Software Requirements Specification

for

Financial Management System (FMS)

Version 1.0 approved

Prepared by Shashank Singh, Rudrapratap Singh & Raj V. Singh

Thakur College of Engineering & Technology

Jan 22nd, 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
nil	nil	nil	nil

1. Introduction

1.1 Purpose

The product described is the Financial Management System (FMS), designed to manage and streamline financial transactions and payment processing between organizations. This SRS focuses on the core functionalities of payment tracking, invoice generation, and financial record management, aiming to improve operational efficiency and financial transparency.

1.2 Document Conventions

This SRS follows the IEEE 830-1998 standards for Software Requirements Specifications.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for different readers, including developers (who focus on technical aspects and system design), project managers (who are interested in the project's scope and timelines), marketing staff (who need to understand the system's features), testers (who check the system's functionality), and documentation writers (who create user guides). The document starts with an introduction and system overview for general understanding, then provides specific details on functionality, followed by sections on design and testing. The recommended reading order begins with the overview, then moves on to detailed requirements, design, and testing sections, depending on the reader's role.

1.4 Project Scope

The Financial Management System (FMS) is designed to streamline and automate the management of payments and financial transactions between organizations. Its primary purpose is to track payments, generate invoices, and manage financial records efficiently, reducing manual effort and minimizing errors. The system aims to enhance operational efficiency, improve financial transparency, and provide timely reporting to all stakeholders. By aligning with the goal of delivering a seamless, user-friendly financial experience, the FMS supports the broader business strategy of improving organizational management and enhancing overall satisfaction.

1.5 References

- Vision and Scope Document (v1.0, Jan 2025) – Internal repository.
- IEEE 830-1998 Standard – IEEE Xplore.
- User Interface Style Guide (v1.2, Jan 2025) – Internal repository.
- System Design Specifications (v1.0, Jan 2025) – Internal repository.

2. Overall Description

2.1 Product Perspective

This product, the Financial Management System (FMS), is a new, self-contained system designed to streamline and automate the management of financial transactions and payment processing. It is not a replacement for any existing systems but aims to address the need for efficient, accurate, and transparent management of financial records. The FMS operates independently but may be integrated with other systems in the future for broader functionality. The system is designed to improve operational efficiency, reduce manual errors, and provide timely reporting, benefiting organizations looking to enhance their financial management processes.

2.2 Product Features

The Financial Management System (FMS) includes several key features designed to streamline financial transactions. These features include payment tracking, invoice generation, financial record management, and reporting. The system allows users to efficiently process payments, generate accurate invoices, maintain financial records, and generate timely reports for administrators. These core functionalities improve financial transparency, reduce manual errors, and enhance operational efficiency.

2.3 User Classes and Characteristics

The Financial Management System (FMS) is designed for various user classes with different roles and access levels. Administrators have full access, managing system settings, user roles, and financial operations. Finance Managers handle daily financial tasks like payment tracking and report generation, with limited access to system settings. Accounting Staff have restricted access, mainly entering transactions and reviewing basic financial data. Auditors have read-only access to financial records for compliance purposes. Administrators and Finance Managers are the most critical user classes, while Accounting Staff and Auditors play supportive but essential roles with more limited access.

2.4 Operating Environment

The Financial Management System (FMS) is built for cross-platform compatibility across Windows, Linux, macOS, Android, and iOS. The system is designed to work on the latest versions of these operating systems and requires an internet connection for data synchronization.

2.5 Design and Implementation Constraints

The development of the Financial Management System (FMS) will be influenced by several factors, including corporate and regulatory policies. The system must comply with Indian government laws, such as taxation regulations (e.g., GST compliance) and other financial laws, ensuring proper handling of financial data and reporting. It will also adhere to data privacy regulations like GDPR or HIPAA where applicable. The system will be built using Flutter for cross-platform compatibility, which may limit the use of platform-specific features. Security measures, including encryption and secure authentication via Firebase, will restrict certain tool and protocol choices. Integration with external applications, such as payment gateways, will be constrained by their APIs, and the system must meet specific performance and scalability requirements. Additionally, adherence to internal design conventions and programming standards will guide the development process to ensure consistency, maintainability, and regulatory compliance.

