Software Requirements Specification for Hisab Khata

Prepared by: Abhishek Sharma

Github: @4bhishekshar ma

Date Created: November 13, 2025

Table of Contents

Table of Figures:

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the Hisab Khata mobile and web application. Hisab Khata is a credit/due management and transaction tracking system designed for small to medium-sized businesses (shopkeepers) and their customers in Nepal.

The document specifies all functional and non-functional requirements for the Hisab Khata system.

1.2 Document Conventions

This document follows the following conventions:

- **Priority Levels:**
  - **Critical:** Must be implemented for system to function
  - **High:** Important features that significantly impact user experience
  - **Medium:** Desirable features that enhance functionality
  - **Low:** Optional features for future versions

- **Assumed Inheritance:** High-priority requirements for system features imply that all sub-requirements inherit that priority unless explicitly stated otherwise.

1.3 Intended Audience and Reading Suggestions

This SRS is organized to serve different stakeholders:

- **For Developers:**
  - Read External Interface Requirements and System features for detailed functional specifications.
  - Reference Design and implementation constraints for technical constraints
  - Review appendix B for system architecture and data models.
- **For Project Management**
  - Start with a product scope for project overview

- Review overall description for high level understanding
- Focus on Section 4 for future breakdown and priorities

- **For Quality Assurance/Testers:**
  - System features provide detailed test scenario
  - Non-Functional requirement issues wise performance and security criteria
  - Appendix C lists items requiring clarification

- **For End Users and Stakeholders:**
  - Scope explains business value
  - Product functions provide the feature overview
  - Use cases describe different user roles

1.4 Product Scope

Hisab Khata is a comprehensive credit/due management and transaction tracking platform designed to digitize and simplify the traditional "Baki Khata" (credit ledger) system commonly used by small businesses in Nepal.

Business Context: Small shopkeepers in Nepal traditionally maintain handwritten ledgers to track customer credit transactions.
This manual process is:

- Error-prone and time-consuming
- Lacks transparency for customers
- Difficult to track payment reminders
- Prone to disputes and loss of records

Product Vision: Hisab Khata aims to provide a digital, mobile-first solution that:

- Eliminates paper-based credit tracking
- Provides real-time visibility to both businesses and customers
- Automates payment reminders and calculations
- Enables digital payment integration
- Builds trust through transparent record-keeping

Key Benefits:

- **For Business Owners:**
    - Reduce time spent on manual bookkeeping by ~70%
    - Minimize calculation errors and disputes
    - Automated payment reminders increase collection rates
    - Real-time analytics for better business decisions
    - AI-powered features for faster data entry

- **For Customers:**
    - Complete transparency of credit history
    - Convenient digital payment options
    - Payment reminders to avoid late payments
    - Build credit reputation through loyalty points
    - Easy access to transaction statements

2 Overall Description

2.1 Product Perspective

Hisab Khata is a new, self-contained product developed specifically for the Nepali market. It is not a replacement or extension of any existing system.

