INSTITUTE FOR ADVANCED COMPUTING & SOFTWARE DEVELOPMENT (IACSD), AKURDI, PUNE, MAHARASHTRA

Documentation On

Bankezy

(Banking Management System)

PG-DAC March 2023

Submitted By:

Group No: 11

Roll No.	Name:
233015	Arpit Jain
233059	Saurabh P. Patil

Mrs. Sonali Mogal
Project Guide

Mr. Rohit Puranik
Centre Coordinator

TABLE OF CONTENTS:

List Of Figures:

ABSTRACT:

The "Bankezy.com Bank Management System " is a comprehensive project that focuses on augmenting the functionalities available to the users using Bankezy application. This report presents the step-by-step implementation of features that allow users to view user details, manage accounts, and access relevant functionalities. The project employs technologies such as Spring MVC, Java EE, JSP, and MySQL to achieve these goals. By extending the existing capabilities, the project aims to provide a more robust and user-friendly platform for bank Users, enhancing their ability to perform tasks efficiently.

The **Bankezy.com** is a comprehensive web-based application designed to streamline and automate various banking operations. This project aims to provide an efficient and user-friendly platform for customers to manage their accounts, perform transactions, and access banking services online.

The project encompasses a range of functionalities, including user registration, account management, fund transfers, transaction history tracking, and more. Through a secure and intuitive interface, users can seamlessly navigate through different features, ensuring a convenient and reliable banking experience.

This project report outlines the detailed design, implementation, and functionality of the Bankezy.com Bank Application. It presents the project's objectives, scope, methodologies, and key features. The report delves into the technical aspects of the system, discussing technologies used, database design, user interfaces, and security measures

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, **Mrs.Sonali Mogal** for providing us with the right guidance and advice at the crucial juncture sand for showing me the right way. I extend my sincere thanks to our respected **Centre Co-Ordinator Mr.Rohit Puranik**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Saurabh P Patil. (233059)

Arpit Jain (233015)

Introduction

Bankezy - A Modern Banking Application

In today's digital era, the banking industry has witnessed a significant transformation with the advent of modern technologies. The project "Bankezy" is a comprehensive banking application designed to streamline and enhance various banking operations, offering users an efficient and user-friendly experience.

Bankezy aims to bridge the gap between traditional banking practices and the demands of modern customers. The application integrates features that cater to both customers and bank employees, ensuring smooth financial transactions, convenient account management, and robust security measures.

Bankezy offers a range of features for customers, including account creation, funds transfer, transaction history, and payment processing. Additionally, users can access their account details, check balances, and manage their profiles seamlessly. The application also provides bank employees, particularly managers, with advanced privileges to oversee user data and account activities.

Built on Java EE and Spring MVC frameworks, Bankezy leverages the power of Java programming and Spring's extensive libraries to create a modular and scalable architecture. The integration of JSP for the user interface ensures a dynamic and interactive experience. The use of MySQL as the backend database management system guarantees data integrity and efficiency.

Bankezy offers several advantages over traditional banking methods. Customers can perform transactions anytime, anywhere, reducing the need for physical visits to the bank. The secure login and authentication mechanisms ensure data privacy, while the real-time updates and notifications keep users informed about their financial activities.

Project Objective

The objective of the project is to develop a modern banking application, "Bankezy," that offers seamless online banking services to users. It aims to provide an intuitive user interface for account management, transactions, and payments. Additionally, the project aims to enhance security features to ensure data privacy and secure financial operations.

Project Overview

"Bankezy" is a comprehensive banking application designed to revolutionize traditional banking practices. It offers customers the convenience of online transactions, account management, and payment processing. The application targets modern users who seek digital solutions for their financial needs. With a user-friendly interface, advanced security measures, and real-time updates, "Bankezy" aims to enhance the overall banking experience.

Project Scope

The scope of the project encompasses the development of a feature-rich banking application, "Bankezy," that caters to both customers and bank employees. Customers can create accounts, perform transactions, view transaction history, and manage their profiles. For bank employees, especially managers, the application provides capabilities to oversee user data and account activities. The project includes backend development using Java EE and Spring MVC, frontend design with JSP, and integration with MySQL for data management. The application's scope also extends to implementing robust security measures, including user authentication and data encryption, to ensure the safety and privacy of financial operations.

1.1 Modules

The project consists of several modules that collectively form the "Bankezy" banking application. Each module serves a specific purpose and contributes to the overall functionality of the application. The main modules present in the project are:

1. User Authentication and Security:

This module handles user registration, login, and security features. It ensures that user accounts are securely created, passwords are encrypted, and user access is authenticated using tokens or session management.

2. User Profile Management:

Users can view and update their profile information, including personal details, contact information, and security settings. This module also includes functionality for account verification and password reset.

3. Account Management:

This module allows users to create various types of accounts, such as savings or checking accounts. Users can view their account balances, transaction history, and manage account preferences.

4. Transaction Processing:

Users can perform financial transactions like deposits, withdrawals, transfers, and payments. The module ensures secure and accurate transaction processing while maintaining user account balances.

5. Payment Processing:

Users can make payments to beneficiaries or other accounts, both within the bank and to external entities. This module supports different payment methods and provides transaction tracking.

6. Transaction History:

The application maintains a comprehensive record of all user transactions, providing users with an overview of their financial activities. This module enables users to review past transactions and monitor their financial history.

7. Employee Portal:

This module is exclusively for bank employees, particularly managers. It allows employees to access user information, perform administrative tasks, and manage accounts. It includes features like user search, account details, and account deletion.

8. Dashboard and Analytics:

The dashboard provides users with an overview of their accounts, recent transactions, and account summaries. Analytics tools can be integrated to offer insights into spending patterns and account trends.

9. User Interface (UI):

This module deals with the frontend development, including designing user-friendly interfaces using JSP templates, CSS, and JavaScript. It focuses on creating an intuitive and visually appealing user experience.

10. Database Management:

This module involves designing and managing the database schema using MySQL. It includes creating tables for users, accounts, transactions, and payment history, as well as implementing data integrity and relationships.

These modules collectively form the core components of the "Bankezy" banking application, offering users a comprehensive and secure platform for their financial activities while providing bank employees with tools for efficient account management.

System Analysis

System analysis is a crucial phase in the software development life cycle that involves studying and evaluating an existing system or process to identify its strengths, weaknesses, and requirements. It aims to understand the system's functionality, processes, data flow, and interactions with various stakeholders. This analysis provides the foundation for designing and implementing an improved solution that meets user needs more effectively.

In the context of the project, the existing system refers to the current state of the bank application, including its functionalities, user interactions, and technical architecture. It likely involves manual processes for user registration, account management, and transactions. The existing system might lack automation, have limited user authentication and security measures, and could be prone to errors due to manual intervention.

The proposed system, on the other hand, outlines the improvements and enhancements that will be made to the existing system. It leverages modern technologies, software architecture, and best practices to create a more efficient and user-friendly banking application. The proposed system aims to automate user registration, provide a secure and seamless authentication process, enable comprehensive account management, facilitate various types of transactions, and offer detailed transaction history.

Key enhancements in the proposed system include:

1.User Authentication and Security Enhancement: Implement robust authentication mechanisms such as token-based authentication and multi-factor authentication to ensure secure user access.

2. Account Management: Introduce an intuitive interface for users to create and manage different types of accounts. Provide options to view account balances, transaction histories, and other relevant details.

3. Transaction Processing: Streamline the transaction process by automating deposits, withdrawals, transfers, and payments. Implement validation checks to ensure accurate and error-free transactions.

4. Employee Portal: Develop a dedicated portal for bank employees with restricted access to view and manage user details, enhancing administrative efficiency.

5. Dashboard and Analytics: Design a user-friendly dashboard to display key account information, recent transactions, and analytics. Provide insights into spending patterns and financial trends.

6. Database Management: Optimize the database schema for efficient data storage and retrieval. Ensure data integrity and implement relationships between various tables.

7. User Interface (UI) Enhancement: Create an intuitive and responsive user interface using JSP templates, CSS, and JavaScript, offering a seamless user experience across devices.

The proposed system aims to transform the bank application into a modern and efficient platform that provides users with secure and convenient financial services. Through a systematic analysis of the existing system and the subsequent implementation of the proposed system, the project seeks to enhance user satisfaction, streamline operations, and ensure data accuracy and security.

System Requirement Specification(SRS)

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed and comprehensive overview of the requirements, features, and specifications for the development of the Bankezy, Banking Management System. This document serves as a crucial reference for stakeholders, developers, testers, and project managers involved in the project's lifecycle.

1.2 Scope

The Bankezy, Banking Management System aims to revolutionize the banking experience by offering users a cutting-edge, secure, and user-friendly platform for managing their accounts and conducting transactions. Furthermore, the system empowers authorized bank employees to seamlessly manage user accounts, transactions, and operational aspects. The scope of this project encompasses the entire software development process, from design and development to testing and deployment.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **API:** Application Programming Interface
- **UI:** User Interface
- **DBMS:** Database Management System

2. Overall Description

2.1 Product Perspective

Bankezy represents an independent and self-contained web application that interfaces with a robust MySQL backend database. It is designed to provide customers with a unified platform to efficiently manage their accounts, perform secure transactions, and communicate effectively with bank employees. In parallel, the system equips bank employees with tailored functionalities to ensure optimal oversight and management of user activities.

2.2 User Classes and Characteristics

1. **Customers (Users):** Customers represent individuals who possess bank accounts and require a seamless, secure, and modern platform to manage their accounts, initiate transactions, and stay updated on their financial activities.

2.3 Operating Environment

Bankezy is designed to be accessible across a variety of modern web browsers, ensuring cross-device compatibility that caters to both desktop and mobile users. The application will be hosted on a secure web server to guarantee uninterrupted availability and robust performance.

2.4 Design and Implementation Constraints

- **Technology Stack:** The software architecture will leverage the power of Java for backend development, utilizing its versatility and scalability. MySQL will serve as the core database management system, ensuring efficient data storage and retrieval. The frontend will be developed using HTML, CSS, Bootstrap, and FontAwesome to create a visually appealing and intuitive user experience.
- **Security Measures:** Bankezy will implement a multi-layered security framework encompassing data encryption, secure authentication mechanisms, and role-based access control (RBAC) to safeguard sensitive information and user privacy.
- **Scalability:** The system will be designed with scalability in mind, ensuring that it can accommodate future growth in user base and transaction volume without compromising performance.

2.5 User Documentation

User satisfaction and ease of use are paramount. To this end, comprehensive user documentation will be developed. This documentation, including an exhaustive user manual, will provide clear instructions, illustrated examples, and step-by-step guides to navigate the application's features seamlessly.

3. System Features

3.1 User Registration and Authentication

The system streamlines the user registration process, allowing customers to provide their personal details, including names and email addresses. A secure email verification mechanism is implemented to confirm the authenticity of user email addresses. Passwords are securely hashed and stored in the database to ensure data security and privacy.

3.2 Account Management

Customers have the power to view their account details, including real-time balances, account numbers, and types. The system enables users to proactively update their contact information and customize their account preferences according to their needs.

3.3 Transactions

With a user-centric focus, Bankezy empowers customers to initiate fund transfers between their accounts seamlessly. The system's transaction history feature provides a comprehensive overview of transaction types, amounts, sources, statuses, and reasons, enhancing transparency and accountability.

3.4 Payments

The payment functionality within Bankezy offers customers the convenience of securely transferring funds to beneficiaries. The system meticulously records beneficiary details, transaction amounts, payment statuses, and reference numbers, providing users with a robust payment history for their records.

4. External Interface Requirements

4.1 User Interfaces

Bankezy's user interface is thoughtfully designed to seamlessly blend aesthetics and user-friendliness. Leveraging the capabilities of HTML, CSS, Bootstrap, and FontAwesome, the application offers a visually appealing and intuitive experience that prioritizes user engagement and satisfaction.

4.2 API Interfaces

The Bankezy application interfaces through meticulously documented API endpoints, facilitating seamless interaction with core functionalities. These APIs encompass a range of critical processes, including user registration, authentication, account management, transaction processing, and payment handling.

5. Functional Requirements

5.1 User Registration and Authentication

- **Requirement:** Users can initiate the registration process by providing their personal details, including names and email addresses.
- **Requirement:** Email verification tokens are generated and sent to users for confirming their email addresses securely.
- **Requirement:** User passwords are transformed into secure hashes using industry-standard encryption techniques and stored securely in the database.

5.2 Account Management

- **Requirement:** Customers can easily access and view detailed account information, including current balances, account numbers, and types.
- **Requirement:** Users can proactively update their contact information and customize account preferences to suit their evolving needs.

5.3 Transactions

- **Requirement:** Customers are empowered to initiate fund transfers between their accounts using a seamless and intuitive process.
- **Requirement:** The system meticulously logs transaction details, including types, amounts, sources, statuses, and reasons, creating a comprehensive transaction history.

5.4 Payments

- **Requirement:** Customers can efficiently transfer funds to designated beneficiaries using a secure and user-friendly payment functionality.
- **Requirement:** The system maintains a detailed payment history, recording beneficiary details, transaction amounts, payment statuses, and reference numbers.

5.5 Employee Access and Management

- **Requirement:** Authorized bank employees can access and manage user accounts and transactions based on their designated roles and responsibilities.
- **Requirement:** Administrative employees possess enhanced privileges, allowing them to verify accounts, review transactions, and manage user data within their authorized scope.

6. Non-Functional Requirements

6.1 Performance

- **Requirement:** The system must exhibit exceptional performance, catering to a potentially large number of concurrent users without compromising on responsiveness.
- **Requirement:** Transaction processing times should be minimal to ensure seamless and efficient user experiences during fund transfers.

