**Project Title: FinWise Manager Software Requirements Specifications**

**Date: 02/22/2024**

**Version: 1.0**

**Prepared By: Alex Nikirk , Albert Mitchell**

**Group Name: SmartSpend**

**Group# 27**

[1. Introduction 2](#_Toc837804147)

[1.1 Purpose 2](#_Toc2071147539)

[1.2 Scope 2](#_Toc2091113388)

[1.3 Definitions, Acronyms and Abbreviations 3](#_Toc436125354)

[1.4 References 3](#_Toc1359173043)

[1.5 Overview 3](#_Toc1169316327)

[2. Specific Requirements 3](#_Toc769538001)

[2.1 External Interface Requirements 3](#_Toc1012412547)

[2.1.1 User Interfaces 3](#_Toc455666837)

[2.1.2 Hardware Interfaces 4](#_Toc1172203451)

[2.1.3 Software Interfaces 4](#_Toc1934439031)

[2.1.4 Communication Protocols 4](#_Toc2079193701)

[2.2 Functional Requirements 4](#_Toc939210608)

[2.3 Non-Functional Requirements 5](#_Toc80006529)

[2.3.1 Reliability 6](#_Toc385595116)

[2.3.2 Availability 6](#_Toc961140607)

[2.3.3 Security 7](#_Toc1144188779)

[2.3.4 Maintainability 7](#_Toc580493065)

[2.3.5 Portability 9](#_Toc631770800)

[2.3.6 Performance 9](#_Toc464789058)

[2.4 Design Constraints 9](#_Toc103829234)

# **1. Introduction**

## **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to present a detailed description of the FinWise Manager, a personal finance management application. This document is intended for the development team members to guide the development process. It will also serve as an agreement on what the application is expected to do between the stakeholders and the development team. The primary goal of this document is to ensure that clear requirements are communicated and understood by all parties involved in the project.

## **1.2 Scope**

FinWise Manager is designed to empower users with tools to manage their personal finances effectively. The application will allow users to track their income, expenses, savings, and financial goals in a user-friendly interface. Features will include account registration and management, transaction logging, budget creation and monitoring, financial reporting, and insights based on user data. The application aims to support users in making informed financial decisions, encouraging better spending habits, and achieving financial stability. FinWise Manager will be accessible as a web application and may extend to mobile platforms to ensure maximum accessibility and convenience.

## **1.3 Definitions, Acronyms and Abbreviations**

SRS: Software Requirements Specification

API: Application Programming Interface

UI: User Interface

UX: User Experience

GUI: Graphic User Interface

FR: Functional Requirements

## **1.4 References**

**None required for this version**

## **1.5 Overview**

The remainder of this document is organized as follows: Section 2 details the specific requirements of the application, including external interface requirements, functional and non-functional requirements, and design constraints. This includes descriptions of the user interfaces, hardware and software interfaces, communications protocols, and detailed specifications of the application features and performance expectations.

# **2. Specific Requirements**

## **2.1 External Interface Requirements**

### 2.1.1 User Interfaces

The FinWise Manager application will feature a user-centric design prioritizing ease of navigation and efficiency. The UI will include:

* **Dashboard**: A central hub offering a quick overview of the user's financial status, including recent transactions, current balance, and budget insights, with interactive elements for a detailed view.
* **Transaction Management**: A dedicated interface for logging, viewing, and categorizing transactions, enhancing user engagement with sortable lists and filters.
* **Budgeting Tool**: Interactive tools for setting and adjusting monthly budgets across various categories, integrated with visual cues to indicate budget status.
* **Reports and Insights**: Graphical representations and analytics providing insights into spending patterns, budget compliance, and financial health over selectable periods.

