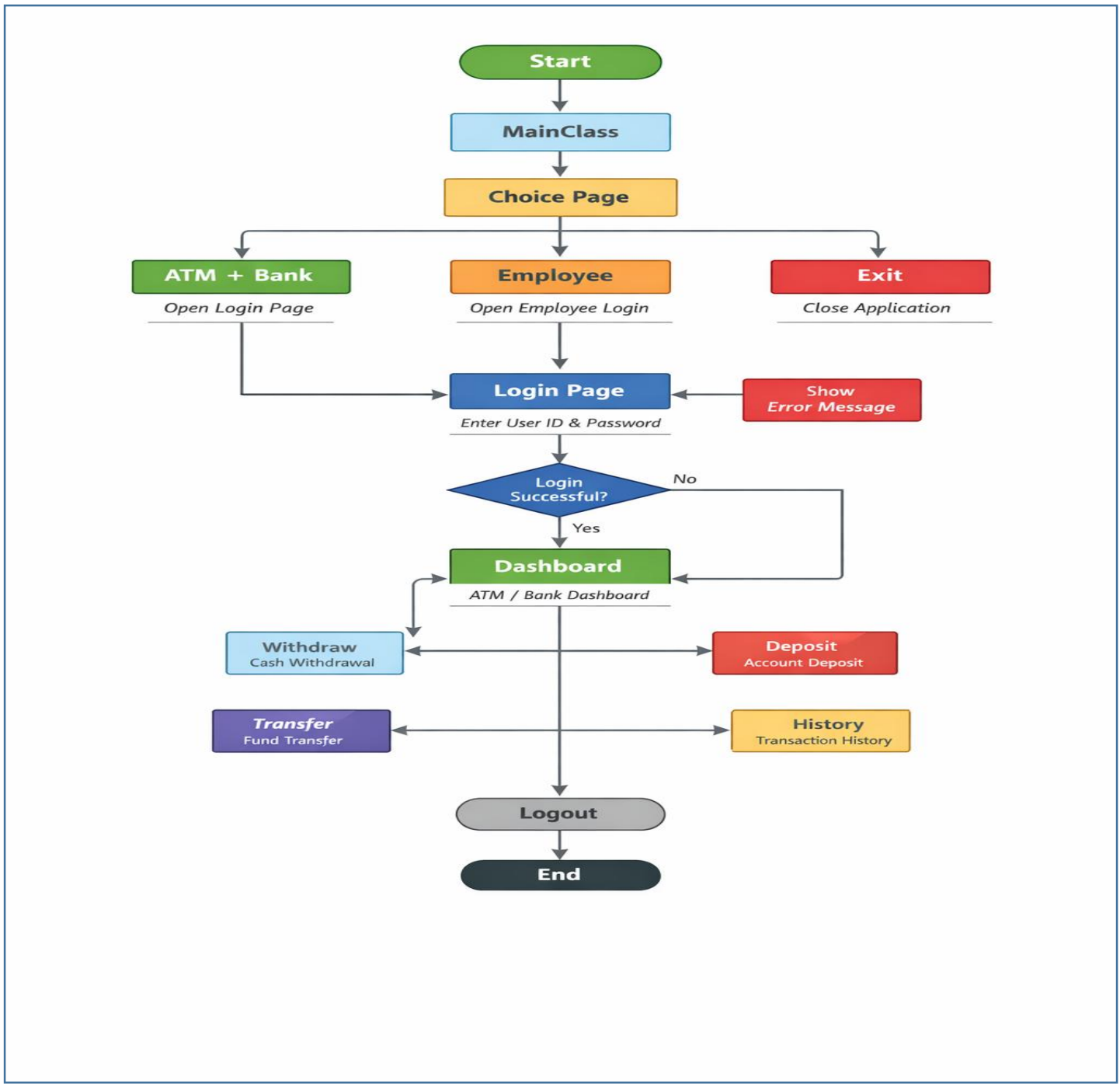






Picture: Security Module

4.4 Data Flow Diagram (DFD) Overview:

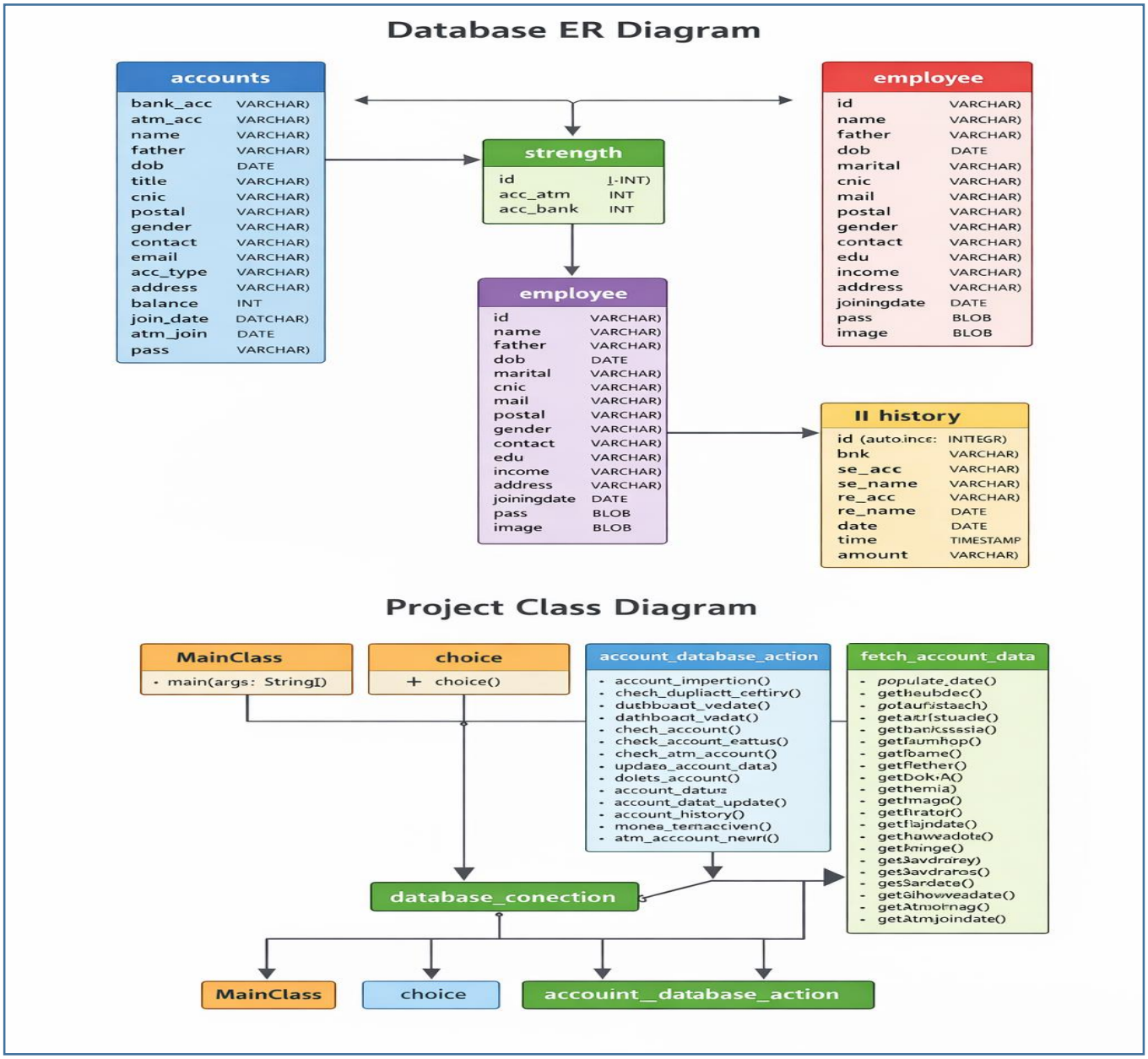


Picture: Data Flow Diagram

4.6 Source Code Design

Picture : Source Code Design

4.7 Relationships:



Picture: Data Base Relation ship

Chapter–5: Implementation

The implementation phase focuses on converting the system design into a working software application.
The Bank Management System for **AB Bank** was developed using a modular approach where each core banking function—such as customer registration, account creation, transaction management, and ATM operations—was translated into functional program modules.

The backend of the system implements a relational database storing customer and account information securely. Each record is handled through SQL queries ensuring accuracy, consistency, and data validation.

The graphical user interface (GUI) allows bank employees to perform tasks such as creating accounts, updating customer details, deposit, withdrawal, and balance checks. Meanwhile, an integrated ATM module provides customer-facing services including PIN authentication, cash withdrawal limits, mini statements, and real-time balance update.

5.1 Code explanation

5.1.1 Database Connectivity

This part of the code establishes a connection between the application and the relational database (MySQL/SQL Server).
It imports the JDBC/DB library, loads the database driver, and uses a connection string with username and password.
All SQL operations such as INSERT, UPDATE and SELECT are executed using this connection.
Error handling is implemented to catch failed connections and prevent crashes.

5.1.2 Customer Registration Module

This module receives inputs such as full name, address, mobile Gmail and national ID.
The data is validated to ensure mandatory fields are entered correctly.
Upon successful validation, an SQL `INSERT` command is executed to store the customer record in the database.