6.2 Security

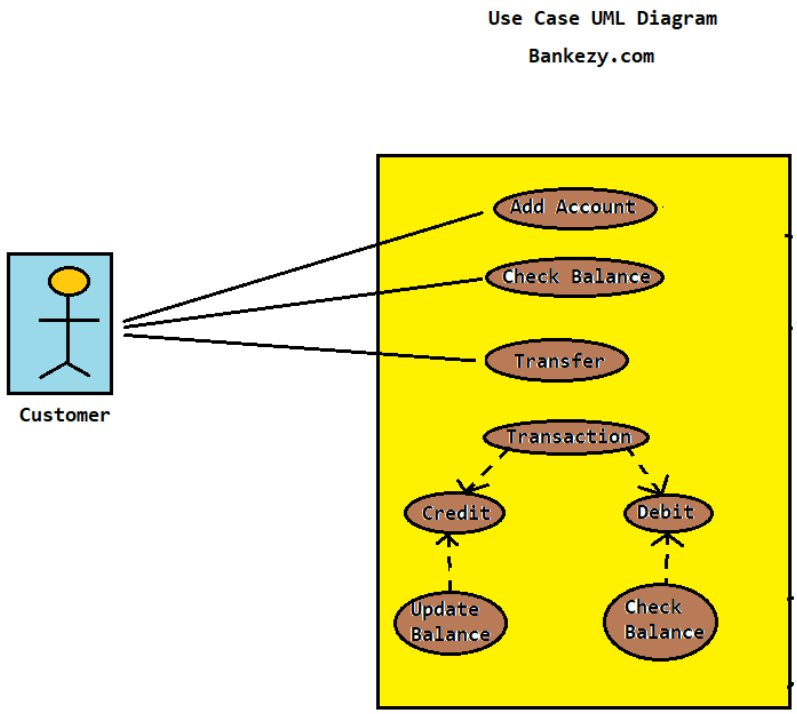
- **Requirement:** Data transmission within the system must be encrypted using robust cryptographic protocols to ensure secure communication between clients and servers.
- **Requirement:** Multi-factor authentication mechanisms and secure password policies must be implemented to validate user identities and prevent unauthorized access.
- **Requirement:** Role-based access control (RBAC) mechanisms must be enforced rigorously to restrict unauthorized access to sensitive data and functionalities.

6.3 Reliability

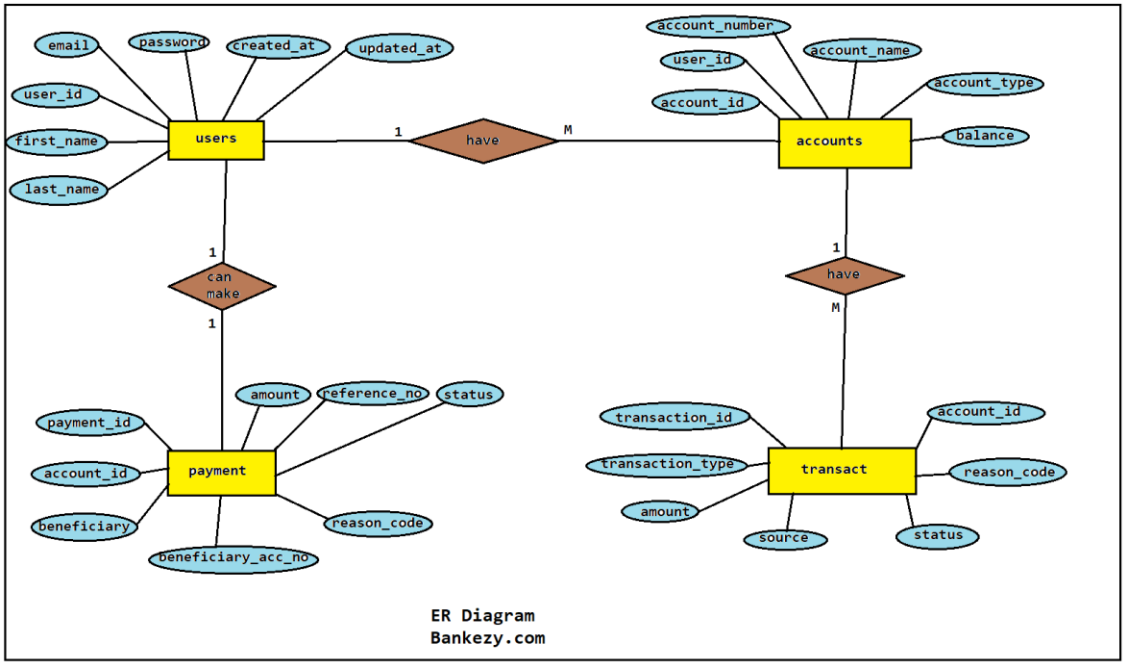
- **Requirement:** The system should be operational 24/7, guaranteeing consistent availability to users with minimal scheduled downtime for necessary maintenance tasks.

7. System Models

7.1 Use Case Diagram



7.2 E-R diagram:



8. Appendixes

8.1 Glossary

- **SRS:** Software Requirements Specification
- **API:** Application Programming Interface
- **UI:** User Interface
- **DBMS:** Database Management System

System Design

System design is a critical phase in the software development life cycle that focuses on converting the requirements specified in the software requirements specification (SRS) into a detailed blueprint for the entire system. It involves creating a structured plan for how the software will be developed, organized, and implemented. System design encompasses various aspects, including architectural design, database design, user interface design, and more. The primary goals of system design are to ensure the system's functionality, scalability, maintainability, and performance**.**

2. Input Design:

Input design is a crucial component of system design that deals with how data is entered into the system. It involves defining the methods and techniques for capturing and entering data accurately and efficiently. Input design aims to minimize errors, reduce user effort, and ensure the completeness and accuracy of the entered data. Proper input design enhances user experience and system usability.

Key Aspects of Input Design:

2.1 Data Entry Methods:

Selecting appropriate data entry methods based on the type of data and user preferences. Methods can include keyboard input, barcode scanning, voice recognition, and more.

2.2 Data Validation:

Implementing validation checks to ensure that the entered data is accurate and conforms to specified formats. Validation helps prevent errors and ensures data integrity.

2.3 Data Verification:

Incorporating verification techniques to double-check data accuracy, such as re-entering critical data or confirming data with a second person.

2.4 Data Compression:

Utilizing data compression techniques to reduce the amount of data to be entered without losing essential information.

2.5 Data Security:

Implementing security measures to protect sensitive data during input. This can include encryption and user authentication.

2.6 User-Friendly Interfaces:

Creating intuitive and user-friendly interfaces that guide users through the data entry process and provide helpful prompts and feedback.

2.7 Error Handling:

Designing error handling mechanisms to handle incorrect or incomplete data entries. Clear error messages and corrective actions should be provided to users.

3. Output Design:

Output design focuses on how information is presented to users or other systems. It involves determining the format, layout, and medium of output data to ensure its usefulness and understandability. The primary goal of output design is to provide users with relevant and actionable information.

Key Aspects of Output Design:

3.1 Information Presentation:

Selecting appropriate ways to present information, such as reports, graphs, charts, or textual summaries. The format should align with the users' needs and the nature of the data.

3.2 Layout and Formatting:

Designing clear and well-organized layouts that enhance readability. Proper formatting, headings, and section divisions help users quickly understand the content.

3.3 Data Summarization:

Summarizing large sets of data into meaningful insights that users can easily comprehend. Aggregating data into totals, averages, or percentages can help users make informed decisions.

3.4 Filtering and Sorting:

Providing options to filter and sort output data based on user preferences. This enables users to focus on specific aspects of the information.

3.5 Customization:

Allowing users to customize the output based on their requirements. Users may want to select specific fields or parameters to generate tailored reports.

3.6 Consistency:

Maintaining consistency in the presentation of output across different modules of the system. Consistent fonts, colors, and layouts contribute to a unified user experience.

3.7 Accessibility:

Ensuring that output is accessible to all users, including those with disabilities. Compliance with accessibility standards enhances inclusivity.

3.8 Distribution Medium:

Selecting the appropriate medium for distributing output, such as printed reports, digital files, or interactive web-based displays.

4. Conclusion:

System design, input design, and output design are interconnected aspects that play a crucial role in developing effective and user-friendly software systems. Input design focuses on capturing data accurately and efficiently, while output design ensures the meaningful presentation of information. Both design phases contribute to the overall success of a software application by enhancing user experience, data accuracy, and decision-making capabilities. A well-designed system, combined with thoughtful input and output design, results in a powerful and user-centric software solution**.**

Database Design

Designing a database for a bank application involves creating tables that accurately represent the various entities, their relationships, and the attributes associated with them. Here's a database design for the given project, based on the provided schema:

Database Design:

1. Table: users
   - user_id (Primary Key)
   - first_name
   - last_name
   - email
   - password
   - token
   - code
   - verified
   - verified_at
   - created_at
   - updated_at

2. Table: accounts
   - account_id (Primary Key)
   - user_id (Foreign Key references users)
   - account_number
   - account_name
   - account_type
   - balance
   - created_at
   - updated_at

3. Table: transaction_history

- transaction_id (Primary Key)
- account_id (Foreign Key references accounts)
- transaction_type
- amount
- source
- status
- reason_code
- created_at

4. Table: payments

- payment_id (Primary Key)
- account_id (Foreign Key references accounts)
- beneficiary
- beneficiary_acc_no
- amount
- reference_no
- status
- reason_code
- created_at

Database Relationships:

- One user can have many accounts (one-to-many relationship between users and accounts).
- One account can have many transaction records (one-to-many relationship between accounts and transaction_history).
- One account can have many payment records (one-to-many relationship between accounts and payments).

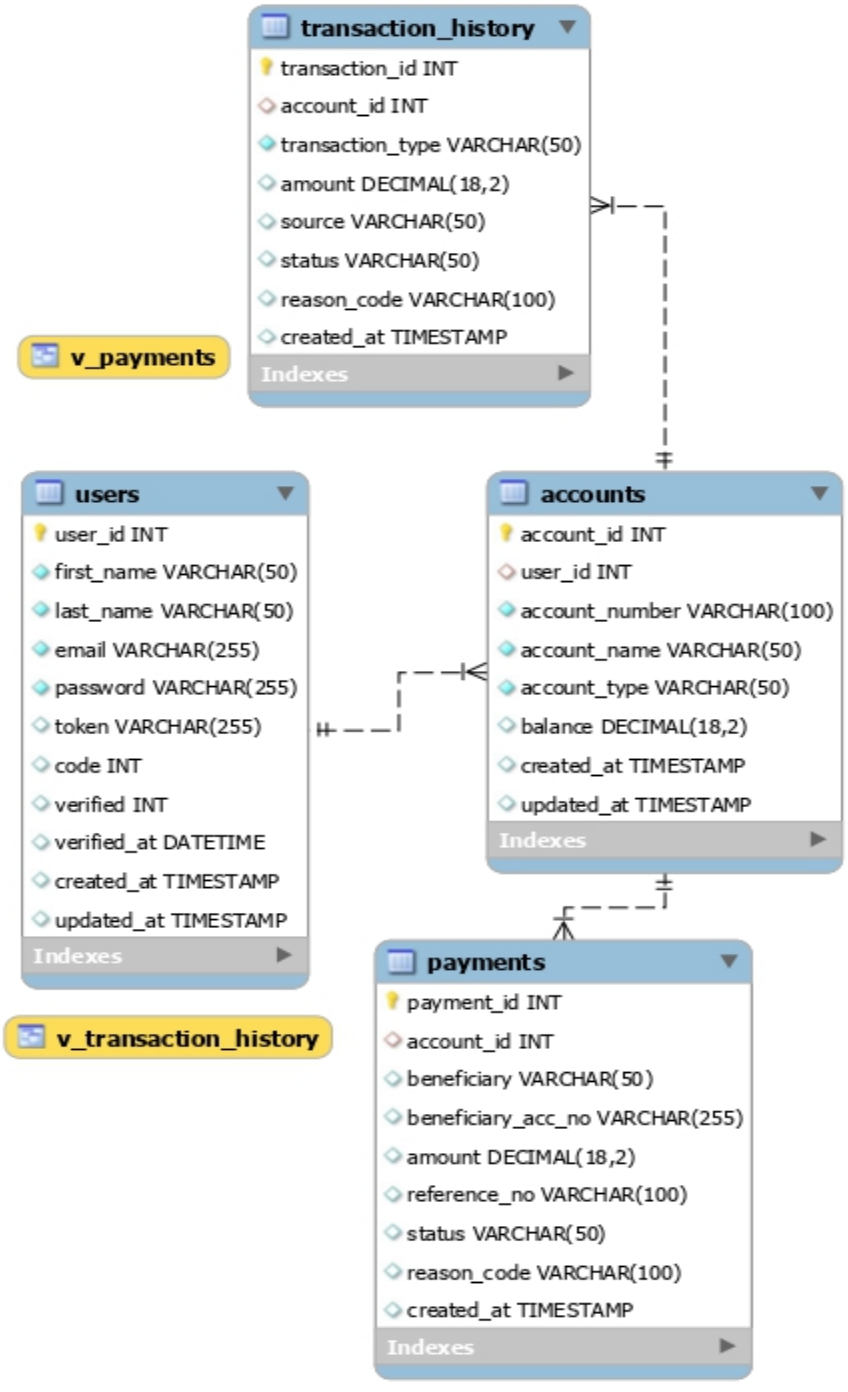
Database Constraints:

- Foreign key constraints ensure referential integrity between related tables.
- Unique constraints on email and account_number fields to ensure uniqueness.

