Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements for FinVerse, a web-based trading platform that provides stock and currency exchange simulation. The platform integrates Artificial Intelligence (AI) to provide trading insights and includes a real payment gateway to manage user wallets with credit/debit card transactions. This document serves as a guide for developers, stakeholders, and testers to ensure alignment with the project goals.

1.2 Scope

FinVerse is designed as a web application that enables users to register, log in, and perform currency and stock exchange transactions. Users will be able to deposit money into their wallet using real payment gateways and utilize these funds for trading activities. The system will also feature AI-driven recommendations for portfolio management and trading decisions. Admins will have a dashboard to manage users and content. The application will provide a secure, responsive, and user-friendly interface accessible via modern browsers.

1.3 Definitions, Acronyms, and Abbreviations

API: Application Programming Interface
AI: Artificial Intelligence
UI: User Interface
UX: User Experience
DBMS: Database Management System
PCI-DSS: Payment Card Industry Data Security Standard

1.4 References

- FinVerse Project Plan (University Submission)
- IEEE SRS Standards
- Payment Gateway Documentation (e.g., Stripe, PayPal)
- Stock Market APIs (e.g., Alpha Vantage, Yahoo Finance)

1.5 Overview

This document outlines the functional and non-functional requirements of the FinVerse application. It includes system models such as use case diagrams, entity-relationship diagrams, and sequence diagrams to demonstrate system behavior and relationships. Appendices provide glossary and additional supporting information.

2. Overall Description

2.1 Product Perspective

FinVerse is an independent web application but relies on external APIs for stock and currency data and third-party payment gateways for wallet transactions. It integrates AI modules for decision-making and financial recommendations. The product is modular and follows an MVC-based architecture.

2.2 Product Functions

- User registration and authentication
- Trading of stocks and currencies (buy/sell)
- Wallet management with credit/debit card payment integration
- Portfolio tracking with profit/loss calculation
- AI-driven recommendations for trading
- Admin dashboard for monitoring users and content

2.3 User Classes and Characteristics

- End Users: Individuals performing trades, need an intuitive UI
- Admins: Oversee system and user management
- External Systems: Payment gateways and stock APIs

2.4 Operating Environment

The system will operate as a web application accessible via major browsers (Chrome, Firefox, Edge). Backend will run on a server with DBMS (MySQL/PostgreSQL). The application requires HTTPS for secure communications.

2.5 Design and Implementation Constraints

- Limited to academic project timeframe
- Compliance with PCI-DSS for payment security
- Dependency on third-party APIs (rate limits, downtime)

2.6 User Documentation

User manuals, online FAQs, and an admin guide will be provided.

3. Specific Requirements

3.1 Functional Requirements

1: The system shall allow users to register and log in securely.
2: The system shall allow wallet management including deposits via payment gateways.
3: The system shall enable users to buy/sell currencies and stocks.
4: The system shall track user portfolio and display profit/loss.
5: The system shall provide AI-driven trading recommendations.
6: The system shall include an admin dashboard for user/content management.

3.2 Non-Functional Requirements

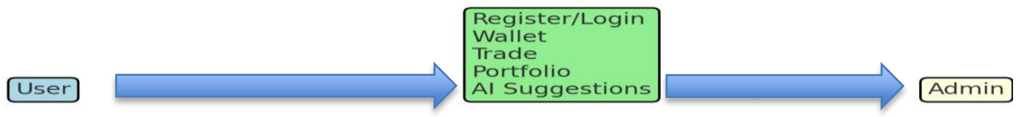
1: Security – Encrypted communications (HTTPS, TLS) and PCI-DSS compliance.
2: Performance – Handle 1000 concurrent users with <2s response.
3: Usability – Mobile responsive and user-friendly interface.
4: Availability – 99% uptime target.

