BETHLEHEM UNIVERSITY
جامعة بيت لحم

Software Requirements Specification

for

Chef’s Cabinet - Food Management and Tracking System

December 11, 2023

Prepared By

Angela Salem

Aya Altamimi

Daniella Anastas

Submitted in partial fulfillment Of the requirements of SWER311 - Software Requirements Engineering

Software Engineering

Outline

1. Introduction

1.0 overview ——————————————— 4

1.1 Purpose ——————————————— 5

1.2 Scope ——————————————— 5

1.3 Intended Audience ——————————————— 7

2. Overall Description——————————————— 8

2.1 Product Perspective ——————————————— 8

2.2 Product Functions——————————————— 9

2.3 User Classes and Characteristics——————————————— 10

2.4 Operating Environment ——————————————— 11

2.5 Design and Implementation ——————————————— 12

2.6 User Documentation ——————————————— 13

2.7 Assumptions and Dependencies ——————————————— 14

3. System Features ——————————————— 15

3.1 Real-time Inventory Tracking ——————————————— 16

3.2 Automatic Inventory Deduction from Meal Entry ——————————————— 17

3.3 Machine Learning for Predictive Analysis ——————————————— 18

3.4 Inventory Exchange and Trade System ——————————————— 20

3.5 Surplus Redistribution and Reward System ——————————————— 21

4. External Interface Requirements —--- 24

4.1 User Interfaces —-- 25

4.3 Software Interfaces —--- 26

4.4 Communication Interfaces —--- 26

5. Non-functional Requirements —-- 27

5.1 Performance Requirements —-- 28

5.2 Security Requirements —-- 29

5.3 Reliability and Availability —--- 30

5.4 Maintainability —-- 31

5.5 Portability —-- 32

6. Other Requirements —--- 34

6.1 Legal and Regulatory Requirements —--- 34

6.2 Other Requirements —--- 35

7. Appendices —-- 36

7.1 Glossary

1. Introduction

1.0 Project Overview

"Chef's Cabinet" emerges as an essential solution in the realm of restaurants, where the significant costs associated with food waste prompt a strategic approach. Recognizing the need to enhance cost efficiency and reduce losses, the system provides an accurate alternative to the traditional, time-consuming methods of recording waste on paper or Excel. By monitoring and analyzing the highest-cost areas, the Chef's Cabinet becomes a crucial tool for determining inconsistencies and taking the right approach to reduce waste. This not only leads to yearly savings but also makes the restaurant kinder to the environment by reducing its carbon footprint. In addition to waste tracking; the system serves as a comprehensive tool for food consumption assessment, enabling the recording of production and serving quantities. To put it simply, Chef's Cabinet is a game-changing tool that encourages sustainable behaviors in restaurants while streamlining their productivity.

1.1 Purpose

Chef's Cabinet project aims to achieve three main goals:

1. minimizing food waste that results in public diners, therefore less environmental pollution and more social impact. see 3.1 Feature 1
2. enhance inventory management of public restaurants
3. increase restaurants profits. see 3.4 Feature 5

1.2 Scope

Chef's Cabinet project gives an innovative software solution that exceeds traditional methods to food waste management in restaurants. The deliverables include a user-friendly interface that provides real-time insights

into overall food waste metrics, financial values, and environmental impacts. The system will also offer a predictive element that will enable restaurant staff to make wise decisions to minimize waste.

Included functionalities cover:

1- Inventory Management: Tracking, updating, and suggesting actions on inventory items, including expiration date alerts and machine learning-based optimization.

2- Surplus Management: Facilitating surplus item donation tracking, verification, and incentivizing through reward systems.

3- Integration with External Systems: Connecting to billing systems, machine learning models, and government verification portals.

4- User Interaction Interfaces: User-friendly interfaces for inventory management, surplus tracking, and data input functionalities.

The Excluded functionalities :

1- External Logistics: Operations beyond the software's domain, such as the actual transportation of surplus items or integration with external logistics.

2- Physical hardware requirements: for instance, the devices with cameras for image capture.

