

Winter  
2026

# Software Requirements Specification

<SMART STOCK>

Parsa Tahmasebi  
Jun Ho Jeon

Andy wei

Nikan Eidi

Ivan Castro

<SMART STOCK PEOPLE> | Group 4

# Executive Summary

## Background

## Description

Company Value Add

End-User Value Add

## Scope

What is Included

What is Not Included

## Justification

## Table of Contents

<b>EXECUTIVE SUMMARY.....</b>	<b>1</b>
<b>BACKGROUND.....</b>	<b>1</b>
<b>DESCRIPTION.....</b>	<b>1</b>
COMPANY VALUE ADD.....	1
END-USER VALUE ADD.....	1
<b>SCOPE.....</b>	<b>1</b>
WHAT IS INCLUDED.....	1
WHAT IS NOT INCLUDED.....	1
<b>JUSTIFICATION.....</b>	<b>1</b>
 <b>SECTION 1.....</b>	 <b>4</b>
<b>1.1 DOCUMENT AUTHORS.....</b>	<b>4</b>
<b>1.2 DOCUMENT REVISION HISTORY.....</b>	<b>4</b>
<b>1.3 DOCUMENT PURPOSE.....</b>	<b>5</b>
<b>1.4 AUDIENCE.....</b>	<b>5</b>
<b>1.5 GROUP AGREEMENT.....</b>	<b>5</b>
Team #.....	5
Project Title.....	5
Project Time Frame.....	5
Team Members.....	5
Team Leadership.....	5
Team Functions/Roles.....	5
Team Meetings.....	5
Team Problems.....	5
Team Commitment.....	5
 <b>SECTION 2.....</b>	 <b>6</b>
<b>2.1 PROJECT PROPOSAL.....</b>	<b>6</b>
2.1.1 PROJECT BACKGROUND.....	6
2.1.2 PROBLEM STATEMENT.....	6
2.1.3 PRODUCT VISION.....	6
<b>2.2 STAKEHOLDERS AND USERS.....</b>	<b>6</b>
<b>2.3 PROJECT SCOPE.....</b>	<b>6</b>
<b>2.4 SYSTEM RISKS.....</b>	<b>6</b>
<b>2.5 OPERATING ENVIRONMENT.....</b>	<b>6</b>
<b>2.6 FUNCTIONAL REQUIREMENTS.....</b>	<b>6</b>

<b>2.7 NONFUNCTIONAL REQUIREMENTS.....</b>	<b>6</b>
<b>2.8 UI/UX INTERFACE MOCK-UPS.....</b>	<b>6</b>
 <b>SECTION 3.....</b>	 <b>7</b>
<b>3.1 WORKFLOW DIAGRAMS.....</b>	<b>7</b>
<b>3.2 DATA MODELLING AND FLOW.....</b>	<b>7</b>
<b>3.3 USE CASE SCENARIOS.....</b>	<b>7</b>
<b>3.4 ACTIVITY DIAGRAMS.....</b>	<b>7</b>
<b>3.5 BUSINESS RULES.....</b>	<b>7</b>
 <b>SECTION 4 – DOMAIN CLASS.....</b>	 <b>8</b>
 <b>SECTION 5 – DATABASE.....</b>	 <b>8</b>
 <b>SECTION 6 – PROJECT MANAGEMENT.....</b>	 <b>8</b>
<b>6.1 WORK BREAKDOWN STRUCTURE.....</b>	<b>8</b>
<b>6.2 MILESTONES &amp; ACCEPTANCE CRITERIA.....</b>	<b>8</b>
 <b>SECTION 7 – PRODUCT BACKLOG &amp; IMPLEMENTATION SCHEDULE.....</b>	 <b>8</b>
 <b>SECTION 8 – CLIENT/FACULTY SIGN-OFF.....</b>	 <b>8</b>