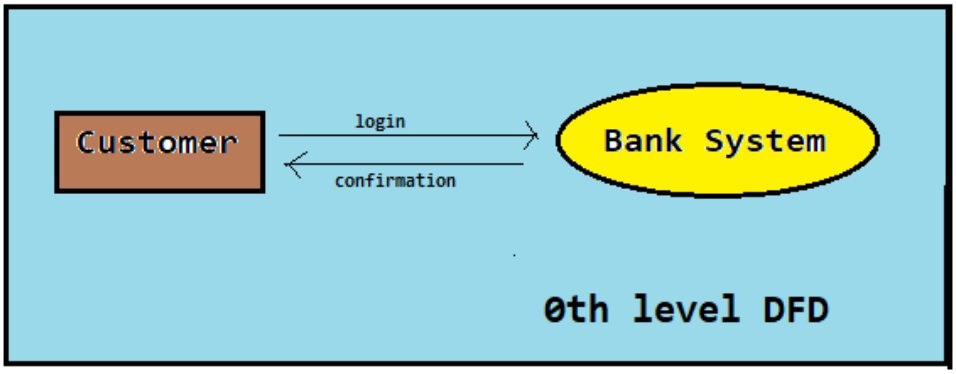
This database design reflects the relationships and attributes defined in the provided schema. It allows for efficient storage and retrieval of user, account, transaction, and payment data. Properly defined constraints ensure data integrity, and relationships ensure accurate representation of associations between entities.

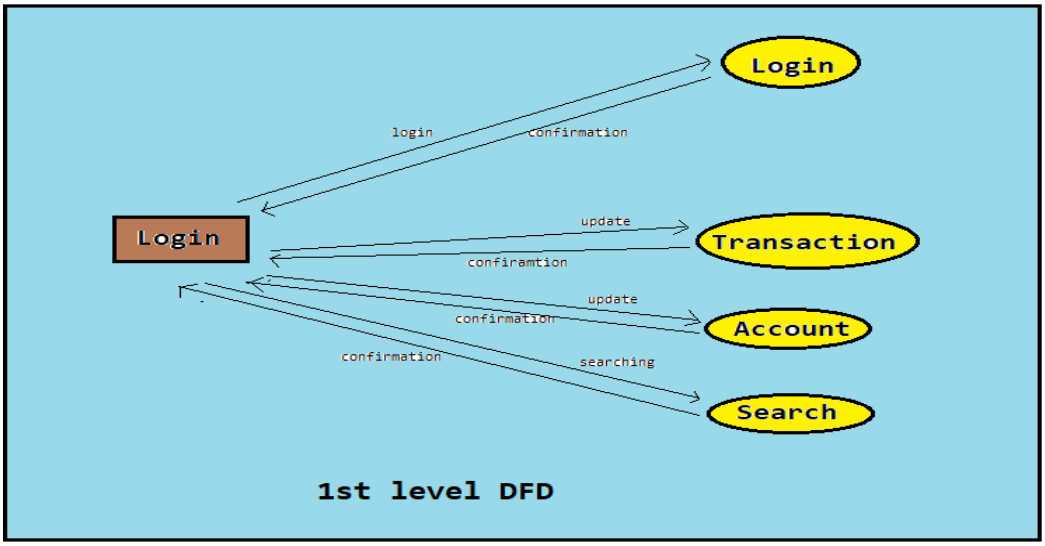
However, please note that this is a high-level overview of the database design. Depending on the specifics of your project, additional considerations such as indexes, data types, and normalization may be necessary to optimize the database further. It's recommended to collaborate with a database designer or administrator to fine-tune the design based on your project's requirements and performance needs.

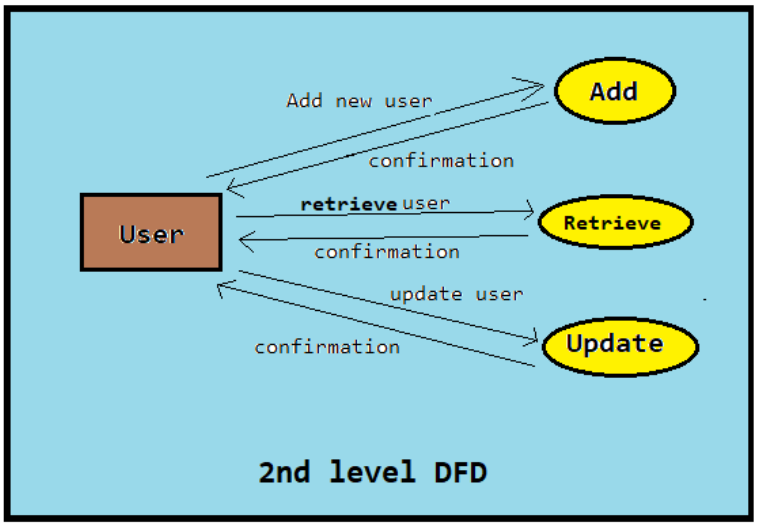
ER Diagram


transaction_history
transaction_id INT
account_id INT
transaction_type VARCHAR(50)
amount DECIMAL(18,2)
source VARCHAR(50)
status VARCHAR(50)
reason_code VARCHAR(100)
created_at TIMESTAMP
Indexes
v_payments
users
user_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
email VARCHAR(255)
password VARCHAR(255)
token VARCHAR(255)
code INT
verified INT
verified_at DATETIME
created_at TIMESTAMP
updated_at TIMESTAMP
Indexes
v_transaction_history
accounts
account_id INT
user_id INT
account_number VARCHAR(100)
account_name VARCHAR(50)
account_type VARCHAR(50)
balance DECIMAL(18,2)
created_at TIMESTAMP
updated_at TIMESTAMP
Indexes
payments
payment_id INT
account_id INT
beneficiary VARCHAR(50)
beneficiary_acc_no VARCHAR(255)
amount DECIMAL(18,2)
reference_no VARCHAR(100)
status VARCHAR(50)
reason_code VARCHAR(100)
created_at TIMESTAMP
Indexes

DFD Diagram


Customer
login
confirmation
Bank System
0th level DFD


Login
login
confirmation
Login
update
confiramtion
Transaction
update
confirmation
Account
confirmation
searching
Search
1st level DFD


Add
Add new user
confirmation
User
retrieve user
confirmation
Retrieve
update user
confirmation
Update
2nd level DFD

Sequence Diagram

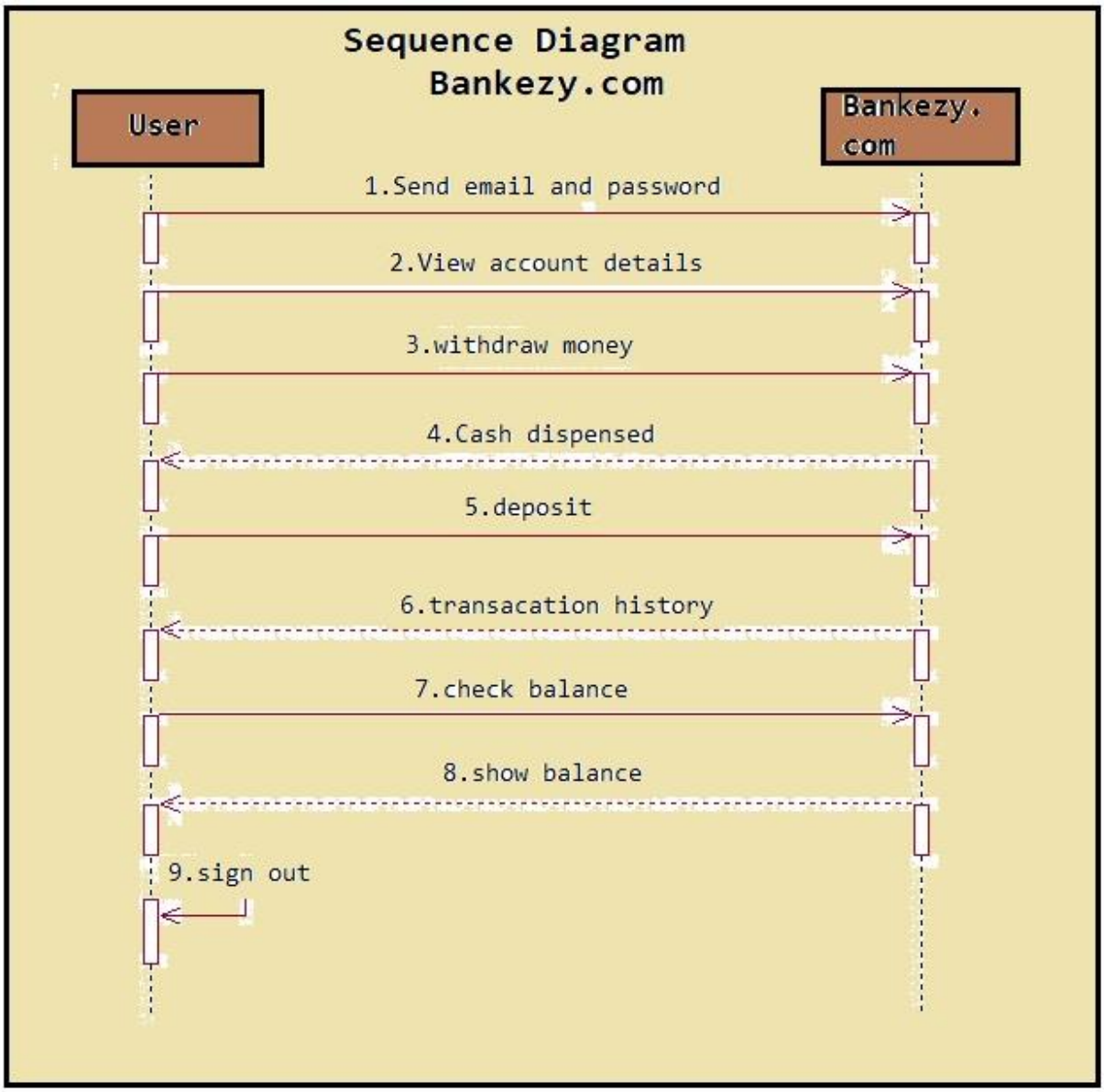


Table Structure

1.Database: bankezy

MySQL 8.0 Command Line Client

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| advjava            |
| advjava2           |
| advjava3           |
| advjava4           |
| avdjava2           |
| bankezy            |
| bankezy_hmail      |
| bankezy_new        |
| demo_bank_v1       |
| dotnetemplmvc      |
| emp                |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
| testing            |
| transflower        |
+--------------------+
18 rows in set (0.01 sec)