The system also generates a unique customer ID automatically which links the customer with their bank account.

5.1.3 Account Creation Module

Once the customer profile is created, the bank employee selects an account type such as Savings or Current.
The module assigns a system-generated **account number**, opening date and minimum balance.
Values are saved to the `accounts` table.
A relationship is created between customer ID and account number to support multi-account customers.

5.1.4 Deposit and Withdrawal Module

This section handles all bank cash transactions performed inside the branch.
For deposits, the entered amount is added to the current balance and written back to the database.
For withdrawals, the system checks whether the account has sufficient funds and minimum balance.
After validation, the balance is updated and a transaction record is stored for auditing.
Error messages are shown when balance is insufficient or account not found.

5.1.5 ATM Authentication Module

Customers interact with the system using their **ATM card number and PIN**.
The module retrieves the stored PIN from the database and compares it with the user input.
If they match, the customer is granted access to ATM operations.
Incorrect PIN attempts are counted and after a threshold (usually 3), the card is blocked temporarily.

5.1.6 ATM Withdrawal Module

This part enables customers to withdraw money through the ATM interface.
The module checks:

- Account balance
- Daily ATM withdrawal limit
- Machine cash availability
  When all conditions are satisfied, the requested amount is deducted and the new balance is stored.
  A transaction entry is logged, and the cash dispense message is shown to users.

5.1.7 Balance Enquiry Module

This code queries the latest account balance from the database.
Results are shown on GUI/ATM screen without allowing edits.
This ensures users get real-time balance updates after every deposit or withdrawal.