# Section 1

## 1.1 Document Authors

- Parsa Tahmasebi
- Jun Ho Jeon
- Andy wei
- Nikan Eidi
- Ivan Castro

## 1.2 Document Revision History

WEEK	DATE	Revisions
1		●
2		●
3	Jan 25, 2026	<ul style="list-style-type: none"><li>● Section 1</li><li>● Section 2</li></ul>
4	Jan 28,2026	<ul style="list-style-type: none"><li>● Section 2.5, 2.3</li></ul>
5		●
6		●
7		●
8		●
9		●
11		●
12		●
13		●
14		●

## 1.2 Document Purpose

The Smart Stock development team utilizes this Software Requirements Specification as the primary technical reference to ensure all project objectives are met with precision. The document establishes a comprehensive framework for addressing the operational challenges of small scale restaurants through AI-driven demand forecasting and NLP-based assistance systems. By defining the operational boundaries of a cloud-native environment alongside a BYOD model, this SRS serves as the fundamental guide for Quality Assurance and system validation throughout the two semester development lifecycle. All technical resources are focused on the core objectives of reducing food waste and optimizing procurement efficiency, ensuring the project remains within scope while delivering high-quality, actionable results.

## 1.4 Audience

Our target audience would consist of small-scale restaurants, startup food businesses, and low budget independent owners who rely on manual or spreadsheet based inventory management due to limited budgets and technical resources for managing inventory. This includes business owners, managers, and staff handling inventory with limited technical experience, needing a simple, low-cost system that will help with food waste and prevent stock shortages. Smart Stock is designed to improve these businesses quality of life with AI-assisted inventory intelligence that was only accessible to larger restaurant chains.

## 1.5 Group Agreement

### Team 4

#### Project Title

Smart Stock

#### Project Time Frame

3 months and 2 weeks

#### Team Members

- Parsa Tahmasebi
- Jun Ho Jeon
- Andy wei
- Nikan Eidi
- Ivan Castro

#### Team Leadership

#### Team Functions/Roles

#### Team Meetings

#### Team Problems

### Team Commitment

The undersigned members agree to work together on the project until the end of the PRJ666 next Semester. They recognize that as a team and individually they are equally responsible for the quality of all deliverables.

Name	Date	Signature
Nikan Eidi	01/23/2026	NE
Parsa Tahmasebi	01/23/2026	PT
Jun Ho Jeon	01/23/2026	jun
Ivan Castro	01/23/2026	Ivan Castro
Andy Wei	01/23/2026	andy

## Section 2

### 2.1 Project Proposal

#### 2.1.1 Project Background

This project aims to improve the daily operations of small scale companies such as local restaurants in regards to inventory management. Unlike big corporations where the ordering of products has a systematic approach due to collaboration of multiple companies to keep the system going, smaller companies have to manually order and handle the supplies.

In order to improve the situation our group has decided to create a solution for this issue while keeping in mind the limitations of those smaller companies. However before explaining the whole idea we first need to take a look at why we are doing this in the first place. No matter the amount of precision or attention to details we as humans always make mistakes as the sheets might get damaged or any other reason which can result to loss of the inventory counts. However by inputting them into a well built system and having copies of the documents safely stored in different locations in daily/weekly basis we can prevent a lot of the issues from happening.

Further more to help with the process and make it even easier for employees and members of the company, we have thought of adding a touch of AI to the project which will be focused on making the data more understandable and process of ordering easier.

#### 2.1.2 Problem Statement

The primary challenge small-scale restaurants faced was the absence of accessibility of automated infrastructure for inventory management. In an industry where **profit margins** are **notoriously thin**, the **lack of precision in tracking materials** such as food items can cause **financial** and **operational burden**. Currently, these most of these small businesses operate within a “**technological gap**”, being forced to decide between primitive manual methods and expensive enterprise-level software that is both financially and functionally an issue.

The issue mostly lies on the manual tracking done by a human, and being human nature there would always be mistakes somewhere. Most restaurant startups use paper logs, physical counts, and basic spreadsheets to manage their stock. These methods can always be flawed, being **time consuming** and highly susceptible to **human error**, especially when performed by a staff during high pressure shifts. This inaccuracy could lead to issues such as **overstocking** or **sudden stockouts** which **constricts** restaurants from gaining that more revenue.

