**Software Requirements**

**Specification**

**for**

**VAULTY**

**Version 1.0 approved**

**Prepared by**

**Mohsin Javeed 271045510**

**Aden Ali Agha 271045522**

**Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.**

**Software Requirements Specification for Vaulty Page ii**

# Table of Contents

[Table of Contents ii](#_Toc8033)

[Revision History ii](#_Toc8034)

[1. Introduction 1](#_Toc8035)

[1.1 Purpose 1](#_Toc8036)

[1.2 Document Conventions 1](#_Toc8037)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc8038)

[1.4 Product Scope 1](#_Toc8039)

[1.5 References 1](#_Toc8040)

[2. Overall Description 2](#_Toc8041)

[2.1 Product Perspective 2](#_Toc8042)

[2.2 Product Functions 2](#_Toc8043)

[2.3 User Classes and Characteristics 2](#_Toc8044)

[2.4 Operating Environment 2](#_Toc8045)

[2.5 Design and Implementation Constraints 2](#_Toc8046)

[2.6 User Documentation 2](#_Toc8047)

[2.7 Assumptions and Dependencies 3](#_Toc8048)

[3. External Interface Requirements 3](#_Toc8049)

[3.1 User Interfaces 3](#_Toc8050)

[3.2 Hardware Interfaces 3](#_Toc8051)

[3.3 Software Interfaces 3](#_Toc8052)

[3.4 Communications Interfaces 3](#_Toc8053)

[4. System Features 4](#_Toc8054)

[4.1 System Feature 1 4](#_Toc8055)

[4.2 System Feature 2 (and so on) 4](#_Toc8056)

[5. Other Nonfunctional Requirements 4](#_Toc8057)

[5.1 Performance Requirements 4](#_Toc8058)

[5.2 Safety Requirements 5](#_Toc8059)

[5.3 Security Requirements 5](#_Toc8060)

[5.4 Software Quality Attributes 5](#_Toc8061)

[5.5 Business Rules 5](#_Toc8062)

[6. Other Requirements 5](#_Toc8063)

[Appendix A: Glossary 5](#_Toc8064)

[Appendix B: Analysis Models 5](#_Toc8065)

[Appendix C: To Be Determined List 6](#_Toc8066)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

## *This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for the Vaulty.*

## *The purpose of this document is to define the features and scope of the system for use by both developers and stakeholders. This SRS covers the entire system and includes all major components, such as user registration, balance tracking, savings goals, and transaction history.*

## 1.2 Document Conventions

## *This document follows a clear and consistent format for readability and structure. The main text is written in Arial Italic, while section headings are in Times New Roman Bold to distinguish them clearly.*

## *All requirements are listed using unique identifiers (e.g., REQ-1, REQ-2) and are stated clearly to ensure they are specific and unambiguous.*

## *Where applicable, the priority of each requirement (High, Medium, Low) is stated individually. No assumptions are made regarding inheritance of priority from higher-level requirements*

## *.*1.3 Intended Audience and Reading Suggestions

## *This document is intended for developers, testers, supervisors, and potential users—mainly students and young adults interested in saving money.*

## *Readers should start with the introduction and overall description for context, then move to system features and technical sections based on their role.*

## 1.4 Product Scope

## *This software is a digital piggy bank designed to help users especially students and young adults develop better saving habits. It addresses the problem of manual, inconsistent tracking by offering a modern, user-friendly platform to monitor and manage savings efficiently. The goal is to promote financial responsibility through convenience and accessibility.*

## 1.5 References

This Document does not refer to any other documents.

