# Rajiv Gandhi Univesity of Knowledge Technologies, Basar.

# Software Requirements Specification (SRS)

## for

## Agri Vision Web Application

| | | |
|---|---|---|
| Depavath Naresh | Mudavath Kalyan | Mudapally Aravind |
| B200391 | B201174 | B200778 |

October 15,2024.

**Mini Project Guide:**U.Nagamani Madam.

| Version | Date | Author | Description |
|---|---|---|---|
| 1.0 | October 15,2024 | Depavath Naresh,<br>Mudavath Kalyan,<br>Mudapally Aravind. | Initial Draft. |

# Table of Contents

# 1. Introduction

### 1.1 Purpose

The purpose of the Agri Vision Project is to create a comprehensive platform that bridges the gap between farmers, consumers, and sellers in the agriculture industry. By eliminating the need for contractors, the platform ensures that farmers receive a fair price for their goods while providing consumers with fresh, direct-from-farm products. Additionally, the platform empowers farmers with tools to manage their business effectively, check weather conditions, and resolve queries through an integrated chatbot.

### 1.2 Scope

This web application includes multiple features for various user roles:

- Farmers can sell their crops and grains, buy farming supplies, rent machines, and access real-time weather updates.
- Sellers can manage their inventory and sell agricultural products like seeds and pesticides.
- Consumers can browse and buy products directly from farmers, manage their profiles, and view their carts.
- A GPS-enabled weather system allows farmers to check local weather updates, helping them make informed decisions.
- A chatbot provides support and answers to farming-related queries.

This system aims to build a fair, transparent, and efficient platform for all stakeholders in the agricultural ecosystem.

### 1.3 Definitions, Acronyms, and Abbreviations

- **GPS**: Global Positioning System
- **UI**: User Interface
- **SRS**: Software Requirements Specification
- **API**: Application Programming Interface
- **CRUD**: Create, Read, Update, Delete (operations related to data)

### 1.4 References

- Google map api,Open Street api,etc...

### 1.5 Overview

This document outlines the functional and non-functional requirements of the Agri Vision Project. It covers system design, specific features, external dependencies, user roles, and various operational conditions to provide a clear picture of the system’s functionality.

# 2. Overall Description

### 2.1 Product Perspective

The Agri Vision Project is designed to be a self-contained, modular web application that can be expanded in the future with additional services like predictive farming analytics, supply chain monitoring, or advanced payment gateways. The system will connect multiple stakeholders (farmers, consumers, and sellers) and integrate seamlessly with weather services, GPS-based location services, and secure payment gateways.

### 2.2 Product Functions

Key features of the platform include:

- **Seller Profile Management**: Sellers can add, edit, and remove products from their inventory, as well as update their profile and contact information.
- **Farmer Profile Management**: Farmers can list grains for sale, buy seeds or pesticides, rent farming machinery, and check real-time weather data through an interactive map.
- **Consumer Profile Management**: Consumers can edit their profiles, manage addresses, browse products, and manage their shopping carts.
- **Weather GPS Integration**: Farmers can check weather conditions in their location by clicking on the map, which uses GPS data to show real-time weather updates.
- **Chatbot Integration**: A virtual assistant for answering farmers' questions, ranging from farming tips to platform-related queries.

### 2.3 User Characteristics

- **Farmers**: Typically non-technical users, comfortable with basic web and mobile applications, seeking to sell their produce and buy supplies with minimal hassle.
- **Sellers**: Businesses or individuals familiar with managing products online, responsible for selling farming supplies and renting machinery.
- **Consumers**: End-users or businesses looking to buy fresh agricultural products directly from farmers, managing personal accounts, orders, and payments.
- **Admin**: Technical users responsible for managing the platform, including user permissions, product listings, and overall system performance.

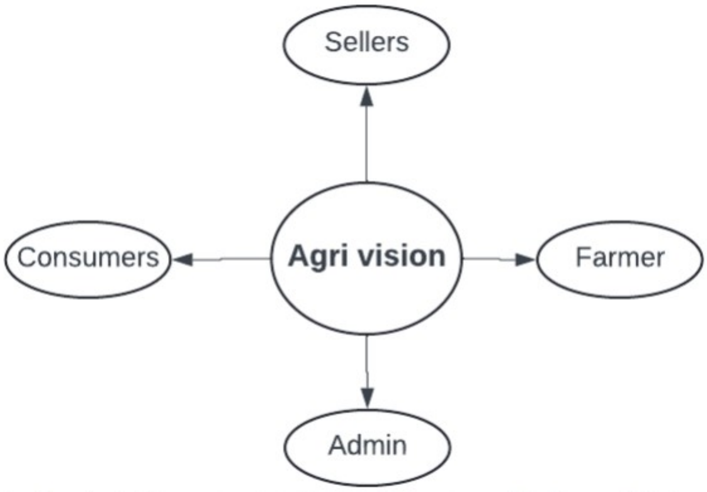


**Figure**:Agri vision connects i.e Farmes,Sellers,Consumers

**2.4 Operating Environment**

- **Browser Compatibility**: The application will run on modern browsers like Chrome, Firefox, Safari, and Edge, supporting both desktop and mobile platforms.
- **Mobile-Friendly**: A responsive UI design ensures usability on smartphones and tablets.
- **GPS Access**: GPS will be required to provide weather updates and other location-based services.
- **Cloud-based Hosting**: The platform will be hosted on cloud infrastructure to ensure scalability and availability.

**2.5 Assumptions and Dependencies**

- A reliable internet connection is required for accessing the platform.
- External APIs for weather data and secure payment processing will be integrated.
- GPS permissions are needed to provide accurate weather data.

# 3. Specific Requirements

## 3.1 Functional Requirements

### 1. User Registration and Authentication

- Users must register with valid email addresses and phone numbers.
- The platform must support different user roles (farmers, sellers, consumers).
- Secure authentication using email verification or two-factor authentication (2FA) should be provided.

### 2. Seller Profile Features

- **Add/Delete Products**: Sellers can upload product details including images, descriptions, pricing, and availability.
- **Edit Profile**: Sellers can edit personal information and company details, including addresses and contact numbers.
- **Inventory Management**: Sellers can update stock quantities and pricing as needed.

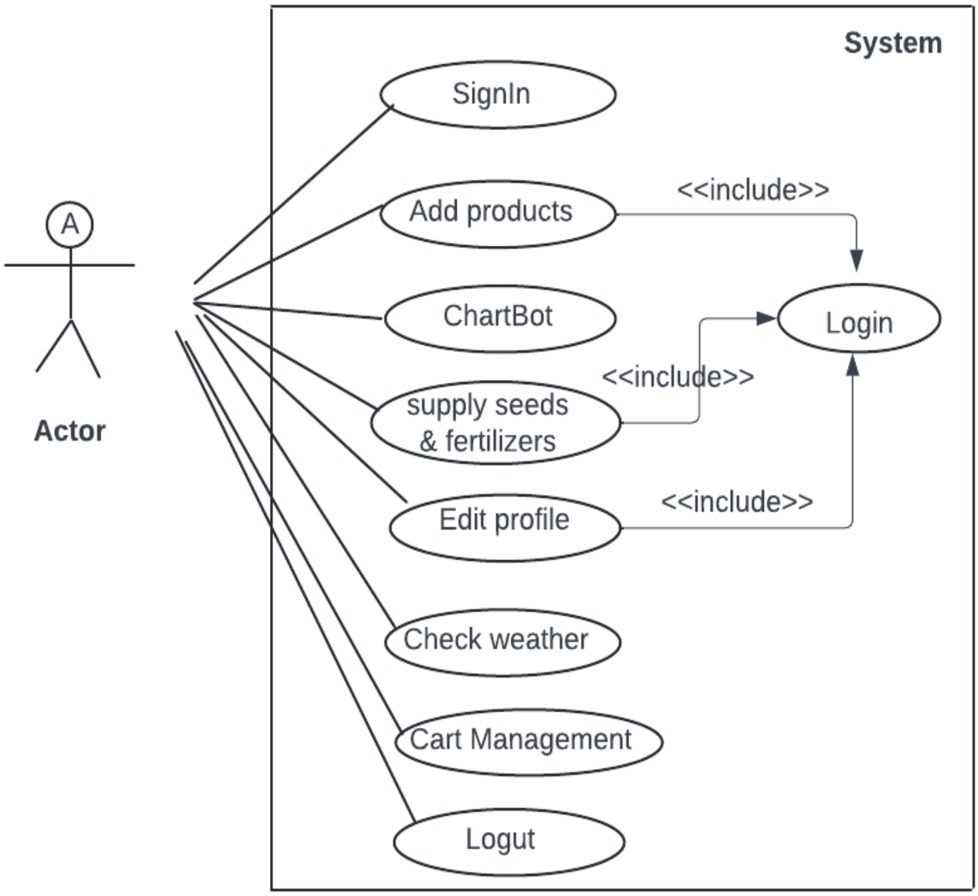


**Figure**:Use case Diagram of Supplier.

### 3. Farmer Profile Features

- **Add/Delete Grains**: Farmers can list their crops and grains for sale.
- **Purchase Seeds/Pesticides**: Farmers can browse, select, and purchase farming supplies like seeds or pesticides from sellers.
- **Rent Machines**: Farmers can rent agricultural machines from sellers, specifying the rental duration.
- **Check Weather via Map**: Farmers can click on a map to view weather updates for their region using GPS data.
- **Edit Profile & Address**: Farmers can update their personal information and contact details.

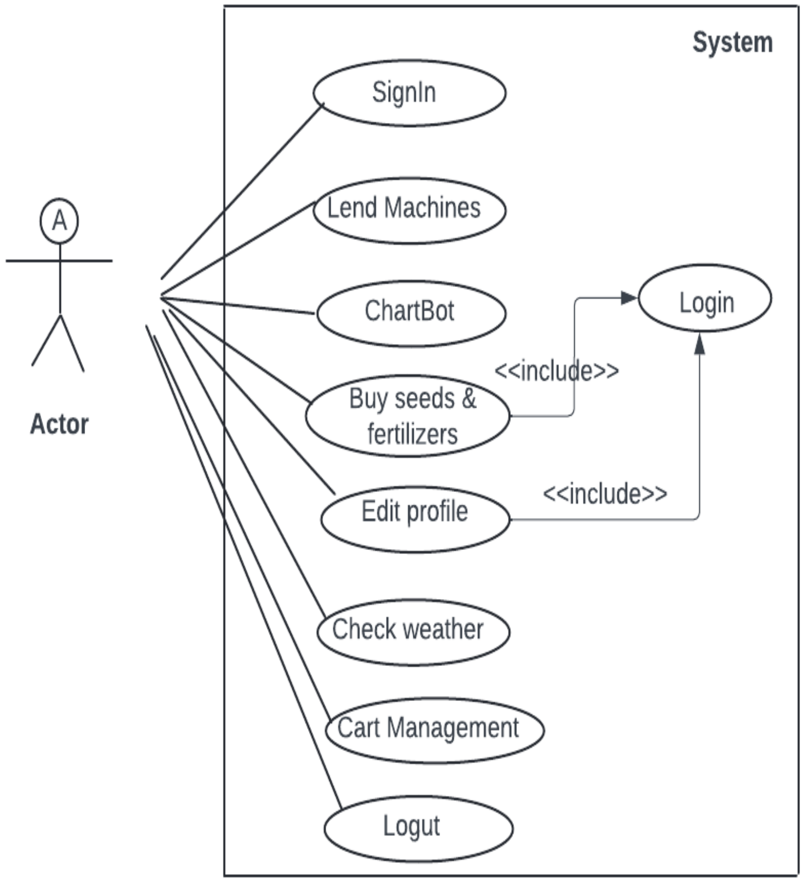


**Figure**:Use case Diagram of Farmer

**4. Consumer Features**

- **Edit Profile & Address**: Consumers can update their profile information, including personal details and delivery addresses.
- **Buy Products from Farmers**: Consumers can browse and purchase products directly from farmers, adding them to their cart.
- **Cart Management**: Consumers can manage their cart, change product quantities, and remove products before proceeding to checkout.
- **Order Summary & Payment**: Consumers receive a detailed order summary and have multiple payment options, including card and cash-on-delivery.

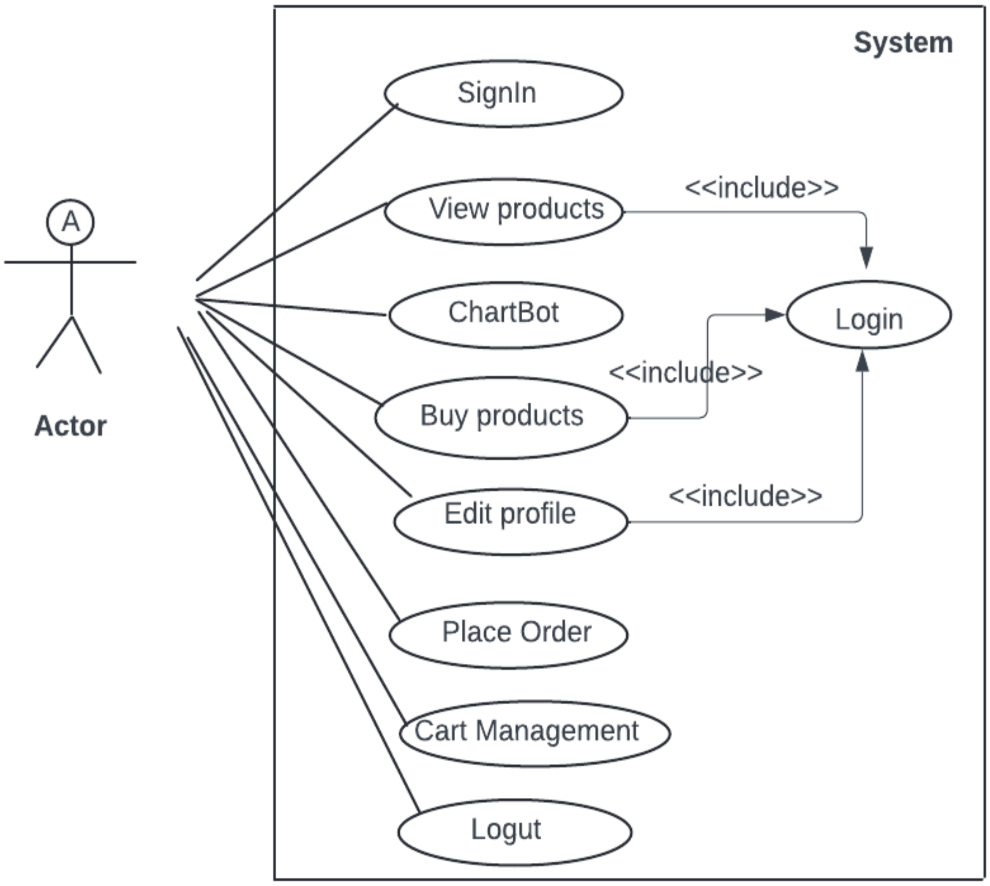
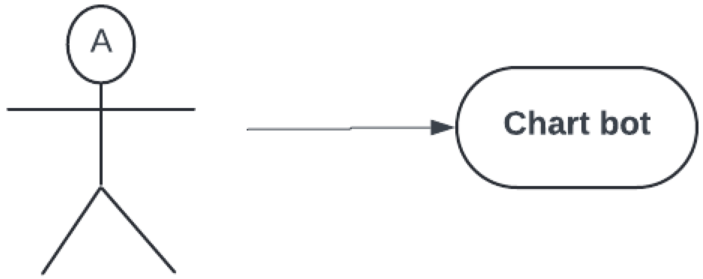


**Figure**:Use case Diagram of Consumer.

### 5. Chatbot Assistance

- **Interactive Help**: The chatbot will help farmers with questions about farming, managing their products, or technical issues related to the platform.
- **24/7 Availability**: The chatbot will be available round-the-clock to assist users.



Authorised login will access chartbot.

# 4. System Features

### 4.1 Seller Profile Management

- **Priority**: High
- **Functional Requirements**:
  - Add/delete products with images and descriptions.
  - Edit personal and company details.
  - Update address and contact information.

### 4.2 Farmer Profile Management

- **Priority**: High
- **Functional Requirements**:
  - Add/delete grains for sale.
  - Purchase seeds or pesticides.
  - Rent machines and specify rental periods.
  - View weather updates using GPS by clicking on an interactive map.

### 4.3 Consumer Profile Management

- **Priority**: High
- **Functional Requirements**:
  - Edit profile and addresses.
  - Browse and purchase products from farmers.
  - Manage cart: add, remove, or modify product quantities.

### 4.4 Weather GPS

- **Priority**: Medium
- **Functional Requirements**:
  - Access location data using GPS to provide real-time weather updates.
  - Display weather information on a map based on the farmer’s current location.

### 4.5 Chatbot Assistance

- **Priority**: Medium
- **Functional Requirements**:
  - Provide natural language responses to user queries.
  - Assist with farming tips, platform navigation, and troubleshooting.

# 5. Non-functional Requirements

### 1. Performance

- The system must handle at least 1,000 concurrent users without performance degradation.

### 2. Security

- Ensure secure transactions using SSL encryption.
- Implement role-based access control (RBAC) to protect sensitive data.

- Personal data must comply with local privacy laws (e.g., GDPR).

**3. Usability**

- The platform must be easy to use, especially for non-technical farmers.
- Provide tooltips and tutorials to guide first-time users.

**4. Availability and Reliability**

- The platform must ensure 99.9% uptime, minimizing downtime for farmers during peak seasons.

## 6. Appendices

- **Use Case Diagrams**: Visual representation of user interactions with the system.
- **Glossary**: A list of specialized terms used in agriculture or technology within the application

# Agri-VisioN: Agriculture Based Web Application

**A Project Report**

*Submitted by*

**Depavath Naresh [B200391]**

**Mudavath Kalyan [B201174]**

**Mudapally Aravind [B200778]**

Under the Guidance of

**U Nagamani**

**Department of Computer Science and Engineering**

Rajiv Gandhi University of Knowledge Technologies, Basar

Telangana – 504107

# CERTIFICATE

This is to certify that **Depavath Naresh (ID No. B200391)**, **Mudavath Kalyan (ID No. B201174)**, and **Mudapally Aravind (ID No. B200778)** have successfully completed the project titled **"Agri Vision: Agriculture Based Web Application"** at **RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES** under my supervision and guidance in the fulfillment of requirements for the **V semester, Bachelor of Technology** of RGUKT, Basar.

