

**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 | 4 |
| 1.1 | <i>Project Overview</i> | 4 |
| 1.2 | <i>Purpose and Scope</i> | 4 |
| 1.3 | <i>Audience</i> | 5 |
| 1.4 | <i>Key Features and Benefits</i> | 5 |
| 1.5 | <i>System Requirements</i> | 6 |
| 1.6 | <i>System Limitations</i> | 6 |
| 2 | System Architecture | 7 |
| 2.1 | <i>High-Level Overview</i> | 7 |
| 2.2 | <i>Technology Stack</i> | 7 |
| 3 | Database Design | 8 |
| 3.1 | <i>Entity-Relationship Diagram (ERD)</i> | 8 |
| 3.2 | <i>Sub-Diagrams</i> | 9 |
| 3.2.1 | Farmer Entity Sub-Diagram | 9 |
| 3.2.2 | Consumer Entity Sub-Diagram | 10 |
| 3.2.3 | Suppliers Entity Sub-Diagram | 10 |
| 4 | Class Design | 11 |
| 4.1 | <i>Overview</i> | 11 |
| 4.2 | <i>Class Diagram</i> | 11 |
| 4.3 | <i>User Class Design</i> | 12 |
| 4.4 | <i>Product Class Design</i> | 13 |
| 4.5 | <i>Weather Class Design</i> | 13 |
| 5 | Interaction Design | 14 |
| 5.1 | <i>Sequence Diagram</i> | 14 |
| 5.2 | <i>Sub-Diagrams</i> | 16 |
| 5.2.1 | Farmer Sequence Diagram | 16 |
| 5.2.2 | Consumer Sequence Diagram | 17 |
| 5.2.3 | Supplier Sequence Diagram | 19 |
| 6 | Data Flow Design | 21 |
| 6.1 | <i>Data Flow Diagram</i> | 21 |
| 6.1.1 | Level 0 | 21 |
| 6.1.2 | Level 1 | 22 |
| 6.2 | <i>Sub-Diagrams</i> | 22 |
| 6.2.1 | Weather Data Collection Sub-DFD | 23 |
| 7 | Activity Design | 23 |
| 7.1 | <i>Activity Diagram</i> | 23 |
| 7.2 | <i>Sub-Diagrams</i> | 25 |
| 7.2.1 | Activity Diagram: Admin Process | 25 |
| 7.2.2 | Activity Diagram: User Registration Process | 25 |
| 8 | Frontend Images | 26 |

| | | |
|-----|----------------------------------|----|
| 8.1 | <i>Frontend Images</i> | 26 |
| 8.2 | <i>Conclusion</i> | 30 |

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:

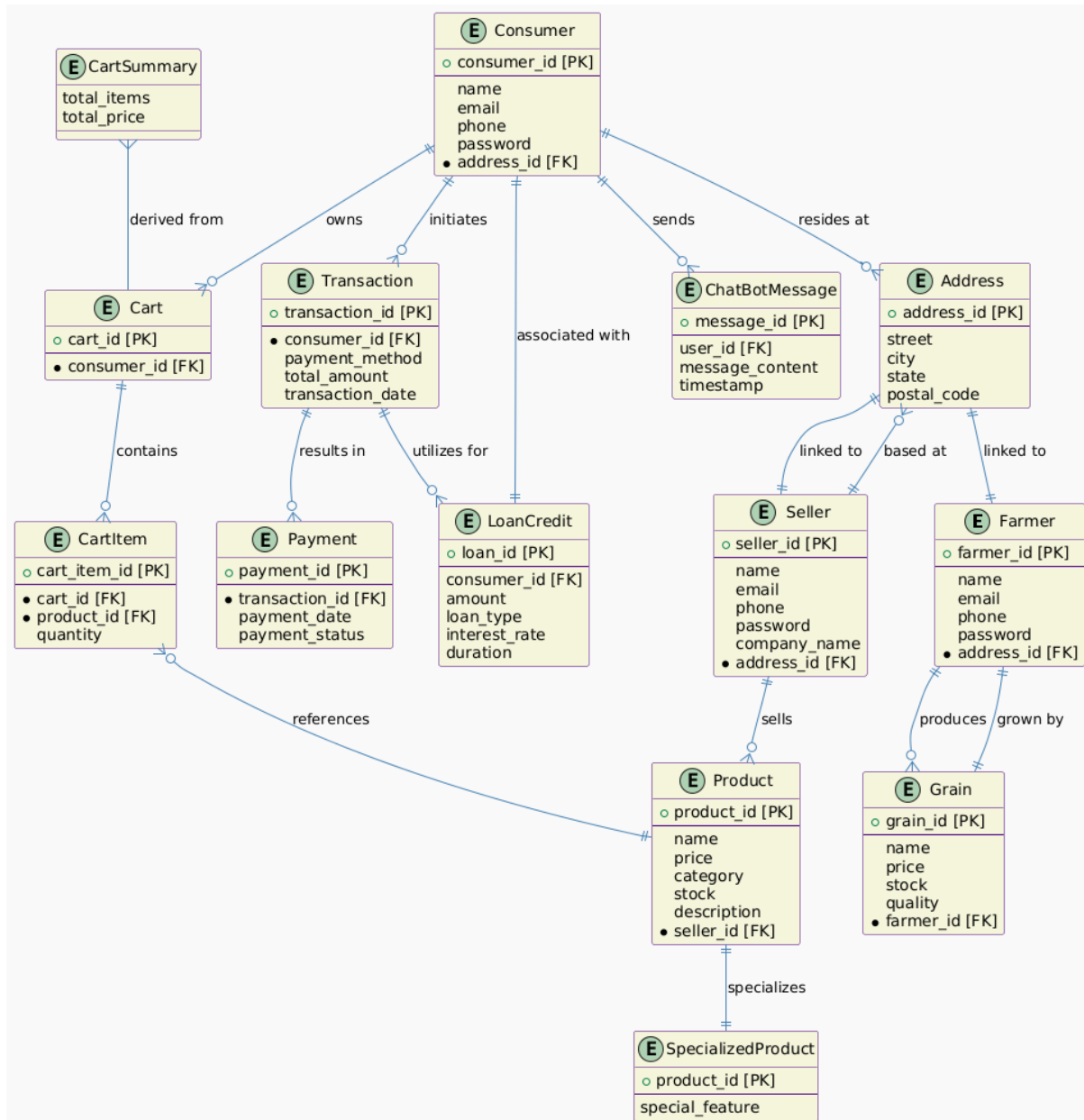


Figure 1: Main Entity-Relationship Diagram (ERD)

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.