3- physical hardware development : the system will not encompass the creation of new kitchen appliances, electronic sensors for inventory monitoring, or modifications to existing cooking equipment.

The project acceptance criteria ensure that the project meets the necessary requirements and aligns with the goals and objectives of the stakeholders involved. This includes approval of the project's budget by the user, ensuring it aligns with their expectations. The user-friendly interface ensures that restaurant employees can effortlessly navigate through the system, making it a practical tool in their daily tasks. Finally, the project's alignment with sustainability goals can be evaluated by reducing greenhouse gas emissions.

1.3. Intended Audience

The SRS document for Chef's Cabinet is created to accommodate a wide set of stakeholders who are involved in the development, implementation, and administration of the system. The identified audience includes:

Audience	SRS Document Use
Software Developers	Understand the coding rules, system architecture, and technical specifications before implementing them.
Project Managers & Team Leads	Create project plans, assign tasks, and track progress.
Food Waste Management Experts	They help in project organization by applying their expertise and industry best practices to successfully reduce food waste.
Marketing and Sales Teams	Gather information on market trends, competitive products, and customer preferences, ensuring that our system meets market and customer demands.

2. Overall Description

2.1 Product Perspective

Chef's Cabinet stands as a clever solution redefining food inventory management, ready to smooth processes and encourage waste reduction in the restaurant industry.

The system consists of two integral modules:

- *Mobile Application for Users:*

  The mobile application lets restaurant staff and individuals input grocery lists effortlessly, track food inventory, and receive notifications about close expiration dates.

- *Web Portal for Restaurant Managers:*

  The officials can get full analytics and insights using this portal. Managers can follow food consumption behaviors, categorize waste data, and analyze them over particular time periods.

The product will integrate with production planning systems:

Chef's Cabinet is integrated with production planning systems. The system will retrieve the daily menu information from those production planning systems. This allows Chef’s Cabinet to seamlessly incorporate menu details, record statistics on wasted food, and analyze the waste for each item. This integration improves the efficiency of production planning by benefiting from the insights gathered by Chef’s Cabinet without the need for manual data transfers.

2.2 Product Functions

Chef’s Cabinet is characterized with a range of functions aimed at optimizing food waste management and enhancing operational efficiency within the restaurant environment.

- **Real-time Food Inventory Monitoring:**

  The system enables users to monitor their food inventory in real-time, providing a full overview of available items and their quantities.

- **Waste Tracking and Analysis:**

  The system records and analyzes statistics on the amount of wasted food, categorizing the waste for each item and providing valuable insights for waste reduction strategies.

- **Smart Demand Prediction:**

  AI algorithms analyze consumption patterns, user preferences, and external factors to predict future demands accurately. This helps in enhancing inventory management, minimizing overstocking or understocking, and reducing the risk of perishable items going to waste.

2.3. User Classes and Characteristics

User	Class	Description
Restaurant Manager	Management Role	Individuals responsible for managing restaurant operations and having a high knowledge of computers to make plans, monitor system, and interpret data.
Kitchen Staff	Operations Team	They'll be recording the amount of consuming, and monitoring food wastage. They're required to have computer skills for system interaction and data entry.
End Users	Restaurants Guests	Customers who use the mobile app are expected to have an intermediate level of computer knowledge for using the app features, inputting data, and receiving notifications.
System Administrators	Technical Management	Overseeing system integration, maintenance, and troubleshooting, system administrators require an advanced level of technical expertise.
AI Specialists	Machine Learning Experts	They maintain AI models, having an expert level of knowledge in machine learning, algorithms, and system optimization.

2.4. Operating Environment

Chef's Cabinet operates in a specified environment to ensure optimal performance and compatibility.

- **Server-side Components:**

    ➢ The server-side components of Chef's Cabinet must operate within a Linux operating system environment. It is optimized for compatibility with distributions such as Ubuntu Server 18.04+ and CentOS 7+.

    ➢ Hardware specifications for the server include a minimum of 8GB RAM, a quad-core processor, and at least 100GB of storage space.