mysql> use bankezy;
Database changed
mysql> show tables;
+-----------------------+
| Tables_in_bankezy     |
+-----------------------+
| account               |
| accounts              |
| payment               |
| payments              |
| transact              |
| transaction_history   |
| users                 |
| v_payments            |
| v_transaction_history |
+-----------------------+
9 rows in set (0.00 sec)
```

2.Table payments:

3.Table users:

4.Table Transact:

5.Table transaction_history:

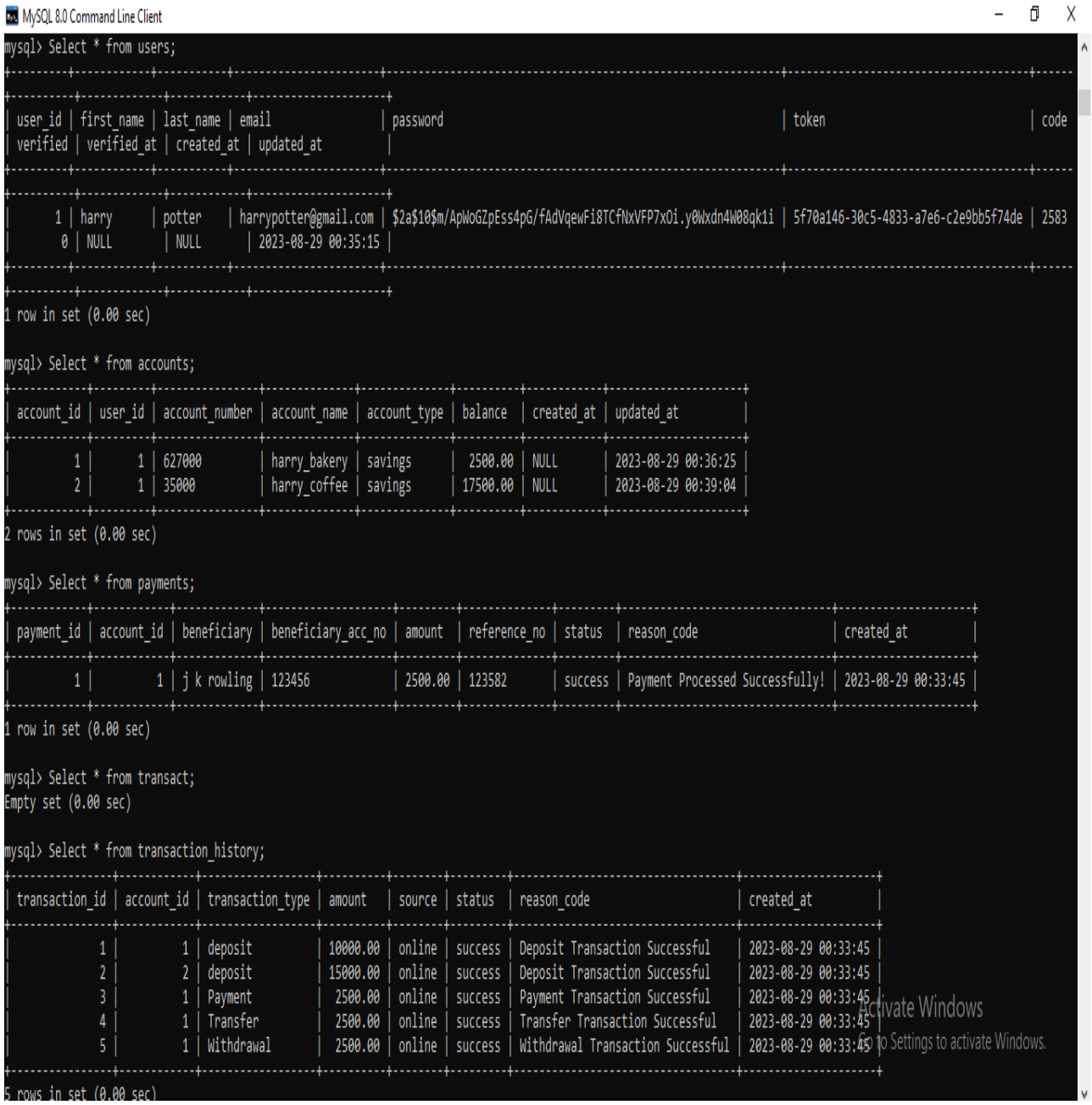
MySQL 8.0 Command Line Client

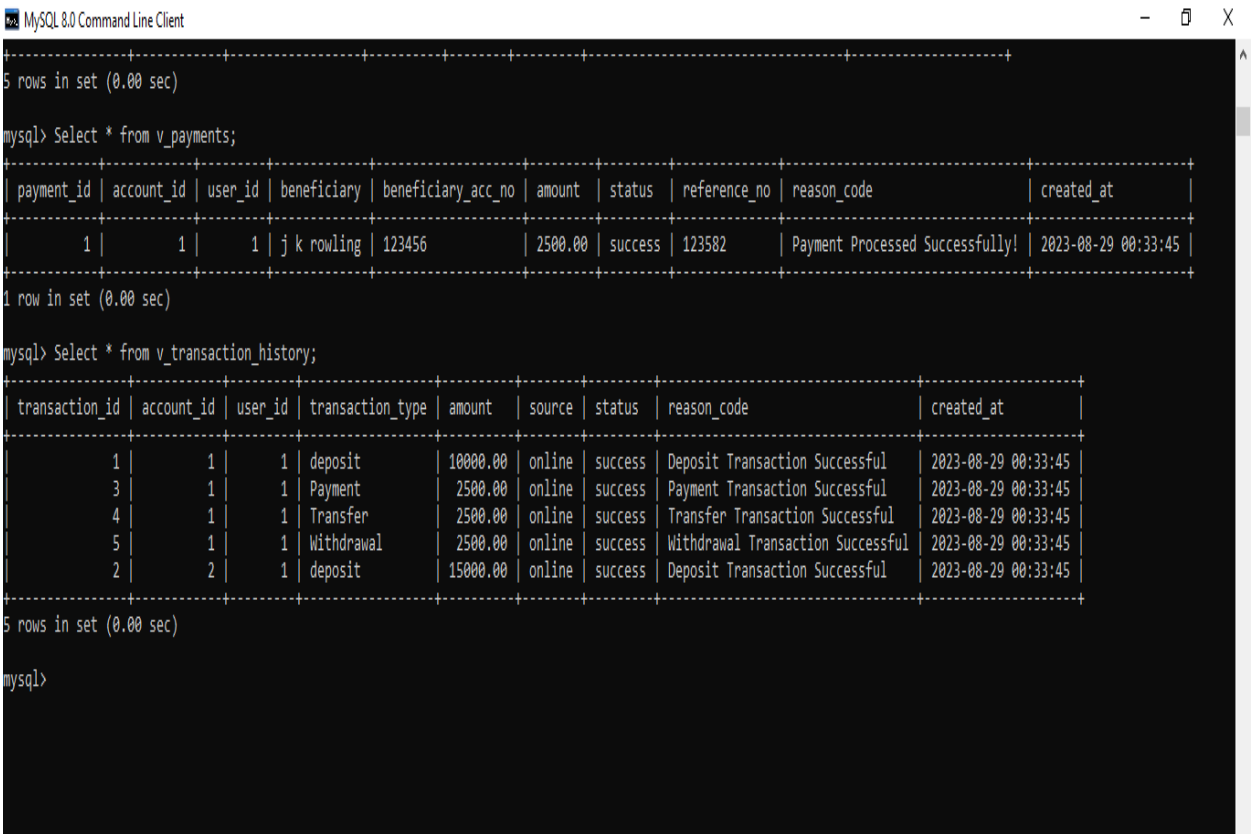
```
Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use bankezy;
Database changed
mysql> desc users;
+-------------+--------------+------+-----+-------------------+-------------------+
| Field       | Type         | Null | Key | Default           | Extra             |
+-------------+--------------+------+-----+-------------------+-------------------+
| user_id     | int          | NO   | PRI | NULL              | auto_increment    |
| first_name  | varchar(50)  | NO   |     | NULL              |                   |
| last_name   | varchar(50)  | NO   |     | NULL              |                   |
| email       | varchar(255) | NO   | UNI | NULL              |                   |
| password    | varchar(255) | NO   |     | NULL              |                   |
| token       | varchar(255) | YES  |     | NULL              |                   |
| code        | int          | YES  |     | NULL              |                   |
| verified    | int          | YES  |     | 0                 |                   |
| verified_at | datetime     | YES  |     | NULL              |                   |
| created_at  | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| updated_at  | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-------------+--------------+------+-----+-------------------+-------------------+
11 rows in set (0.04 sec)

mysql>
```

MySQL 8.0 Command Line Client

```
mysql> desc payments;
+--------------------+---------------+------+-----+---------+----------------+
| Field              | Type          | Null | Key | Default | Extra          |
+--------------------+---------------+------+-----+---------+----------------+
| payment_id         | int           | NO   | PRI | NULL    | auto_increment |
| account_id         | int           | YES  | MUL | NULL    |                |
| beneficiary        | varchar(50)   | YES  |     | NULL    |                |
| beneficiary_acc_no | varchar(255)  | YES  |     | NULL    |                |
| amount             | decimal(18,2) | YES  |     | NULL    |                |
| reference_no       | varchar(100)  | YES  |     | NULL    |                |
| status             | varchar(50)   | YES  |     | NULL    |                |
| reason_code        | varchar(100)  | YES  |     | NULL    |                |
| created_at         | timestamp     | YES  |     | NULL    |                |
+--------------------+---------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)

mysql> desc transact;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| transaction_id   | int          | NO   | PRI | NULL    |       |
| account_id       | int          | NO   |     | NULL    |       |
| amount           | double       | NO   |     | NULL    |       |
| created_at       | datetime(6)  | YES  |     | NULL    |       |
| reason_code      | varchar(255) | YES  |     | NULL    |       |
| source           | varchar(255) | YES  |     | NULL    |       |
| status           | varchar(255) | YES  |     | NULL    |       |
| transaction_type | varchar(255) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
8 rows in set (0.01 sec)

mysql>
mysql> Desc transaction_history;
+------------------+---------------+------+-----+---------+----------------+
| Field            | Type          | Null | Key | Default | Extra          |
+------------------+---------------+------+-----+---------+----------------+
| transaction_id   | int           | NO   | PRI | NULL    | auto_increment |
| account_id       | int           | YES  | MUL | NULL    |                |
| transaction_type | varchar(50)   | NO   |     | NULL    |                |
| amount           | decimal(18,2) | YES  |     | NULL    |                |
| source           | varchar(50)   | YES  |     | NULL    |                |
| status           | varchar(50)   | YES  |     | NULL    |                |
| reason_code      | varchar(100)  | YES  |     | NULL    |                |
| created_at       | timestamp     | YES  |     | NULL    |                |
```

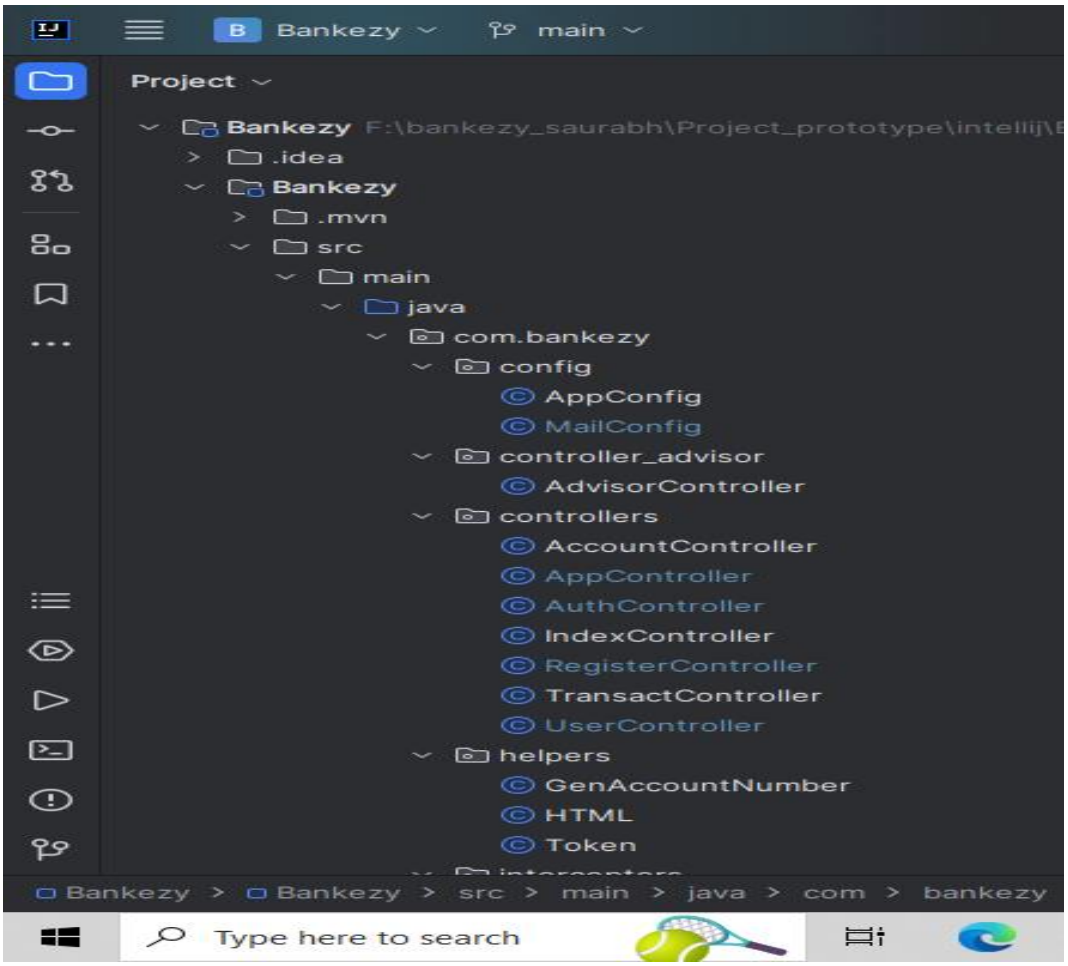
```
mysql> Desc v_payments;
+--------------------+---------------+------+-----+---------+-------+
| Field              | Type          | Null | Key | Default | Extra |
+--------------------+---------------+------+-----+---------+-------+
| payment_id         | int           | NO   |     | 0       |       |
| account_id         | int           | NO   |     | 0       |       |
| user_id            | int           | NO   |     | 0       |       |
| beneficiary        | varchar(50)   | YES  |     | NULL    |       |
| beneficiary_acc_no | varchar(255)  | YES  |     | NULL    |       |
| amount             | decimal(18,2) | YES  |     | NULL    |       |
| status             | varchar(50)   | YES  |     | NULL    |       |
| reference_no       | varchar(100)  | YES  |     | NULL    |       |
| reason_code        | varchar(100)  | YES  |     | NULL    |       |
| created_at         | timestamp     | YES  |     | NULL    |       |
+--------------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

mysql> Desc v_transaction_history;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| transaction_id   | int           | NO   |     | 0       |       |
| account_id       | int           | NO   |     | 0       |       |
| user_id          | int           | NO   |     | 0       |       |
| transaction_type | varchar(50)   | NO   |     | NULL    |       |
| amount           | decimal(18,2) | YES  |     | NULL    |       |
| source           | varchar(50)   | YES  |     | NULL    |       |
| status           | varchar(50)   | YES  |     | NULL    |       |
| reason_code      | varchar(100)  | YES  |     | NULL    |       |
| created_at       | timestamp     | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