### **2.1.2 Hardware Interfaces**

FinWise Manager is primarily a web-based application, there are minimal hardware interface requirements. It should be compatible with standard computing devices including desktops, laptops, tablets and smartphones with internet access

### 2.1.3 Software Interfaces

The application will interface with web and mobile platforms, to make sure that they are compatible with Web servers (Apache), Database Systems (MongoDB, MySQL), Mobile Operating Systems (support for both iOS and Android) and Third-Party API’s (financial data services for importing transactions, balance reports and email services to send the user notifications

### **2.1.4 Communication Protocols**

FinWise will use HTTP/HTTPS for secure web communication, WebSockets, for real time updates within the web and mobile applications, Mobile-specific protocols that allow the user to receive push notifications

## **2.2 Functional Requirements**

**Account Management**

**FR1.** **User Registration**: This allows the user to create a personal account by supplying vital information such as an email and password. The system will verify the email address.

**FR2. User Login/Logout**: This enables users to securely access their account using their email and password, and to sign out from their account. The system ensures secure session management.

**FR3. Profile Management**: Users can update their personal profile information, including changing their password. The system provides options for editing user details.

**FR4. Add Transactions**: Users can enter details of their income and expenses, including amount, category, and date. The system allows for the manual entry of transaction details.

**FR5. View Transactions**: This feature allows users to see the history of their transactions. Users can filter transactions by date, category, or type.

**FR6. Edit/Delete Transactions: Users can modify or remove transactions from their history.** The system provides options for editing or deleting entries.

**FR7. Set Budgets**: Users can establish budgets for various spending categories. The system facilitates the creation of budget limits.

**FR8. View Budgets**: This enables users to compare their actual spending against their set budgets. The system displays budget statuses.

**FR9. Budget Notifications**: The system alerts users when their spending approaches or exceeds their budgeted amounts through notifications.

**FR10. Spending Reports**: Users can generate reports that visualize their spending over time. The system provides graphical reports categorized by spending type.

**FR11. Financial Insights**: The system analyzes spending patterns to offer users personalized financial advice and potential savings opportunities.

**FR12. Data Encryption**: Ensures the encryption of sensitive data, including passwords and financial information, for user security.

**FR13. Two-Factor Authentication (Optional)**: Offers an additional security layer through a two-step verification process, optional for users.

**FR14. Export Data**: Allows users to download their transaction and budget data in a CSV format. The system supports data export functionality.

**FR15. Customer Support**: Provides a feature for users to contact support for help or to report issues within the application. The system includes a support mechanism.

## **2.3 Non-Functional Requirements**

* **Reliability**: This ensures the system operates reliably, accurately handling data and automatically recovering from common failures to minimize downtime.
* **Availability**: FinWise Manager will be available for use at all times, targeting a 99.9% uptime, with exceptions for scheduled maintenance.
* **Security**: The system safeguards user data using top-tier encryption techniques, complies with privacy regulations, and employs secure authentication methods to deter unauthorized access.
* **Maintainability**: Designed for straightforward updates and corrections, the system facilitates the rapid deployment of enhancements and fixes without significantly impacting user access.
* **Portability**: The application is designed to function seamlessly across various web browsers and mobile devices, ensuring users have a consistent experience regardless of platform.
* **Performance**: FinWise Manager is optimized for speed, striving for a swift response to user inputs, aiming for all operations to complete within 2 seconds under standard usage conditions.

### **2.3.1 Reliability**

The reliability requirements of FinWise Manager will ensure the system's consistent performance and resilience, focusing on error handling mechanisms and data integrity safeguards:

* **Error Handling**: The system will manage errors, providing users with clear, informative messages and logging details for system administrators. It aims to prevent application crashes and minimize user disruption.
* **Data Integrity**: FinWise Manager will implement checks and balances to maintain the accuracy and consistency of data throughout its lifecycle, using transactional processes where necessary to prevent data loss or corruption.

### **2.3.2 Availability**

Ensuring the continuous availability of FinWise Manager is paramount to providing a reliable financial management service. FinWise Manager should be available to users 24/7, with minimal planned maintenance. Unplanned downtime should be minimized and promptly addressed to ensure continuous service availability. Users should be notified in advance of any planned maintenance, and measures should be in place to minimize disruption during these periods. A robust and scalable infrastructure should support FinWise Manager's availability requirements, enabling the system to handle increased user loads without impacting performance.