- **Client-side Components:**

➢ The client-side components are expected to function seamlessly in common web browser environments using Secure Sockets Layer/ Transport Layer Security cryptographic protocols. The minimum encryption level required is 128 bits. The supported browsers include:

  - Apple Safari 7+
  - Google Chrome 44+
  - Microsoft Internet Explorer 10+
  - Mozilla Firefox 40+

➢ Client devices should have a minimum of 4GB RAM, a dual-core processor, and a screen resolution of 1024x768 pixels for optimal user experience.

2.5. Design and Implementation Constraints

Chef's Cabinet operates within certain constraints to ensure a effective design and implementation. These constraints are essential for the proper functioning of the software system.

- The system depends on a strong data store to manage information related to food inventory, wastage, and user interactions. All Chef's Cabinet functionalities are accessible via a RESTful API to enable easy communication and integration with other systems.

- Chef's Cabinet requires continuous operation to support immediate food waste tracking. The tracking system is designed to function 24/7, ensuring that users can access and update information at any time.

- Chef's Cabinet should be designed to accommodate with increasing user data and system usage. The choice of cloud infrastructure, such as AWS or Azure, and the implementation of containerization using Docker, can ease scalability and efficient resource use.

- About image recognition integration, the system has a maximum processing speed for image analysis and categorization. The system may experience processing delays if this limit is exceeded,, which would negatively affect the user experience. To prevent disruption with the functionality of image recognition, usage must accommodate to certain restrictions.

2.6 User Documentation

The user documentation for Chef's Cabinet will include comprehensive guides aimed at facilitating a seamless experience for end-users. This documentation will consist of:

1. User Manuals:

   Manuals will guide users on how to navigate Chef's Cabinet, record food items, use the image recognition feature, and explain waste analysis reports. The manuals will be structured for easy understanding, with step-by-step instructions and pictures.

2. FAQs & Troubleshooting Guides:

   FAQs section will be housed on the website , it will answer common questions and provide quick solutions to user concerns. Troubleshooting guides will offer help in resolving issues users may encounter.

2.7. Assumptions and Dependencies

We, as a project team, have identified key assumptions and dependencies that shape the success and functionality of the software system.

1. ***Assumptions***

   AI Integration:

   Assumption: Chef's Cabinet smart features, like AI waste analysis, rely on good image recognition. This means the AI can understand and help reduce waste properly.

   Rationale: To really cut down on waste, the AI needs to see and understand what's being thrown away. If the image recognition is reliable, the waste reduction strategies will work better.

2. ***Dependencies***

   Billing System Integration:

   Chef's Cabinet depends on data from the billing system to assign financial values to wasted food. This integration ensures precise calculations of the economic impact of food wastage on the restaurant's finances.

3. System Features

3.1 Feature 1:

Real-time Inventory Tracking

Description:

This feature enables the real-time tracking of inventory within the restaurant's storage system, providing accurate and up-to-date information on available stock levels.

Functionality:

Real-time Updates: The system captures incoming and outgoing inventory data, ensuring instantaneous updates of quantities and statuses.

Identification and Location: Users can easily identify available items, their quantities, and precise storage locations within the facility.

Expiration Date Monitoring: The system actively monitors expiration dates, triggering alerts to users when inventory quantities reach predefined thresholds or when items approach their expiration dates.

User Interaction:

- Users access the inventory tracking feature through a user-friendly interface, viewing categorized lists of available items, their quantities, and details. They can perform actions such as adding new items, updating quantities, and setting low-stock alerts.

- ***Notifications :*** Alerts and notifications are delivered to users through the application interface, email, or SMS as his set preferences and based on predefined urgency levels (critical, informational, etc.).

3.2 Feature 2:

Automatic Inventory Deduction from Meal Entry

Description:

This feature establishes a connection between the billing system and the inventory management system to facilitate the deduction of inventory items as they are used in orders.

Functionality:

1. Integration between Systems: Establishes seamless integration between the billing and inventory systems through API keys or system access points. And assists restaurant staff with seamless integration through various support channels, including IT guidance, instructional materials, or step-by-step instructions.