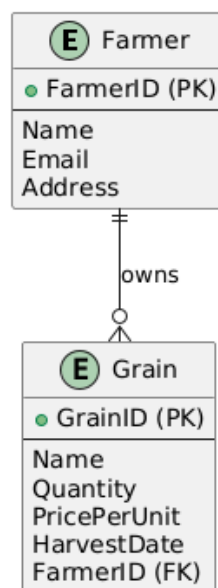


Figure 2: Farmer Entity-Relationship Diagram (ERD)

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.

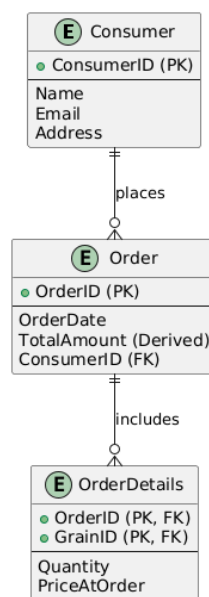


Figure 3: Consumer Entity-Relationship Diagram (ERD)

3.2.3 Suppliers Entity Sub-Diagram

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

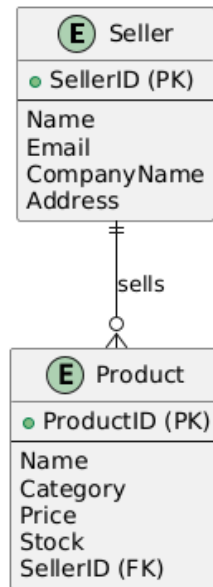


Figure 4: supplier Entity-Relationship Diagram (ERD)

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.

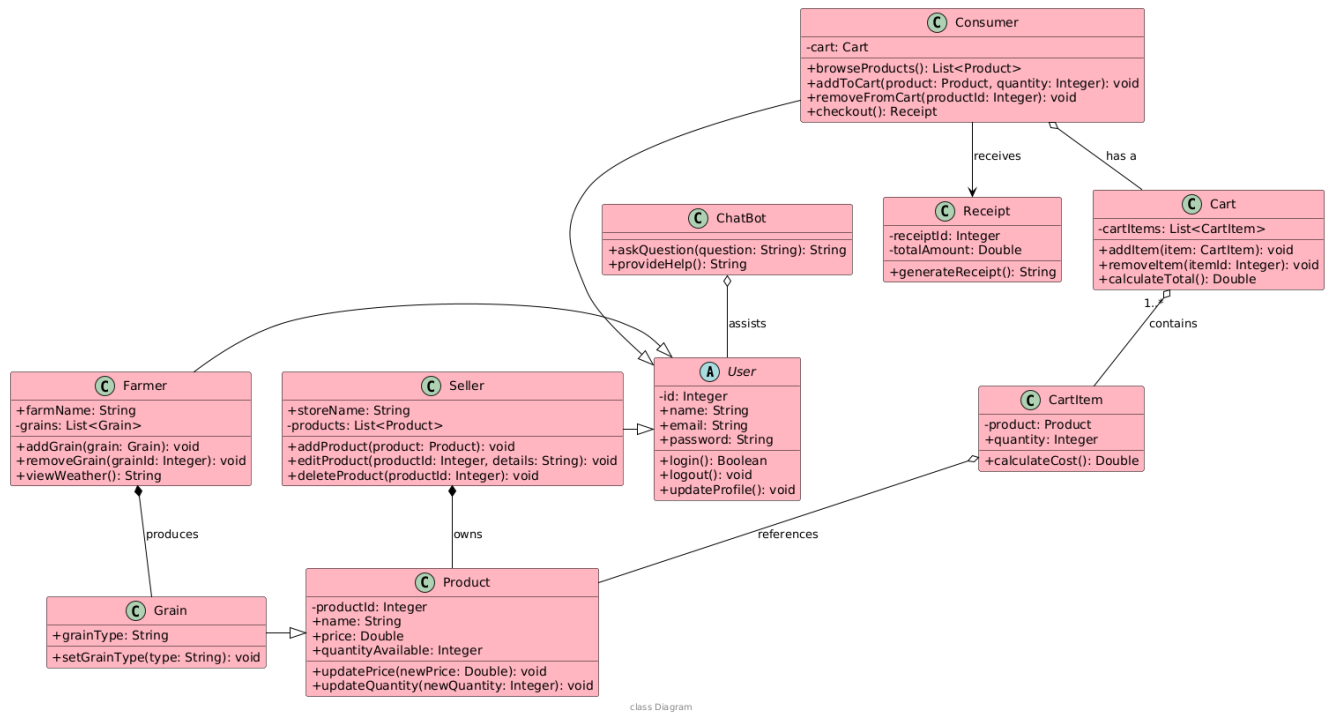


Figure 5: Main Class Diagram for Agri Vision

- 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.