**Signature of the Guide** **Head of the Department**

**Signature of Examiners:**

1) ______________________________________________

2) ______________________________________________

# ACKNOWLEDGEMENT

I express my gratitude to my guide **Ms.Nagamani Uddamari**, Department of Computer Science and Engineering, for her encouragement, valuable suggestions, and guidance during the design, development, and implementation of this project.

I also express my gratitude to **Dr. Venkat Ramana**, Head of the Department, Department of Computer Science and Engineering, for his valuable guidance and encouragement during the course of this project and helping us complete it successfully.

I express my grateful thanks to all the staff members of the **Department of Computer Application** for their valuable assistance, encouragement, and cooperation during this wonderful learning experience.

Finally, I am also grateful to my parents and friends for their support, encouragement, and backing to achieve the goal of completing this project successfully.

# Contents

# Declaration

We hereby declare that the project report entitled **“Agri Vision: Agriculture Based Web Application”** is an authentic record of our own work carried out under the supervision of **MRS.NAGAMANI**, Assistant Professor, Department of Computer Science and Engineering, RGUKT, Basar.

Place: Basar
Date: December 2024

**Depavath Naresh (B200391)**
**Mudavath Kalyan (B201174)**
**Mudapally Aravind (B200778)**

# Abstract

**Agri Vision** is an innovative web-based application designed to transform the agricultural landscape by bridging the gap between farmers, consumers, and sellers on a single platform. The primary objective of this platform is to eliminate the traditional intermediaries that often exploit both farmers and consumers, leading to unfair pricing and inefficiencies in the supply chain. By creating a direct connection between the stakeholders, the platform ensures fair pricing mechanisms, better market access, and seamless communication.

The application caters to the specific needs of its users through role-specific profiles, ensuring personalized experiences for farmers, consumers, and sellers. Farmers can showcase their produce, access market trends, and connect with buyers directly, while consumers benefit from fresh produce at reasonable prices. Sellers can use the platform to expand their business by connecting with a broader customer base.

A key feature of the platform is its e-commerce capability, enabling users to buy and sell agricultural products with ease. The integration of real-time weather updates allows farmers to make informed decisions regarding crop planning, harvesting, and storage. Additionally, the platform is equipped with an AI-powered chatbot to provide 24/7 assistance, answering user queries, and offering recommendations to improve their overall experience.

**Agri Vision** leverages cutting-edge technology to address the persistent challenges in the agricultural sector, fostering transparency, efficiency, and sustainability. By empowering farmers and ensuring fair trade practices, the platform aspires to build a thriving agricultural ecosystem that benefits all stakeholders.

**Keywords:** Agriculture, Web Application, E-commerce, Farmers, Consumers, Sellers, Transparency, Sustainability, AI Assistance

# Chapter 1

# Introduction

## 1.1 Objectives

The *Agri Vision* project aims to solve critical challenges faced by the agricultural sector. The key objectives include:

- **Empower Farmers:**
  - Provide farmers with a platform to list their products, reach a wider audience, and access fair pricing without relying on intermediaries.
  - Enable farmers to connect directly with consumers and sellers, thereby improving market access and income.
- **Streamline Transactions:**
  - Create an efficient and secure marketplace for farmers, consumers, and sellers to buy and sell agricultural products.
  - Simplify the transaction process by integrating secure payment gateways and real-time order tracking.
- **Weather Insights:**
  - Integrate real-time weather updates to help farmers plan their activities effectively, reducing risks from unforeseen weather events.
  - Provide location-based weather forecasts, enabling farmers to make informed decisions about planting, irrigation, and harvesting.

- **Consumer Access:**
  - Enable consumers to directly purchase fresh, organic produce from farmers, bypassing middlemen.
  - Provide an easy-to-use platform for consumers to browse, compare, and buy agricultural products at fair prices.

## 1.2 Scope

*Agri Vision* aims to bridge the gap between farmers, sellers, and consumers by leveraging technology to eliminate inefficiencies. The platform addresses challenges such as market fragmentation, lack of information, and poor communication among stakeholders.

### Bridging the Communication Gap

Farmers, consumers, and sellers can interact directly using role-specific profiles:

- **Farmers:** List products with pricing, quantity, and quality details.
- **Consumers:** Purchase fresh produce directly and negotiate prices.
- **Sellers:** Market products such as fertilizers and agricultural tools.

Real-time notifications ensure smooth communication regarding orders, availability, and weather updates.

### Reducing Market Fragmentation

*Agri Vision* consolidates fragmented markets into a single, accessible platform:

- Provides farmers with access to distant buyers.
- Reduces dependency on intermediaries by integrating e-commerce tools.
- Standardizes pricing and quality benchmarks across regions.

## Empowering Farmers

Farmers gain access to tools and information that optimize operations:

- **Market Analytics:** Insights into pricing trends and demand.
- **Inventory Management:** Track stock and manage harvests efficiently.
- **Weather Updates:** Forecasts aid in planning and decision-making.

## Enhancing Consumer Access

Consumers can find and purchase fresh produce conveniently:

- Search filters for quality, region, and availability.
- A user-friendly interface for browsing, ordering, and tracking shipments.

## Streamlining Operations for Sellers

*Agri Vision* simplifies operations for sellers of equipment and fertilizers:

- **Digital Catalogs:** Showcase products with details and pricing.
- **Order Tracking:** Real-time updates on inventory and fulfillment.
- **Payment Integration:** Secure gateways for seamless transactions.

## Promoting Sustainability

The platform reduces inefficiencies to create a sustainable ecosystem:

- Direct sales reduce transportation, carbon emissions, and food spoilage.
- Encourages organic farming by supporting modern techniques and practices.

# Chapter 2

# System Architecture

## 2.1 Overview

The *Agri Vision* system comprises three main layers:

- **Frontend:** Built with React.js for an interactive user experience.
- **Backend:** Node.js with Express.js handles business logic, APIs, and authentication.
- **Database:** MongoDB stores user profiles, products, orders, and weather data.

## 2.2 Components

1. **User Management:** Role-specific features for farmers, sellers, and consumers.
2. **Interactive Maps:** Real-time weather and location-based data using OpenStreetMap and OpenWeather APIs.
3. **E-commerce Module:** Secure and seamless payment gateways with PayPal integration.
4. **Chatbot:** AI-powered chatbot for user assistance and query resolution.

# Chapter 3

# Database Design

## 3.1 Entity-Relationship Diagram (ERD)

The database schema for *Agri Vision* is designed to ensure optimal performance and scalability. It includes the following entities:

- **Users:** This table stores the details of all users on the platform, including:
  - **userID:** A unique identifier for each user.
  - **name:** Full name of the user.
  - **address:** Address details for delivery and location-based services.
  - **role:** Specifies the role of the user (e.g., Farmer, Consumer, Seller).
  - **contact:** Contact information such as phone number and email.
- **Products:** This table holds the inventory details for farmers and sellers:
  - **productID:** Unique ID assigned to each product.
  - **category:** Type of product (e.g., Vegetables, Fertilizers, Tools).
  - **price:** The selling price of the product.
  - **availability:** Stock information (e.g., quantity available).
  - **sellerID:** Links the product to the seller who listed it.
- **Orders:** Tracks order-related information, including:
  - **orderID:** Unique identifier for each order.

  - **status:** Status of the order (e.g., Pending, Completed, Canceled).
  - **consumerID:** Links the order to the consumer who placed it.
  - **paymentInfo:** Payment details, including transaction IDs and modes.
  - **deliveryDate:** Scheduled date for order delivery.

- **Weather Data:** This table provides weather-related information for users:
  - **locationID:** A unique ID representing the geographical location.
  - **temperature:** Current temperature readings.
  - **rainfall:** Rainfall predictions or historical data.
  - **forecast:** Short-term and long-term weather forecasts.

## 3.2 Relationships

The relationships between entities ensure smooth data flow and integrity:

- **Farmers and Products:** Each farmer can list multiple products, linking the **Users** table to the **Products** table via **sellerID**.
- **Consumers and Orders:** A consumer can place multiple orders, linking the **Users** table to the **Orders** table via **consumerID**.
- **Weather Data and Locations:** Weather data is tied to user-specific locations, enabling tailored weather information for each user.

# Chapter 4

# Class Design

## 4.1 Object-Oriented Approach

The system design follows an object-oriented approach for modularity and reusability. Key classes include:

- **User Class:**
  - Attributes:
    - **userID:** Unique identifier for users.
    - **name, role, and contact:** Core user details.
    - **address:** Stores user location for service delivery.
  - Methods:
    - **registerUser():** Handles user registration.
    - **updateProfile():** Allows users to update their details.
- **Product Class:**
  - Attributes:
    - **productID, category, price, availability, sellerID.**
  - Methods:
    - **addProduct():** Enables sellers to list new products.
    - **updateStock():** Updates stock availability after a sale.

- **Order Class:**
  - Attributes:
    - **orderID, status, consumerID, paymentInfo, deliveryDate.**
  - Methods:
    - **createOrder():** Handles new order creation.
    - **trackOrder():** Provides updates on order status.
- **Weather Class:**
  - Attributes:
    - **locationID, temperature, rainfall, forecast.**
  - Methods:
    - **getWeatherData():** Fetches current weather updates.
    - **generateForecast():** Provides tailored weather predictions.

# Chapter 5

# Frontend Design

## 5.1 Features

The frontend design of *Agri Vision* prioritizes user accessibility and functionality:

- **Role-Based Dashboards:**
  - Farmers can manage their product listings and view weather data.
  - Consumers have access to product search, purchase, and tracking features.
  - Sellers can manage inventory, track orders, and update product details.
- **Cart and Checkout Pages:**
  - A user-friendly shopping cart allows consumers to add, edit, or remove items.
  - Secure checkout integrates with payment gateways like PayPal and Razorpay.
- **Interactive Weather Map:**
  - Displays real-time weather data for specific locations.
  - Uses OpenWeather API for accurate weather forecasts.

## 5.2 Technologies Used

The frontend of *Agri Vision* uses modern technologies to create a user-friendly and efficient platform. The main technologies include:

- **React.js:**
  - Used to build an interactive and responsive user interface.
  - Helps in creating reusable components, like role-based dashboards and weather maps.
  - Makes the platform faster by using a virtual DOM for updates.
- **Bootstrap:**
  - Makes the platform look neat and professional with pre-designed layouts and components.
  - Ensures the design works well on all devices, like desktops, tablets, and smartphones.
- **API Integration:**
  - Connects the frontend to the backend and external services, like weather APIs.
  - Helps farmers get real-time weather updates.
  - Provides secure online payments using trusted payment gateways.

—

# Chapter 6

# Future Scope

In the future, *Agri Vision* can grow and improve with these features:

- **Mobile App:**
  - A mobile app will make the platform easy to use for everyone, even in rural areas.
  - Offline features will let farmers access important information without an internet connection.
- **AI-Based Recommendations:**
  - AI can suggest the best crops to plant, when to plant them, and how to manage pests based on weather and market data.
  - It can also help sellers understand what customers want.
- **Language Support:**
  - Adding support for local languages will make the platform accessible to people from different regions.

These improvements will make *Agri Vision* even more helpful, inclusive, and powerful for the agricultural community.

# Chapter 7

# Software Testing

Software testing is the process of evaluating the software against the requirements gathered from users and system specifications. Testing is conducted at various levels of the software development lifecycle (SDLC) or at the module level in the program code. It helps ensure the software behaves as expected and meets the required functional and non-functional specifications. Testing in *Agri Vision* follows a structured approach that includes both validation and verification processes.

## 7.1 Unit Testing

Unit testing is a technique in which individual software modules or components are tested in isolation to ensure that they function correctly. In the context of *Agri Vision*, unit testing is performed by the developers themselves to detect any issues in the standalone units or components of the application before they are integrated with other components. The main goal is to ensure the functional correctness of these modules and to identify defects early in the development phase.

### Advantages of Unit Testing:

- **Reduces Defects:** Unit testing ensures that newly developed features function correctly and reduces bugs when changing existing functionality.
- **Cost-Effective:** It helps detect defects early in the development process, reducing the overall cost of testing.

- **Improved Design:** Unit testing encourages better software design by identifying potential design flaws early on.
- **Better Refactoring:** Allows for safe refactoring of code by ensuring that existing functionality is not broken.
- **Quality Assurance:** When integrated with the build process, unit tests provide immediate feedback on the quality of the build.

### Unit Testing Techniques:

Unit testing in *Agri Vision* uses the following techniques:

- **Black Box Testing:** This technique focuses on testing the user interface, input, and output functionality without knowledge of the internal workings of the module. In *Agri Vision*, black box testing is used to validate the functionality of key features like product listings, cart functionality, and order management.
- **White Box Testing:** White box testing is used to test the internal functions and logic of the modules. This includes checking the correctness of algorithms used for product pricing, user authentication, and payment processing.

## 7.2 Black Box Testing

Black box testing is a software testing method where the functionality of an application is tested based on its specifications, without any knowledge of the internal code or structure. In *Agri Vision*, black box testing is used to ensure that the system functions according to the specified requirements.

### Advantages of Black Box Testing:

- **Specification-Based Testing:** Black box testing focuses on the software's functionality from an end-user perspective, ensuring that the system meets all functional requirements and behaves as expected.

- **Independent Testing:** Black box testing is typically performed by independent testers who are not involved in the development of the application, ensuring unbiased results.
- **Applicable to All Levels:** Black box testing can be applied at various stages of software testing, including unit, integration, system, and acceptance testing.

### Testing Approach for *Agri Vision*:

In *Agri Vision*, black box testing is performed across several areas:

- **User Interface Testing:** Validates that the user interface (UI) is intuitive and functions as expected, such as browsing products, adding items to the cart, and checking out.
- **Functional Testing:** Ensures that all features of the system, including product listings, weather updates, and payment gateways, perform as expected.
- **End-to-End Testing:** Simulates real-world user scenarios, such as a consumer purchasing products directly from farmers and processing payments.

## 7.3 Testing Sequence for *Agri Vision*

The testing sequence for *Agri Vision* follows a structured approach to ensure comprehensive validation of the software's functionality:

- **Unit Testing:** Each module (e.g., user registration, product listing, payment processing) is tested individually to ensure correctness before integration.
- **Integration Testing:** After individual module testing, integration testing is conducted to ensure that different modules interact correctly, such as the communication between the user profile and the shopping cart system.

- **System Testing:** The entire system is tested as a whole to ensure that all modules work together seamlessly and the system functions as expected. This includes testing of end-to-end workflows like product purchasing, order tracking, and weather updates.
- **User Acceptance Testing (UAT):** The system is tested by real users (farmers, consumers, and sellers) to ensure that it meets their needs and expectations before going live.

## 7.4 Conclusion of TESTING

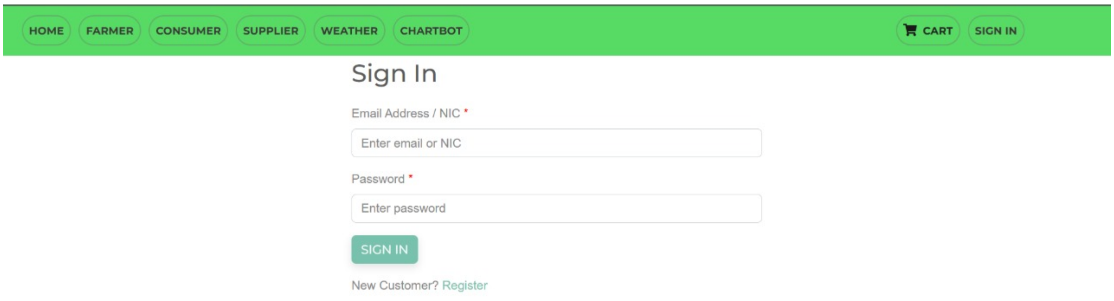
Software testing plays a crucial role in ensuring the success of the *Agri Vision* platform. By performing unit testing, black box testing, and integration testing at different stages of the software development life cycle, the platform is ensured to function as intended. These testing strategies help identify and resolve issues early in development, leading to a more stable and reliable product for end users. The detailed and systematic approach to testing ensures that the *Agri Vision* platform delivers a seamless, user-friendly experience for farmers, consumers, and sellers.

# Chapter 8

# Snapshots of the System

In this chapter, we present snapshots of key features of the *Agri Vision* platform, illustrating its functionality and user interface.

## 8.1 Login Page



Sign In

Email Address / NIC *

Enter email or NIC

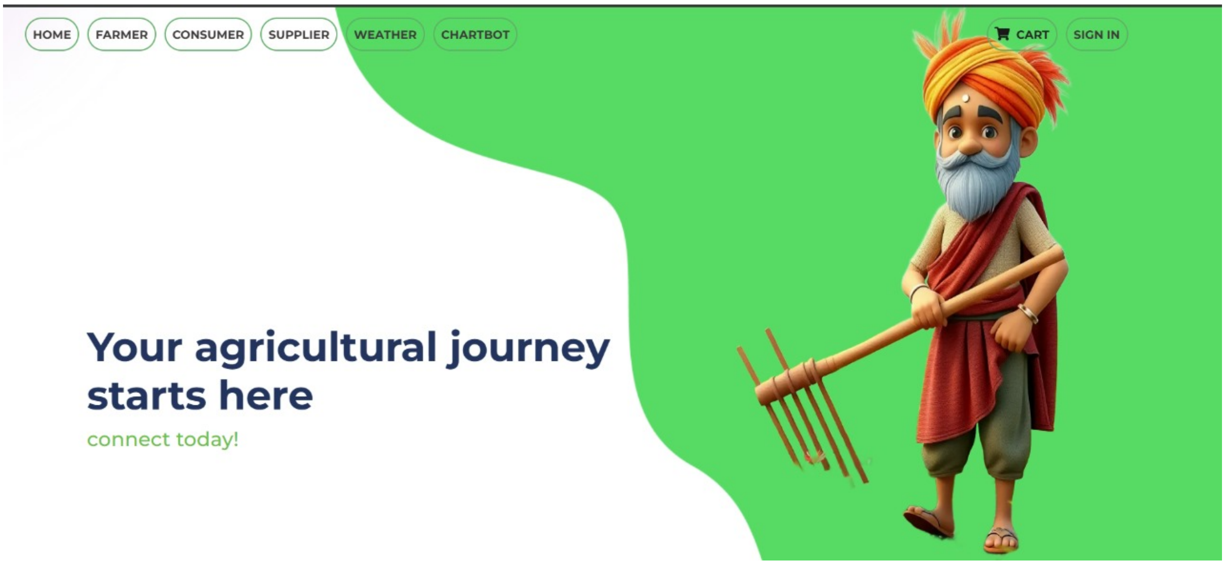
Password *

Enter password

New Customer? Register

Figure 8.1: Login Page: Allows users to access their accounts securely by entering their credentials.