2. Real-time Updates:Retrieves order details, such as date and meal, from the connected billing system when an order is assigned. Automatically adjusts the inventory by deducting pre-entered ingredient quantities associated with the specific order.

User Interaction:

One-time Entry of every meal ingredient Quantities:

- Users input quantities of ingredients used in each meal during the system's setup phase or upon implementation.
- Allows for modifications in the system when there are changes to the menu or alterations in ingredient quantities, ensuring accurate inventory deductions.

Integration Guidance:

- Guides restaurant staff through the integration process using clear instructions, video resources, or interactive tutorials.

3.3 Feature 3:

Machine Learning for Predictive Analysis

Description:

This feature utilizes machine learning algorithms to predict potential food spoilage based on historical data, trends, and expiration dates. The objective is to provide users with proactive recommendations and insights to prevent waste.

Functionality:

- ***Data Analysis:*** The system analyzes comprehensive historical data, including item purchase dates, storage conditions, seasonal trends, and item-specific attributes, to train and refine machine learning models.
- ***Prediction Models:*** Machine learning models employ regression and pattern recognition algorithms to predict the likelihood of items spoiling within a given timeframe. These models continuously learn and adapt based on new data inputs.
- ***Recommendations:*** Offers users actionable recommendations tailored to specific scenarios, such as:
    - Seasonal Recommendations: During tourism seasons, the system suggests optimized food quantities to cater to increased demand, minimizing waste and ensuring adequate supply.
    - Usage Suggestions: Recommends using specific items in meal preparation or adjusting stock levels based on predicted spoilage likelihood.

Constraints:

- Accuracy of predictions might vary based on the quality and completeness of historical data.
- The system's effectiveness relies on continuous learning and refinement of prediction models based on real-time data.

3.4 Feature 4:

Inventory Exchange and Trade System

Description:

Enables restaurants to exchange or sell surplus inventory to other establishments, fostering a mutually beneficial system for surplus utilization.

Functionality:

Matching Surplus with Demand:

- Matches restaurants holding surplus items with establishments in need of those specific items.
- Facilitates the trade or sale of surplus inventory between interested parties.

User Interaction:

- **Listing Surplus Inventory:**

  Allows restaurants to list available surplus inventory items within the system.

- **Negotiation and Transaction:**

  Facilitates communication, negotiation, and transaction processes between participating establishments.

3.4 Feature 5:

Surplus Redistribution and Reward System

Description:

The system incentivizes restaurants to donate surplus food by providing rewards and facilitating tax benefits. It encourages surplus food donations to support those in need while offering tangible benefits to contributing establishments.

Functionality:

Point-based Reward System:

- Establishes a points system rewarding restaurants based on the quantity and quality of donated surplus food items.
- Points are earned for each donation and contribute to the restaurant's reward status.

Surplus Donation Tracking:

- Enables restaurants to efficiently monitor surplus food availability within the system.
- Facilitates seamless reporting and documentation of surplus food items ready for donation.

Enhanced Surplus Donation Verification:

- Implements a verification process beyond image upload for surplus food donations.

- Designated government representatives physically verify and validate surplus donations.
- Real-time confirmation within the system ensures authenticity and compliance with donation standards.

Tax Exemptions and Benefits:

Automated Verification Process:

- Captured images of surplus food donations uploaded by restaurants are forwarded to the government's system.
- Enhanced verification includes physical validation by authorized personnel, ensuring adherence to donation criteria.

Points Calculation and Tax Exemption:

- Upon successful government verification, earned points are calculated and awarded to the restaurant within the system.
- Accumulated points contribute to the restaurant's eligibility for tax exemptions, aligning with government policies.

User Interaction:

Surplus Food Donation Capture:

- Restaurants capture surplus food donation images via the system's interface.
- Images serve as visual proof and are uploaded directly to the system for donation documentation.

Enhanced Verification Process:

- The system guides users through a verification process where authorized personnel confirm and authenticate surplus donations.
- Real-time verification status updates are provided within the system for transparency and audit purposes.

