Logo, company name

Description automatically generated

**PUSL3190 Computing Individual Project**

**Project Interim Report**

**Smart Agriculture System**

Supervisor:

Name:

Plymouth Index Number:

Degree Program: BSc. (Hons) Software Engineering

**Content**

[Abstract iv](#_Toc160818415)

[Chapter 01 – Introduction 1](#_Toc160818416)

[1.1 Introduction 1](#_Toc160818417)

[1.2 Problem Definition 2](#_Toc160818418)

[1.3 Project Objectives 4](#_Toc160818419)

[Chapter 02 – System Analysis 6](#_Toc160818420)

[2.1 Facts Gathering Techniques 6](#_Toc160818421)

[2.2 Existing System 8](#_Toc160818422)

[2.3 Use Case Diagram 10](#_Toc160818423)

[2.4 Drawbacks of the existing system 11](#_Toc160818424)

[Chapter 03 – Requirement Specification 13](#_Toc160818425)

[3.1 Functional Requirements 13](#_Toc160818426)

[3.2 Non-Functional Requirements 13](#_Toc160818427)

[3.3 Hardware Requirements 14](#_Toc160818428)

[3.4 Software Requirements 14](#_Toc160818429)

[Chapter 04 – Feasibility Study 15](#_Toc160818430)

[4.1 Operational Feasibility 15](#_Toc160818431)

[4.2 Technical Feasibility 15](#_Toc160818432)

[4.3 Outline Budget 16](#_Toc160818433)

[Chapter 05 – System Architecture 17](#_Toc160818434)

[5.1 Class Diagram 17](#_Toc160818435)

[5.2 ER Diagram 17](#_Toc160818436)

[5.3 High-Level Architectural Diagram 18](#_Toc160818437)

[5.4 Networking Diagram 19](#_Toc160818438)

[Chapter 06 – Development Tools and Techniques 20](#_Toc160818439)

[6.1 Development Methodology 20](#_Toc160818440)

[6.2 Programming Languages and Tools 21](#_Toc160818441)

[6.3 Third Party Components and Libraries 22](#_Toc160818442)

[6.4 Algorithms 24](#_Toc160818443)

[Chapter 07 – Discussion 26](#_Toc160818444)

[References 28](#_Toc160818445)

[Appendix 29](#_Toc160818446)

**No table of figures entries found.**

# Abstract

# Chapter 01 – Introduction

## Introduction

in the rapidly evolving landscape of the food industry, the combination of automation and image detection technologies carries a huge potential.to streamline process and enhance efficiency. This interim report depicts the intersection of automation and food image detection, aiming to address key challenges and the opportunities in these promises.

## Problem Definition

The area that I am facing is the food industry and it faces several challenges including the need for accurate method of inventory management, and consumer engagement. Traditional manual approaches like food inspection and inventory tracking are often time-consuming, labor intensive and prone to human error. Infact, with the development of the demand for transparency and sustainability, there is a heightened seeking for robust system to ensure product quantity and traceability.

In the context of self-service restaurants, they are facing challenges in streamlining the ordering, food analysis, and billing processes due to reliance on non-automated systems.

These systems are processed by manual paper-based workflows which lead to inefficiencies, errors, and delays, by consequencing customer dissatisfaction. Manual order taking, handwritten tickets, and manual.

Services which are done in existing restaurants like ordering serving billing can occur in accurate results .

The people who work in restaurants—owners, managers, and employees—bear the brunt of these difficulties because they deal with mistakes and inefficiencies. Self-service restaurants are looking more and more for fully automated systems that use image processing for food recognition, RFID/NFC for tray and customer identification, and QR code systems for quick and safe payment transactions in order to address these problems.

Through digitization, certain processes can be implemented digitally, improving data transparency and facilitating more efficient business analysis. Self-service restaurants can lower labor costs, increase customer satisfaction, and improve operational efficiency by automating the billing, food analysis, and payment processes.

Project Objectives

# 1.Develop automated system for food detection:

# By training models on enormous datasets of food photos, the initiative aims to automate the process of assessing food by utilizing sophisticated image detection techniques including computer vision and machine learning.

# Develop a cutting-edge image processing system that can recognize and categorize food items placed on trays at a self-serve buffet. Sophisticated algorithms will be employed by technology to ensure accurate and efficient food recognition, hence enhancing the dining experience for customers.

# 2.Provide food analysis integration with billing system:

# Using an image processing system, a billing system that integrates a food analysis module and calculates cost can be replicated.

