# Software Requirements Specification

# PRJ566 – Winter 2025

# PRJ566 – Team No:

Team 06

# Name of Project:   Restaurant Management Application – ChowHub

# Project Leader:

Mostafa Hasanalipourshahrabadi

**Last updated:**

**Team Members:**

Tingchen Tsao - 107253239

Lily Huang - 180923211

Furkan Bas - 121540215

Saad Ghori – 155972227

# TABLE OF CONTENTS

1. **Introduction/Overview - Document Information**
   1. **Document Authors**
   2. **Revision History**
   3. **Document Conventions**
   4. **Document Purpose**
   5. **Intended Audience**
   6. **Group Agreement**
2. **Project Overview**
   1. **Project Proposal**
   2. **Stakeholders and Users**
   3. **Project Scope**
   4. **System risks**
   5. **Operating Environment**
   6. **Functional Requirements**
   7. **Nonfunctional Requirements**
   8. **UI/UXD Interface Mockups**
3. **Process & Data Modeling**
   1. **UML Modeling: DFDs & Activity Diagrams**
   2. **Use Case Specification** 
      1. **Business Rules**
      2. **System Use Case Diagrams**
      3. **Use Case Description Tables**
4. **Domain Class Diagram**
5. **Database (Select either 5.1 or 5.2)**
   1. **RDBMS Artifacts**
      1. Scripts to create, populate, delete tables
      2. Data Dictionary
   2. **NoSQL Artifacts**
6. **Work breakdown Structure (WBS)**
7. **Milestones & Acceptance Criteria**
8. **Implementation Schedule (Agile/Waterfall)**
9. **Client / Faculty Sign-off**

# 1 - Introduction/Overview - Document Information

## 1.1 Document Authors

Mostafa Hasanalipourshahrabadi, Tingchen Tsao, Lily Huang, Furkan Bas, Saad Ghori

## 1.2 Revision History

|  |  |
| --- | --- |
| Week 03 | 1. Introduction/Overview  1.1 Document Authors  1.2 Revision History  1.3 Document conventions  1.4 Document purpose  1.5 Intended audience  1.6 Group agreement  2.1 Project proposal |
| Week 04 | 2.2 Stakeholders and Users  2.3 Project Scope  2.4 System Risks  2.5 Operating Environment |
| Week 05 |  |
| Week 06 |  |
| Week 07 |  |
| Week 08 |  |
| Week 09 |  |
| Week 10 |  |
| Week 11 |  |
| Final |  |

## 1.3 Document Conventions

Any text in red indicates an exception or error.

Any text in blue is in-progress.

Any text highlighted in yellow is an important point.

Any text in green was recently added.

Any text *italicized* represents definitions.

Any text with ~~strike-through~~ is deleted.

## 1.4 Document Purpose

The purpose of this Software Requirements Specification (SRS) document is to clearly outline the requirements for developing the ChowHub Restaurant Management Application. It provides a detailed description of the system’s features, goals, and limitations, ensuring everyone involved in the project understands what needs to be built and how it will work.

This document is meant to guide the development team as they design, build, and test the application, while helping the project manager plan resources and timelines. It also ensures that restaurant stakeholders know what to expect from the application and can refer to it for any future updates or improvements. By clearly listing the project’s requirements, this SRS helps avoid misunderstandings, reduces risks, and makes it easier for everyone to work together effectively.

## 1.5 Intended Audience

The intended audience for this Software Requirements Specification (SRS) document includes everyone involved in the project. The development team—software developers, testers, and designers—will use it to build and test the system according to the requirements. The project manager will refer to it to manage resources, track timelines, and ensure the project stays on schedule. Restaurant owners or managers, as clients, may review the document to ensure the system meets their needs. The quality assurance team will use the requirements to check that the system works correctly, is user-friendly, and reliable. System integrators, who connect ChowHub to tools like POS platforms, will rely on it to ensure smooth integration. Technical writers might use the SRS to create user manuals or training materials. Lastly, future teams or vendors working on updates or expansions can refer to the document to understand the original system requirements and keep everything consistent. This SRS helps all parties work together effectively and stay aligned.

## 1.6 Group Agreement

**TEAM AGREEMENT**

**Team #:** 06

**Project Title:** ChowHub

**Project Time Frame:** January 6, 2025 – August 15, 2025

**Team Members:**

Furkan Bas

Lily Huang

Saad Ghori

Tingchen Tsao

**Team Leadership:** Mostafa Hasanalipourshahrabadi

**Team Functions:**

* Attend all scheduled online meetings on the team's channel on MS Teams.
* Communicate and collaborate through MS Teams chat or in the issues/discussion section of GitHub.
* Complete all assigned issues for the deliverables of the project by the designated soft deadline.
* Communicate in advance on the MS Teams channel or chat if there are any challenges or difficulties with any assigned tasks.
* Assist and support other team members to foster a collaborative and stress-free environment within the team.
* Submit the deliverable for the week by the soft deadline of 12:00 PM every Sunday. The group will review and finalize the work by 6:00 PM every Sunday.
* The group leader will submit the deliverable and review the GitHub repository by 9:00 PM every Sunday.
* If any member wishes to make a change after the leader has submitted, they must inform the team and obtain the leader's approval before making the change.