4. External Interface Requirements

4.1 User Interfaces

the system will have a view for manual data entry in case having and external accounting system doesn't apply. Moreover, it will facilitate data entry by using the machine learning. Therefor, it can predict what inventory will be needed for data entry and exempt the user from routine tasks. Yet, the user is always allowed to edit and reenter data.

Also, as we mentioned that the system will have a web app view, it is necessary to make data analysis view as useful to let the user see exactly, how much their business consumes inventory, how much produce it needs and whether they contribute enough to the food sharing system.

Another user interaction will be present when the user uses the reward system. This view shall allow clear representation of the users' contribution to prevent food waste, and also show them the available services they can benefit from as restaurants.

Lastly, a "Food Donation Form" view. This interface is for restaurants to input details of the surplus food available for donation. Fields for item name, quantity, expiration date, and any special instructions, with the option to specify dietary information (e.g., vegetarian, gluten-free).

4.2 Software Interfaces:

the system shall not directly interact with other systems in its environment, the only interaction is realized by exchanging mutual data mentioned in the section below.

4.3 Communication Interfaces:

As end users use other systems for accounting purposes, Chef's Cabinet may use some of the other system's database for mutual data. This data represents the amount of food in inventory, and the amount of food sold and their price. This data will be used as described in 3.3 Feature 3.

As for the protocols, the system will use basic HTML and FTP protocols to achieve the communication between other databases and the system database.

5. Non-functional Requirements

5.1 Performance Requirements

- **Response Time:**

  - ***User Interactions***: The system should respond promptly to user interactions within 2 seconds for routine actions such as adding items, updating quantities, and receiving alerts.

  - ***Machine Learning Analysis:*** Machine learning predictions and recommendations should be generated within 5 -7 seconds of a user request. This includes suggesting meals to prevent waste or recommending items based on seasonal demands.

  - ***Notification Delivery:*** Critical notifications, such as expiring items, should be delivered instantly. Non-critical notifications can have a maximum delay of 30 seconds.

- **Throughput**:

  - The system shall handle a peak load of 300- 500 concurrent users.
  - The system shall process at least 60 transactions per second during peak load.

(see 5.2)

- **Resource Utilization:**
  - The system should maintain an average CPU utilization of less than 70% during normal operation.
  - The app shall not utilize more than 80% of CPU and memory resources under peak load, maintaining smooth application performance.

5.2 Security Requirements

- **Authentication and Authorization:**
  - The system must employ robust authentication protocols like OAuth 2.0 or OpenID Connect to authenticate users securely.
  - Two-factor authentication (2FA) shall be mandatory for sensitive operations, including financial transactions and access to critical data.
- **Password Security:**
  - User passwords shall be securely stored using industry-standard hashing algorithms such as bcrypt or Argon2 to prevent unauthorized access in case of a data breach.
- **Granular Access Controls:**
  - The system must enforce granular access controls based on user roles and permissions, ensuring that access to sensitive functionalities and data is restricted to authorized personnel only.
  - Data transmissions between the application and external systems shall be encrypted using HTTPS to ensure secure communication channels.
- **Auditing and Logging:**
  - The system must maintain detailed logs of all user activities and system events in a comprehensive and tamper-proof manner.
  - Security logs shall be retained for a minimum of 12 months, allowing authorized personnel to perform audits and investigations as needed.

- **Vulnerability Management:**
  - The system must undergo routine security assessments, including vulnerability scans, to proactively identify and mitigate potential security risks.
  - Identified vulnerabilities shall be addressed promptly through the deployment of updates and patches to maintain a secure environment and prevent exploitation.

5.3 Reliability and Availability

- **Availability**:
  - System Uptime: The system must maintain an operational status with an uptime of 99.9%, allowing a maximum of 8.76 hours of downtime per year, excluding scheduled maintenance periods.
  - Scheduled Downtime: The system shall limit scheduled maintenance periods to a total of 2 hours per month, ideally during off-peak hours. This scheduled downtime is separate from the overall uptime requirement.

- **Data Integrity:**
  - The app should ensure data integrity through checksums .
  - System backups should be performed daily and stored offsite.