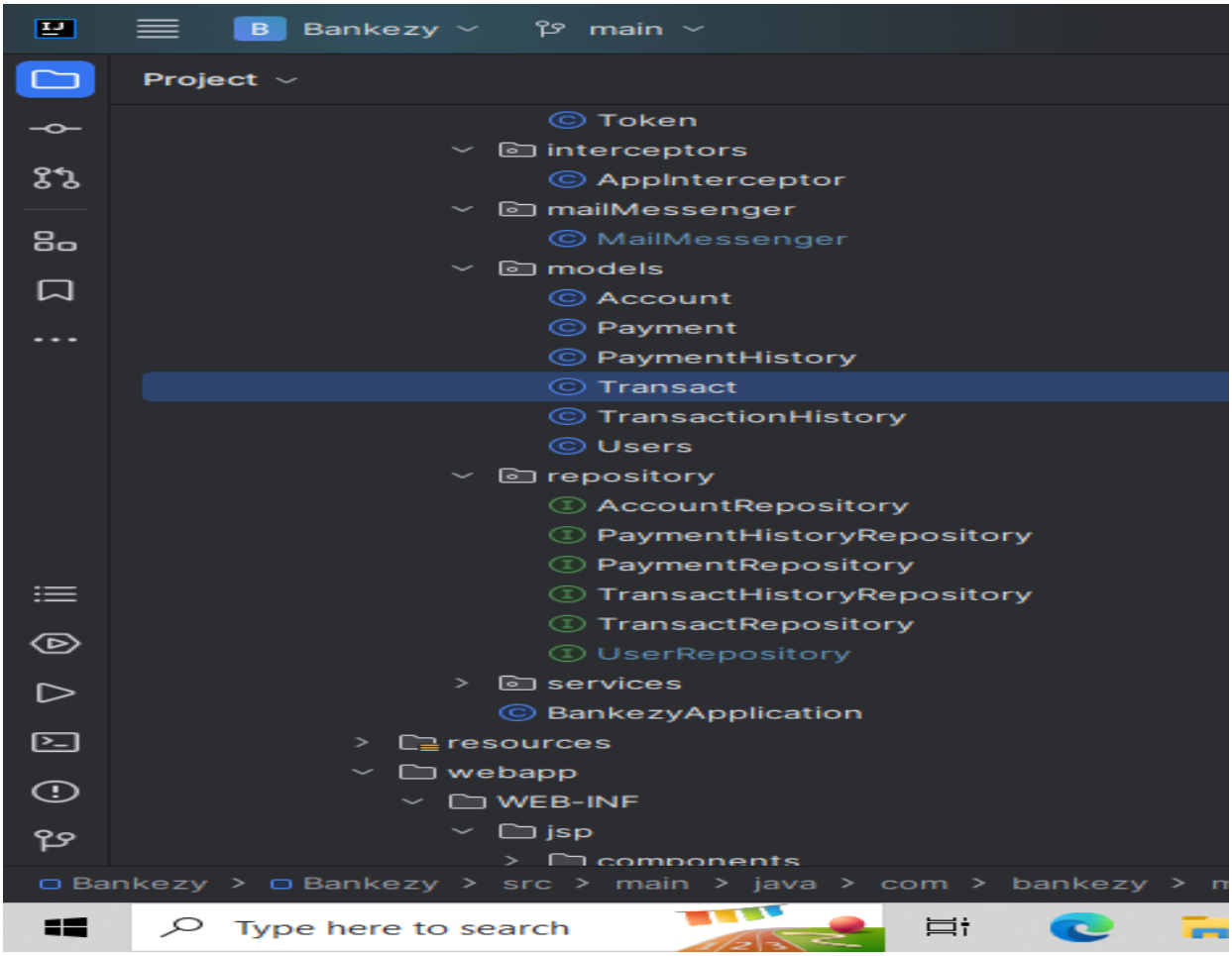

MySQL 8.0 Command Line Client
mysql> Select * from users;
+---------+------------+-----------+--------------------+--+--------------------------------------+------+
| user_id | first_name | last_name | email | password | token | code |
| verified | verified_at | created_at | updated_at |
+---------+------------+-----------+--------------------+--+--------------------------------------+------+
| 1 | harry | potter | harrypotter@gmail.com | $2a$10$m/ApWoGZpEss4pG/fAdVqewFi8TCfNxVFP7xOi.y0Wxdn4W08qk1i | 5f70a146-30c5-4833-a7e6-c2e9bb5f74de | 2583 |
| 0 | NULL | NULL | 2023-08-29 00:35:15 |
1 row in set (0.00 sec)
mysql> Select * from accounts;
account_id	user_id	account_number	account_name	account_type	balance	created_at	updated_at
1	1	627000	harry_bakery	savings	2500.00	NULL	2023-08-29 00:36:25
2	1	35000	harry_coffee	savings	17500.00	NULL	2023-08-29 00:39:04
2 rows in set (0.00 sec)
mysql> Select * from payments;
payment_id	account_id	beneficiary	beneficiary_acc_no	amount	reference_no	status	reason_code	created_at
1	1	j k rowling	123456	2500.00	123582	success	Payment Processed Successfully!	2023-08-29 00:33:45
1 row in set (0.00 sec)
mysql> Select * from transact;
Empty set (0.00 sec)
mysql> Select * from transaction_history;
transaction_id	account_id	transaction_type	amount	source	status	reason_code	created_at
1	1	deposit	10000.00	online	success	Deposit Transaction Successful	2023-08-29 00:33:45
2	2	deposit	15000.00	online	success	Deposit Transaction Successful	2023-08-29 00:33:45
3	1	Payment	2500.00	online	success	Payment Transaction Successful	2023-08-29 00:33:45
4	1	Transfer	2500.00	online	success	Transfer Transaction Successful	2023-08-29 00:33:45
5	1	Withdrawal	2500.00	online	success	Withdrawal Transaction Successful	2023-08-29 00:33:45
5 rows in set (0.00 sec)
Activate Windows
Go to Settings to activate Windows.

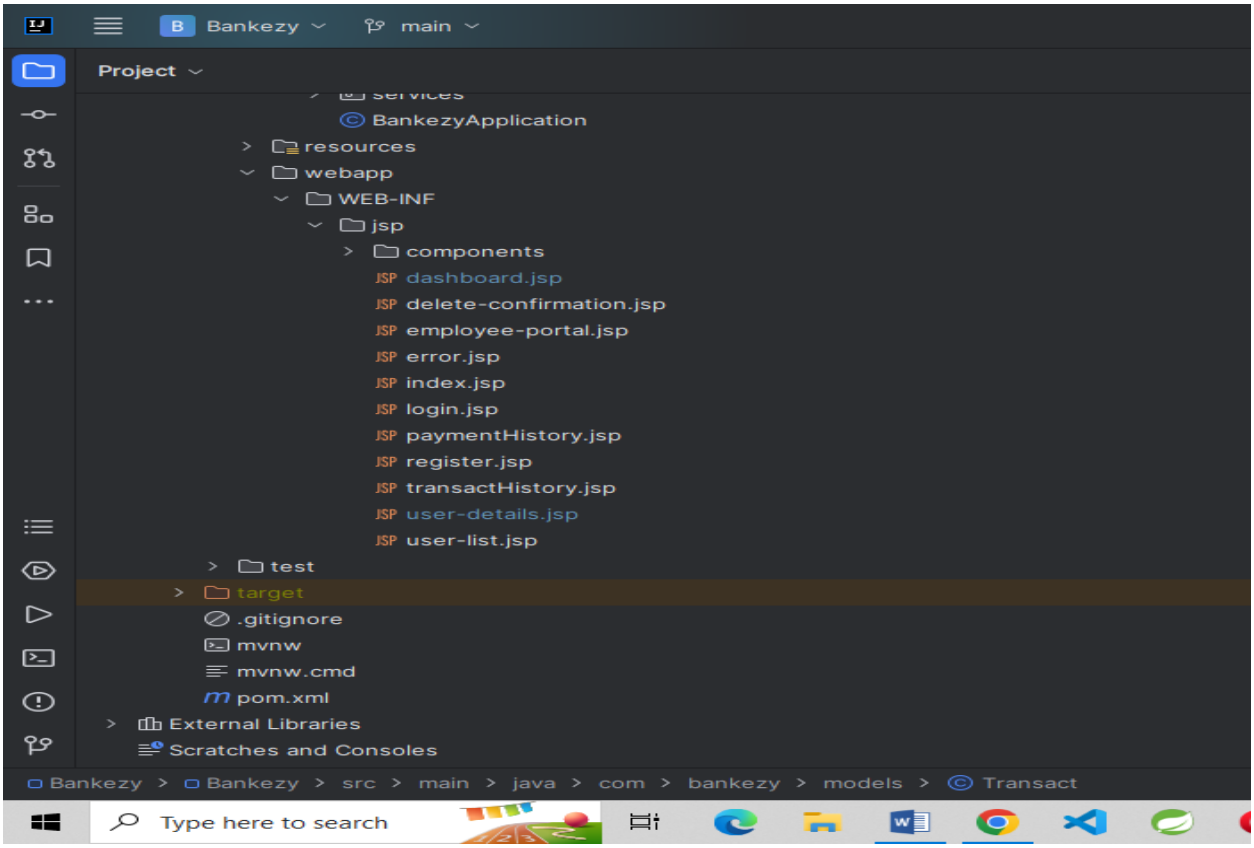

MySQL 8.0 Command Line Client
5 rows in set (0.00 sec)
mysql> Select * from v_payments;
| payment_id | account_id | user_id | beneficiary | beneficiary_acc_no | amount | status | reference_no | reason_code | created_at |
| 1 | 1 | 1 | j k rowling | 123456 | 2500.00 | success | 123582 | Payment Processed Successfully! | 2023-08-29 00:33:45 |
1 row in set (0.00 sec)
mysql> Select * from v_transaction_history;
transaction_id	account_id	user_id	transaction_type	amount	source	status	reason_code	created_at
1	1	1	deposit	10000.00	online	success	Deposit Transaction Successful	2023-08-29 00:33:45
3	1	1	Payment	2500.00	online	success	Payment Transaction Successful	2023-08-29 00:33:45
4	1	1	Transfer	2500.00	online	success	Transfer Transaction Successful	2023-08-29 00:33:45
5	1	1	Withdrawal	2500.00	online	success	Withdrawal Transaction Successful	2023-08-29 00:33:45
2	2	1	deposit	15000.00	online	success	Deposit Transaction Successful	2023-08-29 00:33:45
5 rows in set (0.00 sec)
mysql>

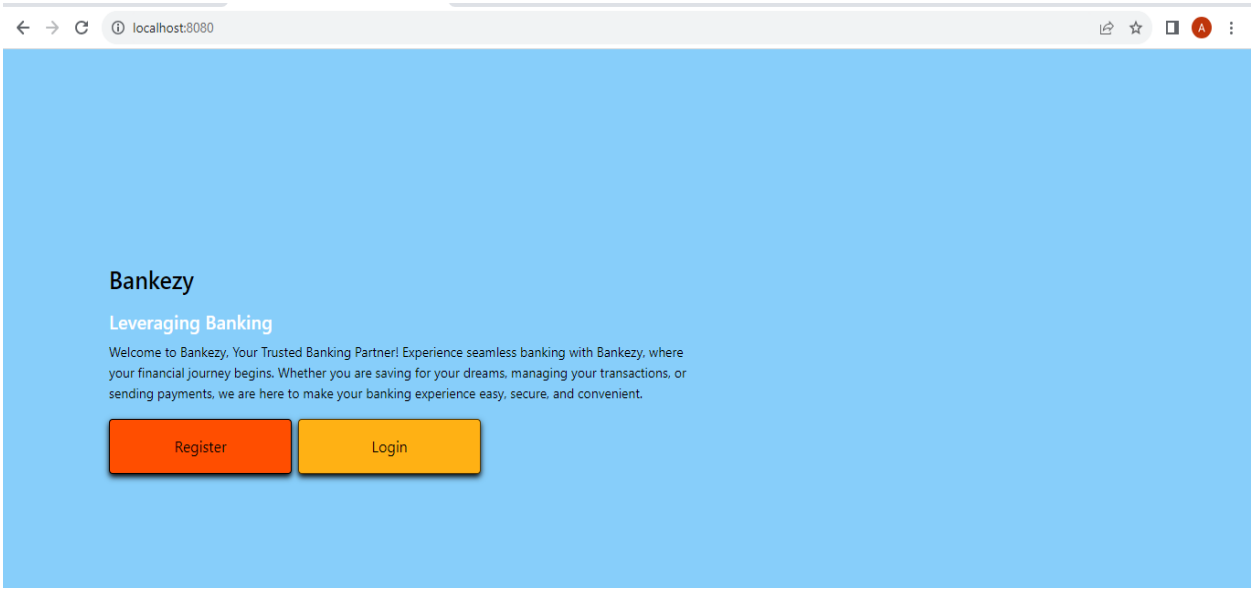
Project Diagrams

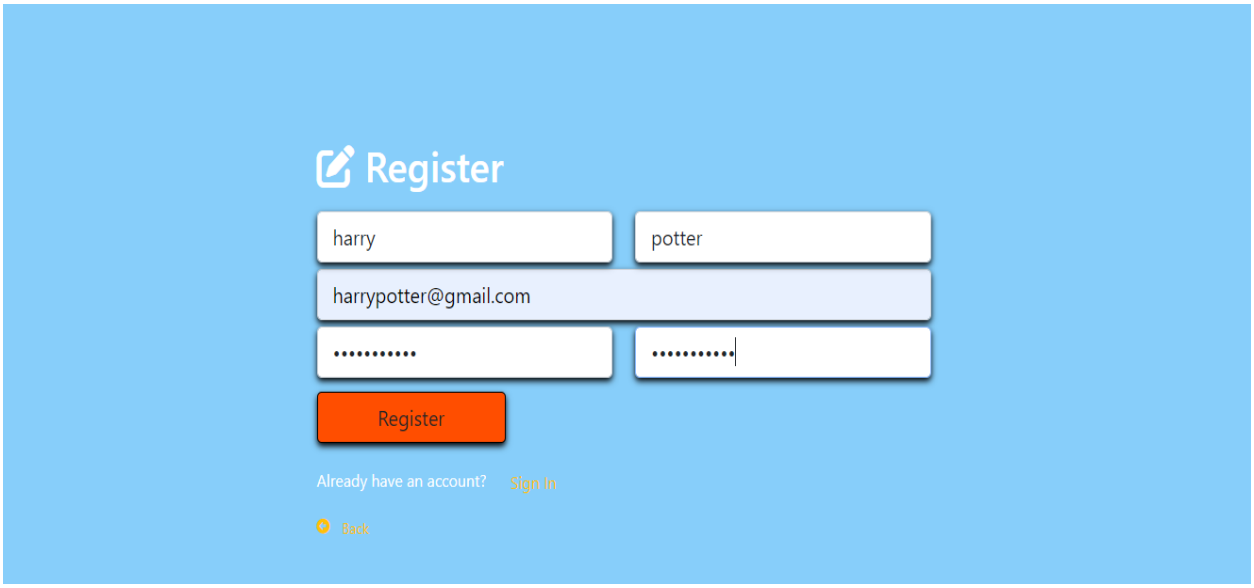
Folder Structure :

