

INTERNATIONAL UNIVERSITY
VIETNAM NATIONAL UNIVERSITY – HCMC



Final Report - Group 7

Instructor: *Nguyen Thuy Loan*

Course: *Principles of Database Management*

TOPIC #30 - GROUP 7:
ONLINE BANKING SYSTEM DATABASE

Members:

Huỳnh Phương Đại - ITITWE21039

Nguyễn Anh Trí - ITCSIU22152

Nguyễn Huỳnh Minh Đức - ITCSIU22218

Châu Thịnh - ITCSIU22240

Trương Quang Tường - ITITI20130

Nguyễn Trọng Thuận - ITCSIU22223

Contents

CHAPTER 1/ PROJECT DESCRIPTION:	3
1.1 INTRODUCTION	3
1.2 LITERATURE REVIEW	3
1.3 METHODOLOGY	4
1.4 INITIAL APPROACH	4
Entities:	4
Functions:	5
1.5 RESULT	7
1.6 PROJECT DEMONSTRATION	8
1.7 DISCUSSION	17
1.8 CONCLUSION	18
CHAPTER 2/ PROJECT TIMELINE:	18
CHAPTER 3/ PROJECT ANALYSIS:	19
3.1 DIAGRAMS SHOWING RELATIONSHIPS FOR ONLINE BANKING DATABASE [3]	20
3.2 PRIMARY ENTITIES	21
3.3 SECONDARY ENTITIES	22
3.4 DATABASE AND TABLE CREATION	23
3.5 NORMALIZATION	23
CHAPTER 4/ FUTURE PLANS:	26
1. Security	26
2. Interface	27
REFERENCES:	27

LIST OF FIGURES

Figure 1. Register page for creating a new bank account.....	10
Figure 2. Login Screen.....	11
Figure 3. App interface after logging in.....	12
Figure 4. Account information screen.....	13
Figure 5 Transition interface.....	14
Figure 6 Customer information interface.....	15
Figure 7 Transition history display.....	16
Figure 8 Saving Account Information's screen.....	17
Figure 9 Create a saving account by inputting the amount of money and choosing duration.....	17
Figure 10 Invalid transaction occurs when input invalid amount.....	18
Figure 11. Transaction successful notification.....	19
Figure 12.Login success announcement.....	20
Figure 12: Project development timeline.....	22
Figure 13: ERD shows the relationship between online banking.....	23
Figure 14: Schema diagram converted from the previous ERD.....	24

Abstract

The "Online Banking System Database" project, developed by a team of pre-graduate students under Dr. Nguyen Thi Thuy Loan's guidance focuses on establishing a secure, efficient database for modern banking operations. Prioritizing the safeguarding of customer credentials and account information, it addresses critical aspects of digital banking security and efficiency. This academic initiative, intended for educational purposes, aims to underpin online banking with a robust, secure system, facilitating safe account access and financial transactions. The report encompasses a comprehensive analysis, system design, and prospective enhancements, reflecting the collaborative educational endeavor of the group.

We express our sincere gratitude to the Teacher Assistant team for their invaluable guidance and insightful responses during the development of our system, which significantly contributed to its successful completion.

CHAPTER 1/ PROJECT DESCRIPTION:

1.1 INTRODUCTION

The Online Banking System Database is a cornerstone of contemporary banking operations. It serves as a secure storage and authentication system for customer login credentials and account information. This database is instrumental in maintaining the confidentiality, integrity, and availability of digitized banking services.

1.2 LITERATURE REVIEW

The project "Online Banking System Database" aims to create a secure and efficient database for managing online banking operations. It includes entities like customers, accounts, transactions, and savings accounts, and functions for adding customers and accounts, handling transactions, and managing savings accounts. The project also involves normalizing the database to the Fourth Normal Form (4NF) to reduce redundancy and improve data integrity. Plans include enhancing security features, improving the user interface, and implementing new functionalities for a better customer experience. This comprehensive approach ensures the reliability and security of the online banking system.

1.3 METHODOLOGY

- a. Entity Analysis: Identifying primary entities (Customer, Account, Transaction, savings account, etc.) and their attributes.
- b. Function Implementation: Developing functions for operations like adding customers, accounts, and transactions, and managing customer-account relationships.
- c. Database Design: Utilizing Microsoft SQL Server 2014 for database design and data insertion.
- d. Normalization Process: Applying normalization principles up to the Fourth Normal Form (4NF) to ensure data integrity and reduce redundancy.
- e. System Testing: Conducting software testing for bug identification and fixing.

1.4 INITIAL APPROACH

Entities:

To sum up, five entities and seven functions are primarily focused on in this project to provide secure access to customers' accounts and empower them to conduct financial transactions safely.

1. **Customer:** *This table will store information about the bank's customers, such as their name, address, phone number, email address, and other relevant details.*
2. **Account:** *This table will store information about the bank accounts held by each customer, such as the account number, account type, balance, and other relevant details.*
3. **Transaction:** *This table will store information about all transactions on the bank accounts, such as the date, time, amount, and other relevant details.*
4. **Saving Account:** *This table will store information about the saving account containing the information of date, amount, rate, and duration*
5. **Customer Account:** *This table will store the information between the Customer table and the Account table based on Customer_id and Account_No*

Functions:

1. **Add Customer:** *This function will allow a bank employee to add a new customer to the system. The function will require the following parameters:*
 - a. *CustomerID: INT IDENTITY(100,1) (PRIMARY KEY)*
 - b. *Gender: VARCHAR(10)*
 - c. *City: VARCHAR(225)*
 - d. *Street: VARCHAR(225)*
 - e. *Email: VARCHAR(225)*
 - f. *Date_of_birth: DATE*
 - g. *Fullname: VARCHAR(225)*
 - h. *Contact_number: VARCHAR(20)*
 - i. *PIN: VARCHAR(6)*

2. *Add Account: This function will allow a bank employee to add a new account for a customer. The function will require the following parameters:*
 - a. *Account_No: INT IDENTITY(1,1) (PRIMARY KEY)*
 - b. *CustomerID: INT (FOREIGN KEY REFERENCES Customer(CustomerID))*
 - c. *Balance: DECIMAL(10,2)*
 - d. *Open_date: DATE*
3. *Add new Transaction: This function will allow a bank employee to add a new transaction to another's account. The function will require the following parameters:*
 - a. *Transaction_id: INT IDENTITY(1,1) (PRIMARY KEY)*
 - b. *Customer_id: INT(FOREIGN KEY REFERENCES Customer(CustomerID))*
 - c. *Amount: DECIMAL(10,2)*
 - d. *Transaction_date: DATE*
4. *Add CustomerAccount: This function will allow the bank employee to create an Account Number for the customer through Customer ID.*
 - a. *CustAcc_id: IDENTITY(1,1) PRIMARY KEY*
 - b. *Account_No(FOREIGN KEY REFERENCES Accounts(Account_No))*
 - c. *CustomerID:INT(FOREIGN KEY REFERENCES Customer(CustomerID))*

CONSTRAINT UC_CustomerAccount UNIQUE (Customer_id, Account_No)
5. *Add Saving Account: This function will allow a bank employee to add a new savings account to a customer's account. The function will require the following parameters:*
 - a. *Saving_Acc_id: INT IDENTITY (1,1)(PRIMARY KEY)*
 - b. *Initial_Amount: MONEY*
 - c. *Interest_rate: FLOAT*
 - d. *Open_date: DATE*
 - e. *Duration: Time*
 - f. *Account_No: INT (FOREIGN KEY REFERENCES Accounts(Account_No))*

6. *Create SavingsAccount: This function will allow a customer to enter their information and create a Savings account on their own. A Customer can only create and maintain one account at a time.*

7. *Check SavingsAccount: This function will allow the system to check for duplication in case more than one Savings account was assigned to a Customer in the same time frame.*

1.5 RESULT

- a. Implementation Success: The developed system serves as a foundational platform for future enhancements, adhering rigorously to the principles of object-oriented programming and object-oriented design and analysis. Consequently, the system incorporates essential functionalities, including account login, account inquiries, and monetary transactions, establishing a robust base for further functional expansion and refinement. This approach underscores the system's adaptability and scalability in meeting evolving banking requirements.
- b. Performance metrics: During the software testing phase, comprehensive evaluations were conducted to ensure robust system performance. These evaluations included tests for empty and invalid inputs, resistance to various data type injections, and checks for data inconsistencies and duplications. This rigorous testing regimen was crucial in refining the system, ensuring its smooth operation post-deployment. While acknowledging the potential existence of residual errors and bugs, the team has diligently worked within the project's timeline to deliver a functional and reliable product.
- c. Challenges and Resolutions: The development cycle has no doubt run into several problems and challenges. Problems range from minor software discrepancies such as trying to work on the same project without using GitHub to a plethora of plugin errors and version mismatches of the plugins.