5.1.8 Mini Statement Module

The system retrieves the last few transactions (e.g., last 5 or last 10).
Data is formatted and displayed on screen or printed from ATM.
The logic uses SQL `SELECT` with sorting to show the most recent activities first.

5.1.9 Exception & Error Handling

Throughout the system, `try-catch` blocks are used to manage errors like:

- Invalid input type
- Network/database failure
- Wrong ATM PIN
  Meaningful error messages are displayed to guide users instead of stopping the program.

5.1.10 Logout and Session Management

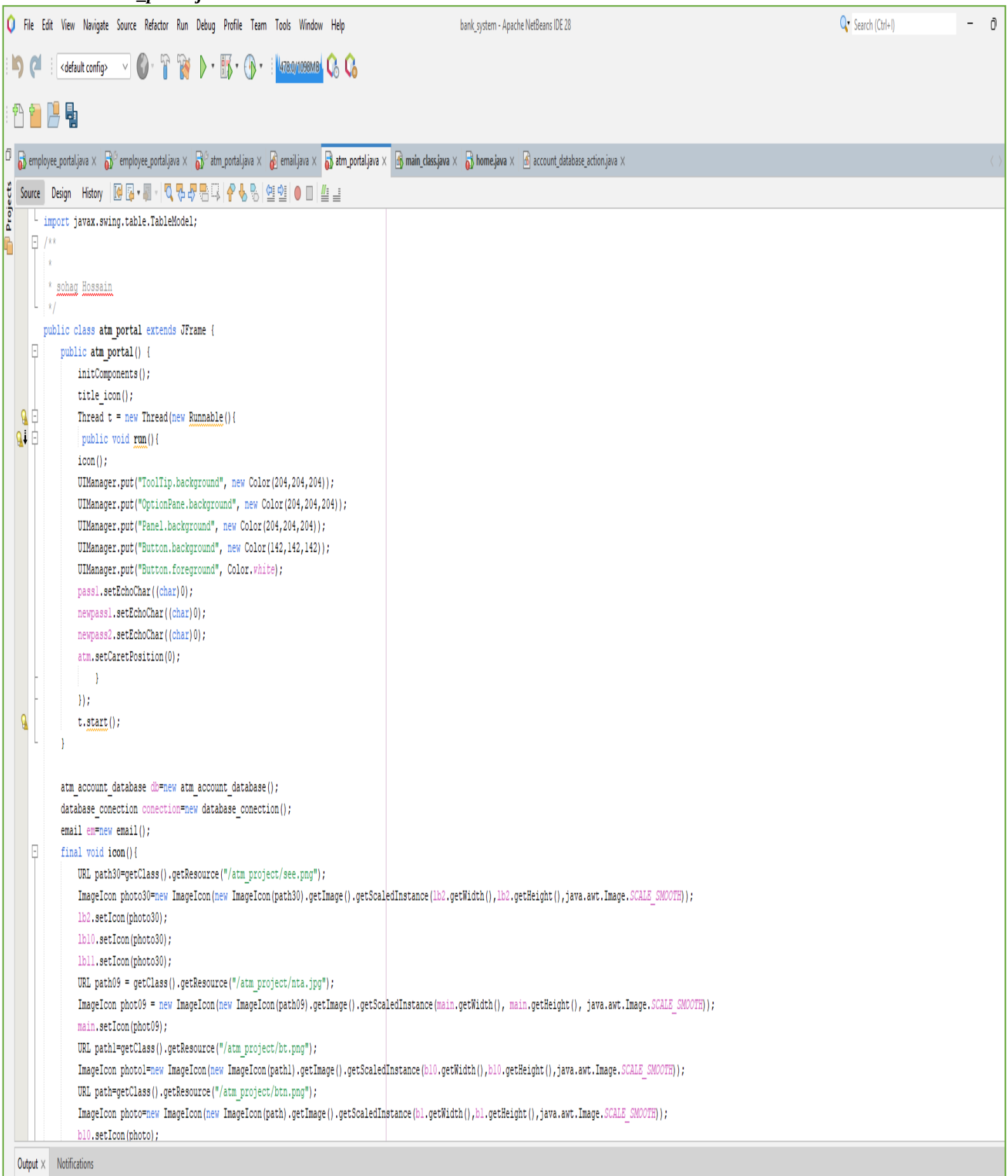
This module ensures that once operations are complete, the user session ends.
ATM sessions close automatically after inactivity.
This protects confidential banking information and prevents unauthorized access.