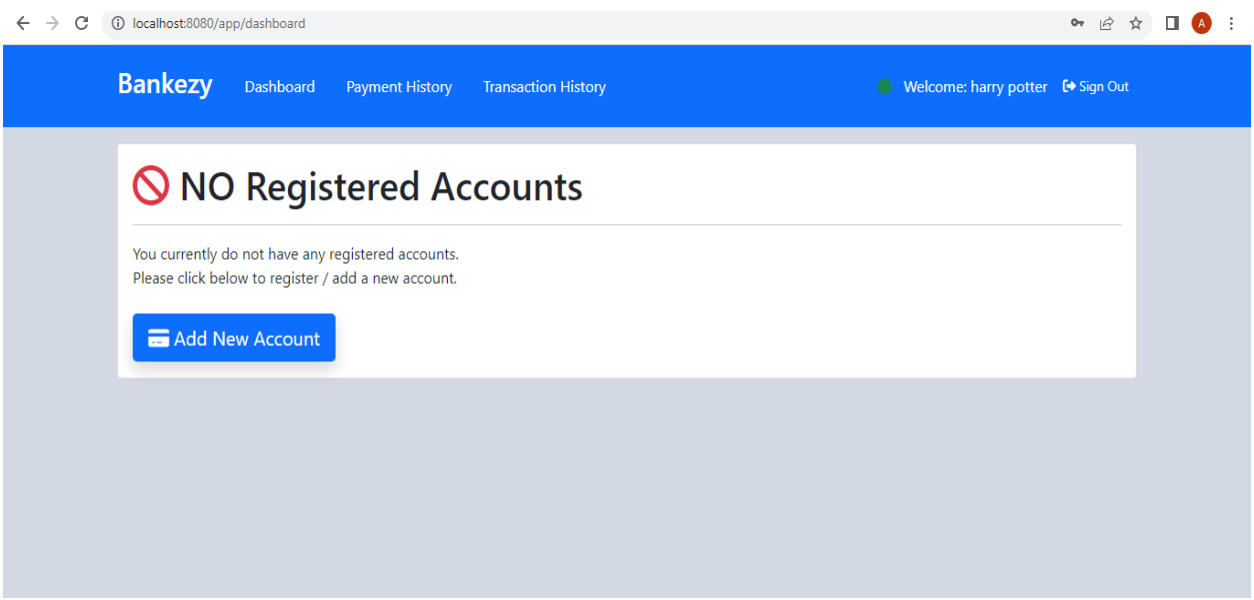


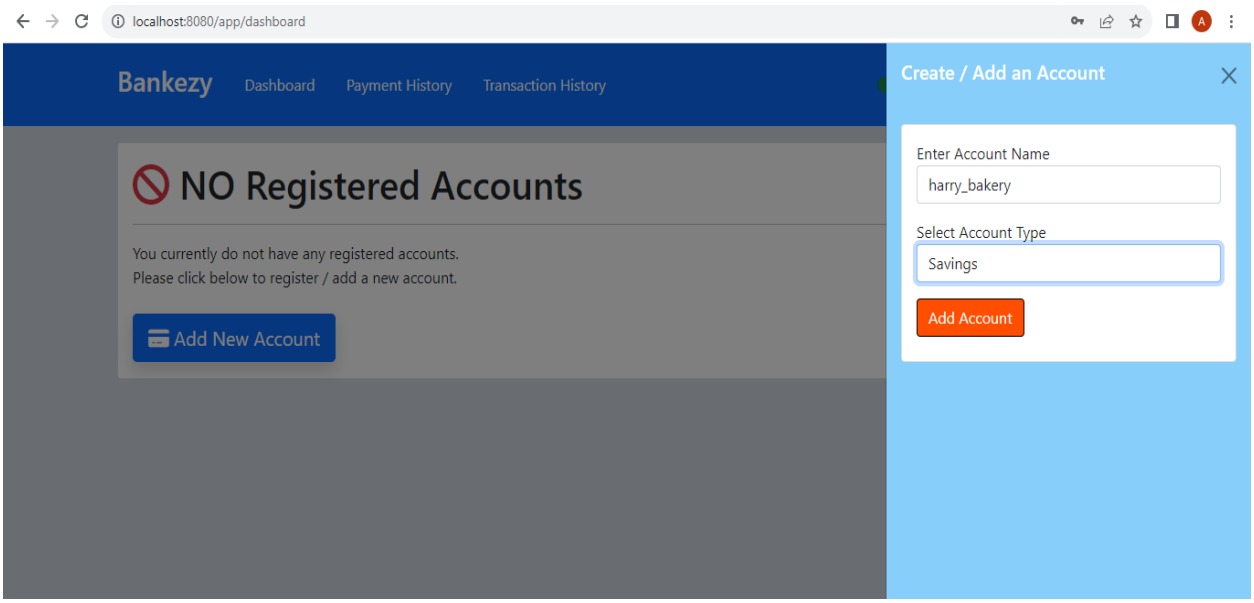

Bankezy
main
Project
Token
interceptors
AppInterceptor
mailMessenger
MailMessenger
models
Account
Payment
PaymentHistory
Transact
TransactionHistory
Users
repository
AccountRepository
PaymentHistoryRepository
PaymentRepository
TransactHistoryRepository
TransactRepository
UserRepository
services
BankezyApplication
resources
webapp
WEB-INF
jsp
components
Bankezy > Bankezy > src > main > java > com > bankezy
Type here to search

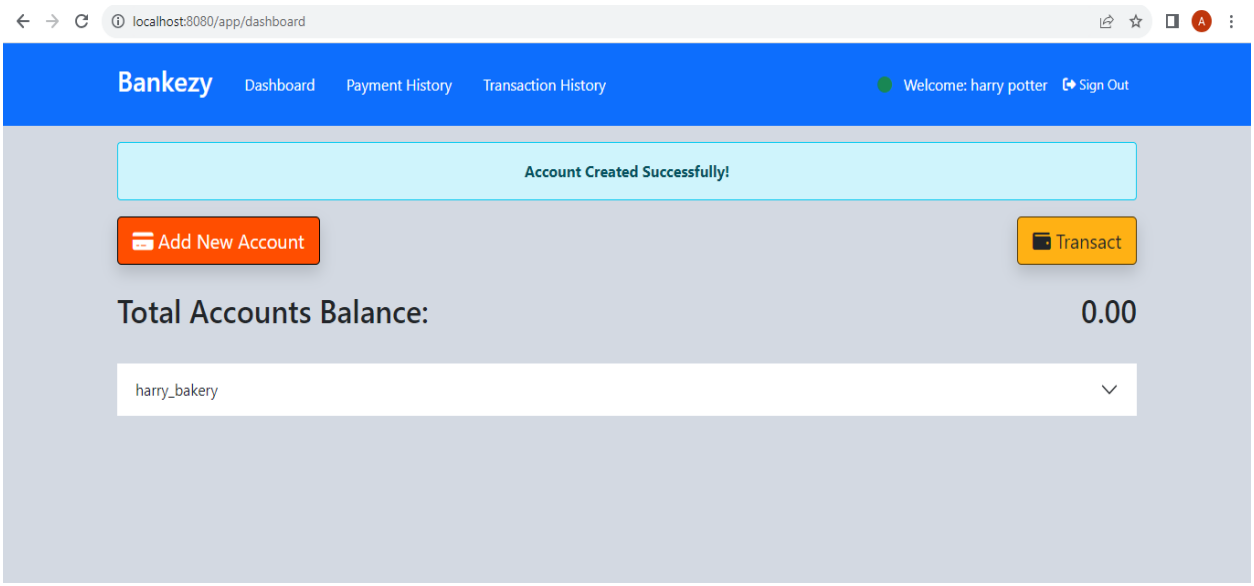

Bankezy
main
Project
BankezyApplication
resources
webapp
WEB-INF
jsp
components
dashboard.jsp
delete-confirmation.jsp
employee-portal.jsp
error.jsp
index.jsp
login.jsp
paymentHistory.jsp
register.jsp
transactHistory.jsp
user-details.jsp
user-list.jsp
test
target
.gitignore
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
Bankezy > Bankezy > src > main > java > com > bankezy > models > Transact
Type here to search

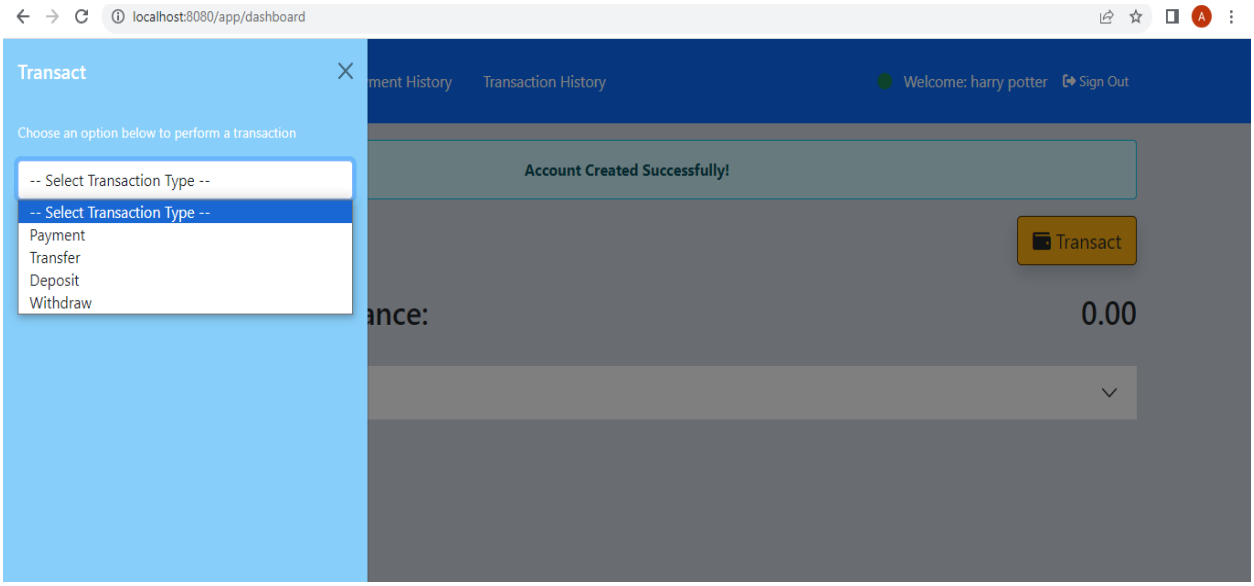

localhost:8080
Bankezy
Leveraging Banking
Welcome to Bankezy, Your Trusted Banking Partner! Experience seamless banking with Bankezy, where your financial journey begins. Whether you are saving for your dreams, managing your transactions, or sending payments, we are here to make your banking experience easy, secure, and convenient.
Register
Login

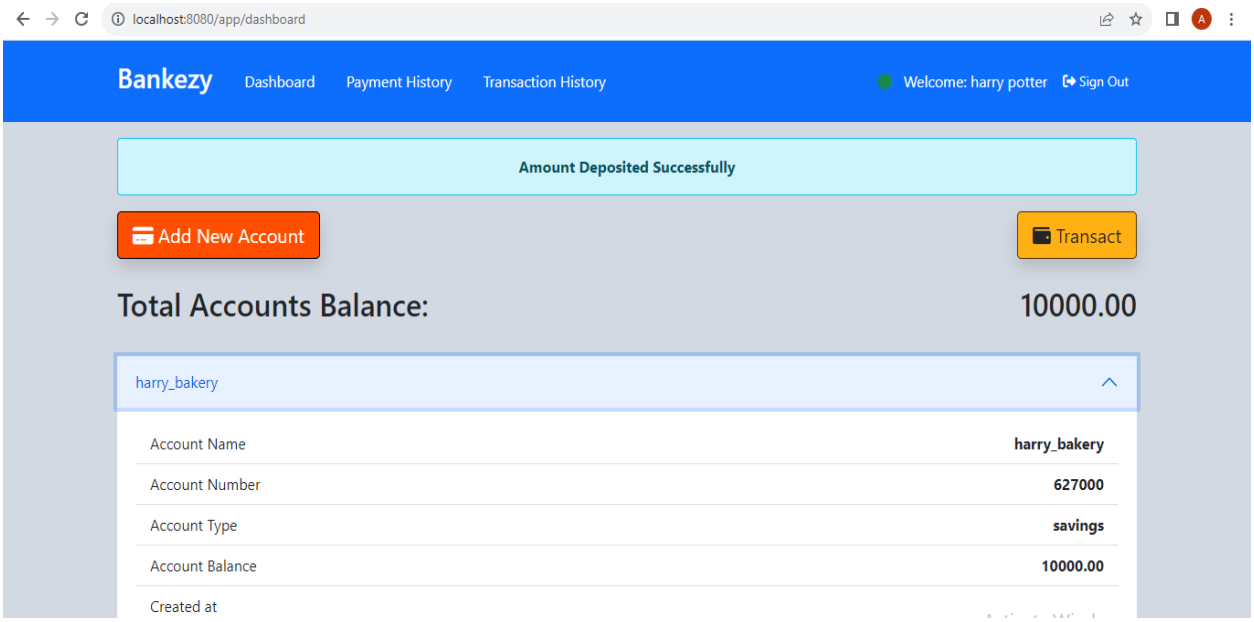

Register
harry
potter
harrypotter@gmail.com
Register
Already have an account?
Sign In
Back

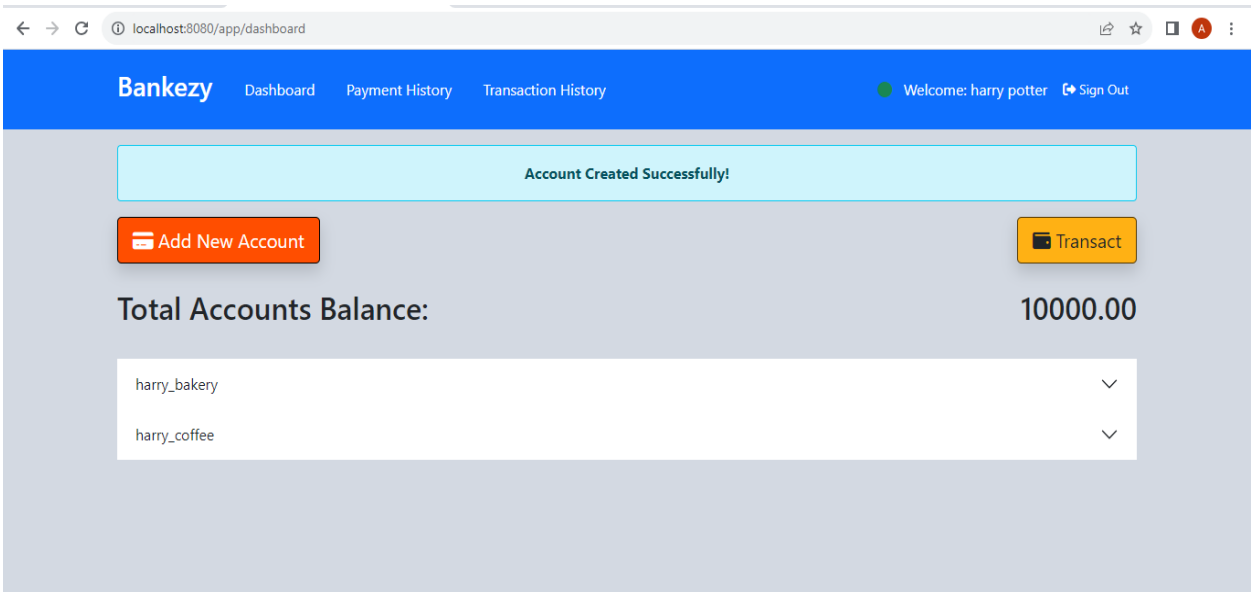

localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
NO Registered Accounts
You currently do not have any registered accounts.
Please click below to register / add a new account.
Add New Account

