Project Report On

BANKING SYSTEM

By

Krishna Pensalwar (SIB 32)

Suyog Pawar (SIB 31)

Om Nichal (SIB 20)

Sunandan Phalke (SIB 33)

*Under the mentorship*

*Of*

Prof. Prachi Nilekar



Department of Information Technology

International Institute of Information Technology

Academic Year 2022 – 2023

Savitribai Phule Pune University, Pune



Department of Information Technology

International Institute of Information Technology

CERTIFICATE

This is to certify that a mini project work is carried out on

“BANKING SYSTEM”

Submitted by :

Krishna Pensalwar (SIB 32)

Suyog Pawar (SIB 31)

Om Nichal (SIB 20)

Sunandan Phalke (SIB 33)

Is a mini project work carried out by Krishna Pensalwar,

Suyog Pawar, Om Nichal , Sunandan Phalke under the mentorship of Prof. Prachi Nilekar and it is submitted towards partial fulfilment of requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Information Technology).

Prof. Dr. Jyoti Surve

Prachi Nilekar Head of Department

Information Technology

Place : Pune :: Date : 19-05-2023

Acknowledgement

With immense pleasure, I am presenting this report on mini project on “Banking System” as a part of the curriculum of S.E of Information Technology at INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, HINJEWADI, PUNE. It gives me proud privilege to complete this report work under the valuable mentorship of Prof. Prachi Nilekar. I am also extremely grateful to Dr. Jyoti Surve (HoD IT) and Dr. Vaishali Patil, Principal I2IT for providing all facilities and help for smooth progress of report work.

I would also like to thank all the staff members of I2IT (Information Technology) Department, Management, friends and family members who have directly or indirectly guided and helped me for the preparation of this report of this report and gave me an unending support right from the stage idea was convinced.

Signature

Krishna Pensalwar

Suyog Pawar

Om Nichal

Sunandan Phalke

INDEX

Serial No. Topic Page No.

1. Detailed problem statement

BANKING SYSTEM 5

1. Detailed information about

Functionalities in BASIC

BANKING SYSTEM 7

1. Hardware and Software

Requirements 9

1. ER diagrams 11
2. ER to Table conversion 12
3. Functional dependencies 17
4. Code Output 19
5. Conclusion 22

CHAPTER 1

**Detailed Problem Statement**

The goal of this project is to develop a comprehensive banking system application in Java that automates various banking operations and provides a user-friendly interface for customers.

The application should address the following requirements:

1. User Registration and Authentication:

* Customers should be able to register with the bank by providing their personal details, such as name, address, contact information, and identification documents.
* The system should authenticate customers during login using secure credentials, such as a username and password.

1. Account Management:

* Customers should be able to open various types of accounts, such as savings, checking, and fixed deposit accounts.
* Each account should have a unique account number and associated customer information.
* Customers should be able to perform operations related to their accounts, such as depositing funds, withdrawing funds, transferring funds between accounts, and viewing account balances and transaction history.

3. Transaction Processing:

* The system should facilitate secure and reliable transactions, ensuring data integrity and confidentiality.
* The system should maintain transaction records, including details such as the transaction amount, date, time, and involved accounts.

1. Error Handling and Exceptional Cases:

* The system should handle error conditions gracefully, displaying meaningful error messages and guiding users on how to resolve them.
* Exceptional cases, such as insufficient funds during a withdrawal or an invalid account number during a transfer, should be handled appropriately.

Overall, the banking system application should aim to streamline banking operations, enhance customer satisfaction, and ensure the secure management of financial transactions.

CHAPTER 2

**Detailed Information about the Functionalities of the Banking System**

1. User Account Creation and Login

The banking system application allows users to create an account and subsequently log in to access their personal banking services. This functionality ensures that customers can securely interact with the application and perform various banking operations.

2. Checking Balance and Accessing Account Information

This functionality allows users to check their account balance and access detailed information about their accounts. By providing this feature, the banking system application empowers users to stay updated on their financial status and monitor their account activities.

3. Depositing and Withdrawing Money

This functionality allows users to deposit funds into their accounts and withdraw money when needed. By providing this feature, the banking system application facilitates the movement of funds, empowering users to manage their finances conveniently.

4. Transaction Lookup

This functionality allows users to look up and review their previous transactions within the banking system application. By providing this feature, users can easily track and analyze their financial activities, helping them monitor their spending, reconcile accounts, and maintain accurate records

5. Account-to-Account Transfer

The account-to-account transfer functionality enables users to transfer funds between their own accounts or to other users' accounts within the banking system application. This feature provides a convenient and secure way for users to manage their finances, make payments, and facilitate transactions.

6. User Log Out

The user log out functionality allows users to securely sign out of their accounts within the banking system application. This feature ensures the privacy and security of user data by terminating the active session and preventing unauthorized access to the user's account.