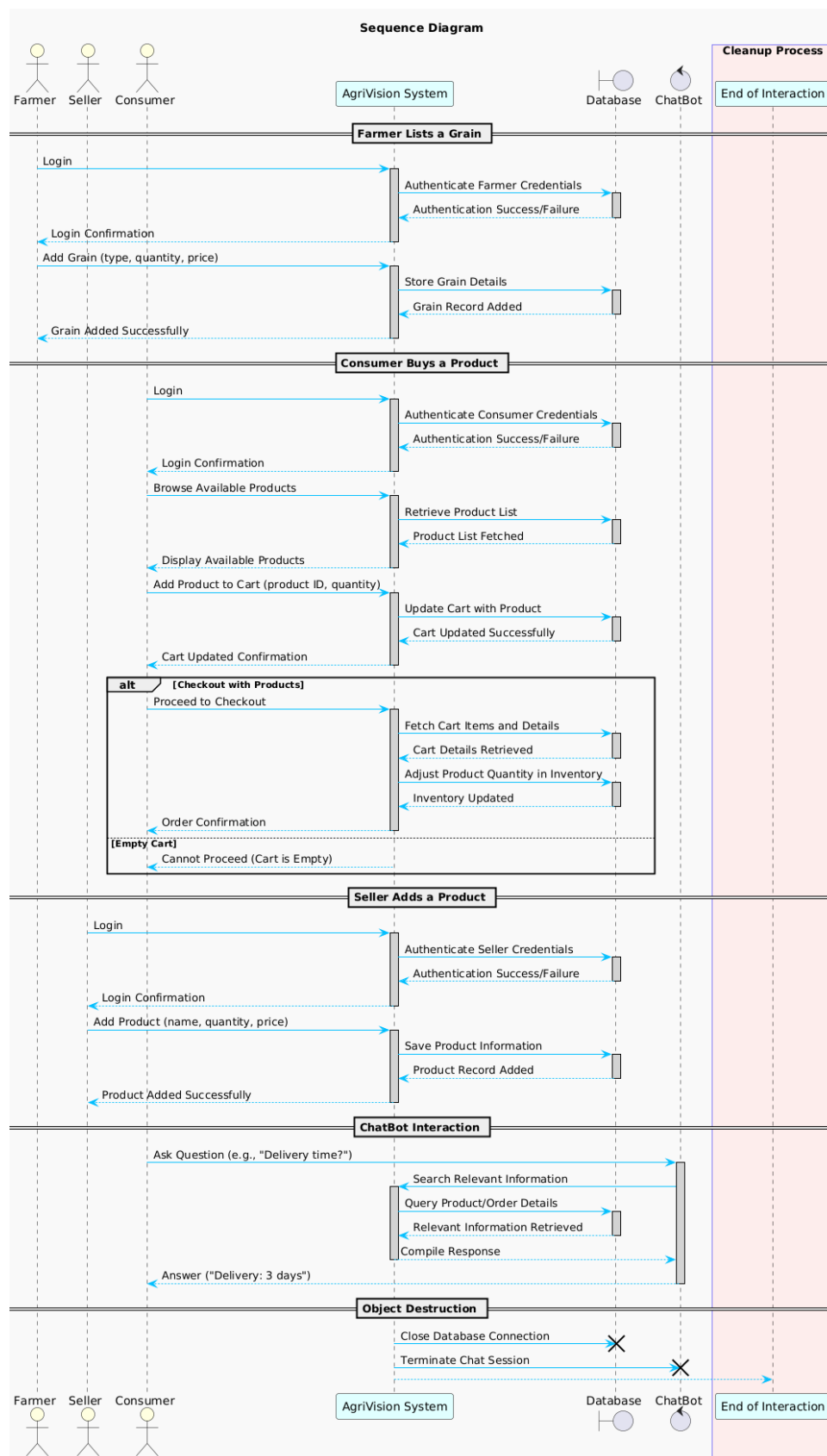


Figure 6: Sequence Diagram

5.2 Sub-Diagrams

5.2.1 Farmer Sequence Diagram

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

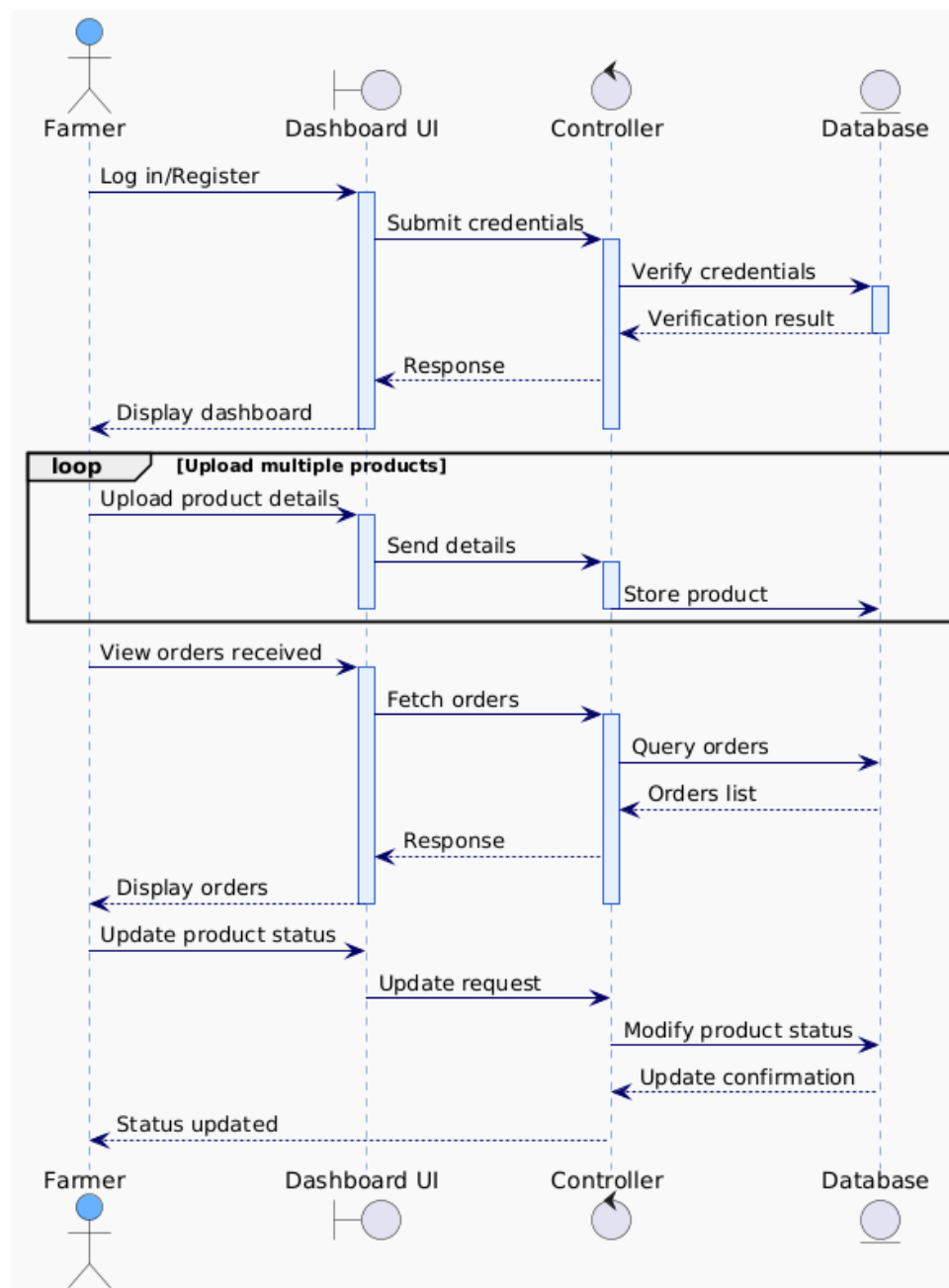


Figure 7: Sequence for Farmer

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.

