

EUSBC Software Requirements Specification (SRS)

Version: 1.0

Date: 11/01/2026

Author: Egbie Uku



1. Introduction.....	5
1.1 Purpose.....	5
1.2 System Overview.....	5
1.3 Intended Audience.....	5
2. Overall Description.....	6
2.1 Product Overview.....	6
2.2 Simulation Scope.....	6
2.3 Target Users.....	6
3. Core Features and Functional Requirements.....	7
3.1 Account Creation and Management.....	7
3.2 Wallet and Bank Account Management.....	7
3.3 Interest Management.....	7
3.3.1 Interest Configuration.....	8
3.3.2 Interest Application.....	8
3.3.3 Default Behaviour.....	8
3.4 Administrative Configuration.....	8
3.5 Funds and Transaction Management.....	8
3.5.1 Transaction Authentication.....	8
3.5.2 Add Funds.....	8
3.5.3 Transfers.....	9
3.5.4 Recurring Transfers.....	9
3.5.5 Transaction Limits.....	9
3.6 Dashboard, Security, and Notifications.....	9
3.7 Credit Card Management.....	9
4. Non-Functional Requirements.....	10
4.1 Security.....	10
4.2 Performance.....	10
4.3 Reliability and Availability.....	10
4.4 Scalability.....	11

4.5 Maintainability and Configurability.....	11
4.6 Auditability and Compliance.....	11
5. Out of Scope.....	11
5.1 Account Management.....	11
5.2 Wallet and Bank Account Management.....	12
5.3 Funds and Access Control.....	12
6.1 Bank Account Management.....	12
6.2 Funds Management.....	13
6.3 Reporting and Analytics.....	13
7.1 Assumptions.....	13
7.2 Constraints.....	14

1. Introduction

1.1 Purpose

This document defines the functional and non-functional requirements for a secure, web-based financial management system. The purpose of this Software Requirements Specification (SRS) is to provide a clear, structured description of the system's expected behaviour, constraints, and capabilities, serving as a reference for design, development, testing, and validation activities.

1.2 System Overview

The application is a web-based platform that enables users to securely manage accounts, wallets, linked bank accounts, and financial transactions. Users can add and transfer funds, manage stored payment methods, configure transaction limits, schedule recurring transfers, and accrue interest based on configurable, bank-specific rules.

In addition to end-user functionality, the system provides administrative capabilities for managing users, configuring system-wide settings, monitoring activity, and generating operational and audit reports.

1.3 Intended Audience

This document is intended for the following audiences:

- Software developers
- Testers and quality assurance engineers
- System architects
- Security engineers
- Technical reviewers and maintainers
- Independent learners and hobbyists seeking to understand the system design

2. Overall Description

2.1 Product Overview

EUSBC is a virtual banking platform that enables users to create and manage virtual bank accounts, wallets, and payment cards within a simulated financial environment. Users can fund virtual accounts and wallets, transfer funds between supported accounts on the platform, manage linked cards, disable card usage, and delete cards as needed.

The application supports internal fund transfers between active bank accounts and wallets registered within the platform. All financial entities and balances within the system are virtual and do not represent real currency or real-world financial transactions.

2.2 Simulation Scope

EUSBC is a simulated banking system intended for educational, experimental, and demonstrative purposes only. The platform does not integrate with real financial institutions, payment processors, or external banking networks. All balances, transactions, and interest calculations are virtual and have no real-world monetary value.

2.3 Target Users

The application is intended for:

- Individuals interested in exploring or learning about digital banking concepts
- Developers and testers evaluating virtual financial workflows
- Technical users interacting with a simulated banking environment

Users are expected to have basic computer literacy and proficiency in the English language.

3. Core Features and Functional Requirements

This section defines the core functional requirements (FR) of the system. Each requirement describes mandatory system behaviour.

3.1 Account Creation and Management

FR-01 The application shall allow users to create an account.

FR-02 The application shall allow users to authenticate securely using supported authentication mechanisms.

FR-03 The application shall allow users to log in and log out of the application.

FR-04 The application shall allow users to verify their email address.

FR-05 The application shall provide a configuration option to bypass email verification for testing environments.

FR-06 The application shall allow users to reset their passwords.

FR-07 The application shall allow users to update their profile information.

FR-08 The application shall allow users to select a preferred currency for displaying balances in wallets and bank accounts.

FR-09 The application shall provide access to two-factor authentication (2FA) recovery code management.

3.2 Wallet and Bank Account Management

FR-10 The application shall automatically create a single wallet for each user upon account creation.

FR-11 The application shall allow users to create a bank account by selecting from a predefined list of supported banking institutions.

FR-12 The application shall allow users to add up to three credit cards to their wallet.