CHAPTER 3

**Hardware and software requirements**

For a banking project in Core Java with MySQL, you will need the following hardware and software requirements:

Hardware Requirements:

1. Computer: A reliable computer capable of running the required software smoothly.

2. Processor: A modern processor with multiple cores for efficient performance.

3. Memory (RAM): At least 4GB of RAM to handle the development environment and database operations.

4. Storage: Sufficient hard disk space to install the required software and store project files.

Software Requirements:

1. Java Development Kit (JDK): Install the latest version of JDK, which includes the Java compiler and runtime environment.

2. Integrated Development Environment (IDE): Choose an IDE for Java development, such as Eclipse, IntelliJ IDEA, or NetBeans. Install and set up the IDE.

3. MySQL Database: Download and install the MySQL database server on your development machine. You can choose the community edition from the official MySQL website (https://dev.mysql.com/downloads/).

4. MySQL Connector/J: This is the official MySQL JDBC driver that allows Java applications to connect to the MySQL database. Download the latest version from the MySQL website.

Additional Libraries and Frameworks:

1. Apache Maven or Gradle (optional): These build automation tools can help manage project dependencies and simplify the build process.

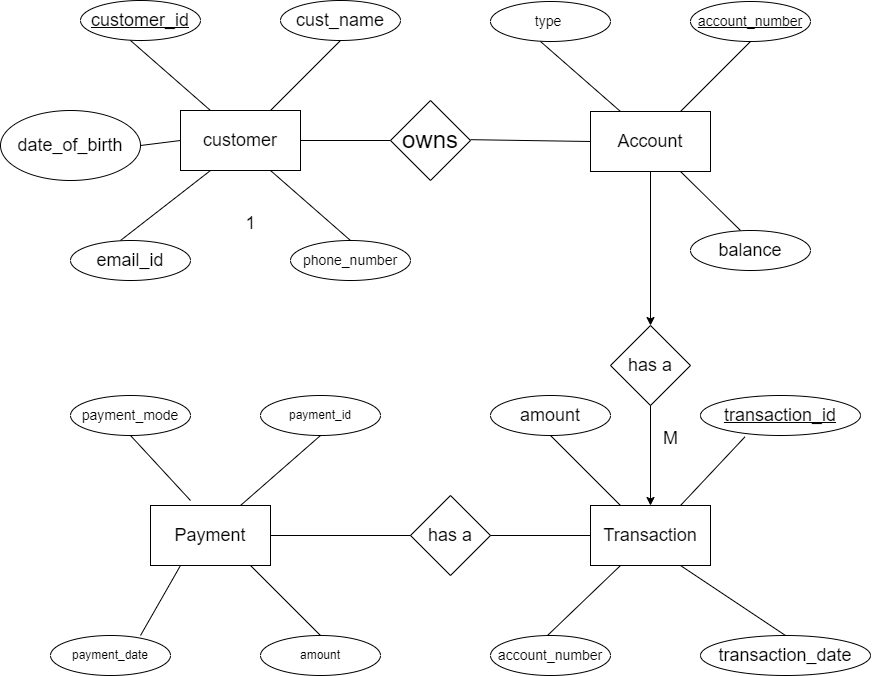
It's important to note that this setup assumes a simple banking project without advanced features. However, as we develop the project, we may need to incorporate additional libraries or frameworks for specific functionalities.

Followed secure coding practices and implement appropriate security measures, such as input validation, encryption, and

secure communication protocols, to ensure the safety and integrity of banking transactions and user data.

CHAPTER 4

**ER diagram**



CHAPTER 5

**ER to Table conversion**

**Creation of tables**

mysql> create database miniproject;

Query OK, 1 row affected (0.06 sec)

mysql> use miniproject

Database changed

mysql> create table customer ( customer\_id int Primary key ,

-> cust\_name varchar(100),

-> phone\_number varchar(15),

-> email\_id varchar(50),

-> date\_of\_birth Date );

Query OK, 0 rows affected (0.07 sec)

mysql> desc customer;

+---------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+--------------+------+-----+---------+-------+

| customer\_id | int | NO | PRI | NULL | |

| cust\_name | varchar(100) | YES | | NULL | |

| phone\_number | varchar(15) | YES | | NULL | |

| email\_id | varchar(50) | YES | | NULL | |

| date\_of\_birth | date | YES | | NULL | |

+---------------+--------------+------+-----+---------+-------+

5 rows in set (0.06 sec)

mysql> create Table Account ( Account\_Number Int Primary Key ,

-> Acc\_type varchar(100),

-> Balance int ,

-> customer\_Id Int ,

-> Foreign Key (customer\_id) References Customer(customer\_id)

-> );

Query OK, 0 rows affected (0.06 sec)

mysql> desc account;

