Software Requirements Specification (SRS) - Bankezy

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. **Bank Employees:** This user class comprises authorized bank personnel, including administrators and customer service representatives. These employees hold distinct

roles that grant them access to specific functionalities for managing user accounts, transactions, and system operations.

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.

3.5 Employee Access and Management

Authorized bank employees with defined roles can access and manage user accounts and transactions with precision. Administrative employees possess enhanced privileges, enabling them to verify accounts, review transactions, and manage user data, ensuring the integrity and efficiency of system operations.

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.

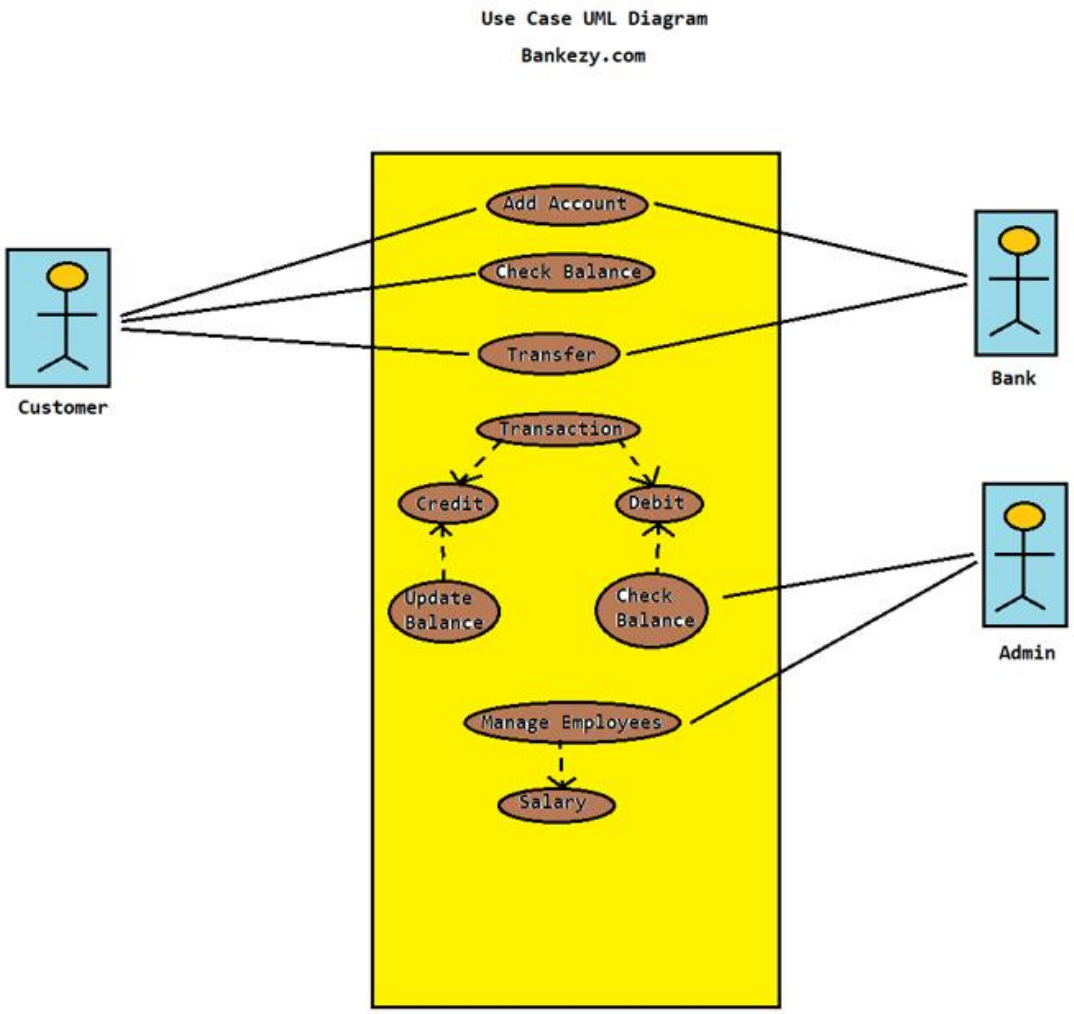
- **Requirement:** Role-based access control (RBAC) mechanisms must be enforced rigorously to restrict unauthorized access to sensitive data and functionalities.