3.3 Interface Requirements

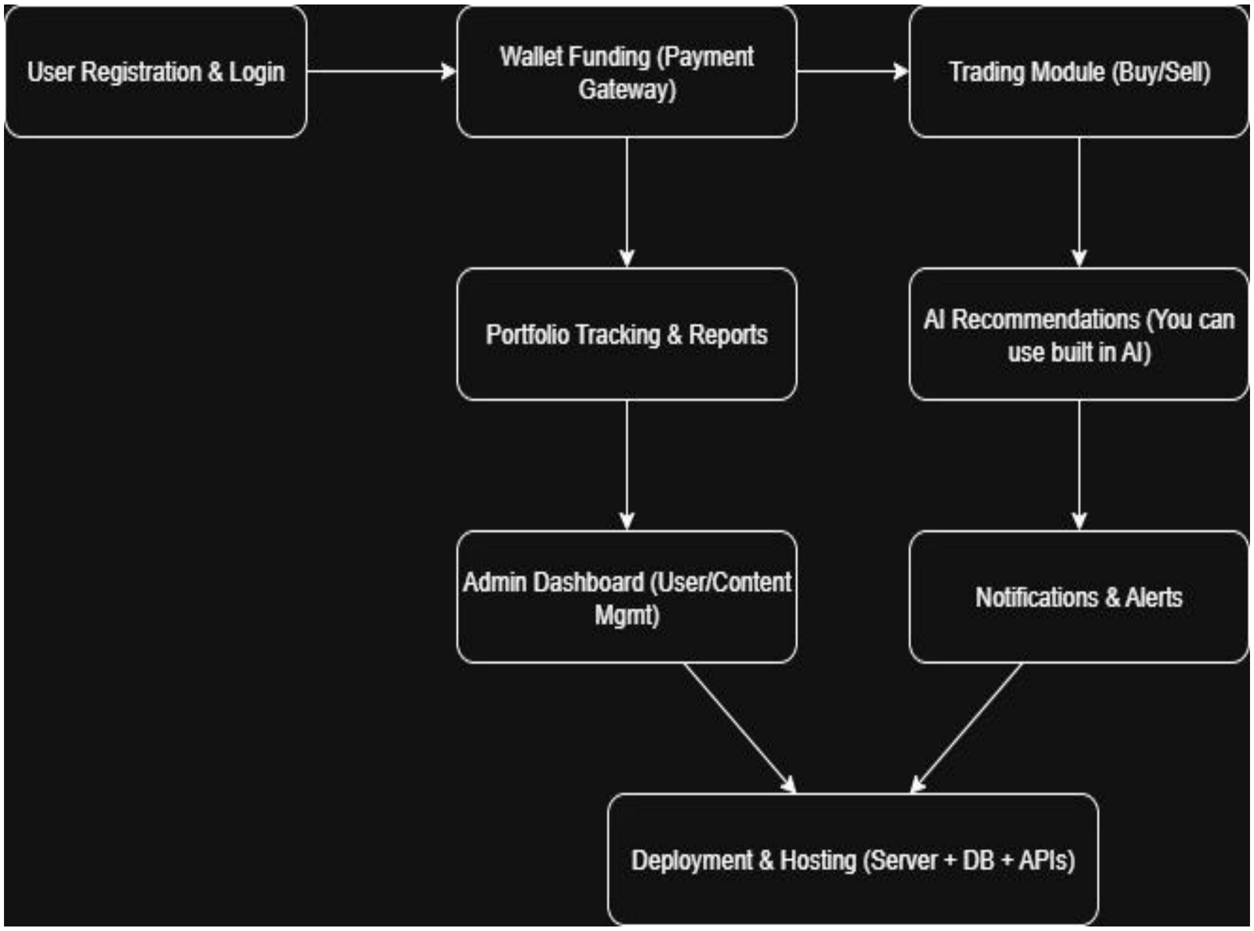
- User Interface: Responsive UI with trading dashboard, wallet view, AI suggestions.
- Hardware Interface: Standard PC/mobile devices with browsers.
- Software Interface: APIs (stock data, AI models, payment gateways).
- Communications Interface: HTTPS, WebSocket for live updates.