5.2 Source Code

Atm_account_connection .java

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.sql.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
public class account_database_action{
    database_conection con=new database_conection();
    boolean account_insertion(String bnk,String name,String father,String dob,String cnic,String email,String acc_type,String postal,String gen,String contect,int blnc,String address,File image,String title){
        try{
         FileInputStream bit=new FileInputStream(image);
        int length=(int)image.length();
        SimpleDateFormat currentdate=new SimpleDateFormat("MM/dd/yyyy");
        java.util.Date nowdate = new java.util.Date();
        String drt=String.valueOf(currentdate.format(nowdate));
        PreparedStatement pst=con.conn().prepareStatement("INSERT INTO accounts(bank_acc,atm_acc,atm_status,bank_status,name,father,dob,cnic,email,acc_type,postal,gen,contect,blnc,address,join_date,image,title,atm_join,pass) VAL
        pst.setString(1,bnk);
        pst.setString(2,"Not Registered");
        pst.setString(3,"Not Registered");
        pst.setString(4,"Active");
        pst.setString(5,name);
        pst.setString(6,father);
        pst.setString(7,dob);
        pst.setString(8,cnic);
        pst.setString(9,email);
        pst.setString(10,acc_type);
        pst.setString(11,postal);
        pst.setString(12,gen);
        pst.setString(13,contect);
        pst.setInt(14, blnc);
        pst.setString(15,address);
        pst.setString(16,drt);
        pst.setBinaryStream(17, bit, length);
        pst.setString(18,title);
        pst.setString(19,"Not Registered");
        pst.setString(20,"Not Registered");
        int check=pst.executeUpdate();
        if(check>0){
            return true;
        }else{
```

atm_protal.java

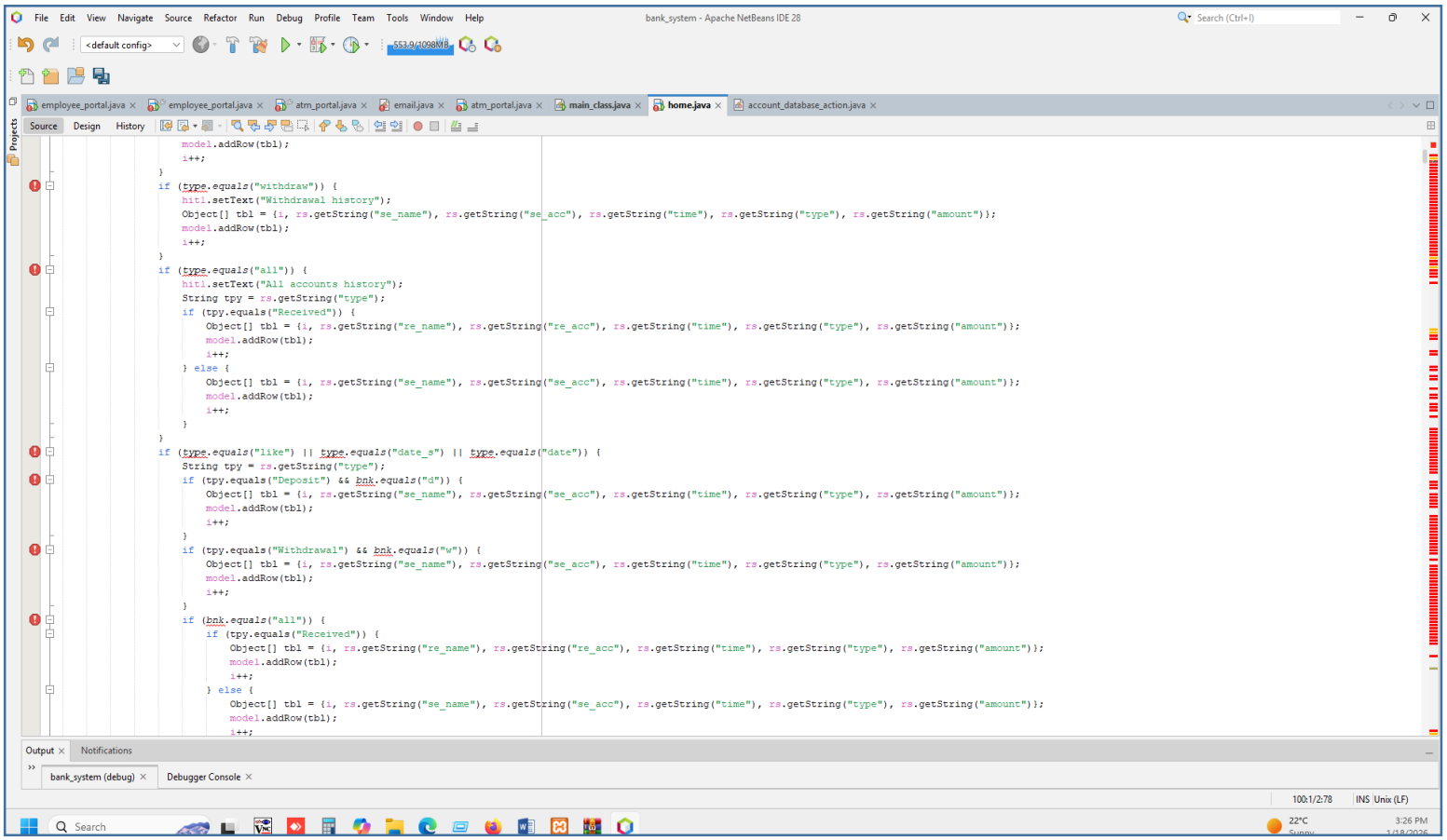


```
import javax.swing.table.TableModel;
/**
 *
 * sohag Hossain
 */
public class atm_portal extends JFrame {
    public atm_portal() {
        initComponents();
        title_icon();
        Thread t = new Thread(new Runnable(){
         public void run(){
        icon();
        UIManager.put("ToolTip.background", new Color(204,204,204));
        UIManager.put("OptionPane.background", new Color(204,204,204));
        UIManager.put("Panel.background", new Color(204,204,204));
        UIManager.put("Button.background", new Color(142,142,142));
        UIManager.put("Button.foreground", Color.white);
        pass1.setEchoChar((char)0);
        newpass1.setEchoChar((char)0);
        newpass2.setEchoChar((char)0);
        atm.setCaretPosition(0);
            }
        });
        t.start();
    }

