# Telhan Sathi — Real-time Auction & Farmer Marketplace

**Software Requirements Specification (SRS)**
**Document Version:** 1.1
**Last Updated:** 2026-01-08
**Authors:** Telhan Sathi Team

---

## 1. Introduction

### 1.1 Purpose

This Software Requirements Specification (SRS) document defines the **functional** and **non-functional** requirements of the **Telhan Sathi** platform. It focuses on: - Farmer-side features (10 major modules) - Buyer-side features - AI/ML capabilities (ARIMA forecasting & profitability calculations)

This document is intended for **product owners, developers, QA teams, and project managers**.

### 1.2 Scope

Telhan Sathi is a **web-based farmer marketplace and real-time auction platform**. Farmers list crop lots for auction, and buyers bid competitively. The platform supports: - Real-time auctions (NILAMI) - Communication between buyers and farmers - Forecasting & profit simulation - Subsidy information - Coin-based redemption system

> ⚠️**Important:** Payments and physical delivery are handled **outside the platform** by farmers and buyers directly. This SRS does not define payment or delivery processing.

### 1.3 Definitions, Acronyms & Abbreviations

- **NILAMI** – Auction subsystem
- **ARIMA** – AutoRegressive Integrated Moving Average
- **OTP** – One-Time Password
- **IoT** – Internet of Things
- **SKU** – Stock Keeping Unit

### 1.4 References

- Project README & repository
- Business requirements from product team

---

# 2. Overall Description

## 2.1 Product Perspective

Telhan Sathi is a single integrated system supporting **Farmers** and **Buyers**. It integrates with: - SMS/OTP services - Weather APIs - Optional IoT (MQTT / HTTPS) - AI/ML forecasting services

The system consists of a **responsive web frontend** and **server-side REST + WebSocket APIs**.

## 2.2 User Roles & Characteristics

- **Farmer:** Lists auctions, receives bids, chats with buyers, redeems coins
- **Buyer:** Browses auctions, places bids, communicates with farmers
- **Admin:** Manages subsidies, moderation, dispute support
- **System:** AI/ML components for forecasting and analytics

## 2.3 Operating Environment

- Web browsers (desktop & mobile)
- Backend: Python (Flask), SQL database
- Optional IoT devices

## 2.4 Design Constraints

- Real-time bidding requires WebSockets / Socket.IO
- Secure handling of credentials and secrets

---

# 3. Functional Requirements

## 3.1 Farmer-Side Features (10 Modules)

### 3.1.1 Authentication & Onboarding

- **FR-F-1:** Farmer Registration via phone & OTP
- **FR-F-2:** Profile setup (location, crops, land size)
- **FR-F-3:** Login & account recovery

### 3.1.2 Dashboard & Home

- **FR-F-4:** Dashboard overview (auctions, bids, coins, messages)
- **FR-F-5:** Real-time notifications

### 3.1.3 Auction Management (NILAMI)

- **FR-F-6:** Create auction (crop, quantity, grade, reserve price)
- **FR-F-7:** Edit auction before first bid

- **FR-F-8:** Publish / schedule auctions

#### 3.1.4 Bid Management

- **FR-F-9:** Receive bids in real time
- **FR-F-10:** Bid history & leaderboard
- **FR-F-11:** Accept / reject bid and conclude auction

#### 3.1.5 Communication

- **FR-F-12:** Chat with buyers
- **FR-F-13:** Chat history & attachments

#### 3.1.6 Profit Simulator (AI)

- **FR-F-14:** Run ARIMA-based profit simulations
- **FR-F-15:** Compare scenarios

#### 3.1.7 Crop Economics

- **FR-F-16:** Crop comparison dashboard
- **FR-F-17:** Cost breakdown

#### 3.1.8 Weather & Field Monitoring

- **FR-F-18:** Weather alerts
- **FR-F-19:** IoT sensor integration (optional)

#### 3.1.9 Subsidies & Benefits

- **FR-F-20:** Browse subsidy programs
- **FR-F-21:** Eligibility checks

#### 3.1.10 Redemption Store

- **FR-F-22:** Redeem coins
- **FR-F-23:** Earn coins via activities

---

### 3.2 Buyer-Side Features

- **FR-B-1:** Buyer registration
- **FR-B-2:** Login & account management
- **FR-B-3:** Buyer dashboard
- **FR-B-4:** Search & filters
- **FR-B-5:** Place bids
- **FR-B-6:** Track bids
- **FR-B-7:** Counter-offer negotiation
- **FR-B-8:** Watchlist & alerts

- **FR-B-9:** Chat with farmers
- **FR-B-10:** Profile & address management

---

### 3.3 AI/ML Features

- **FR-A-1:** ARIMA price forecasting
- **FR-A-2:** Forecast visualization & guidance
- **FR-A-3:** ROI & profitability calculation
- **FR-A-4:** Scenario comparison

---

## 4. Non-Functional Requirements

### 4.1 Performance

- Bid latency $\leq$ 2 seconds
- Page load $\leq$ 3 seconds

### 4.2 Scalability

- Horizontal scaling support

### 4.3 Availability & Reliability

- 99.5% uptime
- Daily backups

### 4.4 Security

- Secure password hashing
- OTP expiry
- Role-based access control
- TLS encryption

### 4.5 Privacy

- Minimal personal data usage

### 4.6 Usability

- Multilingual support
- Mobile-friendly UI

### 4.7 Maintainability

- Structured logging & monitoring

### 4.8 Compliance

- Clear guidance for **off-platform payment & delivery**

---

## 5. System Architecture Overview

- Web frontend + REST APIs
- WebSocket for real-time bidding
- Background workers
- SQL database
- Optional MQTT / IoT integration

---

## 6. Data Model (Key Entities)

- User
- Auction
- Bid
- ChatMessage
- CoinTransaction
- Subsidy
- Forecast

---

## 7. External Interfaces

- SMS / OTP service
- Weather API
- IoT Broker
- Object storage

---

## 8. AI/ML Implementation Notes

- Periodic ARIMA retraining
- Confidence intervals
- Model metadata transparency
- No personal data usage

---

## 9. Use Cases

- **UC-F-1:** Farmer auction lifecycle
- **UC-B-1:** Buyer bidding lifecycle

- **UC-A-1:** Profit simulation

---

## 10. Acceptance Criteria

- Real-time bidding works within SLA
- Auction closes correctly
- Forecast accuracy & clarity
- Chat & notifications are reliable

---

## 11. Risks & Mitigations

- Fraud → OTP, reputation, moderation
- Disputes → chat records & guidance
- Model drift → retraining

---

## 12. Appendix

- Sample APIs: `/api/v1/auctions`, `/api/v1/bids`, `/api/v1/forecast`, `/api/v1/chats`

---

**End of SRS v1.1**