## 8.2 Homepage (Hero Section)



Your agricultural journey starts here

connect today!

Figure 8.2: Homepage: The hero section introduces the platform and highlights its key features.

## 8.3 Admin Dashboard

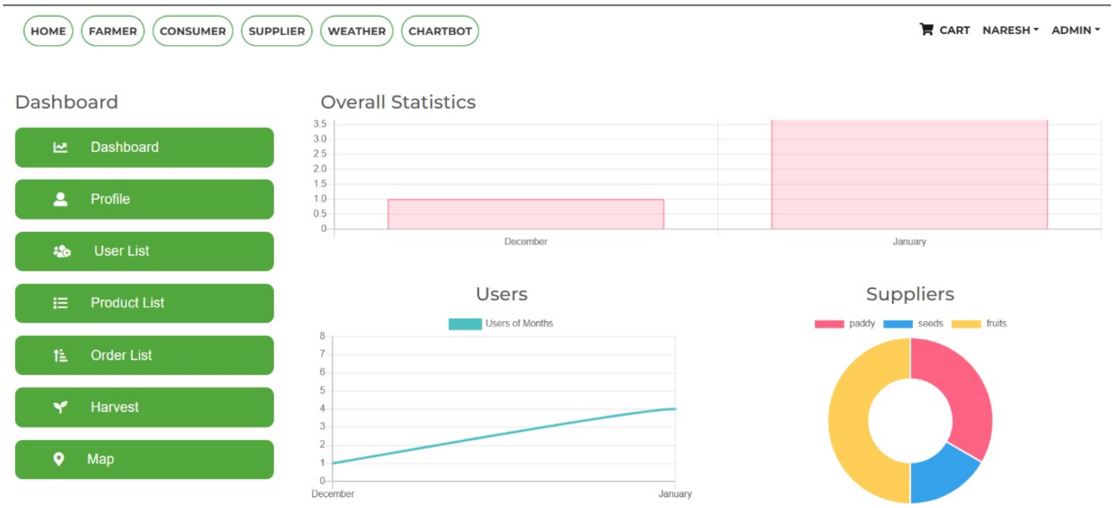
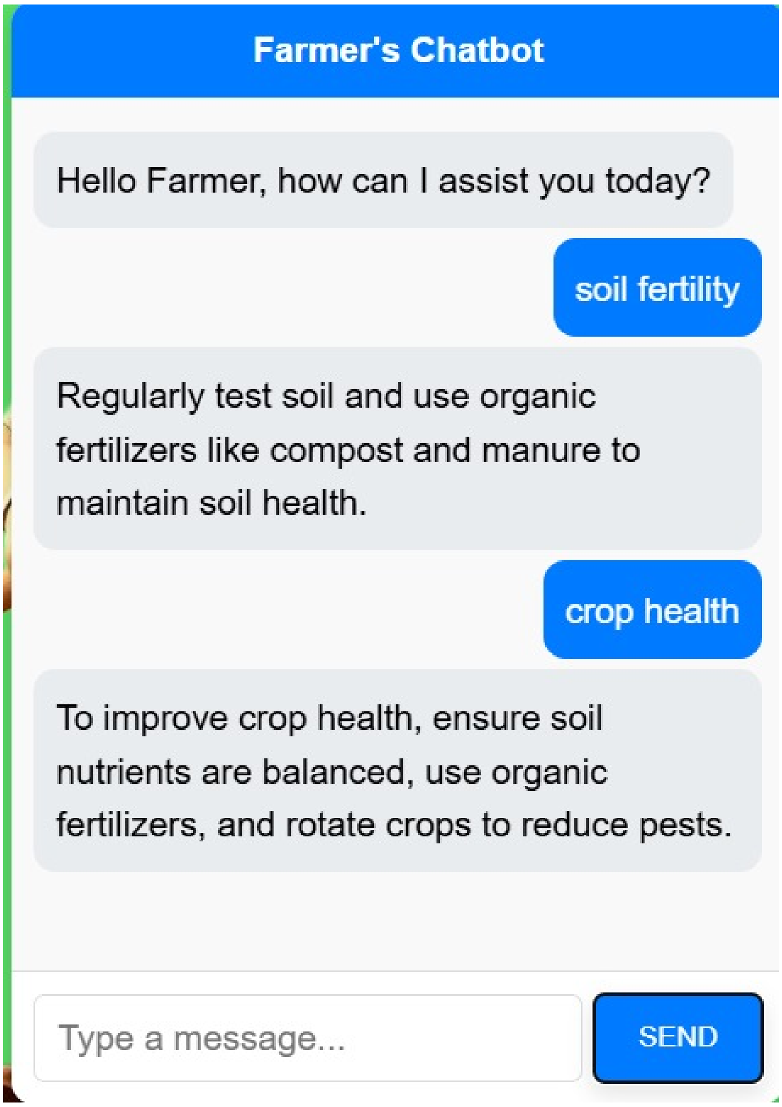


Figure 8.3: Admin Dashboard: Provides tools for managing users, products, and services.

## 8.4 Chatbot Interface



Hello Farmer, how can I assist you today?

soil fertility

Regularly test soil and use organic fertilizers like compost and manure to maintain soil health.

crop health

To improve crop health, ensure soil nutrients are balanced, use organic fertilizers, and rotate crops to reduce pests.

Figure 8.4: Chatbot Interface: Assists users in navigating the platform and resolving queries.

## 8.5 Interactive Map

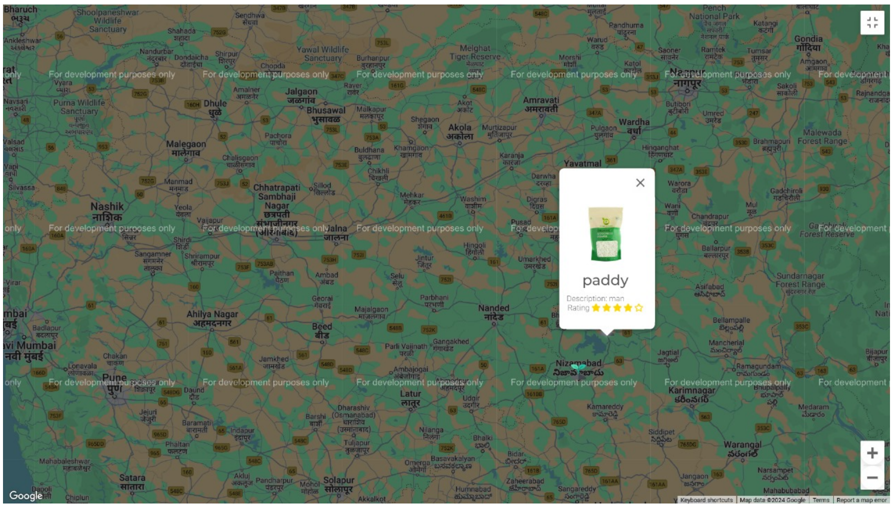
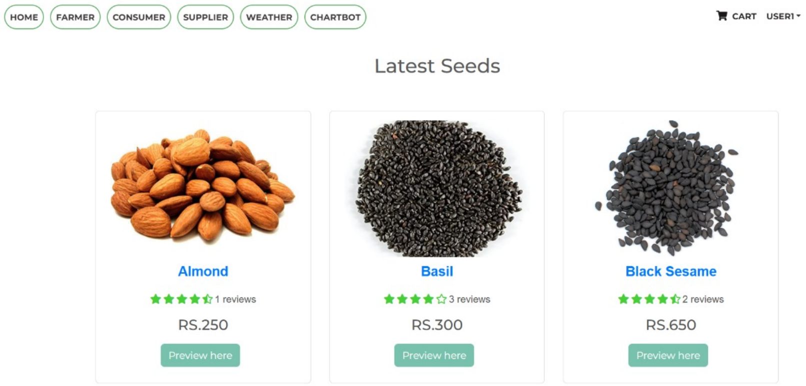


Figure 8.5: Interactive Map: Shows farmer locations and real-time weather updates using APIs.

## 8.6 Products Page



Latest Seeds

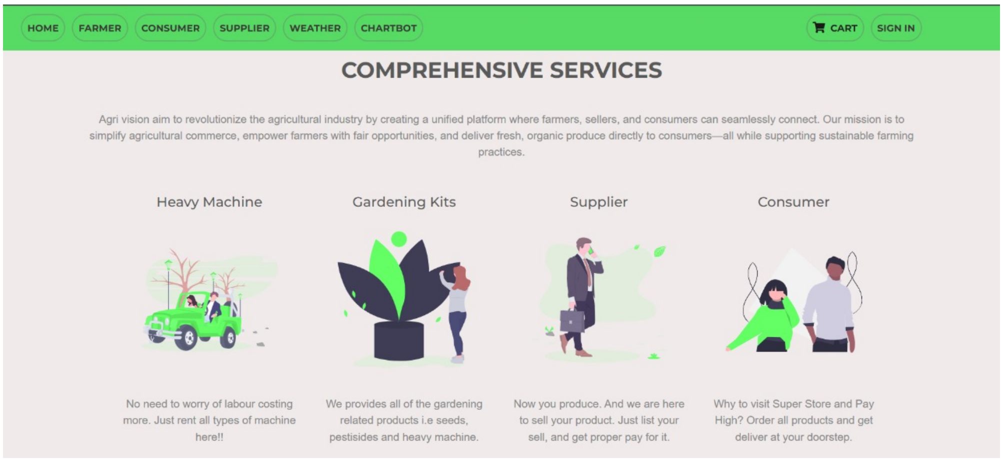
Almond
★★★★☆ 1 reviews
RS.250
Preview here

Basil
★★★★☆ 3 reviews
RS.300
Preview here

Black Sesame
★★★★☆ 2 reviews
RS.650
Preview here

Figure 8.6: Products Page: Allows users to browse available products with details like price and quantity.

## 8.7 Services Page



COMPREHENSIVE SERVICES

Agri vision aim to revolutionize the agricultural industry by creating a unified platform where farmers, sellers, and consumers can seamlessly connect. Our mission is to simplify agricultural commerce, empower farmers with fair opportunities, and deliver fresh, organic produce directly to consumers—all while supporting sustainable farming practices.

Heavy Machine

No need to worry of labour costing more. Just rent all types of machine here!!

Gardening Kits

We provides all of the gardening related products i.e seeds, pestisides and heavy machine.

Supplier

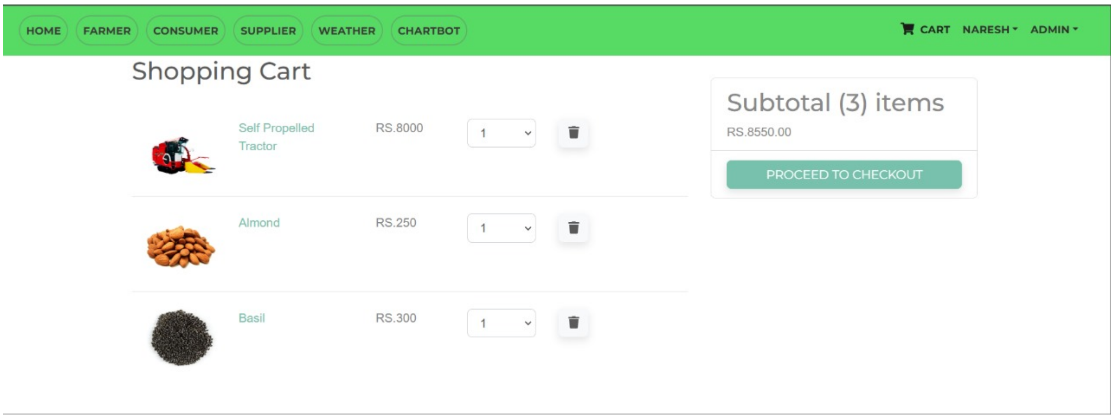
Now you produce. And we are here to sell your product. Just list your sell, and get proper pay for it.

Consumer

Why to visit Super Store and Pay High? Order all products and get deliver at your doorstep.

Figure 8.7: Services Page: Displays the range of services provided by the platform, including product listing, weather updates, and more.

## 8.8 Cart Page



Shopping Cart

| | | |
|---|---|---|
| Self Propelled Tractor | RS.8000 | 1 |
| Almond | RS.250 | 1 |
| Basil | RS.300 | 1 |

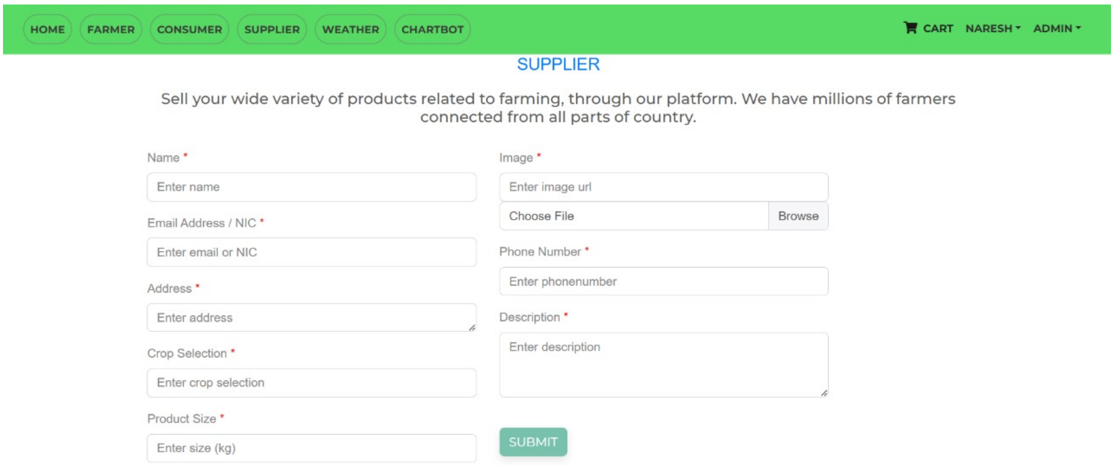
Subtotal (3) items

RS.8550.00

PROCEED TO CHECKOUT

Figure 8.8: Cart Page: Enables consumers to view and manage selected products before checkout.

## 8.9 Supplier Dashboard



SUPPLIER

Sell your wide variety of products related to farming, through our platform. We have millions of farmers connected from all parts of country.

Name *
Enter name

Email Address / NIC *
Enter email or NIC

Address *
Enter address

Crop Selection *
Enter crop selection

Product Size *
Enter size (kg)

Image *
Enter image url
Choose File Browse

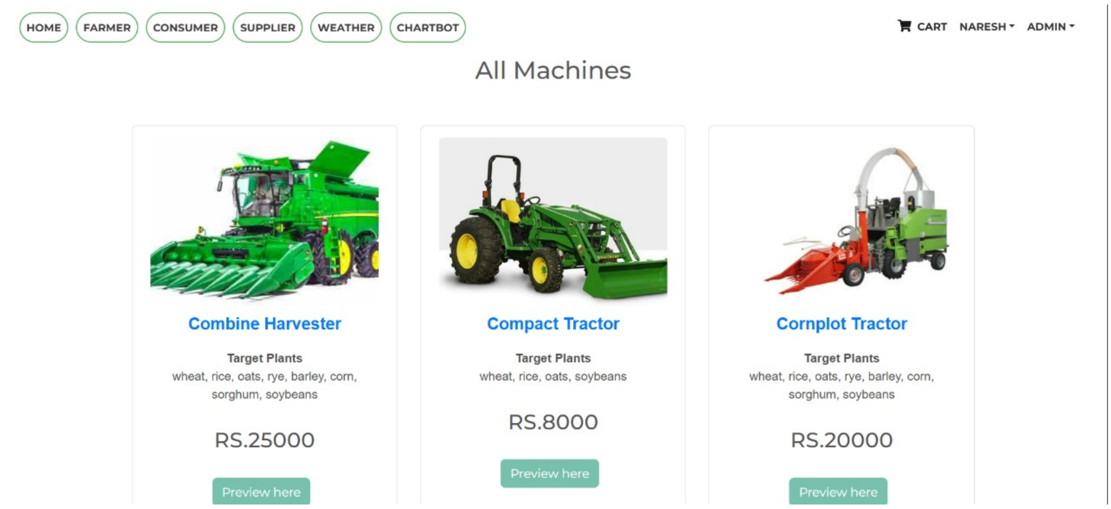
Phone Number *
Enter phonenumber

Description *
Enter description

SUBMIT

Figure 8.9: Supplier Dashboard: Allows suppliers to manage their product inventory and orders efficiently.

## 8.10 Machine management and Lending system



All Machines

**Combine Harvester**
Target Plants
wheat, rice, oats, rye, barley, corn, sorghum, soybeans
RS.25000
Preview here

**Compact Tractor**
Target Plants
wheat, rice, oats, soybeans
RS.8000
Preview here

**Cornplot Tractor**
Target Plants
wheat, rice, oats, rye, barley, corn, sorghum, soybeans
RS.20000
Preview here

Figure 8.10: Machinery: Allows suppliers and farmers to lend Machines to manage inventory efficiently.

# Chapter 9

# Final conclusion and Overview

## 9.1 Conclusion

*Agri Vision* solves many problems in agriculture by bringing farmers, consumers, and sellers together on one platform. The main benefits are:

- **For Farmers:**
  - Farmers can sell their products directly to buyers, avoiding middlemen.
  - Real-time weather updates help farmers plan better and avoid losses.
- **For Consumers:**
  - Consumers get fresh produce directly from farms at fair prices.
  - They can see transparent pricing and avoid overpaying.
- **For Sellers:**
  - Sellers can manage their stock, process orders, and receive payments easily.
  - The platform helps them reach more customers and grow their business.