# The billing system's real-time functioning will guarantee pricing adjustments and open communication between the eatery and its customers.

# RFID Coordination

# To easily identify trays, use RFID technology. To establish a unique connection between the tray and the consumer, each tray must have an RFID tag attached to it. RFID integration enhances the customized dining experience, reduces error rates, and increases operational performance.

# 3. QR Code Generation

# By reducing wait times at the payment counter, it aims to improve overall customer demands by facilitating rapid and safe transactions.

# Include a feature that compiles billing data and creates QR codes for each tray. With these QR codes, customers will have a simple and secure way to complete their payments. Each QR code will contain the total cost of the relevant tray.

# Payment Integration

# Include a secure payment system to facilitate easy and secure transactions between customers and the restaurant. Include a secure payment system to facilitate easy and secure transactions between customers and the restaurant. Since multiple payment methods are supported with the payment integration, customers will have flexibility and convenience.

# Interface That's Easy to Use

# Make an interface that is simple to use and intuitive for customers and staff at restaurants. The UI of the system is designed to be user-friendly, easily navigable, and to offer a positive experience to all users.

# Use in Self-Serve Restaurants

# Install the automated buffet system with ease in a range of self-service dining environments. In order to ensure a smooth implementation of the new technology and maximize its benefits, restaurant staff members must get intensive training.

# Chapter 02 – System Analysis

## 2.1 Facts Gathering Techniques

Facts Gathering techniques on automatic self-service and food detection process involve several strategies to understand the current landscape and challenges that faced here are some effective techniques.

2.1 Observational Studies – Observed some of the vedios of restuarents and supermarkets where there is automated checkout system and food detection technologies in action. It was much better guidance to identify usability issues, technical challenges, and operational workflows.

1.Intervies – Have gone Through interviews with owners of some restaurants. This technique was helpful to gather insights into preferences, experiences regarding the situation and background of processes which happens in a restaurant and what they are willing to have in future to make this process easy.

## 2.2 Existing System

## Using Computer Vision to Identify Food:

## The goal of research is to create models and algorithms that can reliably identify and detect different kinds of food. In order to interpret photos and identify various food products, deep learning techniques like convolutional neural networks (CNNs) are used.

## The use of deep learning techniques to identify food products kept in a refrigerator was investigated in a paper named "Food Recognition and Detection for Smart Refrigerator Applications" (Li et al., 2017). The system was designed to help users keep track of food inventories and expiration dates.

## A different study, "Food Image Recognition Using Convolutional Neural Network" (Yu et al., 2019), suggested using CNN to identify and categorize food photos that were taken with smartphones. It is possible that the system will be coupled with inventory management or meal ordering systems.

## Food Service Automation:

## This field of study focuses on automating order taking, preparation, and billing, among other areas of food service. The goals of automation are to increase productivity, decrease human error, and streamline procedures.

## A touchscreen kiosk-based system for automating the ordering and billing procedure in fast-food restaurants was presented in "Automated Ordering and Billing System for Fast Food Chains" (Kumar et al., 2018). Without assistance from a human, customers can place orders and make payments.

## Researchers created a robotic system that can prepare meals on its own in a study titled "Robotic Kitchen: A Fully Automated Robotic Kitchen Assistant for Assisted Meal Preparation" (Birk et al., 2020). To perform a variety of cooking chores, the device combined robotic manipulation with food detecting skills.

## Combining Food Detection with Invoicing Software:

## Research looks on how to combine billing systems and food detecting technologies to make checkout quick and easy.

## A smart shopping cart with sensors and computer vision technology to recognize objects placed in the cart was proposed in "Smart Shopping Cart with Automated Billing System Using IoT" (Mishra et al., 2020). There is no need for a manual checkout process because the bill is generated automatically by the system.

## An automated checkout system that uses computer vision algorithms to detect and recognize products placed on a conveyor belt was introduced in a research titled "Automated Checkout System Using Computer Vision and Machine Learning" (Chen et al., 2019). The bill is then generated by the system using the goods that it has discovered.

## RFID and NFC-Powered Food Monitoring Devices:

## Research examines how food products can be tracked from manufacturing to consumption using Near Field Communication (NFC) and Radio-Frequency Identification (RFID) technologies. In order to automate checkout procedures and guarantee accurate invoicing based on item identification, these systems can be connected with billing systems.