Though much has been resolved there are still more unsolved problems such as password hashing algorithms and database linkages on other member's computers.

- d. Comparison with Initial Goals: Initially, the project's objective was centered on developing a functional user interface (UI) capable of extracting specific metrics and data from the database. However, over the course of eight lab sessions and nine class meetings, the scope of the project was substantially broadened. The system's capabilities were enhanced to include multiple interrelated windows, aligning more closely with the functionalities found in existing commercial Internet banking applications. This expansion reflects the project's evolution and adaptation to encompass a more comprehensive and market-aligned set of features similar to that of actual online banking applications.

1.6 PROJECT DEMONSTRATION

□ PDM project

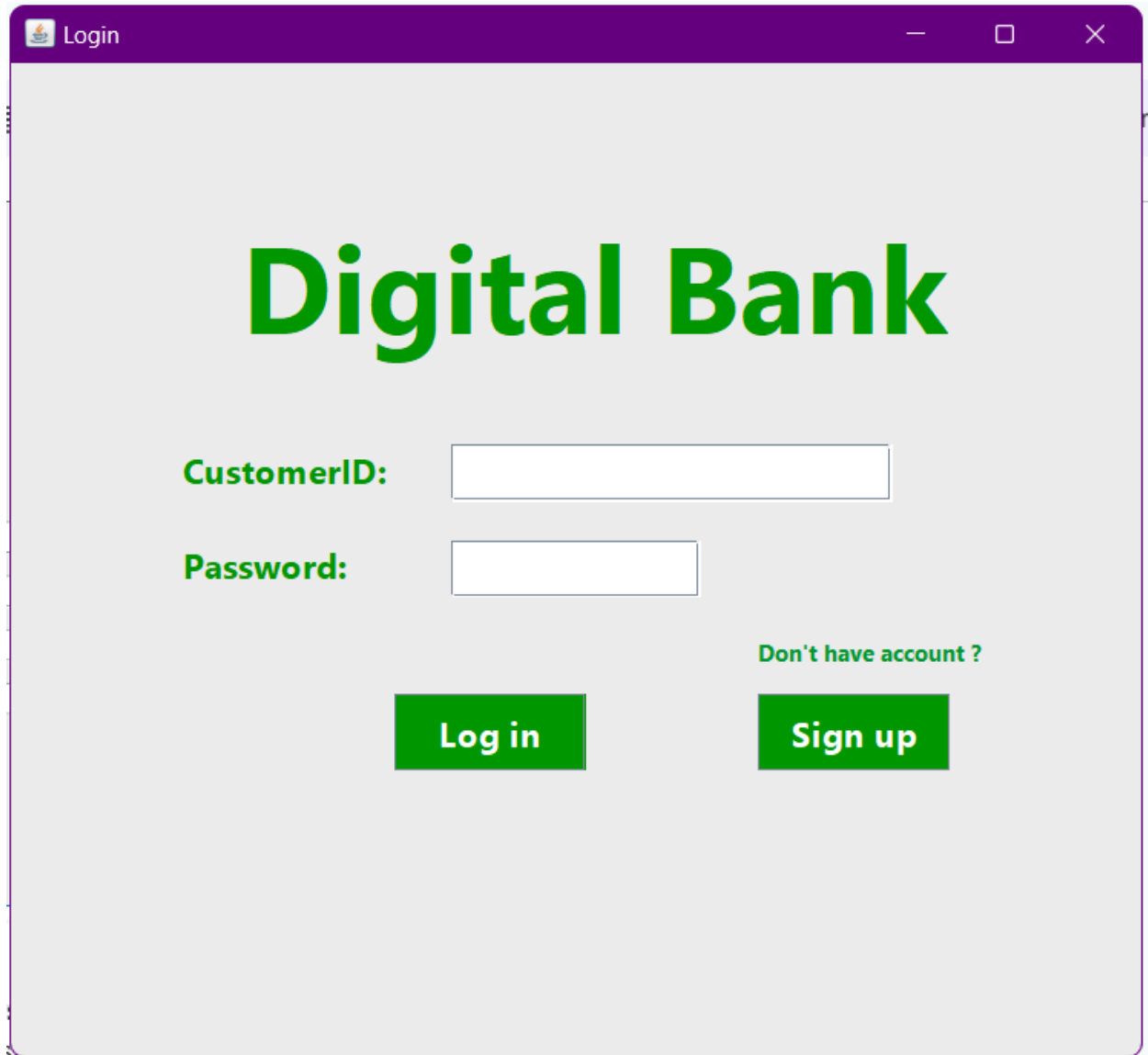
Sign Up

Register Page

Name :*	<input type="text"/>	Date of birth: *	<input type="text"/>
Email: *	<input type="text"/>	Gender: *	<input type="text" value="Select Gender"/>
Password *	<input type="text"/>	Contact Number: *	<input type="text"/>
Verify Password*	<input type="text"/>	City: *	<input type="text"/>
		Street : *	<input type="text"/>

[Already have an account? Log in](#)

Figure 1. Register page for creating a new bank account



The image shows a login window titled "Login" with a purple header bar. The main content area is light gray. At the top, the text "Digital Bank" is displayed in a large, bold, green font. Below this, there are two input fields: "CustomerID:" followed by a white text box, and "Password:" followed by a white text box. To the right of the password field, there is a green link that says "Don't have account ?". At the bottom, there are two green buttons: "Log in" on the left and "Sign up" on the right.

Login

Digital Bank

CustomerID:

Password:

[Don't have account ?](#)