In summary, *Agri Vision* is a smart, user-friendly system that makes agricultural trade fair, transparent, and efficient for everyone.

—

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
BASAR,NIRMAL
TELANGANA-504107

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

# Software Design Document

for

# Agri vision

Prepared by:

Depavath Naresh [B200391 ]

Mudavath kalyan [B201174]

Mudapally Aravind. [B200778]

# Contents

# 1 Introduction

## 1.1 Project Overview

AgriVision is an innovative web application designed to revolutionize the agriculture industry by directly connecting farmers, sellers, and consumers on a single platform. It eliminates the need for intermediaries, ensuring fair pricing and transparent transactions while fostering better communication among stakeholders. The platform offers tailored features for each user role to streamline agricultural operations and enhance productivity.

Farmers can list and manage grains for sale, rent agricultural machinery, and purchase seeds or pesticides directly from sellers. They also have access to real-time weather data through an interactive map, helping them make informed decisions. Sellers can manage their profiles, add or remove products, and provide necessary agricultural supplies to farmers. Consumers can browse available products, purchase directly from farmers, and manage their shopping cart, offering them a seamless buying experience.

The application integrates a powerful AI chatbot to assist users with queries, making the platform user-friendly and accessible. The backend is built with Node.js and Express, while the frontend is developed using React. MongoDB serves as the database, with carefully structured collections to manage users, products, orders, and rentals efficiently.

To enhance the user experience, AgriVision features an intuitive design with dedicated pages for product browsing, cart management, user profiles, and order history. The interactive map for weather insights is implemented using Leaflet.js, and the entire system is deployed using Netlify and Heroku for scalability and reliability.

By connecting farmers, sellers, and consumers directly, AgriVision aims to empower stakeholders, promote transparency, and create a fair agricultural ecosystem. It is a comprehensive solution that leverages modern technology to address challenges in agriculture, making it a vital tool for the agricultural community.

## 1.2 Purpose and Scope

The purpose of the Agri Vision project is to create a unified platform that bridges the gap between farmers, sellers, and consumers in the agriculture industry. By providing features such as interactive weather maps, product management tools, and secure communication channels, the platform aims to streamline agricultural trade and resource management. The project eliminates the need for intermediaries, enabling farmers to sell their products at fair prices and access essential resources like seeds, pesticides, and machinery directly from sellers. Additionally, consumers can purchase high-quality agricultural products with transparency and ease. This fosters a sustainable and efficient agricultural ecosystem.

- **Vulnerability Detection**:The Agri Vision platform is a web-based solution designed to cater to the needs of farmers, sellers, and consumers.

- **Farmers**:Ability to add, delete, and manage agricultural products, view weather updates, and rent machinery or purchase seeds and pesticides..

- **Sellers**:Manage product inventories, edit profiles, and list products for sale to farmers and consumers.

- **Consumers**: Purchase materials directly from farmers, manage cart items, and leave ratings for products.

- **Integration with External Services**:Additional features include a chatbot for user support, an interactive map for weather insights, and a secure, user-friendly interface. The platform will leverage a modular architecture (MERN stack) to ensure scalability, maintainability, and efficient performance. The scope also extends to improving trust, transparency, and communication among all stakeholders while offering a reliable and efficient system for managing agricultural operations and transactions. .

### 1.3 Audience

This document is intended for a diverse group of stakeholders, each with specific interests in the system's design. The following audiences are targeted:

- **Developers**: This document provides a detailed guide on the system's architecture, modules, and interactions, enabling developers to build, extend, and maintain the Agri Vision platform efficiently.

- **System Architects**: System architects can review the proposed design for scalability, security, and performance, ensuring it aligns with the system's goals. The document also provides insights into integration points and infrastructure requirements.

- **Project Managers**: Project managers can utilize this document to oversee project progress, allocate resources effectively, and ensure that the design meets functional and non-functional requirements within the given timeline.

- **Testers**: Testers can use the workflows and design specifications provided in this document to create test cases, ensuring that all system functionalities, including interactive maps, product management, and user roles, are thoroughly validated.

- **Agricultural Stakeholders**: This group includes farmers, sellers, and consumers who will benefit from a high-level understanding of the system's capabilities, such as product management, direct transactions, and weather-based insights.

- **End Users**: A high-level overview in this document will help end users comprehend features like chatbot interactions, weather updates, and seamless buying/selling of agricultural products.

### 1.4 Key Features and Benefits

Agri Vision is designed with several features to connect farmers, sellers, and consumers, ensuring efficient resource management and direct transactions:

- **Interactive Weather Map**: Users can click on a map to access real-time weather updates for their specific location, aiding in better agricultural planning and decision-making.

- **Chatbot Assistance**: The integrated chatbot provides instant support for users, answering queries about the platform's features, processes, and navigation.

- **Role-Based Profiles**: Separate profiles for farmers, sellers, and consumers allow tailored functionalities such as adding/deleting products, renting machines, and managing orders.

- **Streamlined Transactions**: Farmers can sell grains directly to consumers, purchase seeds/pesticides, or rent machinery without intermediaries, ensuring fair pricing and efficiency.

- **Scalable Platform**: Agri Vision is designed to accommodate multiple users and transactions, making it suitable for diverse agricultural communities.

- **Secure and User-Friendly Design**: The platform ensures data privacy for all users while offering a simple interface to encourage widespread adoption and ease of use.

## 1.5 System Requirements

The system operates within the following requirements:

- **Software Requirements**:
  - Server: Linux or Windows-based environment (Ubuntu, CentOS, or Windows Server)
  - Backend: Node.js with Express.js for API handling
  - Frontend: React.js for building dynamic and interactive user interfaces
  - Database: MongoDB for storing user profiles, product information, and transaction data
  - Mapping and Weather APIs: OpenStreetMap and a weather API (such as OpenWeather) for real-time location and weather updates

- **Hardware Requirements**:
  - A server with at least 4 GB of RAM and a multi-core processor for handling concurrent user requests
  - Internet connectivity to fetch weather data, interact with APIs, and support real-time chat features

## 1.6 System Limitations

Although Agri Vision provides a comprehensive platform for the agricultural sector, it has the following limitations:

- **API Dependency**: The system relies on third-party APIs for weather and mapping data. Any changes or outages in these services can impact the functionality.

- **Resource Constraints**: Real-time updates and transactions may experience delays under heavy user loads or insufficient server resources.

- **Geographical Limitations**: Weather data and mapping services may not provide detailed or accurate information for remote or underdeveloped regions.

- **User Onboarding**: Initial adoption may be limited by the digital literacy of the target audience, particularly farmers unfamiliar with technology.

# 2 System Architecture

## 2.1 High-Level Overview

Agri Vision consists of several key components that work together to provide a unified platform for farmers, sellers, and consumers:

- **Frontend**: The user interface (UI) for interactions, built using React.js. It includes components for maps, weather information, user profiles, and product management.
- **Backend**: The server-side logic is implemented with Node.js and Express.js, handling API requests, authentication, and business logic.
- **Database**: MongoDB is used to store user profiles, product details, orders, and transaction history in a scalable and efficient manner.
- **Mapping and Weather Services**: OpenStreetMap integrated with Leaflet provides an interactive map interface, while a weather API fetches real-time weather data.
- **ChatBot Service**: AI-powered chatbot for assisting users in navigating the platform and addressing their queries.

## 2.2 Technology Stack

- **Frontend**: React.js for creating a dynamic and interactive user experience.
- **Backend**: Node.js and Express.js for server-side processing and API management.
- **Database**: MongoDB for managing data related to users, products, orders, and weather reports.
- **Mapping Service**: OpenStreetMap and Leaflet.js for rendering maps and enabling location-based features.
- **Weather Service**: Integration with a weather API such as OpenWeather for real-time weather updates.
- **ChatBot**: AI chatbot built with frameworks like Dialogflow or Rasa for user assistance.

# 3 Database Design

## 3.1 Entity-Relationship Diagram (ERD)

The **Entity-Relationship Diagram (ERD)** defines the structure of the database, the relationships between various entities, and their attributes. Below is the diagram that explains the database structure for Agri Vision:

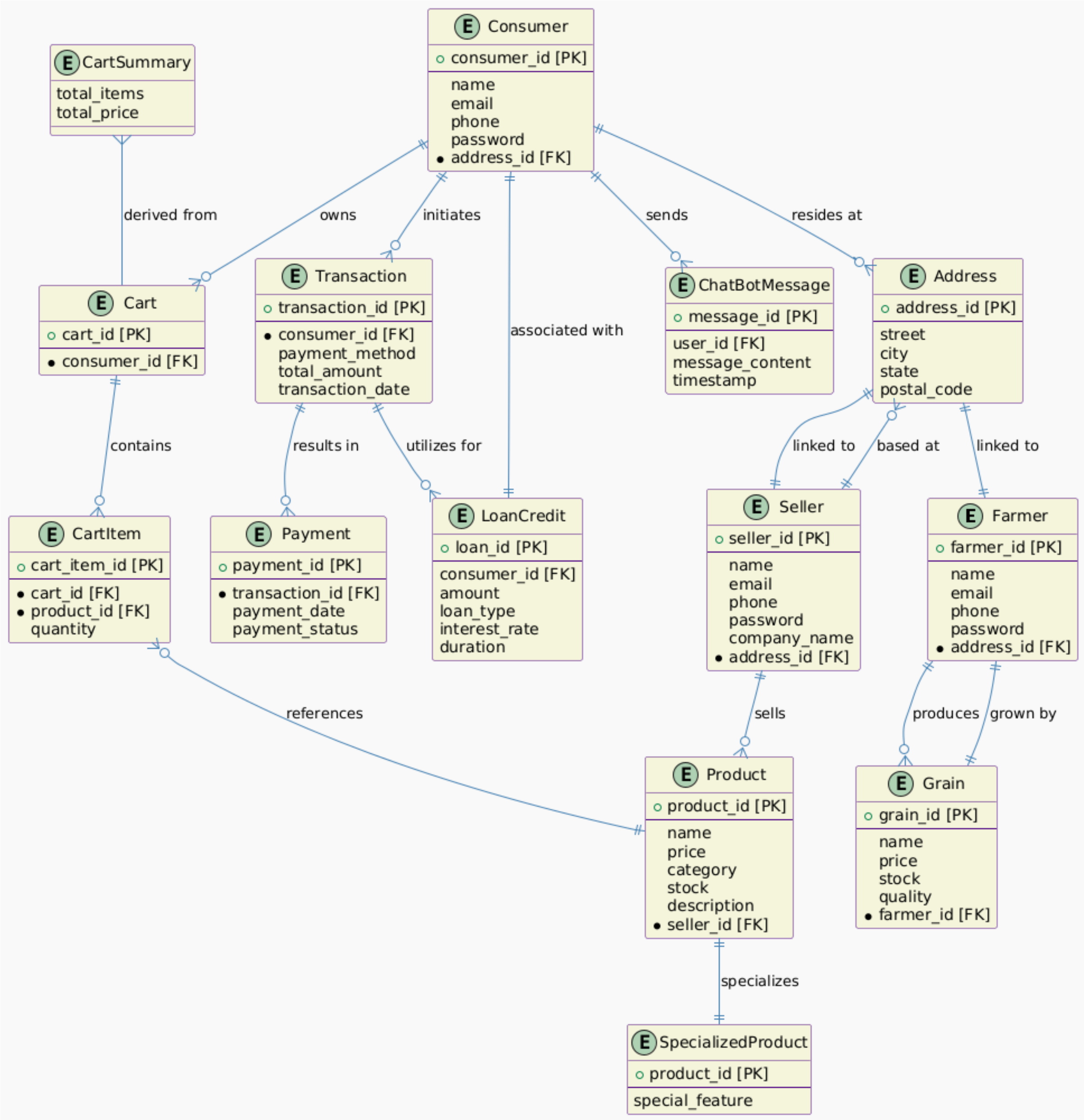


Figure 1: Main Entity-Relationship Diagram (ERD)

**Explanation**:

- **User Entity**: Represents all system users, including farmers, sellers, and consumers, with attributes like `userID`, `name`, `email`, `password`, `role`, and `address`.

- **Product Entity**: Represents products listed for sale by farmers or sellers with attributes like `productID`, `name`, `description`, `category`, `price`, and `quantity`.
- **Order Entity**: Represents orders placed by consumers, including attributes like `orderID`, `productID`, `consumerID`, `quantity`, and `orderStatus`.
- **Weather Data Entity**: Represents weather information for various locations, including attributes like `locationID`, `latitude`, `longitude`, `temperature`, and `forecast`.
- **Rental Machine Entity**: Represents machinery available for rent, with attributes like `machineID`, `name`, `description`, `rentalPrice`, and `ownerID`.
- **ChatBot Log Entity**: Represents user interactions with the chatbot, including attributes like `logID`, `userID`, `timestamp`, and `queryResponse`.

**Relationships**:

- A `User` can list multiple `Products` or `Rental Machines`.
- A `Consumer` can place multiple `Orders`.
- Weather data is linked to geographic locations, enabling weather reporting for specific regions on the map.
- A `ChatBot Log` is associated with a specific `User`.

## 3.2 Sub-Diagrams

### 3.2.1 Farmer Entity Sub-Diagram

This diagram further breaks down the **User Entity** to show how the farmer's profile and preferences are managed.

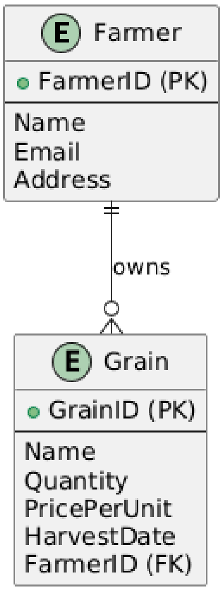


Figure 2: Farmer Entity-Relationship Diagram (ERD)

**Explanation**:

- The **Farmer Entity** includes attributes such as `farmerID`, `name`, `contactInfo`, `farmDetails`, and `grainList`.
- `FarmDetails` contains embedded fields such as farm size, type of crops grown, and location.
- Farmers can add or delete grains from the `grainList`, which is dynamically linked to their profile.

### 3.2.2 Consumer Entity Sub-Diagram

This diagram breaks down the **Order Entity**, representing how consumer orders are managed within the system.

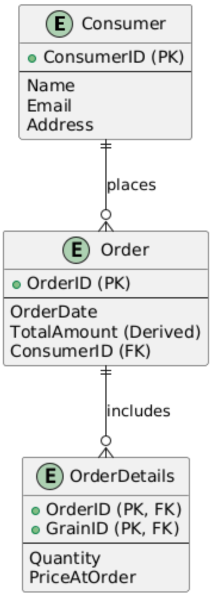


Figure 3: Consumer Entity-Relationship Diagram (ERD)

### 3.2.3 Suppliers Entity Sub-Diagram

This diagram breaks down the **Order Entity**, representing how consumer orders are managed within the system.

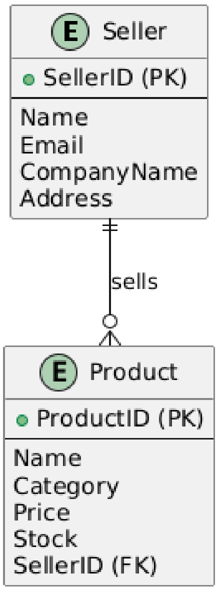


Figure 4: supplier Entity-Relationship Diagram (ERD)

**Explanation**:

- The **Order Entity** has attributes like `orderID`, `consumerID`, `productID`, `quantity`, and `orderStatus`.
- `OrderStatus` includes states such as *pending*, *dispatched*, and *delivered*.
- Each order is associated with a specific consumer, and the `consumerID` links the order back to the user's profile.

# 4 Class Design

## 4.1 Overview

The Class Design section defines the object-oriented structure of the Agri Vision system, detailing the classes, their attributes, methods, and the relationships between them. The primary goal is to ensure modularity and reusability while maintaining a clear separation of concerns between the different roles in the agricultural ecosystem (farmers, suppliers, consumers, etc.).

## 4.2 Class Diagram

The **Class Diagram** provides a high-level view of the main system classes, their relationships, and interactions within the Agri Vision platform.

**Explanation**:

- The **User Class** manages the user-related data, including farmers, consumers, and suppliers.
- The **Product Class** represents the products (grains, seeds, pesticides, etc.) listed by suppliers and available for farmers and consumers.