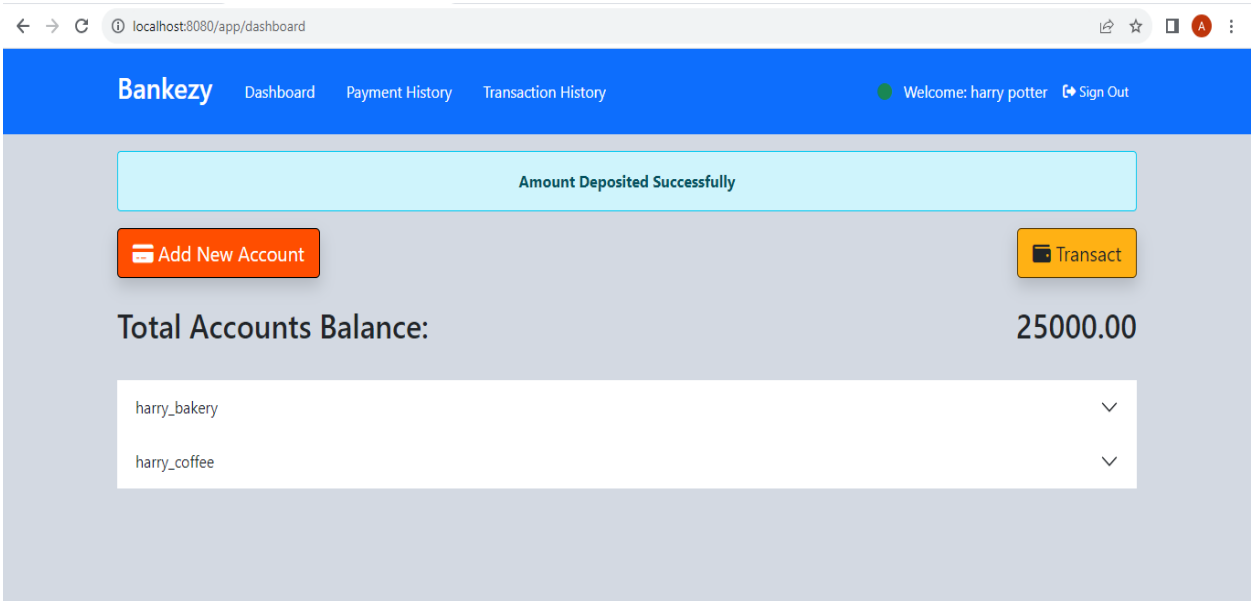

localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
NO Registered Accounts
You currently do not have any registered accounts.
Please click below to register / add a new account.
Add New Account
Create / Add an Account
Enter Account Name
harry_bakery
Select Account Type
Savings
Add Account

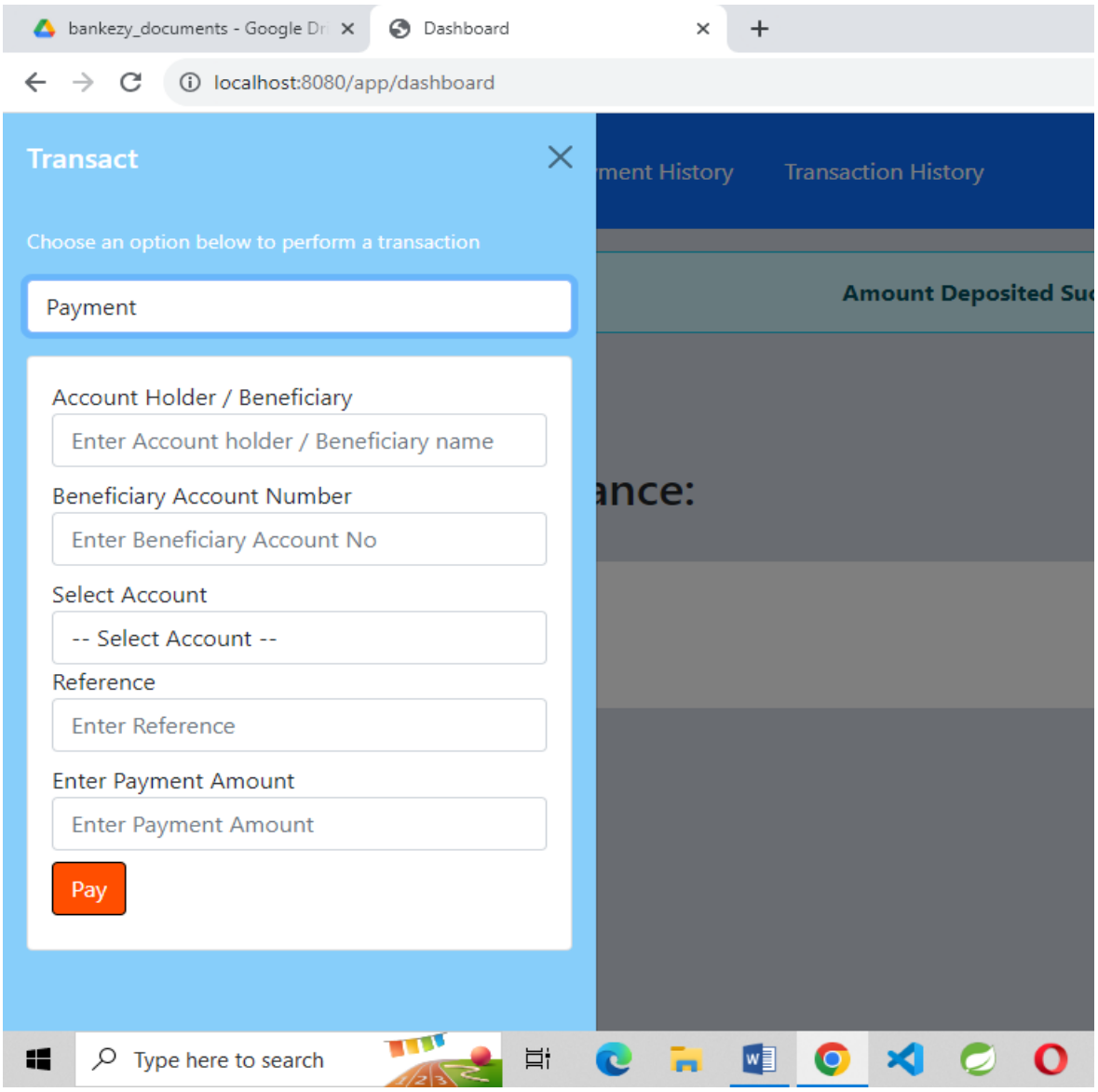

localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Account Created Successfully!
Add New Account
Transact
Total Accounts Balance:
0.00
harry_bakery

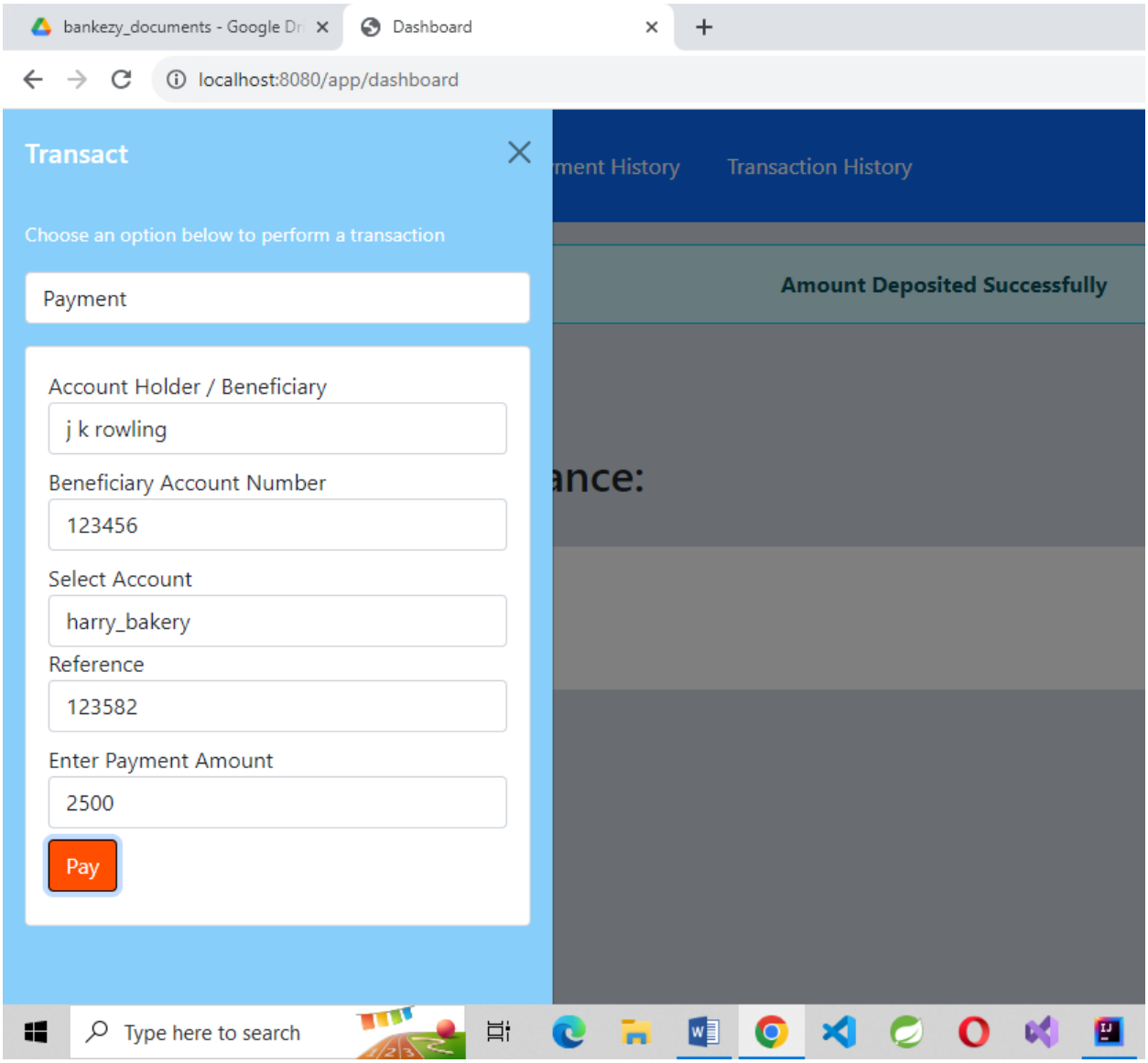

localhost:8080/app/dashboard
Transact
Choose an option below to perform a transaction
-- Select Transaction Type --
-- Select Transaction Type --
Payment
Transfer
Deposit
Withdraw
ment History
Transaction History
Welcome: harry potter
Sign Out
Account Created Successfully!
Transact
ance:
0.00

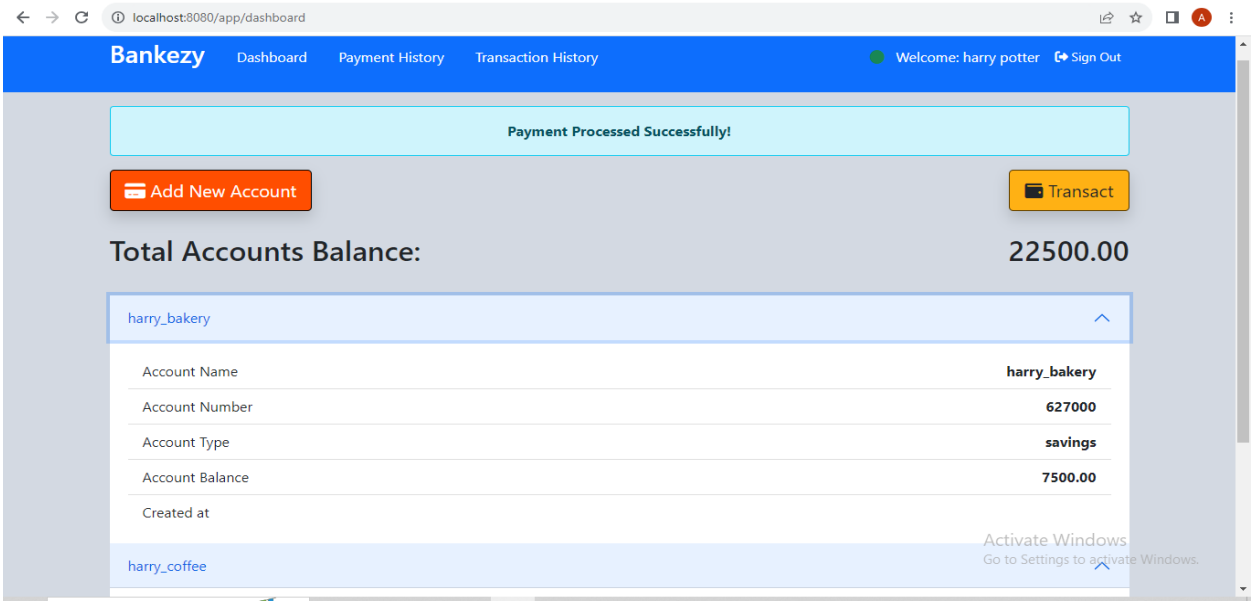

localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Amount Deposited Successfully
Add New Account
Transact
Total Accounts Balance:
10000.00
harry_bakery
Account Name
harry_bakery
Account Number
627000
Account Type
savings
Account Balance
10000.00
Created at