**Team Meetings:**

* The team will meet three times a week - Tuesday at 7:00 PM, Wednesday at 1:25 PM, and Sunday or Saturday at 1:00 PM, depending on team availability.
* The meetings will be on MS Teams, Team 06 channel.
* On Wednesdays, the team will meet with the professor to discuss and review the progress of the project. The other two meetings will be held within the team to discuss upcoming tasks, address any issues or concerns, and plan for the next steps.

**Team Problems:**

* **Conflict Resolution:** If conflicts arise, the team will address them through open discussion and mediation among its members and the leader. Should the issue remain unresolved, the matter will be escalated to the course instructor for further guidance.
* **Workload Distribution:** Tasks will be distributed equitably among all team members. If a team member is having trouble with their workload, they are encouraged to communicate with the team on the Teams’ channel, allowing others to collaborate and help.
* **No-Show Policy:** If a team member fails to complete the tasks assigned to them and does not communicate or attend meetings, the remaining team members have the authority to recommend their exclusion from the team, after notifying the course instructor.

**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 responsible for the quality of all deliverables.**

**Name Date**

|  |  |
| --- | --- |
| Saad Ghori | 01-26-2025 |
| Mostafa Hasanalipourshahrabadi | 01-26- |
| Tingchen Tsao | 01-26-2025 |
| Lily Huang | 01-26-2025 |
| Furkan Bas | 01-26-2025 |

# ShapeShapeShapeShapeShape2 - Project Overview

## 2.1 Project Proposal

**Project Background**

The restaurant industry has long faced operational challenges stemming from inefficient inventory management, fragmented systems, and difficulty in adapting to dynamic customer demands. Traditional approaches to managing inventory, employee shifts, menu operations, and supplier coordination often result in waste, overstocking, ingredient shortages, order cancellations, and dissatisfied customers. Additionally, restaurants often lack real-time insights into sales performance and ingredient usage, making it harder to optimize operations or make data-driven decisions. This project seeks to address these inefficiencies through a streamlined, centralized solution—ChowHub. ChowHub integrates inventory management, automated menu updates, advanced sales analysis, supplier coordination, employee management, shift tracking, and secure POS integration, tailored specifically for the fast-paced, small-to-mid-sized restaurant environment.

**Problem Statement**

|  |  |
| --- | --- |
| The Problem of: | Inefficient inventory tracking, ingredient shortages, overstocking, waste, fragmented operational systems, and limited data-driven decision-making. |
| Affects: | Restaurant owners, managers, staff (including waitstaff and kitchen staff), and ultimately, customers. |
| The impact of which is: | Increased operational costs, loss of revenue due to waste or missed sales opportunities, reduced employee productivity, lower customer satisfaction, and difficulty in scaling operations effectively. |
| A successful solution would: | * Reduce waste through real-time inventory tracking and customizable low-stock alerts. * Automate menu updates based on ingredient availability to prevent unfulfilled orders. * Streamline employee and shift management, reducing manual administrative tasks. * Improve operational efficiency by integrating with existing POS systems to avoid costly replacements. * Enhance customer satisfaction by ensuring timely service, consistent food quality, and minimal operational disruptions. * Provide actionable analytics, such as identifying top-selling menu items, peak sales times, and cost-saving opportunities. * Prioritize security and privacy by implementing strong encryption standards and enabling on-premises data storage. |

**Product Vision**

|  |  |
| --- | --- |
| For | Small-to-mid-sized restaurant owners or managers who need a cost-effective and efficient way to handle inventory, supplier coordination, menu operations, and employee management. |
| Who | Require a streamlined, integrated system to track inventory, manage suppliers, assign shifts, and optimize backend operations while improving customer experience. |
| The Product Name | ChowHub |
| That | Provides real-time inventory tracking, automated menu management, advanced analytics, shift management, and secure user authentication—all seamlessly integrated with POS systems. |
| Unlike | Traditional systems that focus on singular aspects of restaurant management, require costly replacements, or lack customization, |
| Our product | Offers a centralized, scalable, and secure solution that enhances operational efficiency, reduces waste, improves profitability, and prioritizes data privacy—all without disrupting existing workflows. |