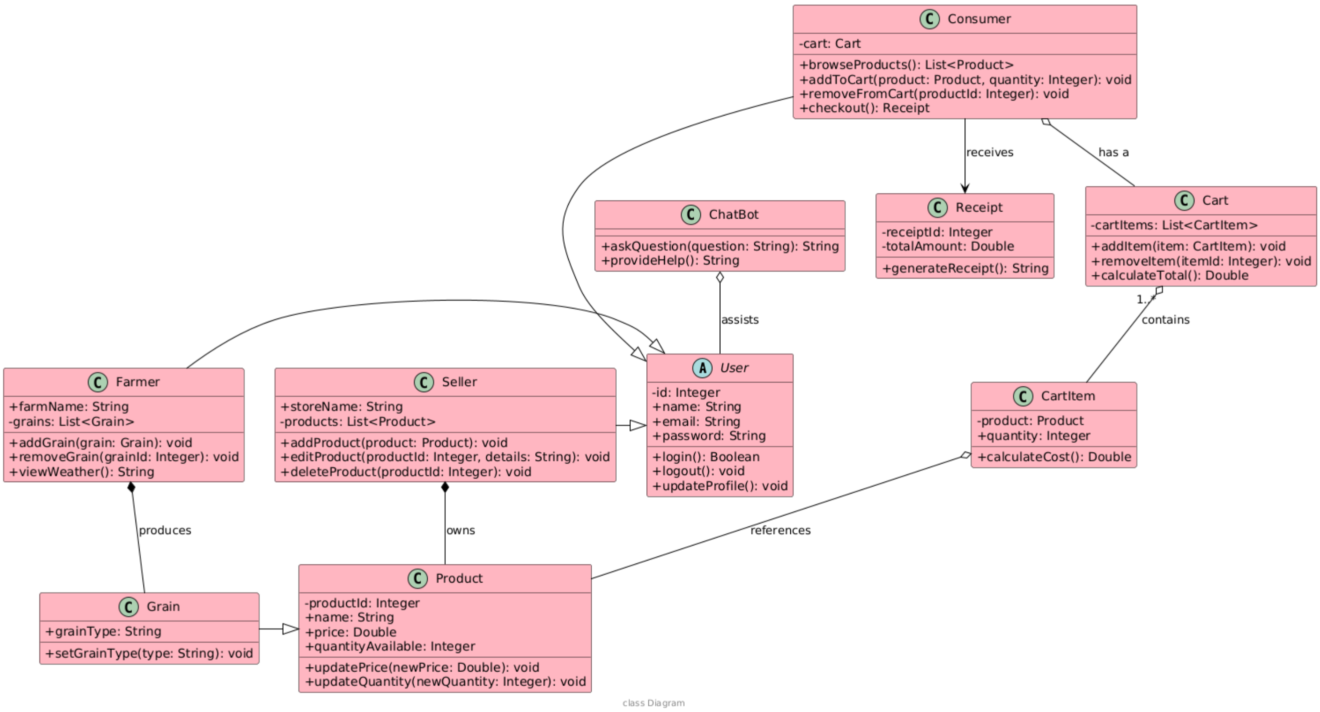

Consumer
-cart: Cart
+browseProducts(): List<Product>
+addToCart(product: Product, quantity: Integer): void
+removeFromCart(productId: Integer): void
+checkout(): Receipt
receives
has a
ChatBot
+askQuestion(question: String): String
+provideHelp(): String
Receipt
-receiptId: Integer
-totalAmount: Double
+generateReceipt(): String
Cart
-cartItems: List<CartItem>
+addItem(item: CartItem): void
+removeItem(itemId: Integer): void
+calculateTotal(): Double
1..*
contains
assists
Farmer
+farmName: String
-grains: List<Grain>
+addGrain(grain: Grain): void
+removeGrain(grainId: Integer): void
+viewWeather(): String
Seller
+storeName: String
-products: List<Product>
+addProduct(product: Product): void
+editProduct(productId: Integer, details: String): void
+deleteProduct(productId: Integer): void
User
-id: Integer
+name: String
+email: String
+password: String
+login(): Boolean
+logout(): void
+updateProfile(): void
CartItem
-product: Product
+quantity: Integer
+calculateCost(): Double
produces
owns
references
Grain
+grainType: String
+setGrainType(type: String): void
Product
-productId: Integer
+name: String
+price: Double
+quantityAvailable: Integer
+updatePrice(newPrice: Double): void
+updateQuantity(newQuantity: Integer): void
class Diagram

Figure 5: Main Class Diagram for Agri Vision

- The **Transaction Class** tracks transactions between farmers, consumers, and suppliers (e.g., purchase, sale, rental).
- The **Weather Class** provides weather data associated with a user's location (through the interactive map).
- Relationships between classes ensure proper data flow, such as the interaction between the **User Class** and the **Transaction Class** for managing purchases and sales.

### 4.3 User Class Design

**Explanation**:

- **Attributes**:
  - `userID`: A unique identifier for each user (farmer, supplier, consumer).
  - `userType`: Specifies whether the user is a farmer, supplier, or consumer.
  - `address`: The physical address of the user.
  - `preferences`: Stores user-specific preferences, such as weather notifications and product recommendations.
- **Methods**:
  - `registerUser()`: Handles user registration for different user types.
  - `updateProfile()`: Updates user details such as address or preferences.

  - `viewProductList()`: Displays available products for purchase or rent (specific to user type).

## 4.4 Product Class Design

**Explanation**:

- **Attributes**:
  - `productID`: Unique identifier for each product.
  - `productName`: The name of the product (e.g., wheat, pesticides).
  - `price`: Price of the product.
  - `category`: Specifies the type of product (e.g., grain, pesticide, machinery).
  - `availability`: Indicates whether the product is available for sale or rent.
- **Methods**:
  - `addProduct()`: Allows suppliers to add new products to the marketplace.
  - `updateProduct()`: Enables suppliers to edit product details (e.g., price, description).
  - `removeProduct()`: Allows suppliers to remove products from the marketplace.

## 4.5 Weather Class Design

**Explanation**:

- **Attributes**:
  - `location`: The geographical location (latitude, longitude) of the user.
  - `temperature`: The current temperature at the user's location.
  - `humidity`: The current humidity level at the user's location.
  - `forecast`: Weather forecast data for the next 24 hours.
- **Methods**:
  - `getWeatherData()`: Retrieves the current weather data based on the user's location.
  - `getWeatherForecast()`: Retrieves the 24-hour weather forecast for the user's location.

# 5 Interaction Design

## 5.1 Sequence Diagram

The **Sequence Diagram** provides a detailed view of how system components interact during specific operations. It visualizes the order of interactions and highlights the system's dynamic behavior.

**Explanation**:

- The sequence diagram illustrates the process of user registration in the system:
    - The user initiates the process by submitting their registration details through the front-end interface.
    - The server validates the user's input, ensuring all required fields are correctly filled and follow predefined rules.
    - Once validated, the server communicates with the database to store the user's information.
    - The user receives feedback confirming successful registration.

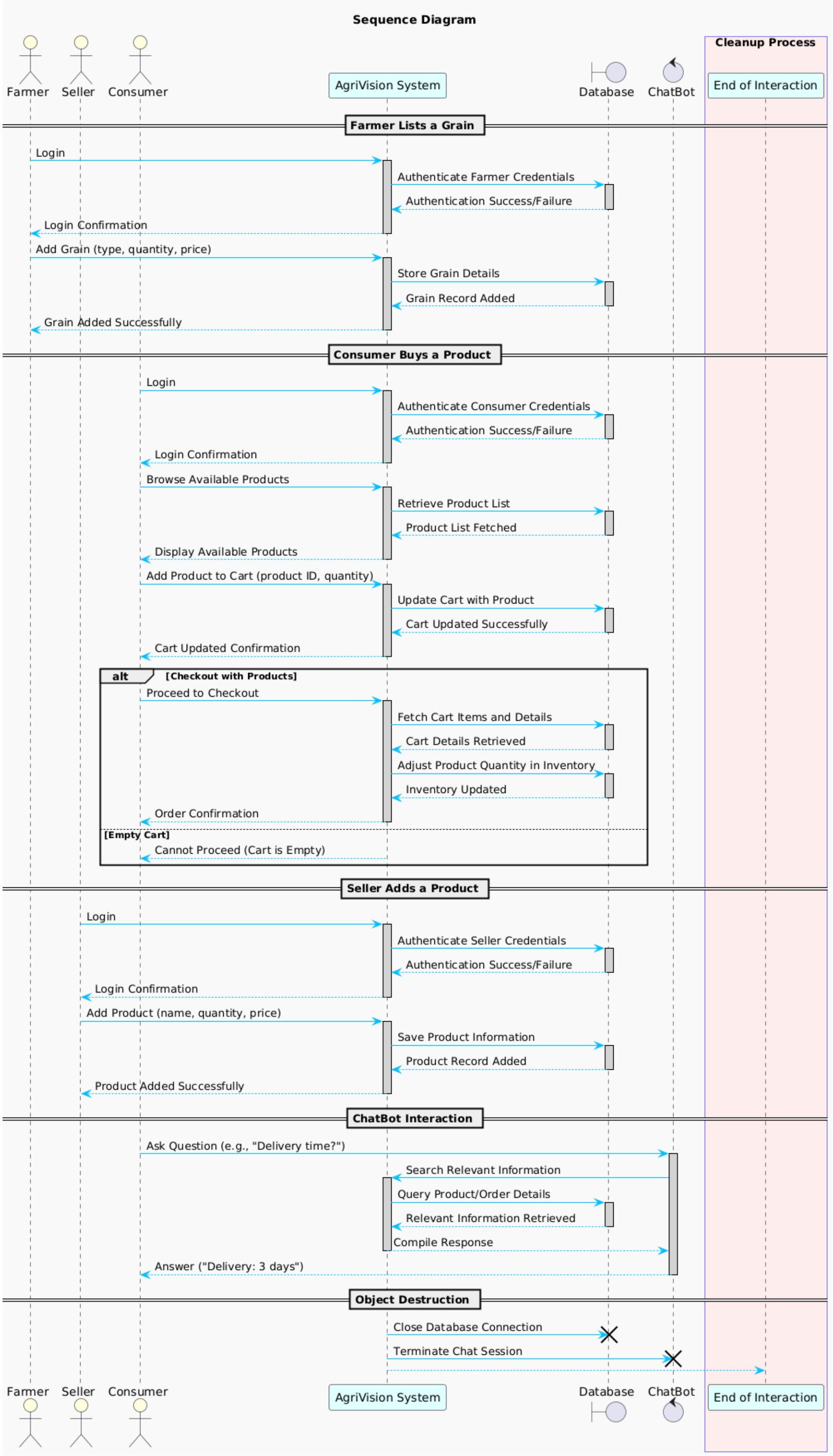

Sequence Diagram
Cleanup Process
Farmer
Seller
Consumer
AgriVision System
Database
ChatBot
End of Interaction
Farmer Lists a Grain
Login
Authenticate Farmer Credentials
Authentication Success/Failure
Login Confirmation
Add Grain (type, quantity, price)
Store Grain Details
Grain Record Added
Grain Added Successfully
Consumer Buys a Product
Login
Authenticate Consumer Credentials
Authentication Success/Failure
Login Confirmation
Browse Available Products
Retrieve Product List
Product List Fetched
Display Available Products
Add Product to Cart (product ID, quantity)
Update Cart with Product
Cart Updated Successfully
Cart Updated Confirmation
alt
[Checkout with Products]
Proceed to Checkout
Fetch Cart Items and Details
Cart Details Retrieved
Adjust Product Quantity in Inventory
Inventory Updated
Order Confirmation
[Empty Cart]
Cannot Proceed (Cart is Empty)
Seller Adds a Product
Login
Authenticate Seller Credentials
Authentication Success/Failure
Login Confirmation
Add Product (name, quantity, price)
Save Product Information
Product Record Added
Product Added Successfully
ChatBot Interaction
Ask Question (e.g., "Delivery time?")
Search Relevant Information
Query Product/Order Details
Relevant Information Retrieved
Compile Response
Answer ("Delivery: 3 days")
Object Destruction
Close Database Connection
Terminate Chat Session
Farmer
Seller
Consumer
AgriVision System
Database
ChatBot
End of Interaction

Figure 6: Sequence Diagram

## 5.2 Sub-Diagrams

### 5.2.1 Farmer Sequence Diagram

This sub-diagram demonstrates the steps the system follows to detect vulnerabilities.

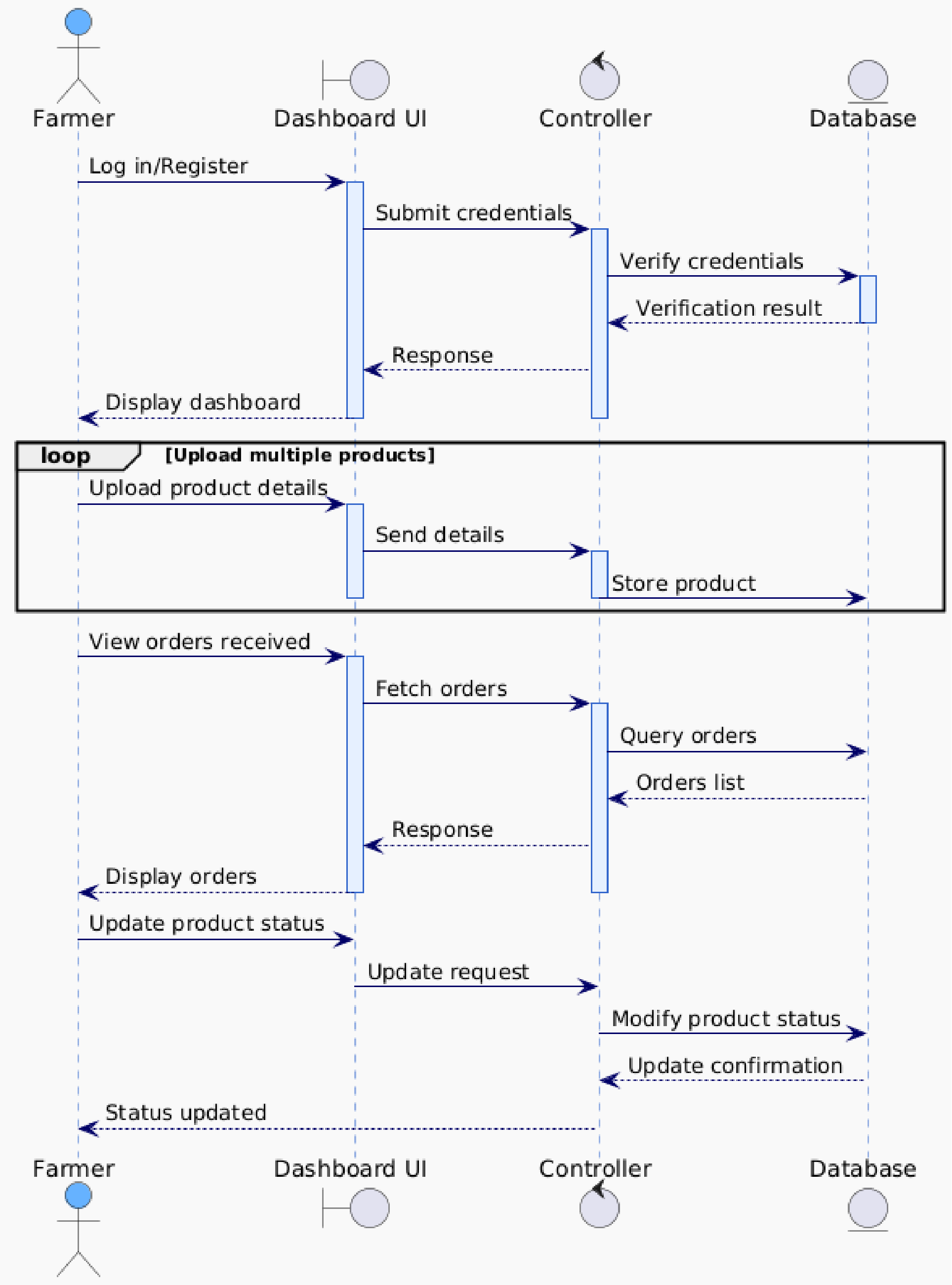


Figure 7: Sequence for Farmer

**Explanation**:

- The sequence begins with the system initiating a scan on websites for vulnerabilities related to agricultural products and services.
- Detected vulnerabilities are sent to the backend for detailed analysis and categorization.

- The backend evaluates the severity of each vulnerability, prioritizing issues that may affect farmer activities.
- Results are stored in the database for further reference and reporting.
- The system triggers notifications to relevant users, including farmers, about critical vulnerabilities in their profiles.

### 5.2.2 Consumer Sequence Diagram

This diagram details how the system handles notifications when a vulnerability is detected.

**Explanation**:

- The system identifies consumers affected by a detected vulnerability.
- The backend prepares the notification message with relevant details about the vulnerability.
- The system sends the notification to consumers via their registered communication channels (e.g., email, SMS).
- Consumers receive the notification and can view more details about the vulnerability by accessing the application.

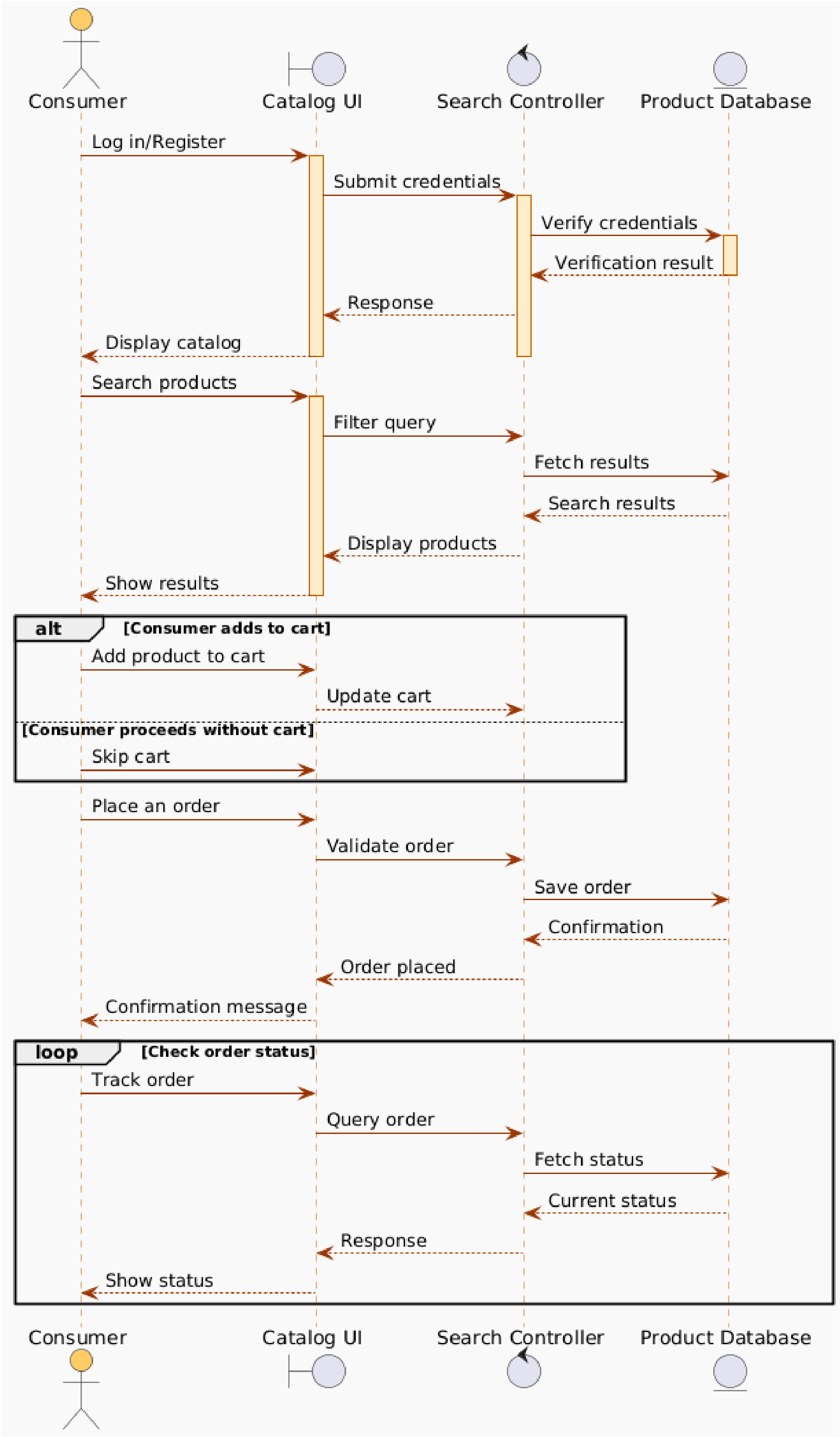


Figure 8: Sequence for Consumers

### 5.2.3 Supplier Sequence Diagram

This diagram demonstrates the steps involved when a vulnerability is detected for a supplier's profile or product.