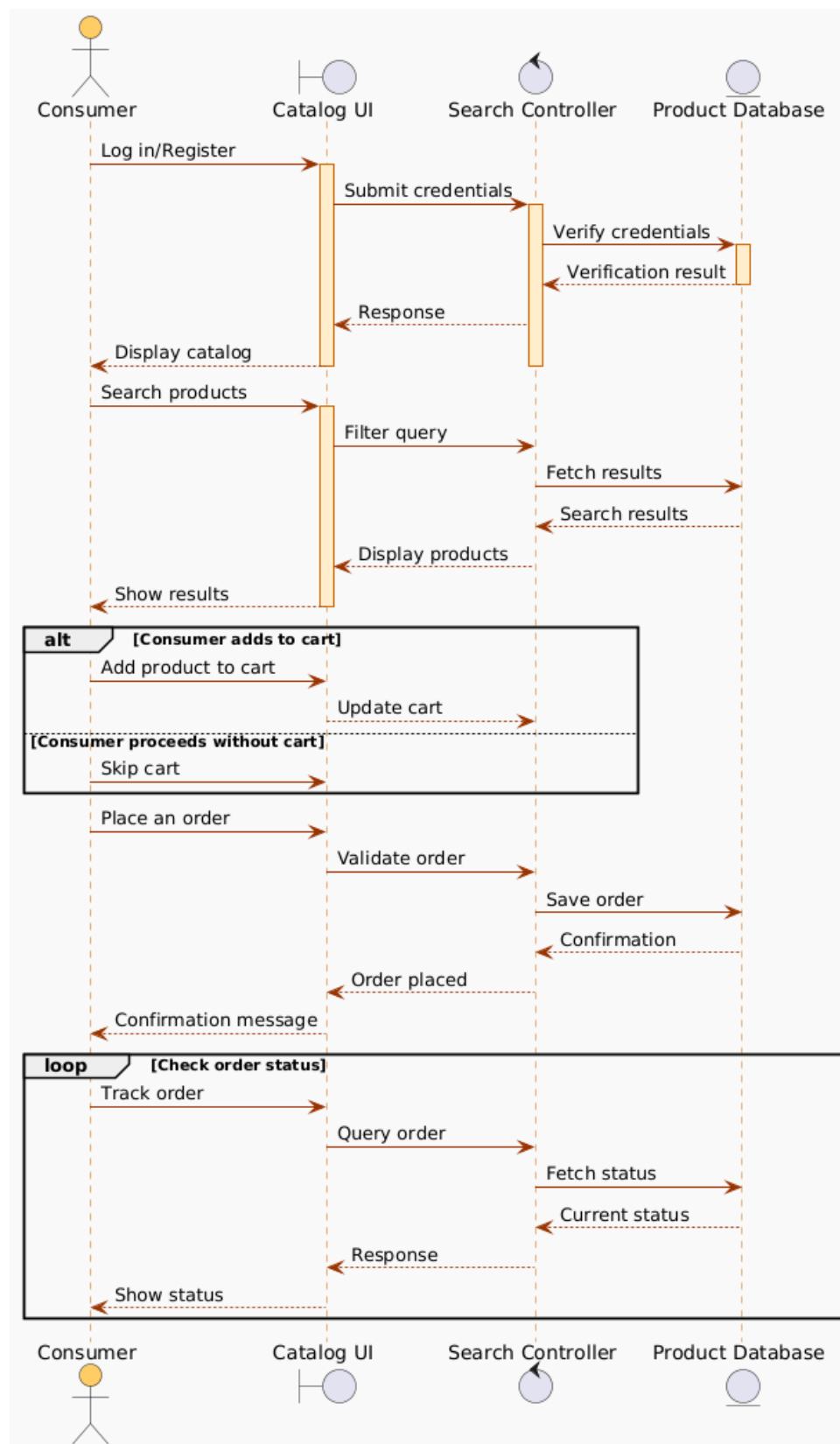


Figure 8: Sequence for Consumers

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.

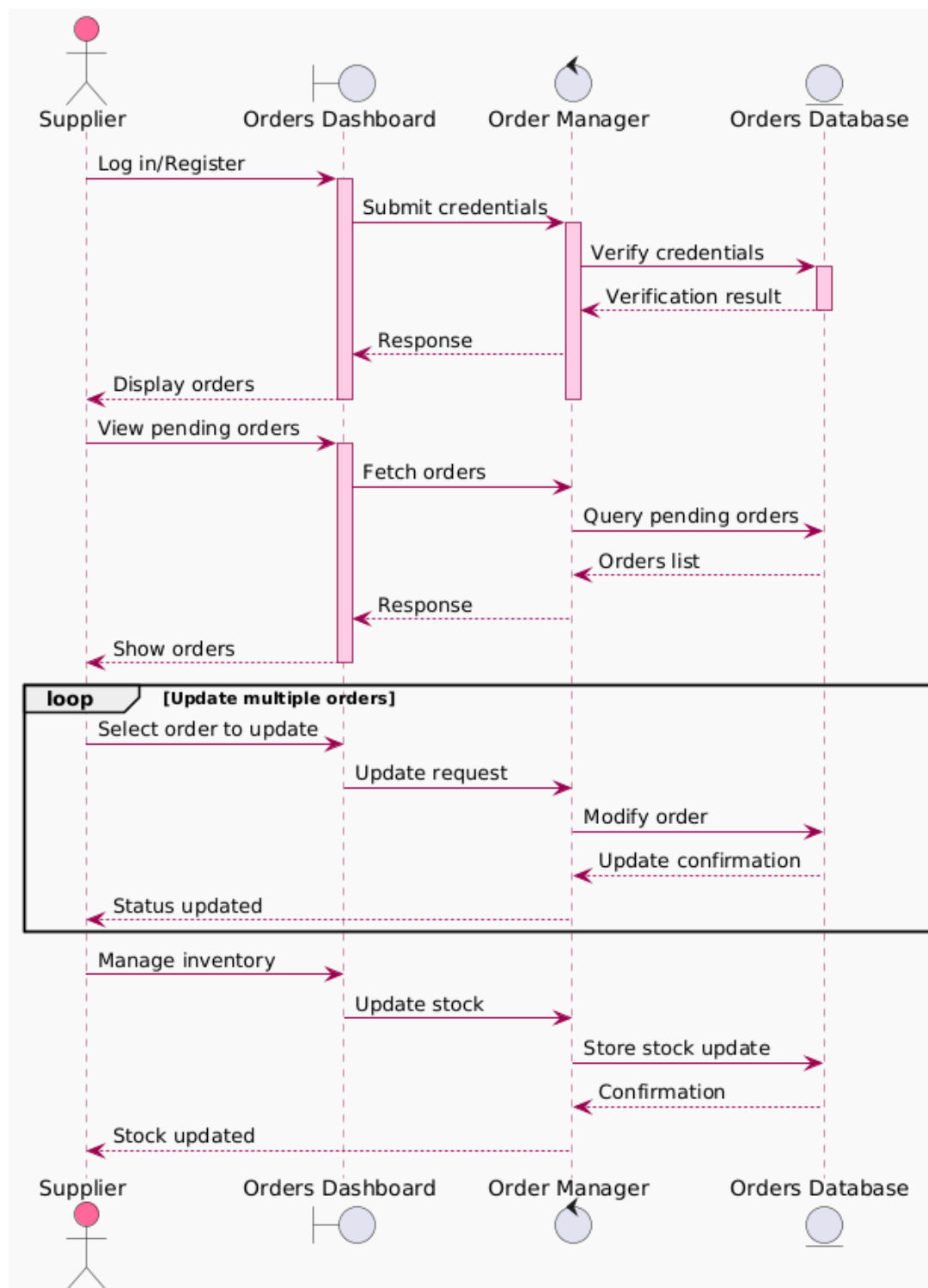


Figure 9: Sequence for Suppliers

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.