Log in **Sign up**

Figure 2. Login Screen

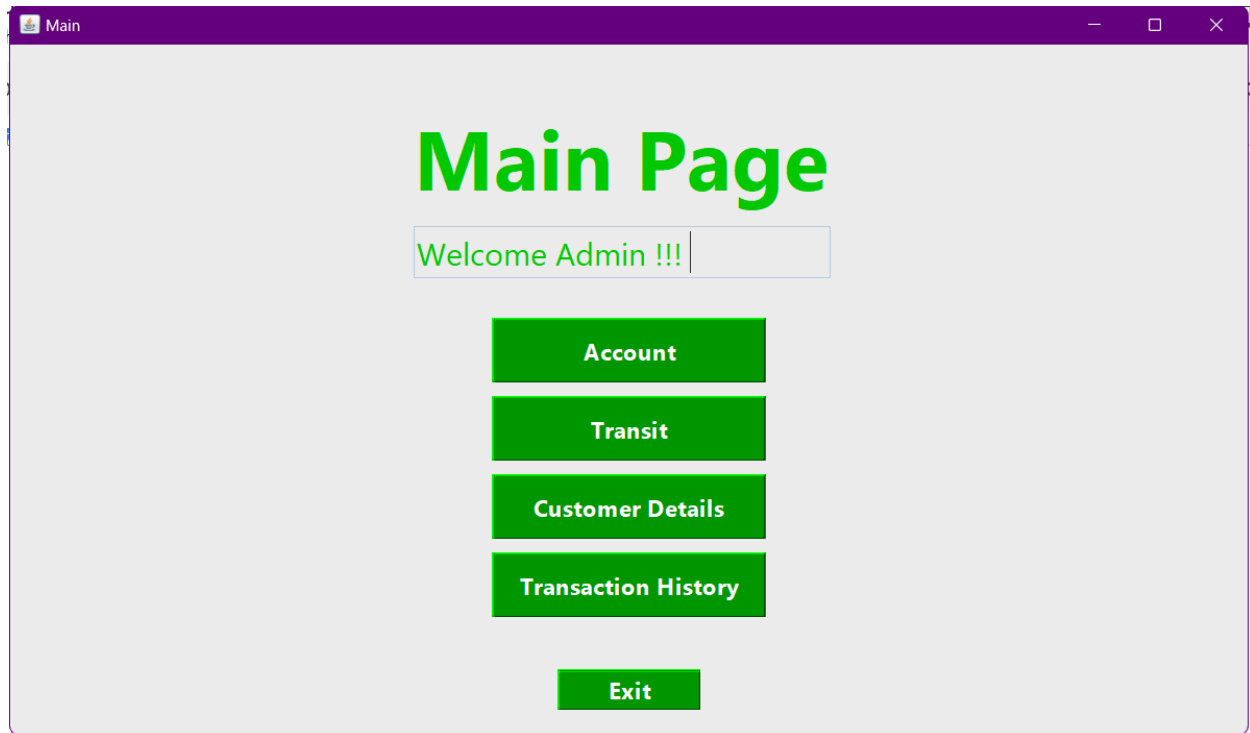


Figure 3. App interface after logging in

Account Information

ACCOUNT

CustomerID:

Account number:

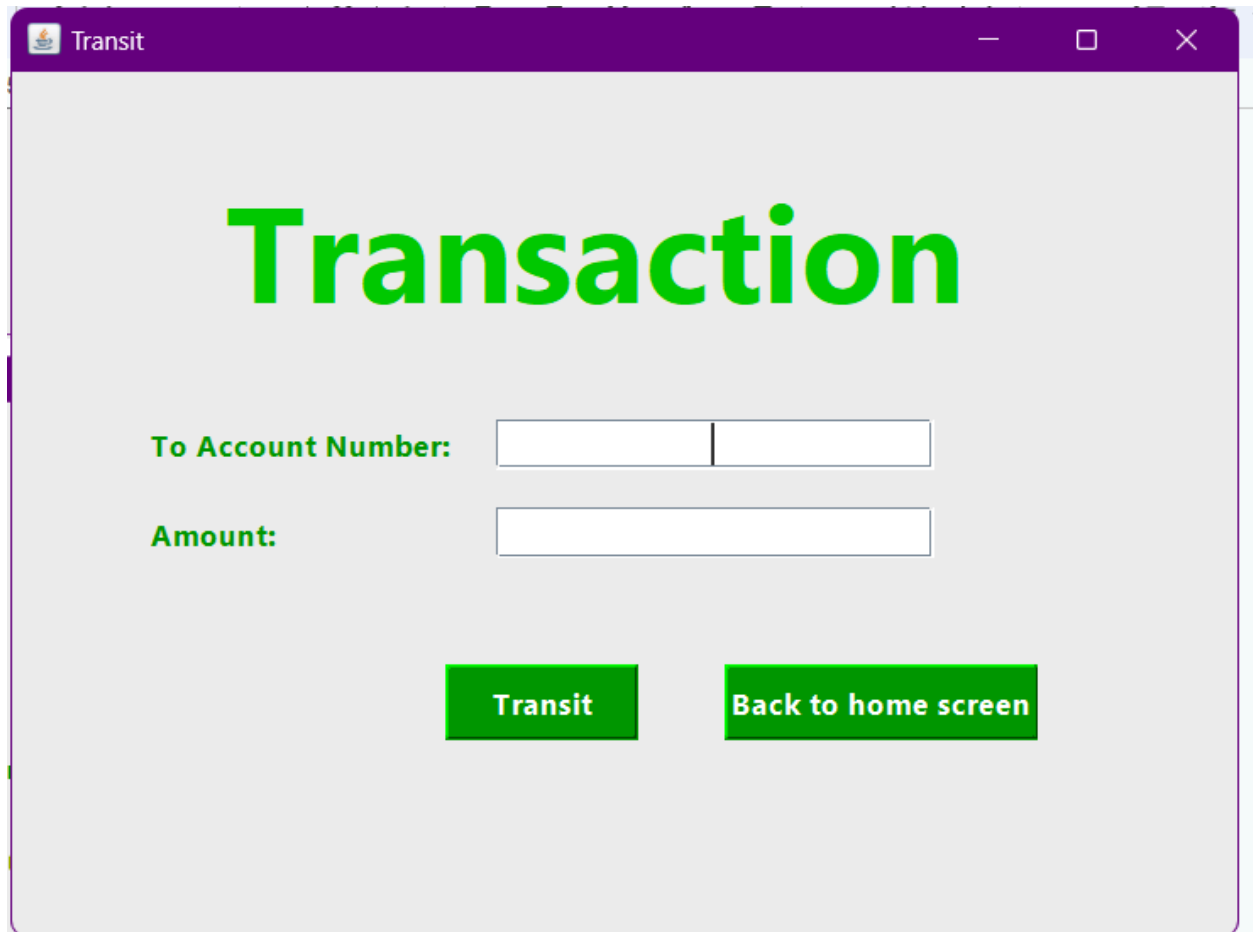
Current Balance:

Create saving accounts

Check saving accounts

Back to home screen

Figure 4. Account information screen



The image shows a web browser window titled "Transit" with a purple header bar. The main content area has a light gray background. At the top, the word "Transaction" is displayed in a large, bold, green font. Below this, there are two input fields. The first is labeled "To Account Number:" in green text and consists of two adjacent white rectangular boxes. The second is labeled "Amount:" in green text and is a single white rectangular box. At the bottom of the form, there are two green buttons with white text. The left button is labeled "Transit" and the right button is labeled "Back to home screen".