**Explanation**:

- The system identifies suppliers whose products or profiles are affected by a detected vulnerability.
- The backend prepares the notification message, which includes the vulnerability details, and identifies if there is a need to update product listings or secure profiles.
- Notifications are sent to the suppliers via registered communication channels (e.g., email, SMS).
- Suppliers receive the notification and can take necessary actions, such as updating their product listings or profiles, to mitigate vulnerabilities.

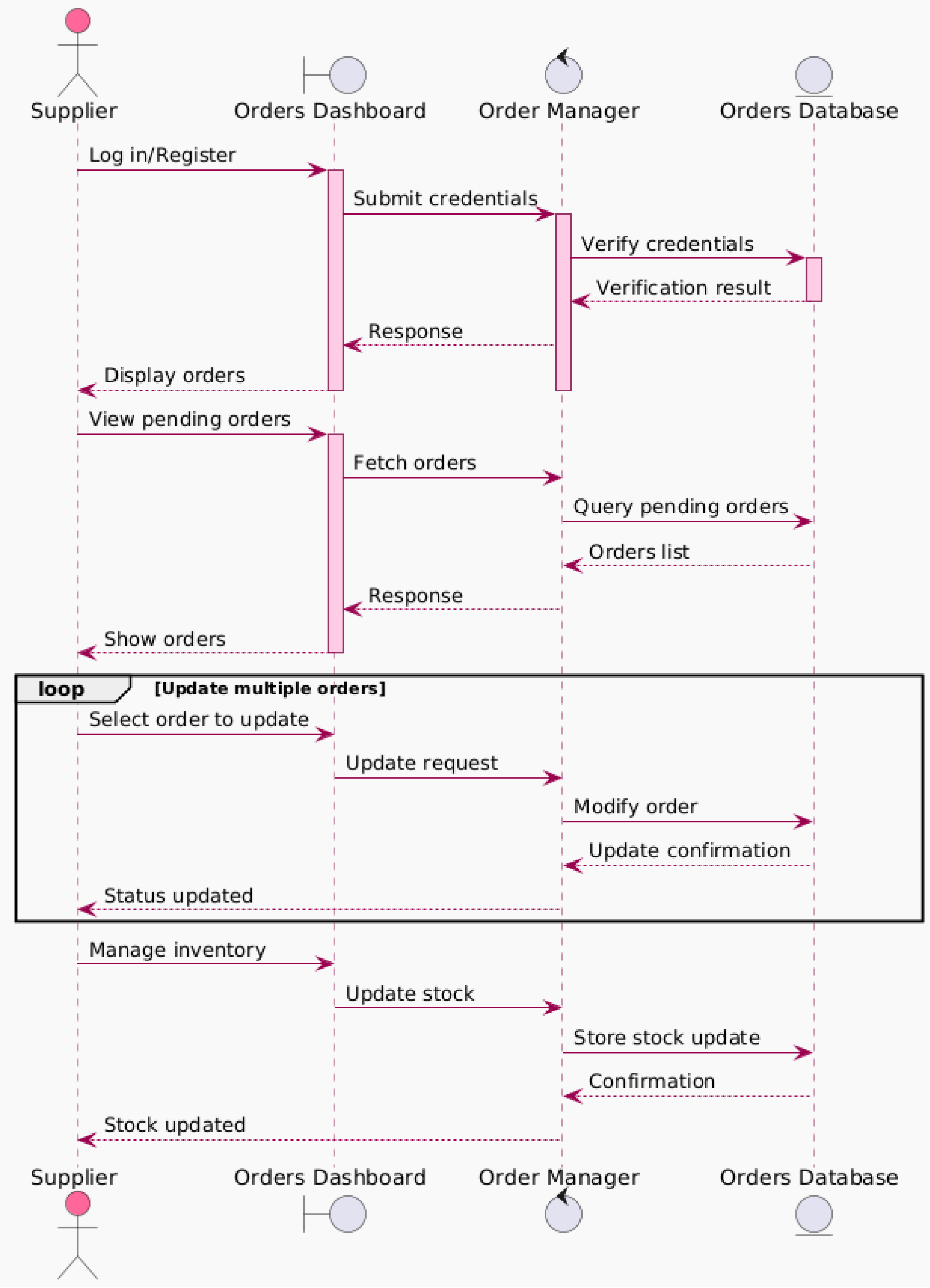


Figure 9: Sequence for Suppliers

# 6 Data Flow Design

## 6.1 Data Flow Diagram

The **Data Flow Diagram (DFD)** provides a high-level overview of how data flows within the system. It shows the input, processes, storage, and output of the data.

### 6.1.1 Level 0

The Level 0 DFD illustrates the system's overall data flow, focusing on the interactions between external entities and the main processes in the Agri Vision project.

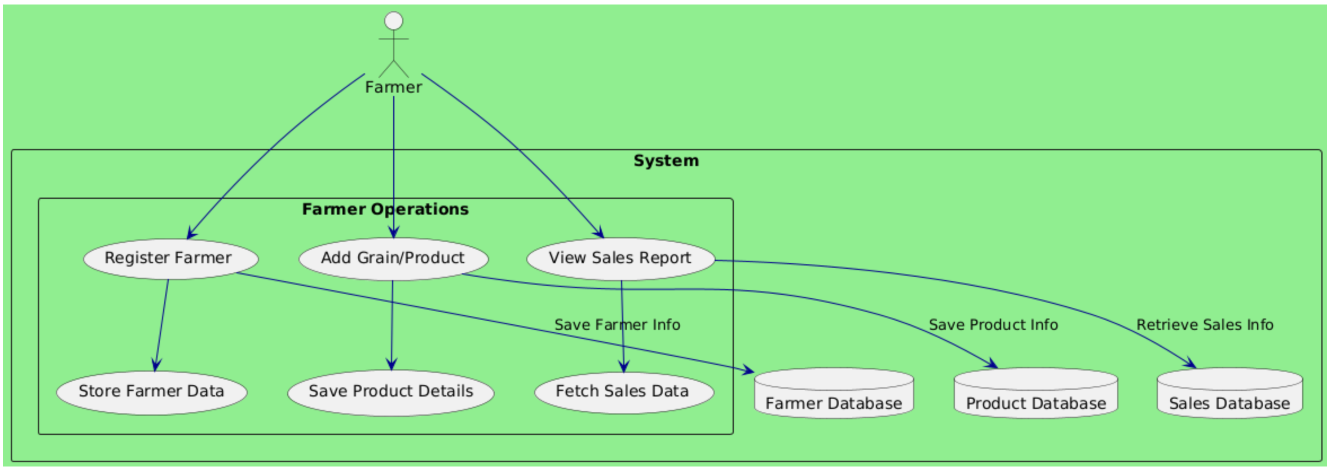


Figure 10: Data flow at farmer

**Explanation**:

- **Entities**:
  - **User**: Interacts with the system, performing actions like registering, managing profiles, and accessing weather updates.
- **Processes**:
  - **Agri Vision System (AVS)**: Handles data processing, including user registrations, product management, weather reporting, and profile management.
- **Data Flow**:
  - Users submit requests such as product updates, profile changes, or location-based weather queries.
  - The system processes the data, stores it in the database, and triggers relevant notifications to the user (e.g., updates on products or weather conditions).

### 6.1.2 Level 1

This sub-diagram details the primary processes within the Agri Vision system, such as user registration, weather data collection, vulnerability detection, and profile management.

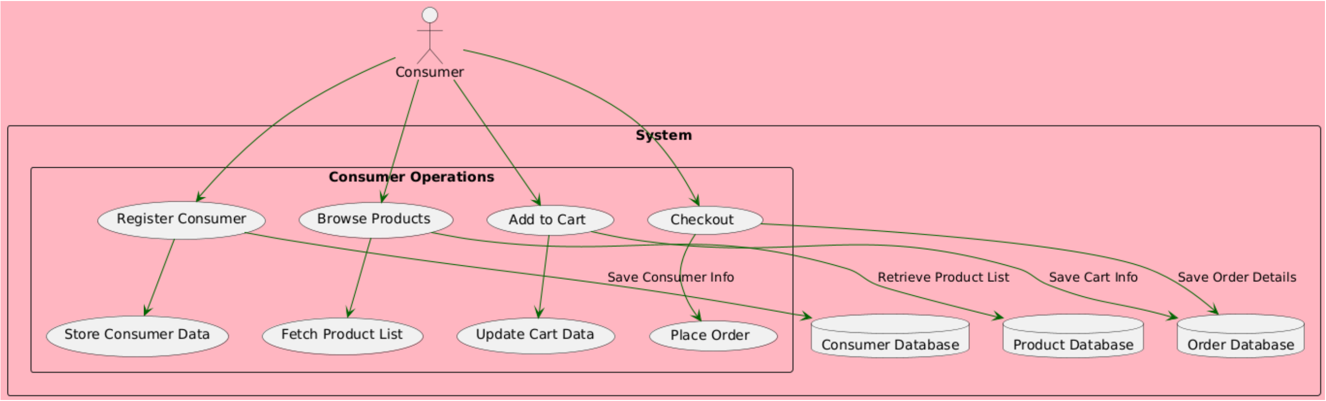


Figure 11: Data flow at Consumer

**Explanation**:

- **Processes**:
  - **User Registration**: Captures the user's details and stores them in the system.
  - **Weather Data Collection**: Gathers weather information based on user location or map interactions.
  - **Vulnerability Detection**: Scans agricultural products for vulnerabilities, like pests or diseases, and updates the database.
  - **Profile Management**: Allows users (farmers, sellers, consumers) to add, edit, or delete profiles and products.
- **Data Flow**:
  - Users submit their registration, product details, or weather queries to the system.
  - The system processes this data, stores it in the database, and triggers notifications (e.g., weather alerts or product updates).

## 6.2 Sub-Diagrams

**Explanation**:

- **Scan Agricultural Products**: Initiates a scan to check for vulnerabilities (e.g., pests, diseases) affecting crops or seeds.
- **Analyze Results**: Processes scanned data to identify risks like pests, soil quality, or environmental issues.

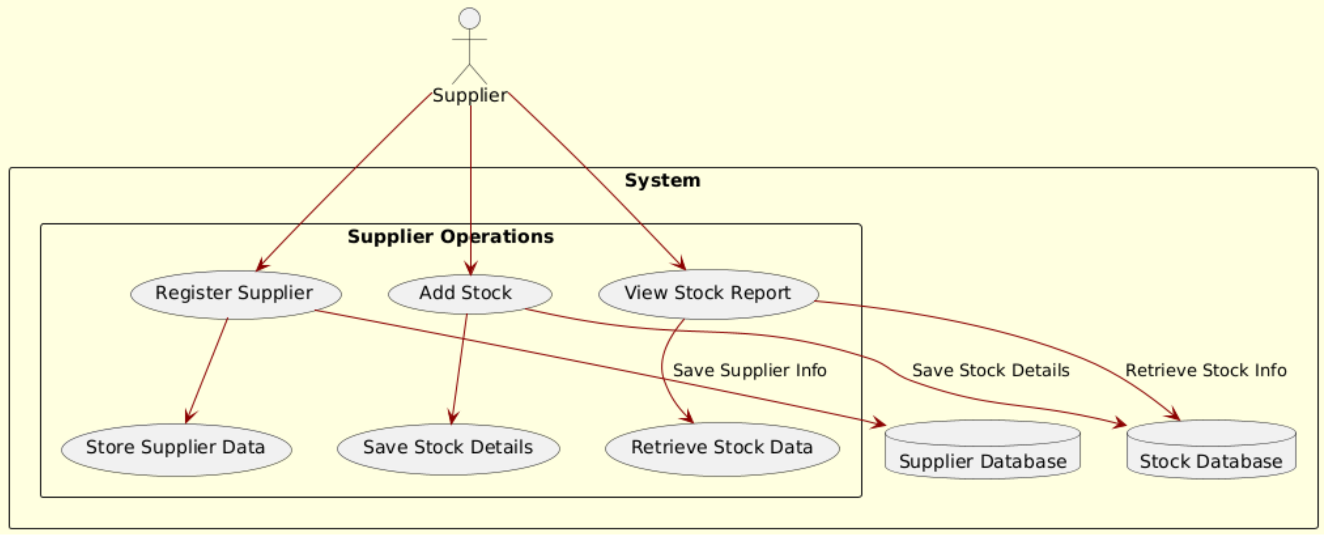


Figure 12: Data flow at supplier.

- **Store Vulnerability Data**: Saves vulnerability findings to the database for further reporting.
- **Data Flow**:
  - The system scans and processes data related to agricultural products.
  - The findings are stored and retrieved as needed, generating alerts or updates.

### 6.2.1 Weather Data Collection Sub-DFD

This sub-diagram focuses on how weather data is collected, processed, and displayed to the user based on location.

**Explanation**:

- **Request Weather Information**: The user submits a location to retrieve the weather data.
- **Fetch Weather Data**: The system connects to weather APIs to collect data.
- **Display Weather Data**: The system processes and presents weather data to the user in the interface.
- **Data Flow**:
  - The user's location is used to fetch relevant weather data.
  - Weather data is processed and displayed on the map or user interface.

# 7 Activity Design

## 7.1 Activity Diagram

The **Activity Diagram** represents the dynamic aspects of the system, illustrating the flow of activities within various processes in the Agri Vision project.

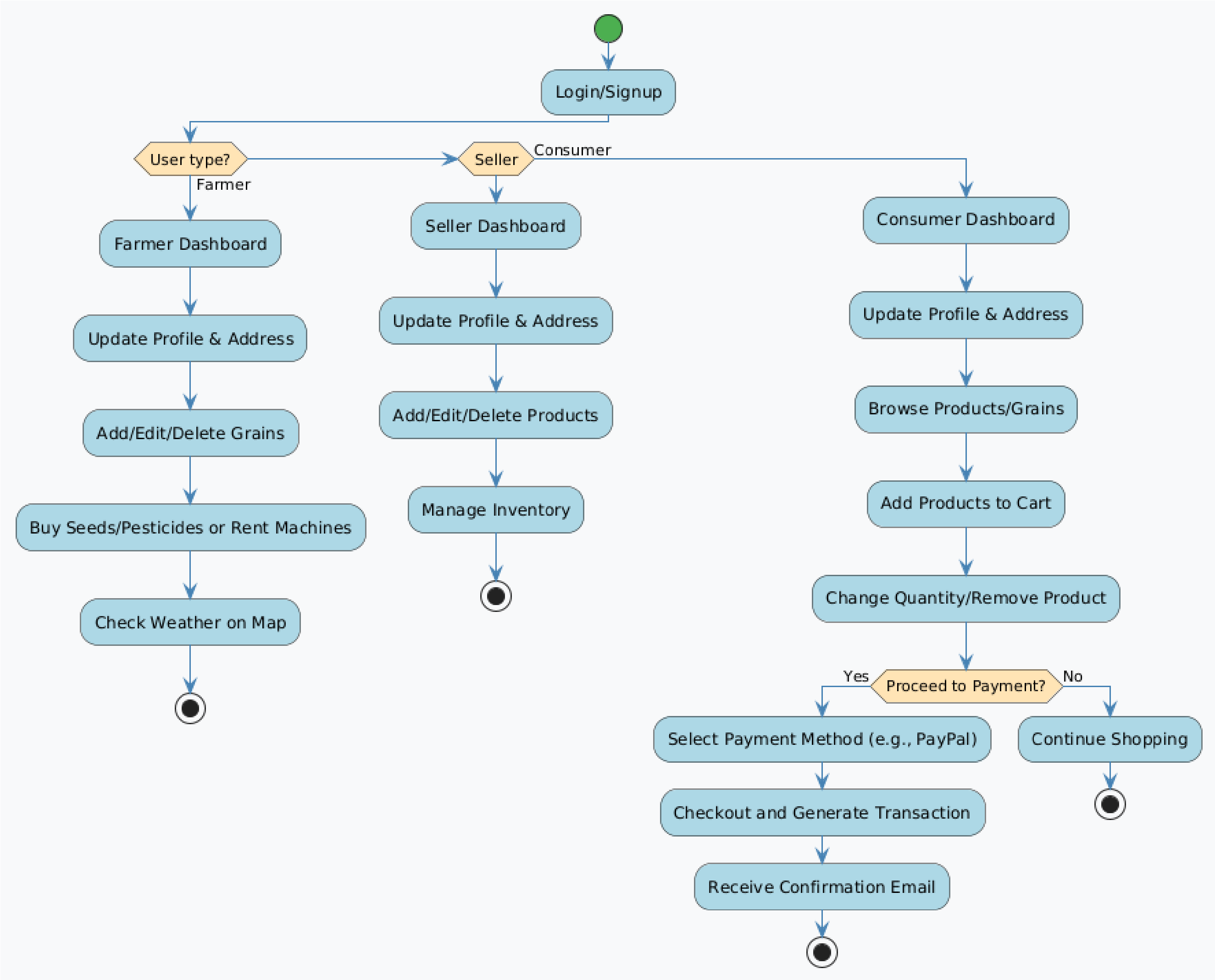


Figure 13: Main Activity Diagram: User Interaction Flow

**Explanation:**

- **User Interaction:** The user initiates the workflow by submitting requests for weather data, product updates, or profile management.
- **Weather Data Retrieval:** Based on the user's location or interaction with the map, the system retrieves weather data from external APIs.
- **Profile Management:** Users interact with their profile (e.g., update address, manage products) and make changes.
- **Vulnerability Detection (if applicable):** The system may also initiate vulnerability scans for products (e.g., crops or seeds) and store the results in the database.
- **Notification Trigger:** Based on the weather data, product updates, or vulnerabilities, notifications are prepared and sent to users.
- **Database Update:** All processed information, such as weather data, product details, or vulnerabilities, is stored in the database for future reference.