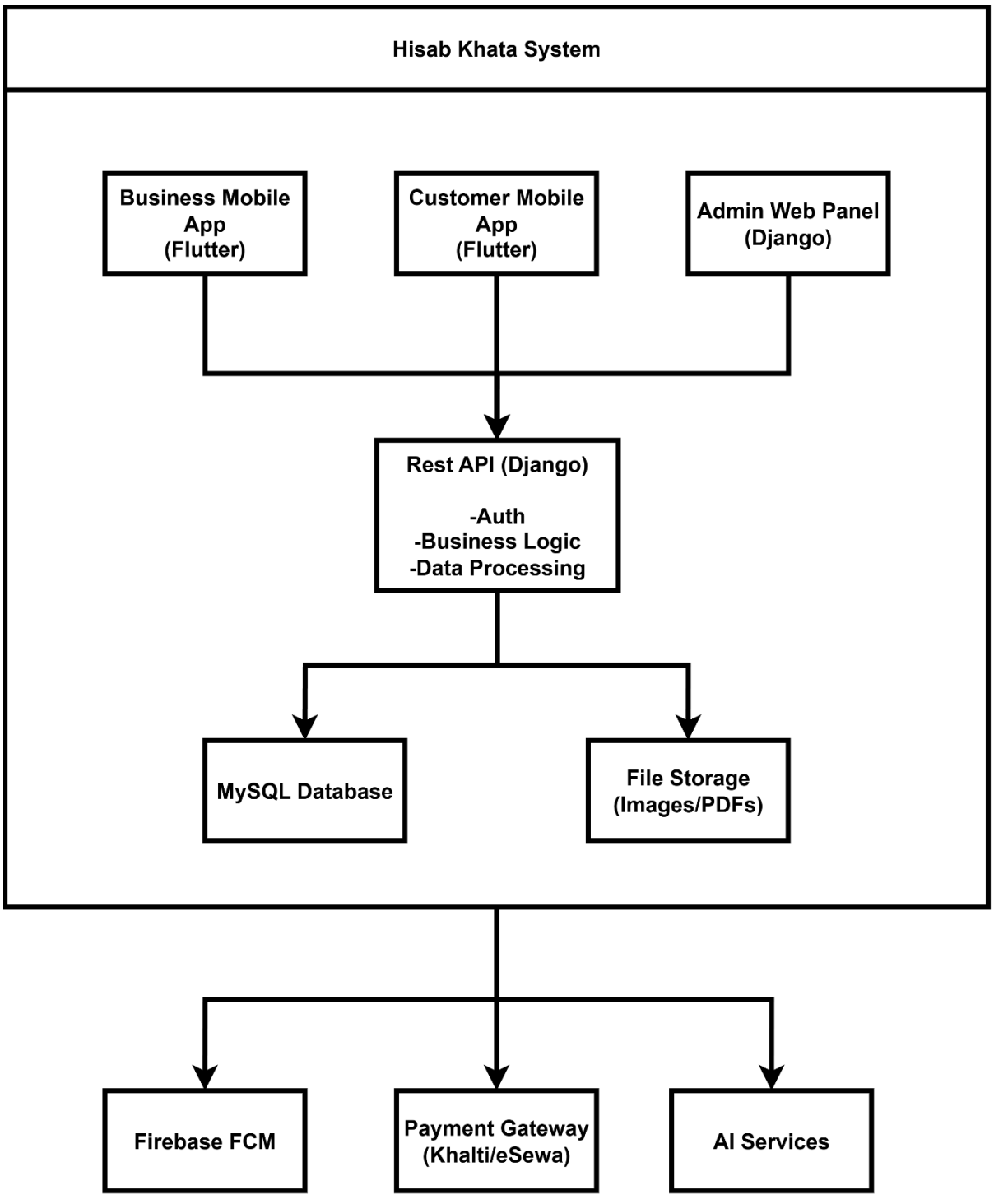

Figure 1: System Context Fig

External System Interfaces:

- **Firebase Cloud Messaging (FCM):**
    - Purpose: Push notifications and real-time chat
    - Integration: Client SDK in Flutter, Admin SDK in Django

- **Payment Gateways:**
    - Khalti: Digital wallet payment processing
    - eSewa: Digital wallet payment processing
    - Integration: RESTful API calls

- **AI/ML Services:**
    - Google ML Kit API: OCR for receipt scanning
    - OpenAI/Gemini API: AI chatbot for financial tips

- **System Boundaries:**
    - The system manages all business-customer credit relationships and transactions
    - Does NOT manage inventory, supply chain, or procurement
    - Does NOT provide accounting/bookkeeping services
    - Does NOT integrate with government tax systems

2.2 Product functions

2.2.1 Authentication and User Management

- Email Registration
- OTP verification
- Login/Logout
- Profile Management
- Session Tracking/Management

2.2.2 Transaction Management

- Record Sales
- Record Credit
- Record Payments
- Auto Calculations
- Transaction History

2.2.3 Customer Management

- Add Customers
- Search Customers
- View Profiles
- Request System

2.2.4 Dashboard and Analytics

- Total Due Summary
- Customer Balances
- Income/Expense
- Charts and Graphs
- PDF Reports

2.2.5 Communication Features

- Real-Time Chat
- Push Notifications
- Payment Reminders
- Budget Alerts
- System Broadcasts

2.2.6 Payment Intregration

- eSewa Gateway
- Khalti Gateway
- Payment Tracking

2.2.7 AI Powered Features

- Receipt Scanning
- Voice Transaction
- AI Chatbot
- Smart Suggestions
- Category Detection

2.2.8 Admin Panel Function

- User Management
- Platform Status
- Broadcasting
- Support Tickets

2.2.9 Localization and UI/UX

- Nepali Language
- English toggle
- Custom Fonts
- Minimal Design
- Bottom Navigation Bar

2.3 User Classes and Characteristics

Hisab Khata servs three distinct user classes:

2.3.1 User Class 1: Business Owner/Shopkeeper

Description: Small to medium-sized business owners who want to digitize their credit tracking system.