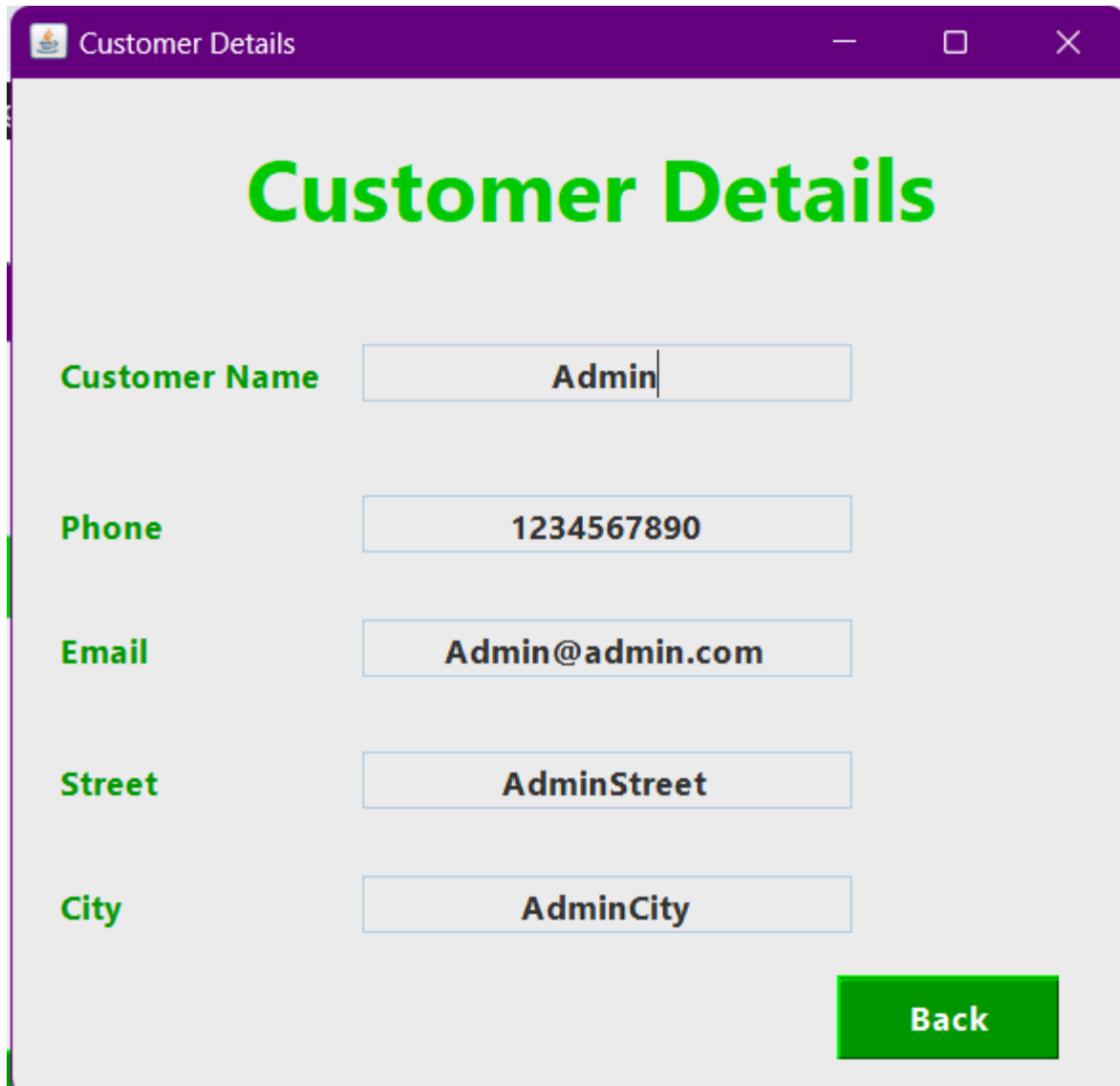
Transaction

To Account Number:

Amount:

[Transit](#) [Back to home screen](#)

Figure 5 Transition interface

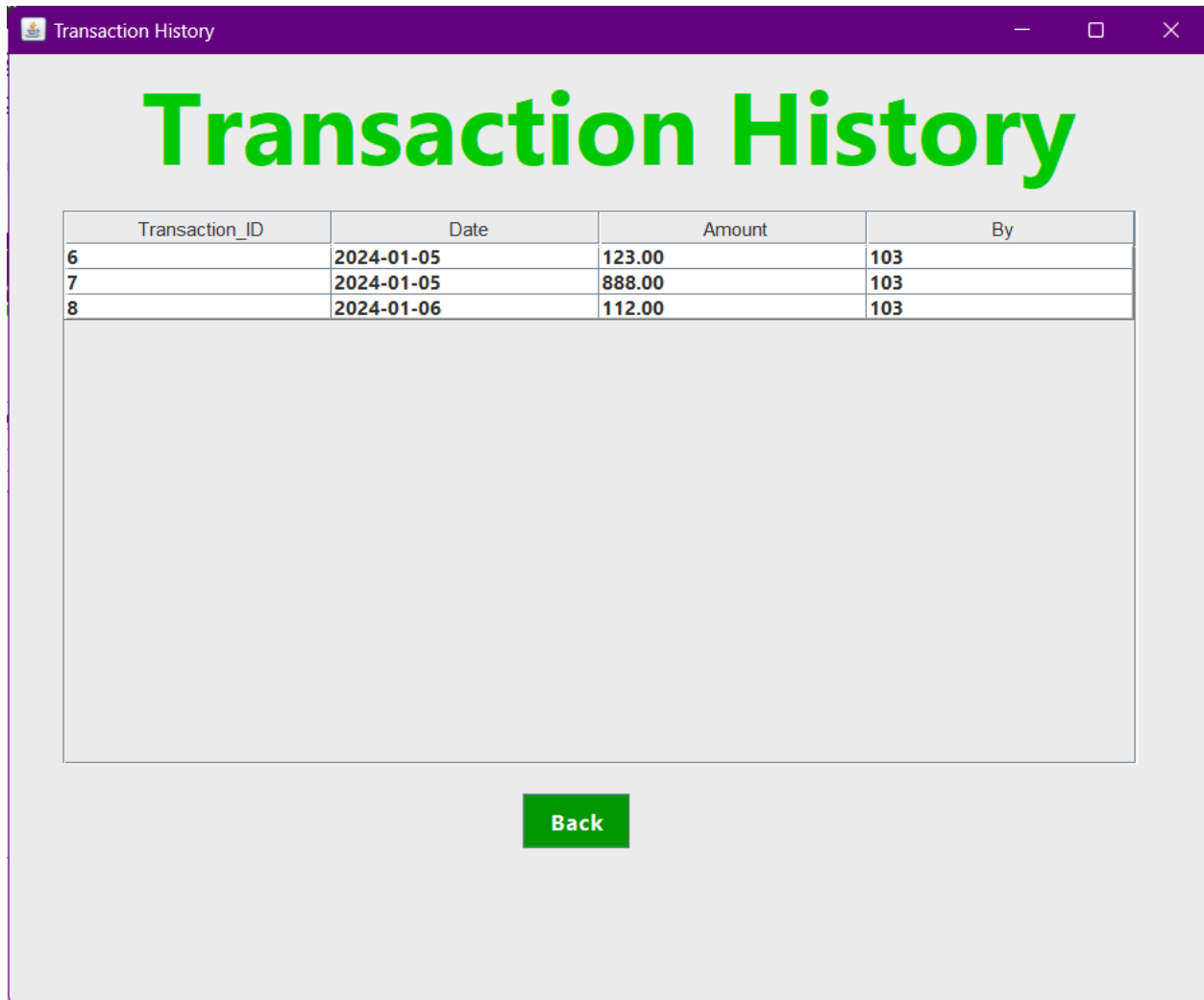


The image shows a web application window titled "Customer Details". The window has a purple header bar with standard window controls (minimize, maximize, close). The main content area has a light gray background. At the top, the title "Customer Details" is displayed in a large, bold, green font. Below the title, there are five input fields, each with a green label to its left. The labels are "Customer Name", "Phone", "Email", "Street", and "City". The input fields contain the following text: "Admin", "1234567890", "Admin@admin.com", "AdminStreet", and "AdminCity" respectively. In the bottom right corner of the form area, there is a green rectangular button with the text "Back" in white.

Field Label	Field Value
Customer Name	Admin
Phone	1234567890
Email	Admin@admin.com
Street	AdminStreet
City	AdminCity

Back

Figure 6 Customer information interface

A screenshot of a software window titled "Transaction History". The window has a purple title bar with standard minimize, maximize, and close buttons. The main content area has a light gray background. At the top, the title "Transaction History" is displayed in a large, bold, green font. Below the title is a table with four columns: "Transaction_ID", "Date", "Amount", and "By". The table contains three rows of data. Below the table is a large, empty rectangular box. At the bottom center of the window is a green button with the text "Back" in white.

Transaction_ID	Date	Amount	By
6	2024-01-05	123.00	103
7	2024-01-05	888.00	103
8	2024-01-06	112.00	103

Back

Figure 7 Transition history display

The screenshot shows a web application window titled "Checking Saving Account". The main heading is "Saving Account" in large green font. Below the heading, there are several input fields and labels:

- Saving Account No**: A text input field containing the number "4".
- Duration**: A text input field containing the number "6", followed by the word "months".
- Open Date**: A text input field containing the date "2024-01-05".
- Close Date**: A text input field containing the date "2024-07-05".
- Initial Amount**: A text input field containing the value "877.0".
- Interest rate**: A text input field containing the value "6.0".
- Expected Amount**: A text input field containing the value "903.31".