The impact of this problem is most severe for independent cafes and startup food entrepreneurs. Because these owners often manage every aspect of the business themselves, from cooking to accounting, this usually leads to the **lack of time** to maintain complex manual logs. Every hour taken from manually auditing a spreadsheet is an hour taken away from more business growth. Especially with how **poor inventory management** becomes a “silent killer” of profitability which then leads to unnecessary expenses that small businesses simply cannot take.

### 2.1.3 Product Vision

Our vision for our product Smart Stock is to develop a cloud-based, AI assisted inventory management system designed specifically to the needs of small scale restaurants and businesses. As a group, we aim to provide a practical, realistic and accessible solution that replaces unreliable manual tracking methods with a platform that delivers accurate and real time insights.

Smart Stock will utilize historical inventory and sales data to perform “smart” demand forecasting, allowing businesses to anticipate stock requirements, minimize food waste, and avoid possible shortages. We aim to incorporate an NLP based assistant that allows users to access information and receive recommendations through natural language queries, making the whole system intuitive for users no matter their technical experience or level.

Ultimately, we envision Smart Stock as an affordable, scalable, and easy to deploy solution to help small businesses and equip them with actionable insights, inventory workflows, reduced waste and sustainable operations.

## 2.2 Stakeholders and Users

Stakeholder Name	Project Role	Internal/External	Engagement Level	Level of Interest	Level of Power
Development Team (Parsa, Jun Ho, Andy, Nikan, Ivan)	System Architects & Developers	Internal	Leading	High	High

Small-Scale Restaurant Owners	Primary Target Audience & Decision Makers	External	Supportive	High	High
Restaurant Managers/Staff	End-Users (Daily Inventory Tracking)	External	Consulted	High	Medium
Course Faculty	Project Evaluators & Supervisors	External	Leading	High	High
Local Food Suppliers	Information Sources for Pricing Comparison	External	Neutral	Low	Low
Cloud/AI Service Providers (AWS, OpenAI)	Infrastructure & API Providers	External	Neutral	Low	Medium

## 2.3 Project Scope

Smart Stock is designed to provide small businesses with an affordable system that offers real time inventory tracking, allowing users to monitor ingredient quantities, receive low-stock and expiry alerts, and maintain accurate digital records while all being processed through AI helping the users understand the data and process of ordering a lot easier, the system will also offer a module that lets users compare ingredient prices from local suppliers and helps the user make financially better decisions. Additionally, Smart Stock is provided with an interactive web-based dashboard, automated reporting tools, and an AI assistant that helps non-technical users navigate through the system efficiently.

However, Smart Stock systems do not provide the business with any functions for processing payments, managing payroll, or handling employee scheduling. The system cannot automate supplier ordering or making final financial transactions for the user, Instead it will provide recommendations and insights on the best course of action for the business, leaving the decision up to the user to decide to follow the recommendations or not. The system will also not support large enterprise-level restaurant chains, as our focus is more on small businesses. Finally, Smart Stock cannot guarantee perfect predictions as AI is dependent on data quality and trends, and is intended to assist, not replace human judgement.

## 2.4 System Risks

- **AI Prediction Accuracy:** The machine learning model might provide incorrect data, leading users to miss out on better procurement deals.
- **API Availability:** Dependency on external supplier APIs is a risk, as access could be blocked or changed, disrupting our daily inventory and price checks.
- **Data Quality Issues:** Incomplete or messy historical pricing data can directly lower the reliability of our AI driven insights.
- **Market Data Drift:** Sudden market shifts or seasonal variations may render the old data useless, leaving the algorithm with mere facades of predictive value.
- **Legal and Privacy Compliance:** Handling vendor pricing or user data carries a risk of accidental non compliance with scraping laws or privacy regulations.
- **System Infrastructure Load:** High frequency web scraping for pricing can put heavy stress on our servers and degrade overall platform performance.
- **API Rate Limiting:** Making too many requests may hit vendor limits or lead to higher operational costs and service interruptions.
- **User Trust and Notifications:** If alerts are delayed or not accurate, users may lose confidence in the app and seek cheaper prices elsewhere.