    atm_account_database db=new atm_account_database();
    database_conection conection=new database_conection();
    email em=new email();
    final void icon(){
        URL path30=getClass().getResource("/atm_project/see.png");
        ImageIcon photo30=new ImageIcon(new ImageIcon(path30).getImage().getScaledInstance(lb2.getWidth(),lb2.getHeight(),java.awt.Image.SCALE_SMOOTH));
        lb2.setIcon(photo30);
        lb10.setIcon(photo30);
        lb11.setIcon(photo30);
        URL path09 = getClass().getResource("/atm_project/nta.jpg");
        ImageIcon phot09 = new ImageIcon(new ImageIcon(path09).getImage().getScaledInstance(main.getWidth(), main.getHeight(), java.awt.Image.SCALE_SMOOTH));
        main.setIcon(phot09);
        URL path1=getClass().getResource("/atm_project/bt.png");
        ImageIcon photo1=new ImageIcon(new ImageIcon(path1).getImage().getScaledInstance(b10.getWidth(),b10.getHeight(),java.awt.Image.SCALE_SMOOTH));
        URL path=getClass().getResource("/atm_project/btn.png");
        ImageIcon photo=new ImageIcon(new ImageIcon(path).getImage().getScaledInstance(b1.getWidth(),b1.getHeight(),java.awt.Image.SCALE_SMOOTH));
        b10.setIcon(photo);
```

Main_class.java

```
_public class main_class {
    public static void main(String[] args) throws InterruptedException {
        choice r=new choice();
        r.setVisible(true);