Characteristics:

- **Technical Expertise: Low to Medium**
  - Comfortable with smartphone usage
  - May not be tech person
  - Need simple, intuitive interfaces
- **Frequency of Use: Daily (5-10 times per day)**

- **Primary Goals:**
  - Track customer credit transactions
  - Manage customer payments
  - View business analytics
  - Send payment reminders

- **Demographics:**
  - Age: 18 – 60+ years
  - Education: Above 7 class
  - Location: Urban, semi-urban areas in Nepal
  - Business Types: Grocery stores, medical shops, hardware stores, etc.

2.3.2 User Class 2: Customer

Description: Individuals who purchase goods/services on credit from businesses using Hisab Khata.

Characteristics:

- **Technical Expertise: Low to Medium**
  - Basic smartphone users
  - Prefer simple interfaces

- **Frequency of Use: Weekly to Monthly**
  - Check dues periodically
  - Make payments as needed

- **Primary Goals:**
  - View current dues
  - Make digital payments
  - Download transaction history
  - Manage multiple business relationships

- **Demographics:**
  - Age: 18 – 55+ years
  - Education: Class 7 above
  - Location: Urban and semi-urban Nepal

2.3.3 User Class 3: System Administration

Description: Technical staff responsible for platform management, user support, and system health monitoring.

Characteristics:

- Technical Expertise: High
    - IT professionals
    - Understand system architecture
    - Can troubleshoot technical issues

- Frequency of Use: Daily (6+ hours)

- Primary Goals:
    - Monitor platform health
    - Manage user accounts
    - Respond to support tickets
    - Detect fraudulent activity
    - Generate platform reports

- Demographics:
    - Age: 22-40 years
    - Education: IT/Computer background
    - Location: Office

2.4 Operating Environment

2.4.1 Clint Side Environment

Mobile Application (Flutter):

- **Supported Platforms:**
  - Android: Version 7.0 and above
  - iOS: iOS 12.0 and above

- **Device Requirements:**
  - Minimum RAM: 2 GB
  - Storage: 100 MB free space
  - Camera: Required for receipt scanning
  - Microphone: Required for voice transactions

- **Recommended Devices:**
  - Budget Android smartphones (Rs. 10,000 above range)
  - Common in Nepal: Xiaomi, Samsung, Realme, Nokia, Apple

- **Network Requirements:**
  - Internet: 2G/3G/4G/WiFi
  - Offline capability with background sync
  - Bandwidth: Low

Web Application (Admin Panel):

- **Supported Browsers:**
  - Google Chrome
  - Microsoft Edge
  - Safari

- **Desktop Requirements:**
  - Operating System: Windows 10+, macOS 10.14+, Linux (Ubuntu 20.04+)
  - RAM: 4 GB minimum
  - Internet: Stable broadband connection

2.4.2 Server-Side Environment

- **Production Server:**
  - Database Server: MySQL 8.0+

- **Development Environment:**
  - Backend: Django with Python
  - Frontend: Flutter with Dart
  - Database: MySQL 8.0+
  - Version Control: Git with GitHub
  - IDE: VS Code, Android Studio

2.4.3 Third-Party Service Dependencies

- **Firebase Services:**
  - Firebase Cloud Messaging (FCM) for push notifications

- **Payment Gateways:**
  - Khalti API
  - eSewa API

- **AI/ML Services:**
  - Google ML Kit API (OCR)
  - Google Speech-to-Text API
  - OpenAI API / Google Gemini API (Chatbot)

- **Email Service:**
  - SMTP Server for OTP
  - Email verification and notifications

3 External Interface Requirements

3.1 User Interface

This section describes the user interface requirements for the Hisab Khata mobile application (Flutter) and web admin panel (Django).