6.3 Reliability

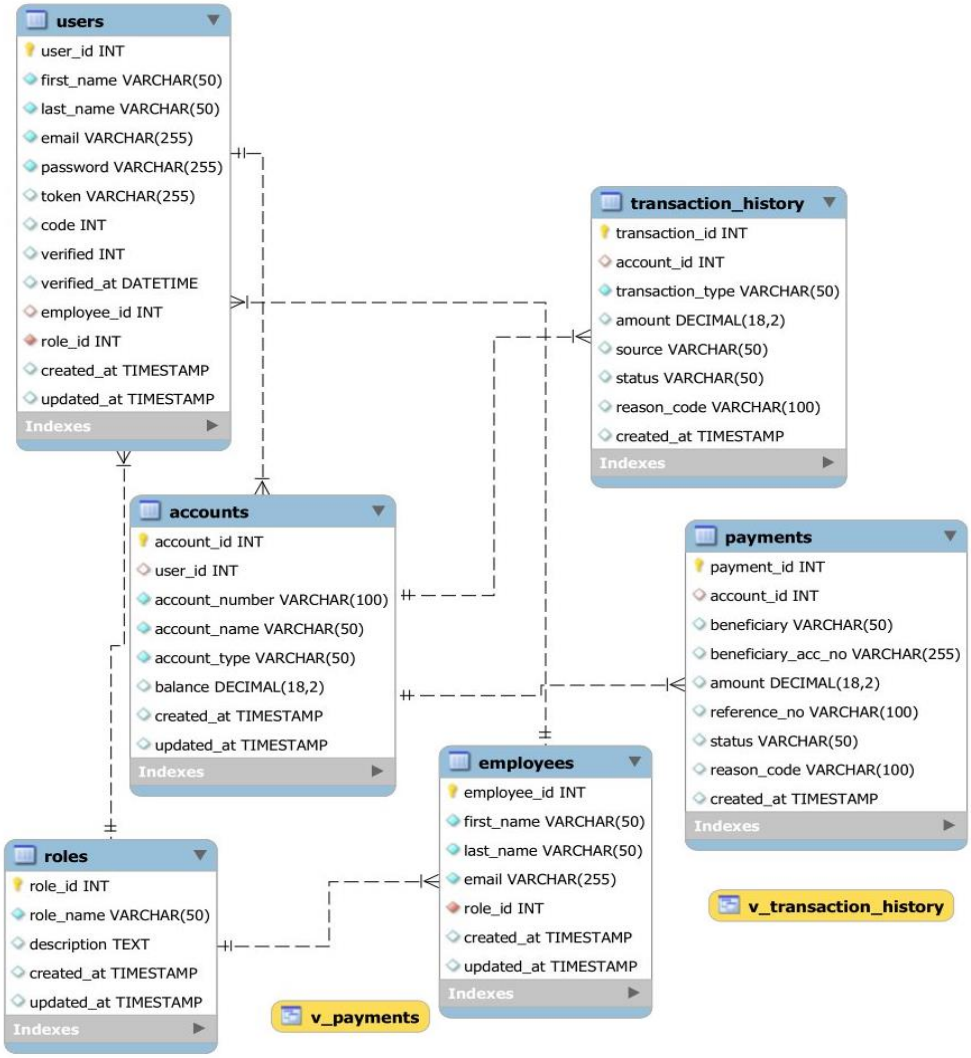
- **Requirement:** The system should be operational 24/7, guaranteeing consistent availability to users with minimal scheduled downtime for necessary maintenance tasks.
- **Requirement:** Comprehensive data backup mechanisms and disaster recovery protocols must be established to ensure data integrity and swift recovery in the event of unforeseen issues.

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

8.2 References

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