+----------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+--------------+------+-----+---------+-------+

| Account\_Number | int | NO | PRI | NULL | |

| Acc\_type | varchar(100) | YES | | NULL | |

| Balance | int | YES | | NULL | |

| customer\_Id | int | YES | MUL | NULL | |

+----------------+--------------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> create Table Transaction ( Transaction\_Id Int Primary Key,

-> Amount Int,

-> Transaction\_Date DateTime,

-> Account\_Number Int,

-> Foreign Key (Account\_Number) References Account(Account\_Number)

-> );

Query OK, 0 rows affected (0.06 sec)

mysql> desc Transaction;

+------------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------------+----------+------+-----+---------+-------+

| Transaction\_Id | int | NO | PRI | NULL | |

| Amount | int | YES | | NULL | |

| Transaction\_Date | datetime | YES | | NULL | |

| Account\_Number | int | YES | MUL | NULL | |

+------------------+----------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> create table Payment ( Payment\_Id Int Primary Key ,

-> Payment\_Type varchar(100) ,

-> Payment\_Date DATETIME ,

-> amount Int,

-> Account\_Number Int,

-> Foreign Key (Account\_Number) References Account(Account\_Number)

-> );

Query OK, 0 rows affected (0.06 sec)

mysql> desc Payment;

+----------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+--------------+------+-----+---------+-------+

| Payment\_Id | int | NO | PRI | NULL | |

| Payment\_Type | varchar(100) | YES | | NULL | |

| Payment\_Date | datetime | YES | | NULL | |

| amount | int | YES | | NULL | |

| Account\_Number | int | YES | MUL | NULL | |

+----------------+--------------+------+-----+---------+-------+

5 rows in set (0.00 sec)

Detailed rules for conversion of ER into table conversion

\

**Rule 1: Strong Entity set having only simple attributes:**

For a strong entity set containing only simple attribute we require to make only one table for that entity in relational model.

• For each entity set we will create a separate table.

• The table includes attributes same as the attributes of the entity set.

• The primary key of the table will be the key attribute of the entity set.

**Rule 2: Set having composite attribute :**

• Only one table will be required for a strong entity set with any number of composite

attributes

• Only the simple attributes of composite attributes are considered while converting.

• For Ex: In reader table. the con1posite attribute 1s name having simple attributes first name, last name. So, we will create first name and last name attribute in reader table instead of attribute name.

**Rule 3: Set having multivalued attribute :**

• For every multivalued attribute, a ne\V relation is formed.

• The relation contains the primary key of the main table and the multivalued

attribute. The primary key of this table is a combination of the primary key of main table and multivalued attribute.

• For Ex: For Reader table user id is the primary key and phone number is multivalued

attribute, so the new table that we will snake will contain user id and phone number. The primary key for this table is combination of user id and phone number.

**Rule 4: Conversion of one to many :**

• For one-to-many relation we take the foreign key which is the primary key of one side and add this foreign key in N side table.

• If there are any attributes present in relationship, then make the attributes too.

• for Ex: The relationship between staff and book is 1 :N. so the foreign key goes on N side i.e., book.

**Rule 5: Conversion of many to many :**

• In case of many to n1any relationship, we make a separate table for the relation itself.

The primary keys of the both tables are taken as foreign key in the relations up table.

• The primary key of the relationship table will be the combination of foreign keys.

• For Ex: Keeps track is relation between a reader and staff. This means each staff maintains record of many readers and the record of a reader is maintained by several staffs. Its primary key is the combination of both the tables i.e. reader and staff.

CHAPTER 6

Functional dependencies

* For Customer table, the primary key is customer\_no so it helps us in uniquely identify a tuple in the table (cust\_name, phone\_number, email\_id, date\_of\_birth). Therefore Customer table is in BCNF because customer\_no being the super key which derives all the other attributes uniquely.
* For staff table, the primary key is staff\_id so, it helps us in uniquely identifying the tuple in the table (name, login\_id). Therefore, staff table is in BCNF because staff\_id being the supewr key which derives all the other attributes uniquely (name, login\_id).
* For Authsys table the primary key is login id. So it helps us in uniquely identifyling a tuple in the table (like password, login id). Therefore, login id table is in BCNF because login id being the super key which derives all the other attributes uniquely (password).
* For keeps Track table, the primary key is combination of staff id and user id. Therefore, Keeps track table is in BCNF because staff id being the super key.
* For readers table, the primary key is user id. So, it helps us in uniquely identifying a tuple in the table (like user id, email, first name, last name and address). Therefore readers table is in BCNF because user id being the super key which derives all the other attributes uniquely (like user id, email, first name, last name and address).
* For reader no table, the primary key is the combination. So, it helps us in uniquely identifying a tuple in the table (like user id and phone no). Therefore, readers no table is in BCNF because the combination of user id and phone no being the super key.
* For books table, the primary is ISBN. So, it helps us in uniquely identifying a tuple in the table (like ISBN, title, price, staff id, pub id). Therefore, books table is in BCNF because ISBN being the super key which derives all the other attributes uniquely (like ISBN, title, price, staff id, pub id).
* For publisher table, the primary key is pub id. So, it helps us in uniquely identifying a tuple in the table (like pub id, name). Therefore, publisher table is in BCNF because pub id being the super key which derives all the other attributes uniquely (like pub id, name).
* For track table, the primary key is the combination of user id and ISBN. So, it helps us in uniquely identifying a tuple in the table (like issue date, return date, due date). Therefore, track table is in BCNF because the combination of user id and ISBN being primary key.