- **Disaster Recovery:**
  - The app should have a disaster recovery plan that can be implemented within 24 hours.
  - Fault Tolerance: The system must incorporate fault-tolerant measures to ensure continued functionality during hardware or software failures.

5.4 Maintainability

- **Code Quality:**
  - The app should be written in accordance with established coding standards.
  - Unit tests should be written to cover at least 80% of the code base.

- **Documentation**:
  - Comprehensive Documentation: The system shall have comprehensive technical documentation covering system architecture, APIs, and user guides.
  - Update Frequency: Documentation must be updated with each significant software release to reflect changes and new features accurately.

- **User Feedback and Feature Requests:**
  - The system shall Actively collect user feedback and feature requests and leverage this input to drive continual app evolution, ensuring alignment with evolving restaurant needs.

- ❖ Continuous Integration and Continuous Delivery (CI/CD)
  - ➢ The system must Implement an automated CI/CD pipeline for streamlined development, testing, and deployment.
  - ➢ The system must Automated code testing, build processes, and staging for efficient and error-free deployment.

- ❖ Monitoring and Logging:
  - ➢ The system must continuously monitor performance metrics, resource utilization, and error occurrences without impacting the functionality of existing features.
  - ➢ The system must collect detailed logs for analysis, aiding in identifying potential issues and optimizing performance without interfering with other functionalities.

- ❖ Security Updates and Patches:
  - ➢ The system must: Undergo regular security assessments and vulnerability scans to proactively identify and address potential risks without compromising existing features.
  - ➢ The system must: Promptly deploy updates and patches to maintain a secure environment and prevent exploitation, ensuring no adverse effects on existing features.

5.5 Portability

- ❖ Database Portability:
  - ➢ The system must: Be compatible with various database management systems like MySQL, PostgreSQL, MongoDB to ensure flexibility in data storage and retrieval across different environments.

- **Source Code Portability:**
  - The system must be developed using a widely accepted and supported programming language to ensure adaptability across different environments and platforms.

- **Compatibility:**
  - The system must offer well-documented and standardized APIs to enable effortless integration with other systems or third-party applications across diverse environments.

  - The system must ensure compatibility with major operating systems such as Windows, macOS, and Linux to facilitate seamless deployment across diverse environments.