2.6 User Documentation

nil

2.7 Assumptions and Dependencies

The Financial Management System (FMS) assumes a secure, internet-connected environment where authorized users access the system with valid credentials. It relies on third-party payment gateways, cloud-based databases, and compliance with financial regulations like GST, GDPR, and HIPAA. The system depends on Firebase Authentication for security, external APIs for payments and tax calculations, and cross-platform compatibility using Flutter. Report generation in PDF and Excel formats requires third-party libraries, while regular data backups ensure reliability.

3. System Features

3.1 Access Control and Security

3.1.1 Description and Priority

The Access Control and Security feature ensures that the application is strictly accessible only within the office environment, preventing access from unauthorized locations or devices. This feature is of High Priority as it directly impacts the security of the system by limiting access to only trusted devices and networks. The system will enforce security protocols to prevent any external or unauthorized users from interacting with the application, ensuring that all data remains secure and protected.
Priority Ratings:

- Benefit: 10
- Penalty: 1
- Cost: 5
- Risk: 7

3.1.2 Stimulus/Response Sequences

When a user attempts to access the application from outside the designated office network or on unauthorized devices, the system will deny access and display a message explaining that the system is restricted to the office environment. When the user accesses the application from an authorized device and network, they will be prompted to log in using their credentials. If the login is successful, the system grants access; otherwise, it will display an error message.

3.1.3 Functional Requirements

REQ-1: The system must verify that the device accessing the application is connected to the office’s internal network or an approved VPN.
REQ-2: The system must deny access if the user is attempting to connect from outside the defined office environment.

REQ-3: The system must ensure that no third-party access is allowed to the system from unauthorized external devices or networks.
REQ-4: The system must log any unauthorized access attempts and alert administrators for review.
REQ-5: The system must require secure authentication for all users attempting to log in and limit access to designated, pre-approved users only.

3.2 Payment Tracking

3.2.1 Description and Priority

The Payment Tracking feature is designed to record and manage payments, allowing authorized users to input, track, and update payment information. This feature is of High Priority since it ensures accuracy in financial operations and helps minimize errors by providing real-time updates and transaction details.
Priority Ratings:

- Benefit: 9
- Penalty: 3
- Cost: 4
- Risk: 5

3.2.2 Stimulus/Response Sequences

When a user inputs payment details, such as the amount, method, and recipient, the system will validate the information and process the transaction. If the payment information is valid, the transaction will be recorded and reflected in the user's account. If any required information is missing or incorrect, an error message will prompt the user to correct the details before proceeding.

3.2.3 Functional Requirements

REQ-1: The system must allow authorized users to enter payment details, including the amount, method, and recipient.
REQ-2: The system must validate the payment details to ensure all necessary fields are completed.
REQ-3: The system must store the transaction details securely and update the recipient’s balance accordingly.
REQ-4: The system must display an error message if there is missing or incorrect information in the payment details.
REQ-5: The system must provide confirmation after a payment has been successfully recorded.

3.3 Report Generation

3.3.1 Description and Priority

The Report Generation feature enables users to generate financial reports, such as payment summaries, transaction histories, and outstanding balances. This feature is of Medium Priority as it supports administrative tasks and decision-making but is not as critical to system functionality as other features. Reports will be customizable, exportable in formats like PDF and Excel, and will assist in reviewing and analyzing financial data.
Priority Ratings:

- Benefit: 7
- Penalty: 4
- Cost: 6
- Risk: 4

3.3.2 Stimulus/Response Sequences

When a user selects a report type and applies filters (e.g., date range or specific student ID), the system will process the request and generate the report. The user will be able to view the report on-screen and download it in their preferred format, such as PDF or Excel. If the data does not meet the selected filters, the system will show an error message indicating no records found.

3.3.3 Functional Requirements

REQ-1: The system must allow users to generate customizable reports by selecting specific filters like date range, payment method, or user ID.
REQ-2: The system must support exporting generated reports in PDF and Excel formats.
REQ-3: The system must display an error message when no data matches the selected report criteria.
REQ-4: The system must allow users to download the generated reports in the desired format.
REQ-5: The system must provide real-time data for report generation, ensuring up-to-date information is reflected in the report.