CHAPTER 7

**Code Output**

-------------------------- Welcome to HDFC Bank ---------------------------

----------------------------------------------------------------------------

1. Login to an Existing account

2. Create a new account

0. Exit

----------------------------------------------------------------------------

Enter the choice code : 2

-------------------- Select the type of the Account ------------------------

1. Saving

2. Current

3. Salary

----------------------------------------------------------------------------

Enter the choice code : 1

Enter your Name : Sunandan

Enter your Age : 20

Enter Opening balance : 50000

Enter IFSC : HINJ007

Enter your PAN : DSWPP784A

Enter your Aadhar : 125478546932

Enter your Mobile : 8536741256

Enter your Address : Hinjewadi

----------------------------------------------------------------------------

Logging in...

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 1

----------------------------------------------------------------------------

Total Balance : 50000.0

Available Balance : 48000.0

Today's remaining Withdrawal Balance : 10000.0

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 2

----------------------------------------------------------------------------

Name : Sunandan

Age: 20

IFSC: HINJ007

Balance : 50,000

PAN : DSWPP784A

Aadhar : 125478546932

Mobile No : 8536741256

Address : Hinjewadi

Account Type : Saving

Account Number : 00002

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 3

-------------- Deposit ----------------

Enter the amount to be deposited : 40000

Deposit Successful!

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 5

----------------------------------------------------------------------------

Date Time Amount Transaction Type

----------------------------------------------------------------------------

20/05/23 3:05:10 pm 50000.0 Deposit

----------------------------------------------------------------------------

20/05/23 3:05:39 pm 40000.0 Deposit

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 4

-------------- Withdrawal ----------------

Enter the amount to be withdrawn : 5500

Withdrawal Successful!

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 4

-------------- Withdrawal ----------------

Enter the amount to be withdrawn : 5500

----------------------------------------------------------------------------

Withdrawal is beyond daily limit.

Please enter a amount less than or equal to 4500.0

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 5

----------------------------------------------------------------------------

Date Time Amount Transaction Type

----------------------------------------------------------------------------

20/05/23 3:05:10 pm 50000.0 Deposit

----------------------------------------------------------------------------

20/05/23 3:05:39 pm 40000.0 Deposit

----------------------------------------------------------------------------

20/05/23 3:05:59 pm 5500.0 Withdrawal

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 6

----------------------------------------------------------------------------

1. Login to an Existing account

2. Create a new account

0. Exit

----------------------------------------------------------------------------

Enter the choice code : 0

----------------------------------------------------------------------------

Data sent to MySQL

After Re-Running the app.

----------------------------------------------------------------------------

Data read from MySQL

----------------------------------------------------------------------------

-------------------------- Welcome to HDFC Bank ---------------------------

----------------------------------------------------------------------------

1. Login to an Existing account

2. Create a new account

0. Exit

----------------------------------------------------------------------------

Enter the choice code : 1

Enter the Bank account number : 00002

Enter the IFSC code : HINJ007

----------------------------------------------------------------------------

Logging in...

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 1

----------------------------------------------------------------------------

Total Balance : 84500.0

Available Balance : 82500.0

Today's remaining Withdrawal Balance : 4500.0

----------------------------------------------------------------------------

----------------------------------------------------------------------------

1. Check Balance

2. Account Info

3. Deposit

4. Withdraw

5. Show transactions

6. Log out

7. A/C Money Transfer

----------------------------------------------------------------------------

Enter the choice code : 6

----------------------------------------------------------------------------

1. Login to an Existing account

2. Create a new account

0. Exit

----------------------------------------------------------------------------

Enter the choice code : 0

----------------------------------------------------------------------------

Data sent to MySQL

CHAPTER 8

**Conclusion and future scope:**

• The project is aimed at making the day to day banking record registry hassle-free and more secured. By securing it with authentication only the account user and the administrator will have complete control over the data.

• By integrating this project with DBMS (MySQL) we enable all the features and security provided by the DBMS such as transaction control and concurrency management make this application scalable and future ready.