# 2. Overall Description

## 2.1 Product Perspective

## *This is a new, self-contained product developed as a standalone application for academic purposes. It is not a follow-up to any existing product or system, nor does it replace any prior digital solution. Its purpose is to digitally simulate the traditional concept of a piggy bank with added tracking and ease-of-use features.*

## 2.2 Product Functions

## *Vaulty lets users create personal accounts to manage savings efficiently. Users can set savings goals, track progress with charts, and categorize deposits. It supports money transfers, including recurring transfers, and offers reminders to encourage saving. The app features expenditure tracking and a points system to reward saving milestones. Users can also play mini games to earn bonus rewards, making saving fun. Secure administrative controls ensure reliable and safe platform access.*

## 2.3 User Classes and Characteristics

## *The expected user classes for the Vaulty are:*

## *All Individuals Trying to Save Money: This broad group includes users of varying ages and technical skills who want a simple way to manage savings.*

## *Adolescents and Financially Conscious Users: Teenagers and others focused on improving saving habits, frequently using features like goal setting and expenditure tracking.*

## *Young Adults and Students: Regular users managing budgets and seeking features like recurring transfers and rewards, with moderate to advanced technical skills.*

## *Users of All Ages, Mainly Young Adults: While open to all, the app primarily targets young adults engaged in financial planning.*

## *The most important user classes are adolescents and young adults, as they drive the app’s core functionality, while the others represent a wider but less targeted audience.*

## 2.4 Operating Environment

## *The software will run on common devices like smartphones and PCs, supporting Android, iOS, Windows, and macOS. It will work as a mobile app or web app in standard browsers. No special hardware or software is required.*

## 2.5 Design and Implementation Constraints

## *Security: The app must implement strong encryption, secure authentication, and protect user data to ensure privacy and prevent unauthorized access.*

## *Usability: The design should balance security features with an intuitive, user-friendly interface, especially for younger users.*

## *Maintenance: The code must follow established standards and conventions to allow easy updates and support by the client’s team.*

## *Integrations: The system will depend on external banking and payment APIs, which may limit functionality and require compliance with their protocols.*

## 2.6 User Documentation

## *The following user documentation components will be delivered along with the Vaulty application to ensure smooth onboarding, usage, and troubleshooting for users:*

## *User Manual: A detailed PDF guide covering all major features of Vaulty, including registration, money transfers, savings goals, and reward systems. It will also contain FAQs and common issue resolutions.*

## *Quick Start Guide: A short, beginner-friendly visual guide (PDF or in-app onboarding slides) to help new users get started with essential functions in under 5 minutes.*

## *In-App Tooltips and Help Bubbles: Contextual help built into the app interface to assist users as they navigate through different sections (e.g., “What’s a savings goal?”).*

## *Tutorial Videos: Short animated videos embedded in the app and accessible via the help center, guiding users through key features like setting goals and earning points.*

## *All documentation will be delivered in English and designed for mobile and desktop readability. The materials will follow standard formatting guidelines to ensure accessibility and consistency across platforms.*

## 2.7 Assumptions and Dependencies

# *It is assumed that users of all ages, especially young adults and adolescents, will have access to compatible devices and basic digital skills.*

# *The app is assumed to integrate with external banking and payment APIs, which may change or limit functionality.*

# *The project depends on standard mobile platforms (Android, iOS) and web browsers being available and up to date.*

# *.*3. External Interface Requirements

## 3.1 User Interfaces

## *Vaulty features a clean, user-friendly interface designed for both mobile and web platforms. Key screens include:*

## *Home: Displays balance, savings progress, and quick actions.*

## *Login/Signup: Simple form with password recovery.*

## *Dashboard: Shows goals, transactions, and reward points.*

## *Goals & Tracking: Allows users to set and monitor savings goals and expenses.*

## *Rewards/Games: Fun mini games to earn bonus points.*

## *All screens follow consistent layouts with standard buttons (Save, Cancel, Help), icons, and responsive design*

## 3.2 Hardware Interfaces

## *Vaulty will run on standard smartphones, tablets, and personal computers. It supports Android and iOS mobile devices, as well as Windows and macOS desktops through web browsers. No specialized hardware is required.*