4. External Interface Requirements

4.1 User Interfaces

The software's user interface (UI) will be designed for simplicity and consistency, adhering to established GUI standards and the product family style guide. It will include a secure login screen with fields for username and password. The interface will feature clear navigation menus that allow users to easily access different sections, such as payment tracking and report generation. Standard buttons such as "Save," "Cancel," and "Help" will be present across all screens, ensuring uniformity and ease of use. Error messages will be displayed in a clear, concise manner, providing guidance for users to resolve issues. The screen layouts will be structured logically, presenting information in an intuitive and easily accessible format. Additionally, keyboard shortcuts will be provided for common actions to enhance efficiency. The software will be responsive across devices, optimizing the UI for both mobile and desktop screens.

4.2 Hardware Interfaces

The software will interface with a range of hardware devices, including desktop computers (Windows, Linux, macOS), mobile devices (Android, iOS), and dedicated office devices such as POS systems, using standardized communication protocols like HTTP/HTTPS for secure data transmission. The software will facilitate data exchanges, including sending and receiving information between the device’s local storage and cloud servers, ensuring real-time transaction updates and synchronization. Hardware interactions will focus on processing user inputs, displaying outputs on screens, and supporting device functionalities, such as touchscreen inputs on mobile devices and desktop navigation. The system will be optimized for efficient operation on supported devices within the office environment, with no specialized hardware required beyond standard devices.

4.3 Software Interfaces

The product will integrate with various software components, including the operating systems (Windows, Linux, macOS, Android, iOS), cloud-based databases, and external libraries. It will use standard communication protocols for data exchange, including HTTP/HTTPS for secure data transmission. The system will interact with the database to manage student payment records, transaction logs, and user authentication information, with data flowing in and out based on user actions and system processes. Input data may include payment details, user credentials, and transaction requests, while output data will include confirmation messages, transaction records, and reports. Services required will include database access, authentication, and real-time synchronization. Data sharing between software components will occur through secure APIs, and any shared data will be managed to ensure consistency. If the system requires shared data to be accessed by multiple components concurrently, implementation constraints, such as the use of a global data area or locking mechanisms, will be specified.

4.4 Communications Interfaces

The product will require communication functions for secure data exchange and user interaction, including email notifications, web browser access, and network server communications. It will use standard communication protocols such as HTTP/HTTPS for secure web communication, with all data encrypted during transmission to ensure security. Email notifications will be formatted in a clear, consistent manner, alerting users about transaction statuses, system updates, and other relevant information. Data transfer will occur in real time, with synchronization mechanisms to ensure that all user actions and updates are reflected across devices and systems. FTP will be used for large file transfers, if needed, with security measures in place to protect the data during transfer. Communication security will be handled using SSL/TLS protocols for encrypted communication, and data will be validated before being transmitted to prevent errors. Data transfer rates will be optimized based on network conditions, ensuring smooth communication across the system.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The product must ensure high performance across various scenarios to maintain efficiency, accuracy, and reliability. Payment transactions should be processed within 2 seconds under normal conditions and within 5 seconds during peak loads of 100+ concurrent users. Financial reports containing up to 10,000 records must be generated within 5 seconds, while larger reports of 100,000+ records should be completed within 10 seconds, ensuring timely decision-making. The system must handle at least 500 transactions per second (TPS) without performance degradation, with database query responses not exceeding 2 seconds for single-user queries and 5 seconds for multi-user queries under heavy load. Real-time synchronization of financial data across clients and servers must occur within 1 second, with discrepancies resolved in 30 seconds. The UI should respond within 200 milliseconds for user interactions and navigate between pages in 1 second under normal conditions, extending to 3 seconds during high load. The system must maintain 99.9% uptime, with downtime limited to 2 hours per month, and automatic failover ensuring recovery within 30 seconds. Security measures, including authentication and encryption, should execute within 1 second, ensuring compliance without performance trade-offs. These requirements guarantee a seamless, high-speed, and reliable financial management experience.