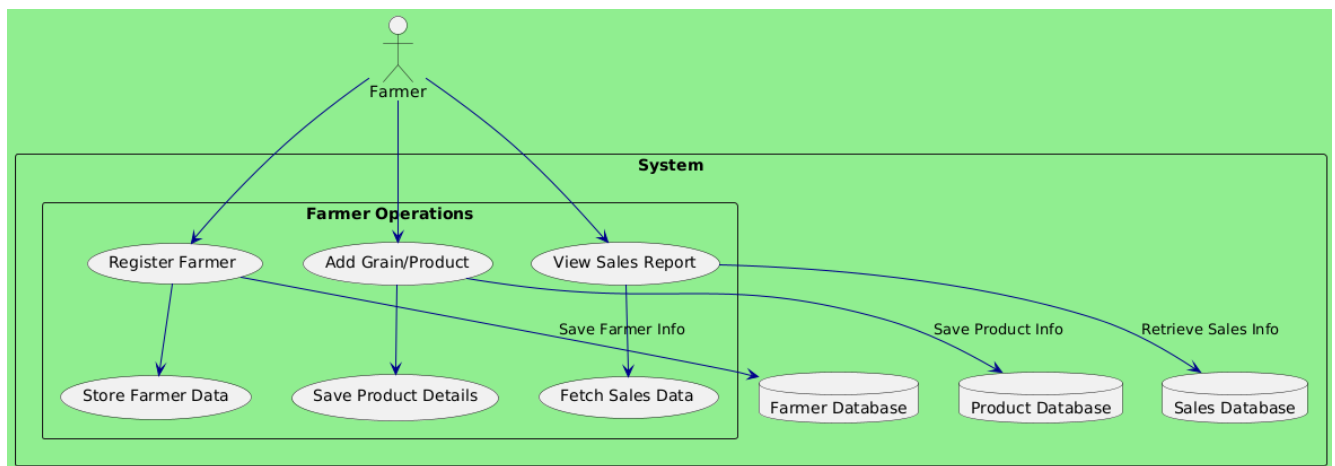


Figure 10: Data flow at farmer

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.

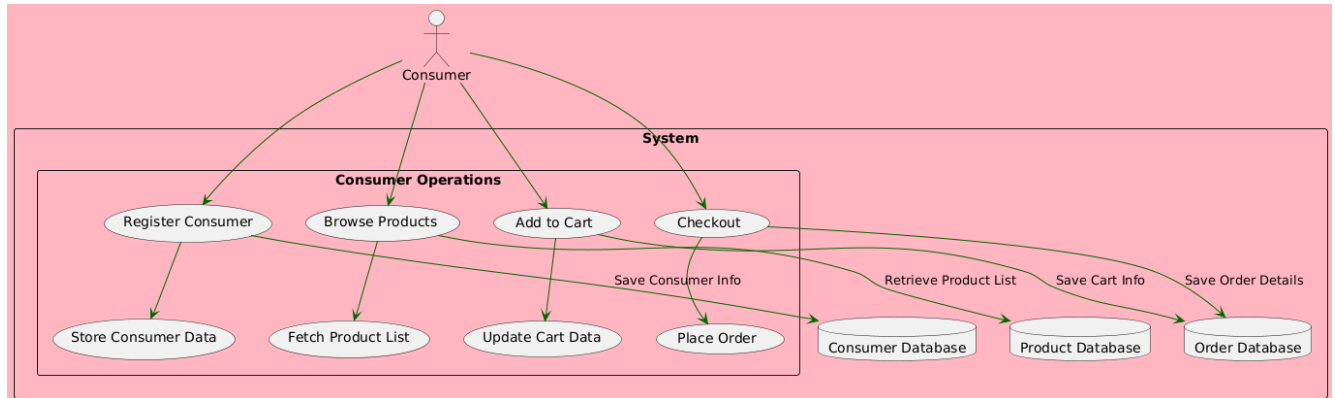


Figure 11: Data flow at Consumer

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.

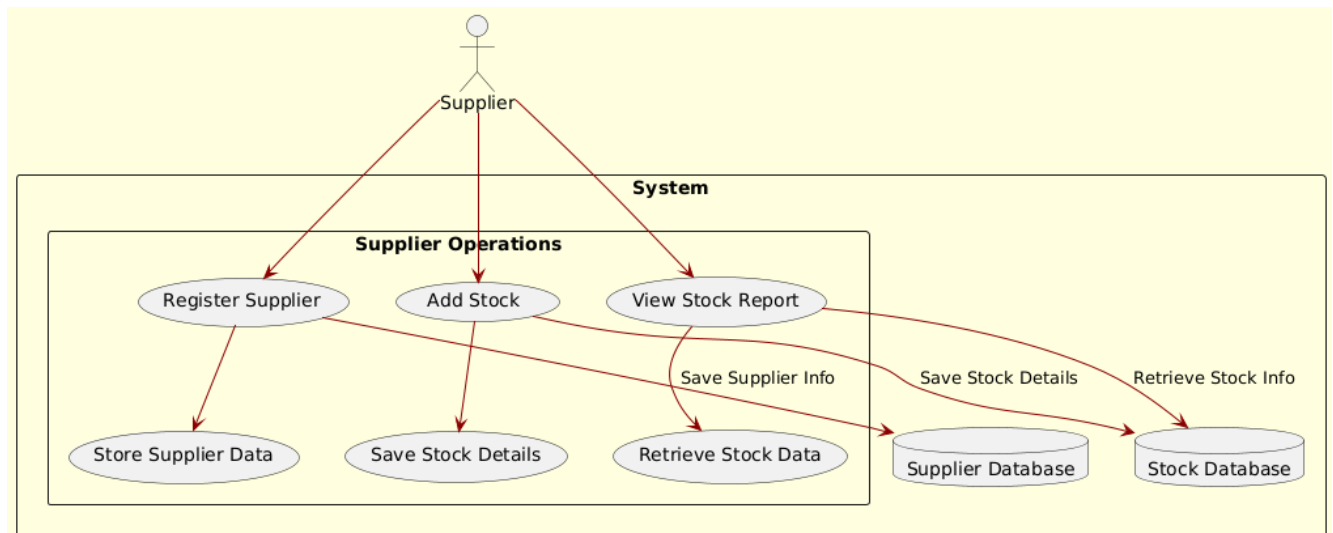


Figure 12: Data flow at supplier.

- **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.