A green "Back" button is located at the bottom right of the form.

Figure 8 Saving Account Information's screen

The screenshot shows a web application window titled "Open A Saving Account". The main heading is "Open Saving Account" in large green font. Below the heading, there are two input fields and labels:

- Amount**: A text input field that is currently empty.
- Duration**: A dropdown menu showing the number "3", followed by the word "months".

At the bottom of the form, there are two green buttons: "Cancel" and "Create".

Figure 9 Create a saving account by inputting the amount of money and choosing duration

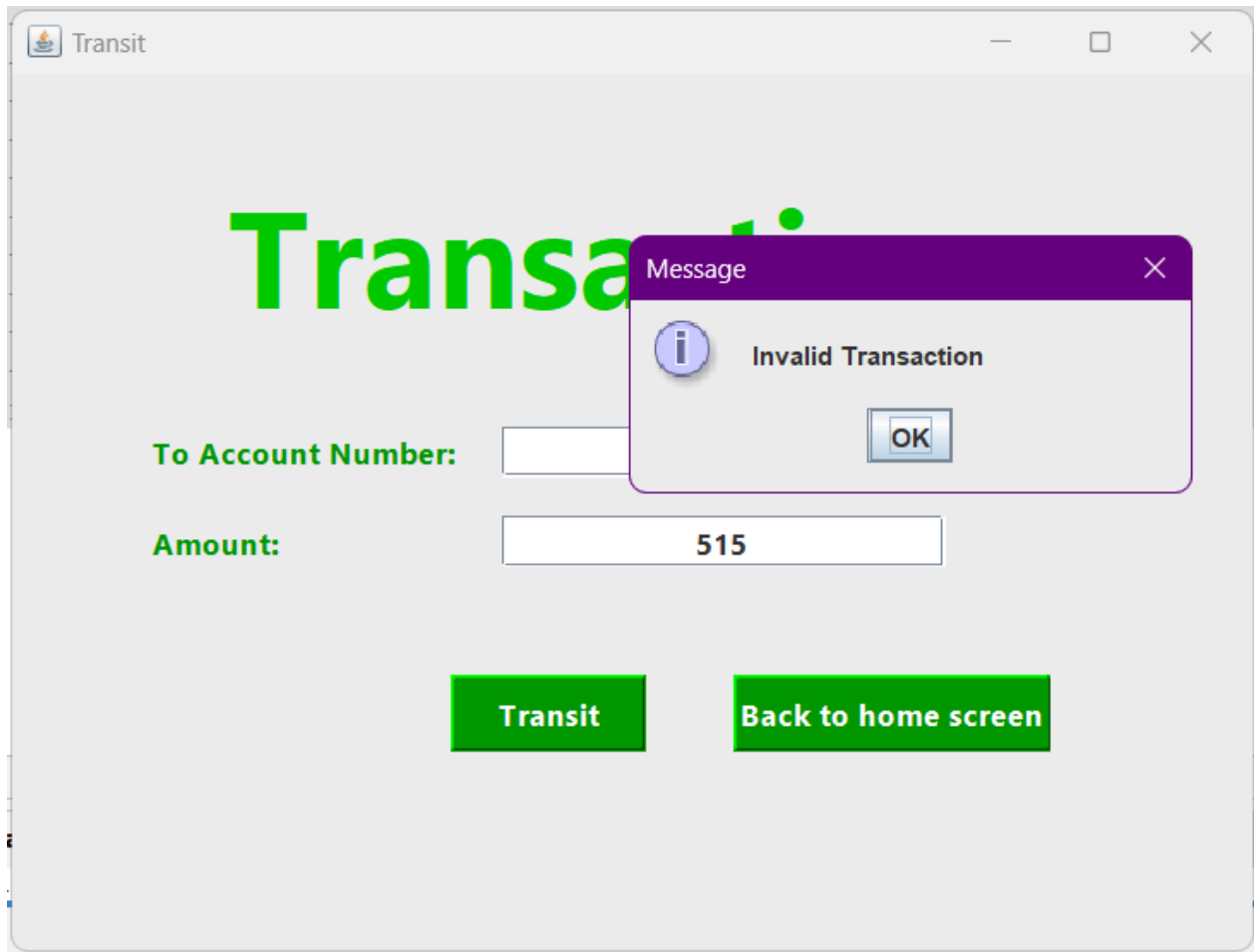


Figure 10 Invalid transaction occurs when input invalid amount

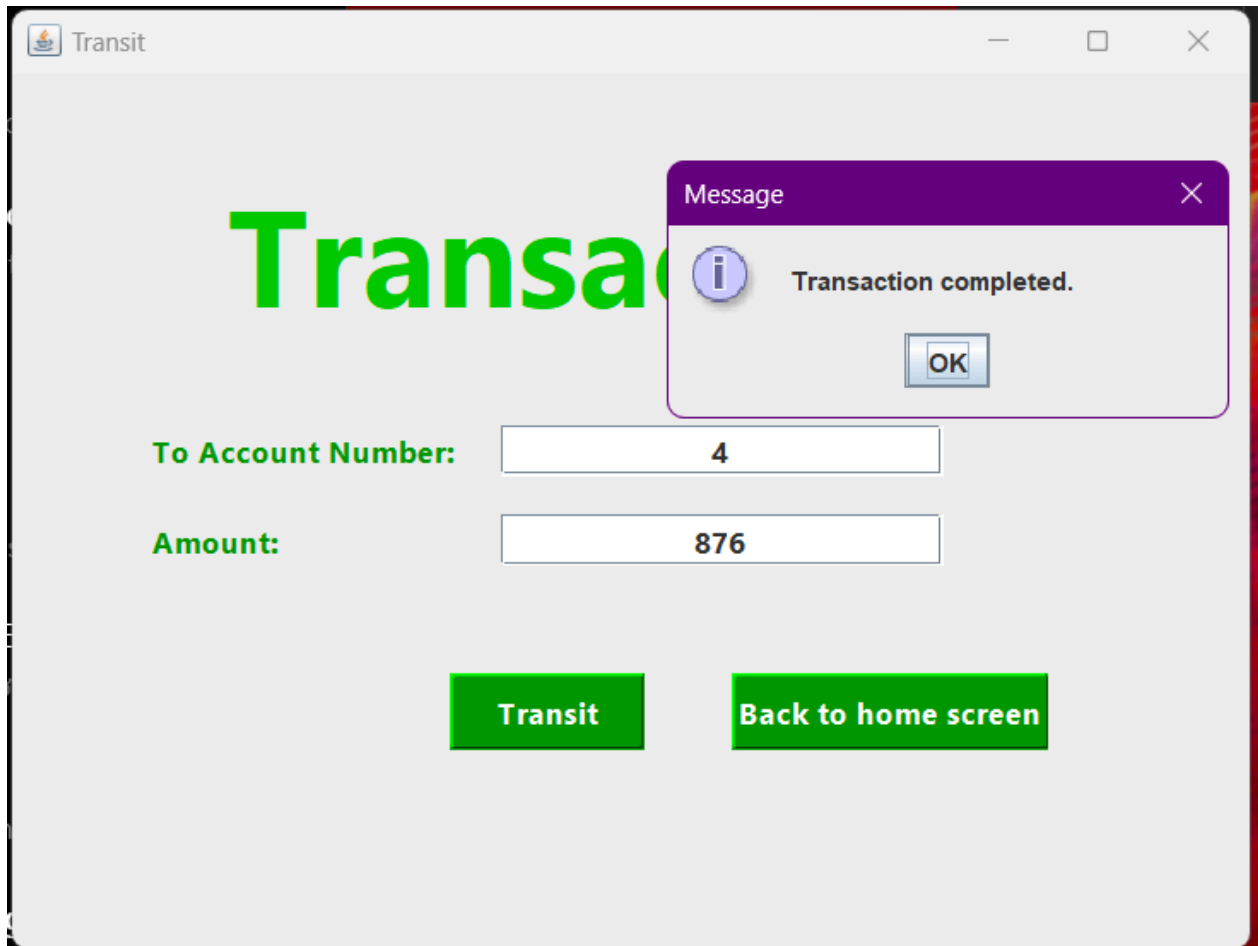


Figure 11. Transaction successful notification

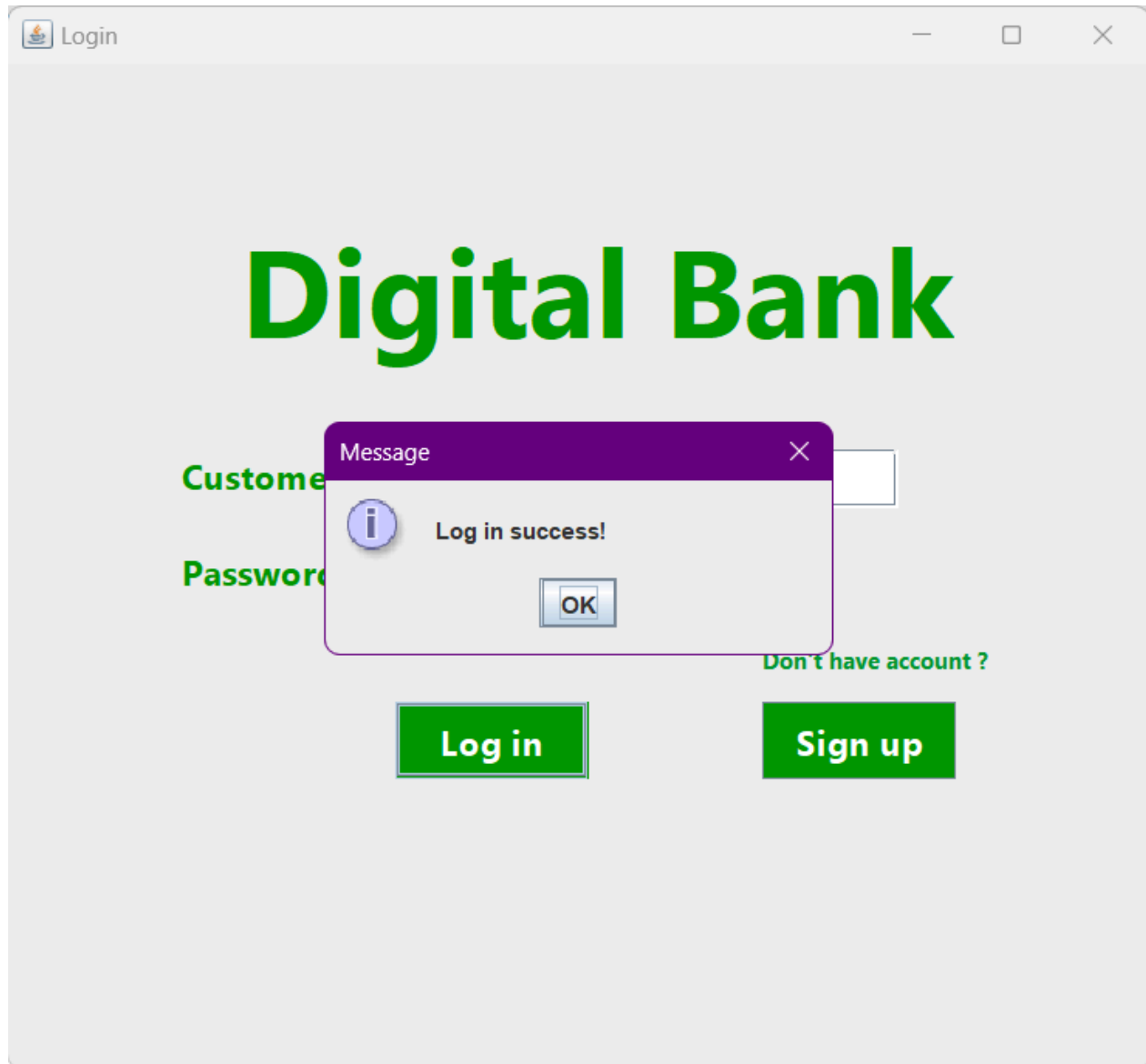


Figure 12.Login success announcement

1.7 DISCUSSION

This project serves as a valuable case study in understanding the interaction between various systems within a basic online banking system database. Our methodology, while not without its limitations, has yielded significant insights into the operational mechanics and security requisites of banking databases. A notable limitation identified is the constrained utility of JavaSwings libraries, especially when contrasted with more advanced UI design libraries like Next UI in React.

This comparison highlights a critical area for enhancement in contemporary database interface design.

1.8 CONCLUSION

This project, focusing on the development of an Online Banking System Database, has been a highly valuable learning experience for our team. It not only equipped us with practical skills in SQL and Java development but also deepened our understanding of database systems. The project's challenges and successes have enhanced our proficiency in designing