## 7.2 Sub-Diagrams

### 7.2.1 Activity Diagram: Admin Process

This diagram elaborates on the internal steps for managing user accounts, updating product information, and overseeing the system's operations.

**Explanation:**

- The admin system begins by receiving requests for user registration or product updates.
- It checks for the validity of the requests, such as ensuring that users are registered or verifying that product details are accurate.
- The system may trigger vulnerability scans for certain products or monitor the system for performance and security issues.
- The admin can update the database with new product listings, user details, or critical system notifications.

### 7.2.2 Activity Diagram: User Registration Process

This diagram illustrates the activity flow for user registration, which is essential for farmers, sellers, and consumers.

**Explanation:**

- The user provides registration details, including personal information and preferences.
- The system validates the input data for correctness and ensures that the data is not already in the system.
- Upon successful validation, the user's details are stored in the database.

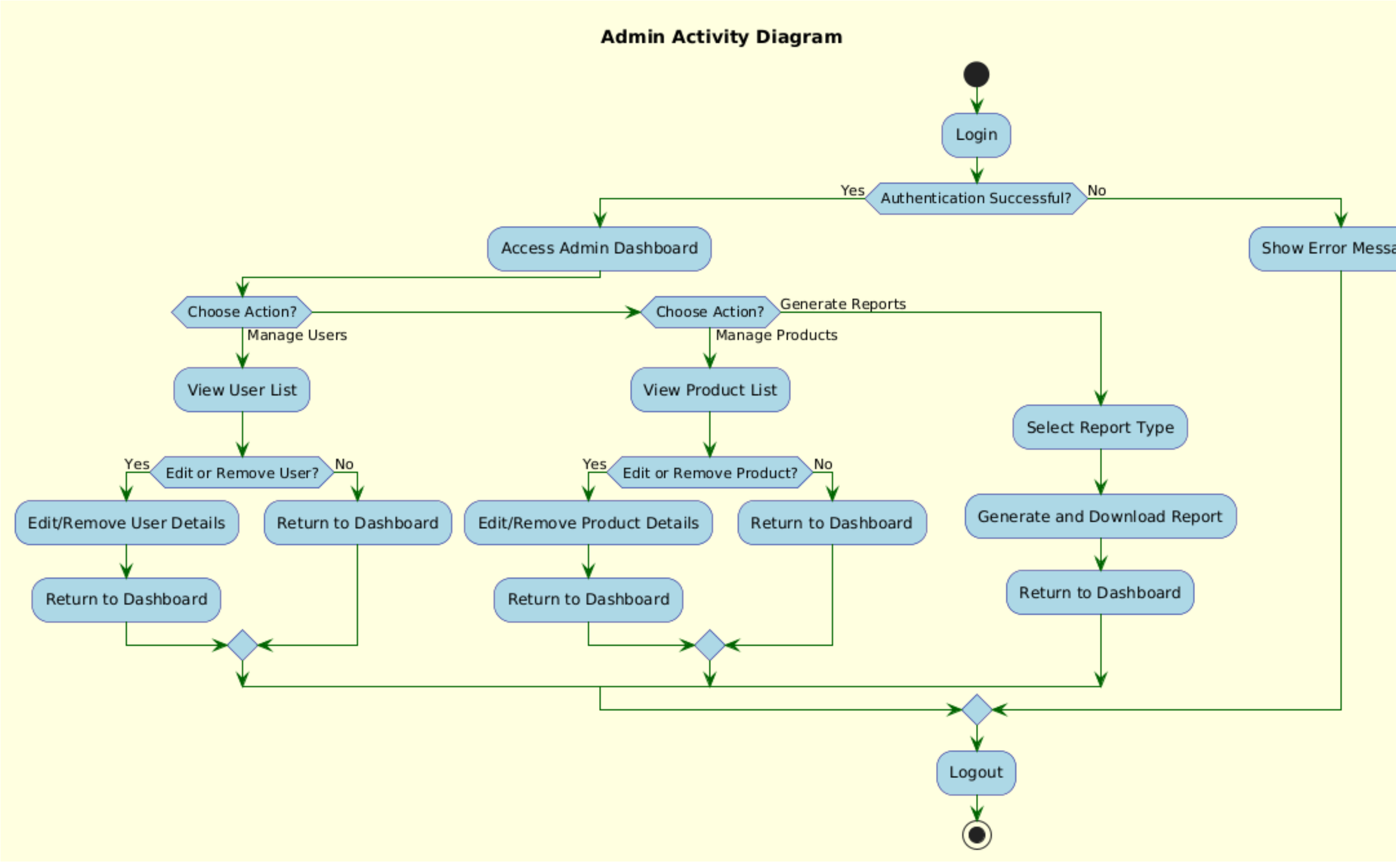


Figure 14: Admin Process Activity Diagram

# 8 Frontend Images

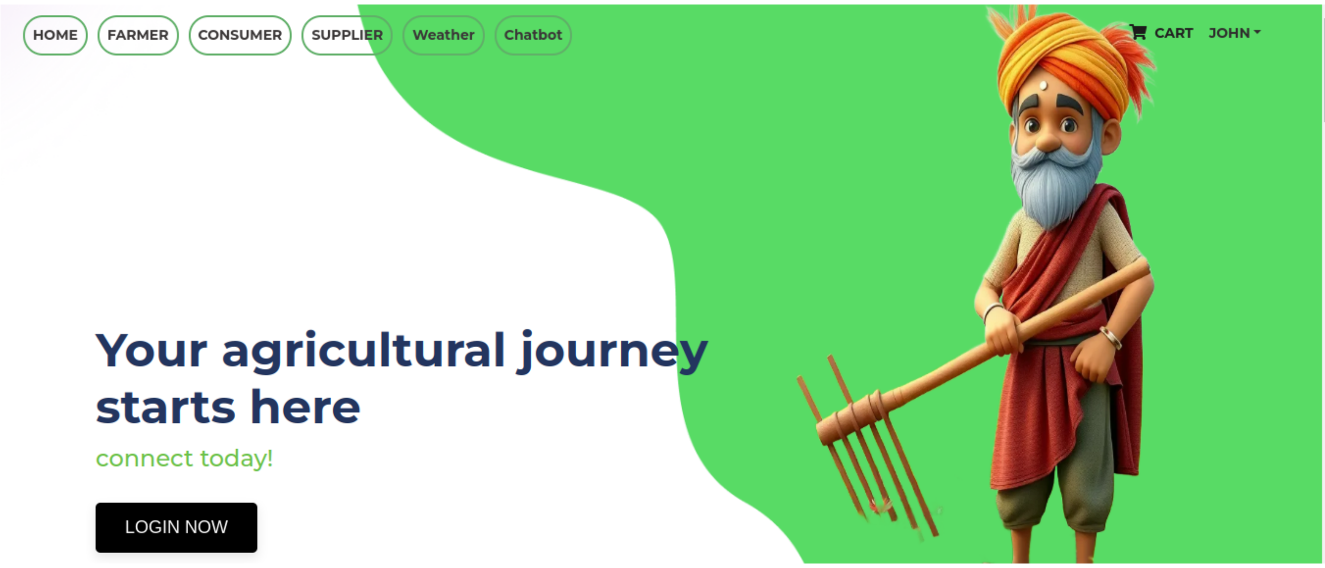
- The subject line is concise and highlights the purpose of the email, such as **Order Confirmation**, **Profile Update**, or **New Product Listing Notification**.
- Key details (e.g., order ID, update summary, or relevant dates) ensure users have all necessary information at a glance.
- The email includes direct links to the Agri Vision dashboard, allowing users to view their orders, update their profiles, or access additional resources.

## 8.1 Frontend Images

Below are sample screenshots of the Agri Vision user interface:

- **Farmer Dashboard:** A streamlined view for farmers to list grains, edit personal/company details, and access farming resources like weather updates or machinery rentals.
- **Seller Interface:** A dedicated space for sellers to manage product inventories, update addresses, and interact with consumers.
- **Consumer Interface:** A user-friendly section for consumers to browse agricultural products, manage carts, and finalize purchases with farmers or sellers.

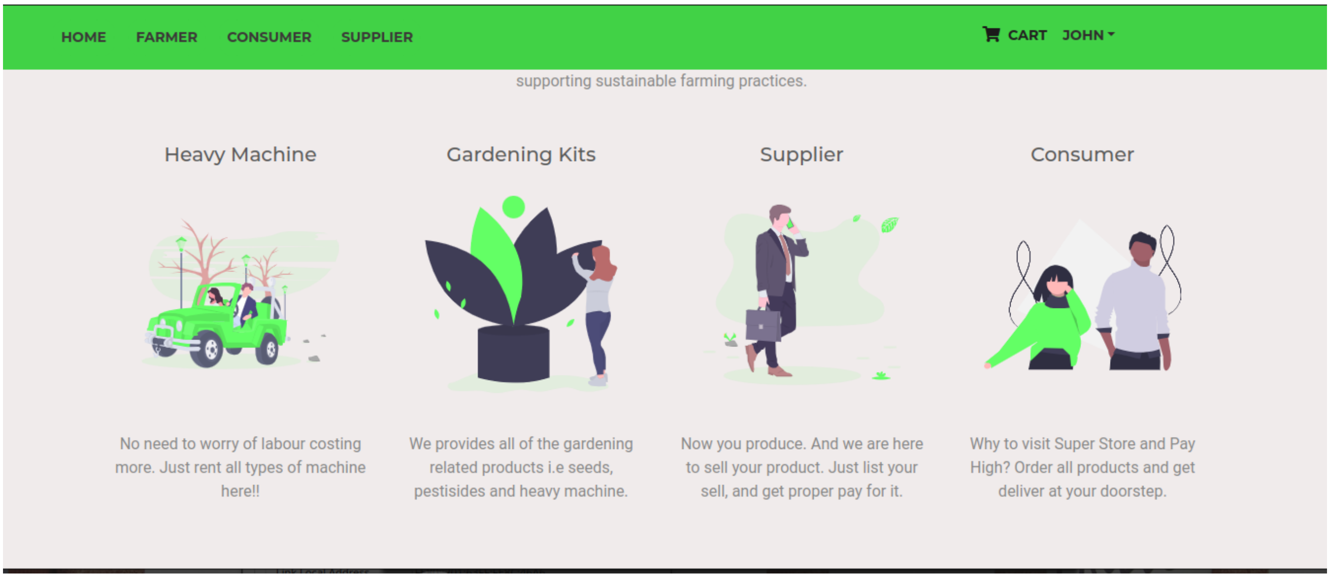
These images illustrate the intuitive and role-specific designs that make the platform efficient and user-friendly for all stakeholders.



Your agricultural journey starts here

connect today!

Figure 15: Agri vision Landing Page



supporting sustainable farming practices.

Heavy Machine

No need to worry of labour costing more. Just rent all types of machine here!!

Gardening Kits

We provides all of the gardening related products i.e seeds, pestisides and heavy machine.

Supplier

Now you produce. And we are here to sell your product. Just list your sell, and get proper pay for it.

Consumer

Why to visit Super Store and Pay High? Order all products and get deliver at your doorstep.

Figure 16: Landing page of AgriVision

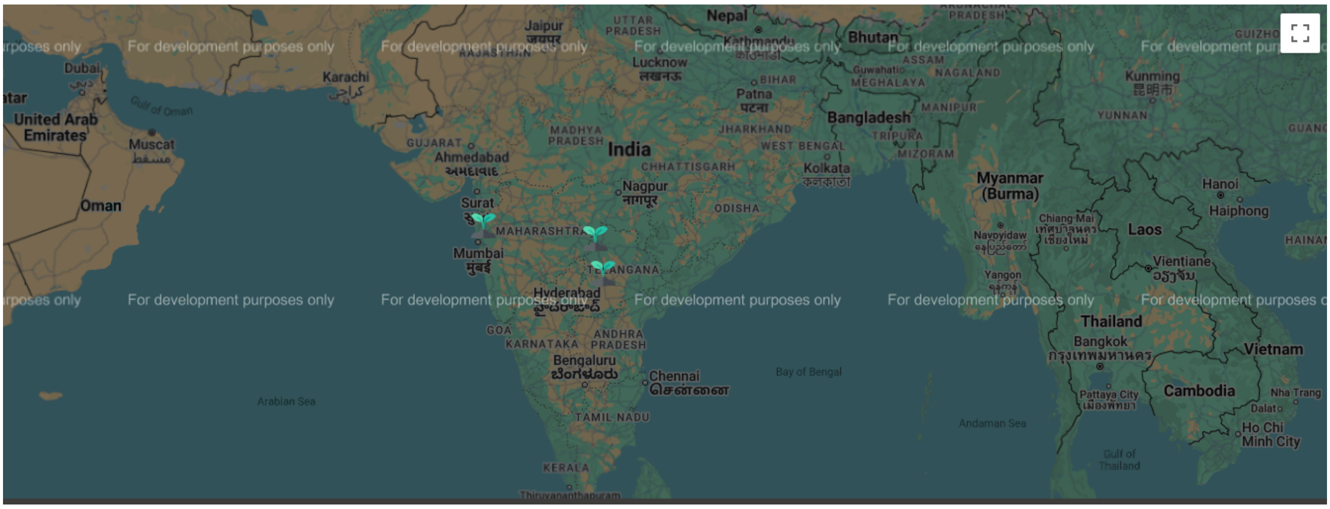
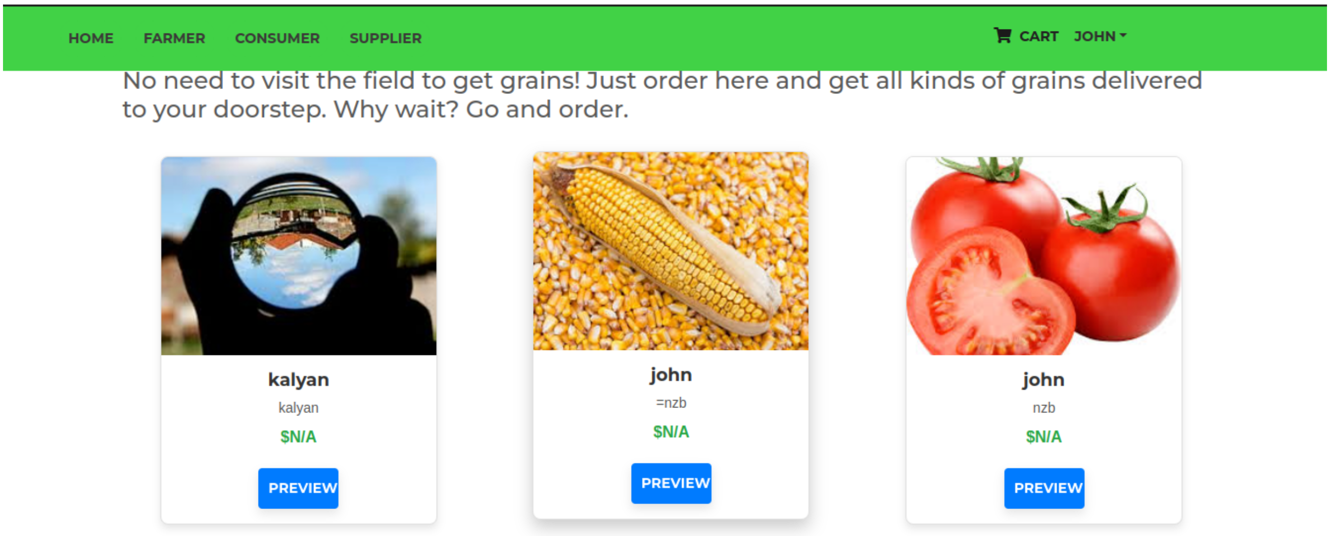


Figure 17: weather map



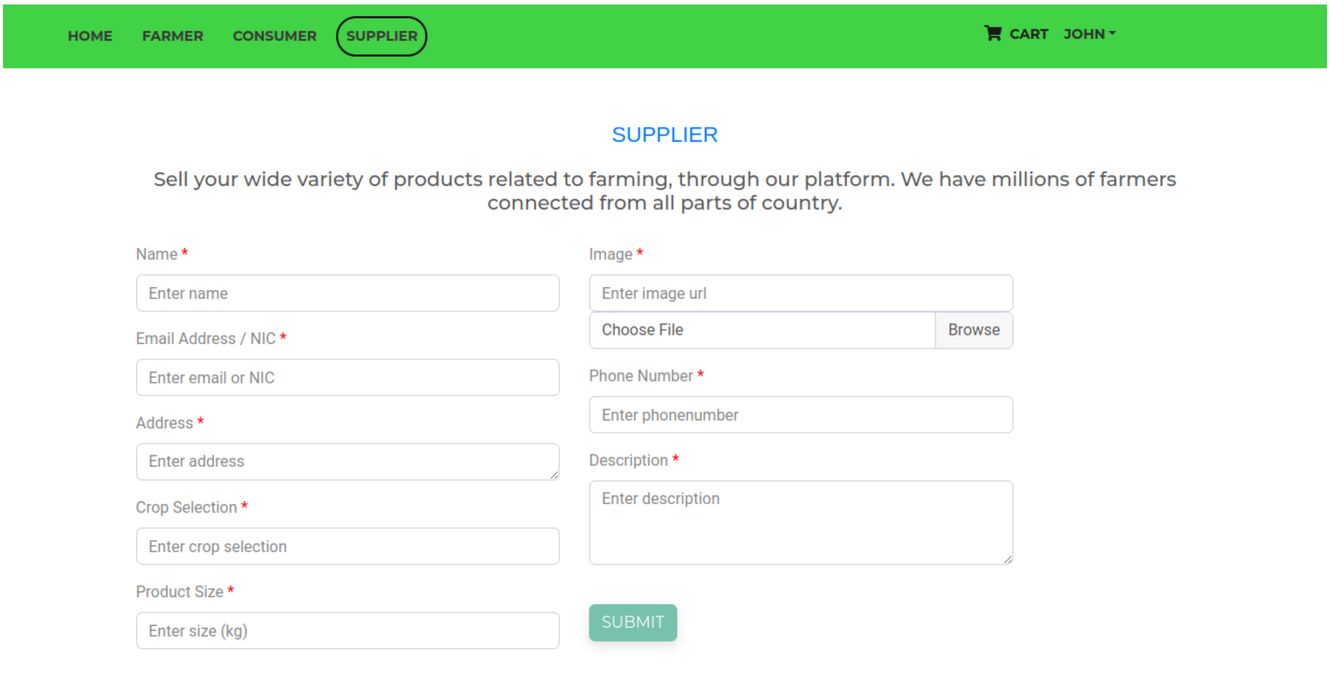
No need to visit the field to get grains! Just order here and get all kinds of grains delivered to your doorstep. Why wait? Go and order.

kalyan
kalyan
$N/A
PREVIEW

john
=nzb
$N/A
PREVIEW

john
nzb
$N/A
PREVIEW

Figure 18: Consumer section



SUPPLIER

Sell your wide variety of products related to farming, through our platform. We have millions of farmers connected from all parts of country.