## An RFID-based system to track food products along the supply chain was proposed in "RFID-Based Food Traceability System for Supply Chain Management" (Zhang et al., 2018). By linking RFID tags to product details and prices, the method enables automatic billing.

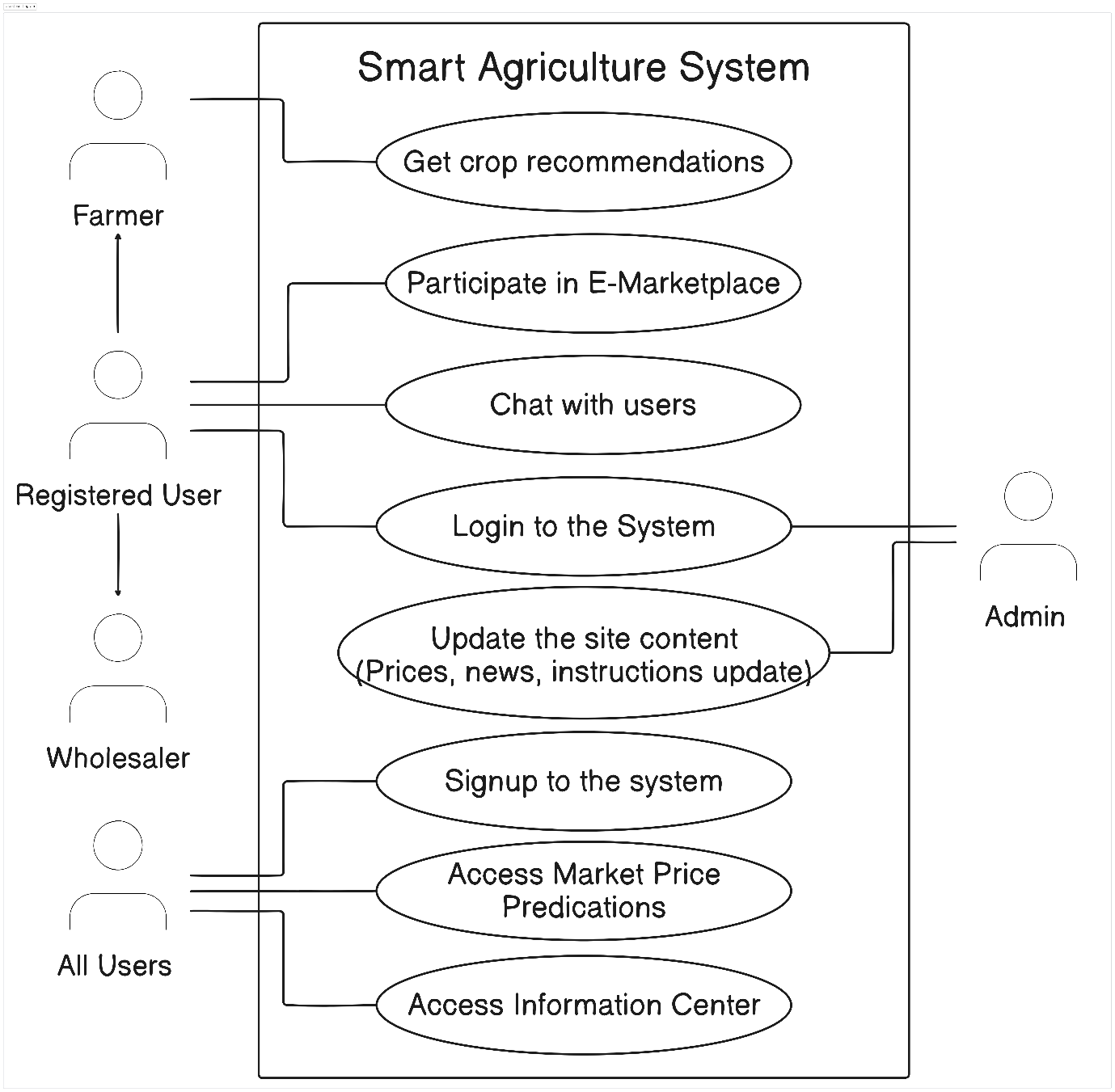
## The article "Mobile Application for Automated Food Ordering and Billing" (Gupta et al., 2019) describes a smartphone application that recognizes food products through image recognition. Through the app, users may directly place orders, with billing determined by the items selected automatically.