and implementing efficient and secure databases, crucial for the banking sector. Our work on this project has implications beyond academic learning, potentially contributing to advancements in digital banking solutions and offering a foundation for further research in database security and user interface design.

CHAPTER 2/ PROJECT TIMELINE

In this section of this report, we present a chronological overview of the project's development phases, highlighting key milestones and deadlines. This timeline offers a clear and structured representation of the project's progression, from inception through to completion.

Start - End Date	Description of Workforce Allocation and Task Execution
Sat 14/10/2023	Assembling the required resources for the projects
Mon 16/10/2023	Identifying what to do and who doing said job
Sat 21/10/2023	First meeting offline to talk about how we approach the project
Thursday 26/10/2023	Rough design of the database on Microsoft SQL Server 2014 and sample data insertion
Sat 28/10/2023	Changes to the logic of the database and drawing
Sat 28/10/2023 - Sun 05/11/2023	Writing mid-term progress report and evaluation of the project progression.
Sun 05/11/2023 - Sat 25/11/2024	Project development on hold due to mid-term
Sat 25/11/2023 - Sun 10/12/2023	User Interaction application development [1]
Mon 11/12/2023 - Fri 15/12/2023	Software testing, bug identification/fixing and final touches.
Sat 23/12/2023	Presentation Day

Figure 12: Project development timeline

CHAPTER 3/ PROJECT ANALYSIS

In this section of this report, we delve into a detailed examination of the project's lifecycle, assessing key aspects such as design, implementation, and key features. This analysis aims to provide a thorough understanding of the project's strengths, challenges, and overall impact, offering valuable insights for future endeavors in similar domains.