Name *
Enter name

Email Address / NIC *
Enter email or NIC

Address *
Enter address

Crop Selection *
Enter crop selection

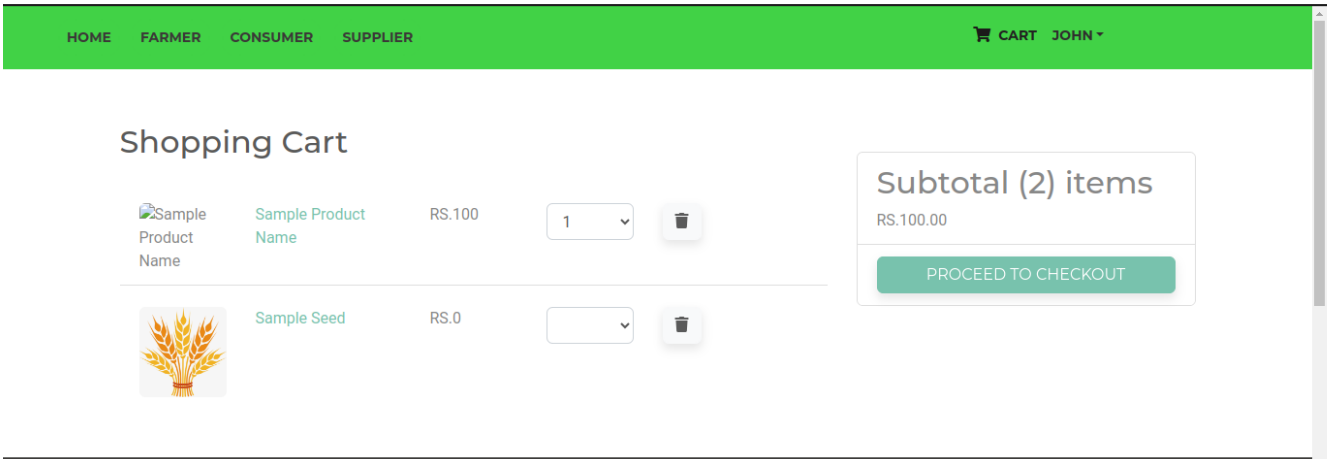
Product Size *
Enter size (kg)

Image *
Enter image url

Phone Number *
Enter phonenumber

Description *
Enter description

Figure 19: Supplier section



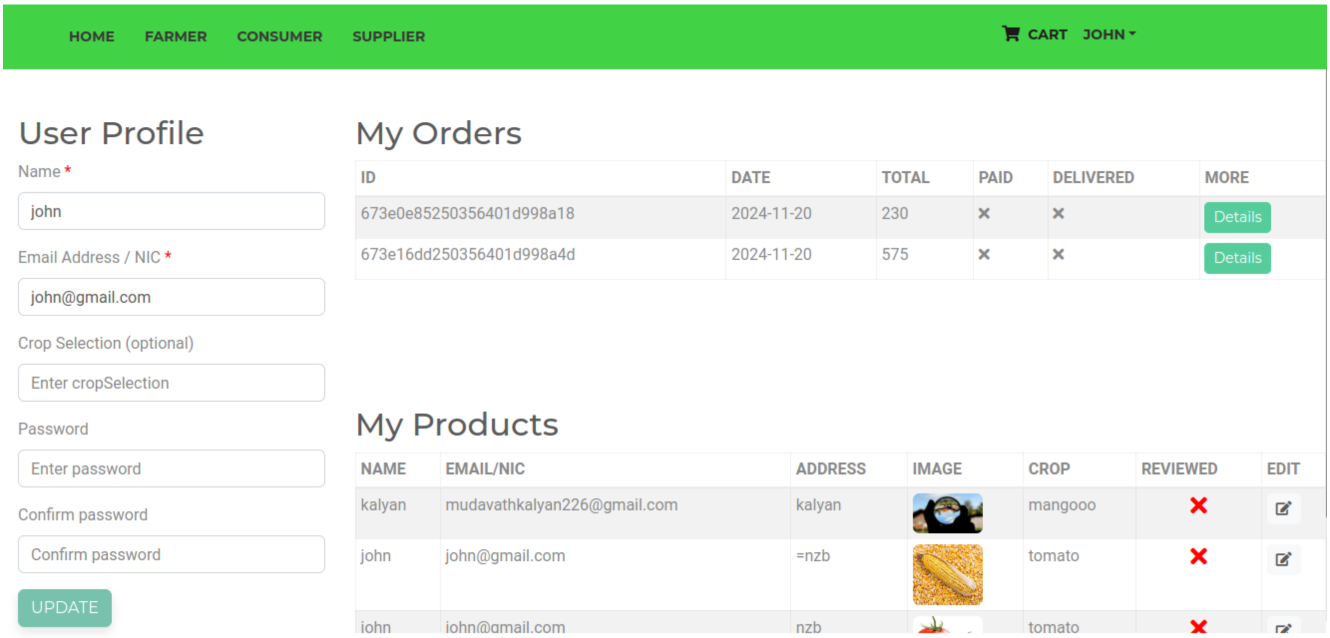
Shopping Cart

| | | | |
|---|---|---|---|
| Sample Product Name | Sample Product Name | RS.100 | 1 |
| | Sample Seed | RS.0 | |

Subtotal (2) items

RS.100.00

Figure 20: Cart section



User Profile

Name *: john

Email Address / NIC *: john@gmail.com

Crop Selection (optional): Enter cropSelection

Password: Enter password

Confirm password: Confirm password

My Orders

| ID | DATE | TOTAL | PAID | DELIVERED | MORE |
|---|---|---|---|---|---|
| 673e0e85250356401d998a18 | 2024-11-20 | 230 | ✗ | ✗ | Details |
| 673e16dd250356401d998a4d | 2024-11-20 | 575 | ✗ | ✗ | Details |

My Products

| NAME | EMAIL/NIC | ADDRESS | IMAGE | CROP | REVIEWED | EDIT |
|---|---|---|---|---|---|---|
| kalyan | mudavathkalyan226@gmail.com | kalyan | | mangooo | ✗ | ✎ |
| john | john@gmail.com | =nzb | | tomato | ✗ | ✎ |
| john | john@gmail.com | nzb | | tomato | ✗ | ✎ |

Figure 21: Profile section

## 8.2 Conclusion

The Agri Vision Platform revolutionizes the agricultural ecosystem by creating a unified space for farmers, sellers, and consumers to connect seamlessly. By eliminating intermediaries, it ensures fair pricing for farmers while offering consumers access to fresh, organic products directly from the source. Farmers can also access essential tools like weather updates and resources such as seeds, pesticides, and machinery rentals, empowering them for better productivity.

Sellers gain an efficient platform to manage inventory and expand their reach, while consumers enjoy simplified interactions with transparent access to agricultural goods. Agri Vision addresses key challenges like market fragmentation and inefficiencies, fostering a fair, sustainable, and digitally driven agricultural ecosystem that empowers communities and promotes equitable growth.

# AgriVision Project: Test Cases Document

## General Information

**Overview**

**Project Name:** AgriVision -Web Application<br>
**Document Type:** Test Case Document<br>
**Author:**<br>
D Naresh B200391<br>
M Kalyan B201174<br>
M Aravind B200778<br>
**Date Prepared:** [06/12/2024]<br>
**Version:** 1.0<br>
**Status:** Final Draft

## Revision History

| Version | Description | Author | Date |
|---|---|---|---|
| 1.0 | Initial Draft | [D Naresh ,M Aravind ,M Kalyan] | [06/12/2024] |

## Test Case Summary

### Feature Overview

- ■ Login Validation and Authentication
- ■ Farmer Features
- ■ Consumer Features
- ■ Seller Features
- ■ Weather Integration
- ■ Chatbot
- ■ Contact us
- ■ Other Features

# Login Validation and Authentication

## TC_001: User Login with Valid Credentials

Description

This test case verifies login functionality with valid credentials.

**Steps:**

➤ Navigate to the login page.

➤ Enter valid username and password.

➤ Click ”Login.”

**Expected Result:**

The user is successfully logged in and redirected to the dashboard.

## TC_002: User Login with Invalid Credentials

Description

This test case checks if the system handles invalid login attempts appropriately.

**Steps:**

➤ Navigate to the login page.

➤ Enter invalid username or password.

➤ Click ”Login.”

**Expected Result:**

An error message is displayed, and login is not permitted.

# Farmer Features

## TC_003: Adding Grains to Farmer Profile

Description

This test case ensures farmers can add grains to their profiles for listing purposes.

**Steps:**

➤ Log in as a farmer.

➤ Navigate to the ”Profile” section.

➤ Click ”Add Grains” and fill in the required details.

➤ Save the changes.

**Expected Result:**

The grain is added successfully, and the farmer’s profile reflects the updated list.

## TC_004: Renting Farming Machinery

Description

This test case verifies the renting functionality for farming machinery.

**Steps:**

➤ Log in as a farmer.

➤ Navigate to the ”Machinery” section.

➤ Select machinery to rent and confirm the transaction.

**Expected Result:**

The farmer successfully rents machinery, and the transaction is recorded.

## TC_005: Adding Products to Cart

Description

This test case ensures consumers can add products to their cart.

**Steps:**

- ➤ Log in as a consumer.
- ➤ Navigate to the product page.
- ➤ Click ”Add to Cart.”

**Expected Result:**

The product is successfully added to the cart.

## TC_006:Removing Items from Cart

Description

This test case ensures consumers can manage items in their cart effectively.

**Steps:**

- ➤ Log in as a consumer.
- ➤ Navigate to a product listing page.
- ➤ click on open cart to ”Enter the Cart”
- ➤ Navigate to the cart page and remove an item.

**Expected Result:**

Items are added and removed from the cart successfully, and the cart total updates accordingly.

## TC_007: Viewing Weather Information via Map API

Description

This test case ensures that weather information is displayed based on user location.

**Steps:**

- ➤ Log in as a farmer or consumer.
- ➤ Navigate to the ”Weather” section.
- ➤ Enter a location or select it on the map.
- ➤ Click ”Get Weather” to display weather details.

**Expected Result:**

The weather information for the specified location is displayed accurately.

## TC_008: Adding Products to Supplier Profile

Description

This test case ensures suppliers can add products to their inventory.

**Steps:**

- ➤ Log in as a supplier.
- ➤ Navigate to the ”Manage Products” section.
- ➤ Click ”Add Product” and fill in product details (e.g., name, quantity, price).
- ➤ Save the changes.

**Expected Result:**

The product is added to the supplier’s profile and listed in the inventory.

# Contact Us Feature

## TC_009: Submitting a Query via Contact Us Form

Description

This test case ensures users can submit queries through the ”Contact Us” form.

**Steps and Details:**

| Step | Details and Explanation |
|---|---|
| 1 | Navigate to the ”Contact Us” page. The link is typically located in the footer or header of the website. |
| 2 | Fill in the required fields such as name, email, and message. Provide a detailed description of the issue or query. |
| 3 | Click ”Submit.” Ensure all required fields are completed before submitting. |
| 4 | Verify that a confirmation message is displayed, indicating successful submission. |

**Expected Result:**

The query is submitted successfully, and the user receives a confirmation message.

## TC_010: Viewing Farmer Locations on Map

Description

This test case ensures users can view farmer locations on the map.

**Steps:**

- ➤ Navigate to the ”Farmer Locations” map page.
- ➤ Zoom and pan the map to locate nearby farmers.
- ➤ Click on a farmer’s marker to view profile and products.

**Expected Result:**

The map displays farmer locations accurately, and clicking on a marker shows relevant details.

# Admin Interface

## TC_011: Viewing Dashboard Statistics

Description

This test case ensures the admin can view statistical data on the dashboard, including user counts, farmer counts, and product statistics.

**Steps and Details:**

| Step | Details and Explanation |
|---|---|
| 1 | Log in as an admin. Use valid admin credentials to access the dashboard. |
| 2 | Navigate to the "Dashboard" section. This section provides a visual summary of key metrics such as the total number of users, farmers, products, and orders. |
| 3 | Verify that all statistics are displayed correctly. Ensure the values are accurate and up-to-date based on the database. |
| 4 | Check the graphical data representation. Validate that the charts and graphs for trends are interactive and accurate. |

**Expected Result:**

The dashboard displays all statistics accurately, with up-to-date data and properly rendered graphs.

## TC_012: Viewing Farmer Profiles

Description

This test case ensures the admin can view and manage farmer profiles.

**Steps and Details:**

| Step | Details and Explanation |
|---|---|
| 1 | Log in as an admin. Use valid admin credentials to access the interface. |
| 2 | Navigate to the "Farmer Profiles" section. This section lists all registered farmers. |
| 3 | Select a farmer's profile from the list. Verify that all details such as name, contact information, and product listings are displayed. |
| 4 | Validate the edit functionality. Update specific information like address or phone number and save changes. Ensure the updates reflect correctly. |

**Expected Result:**

The admin can view detailed farmer profiles, update their information, and ensure changes are saved successfully.

## TC_013: Managing User List

Description

This test case ensures the admin can view and manage the list of registered users.

**Steps and Details:**

| Step | Details and Explanation |
|---|---|
| 1 | Log in as an admin. Use valid admin credentials. |
| 2 | Navigate to the ”User List” section. This section lists all registered users, including their roles (e.g., consumer, farmer, seller). |
| 3 | Select a user to view their details. Verify information such as name, email, role, and account status. |
| 4 | Test the account management functionality. Activate, deactivate, or delete a user account and save changes. Ensure changes are reflected in the user list. |

**Expected Result:**

The admin can view user details and manage user accounts effectively.

## TC_014: Managing Product List

Description

This test case ensures the admin can view, edit, or delete products listed by farmers or sellers.

**Steps and Details:**

| Step | Details and Explanation |
|---|---|
| 1 | Log in as an admin. Use valid admin credentials. |
| 2 | Navigate to the "Product List" section. This section displays all products listed by farmers and sellers. |
| 3 | Select a product to view its details. Ensure details like name, category, price, and stock quantity are displayed. |
| 4 | Test the edit functionality. Modify the product details (e.g., price or quantity) and save changes. Ensure the updates reflect accurately. |
| 5 | Test the delete functionality. Remove a product and verify it no longer appears in the product list. |

**Expected Result:**

The admin can view, edit, or delete products successfully, and all changes reflect accurately in the database.

## TC_015: Viewing and Managing Orders List

Description

This test case ensures the admin can view and manage the list of orders placed by consumers.

**Steps and Details:**

| Step | Details and Explanation |
|---|---|
| 1 | Log in as an admin. Use valid admin credentials. |
| 2 | Navigate to the "Orders List" section. This section lists all orders placed on the platform. |
| 3 | Select an order to view its details. Ensure order details such as consumer name, product name, quantity, and order status are displayed. |
| 4 | Test the order status update functionality. Change the status (e.g., from "Processing" to "Delivered") and save changes. |

**Expected Result:**

The admin can view and update order details successfully, and changes are reflected accurately.

# Conclusion

**Overall Assessment**

The testing process successfully identified and documented critical defects, ensuring the system's functionality aligns with the specified requirements. The application achieved **95% functional compliance**, with the majority of test cases passing. The identified defects were primarily low or medium severity, with no critical blocking issues remaining.

This indicates a robust and reliable foundation for the application, ready for further optimization and refinement in subsequent iterations. Performance and scalability tests highlighted areas where minor optimizations are required to enhance responsiveness under heavy load conditions. Additionally, user interface testing confirmed that the design is responsive and accessible across devices, with minor adjustments needed for cross-browser compatibility.

## Strengths

- ✔ High pass rate across core functional test cases, demonstrating stable and reliable system behavior.
- ✔ Comprehensive coverage of user authentication, file management, and user interface modules.
- ✔ Successful performance under moderate loads, with no significant system crashes or latency issues.
- ✔ Adherence to accessibility guidelines, ensuring inclusivity for all users.

## Identified Weaknesses

- ✗ Some file management operations exhibited minor delays under extreme load conditions.
- ✗ A few defects related to multi-language support need immediate attention.
- ✗ Minor discrepancies in visual elements across different browsers.

## Recommendations

- Resolve all open defects before the next release to maintain high-quality standards.
- Conduct thorough regression testing to confirm that recent fixes do not introduce new issues.
- Perform load testing under higher user volumes to validate scalability for production environments.
- Enhance multi-language support to ensure a seamless user experience for all supported languages.

- Schedule a User Acceptance Testing (UAT) session with stakeholders to validate the application's readiness for deployment.
- Document learnings from this testing cycle to improve future testing strategies and coverage.

## Next Steps

The next phase of the development cycle should focus on the following:

1. Implement fixes for identified issues and optimize performance in targeted areas.
2. Develop additional test cases for newly implemented features or modules.
3. Refine and extend performance testing scenarios for anticipated production workloads.
4. Conduct a final review with key stakeholders to align on release readiness.
5. Prepare detailed user documentation and training materials to facilitate smooth onboarding.

### Final Notes

Overall, the testing phase has provided a strong foundation for the application by addressing critical areas of functionality, performance, and accessibility. With planned improvements and stakeholder engagement, the system is on track to meet user expectations and business requirements for the final release.