## Smart Fridges Using Sensors and Automated Billing:

## The possibility of automated food identification and billing exists with smart refrigerators that are outfitted with cameras and sensors. The creation of such systems to track food inventories, identify expiring products, and enable automated replenishment billing is being researched.

## A smart fridge prototype that uses computer vision and IoT sensors to track food items stored inside was shown in "Smart Fridge with Automated Billing System Using IoT" (Singh et al., 2021). For things that are eaten, the system automatically creates bills and recommends replenishment.

## 2.3 Use Case Diagram



## 2.4 Drawbacks of the existing system

* **Limited Integration of Technologies:** Existing systems often utilize technology in silos, lacking a comprehensive integration that leverages the full potential of AI, ML, and mobile platforms in a cohesive manner. This disjointed approach can limit the effectiveness and accessibility of the technology for all stakeholders in the agriculture sector.
* **Accessibility and Usability Issues:** Many current agricultural technology solutions are not designed with the end-user in mind, particularly smallholder farmers who may have limited technical literacy. The complexity or lack of user-friendly interfaces can deter farmers from adopting these technologies, thereby limiting their benefits.
* **Insufficient Real-Time Data Utilization:** While some systems collect and store agricultural data, they fail to leverage this data effectively for real-time decision-making. The lack of timely insights can prevent farmers from making informed decisions that could enhance their productivity and profitability.
* **Inadequate Direct Market Access:** Many existing platforms do not adequately address the challenges posed by middlemen in the agricultural supply chain, failing to provide farmers with direct access to markets and fair pricing for their produce. This can result in reduced income for farmers and inefficiencies in the supply chain.
* **Data Security and Privacy Concerns:** With the increasing digitization of agricultural data, concerns about data security and privacy have emerged. Existing systems may not offer robust protections for the sensitive information of farmers and other stakeholders, potentially deterring participation.
* **High Operational Costs and Maintenance:** The cost of implementing and maintaining some of the existing agricultural technology solutions such as integrating IOT can be prohibitively high for smallholder farmers and developing economies. This financial barrier can limit the adoption and scalability of such systems.
* **Lack of Comprehensive Support Systems:** Support systems for troubleshooting, technical assistance, and knowledge sharing are often inadequate in existing solutions, leaving users without the necessary support to fully utilize the technology.

# Chapter 03 – Requirement Specification

## 3.1 Functional Requirements

## 

## **Functional Needs: Automated Food Evaluation:**

## **Convolutional neural networks (CNNs) are one sort of image processing technology that the system should use to accurately detect and analyze the various types of food items placed on a tray.**

## **It must be able to handle a wide range of food products with varying sizes, shapes, colors, and configurations so that each is accurately identified.**

## **Real-time results from the food analysis module should enable patrons and restaurant employees to view the flagged items quickly.**

## **Connectivity to the Billing System:**

## **To enable correct cost calculation based on the food items recognized by image processing, the system should smoothly interact with the restaurant's billing system.**

## **Any modifications made by restaurant management should be reflected in the system's dynamic updating of prices and menu items.**

## **Customers should receive itemized bills or receipts from the billing system that include information about every food item identified throughout the self-service procedure.**

## **Tray and Identification of Customers:**

## **The system ought to utilize RFID/NFC technology to provide each tray with a distinct identity and associate it with the corresponding consumer.**

## **Orders and payments made by each customer should be automatically matched to their tray through an error-free identification mechanism.**

## **For a more customized eating experience, customers have to be able to link their payment data, loyalty program information, or other preferences to their tray.**

## **Integration of QR Code Payments:**

## **The system ought to produce QR codes with payment information, such as the total amount owed and transaction data.**

## **To avoid fraud or tampering, QR codes have to be properly encoded and specific to each transaction.**

## **Consumers should be able to start the payment process by using their mobile devices to scan the QR code. Secure payment gateway integration should be used to process the payment.**

## 3.2 Non-Functional Requirements

## **Achievement:**

## **In order to reduce client wait times, the system should be able to conduct food analysis and invoicing transactions quickly.**

## **It ought to be tuned to manage busy times and large numbers of transactions without lagging or experiencing delays in performance.**

## **Dependability:**

## **During restaurant business hours, the system ought to be dependable and accessible with minimal downtime for upgrades or maintenance.**

## **Fault-tolerant methods should be included to handle mistakes gently and guarantee that clients receive continued service.**

## **Safety:**

## **To safeguard sensitive client data and transaction information, such as payment details and personal information, strong encryption should be used.**