5.2 Safety Requirements

The product must implement safeguards to prevent financial loss, data corruption, or unauthorized access that could compromise sensitive financial transactions. The system should include automated data backups every 24 hours to prevent loss due to hardware failures, cyberattacks, or accidental deletions, with recovery processes ensuring data restoration within 30 minutes. To protect against unauthorized access, multi-factor authentication (MFA) must be enforced, and all financial transactions should be encrypted using AES-256 to prevent interception or tampering. In case of system failures, automatic failover mechanisms must ensure continuity with minimal disruption, while unauthorized modifications to financial records should be blocked and logged for review. Compliance with financial and data protection regulations, such as GDPR, HIPAA, and local taxation laws (e.g., GST compliance in India), must be maintained, with periodic security audits to ensure adherence. The system must prevent fraudulent activities by enforcing role-based access control (RBAC), restricting financial actions based on user privileges. Any failed transactions, data mismatches, or anomalies must trigger real-time alerts to administrators for immediate action.

5.3 Security Requirements

The product must enforce multi-factor authentication (MFA) and role-based access control (RBAC) to prevent unauthorized access. All data must be encrypted using AES-256 for storage and TLS 1.3 for transmission. The system must comply with GDPR, HIPAA, PCI-DSS, and local financial regulations, ensuring secure financial transactions. Automatic session timeouts, account lockouts, and audit logs must be implemented for security monitoring. Regular security audits and vulnerability assessments must be conducted to maintain compliance with ISO 27001 and SOC 2. Sensitive data should be protected using anonymization and masking techniques to ensure user privacy.

5.4 Software Quality Attributes

The product must ensure 99.9% availability, supporting 24/7 operation with downtime limited to 2 hours per month. It must be highly reliable, processing 99.99% of transactions accurately without data loss. The system should be portable, running on Windows, macOS, Linux, Android, and iOS with seamless interoperability via REST APIs. Maintainability should be ensured through modular code architecture, allowing updates with minimal downtime. The system must be robust, handling up to 500 concurrent transactions per second without performance degradation. Testability must be achieved with automated unit, integration, and performance testing. Usability should prioritize a user-friendly UI, ensuring new users can complete financial tasks with minimal training. Reusability is key, with shared components for reporting, authentication, and payment processing.

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

Below are the key terms, acronyms, and abbreviations used in the SRS:

- FMS – Financial Management System
- SRS – Software Requirements Specification
- RBAC – Role-Based Access Control
- MFA – Multi-Factor Authentication
- TPS – Transactions Per Second
- GDPR – General Data Protection Regulation
- HIPAA – Health Insurance Portability and Accountability Act
- PCI-DSS – Payment Card Industry Data Security Standard
- ISO 27001 – International Standard for Information Security Management
- SOC 2 – Service Organization Control 2 (Security and Privacy Compliance)
- WCAG 2.1 – Web Content Accessibility Guidelines (Accessibility Compliance)
- TLS 1.3 – Transport Layer Security (Data Encryption)
- AES-256 – Advanced Encryption Standard (Data Security)
- UI – User Interface
- API – Application Programming Interface
- ACID Compliance – Atomicity, Consistency, Isolation, Durability (Database Transactions)
- TBD – To Be Determined (Pending Decision)

Appendix B: Issues List

Below is a dynamic list of open requirements issues that remain to be resolved :

1. Performance Metrics for Extreme Load Scenarios – Further clarification needed on system behavior when exceeding 500 TPS.
2. Localization and Internationalization Support – Pending decision on supported languages and currency formats.
3. Integration with External Tax Calculation Services – Need to finalize APIs and compliance requirements for taxation laws.
4. Disaster Recovery and Failover Testing Strategy – Specifics on real-world failover testing scenarios remain undecided.
5. User Role Definitions and Privileges – Need to confirm the access levels and security constraints for different user roles.
6. Security Compliance Certification Process – Pending timeline and roadmap for achieving ISO 27001, PCI-DSS, and SOC 2 certifications.
7. AI-Driven Fraud Detection Implementation – Further research required to determine feasibility and accuracy thresholds.
8. Database Backup Frequency Customization – Decision required on whether organizations can modify backup intervals.