3.1 DIAGRAMS SHOWING RELATIONSHIPS FOR ONLINE BANKING DATABASE [3]

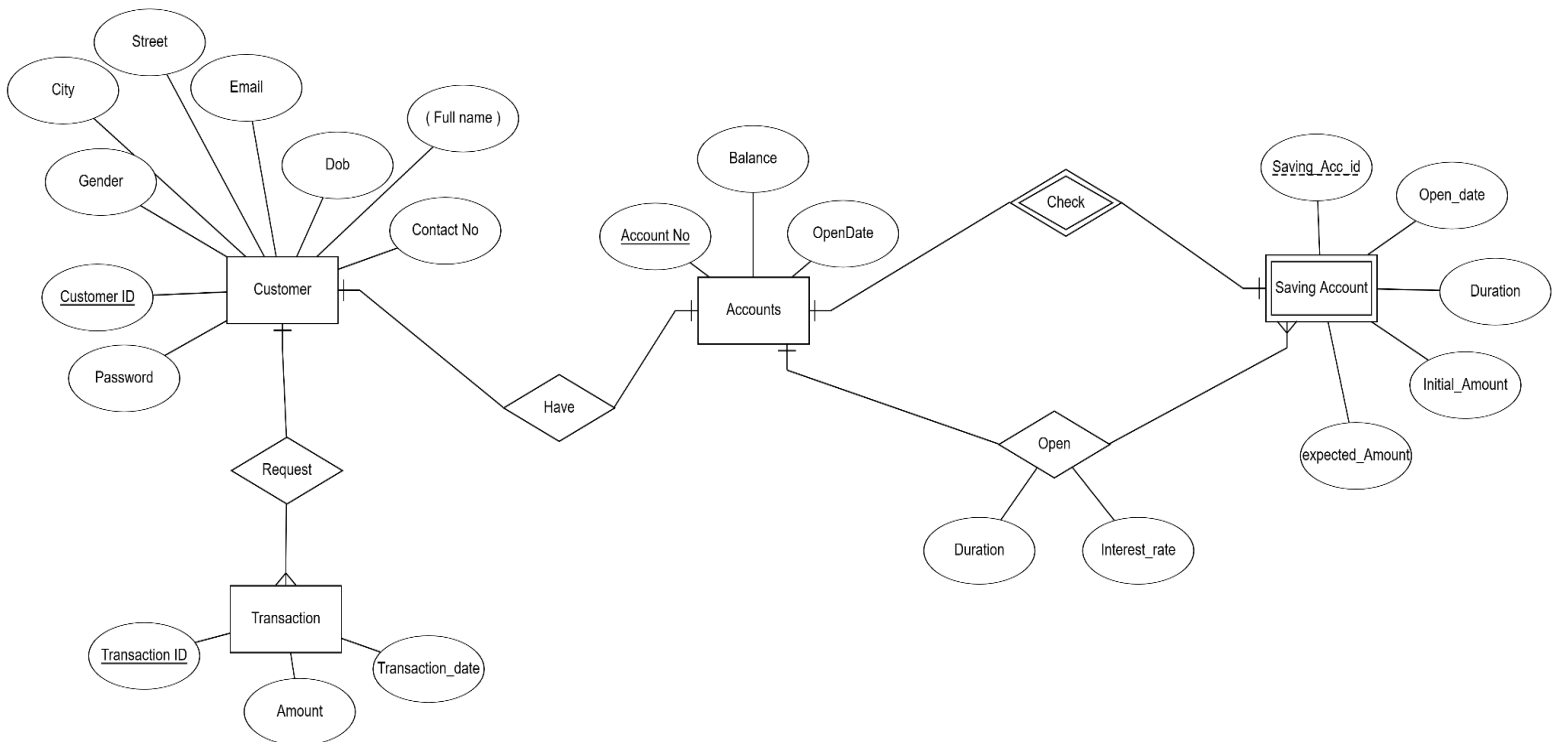


Figure 13: ERD shows the relationship between online banking

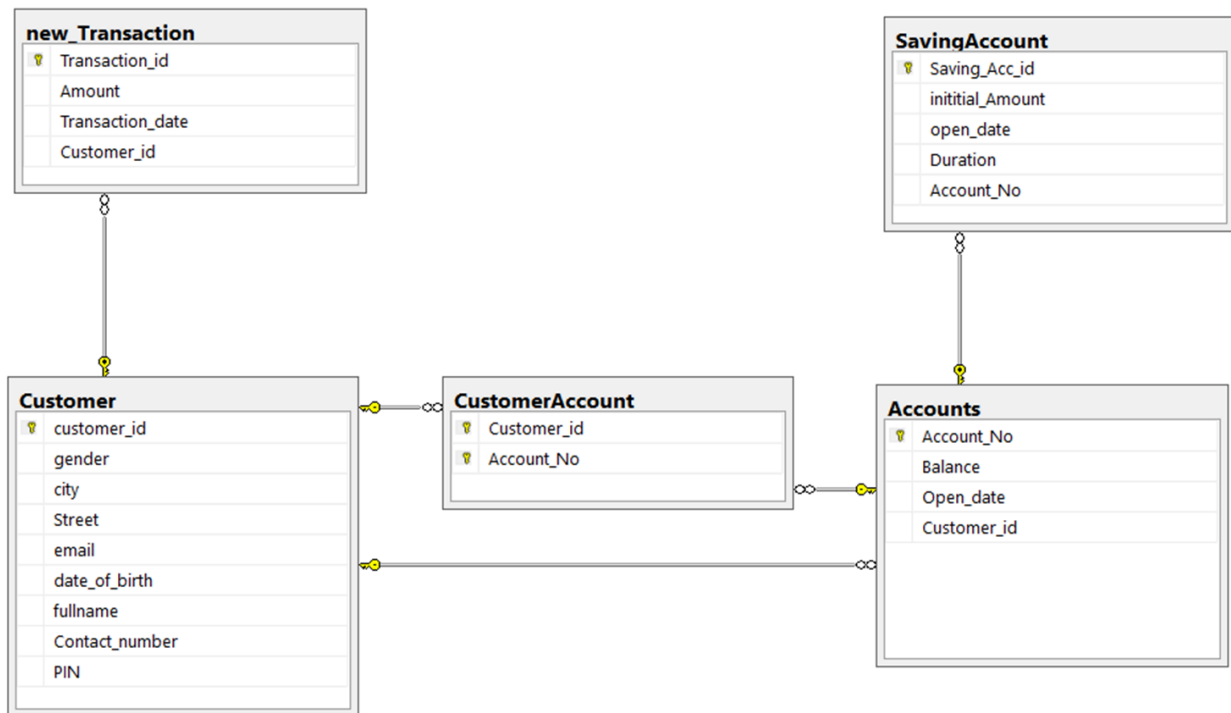


Figure 14: Schema diagram converted from the previous ERD

3.2 PRIMARY ENTITIES

Viewing account balances and transaction history. Initiating fund transfers between their accounts. Making bill payments and managing scheduled payments. Updating personal information, such as contact details.

- a. Account:
 - i. Account_No - A unique identifier for an individual account within the financial institution.
 - ii. Customer_ID - A unique identifier for a customer that may be linked to one or more accounts.
 - iii. Balance - The current amount of funds in the account.

- iv. Open_date: The date that the customer created an account.

b. Customer

- i. Customer_ID - a unique identifier for the customer, used to link to their personal information.
- ii. Gender: Gender refers to either of the two sexes (male and female).
- iii. Date_of_birth: the exact date on which a customer was born, including the day, month, and year
- iv. Fullname - The full name of the customer.
- v. City: The city where the customer lives.
- vi. Street: Street's name where the customer lives.
- vii. Contact_number - phone number used to contact the customer.
- viii. Identification Documents - Official documents used to verify the customer's identity, like a passport or driver's license.
- ix. PIN: the password for the customer to log in to their account.

c. Customer_Account:

- i. CustAcc_id: to make each customer have a unique account.
- ii. Customer_ID - a unique identifier for the customer, used to link to their personal information.
- iii. Account_No - A unique identifier for an individual account within the financial institution.

3.3 SECONDARY ENTITIES

a. new_Transaction:

- i. Transaction ID - A unique identifier for each transaction that occurs on an account.

- ii. Customer_ID - a unique identifier for the customer, used to link to their personal information.
- iii. Amount - The monetary value of the transaction.
- iv. Transaction Date - The date on which the transaction occurred.

b. Saving Account:

- i. Saving_Acc_id - A unique identifier for each saving account.
- ii. Initial_Amount - A principal amount before interest
- iii. Interest_rate -The interest rate percentage after time that the customer can achieve
- iv. Open_date - The date that the customer starts their savings account
- v. Duration - How long it takes, in years, for an investor to be repaid a bond's price by the bond's total cash flows.
- vi. Account_No: A unique identifier for an individual account within the financial institution.