3.1.1 Mobile Application UI(Flutter)

Design Principles:

- Material Design 3 guidelines
- Minimalist and clean interface
- Nepali language support with custom fonts
- Bottom navigation for primary sections

3.1.2 Common UI components

- Bottom Navigation bar
- Authentication Screens
    - Login Screen
    - Registration Screen
    - OTP Verification Screen
- Business Owner Screens
    - Dashboard Screen
    - Add Transaction Screen
    - Customer List Screen
    - Customer Profile Screen
    - Analytic Screen
    - Chat Screen
- Customer Screens
    - Customer Dashboard
    - Transactions History
    - Payment Screen
    - Business Search/Add Screen

- Common Screens
    - Setting Screen
    - Notifications Screens
- Admin Web Panel UI
    - Admin Dashboard

3.2 Hardware Interface

3.2.1 Mobile Device Hardware

- Camera Interface
- Microphone Interface
- Storage Interface
- Network Interface

3.3 Communication Interface

3.3.1 Network Protocol

- HTTP/HTTPS Protocol
- WebSocket Protocol
- RESTful API
- JWT

4 Core System Features

This section describes the functional requirements organized by system features. Each feature is described with priority, stimulus/response sequences, and detailed functional requirements.

4.1 Authentication and Session Management

Purpose: Secure user login, registration, and session handling

Key Features:

- Email-based registration with OTP verification
- JWT token authentication (24-hour access, 30-day refresh)
- Multi-device session management
- Password reset functionality
- Account lockout after 5 failed attempts
- Profile management (name, photo, language preference)

4.2 Transaction Management

Purpose: Record and manage business sales and credit transactions

Key Features:

- Record Normal Sales (fully paid) and Credit Sales (Dew/Udhaaro)
- Record customer payments (cash/digital)
- Automatic calculation of due amounts
- Itemized billing (optional)
- Transaction history with search and filters
- Receipt image upload
- Edit/delete within 24 hours
- Transaction number generation

4.3 Customer Management

Purpose: Manage business-customer relationships

Key Features:

- Add customers via search (by name, email, phone)
- Send/receive connection requests
- Accept/reject/block customers
- View customer profiles with transaction history
- Customer search and filtering
- View total due per customer

4.4 Dashboard and Analytics

Purpose: Real-time business insights and reporting

Key Features:

- **Business Dashboard:**
    - Total due overview across all customers
    - Total collected (monthly)
    - Active customers count
    - Income vs Expense charts
    - Top customers by due amount
    - Highest/Lowest due identification
    - Recent transactions feed

- **Customer Dashboard:**
    - Total due across all businesses
    - Loyalty points balance
    - Connected businesses list
    - Spending analysis

- **Reports:**
    - Generate monthly income/expense PDF
    - Generate customer statement PDF
    - Download due summary report
    - Export data as CSV

4.5 Real Time Chat System

Purpose: Direct communication between businesses and customers

Key Features:

- One-to-one chat per business-customer relationship
- Real-time messaging (Firebase/WebSocket)
- Typing indicators
- Read receipts (sent, delivered, read)
- Unread message count
- Message history with search
- Share transaction links in chat
- Push notifications for new messages

4.6 Payment Gateway Integration

Purpose: Enable digital payments through popular Nepali wallets

Key Features:

- Khalti Integration: Pay via Khalti digital wallet
- eSewa Integration: Pay via eSewa digital wallet
- Cash payment marking (requires business confirmation)
- Payment verification and confirmation
- Payment receipt generation (PDF)
- Payment history tracking
- Digital payment notifications

4.7 AI Powered Features

Purpose: Enhance user experience with AI automation

Key Features:

- **AI Receipt Scanning (OCR):**
  - Capture receipt photo
  - Extract total amount, date, items
  - Auto-fill transaction form
  - Google ML Kit API integration

- **Voice Transaction Entry (Quick-Hisab):**
  - Voice command: "Credit sale to Ram Kumar 500 rupees"
  - Speech-to-Text (Nepali and English)
  - Parse and auto-fill transaction

- **AI Chatbot:**
  - Financial tips and business advice
  - Context-aware conversations
  - OpenAI/Gemini integration

4.8 Notification System

Purpose: Keep users informed via push and in-app notifications

Key Features:

- **Notification Types:**
  - Payment reminders (auto and manual)
  - Due alerts (budget exceeded)
  - Transaction notifications
  - Payment confirmations
  - Chat messages
  - Connection requests
  - System announcements

- **Features:**
    - Push notifications (Firebase FCM)
    - In-app notification center
    - Bulk payment reminders (100 customers at once)