FR-13 The application shall prevent users from adding duplicate credit cards to the same wallet.

FR-14 The application shall prevent users from removing a credit card that contains a non-zero balance.

FR-15 The application shall allow users to schedule recurring transfers between wallets, bank accounts, and credit cards.

3.3 Interest Management

3.3.1 Interest Configuration

FR-16 The application shall support configurable interest settings on a per-bank basis.

FR-17 The application shall support enabling or disabling interest calculations per bank.

FR-18 The application shall support configurable interest rates per bank.

FR-19 The application shall support configurable interest compounding frequencies per bank.

3.3.2 Interest Application

FR-20 The application shall calculate and apply interest only when interest is enabled for the associated bank.

FR-21 The application shall calculate interest using the configured interest rate and compounding frequency for the associated bank.

3.3.3 Default Behaviour

FR-22 If no interest configuration exists for a bank, the system shall apply no interest by default.

FR-23 The application shall support a global default interest configuration as a fallback for banks without explicit settings.

3.4 Administrative Configuration

FR-24 The application shall allow administrators to update interest configuration settings.

FR-25 Changes to interest configuration shall take effect without requiring code changes or application redeployment.

3.5 Funds and Transaction Management

3.5.1 Transaction Authentication

FR-26 The application shall require users to enter a transaction PIN for all financial operations, including adding and transferring funds.

3.5.2 Add Funds

FR-27 The application shall allow users to add funds to their wallet.

FR-28 The application shall allow users to add funds to their bank account.

FR-29 The application shall allow users to add funds to their credit cards.

3.5.3 Transfers

FR-30 The application shall allow users to transfer funds from their wallet to their bank account.

FR-31 The application shall allow users to transfer funds from their wallet to any owned credit card.

FR-32 The application shall allow users to transfer funds from their wallet to another user's wallet.

FR-33 The application shall allow users to transfer funds from their bank account to another user's bank account.

3.5.4 Recurring Transfers

FR-34 The application shall allow users to schedule recurring transfers between wallets, bank accounts, and credit cards.

3.5.5 Transaction Limits

FR-35 The application shall allow users to configure transaction limits on a daily basis.

FR-36 The application shall allow users to configure transaction limits on a weekly basis.

FR-37 The application shall allow users to configure transaction limits on a monthly basis.

3.6 Dashboard, Security, and Notifications

FR-38 The application shall display a dashboard summarising the user's financial data.

FR-39 The application shall ensure that users can only access their own financial data.

FR-40 The application shall provide access to a dashboard for managing 2FA recovery codes.

FR-41 The application shall provide real-time transaction notifications via email or push notifications.

3.7 Credit Card Management

FR-42 The application shall allow users to manually add credit cards using required card details.

FR-43 The application shall allow users to block or unblock credit cards.

FR-44 The application shall allow users to fund individual credit cards or all owned credit cards.

4. Non-Functional Requirements

4.1 Security

- **NFR-01** The application shall require PIN verification for all financial transactions.
- **NFR-02** The application shall enforce authentication for user login and sensitive account actions.
- **NFR-03** The application shall support authentication using two-factor authentication (2FA) recovery codes.
- **NFR-04** The application shall encrypt sensitive data both at rest and in transit.
- **NFR-05** The application shall prevent unauthorised access to wallets, bank accounts, and transaction data.
- **NFR-06** The application shall log all authentication attempts and financial transactions for audit purposes.

4.2 Performance

- **NFR-07** The application shall process standard fund transfers within an acceptable response time under normal operating conditions.
- **NFR-08** The application shall validate transaction limits before committing any financial transaction.
- **NFR-09** The application shall prevent race conditions that could result in inconsistent or negative balances.

4.3 Reliability and Availability

- **NFR-10** The application shall ensure transactional consistency for all balance updates.
- **NFR-11** The application shall not partially complete a financial transaction.
- **NFR-12** The application shall gracefully handle system failures and provide meaningful error messages to users.

4.4 Scalability

- **NFR-13** The application shall support multiple concurrent users performing financial transactions.
- **NFR-14** The application shall allow transaction processing functionality to scale independently of user management functionality.

4.5 Maintainability and Configurability

- **NFR-15** The application shall support configuration flags for enabling or disabling system features.
- **NFR-16** The application shall allow transaction limits and interest rules to be modified without requiring code changes.
- **NFR-17** The application shall be designed using modular components to support future enhancements.

4.6 Auditability and Compliance

- **NFR-18** The application shall record all financial transactions with timestamps and unique identifiers.
- **NFR-19** The application shall retain transaction history for reporting and auditing purposes.
- **NFR-20** The application shall provide administrative access to transaction logs and reports.

5. Out of Scope

This section defines functionality that is **explicitly excluded** from the initial release of the system.