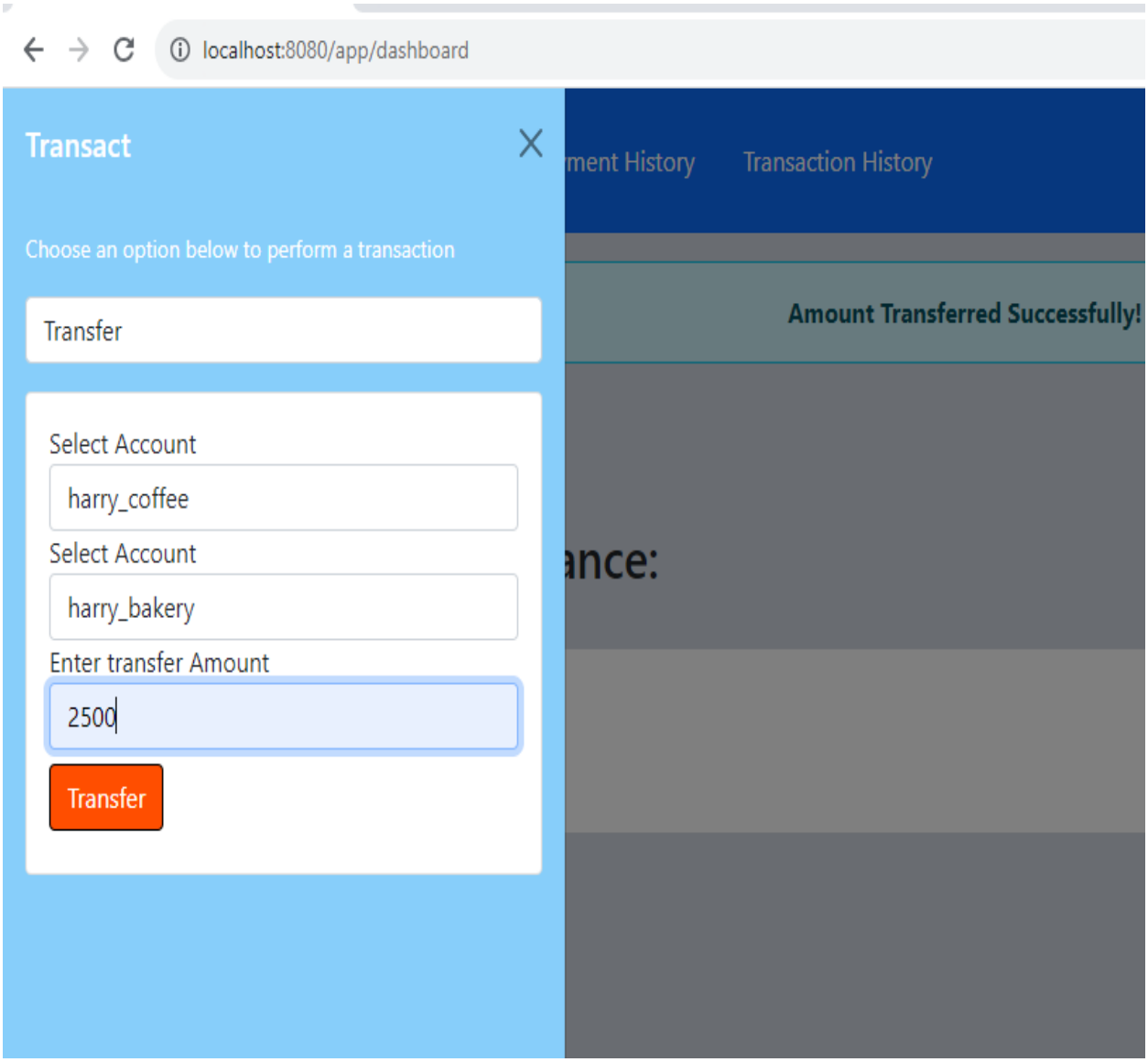

localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Account Created Successfully!
Add New Account
Transact
Total Accounts Balance:
10000.00
harry_bakery
harry_coffee

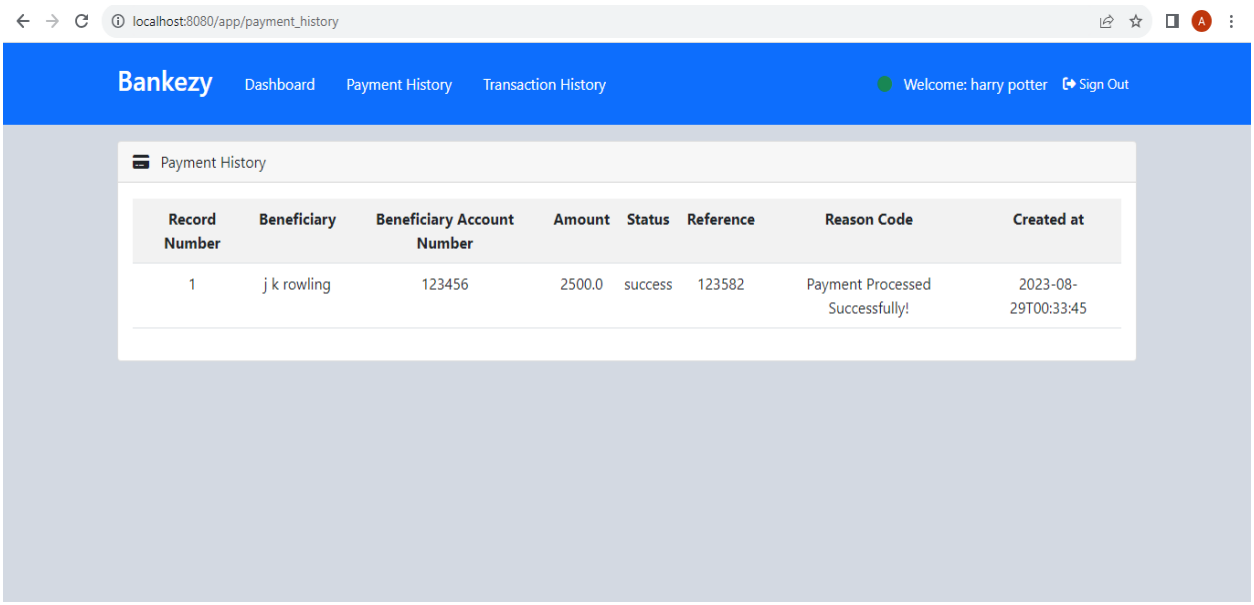

localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Amount Deposited Successfully
Add New Account
Transact
Total Accounts Balance:
25000.00
harry_bakery
harry_coffee

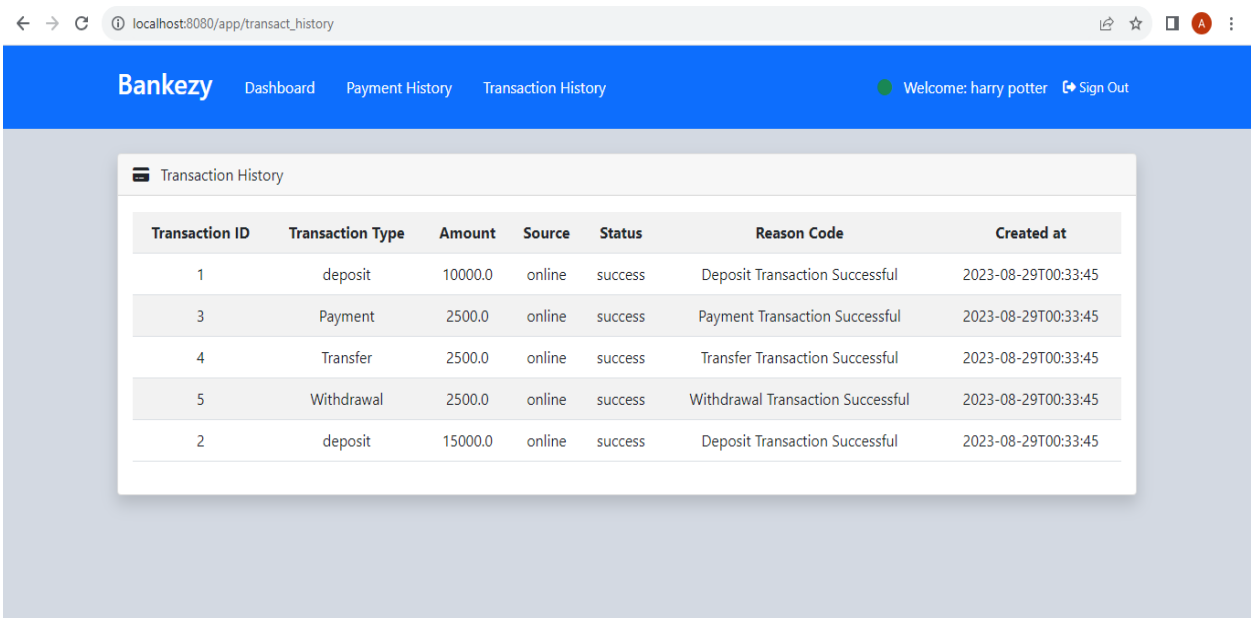

bankezy_documents - Google Dr
Dashboard
localhost:8080/app/dashboard
Transact
Choose an option below to perform a transaction
Payment
Account Holder / Beneficiary
Enter Account holder / Beneficiary name
Beneficiary Account Number
Enter Beneficiary Account No
Select Account
-- Select Account --
Reference
Enter Reference
Enter Payment Amount
Enter Payment Amount
Pay
ment History
Transaction History
Amount Deposited Suc
ance:
Type here to search


bankezy_documents - Google Dr
Dashboard
localhost:8080/app/dashboard
Transact
Choose an option below to perform a transaction
Payment
Account Holder / Beneficiary
j k rowling
Beneficiary Account Number
123456
Select Account
harry_bakery
Reference
123582
Enter Payment Amount
2500
Pay
ment History
Transaction History
Amount Deposited Successfully
ance:
Type here to search


localhost:8080/app/dashboard
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Payment Processed Successfully!
Add New Account
Transact
Total Accounts Balance:
22500.00
harry_bakery
Account Name
harry_bakery
Account Number
627000
Account Type
savings
Account Balance
7500.00
Created at
harry_coffee
Activate Windows
Go to Settings to activate Windows.


localhost:8080/app/dashboard
Transact
Choose an option below to perform a transaction
Transfer
Select Account
harry_coffee
Select Account
harry_bakery
Enter transfer Amount
2500
Transfer
ment History
Transaction History
Amount Transferred Successfully!
ance:


localhost:8080/app/payment_history
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Payment History
Record Number
Beneficiary
Beneficiary Account Number
Amount
Status
Reference
Reason Code
Created at
1
j k rowling
123456
2500.0
success
123582
Payment Processed Successfully!
2023-08-29T00:33:45


localhost:8080/app/transact_history
Bankezy
Dashboard
Payment History
Transaction History
Welcome: harry potter
Sign Out
Transaction History
Transaction ID
Transaction Type
Amount
Source
Status
Reason Code
Created at
1
deposit
10000.0
online
success
Deposit Transaction Successful
2023-08-29T00:33:45
3
Payment
2500.0
online
success
Payment Transaction Successful
2023-08-29T00:33:45
4
Transfer
2500.0
online
success
Transfer Transaction Successful
2023-08-29T00:33:45
5
Withdrawal
2500.0
online
success
Withdrawal Transaction Successful
2023-08-29T00:33:45
2
deposit
15000.0
online
success
Deposit Transaction Successful
2023-08-29T00:33:45

FUTURE SCOPE:

1. **Enhanced Financial Services:** The application can be extended to include more sophisticated financial services such as investment management, loan processing, and financial planning tools. This will provide users with comprehensive banking solutions under one platform.
2. **Mobile Banking App:** Develop a dedicated mobile banking app that allows users to access their accounts, perform transactions, and manage their finances on the go. The app can incorporate biometric authentication for added security and convenience.
3. **Personalized Recommendations:** Implement machine learning algorithms to analyze user transaction data and offer personalized financial recommendations, such as savings strategies, investment opportunities, and budgeting advice.
4. **Integration with Third-party Services:** Integrate the application with external services like payment gateways, e-commerce platforms, and digital wallets to provide users with a seamless experience for online shopping and payments.
5. **Data Analytics and Reporting:** Incorporate data analytics capabilities to generate insightful reports and visualizations for users. This can include expense trends, income analysis, and financial goal tracking.
6. **Security Enhancements:** Implement advanced security measures such as two-factor authentication, biometric identification, and real-time fraud detection algorithms to safeguard user accounts and transactions.

References

1. "MySQL :: MySQL 8.0 Reference Manual." MySQL Documentation. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/.
2. "Java™ Platform, Standard Edition 8 API Specification." Oracle Help Center. [Online]. Available: https://docs.oracle.com/javase/8/docs/api/.
3. "HTML Living Standard." WHATWG. [Online]. Available: https://html.spec.whatwg.org/multipage/.
4. "CSS3 Specification." World Wide Web Consortium (W3C). [Online]. Available: https://www.w3.org/TR/css-2021/.
5. "Bootstrap." Bootstrap. [Online]. Available: https://getbootstrap.com/docs/5.0/getting-started/introduction/.
6. "Font Awesome Icons." Font Awesome. [Online]. Available: https://fontawesome.com/icons.
7. Youtube.com