## **To prevent unauthorized users from accessing the system and to protect user privacy, access control procedures should be put in place.**

## **To protect consumer data, the system must abide by applicable data protection laws, such as GDPR or PCI DSS.**

## **Usability:**

## **Both restaurant employees and patrons should find the user interface to be simple to use and intuitive.**

## **From placing orders to making payments, customers should be able to effortlessly complete the self-service process without help.**

## **It should be easy for restaurant employees to operate the system with minimal training requirements if they have access to clear instructions and user interfaces.**

## **Scalability**

## **As the restaurant expands and can handle more patrons and transactions, the system ought to be built to scale easily.**

## **Long-term scalability and flexibility should be ensured by its capacity to accommodate new features or functionalities without requiring substantial modifications to the underlying architecture.**

## **Sustainability:**

## **Modular components and comprehensive documentation should make it simple to maintain and update the system and enable continuous maintenance tasks.**

## **Upgrades and improvements ought to be implemented seamlessly, causing the least amount of disturbance to the system's functioning and leaving no impact on the user's experience.**

## **To guarantee optimal performance and dependability, regular testing and monitoring should be carried out in order to spot any problems early on and take appropriate action.**

## 3.3 Hardware Requirements

**Web Server Requirements**

## 

## 1. Web Camera: Web cameras are a crucial piece of gear that the self-service restaurant system uses to take pictures of the food that is placed on trays. The image processing module uses the visual input from these cameras to precisely evaluate and identify different food items.

## Specification: Every self-service station will have HD web cams installed, with a minimum resolution of 720p. To take crisp, detailed pictures of food products, cameras should have a large field of vision and autofocus capability.

## 2. Mobile Camera: Using their cellphones to scan QR codes and communicate with the self-service restaurant system, consumers can use mobile cameras as extra image capturing devices. The digital interface and the real world can seamlessly integrate thanks to these cameras.

## Specification: To optimize user accessibility, compatibility with popular smartphone models (iOS and Android) will be guaranteed. High-resolution picture capture and QR code recognition should be supported by cameras.

## 3. Plastic Tray: During the self-service dining experience, plastic trays act as the actual platform for holding food items. These trays have RFID/NFC tags installed for tracking and customer identification. This allows for proper billing inside the system and individualized service.

## Specification: RFID/NFC tags integrated into lightweight, sturdy plastic trays will be used. Trays should be large enough to hold a variety of foods and cutlery while still being comfortable for customers to grasp and ergonomic.

## Tray holders are designed to give plastic trays a firm base so they stay firmly in place throughout the self-service dining experience. The purpose of strategically placing these holders inside the restaurant layout is to maximize client accessibility and convenience.

## Specification: Each self-service station will have sturdy tray holders made of reinforced plastic or stainless steel fitted. To avoid spillage or tipping, holders should have non-slip surfaces and ergonomic designs.

## 5. RFID/NFC Module: Basically, these modules make it easier for customers and trays to be identified in the self-service restaurant system. These modules allow for wireless communication with RFID/NFC tags implanted in plastic trays, allowing for real-time tracking and customer-specific service.

## Specification: Industry-standard protocols (such as ISO/IEC 14443 for NFC) will be supported by dependable and small RFID/NFC devices. Secure communication and smooth integration with the primary system architecture should be supported by the modules.

## 6. Laptop: Synopsis: The self-service restaurant system software is hosted and operated by laptops, which are the main computing devices. These devices offer the processing capacity and networking needed to efficiently handle image processing, invoicing, and UI elements.

## Specification: We'll be using high-performance laptops with solid-state drives (SSDs) for quick storage access, enough of RAM (8GB or more), and multi-core processors (such the AMD Ryzen 5 or Intel Core i5). Additionally, a laptop should have enough USB ports for external devices and built-in Wi-Fi.

## Make sure the hardware specs support the anticipated functionality and performance expectations and are in line with the technical specifications of your project.

## 3.4 Software Requirements

# 1. Image Processing Module: In the self-service restaurant system, the image processing module is crucial for food analysis and identification. It uses the most recent deep learning methods—YOLOv8 in particular—for tasks including object identification and recognition.

# Implementation: The YOLOv8 model is implemented and trained using TensorFlow, a popular machine learning framework. Furthermore, Roboflow is employed to optimize data preparation and augmentation, hence augmenting the precision and resilience of the model.

