**Software Requirements Specification (SRS)** StockTrackPro

Project: StockTrackPro  
Prepared For: Zeyn Caterers  
Author: Amar Salim Batheif  
Date: July 29, 2025

**1. Introduction**

**1.1 Product Name**

**StockTrackPro**

**1.2 Purpose of this Document**

To define the software requirements of StockTrackPro for all stakeholders including the developer, business owner, and testers.

**1.3 Scope**

This document describes the features and behavior of the stock management application StockTrackPro, including user management, stock-taking sessions, inventory logs, permissions, and reporting.

**1.4 Intended Audience**

Developers, QA engineers, stakeholders (e.g., Zeyn Caterers), and future maintainers of the app.

**1.5 System Environment**

* Flutter frontend
* Spring Boot backend
* Firebase authentication and backup
* Android devices
* Optional web dashboard for admins

**2. Overall Description**

**2.1 Product Perspective**

StockTrackPro is a mobile-first inventory system designed to digitize and streamline the stock-taking process.  
It integrates:

* **Flutter** for cross-platform app delivery
* **Spring Boot** backend for REST APIs
* **Firebase** for user authentication and cloud storage
* An **optional admin dashboard** for advanced reporting

It replaces manual spreadsheets and unstructured tracking for businesses like Zeyn Caterers.

**2.2 Product Functions**

* Secure login using Firebase (email/password or OTP)
* Create, update, deactivate products
* Associate products with suppliers
* Perform stock-take sessions (start, input, submit)
* Real-time dashboard: fast-moving, low-stock, out-of-stock
* Activity log of all user actions
* Role-based permissions (admin, manager, staff, viewer)
* Exportable reports in PDF and Excel

**2.3 User Characteristics**

| **Role** | **Technical Skill** | **Access Level** |
| --- | --- | --- |
| Admin | High | Full access to all features and settings |
| Manager | Medium | Can manage inventory and reports |
| Staff | Basic | Perform stock-taking, limited view |
| Viewer | Basic | View-only access to stock info and reports |
| Supplier | N/A | No login access; used as reference metadata only |

**2.4 Constraints**

* MVP supports **only one warehouse**
* Requires active internet for Firebase sync
* No AI or POS integrations
* No camera-based image recognition for counting

**2.5 Assumptions & Dependencies**

* Users have Android devices
* Firebase is operational for authentication and backup
* Admins manage their team and permissions
* Project follows Agile methodology

**3. Functional Requirements**

| **Code** | **Functional Requirement** |
| --- | --- |
| FR-01 | Users shall log in securely via Firebase |
| FR-02 | Admins can create, update, and deactivate products |
| FR-03 | Products shall be associated with suppliers |
| FR-04 | Users can perform full stock-take sessions |
| FR-05 | All stock changes are logged with timestamp and user |
| FR-06 | Permissions are enforced based on user roles |
| FR-07 | Users can generate and export reports |
| FR-08 | The dashboard provides stock insights (fast-moving, low-stock) |
| FR-09 | Admins must approve sensitive actions like negative adjustments |
| FR-10 | Admins can manage users, roles, and permissions internally |

**4. Non-Functional Requirements**

| **Code** | **Non-Functional Requirement** |
| --- | --- |
| NFR-01 | App should load dashboard within 3 seconds |
| NFR-02 | System must be 99.5% available during business hours |
| NFR-03 | Sensitive data is encrypted at rest and in transit |
| NFR-04 | App should support Android devices from 2019 onward |
| NFR-05 | Firebase services should auto-recover within 60 seconds |
| NFR-06 | App must remain usable in low bandwidth conditions |

**5. External Interfaces**

| **Interface** | **Description** |
| --- | --- |
| Firebase Auth | Manages secure user authentication |
| Firebase Firestore | Used for real-time logs and cloud data backup |
| Spring Boot API | Backend for all data operations and REST endpoints |
| File Export Utility | Generates PDF and Excel reports |
| Local Storage | Caches temporary data for offline-friendly use |