5.1 Account Management

- The application shall not support joint or shared accounts.
- The application shall not allow users to create multiple wallets or multiple bank accounts.
- The application shall not support automatic transaction imports.
- The application shall not allow users to download financial data in PDF, CSV, or similar file formats.
- The application shall not integrate with external banks or third-party financial services.
- The application shall support the English language only.

5.2 Wallet and Bank Account Management

- The application shall not create wallets or bank accounts for users who are not registered.
- The application shall not allow users to add more than three credit cards to a wallet.

5.3 Funds and Access Control

- The application shall not allow users to transfer funds to credit cards they do not own.
- The application shall not allow transfers involving another user's credit card, including:
 - transfers from another user's card to another user's bank account,
 - transfers from another user's card to another user's wallet,
 - transfers from another user's card to any other card.
- The application shall not allow unauthenticated or unauthorised users to view, update, or transfer funds.

6. Future Enhancements

This section outlines potential features and improvements that may be considered in future versions of the system. These items are **not part of the initial release**.

6.1 Bank Account Management

- Automatic transaction import from linked bank accounts.
- Different type of accounts e.g saving accounts

6.2 Reporting and Analytics

- Downloadable transaction reports in PDF, CSV, or Excel formats.
- Advanced analytics and transaction insights.
- Visual charts and trends for income and expenses over time.

6.2 Loans and overdraft

- Overdrafts
- Loans

Assumptions & Constraint

This section outlines the assumptions made during the design of the system and the constraints that limit or influence its implementation

7.1 Assumptions

The following assumptions apply to the initial version of the application:

- Users have access to a stable internet connection for operations that require external communication, such as email verification and notification delivery.

- Users register using a valid and unique email address, except in testing environments where verification may be bypassed.
- Users are responsible for maintaining the confidentiality of their authentication credentials.
- Each user owns exactly one wallet and one bank account within the system.
- Each credit card added to the system is owned by the authenticated user.
- Users are authenticated before performing any financial operations.
- All monetary values within the system are represented using a single currency.
- The application operates in a controlled environment where all financial transactions are simulated and not connected to real-world banking systems.

7.2 Constraints

The application is subject to the following constraints:

- The system supports a maximum of one wallet and one bank account per user.
- A maximum of three credit cards may be linked to a user's wallet.
- The system does not integrate with external banks or third-party financial services in the initial release.
- The system supports the English language only.
- Financial operations may be performed only by authenticated and authorised users.
- The initial release targets a web-based platform exclusively.

Functional Requirements (Reference Table)

The table below summarises the functional requirements for easy reference, including **unique IDs, priorities, and test case references**

Req ID	Category	Functional Requirement	Priority	Notes	Test case references
FR-01	User Management	The application shall allow users to create an account.	High	Core functionality	TC-01
FR-02	User Management	The application shall allow users to authenticate securely.	High	Must enforce password/security standards.	TC-02
FR-03	User Management	The application shall allow users to verify their email.	High	Verification required unless bypassed.	TC-03
FR-04	User Management	The application shall allow the user to bypass the email verification by proving a configuration flag in the settings	Medium	Allow the email verification to be bypassed when the flag is set to True in the settings. This for testing and internal purpose only	TC-04
FR-05	User Management	The application shall automatically create a wallet user	High	Wallet linked to the user	TC-05

		registration			
FR-06	Wallet	The application shall allow the user to select a single bank from a set of bank which will then automatically create a bank account upon registration	High	Bank account link to the user	TC-06
FR-07	Bank	The application shall prevent users from adding the same credit card multiple times to the wallet	High	Duplication prevention	TC-07
FR-08	Wallet	The application shall prevent users from removing any card that contains funds.	High	Safety mechanism to avoid fund loss.	TC-08
FR-09	Wallet/Bank/Card	The application shall allow users to add funds to their wallet, bank account and card.	High	Deposition	TC-09
FR-10	Transactions	The application shall allow users to	High	Peer-to-peer transactions.Their	TC-10

		do peer-to-peer transactions i.e. from their wallet to another user wallet		wallet to another wallet transfer	
FR-11	Transactions	The application shall allow users to transfer funds from their wallet to any credit card that belongs to them.	High	Card transactions	TC-11
FR-12	Transactions	The application shall allow users to transfer funds from their bank account to another user's bank account	High	Peer-to-peer transactions. Their bank account to another bank account transfer	TC-12
FR-13	Account Management	The application shall allow users to view and update the balances in their bank accounts, wallets, and credit cards.	Medium	Balance and update management.	TC-13
FR-14	Dashboard	The application shall display a dashboard summarising the user's financial data.	High	Dashboard to show wallet, bank, and card balances.	TC-14