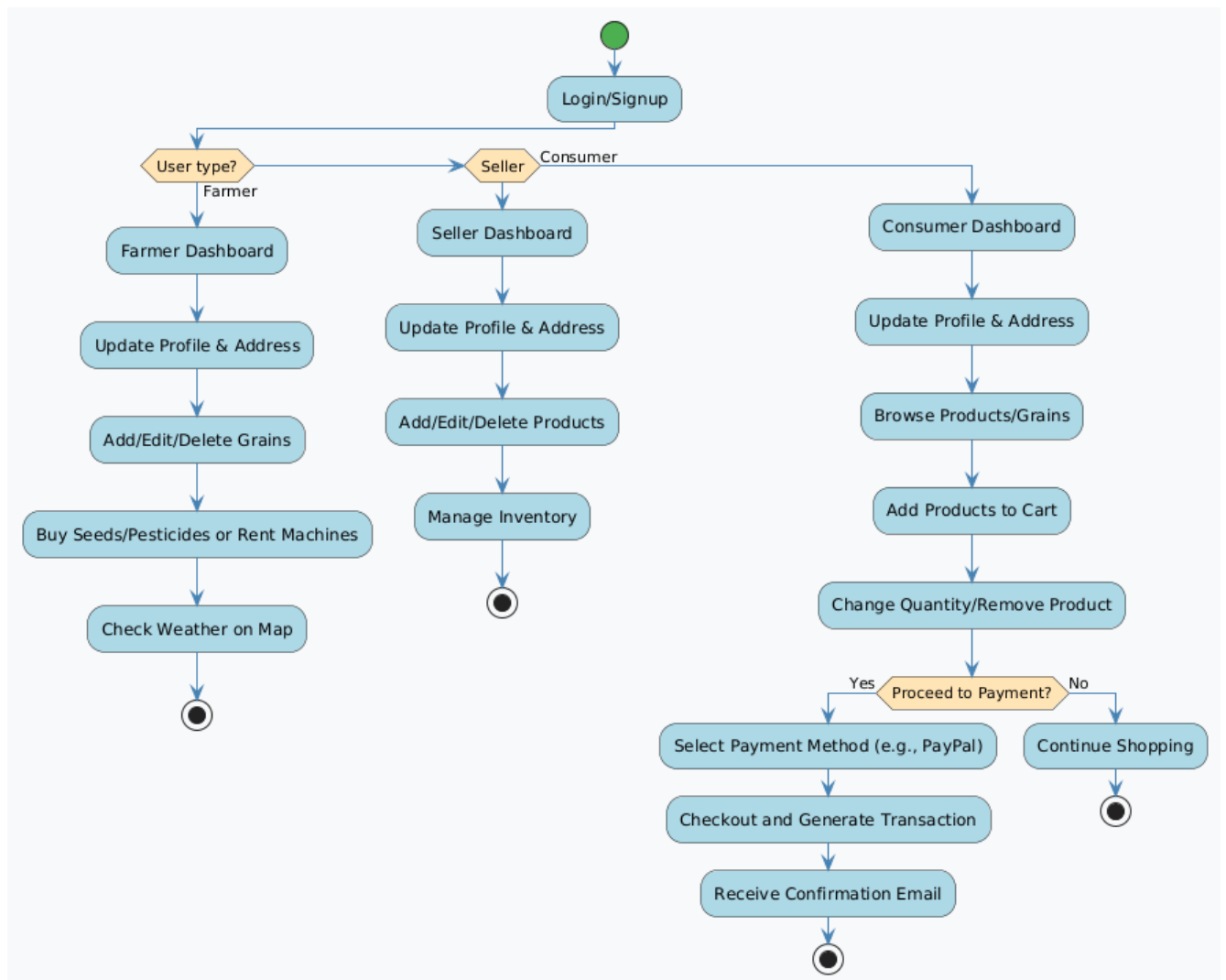


Figure 13: Main Activity Diagram: User Interaction Flow

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.

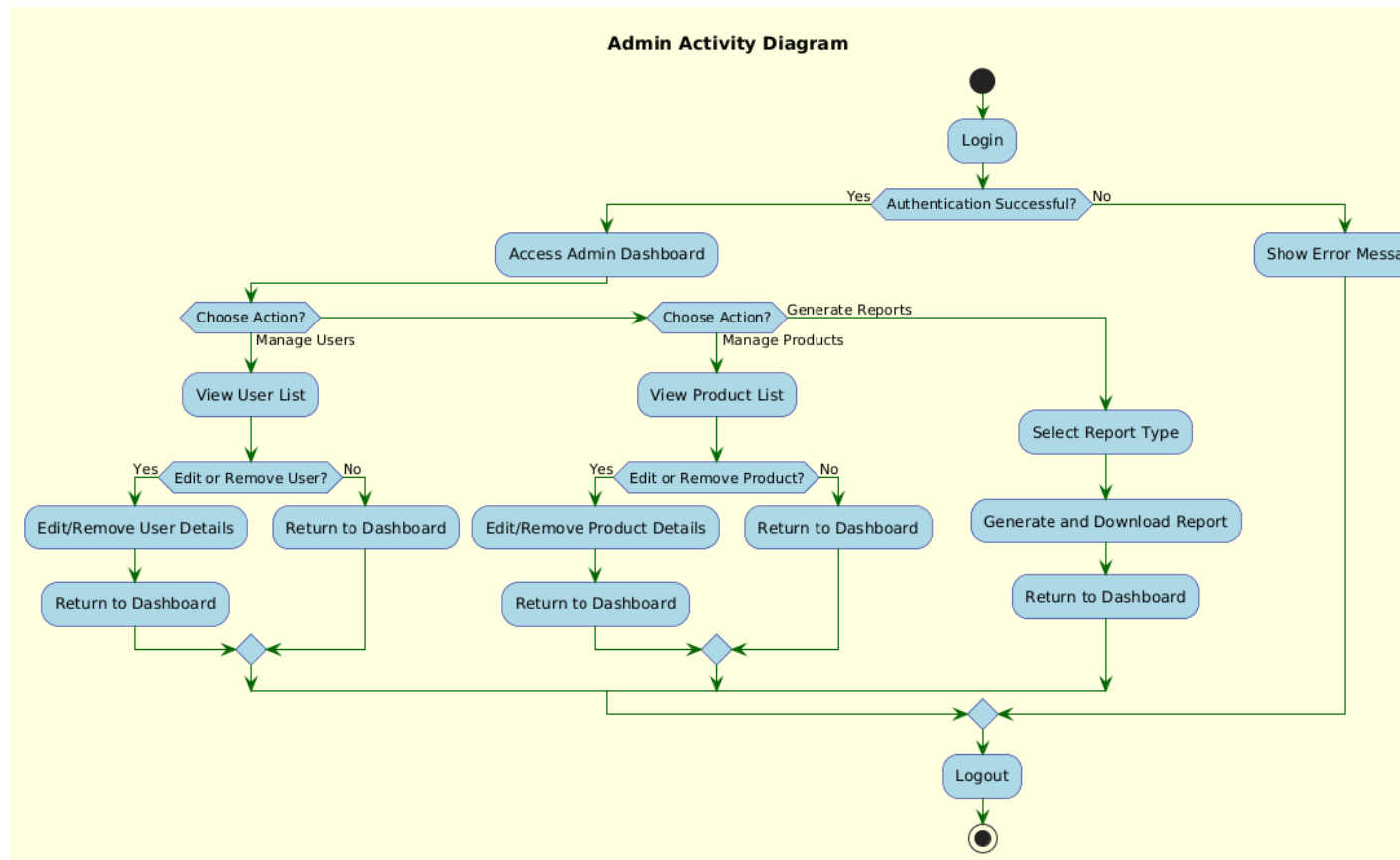


Figure 14: Admin Process Activity Diagram

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.