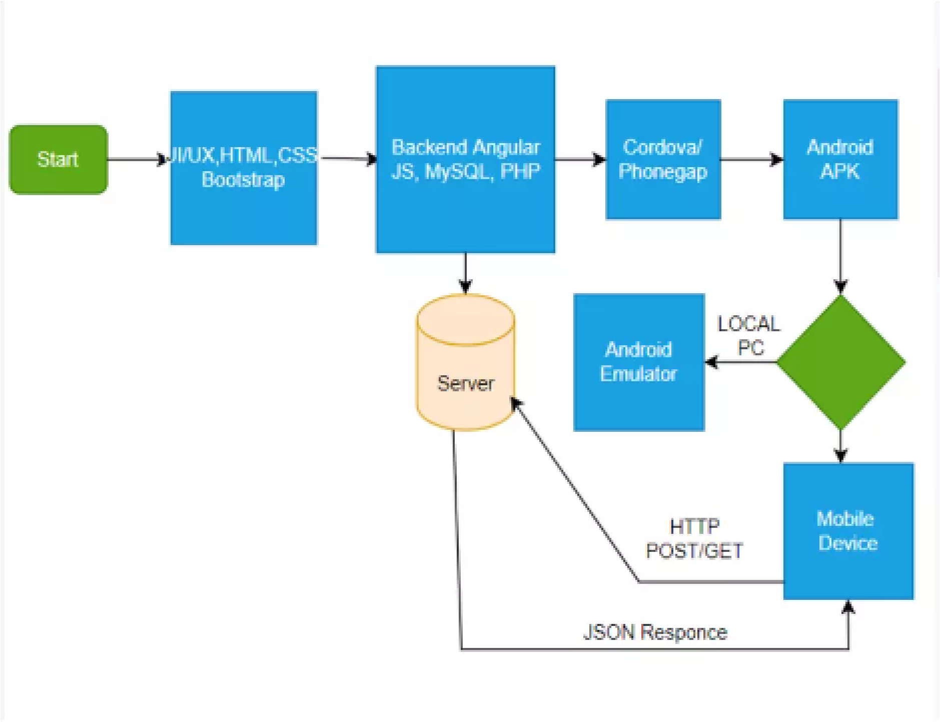

Fig 1: Block Diagram

6. Other Requirements

6.1. Legal and Regulatory Requirements

For Chef's Cabinet to operate legally and properly, it must conform to some legal and regulatory requirements.

1. Data Privacy: The system has to commit to data privacy rules to protect user data. For example, obtaining user permission for data collection, processing, storage, implementing strong user authentication mechanisms to prevent unauthorized access, and staying informed about any updates in relevant requirements.

2. Environmental Regulations: The app needs to adhere with environmental standards related to waste management and sustainability, as it focuses mainly is on minimizing food waste and carbon footprint.
3. Financial Regulations: the app involves financial data and transactions,it must comply with financial laws and security standards to protect
4. financial information. For example, using HTTPS protocol for secure communication over the internet.
5. Localization regulations: As the app may operate in several areas, it has to conform with some language requirements dedicated to these regions.
6. Cultural Aspects: Recognition of cultural differences in the regions where the app is used. Icons, images, and other elements should be culturally appropriate.

6.2. Other Requirements

1. **Database Requirements:** The app shall use scalable database to store data related to user and waste statistics, such as employing MySQL database system that would be hosted on a cloud server. Additionally, backups have to be done on a separate server to ensure data integrity.
2. **Update check:** The app has to send notifications to users and administrators must be triggered for any (legal) changes affecting the app's operation.
3. **Reusability:** Code components and modules should be designed with reusability for efficient development. We can use MVC architecture model for this purpose.
4. **Greenhouse Gas Emissions Calculations**: Do a specific algorithm or equation to calculate the greenhouse gas emissions associated with different food items and waste categories.

7. Appendices

7.1 Glossary

This section provides definitions for key terms and acronyms used throughout the Chef's Cabinet Software Requirements Specification (SRS).

Terms:

Chef's Cabinet: The software solution designed for optimizing food inventory management and waste reduction in the restaurant industry.
AI Integration: The incorporation of artificial intelligence features, including machine learning algorithms, for predictive analysis and smart decision-making.
User Classes: Different roles or categories of individuals interacting with the Chef's Cabinet system, such as Restaurant Manager, Kitchen Staff, End Users, System Administrators, and AI Specialists.
Food Inventory: The stock of food items available within the restaurant at any given time.
Real-time Inventory Tracking: The feature enabling the live monitoring and management of food inventory within the restaurant's storage system.
Surplus Management: The system's functionality that assists in identifying and managing surplus food items, including options for donations or trades.
Machine Learning Analysis: The utilization of machine learning algorithms to predict potential food spoilage based on historical data and trends.
Disaster Recovery: The system's plan and procedures to ensure recovery after a catastrophic event, ensuring minimal downtime.
Fault Tolerance: Measures incorporated within the system to maintain functionality during hardware or software failures.
Data Integrity: The system's capability to maintain and assure the accuracy and consistency of stored data.

Acronyms:

SRS: Software Requirements Specification
API: Application Programming Interface
2FA: Two-Factor Authentication
HTTPS: Hypertext Transfer Protocol Secure
CI/CD: Continuous Integration/Continuous Delivery
MVC: Model-View-Controller
SQL: Structured Query Language

e areas we will concentrate in our cut flower farming business. If need arises we will definitelydd more produce to our list;

- Cut flowers and cut cultivated greens
- Christmas Trees

- Nursery stock crops
- Annual bedding and garden plants
- Potted flowering plants
- Foliage plants
- Potted herbaceous perennials
- Propagative floriculture materials

Main Branch Address

123Plants St., New York

Phone Number

123-456-7890

Email Address

hello@gogledoc.com

Let's Talk