4.9 Admin Panel Features

Purpose: Web-based platform management for administrators

Key Features:

- **User Management:**
    - View all users (businesses and customers)
    - Activate/deactivate accounts
    - Block fraudulent users
    - View user activity logs
    - Search and filter users

- **Analytics:**
    - Platform statistics (users, transactions, revenue)
    - User growth charts
    - Transaction volume trends
    - Business distribution

- **Support:**
    - Support ticket management
    - View and respond to tickets

- **Fraud Detection:**
    - Suspicious activity monitoring
    - Review flagged users
    - Block/unblock users

- **System Settings:**
    - Configure platform settings
    - Broadcast notifications
    - View system logs

4.10 Localization Features

Purpose: Support Nepali and English languages

Key Features:

- Two Languages: Nepali and English
- Nepali Fonts
- Bikram Sambat (Nepali calendar) support
- Currency: NPR (रू / Rs.)
- Language Toggle: Easy switch in settings

4.11 Loyalty Points (Gazab Customer point)

Purpose: Reward reliable customers and encourage engagement

Key Features:

- Gazab Customer Points:
    - Earn points for on-time payments (1 point per NPR 100)
    - Lose points for late payments
    - Points visible to all businesses

- Trust badges based on points:
    - New Customer (0-1)
    - Good Customer (2-5)
    - Reliable Customer (5-8)
    - Trusted Customer (9-10)

5 Other Non-Functional Requirements

5.1.1 Security

- Two-Factor Authentication (2FA): Add an extra step to login using email or SMS for better security.
- Data Encryption: Protect data by coding it while transferring and storing it.
- Secure APIs: Ensure only authorized users can access the system through secure connections.
- Regular Updates: Keep the system safe by fixing security issues regularly.
- Firewall Protection: Use firewalls to block unauthorized access to the system.
- Password Policies: Require strong passwords and periodic password changes for users.

5.1.2 Performance

- App/Page Load Time: Pages should load quickly, in 2 seconds or less.
- Concurrent Users: The system should handle up to 5,000 users at the same time.
- Real-Time Operations: Inventory changes should be updated immediately.
- Load Balancing: Distribute traffic across multiple servers to prevent overloading any single server.

5.1.3 Usability

- Intuitive Design: Make the system simple to use, requiring little training.
- Accessibility: Ensure everyone, including people with disabilities, can use the system.
- Multi-Device Support: The system should work well on computers, tablets, and phones.
- User Customization: Allow users to personalize their dashboards and settings for easier use.
- Clear Navigation: Provide a well-organized menu and options to easily find features.

- Help and Support: Include easy-to-access help documentation and live support options.
- Error Feedback: Provide clear error messages and suggestions for correction.

5.1.4 Reliability

- Uptime Guarantee: The system should be available 99.9% of the time.
- Data Integrity: Keep all data correct and safe.
- Backup and Recovery: Save data daily and recover it quickly if something goes wrong.
- Failover Mechanism: Ensure there's a backup system that can take over in case of a failure.
- Redundancy: Implement redundant systems for critical functions to avoid downtime.
- Error Logging: Record errors and system failures to aid in troubleshooting.
- System Monitoring: Continuously monitor system health to identify potential issues before they affect performance.
- Disaster Recovery: Implement a clear plan to recover the system after major failures.

5.1.5 Maintainability

- Modular Design: Build the system in parts so updates are easy.
- Code Readability: Write the system in a clear way for others to work on easily.
- Version Control: Use version control systems like Git to track changes and manage code.
- Error Handling: Implement a systematic approach for catching and logging errors.
- Easy Debugging: Build the system in a way that makes it easy to locate and fix bugs.
- Flexible Updates: Ensure that updates can be deployed with minimal disruption to the users

6 Other Requirements

6.1 Legal Requirements

- Ensure compliance with local data protection laws in Nepal.
- Maintain audit trails for inventory transactions for at least five or ten years.

6.2 Database Requirements

- Use a relational DBMS such as MySQL, PostgreSQL, MongoDB etc.
- Ensure daily data backup with automated recovery procedures.

6.3 Storage and Backup Requirements

- Minimum storage of 1 TB or more to handle initial data volume.
- Incremental and full backups scheduled weekly or daily.

6.4 Other Requirements beside these requirements

- Comprehensive user training program with supporting documentation.
- Support for multiple languages, including English, Nepali and other applicable languages.