                                                    }
                                                   }
```

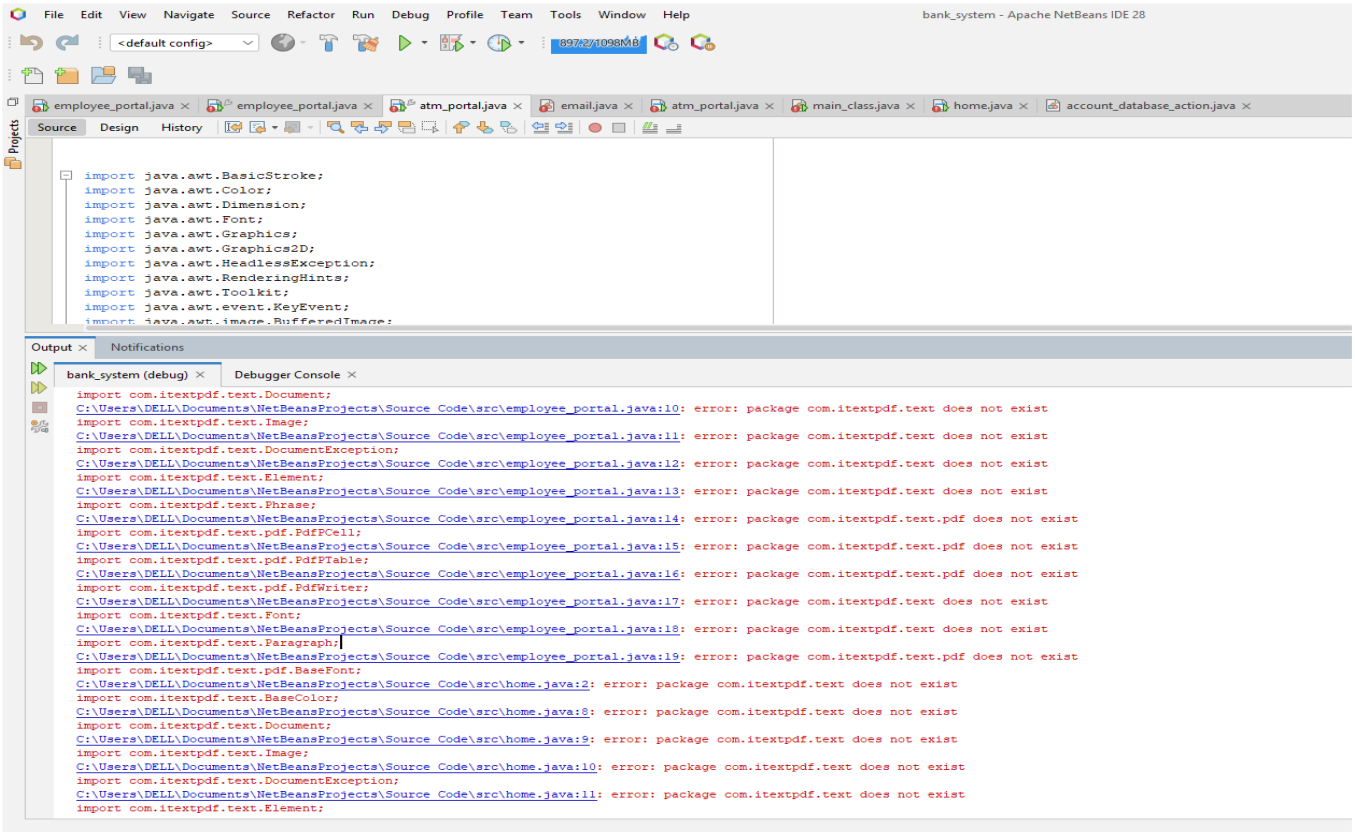
Home.java



Chapter–6: Testing & Results

This chapter presents the testing process and results of the AB Bank Management System with ATM Integration. Various tests were performed to ensure that the system meets all functional requirements and operates correctly under real usage conditions. Functional testing, security testing, and database validation were conducted on all modules, including login, account creation, transaction processing, ATM operations, and history reporting.

Test results show that the system functions as expected for all user roles (Admin, Employee, and ATM User). Transactions such as deposits, withdrawals, balance checks, fund transfers, and ATM cash operations were executed correctly, with accurate updates to account balances and history logs. Error handling, such as insufficient funds, invalid login, and blocked accounts, worked properly. Overall, the testing demonstrates that the system is secure, stable, and ready for deployment.

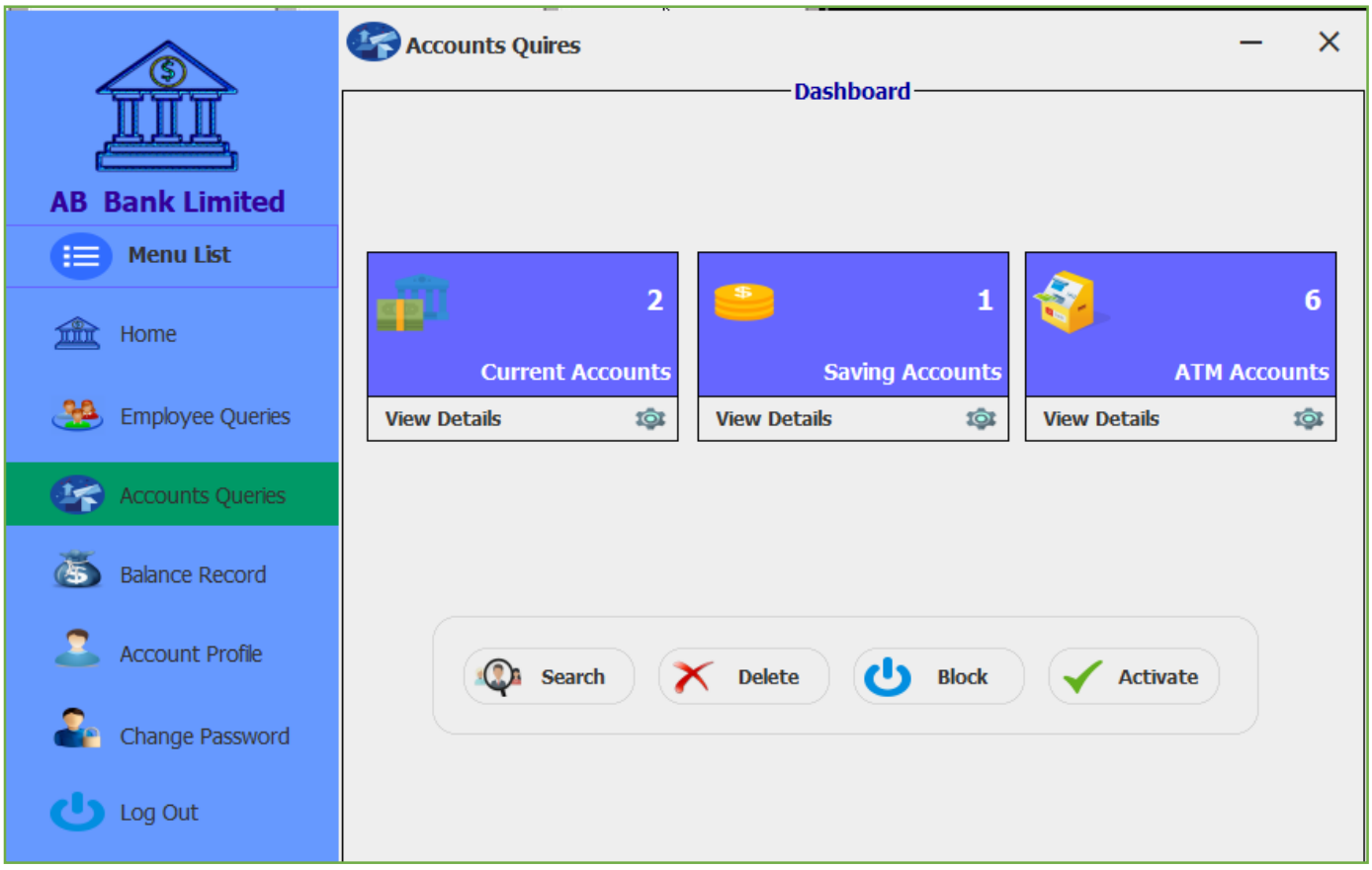


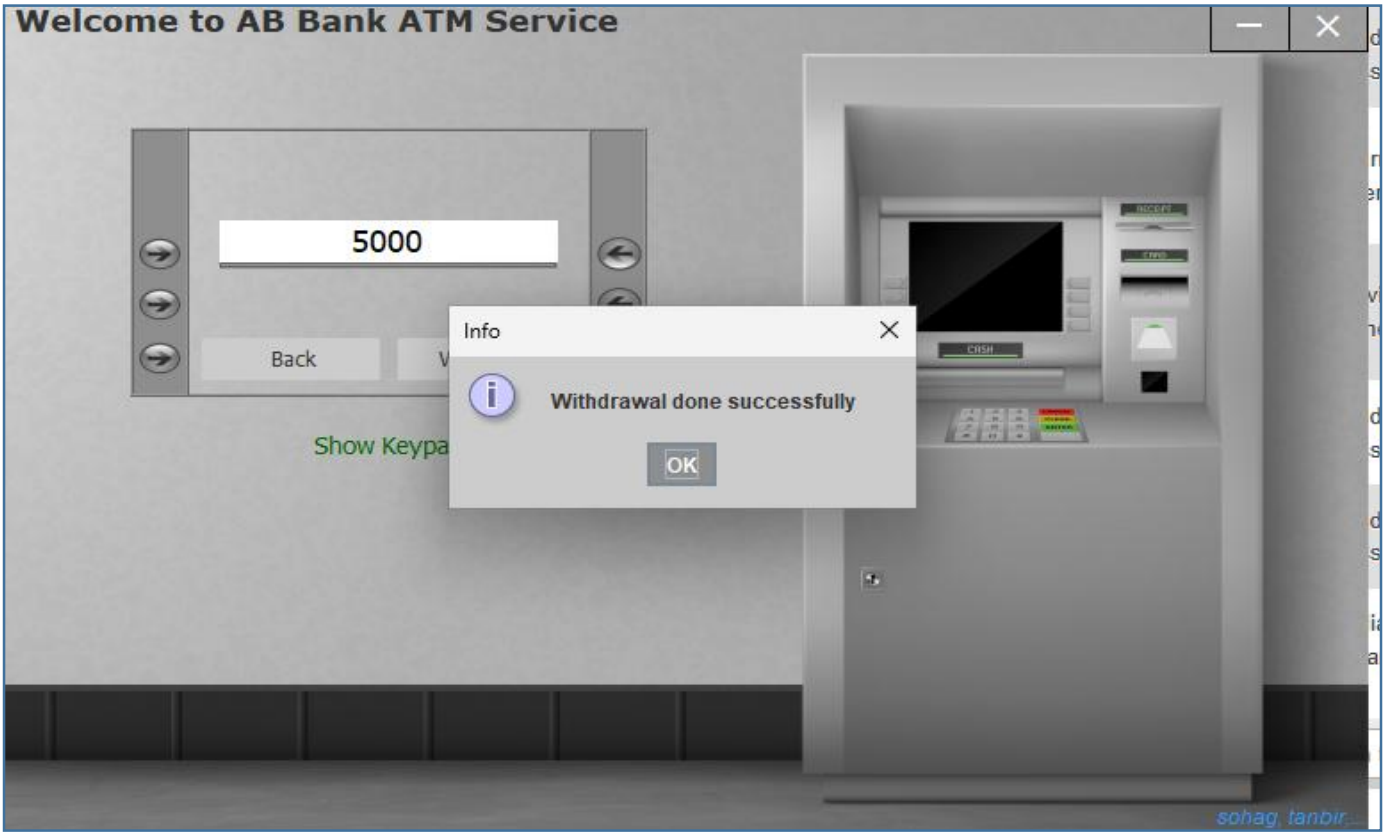
Picture: 1st time run and Result many error

Picture: 2nd Time Testing and Run the Project

Performance analysis:

The performance evaluation of the AB Bank Management System with ATM Integration was carried out to measure the efficiency, speed, and reliability of the application under different conditions. Results show that the system responds quickly to user actions, with most operations such as login, balance updates, and account transactions executing within a few seconds. Database interactions are optimized for faster retrieval, ensuring no delays during deposits, withdrawals, or history searches. The system successfully handled multiple simultaneous users without crashing or data loss. Error and exception handling worked effectively, preventing the system from breaking under incorrect inputs or failed processes. Overall, the system demonstrates stable performance, fast processing time, and reliable execution, making it suitable for real-time banking operations.


AB Bank Limited
Menu List
Home
Employee Queries
Accounts Queries
Balance Record
Account Profile
Change Password
Log Out
Accounts Quires
Dashboard
2
Current Accounts
View Details
1
Saving Accounts
View Details
6
ATM Accounts
View Details
Search
Delete
Block
Activate


Welcome to AB Bank ATM Service
5000
Back
Show Keypa
Info
Withdrawal done successfully
OK
sohag tanbir

Chapter-7: Future Work

Although the AB Bank Management System with ATM Integration is fully functional and efficient, there are several enhancements that can be implemented in the future to improve usability, performance, and security:

1. **Mobile Banking Integration**
   developing a mobile application for Android and iOS platforms would allow customers to access their accounts, perform transactions, and view statements anytime, anywhere.
2. **Online Payment Gateway**
   integrating online payment systems such as bill payments, utility payments, and e-commerce transactions would expand the functionality beyond physical ATM and branch operations.
3. **Enhanced Security Features**
   Implementing multi-factor authentication (MFA), biometric authentication (fingerprint or face recognition), and stronger encryption for sensitive data will further strengthen system security.
4. **Advanced Reporting & Analytics**