## 2.5 Operating Environment

Users can access Smart Stock on the web browser when they are given access to the platform. This makes the system cross-platform and compatible with handheld devices, tablets, and desktop regardless of the operating system. The platform relies on a cloud infrastructure to manage the database storage and ensure scalability for small businesses. We will also integrate AI to power the “Smart Assistant” feature. A stable internet connection is required to ensure real-time data syncing and AI functionality.

## 2.6 Functional Requirements

### 2.6.1 User Story Interviews

### 2.6.2 Core Functionalities

#### Admin Dashboard

- Be able to create employee accounts
- Basic administration access to system

#### Employee Accounts

- Access to inventory availability
- Limited Access to ordering and modifying inventory values

#### AI Inventory management assistant

- Analyze the popularity of certain menu over a specific periods of time

#### Notifications

- Notify the users of the items when the availability count goes below the “minimum” threshold
- Notify the business every month for inventory check across branches

### Data Integration and Storage

- List of items owned by business solely accessible to the admins to add or modify
- List of items categorized as merchandise available in inventory
- Real time information updated

## 2.7 Nonfunctional Requirements

### Usability(UI)

- Responsive Layout that can easily adapt to Mobile, Tablet and Desktop environments
- Reduce bugs and crashes to ensure stability especially during business hours
- Clean interface to facilitate quick inventory counts on mobile devices

### Security

- Strict data isolation to make sure some inventory records are accessible and editable only to business account owners
- Business data information protected through secure authentication methods

### Performance

- Real time data syncing with minimal delay to prevent item discrepancies
- Responsive application interface with immediate responses after a user input

### Maintainability

- Codebase will follow modular design principles to support future feature expansion
- Support easy integration of new AI models or API updates

## 2.8 UI/UX Interface Mock-ups

## Section 3

### 3.1 Data Flow Diagrams

### 3.2 User Stories and related Use Case Scenarios

### 3.3 Activity Diagrams

### 3.4 Business Rules

Business Rule #	Description	Activity Diagram	Related UCS	UI Mock-up
BR1		AD1	UC1	UI 2.7.2
BR2		AD2	UC2	UI 2.7.3
BR3		AD3	UC3	UI 2.7.4
BR4		AD3	UC3	UI 2.7.4
BR5		AD5	UC4	UI 2.7.6
BR6		AD6	UC5	UI 2.7.6
BR7		AD7	UC6	UI 2.7.7
BR8		AD8	UC7	UI 2.7.8
BR9		AD8	UC7	UI 2.7.8
BR10		AD8	UC7	UI 2.7.8
BR11		AD8	UC7	UI 2.7.8
BR12		AD8	UC7	UI 2.7.8
BR13		AD9	UC8	UI 2.7.9
BR14		AD9	UC8	UI 2.7.9
BR15		AD9	UC8	UI 2.7.9
BR16		AD9	UC8	UI 2.7.9
BR17		AD10	AD9	UI 2.7.9
BR18		AD10	AD9	UI 2.7.9
BR19		AD10	AD9	UI 2.7.9
BR20		AD11	UC10	UI 2.7.10
BR21		AD11	UC10	UI 2.7.11
BR22		AD11	UC10	UI 2.7.11
BR23		AD12	UC11	UI 2.7.10
BR24		AD13	UC12	UI 2.7.12

## Section 4 – Domain Class

## Section 5 – Database

## Section 6 – Project Management

### 6.1 Work Breakdown Structure

### 6.2 Milestones & Acceptance Criteria

## Section 7 – Product Backlog & Implementation Schedule

## Section 8 – Client/Faculty Sign-off