# 2. Communication Module for RFID/NFC:

# This module uses RFID/NFC technology to make tray and customer identification easier. It makes it possible for RFID/NFC modules and the main system to communicate seamlessly, guaranteeing precise tracking and customer-specific service.

# Software components in charge of creating and overseeing RFID/NFC connectivity will be developed using Node.js. Effective integration with the hardware components will be accomplished by utilizing pertinent libraries and packages that are compatible with Node.js.

# 3. QR Code Generation Tool:

# The QR code generation module dynamically creates QR codes providing payment details for customer transactions. In the self-service restaurant system, it guarantees safe encoding and scanning capabilities, enabling speedy and secure payment transactions.

# QR codes is generated programmatically using Node.js-based libraries like qrcode or qr-image. In order to create and display QR codes throughout the payment process with ease, these modules will be easily integrated into the system architecture.

# 4. Module for Billing and Payment:

# All billing-related tasks, such as itemized receipts, transaction computations, and payment gateway integration, are handled by the billing and payment module. For consumer transactions, it guarantees safe payment processing and precise billing.

# Implementation: To enable safe payment processing, the system will be connected with either the Stripe API or the PayPal API. Server-side logic for transaction management, billing computations, and payment gateway communication will be developed using Node.js.

# 5. Development of User Interfaces:

# The user interface development process entails using React to create user-friendly and responsive interfaces for both restaurant employees and patrons. It guarantees fluid communication and usage on various platforms and devices.

# Node.js is used on the server-side to manage requests and communicate with back-end services, while React is used for front-end development. To facilitate communication between the server-side functionality and the React-based front end, Node.js will be used to construct RESTful APIs.

# 6. Integrated Development Environment (IDE): Described as a comprehensive toolkit for software development, an IDE offers strong support for JavaScript, Node.js, and React development. Implementation: The main IDEs for software development will be Visual Studio Code or Atom. These IDEs provide a large selection of tools and extensions to improve code quality, productivity, and teamwork throughout the development process.

# Chapter 04 – Feasibility Study

Operational feasibility evaluates the viability and practicality of a given project from an operational perspective. An example of this would be the implementation of a food detection and automatic billing system in self-service restaurants. The following are important elements of operational feasibility, accompanied by examples:

## 4.1 Operational Feasibility

## 

## Evaluation of Current Operations:

## As an illustration, carefully examine how self-service restaurants now operate, paying particular attention to the manual ordering procedure, paper-based billing, and payment options. Determine inefficiencies that cause customer discontent because of manual processes, billing problems, lengthy wait times during peak hours, etc.

## Acceptance by Users:

## I have used surveys or interviews to get input on the planned food detection and automation invoicing system from restaurant owners, managers, employees, and patrons.It has some quit good acceleration on users that can adjust to new technology and the system satisfies their expectations and needs.

## Availability of Resources:

## For instance: Ascertain whether the necessary hardware (such as cameras, RFID/NFC modules), software (such as image processing algorithms, payment gateway integration), and funding are available for the system's implementation. Determine if the restaurant can afford the one-time setup fees as well as the continuing costs of maintenance.

## Technical Proficiency:

## Assess the management team's and employees' technical proficiency in running the suggested system. Assess the need for additional technical staff hire or training in order to assist with system setup and maintenance.

## Integration Difficulties

## Determine the possible obstacles to integrating the food detection and automation billing system with the technologies and procedures already in place in restaurants, like point-of-sale systems. To guarantee flawless operations, evaluate compatibility concerns, data migration needs, and the necessity of seamless integration.

## Adherence to Regulations:

## Verify that the suggested system conforms with all applicable laws, rules, and guidelines regarding money transactions, data privacy, and food safety. Examine how regulatory constraints, such as the necessity of secure payment processing and data encryption to safeguard client information, may affect the architecture and functionality of the system.

Analyzing Risk:

Recognize and evaluate potential risks and impediments, including as technological risks, market competition, and unanticipated occurrences, that may affect the system's successful deployment and operation. Create backup plans to reduce risks and guarantee company operations.

Benefit-Cost Analysis:

To ascertain the financial feasibility and possible return on investment of putting the food detection and automation invoicing system into place, perform a cost-benefit analysis. Examine the anticipated benefits, such as enhanced customer satisfaction, cost savings, and higher efficiency, against the estimated costs of development, implementation, and maintenance.

The adoption of the food detection and automation billing system in self-service restaurants can be decided upon by stakeholders after a full assessment of these operational feasibility considerations and resolution of any issues or problems.