4. System Models

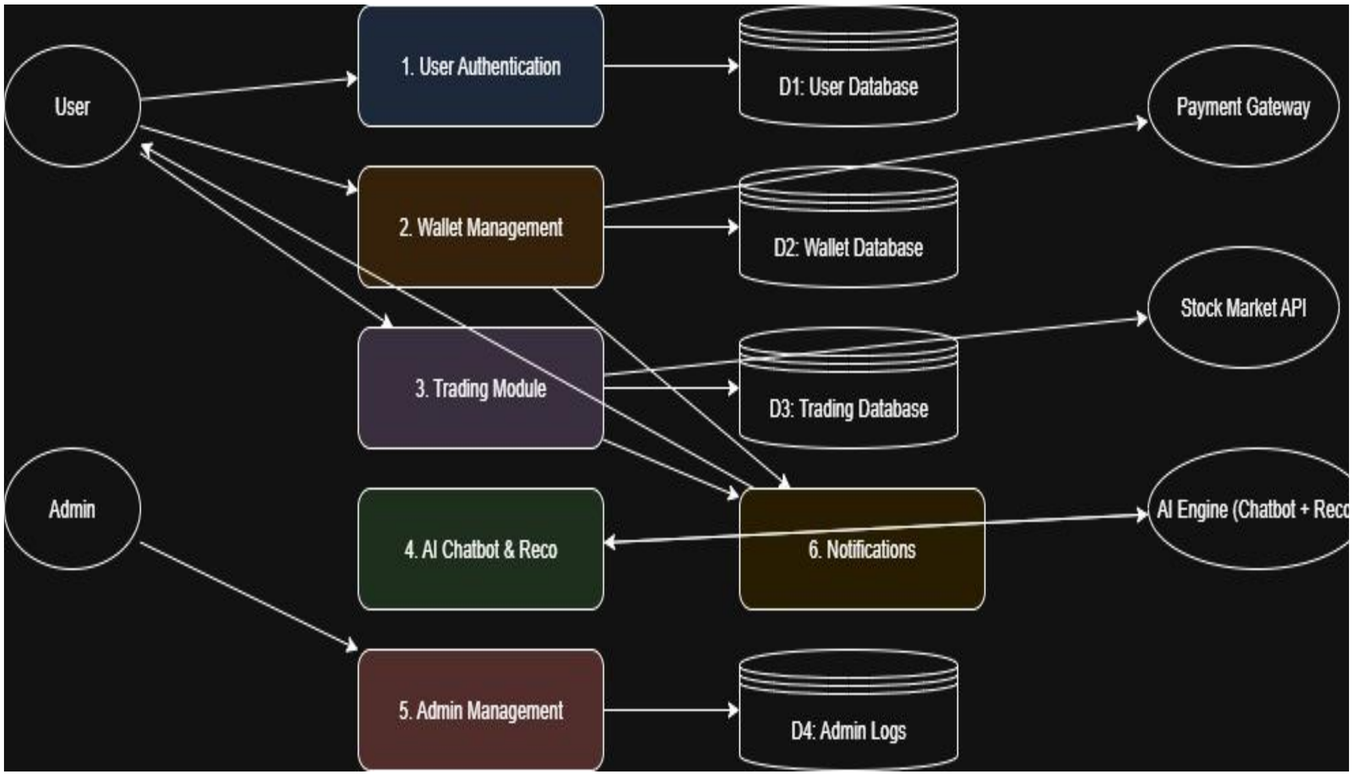
(Diagrams are designed by PM using Draw.io and Canva)



4.1 Work Flow of Website


User Registration & Login
Wallet Funding (Payment Gateway)
Trading Module (Buy/Sell)
Portfolio Tracking & Reports
AI Recommendations (You can use built in AI)
Admin Dashboard (User/Content Mgmt)
Notifications & Alerts
Deployment & Hosting (Server + DB + APIs)

4.2 DFD Level 1:



5. Appendices

Glossary

Forex: Foreign Exchange
API: Interface allowing communication between systems
Wallet: Digital storage for user funds
Portfolio: Collection of assets owned by user
Recommendation System: AI-based suggestion module

Appendix A – Assumptions & Dependencies

A1. Trading remains simulation-only; real financial settlement is not performed on exchanges.
A2. Payment processing relies on a PCI-DSS certified gateway; FinVerse does not store PAN data.
A3. Market data latency and accuracy depend on the chosen data provider.

A4. AI recommendations are advisory and not financial advice; users retain full responsibility.

Appendix B – Requirements Traceability Matrix (Excerpt)

FR-1.1 → UC-Login/Register, Seq-Payment (pre-req)
FR-2.1 → Seq-Payment, Component-Integrations
FR-3.2 → UC-Trade, Component-Core Services
FR-4.1 → Activity-AI, Component-AI Service
NFR-1 → Deployment, Component (security), Interfaces