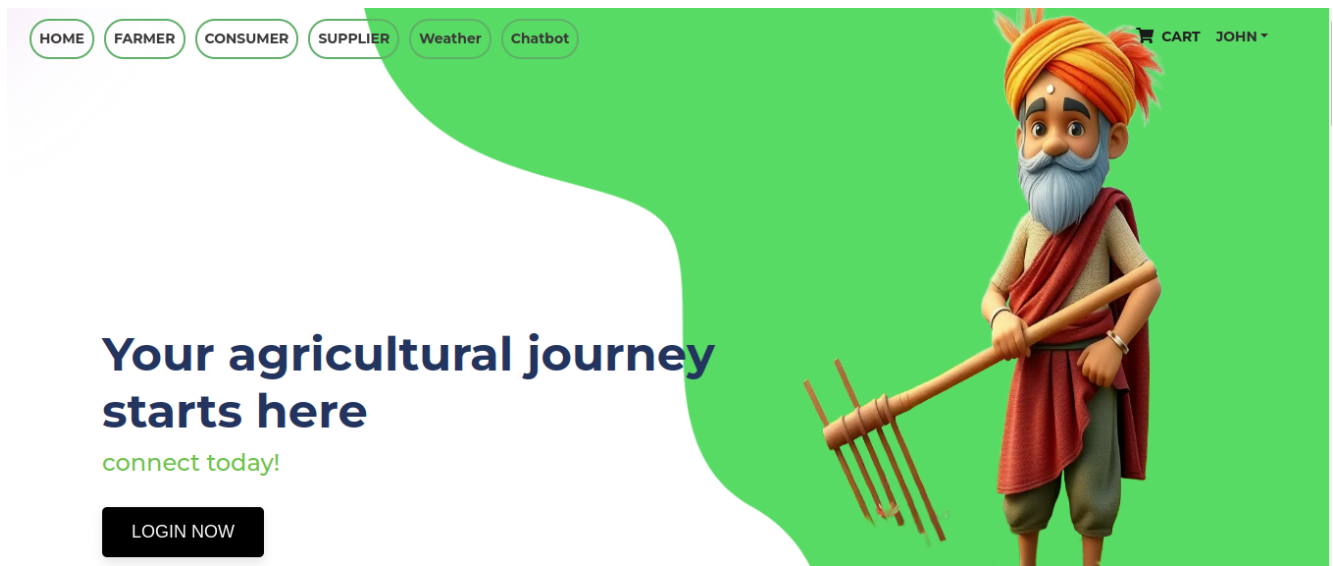


Figure 15: Agri vision Landing Page

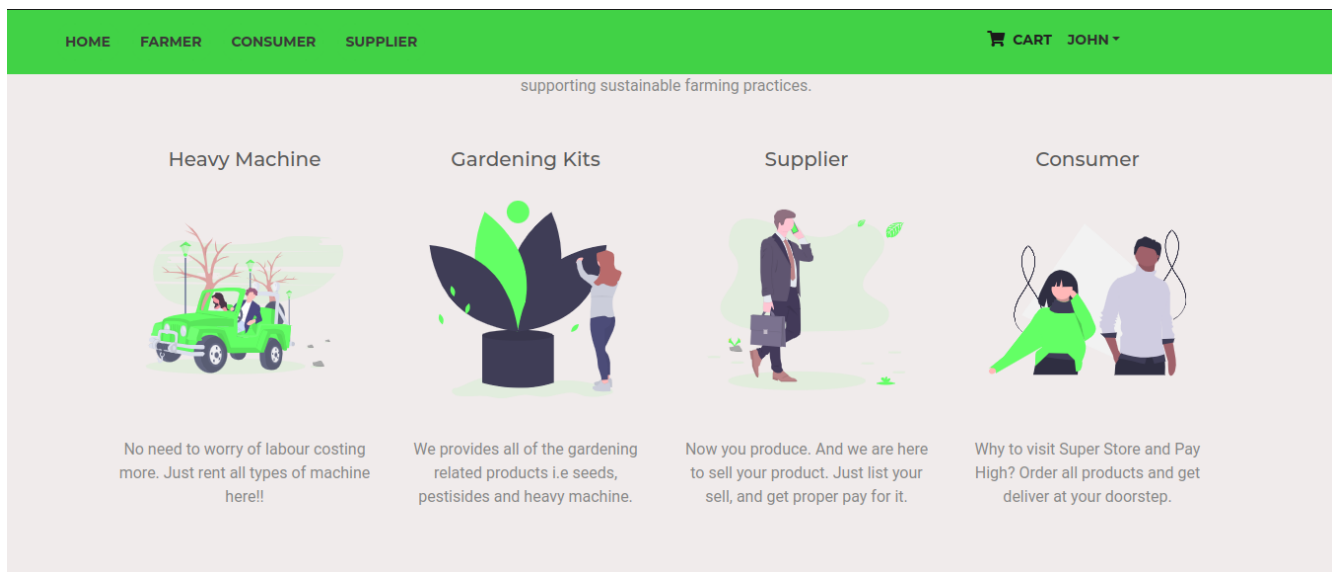


Figure 16: Landing page of AgriVision

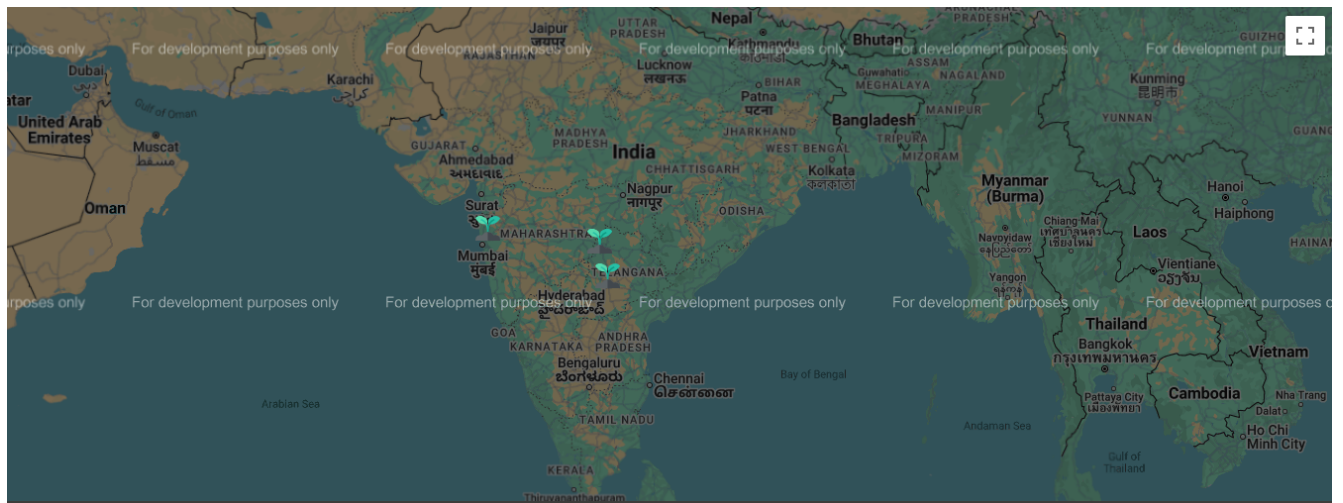


Figure 17: weather map

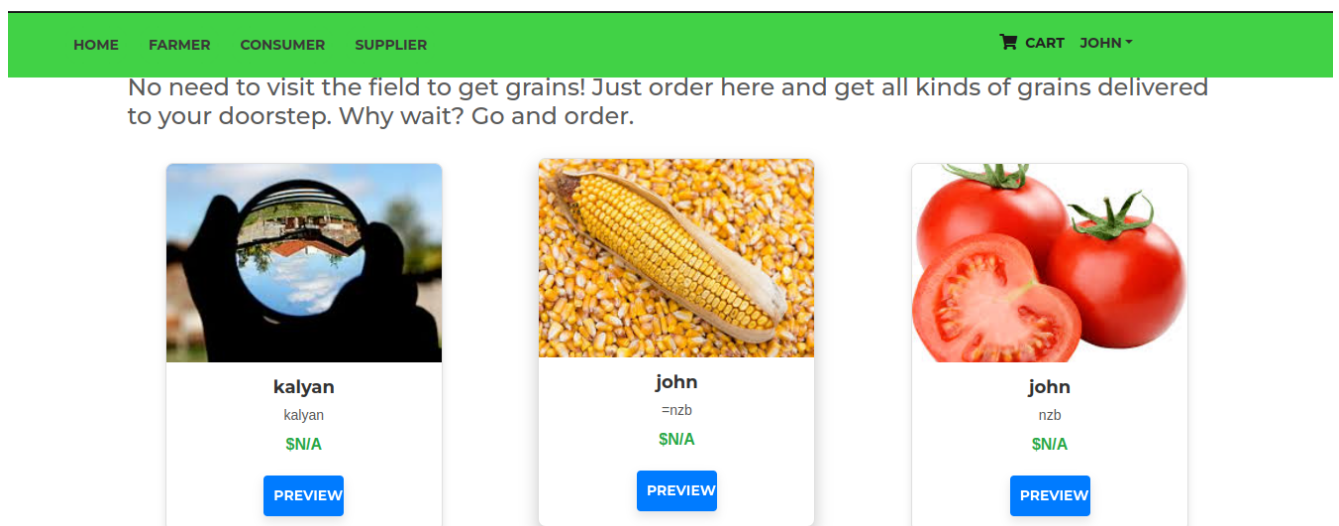


Figure 18: Consumer section

[HOME](#) [FARMER](#) [CONSUMER](#) [SUPPLIER](#) [CART](#) [JOHN](#)