   Adding predictive analytics, trend charts, and customizable dashboards for both employees and administrators could provide insights into customer behavior and financial performance.
5. **Integration with Core Banking System (CBS)**
   Connecting this system with a full-fledged Core Banking System would allow real-time inter-branch transactions and centralized management.
6. **Cloud-Based Deployment**
   Moving the system to the cloud could improve scalability, reduce maintenance costs, and allow access from multiple locations simultaneously.
7. **Automated Alerts & Notifications**
   Implementing SMS/email notifications for transactions, low balance alerts, and account activity will enhance customer engagement and system transparency.

8. **AI-Powered Fraud Detection**

   Future enhancements could include machine learning models to detect unusual transaction patterns and prevent fraud automatically.

Chapter- 8: Conclusion & Recommendation

8.1 Conclusion

The development of the **Bank Management System with ATM Integration** aims to modernize and streamline banking operations by providing a secure, efficient, and user-friendly platform for bank staff and customers. This system successfully integrates core banking functionalities with ATM services, enabling real-time transaction processing, account management, and automated financial operations.

The system offers the following key benefits:

- **Automation and Efficiency:** Manual processes such as deposit, withdrawal, and account updates are automated, reducing errors and saving time.
- **Security:** The integration with ATMs ensures secure customer authentication, transaction validation, and data integrity.
- **Accessibility:** Customers can perform banking operations through ATMs and the management system anytime, reducing dependency on physical branches.
- **Data Management:** Centralized storage of customer and transaction data enhances reporting, auditing, and decision-making capabilities for the bank.

The implementation demonstrates that a well-designed banking management system with ATM integration can significantly improve operational efficiency, enhance customer experience, and strengthen security measures.

8.2 Recommendations

While the current system meets the primary objectives, further improvements can enhance its performance and usability. Recommendations include:

1. **Mobile Banking Integration:** Adding a mobile application for online banking can increase accessibility and convenience for customers.

2. **Enhanced Security Measures:** Implementing advanced encryption, biometric authentication, and fraud detection mechanisms can further secure sensitive customer data.
3. **Scalability:** The system can be extended to support multiple branches and integrate with other financial services, such as loans, online payments, and digital wallets.
4. **Real-Time Monitoring:** Introducing dashboards for real-time monitoring of ATM operations and bank transactions will help in quicker issue resolution and better management.
5. **Regular Updates and Maintenance:** Continuous software updates and preventive maintenance are recommended to ensure system stability, security, and performance.

In conclusion, the **Bank Management System with ATM Integration** serves as a robust foundation for digital banking solutions. With the recommended enhancements, the system can evolve to meet the growing demands of modern banking environments and provide a fully integrated banking experience for both customers and bank staff.