## 4.2 Technical Feasibility

Technical viability evaluates a project's suitability in light of available resources and advancements in technology. An example of this would be the implementation of an automated invoicing system and food detection system in self-service restaurants. The following are important elements of technological feasibility, accompanied by examples:

Technology Evaluation

Assess whether the selected technologies—such as web development frameworks like React and Node.js, RFID/NFC modules, payment gateways like Stripe and PayPal, and image processing algorithms like Roboflow and YOLOv8—are appropriate for the suggested solution. Make sure that these technologies can function as a cohesive unit and fulfill the project's needs.

Requirements for Hardware and Software:

List the hardware and software components—such as cameras for picture taking, RFID/NFC modules for identifying customers, servers for database hosting and payment processing, and software development tools—that are required for the system's implementation. Examine these components' compatibility and availability with the current infrastructure.

Scalability

As an illustration, think about how scalable the suggested system is to handle future expansion and rising demand. Examine whether a rising user base, more transactions, and a larger volume of data can be handled by the selected technologies and architecture without experiencing appreciable performance degradation. By creating flexible, modular parts that are simple to grow or update, you can plan for scalability.

Information Administration:

When it comes to information administration, consider the needs for the system's data processing, management, and storage. Ascertain the kind and amount of data—such as menu items, customer orders, transaction records, and billing information—that are produced by the

food detection and billing procedures. To guarantee data security and integrity, determine whether relational or non-relational databases, data encryption, and backup options are required.

Complexity of Integration:

Evaluate how difficult it will be to integrate the different parts of the system, such as the user interfaces, payment gateways, RFID/NFC communication, QR code creation, and image processing modules. Take into account the requirement for bespoke integration solutions, SDKs, or APIs to enable smooth data exchange and communication between various modules.

Resources for Development:

Assess the availability of qualified engineers, developers, and technical specialists needed to design, build, and implement the system. Determine whether the current team has the requisite experience in software development, systems integration, computer vision, and machine learning. Assess the need for further training or resource hire in order to close any skill shortages.

Risks Associated with Technology:

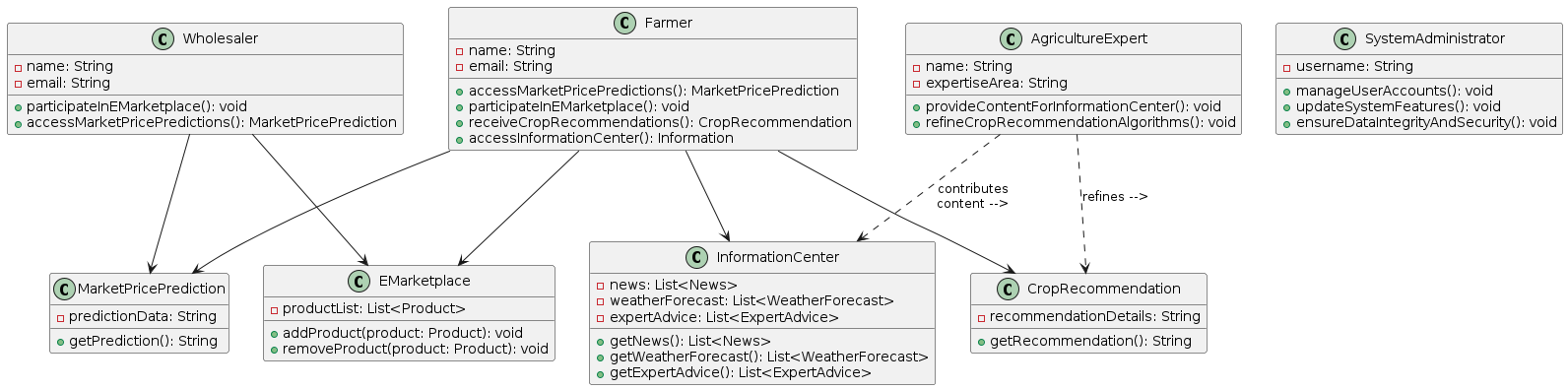
As an illustration, list the possible dangers and difficulties that could arise from adopting and using new technologies, such as incompatibilities, software problems, hardware malfunctions, and security flaws. Create risk-reduction plans that include comprehensive testing, prototyping, and continuing support and maintenance.

Stakeholders can ascertain the technical viability of deploying the food detection and automation billing system in self-service restaurants by carefully evaluating these technical feasibility considerations and resolving any issues or problems.