## *The app interacts with device hardware (e.g., storage, internet connectivity) using standard OS-level APIs. All data transmission will use secure encrypted protocols (e.g., HTTPS) for communication with remote servers. Basic hardware like touch screen, keyboard, or mouse will be used for user interaction.*

## 3.3 Software Interfaces

## *Vaulty will interact with common software components, including operating systems, databases, and external APIs. It is expected to work with standard mobile and web platforms (Android, iOS, Windows, macOS) and connect to a cloud-based database for storing user data. The system may integrate with third-party payment APIs to simulate money transfers and savings.*

## 3.4 Communications Interfaces

# *Vaulty is assumed to use standard internet communication (e.g., HTTPS) for secure data exchange. It may simulate email notifications and basic server interactions. All communication is assumed to be secure and reliable, with no special protocols required.*

# 4. System Features

## *Vaulty allows users to register, log in, and manage money transfers, including recurring ones. Users can set savings goals, track progress, and unlock rewards. The app features a points system that rewards saving habits and offers mini-games to earn bonus points. Daily and weekly financial summaries help users track their spending and savings.*

## 4.1 Money Transfer :

#### 4.1.1 Description and Priority

The Money Transfer feature allows users to deposit, withdraw, and schedule recurring transfers between their Vaulty account and external accounts. This feature is of High priority as it is essential for the core functionality of the app.

#### 4.1.2 Stimulus/Response Sequences

User Action: User selects "Transfer Money" and enters amount and recipient details.

System Response: System verifies the details and prompts for confirmation.

User Action: User confirms transfer.

System Response: System processes the transfer and updates the balance, sending a success notification.

User Action: User cancels the transfer.

System Response: System cancels the transaction and returns to the previous screen.

4.1.3 Functional Requirements

**REQ**-1: The system must allow users to input an amount and recipient for money transfers.

**REQ**-2: The system must validate the transaction details (sufficient funds, valid recipient).

**REQ**-3: The system must support recurring scheduled transfers at user-defined intervals.

**REQ**-4: If an invalid amount or recipient is entered, the system should display an error message and prompt the user to correct it.

**REQ**-5: The system must send a confirmation notification upon successful transaction completion.

## 

## 4.2 Savings Goal:

4.2.1 Description And Priority

The Savings Goals feature enables users to set personal savings targets and track their progress over time. Users can choose a goal amount, define the timeframe, and receive rewards or unlock themes upon completion. This feature is of High priority as it is a core motivational tool to encourage savings.

4.2.2 Stimulus/Response Sequences

User Action: User selects "Create Savings Goal" and enters the goal amount and target date.

System Response: System validates the input (ensuring target date is in the future) and displays the goal overview.

User Action: User confirms the goal creation.

System Response: System creates the goal and adds it to the user's profile with a progress tracker.

User Action: User makes a deposit towards the goal.

System Response: System updates the progress bar and adjusts the remaining balance needed to reach the goal.

User Action: User completes the goal.

System Response: System sends a congratulatory notification and unlocks any associated rewards or app themes.

4.2.3 Functional Requirements

REQ-1: The system must allow users to create a savings goal by entering an amount and target date.

REQ-2: The system must validate that the target date is in the future and that the goal amount is a positive number.

REQ-3: The system must display the user's savings goal with a progress bar that updates as deposits are made.

**4.3 Gamification (Mini Games & Points System)**

4.3.1 Description and Priority

The Gamification feature includes a points system, and mini games designed to encourage users to save money regularly. The points system rewards users for consistent savings, while mini games offer additional opportunities to earn bonus points. These elements aim to make saving money more engaging and rewarding. This feature is of High priority as it drives user engagement and motivates saving behavior.

4.3.2 Stimulus/Response Sequences

User Action: User deposits money into their savings account.

System Response: System awards points based on the deposit amount and updates the user’s points balance.

User Action: User plays a mini-game.