3.4 DATABASE AND TABLE CREATION

A. Database Creation

```
CREATE DATABASE Online_Banking
USE Online_Banking
```

B. Schema Creation

Customer Table:

```
CREATE TABLE Customer (
  customer_id INT IDENTITY(100,1) PRIMARY KEY,
  gender VARCHAR(10),
  city VARCHAR(50),
  Street VARCHAR(50),
  email VARCHAR(100),
  date_of_birth DATE,
  fullname VARCHAR(250),
  Contact_number VARCHAR(20),
  PIN varchar(6)
);
```

new_Transition Table:

```
CREATE TABLE new_Transaction (  
  Transaction_id INT IDENTITY(1,1) PRIMARY KEY,  
  Amount DECIMAL(10, 2),  
  Transaction_date DATE,  
  Customer_id INT,  
  FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id)  
);
```

Account Table:

```
CREATE TABLE Accounts (  
  Account_No INT IDENTITY(1,1) PRIMARY KEY,  
  Balance DECIMAL(10, 2),  
  Open_date DATE,  
  Customer_id INT,  
  FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id)  
);
```

SavingAccount Table:

```
CREATE TABLE SavingAccount (  
  Saving_Acc_id INT IDENTITY (1,1) PRIMARY KEY,  
  initial_Amount money,  
  open_date date,  
  Duration int,  
  Account_No int,  
);
```

CustomerAccount Table:

```
CREATE TABLE CustomerAccount (  
  CustAcc_id INT IDENTITY(1,1) PRIMARY KEY, -- Auto-incrementing primary key for the junction table  
  Customer_id INT NOT NULL,  
  Account_No INT NOT NULL,  
  FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id),  
  FOREIGN KEY (Account_No) REFERENCES Accounts(Account_No),  
  CONSTRAINT UC_CustomerAccount UNIQUE (Customer_id, Account_No) -- To prevent duplicate pairings of customers and accounts  
);
```

3.5 NORMALIZATION

This section outlines the normalization process undertaken to ensure that the Entity-Relationship Diagram (ERD) of our banking system database conforms to the Fourth Normal Form (4NF). This initiative focused on reducing data redundancy, eliminating undesirable dependencies, and enhancing data integrity.

Background:

The banking system database's initial design captured the fundamental relationships between entities such as Customers, Accounts, SavingAccounts, and Transactions. To ensure the efficiency and reliability of the database, it was essential to assess and refine the schema by applying normalization principles up to 4NF.

Process:

The normalization process involved several key steps:

1. **The Analysis of existing structure:** The initial ERD was scrutinized for data anomalies and multi-valued dependencies that could potentially breach 4NF standards.

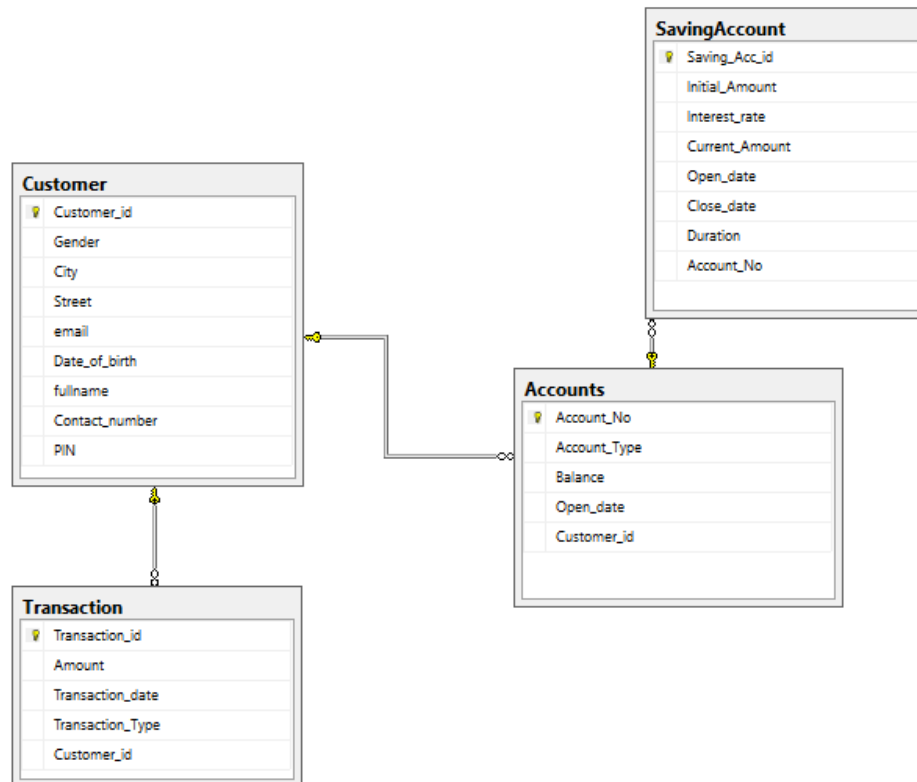


Figure 15: Primitive version of the database.

2. **Identification of Multivalued Dependencies:** Tables were inspected for attributes that contained more than one value per record, indicating a multivalued dependency.
3. **Decomposition of Tables:** To resolve multivalued dependencies, tables were decomposed appropriately. This involved creating new tables and establishing relationships that reflect the logical business rules without redundancy.
4. **Creation of Associative Entities:** To manage many-to-many relationships, associative entities (junction tables) were introduced, providing a normalized approach to linking tables that have multiple associations.

Implementation:

The normalization implementation resulted in the following changes to the ERD:

1. **Introduction of CustomerAccount Table (Fig3.2):** A new junction table, 'CustomerAccount', was created to manage the many-to-many relationship between Customers and Accounts. This table ensures that each pair of customer and account is unique and properly referenced.
2. **Adjustments to SavingAccount Table:** The 'SavingAccount table' was retained with enhancements to ensure that it references the 'Accounts' table correctly and does not hold multivalued dependencies.
3. **Validation of Transaction Table:** The Transaction table was evaluated and confirmed to adhere to 4NF, assuming that each transaction is uniquely linked to one account and one customer.

Outcome:

The normalization process successfully upgraded the banking system's ERD to 4NF. The new design mitigates previous data redundancies and establishes a clear, unambiguous structure. The CustomerAccount junction table specifically resolves the complexities of multiple customers holding joint accounts or individuals possessing multiple accounts, which was a critical aspect of the normalization process.

CHAPTER 4/ FUTURE PLANS

In this section of this report, we will outline the strategic roadmap and forward-looking initiatives for the continuous evolution and enhancement of our online banking system. This will encompass a comprehensive overview of upcoming features, technological advancements, and customer-centric improvements aimed at setting new benchmarks in digital banking excellence.

1. Security

- Enhanced the platform's security framework by integrating robust authentication measures, including the addition of security questions and the implementation of biometric verification features such as FaceID and TouchID.
- Implement new functionalities to streamline the password reset process, providing customers with a user-friendly and secure method to regain access to their accounts in the event of forgotten passwords. These features include step-by-step guided procedures, verification through registered email or phone, and optional security questions for added protection.
- Instituted a system to link customer accounts with their identity cards, significantly enhancing security measures. This linkage ensures a robust verification process, aligning each account with the customer's official identity, thereby reducing the risk of unauthorized access and strengthening overall account integrity.
- Implement password hashing protocols to fortify security against data breaches and malicious attacks. This advanced cryptographic technique transforms passwords into unique hash values, ensuring

that even in the event of a data compromise, the actual passwords remain undecipherable and secure.

2. Interface

Developed a user-centric interface, prioritizing ease of use and intuitive navigation to enhance the overall customer experience. This interface is designed with a clear layout, accessible features, and responsive design elements, ensuring a seamless and engaging interaction for all users.

Revamp the application's background and aesthetics, infusing a contemporary and visually appealing design. This update includes a refreshing color scheme, engaging graphics, and a harmonious layout, all aimed at enriching the user interface and enhancing the app's overall visual appeal.

REFERENCES

[1] [Banking Management System \(youtube.com\)](#)

[2] [Bank Management System | Java Project | Introduction \(youtube.com\)](#)

[3] [ERDPlus](#)