## 2.2 Stakeholders and Users

|  |  |  |
| --- | --- | --- |
| **Stakeholder Name/Identifier** | **Category** | **Role/Responsibilities** |
| **CEO (Chief Executive Officer)** | Administration, Sponsor | Oversees the overall project and ensures resources are allocated; provides executive support and vision. |
| **Construction Manager and Scheduler** | Administration, User | Needs accurate up-to-date information for costing, scheduling, and ensuring smooth execution of the project details. |
| **Administrative Assistant** | User | Assists with project coordination and document management, ensuring smooth communication and information flow. |
| **Schedulers** | User | Responsible for organizing and maintaining the project timeline and coordinating task schedules. |
| **Cost Accountant** | User | Tracks project costs, manages budgets, and ensures financial resources are allocated appropriately. |
| **Project Leader** | Developers | Oversees the development team, ensures project tasks are on track, and manages technical requirements and milestones. |
| **Developers** | Developers | Work on coding, system design, and implementation based on project requirements, collaborating with other team members. |
| **Restaurant Owners/Managers** | End Users, Client | Utilize the ChowHub platform for inventory, menu, and employee management. Ensure the system meets the operational needs of their restaurant. |
| **Quality Assurance (QA) Team** | Developers | Responsible for testing ChowHub to ensure it works as intended, identifying bugs, and verifying that features are functional before release. |
| **System Integrators** | Developers | Work on integrating ChowHub with existing systems such as POS platforms and supplier databases, ensuring compatibility and smooth operation. |
| **End Users (Restaurant Staff)** | End Users | Use ChowHub for tasks like placing orders, tracking ingredients, and managing shift schedules, enhancing the overall restaurant workflow. |
| **Technical Writers** | Documentation | Develop user manuals, training materials, and support documents to guide end-users in effectively using ChowHub. |

## 2.3 Project Scope

## 2.4 System Risks

|  |  |
| --- | --- |
| **Risk** | **Response** |
| Trying to integrate with the POS system may pose an integration risk with not being able to properly integrate into their system. | ChowHub’s features are standalone and don’t need integration into the POS system, but will greatly benefit the ease of use if integrated. |
| There may be compliance risk with storing employee data. | The system wouldn’t deal with sensitive data such as employee social insurance numbers or pay and would just deal with performance associated with employee number. |
| There may be technical risks with not having access to certain technologies such as a POS system or cashier terminal. | We can create and have access to mock POS systems in order get an idea of how we will interface with the POS system or cashier terminal. |
| **There may be risks associated with downtime of their system since if they aren’t able to access their POS then they won’t be able to do orders and lose revenue.** | We can integrate unit testing and a variety of other tests in order to ensure compatibility and integrate on off peak hours of the restaurant. |
| **Resource and time risk. We may not have enough time to dedicate to ChowHub which risks the incompletion of it.** | We can move the scope so that we can complete ChowHub’s creation and then implement the features intended for it later. |

## 2.5 Operating Environment

ChowHub is designed to operate in a restaurant management environment, ensuring seamless integration with existing workflows. The application will function across multiple platforms and environments, providing flexibility for restaurant owners and staff.

**Hardware Requirements**  
- Client Devices: Computers, tablets, or smartphones used by restaurant owners, managers, and staff.  
- POS System Compatibility: Integrate with existing POS terminals without requiring hardware replacement.  
- Server Requirements: Cloud-based (or on-premise) servers capable of handling database storage, analytics processing, and real-time updates.  
**Software Requirements**  
- Operating Systems: Windows, macOS, iOS, and Android.  
- Web Browser Support: The platform will run on modern browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.  
- Database Management: MongoDB for efficient data storage and retrieval.  
**Network Requirements**  
- Internet Connection: Required for real-time updates, cloud synchronization, and remote access.  
- Local Network Support: Restaurants with on-premise servers can operate within a local network in case of internet disruptions.  
**Security Considerations**  
- Data Encryption: Strong encryption will protect sensitive information such as payroll and sales data.  
- Access Control: Role-based access ensures that only authorized users can modify critical data.  
- Data Storage: Sensitive data can be stored locally within the restaurant to enhance privacy and security.

## 2.6 Functional Requirements

## 

## 2.7 Nonfunctional Requirements

Operational, Performance & Security Requirements

## 2.8 UI/UXD Interface Mock-ups

# Process and Data Modeling

## **3.1 UML/DFD Modeling and Data Modeling**

### Activity Diagrams and Data Flow diagram

## **3.2 Business Rules**

|  |  |  |
| --- | --- | --- |
| Business Rule Number | Business Rule Description | Related UC |
| BR01 | User must provide a username, email and password to register for the app. | UC01 |
| BR02 | Post length can be no longer than 300 characters | UC02 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## **3.3 Use Case Specifications with corresponding interface mockups:**

**Each use case needs to have the following:**

1- **Business Rules.**

**2- System Use Case Diagrams.**

**3- Use Case Descriptions.**

**4- Corresponding Mockups**

# Domain Class Diagram

# Database

# Work Breakdown Structure (WBS)

## 

## Work Breakdown Structure

Sample WBS:

Diagram

Description automatically generated

# Milestones and Acceptance Criteria

* 1. Milestone one

Definition

Acceptance Criteria

* …
* ….
* ….
  1. Milestone Two
  2. Milestone Three
  3. ..
  4. …
  5. …
  6. ..
  7. ..
  8. ...etc.

# Implementation Schedule

Implementation Schedule using MS Project (Waterfall)

OR

Product Backlog (Agile-Scrum)

# Client / Faculty Sign-off

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

X .

Name of Client/Rep/Professor