System Response: System tracks gameplay progress and awards bonus points based on performance.

User Action: User completes a saving milestone or a mini-game.

System Response: System sends a notification congratulating the user and updates their points balance.

User Action: User attempts to redeem points for rewards.

System Response: System checks if the user has enough points and, if so, processes the reward redemption.

4.3.3 Functional Requirements

REQ-1: The system must award points for deposits made by the user.

REQ-2: The system must track and display the user's total points balance.

REQ-4: The system must notify users when they earn points from saving or playing mini-games.

REQ-5:The system must provide a variety of mini-games to keep the experience fresh and engaging.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

## *Vaulty should provide a seamless and responsive user experience. The following performance requirements are outlined:*

## *Response Time: The app should process user actions (e.g., money transfers, goal creation) within 2 seconds under normal network conditions.*

## *Transactions: Money transfers should be completed in under 5 seconds, with the system providing immediate feedback to the user upon success or failure.*

## *Goal Updates: Savings goals should update in real-time as users make deposits or changes.*

## *Mini-games: Mini-games should load within 3 seconds and should maintain smooth gameplay at 30 FPS or higher*

## *.*5.2 Safety Requirements

## *As Vaulty is a financial app, safety focuses on data protection and preventing misuse rather than harm. The app must ensure:*

## *User data (including savings, goals, and personal information) is securely stored and cannot be accessed without proper authentication.*

## *All transactions must be logged to allow tracking in case of errors or unauthorized activity.*

## *The system should prevent unauthorized access by enforcing secure login methods.*

## *The app must comply with general data protection practices and avoid actions that could result in data loss or corruption.*

## 5.3 Security Requirements

## *Vaulty must ensure that all user data and transactions are secure and private. Key requirements include:*

## *User Authentication: Secure login using email and password, with optional two-factor authentication.*

## *Data Protection: All sensitive data must be encrypted during transmission and storage.*

## *Access Control: Users can only acsess their own data; admin access is restricted and monitored.*

## *Session Management: Auto logout after a period of inactivity to prevent unauthorized access.*

## 5.4 Software Quality Attributes

Vaulty should be easy to use, reliable, and maintainable. Usability is a top priority, especially for young users. The app should work smoothly across devices (portability) and handle errors without crashing (robustness). It should also be easy to update and test.

## 5.5 Business Rules

# *Only registered users can access personal savings features. Admins can manage system data but cannot access user savings details. Users must be logged in to perform any financial actions. The app should guide users with simple prompts and restrict unauthorized actions.*

# 6. Other Requirements

# *Vaulty has minimal external constraints. It should use a simple database to store user data securely. The app will be in English only. Code should be organized for easy understanding.*

# Appendix A: Glossary

# *SRS (Software Requirements Specification): A document that outlines the functionalities, constraints, and features of a software system.*

# *Vaulty: The name of the digital piggy bank application being developed.*

# *User: Any individual interacting with the Vaulty app, primarily students and young adults saving money.*

# *Admin: A system-level user with restricted access to manage app settings and monitor user activity without viewing personal savings data.*

# *Recurring Transfer: An automated financial transaction that occurs at user-defined intervals.*

# *Savings Goal: A user-defined target to save a specific amount of money within a timeframe.*

# *Points System: A gamified feature that rewards users with points for saving habits and in-app activity.*

# *Mini-Games: Simple interactive games within the app designed to encourage saving by rewarding users with bonus points.*

# *2FA (Two-Factor Authentication): An extra layer of security requiring users to verify identity using a second method beyond a password.*

# *HTTPS (HyperText Transfer Protocol Secure): A secure version of HTTP used for encrypted communication over the internet.*

# *UI (User Interface): The visual layout and interactive elements users engage with in the app.*

# *API (Application Programming Interface): A set of protocols used for building and integrating application software, especially for external services like banking.*

# Appendix B: Analysis Models

Attached With The File Separately

# Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>