### **2.3.3 Security**

The security aspects of FinWise Manager focus on the protection of user data, secure login mechanisms, and precise user permissions:

* **Data Protection**: The system employs encryption for both stored and in-transit user data, conforming to the latest security protocols to deter unauthorized access or breaches.
* **User Authentication**: For accessing their accounts, users will utilize their email and password combination. An option for enhanced security through multi-factor authentication will be available.
* **Authorization Levels**: Defined user roles and access rights ensure individuals interact only with the data and functionalities relevant to their role, safeguarding sensitive information from unwarranted access.

### **2.3.4 Maintainability**

For the FinWise Manager application, software maintenance, updates, upgrades, and testing procedures are crucial components that contribute to the overall reliability, security, and user experience.

1. Software Maintenance and Updates:

* Expectation: Routine maintenance tasks, such as bug fixes, security patches, and minor feature enhancements, are expected to be promptly addressed, ensuring minimal disruption to user access and functionality. Additionally, periodic updates aimed at improving user experience, addressing feedback, or introducing new features should be transparently communicated to users.
* Implementation: A dedicated team will continuously monitor the application for issues, ensuring that a change management process is in place to effectively handle any updates or patches. An agile development approach will allow for timely and iterative improvements based on user feedback, with automated CI/CD pipelines ensuring swift and efficient deployment.

2. Software Upgrades and Documentation:

* Expectation: Major upgrades that enhance or transform the application should occur judiciously, with users well-informed about the changes and any required actions on their part. Comprehensive and regularly updated documentation should accompany these upgrades, providing users with clear guidance and ensuring transparency throughout the process.
* Implementation: The upgrade process will be meticulously planned and executed, with a detailed project plan and thorough testing procedures to ensure a smooth transition. Version control and centralized documentation management systems will be utilized to maintain clear records and ensure that users have access to the relevant information.

3. Testing Procedures:

* Expectation: Rigorous testing procedures that include unit testing, integration testing, and user acceptance testing (UAT) are expected. Testing should be conducted across multiple devices and platforms to ensure compatibility and functionality.
* Implementation: Automated testing tools should be used to streamline the testing process. Test cases should cover a wide range of scenarios, including edge cases and user interactions. The testing environment should closely mirror the production environment to ensure accurate results.

### **2.3.5 Portability**

FinWise Manager aims to be widely accessible, supporting a variety of platforms and devices:

* **Web Browsers**: It will work smoothly on popular web browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, making it easy for users to access the application from desktops and laptops.
* **Mobile Devices**: The application is designed to be mobile-friendly, supporting both iOS and Android devices. This ensures users can easily manage their finances from their phones or tablets, with the interface adjusting to different screen sizes.
* **Operating Systems**: For any potential desktop or mobile app versions in the future, FinWise Manager plans to support the latest versions of Windows and macOS, alongside iOS and Android for mobile users, offering flexibility in how the application can be accessed.

### 2.3.6 Performance

For the performance section of FinWise Manager, the focus is on ensuring the application runs smoothly and efficiently, providing a seamless experience for all users:

* **Response Times**: The application will be designed to respond quickly, aiming for a maximum response time of 2 seconds for any action under normal conditions, ensuring users can navigate and use features without frustrating delays.
* **Simultaneous Users**: FinWise Manager will be capable of supporting a significant number of users simultaneously without degradation in performance, prepared to handle peak usage periods gracefully.

This approach ensures that the application remains reliable and responsive, contributing to a positive user experience even as the user base grows.

## **2.4 Design Constraints**

The design of FinWise Manager will adhere to specific constraints to ensure compliance and optimal performance across platforms:

* **Regulatory Compliance**: The application will comply with financial regulations and data protection laws, such as U.S. financial regulations and data privacy laws (California Consumer Privacy Act (CCPA), affecting how user data is collected, stored, and processed.
* **Cross-Platform Compatibility**: While designed to be accessible on major web browsers and mobile platforms, variations in browser engines and mobile OS versions may limit certain features or designs. Efforts will be made to ensure a consistent user experience across all supported platforms.
* **Performance Optimization**: Given the diverse range of devices, especially mobile, the application will be optimized for low resource consumption to ensure smooth operation on older devices or slower internet connections.

These constraints guide the development and design choices for FinWise Manager, ensuring it delivers a reliable and compliant service to all users.