FR-15	Security	The application shall ensure that users can only access their own financial data.	High	Security and privacy requirement	TC-15
--------------	-----------------	---	------	----------------------------------	-------

Non Functional Requirements (Reference Table)

The table below summarises the functional requirements for easy reference, including **unique IDs, priorities, and test case references**

Req ID	Category	Non Functional Requirement	Priority	Notes	Test case references
NFR-01	Security	The application shall require PIN verification for all financial transactions.	High	Applies to wallet, bank, and card transactions	NTC-01
NFR-02	Security	The application shall enforce authentication for user login and sensitive account actions.	High	Includes login, profile updates, and transactions.	NTC-02
NFR-03	Security	The application shall support authentication using two-factor authentication (2FA) recovery codes.	Medium	Backup access mechanism	NTC-03
NFR-04	Security	The application shall encrypt sensitive data both at rest and in transit.	High	Industry-standard encryption required and encrypted database	NTC-04
NFR-05	Security	The application shall prevent unauthorised	High	Access control enforcement	NTC-05

		access to wallets, bank accounts, and transaction data.			
NFR-06	Security	The application shall log all authentication attempts and financial transactions for audit purposes.	High	Required for compliance and monitoring.	NTC-06
NFR-07	Security	The application shall prevent users from adding the same credit card multiple times to the wallet	High	Duplication prevention	NTC-07
NFR-08	Performance	The application shall process standard fund transfers within an acceptable response time under normal operating conditions.	High	Performance SLA dependent.	NTC-08
NFR-09	Performance	The application shall validate transaction limits before committing any financial transaction.	High	Prevents overdraft and abuse.	NTC-09
NFR-10	Performance	The application shall prevent race conditions that could result in inconsistent	High	Requires transactional locking or atomic operations.	NTC-10

		or negative balances.			
NFR-11	Reliability	The application shall ensure transactional consistency for all balance updates.	High	ACID-compliant transactions.	NTC-11
NFR-12	Reliability	The application shall not partially complete a financial transaction.	High	Rollback required on failure	NTC-12
NFR-13	Reliability	The application shall gracefully handle system failures and provide meaningful error messages to users.	Medium	User experience requirement	NTC-13
NFR-14	Scalability	The application shall support multiple concurrent users performing financial transactions.	High	Concurrent transaction handling required.	NFC-14
NFR-15	Maintainability	The application shall allow transaction processing functionality to scale independently of user	Medium	Supports horizontal scaling.	NFC-15

		management functionality.			
NFR-16	Maintainability	The application shall be designed using modular components to support future enhancements.	Medium	Configuration-driven behaviour.	NFC-16
NFR-17	Maintainability	The application shall be designed using modular components to support future enhancements.	Medium	Improves extensibility and maintenance.	NFC-17
NFR-18	Auditability	The application shall record all financial transactions with timestamps and unique identifiers.	High	Required for audit trails.	NFC-18
NFR-19	Auditability	The application shall retain transaction history for reporting and auditing purposes.	High	Retention policy applies.	NFC-19
NFR-20	Auditability	The application shall provide administrative access to transaction logs and reports.	Medium	Restricted admin-only access.	NFC-20

Glossary

This section defines key terms and concepts used throughout this document to ensure clarity and consistency.

Authentication

The process of verifying a user's identity using approved credentials or authentication mechanisms.

Balance

The current amount of funds available in a wallet, bank account, or credit card.

Bank Account

A system-managed virtual account associated with a user for storing and transferring funds internally within the platform.

Configuration Flag

A configurable setting used to enable, disable, or modify specific system behaviour without requiring code changes.

Credit Card

A virtual payment card added by a user to their wallet for funding and transfer purposes within the system.

Dashboard

A user interface that provides a consolidated view of a user's financial data, including balances and recent activity.

Email Verification

A process used to confirm ownership of a user-provided email address before granting full system access.

Funds

Virtual monetary value stored within wallets, bank accounts, or credit cards in the system.

Functional Requirement

A statement that defines a specific behaviour or capability the system must provide.

Future Enhancements

Planned or potential features that are not part of the initial release but may be implemented in later versions.

Interest

A calculated increase applied to eligible account balances based on configured rates and compounding rules.

Non-Functional Requirement (NFR)

A requirement that defines a quality attribute of the system, such as security, performance, availability, or scalability.

Out of Scope

Features, behaviours, or integrations that are explicitly excluded from the initial version of the system.

Peer-to-Peer (P2P) Transfer

The transfer of funds between two different users within the system.

Transfer

The movement of funds between wallets, bank accounts, or credit cards within the system.

User

An individual who has registered an account and is authorised to interact with the system

Wallet

A virtual account automatically created for each user to store funds and manage linked credit cards.