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 19: Supplier section

[HOME](#) [FARMER](#) [CONSUMER](#) [SUPPLIER](#) [CART](#) [JOHN](#)

Shopping Cart

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

Subtotal (2) items

RS.100.00

PROCEED TO CHECKOUT

Figure 20: Cart section

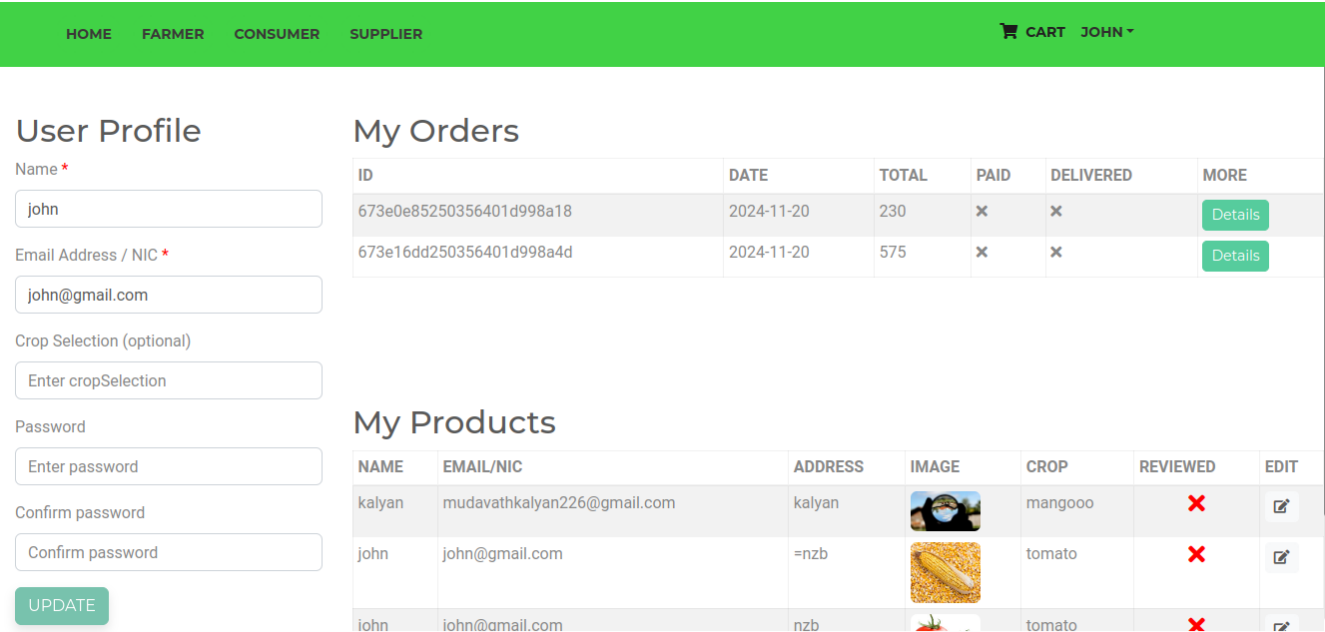


Figure 21: Profile section

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.