## 4.3 Outline Budget

# Chapter 05 – System Architecture

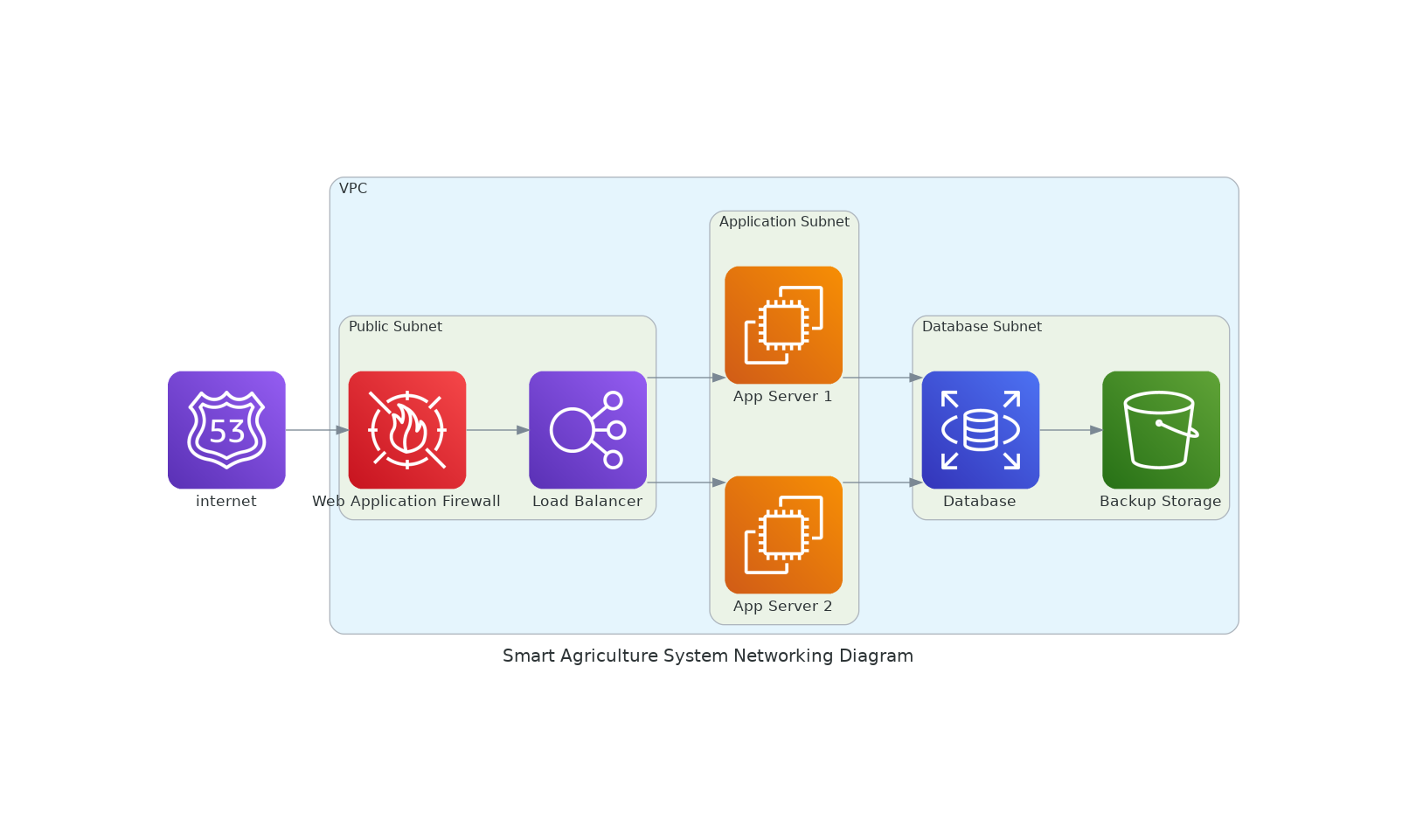
## 5.1 Class Diagram



## 5.2 ER Diagram

## 5.3 High-Level Architectural Diagram

## 5.4 Networking Diagram



# Chapter 06 – Development Tools and Techniques

## 6.1 Development Methodology

## 6.2 Programming Languages and Tools

## 6.3 Third Party Components and Libraries

## 6.4 Algorithms

# Chapter 07 – Discussion

# References

# Appendix