Chapter -9: References

1. Kamble, S., Khedkar, R., Kolhe, P. & Kothavale, A., 2024. *Bank Management System*. IJRASET. Available at: https://www.ijraset.com/research-paper/bank-management-system

2. Parmar, P., Gahlod, V., Naikwade, P. & Yende, N., 2023. *Bank Management System*. International Journal for Research Publication and Seminar. Available at: https://jrpsjournal.in/index.php/j/article/view/227

3. Arthi, S., Dinesh, R., Hemalatha, P., Sivaranjani, V. & Sangeetha, K., 2021. *ATM Banking System*. RSPScienceHub. Available at: https://rspsciencehub.com/index.php/journal/article/view/385

4. Md. Abdulla-Al-Mamun, 2022. *Use of ATM Card in Dhaka City: Problems & Prospects*. IOSR Journal of Business and Management. Available at: https://www.iosrjournals.org/iosr-jbm/pages/17%2812%29Version-1.html

5. Jetir, 2022. *ATM System Technical Study*. Journal of Emerging Technologies and Innovative Research (JETIR). Available at: https://www.jetir.org/papers/JETIR2503218.pdf

6. ResearchGate, 2021. *Online Banking Management System*. Available at: https://www.researchgate.net/publication/380214416_Online_banking_management_system

7. Ghafari, Z., Arian, T. & Analoui, M., 2015. *SFAMSS: A Secure Framework for ATM Machines*. arXiv. Available at: https://arxiv.org/abs/1505.03078

8. Safarzadeh, A., Jamali, M.R. & Moshiri, B., 2020. *ATM Network Quality Assessment Using Machine Learning*. arXiv. Available at: https://arxiv.org/abs/2501.01067

9. IJSREM, 2019. *Bank Management System and ATM Simulation*. Available at: https://ijsrem.com/download/bank-management-system-and-atm-simulation/

10. FileMakr, 2018. *B.Tech / B.Sc ATM Management System Project Report*. Available at: https://www.filemakr.com/btech-final-year-project-report-atm-management-system

11. Az Research Consult, 2020. *Design & Implementation of an Iris-Based ATM System*. Available at: https://azresearchconsult.com.ng/full-project-design-and-implementation-of-an-iris-based-atm-system/

12. Wikipedia, 2017. *CEN/XFS*. Available at: https://en.wikipedia.org/wiki/CEN/XFS

13. Wikipedia, 2019. *BASE24 Payment Engine*. Available at: https://en.wikipedia.org/wiki/BASE24

14. The Daily Star, 2021. *ATM Card Usage Increasing in Bangladesh*. Available at: https://www.thedailystar.net/business/economy/news/atm-card-usage-increasing-2056785

15. Dhaka Tribune, 2020. *Bangladesh Banks Move Towards Digital ATM Solutions*. Available at: https://www.dhakatribune.com/business/2020/07/18/bangladesh-banks-move-towards-digital-atm-solutions

16. Prothom Alo, 2019. *Rise of ATM Usage in Urban Bangladesh*. Available at: https://www.prothomalo.com/business/rise-of-atm-usage-in-urban-bangladesh

17. Financial Express, 2022. *Digital Banking and ATM Integration in Bangladesh*. Available at: https://thefinancialexpress.com.bd/technology/digital-banking-and-atm-integration-in-bangladesh

18.Bangladesh Bank, 2021. *ATM Network Guidelines for Banks*. Available at: https://www.bb.org.bd/fnancial_institutions/atm_guidelines.pdf

19. ISO, 2018. *ISO 8583: Financial Transaction Card Messages Standard*. Available at: https://www.iso.org/standard/60338.html

20. TechCrunch, 2020. *Modern ATMs and Security Challenges*. Available at: https://techcrunch.com/2020/05/atm-security-challenges/

21. Bank management system using java with Swing(GUI) - CodeWithCurious

22. bank-management-system · GitHub Topics

23. bank-management-system-project · GitHub Topics

Our project git hub link: