**Practical No 3:**

**Modeling UML Use Case Diagrams and Capturing Use Case Scenarios for Agriculture Management System**

In this practical, we will model a **UML Use Case Diagram** and capture **Use Case Scenarios** for an **Agriculture Management System (AMS)**.

**Step 1: Identify Actors**

Before we create the Use Case diagram, let’s first identify the **actors** (users or systems that interact with the AMS). For an Agriculture Management System, the following actors can be identified:

* **Farmer**: Main user of the system, responsible for managing crops, livestock, and resources.
* **Agricultural Consultant**: Provides expert advice, helps with crop management, and recommends improvements.
* **Admin**: Manages the overall system, user roles, and keeps track of farm performance.
* **Supplier**: Provides resources like seeds, fertilizers, tools, and machinery to the farmers.
* **System**: The software system itself that performs automated tasks, data analysis, and generates reports.
* **Government/Regulatory Agency**: Might interact with the system to monitor the agricultural outputs for regulations and subsidies.

**Step 2: Identify Use Cases**

For each actor, we will identify the **use cases**, which are the tasks or actions that the actors will perform in the system.

Here are some possible **use cases** for the **Agriculture Management System**:

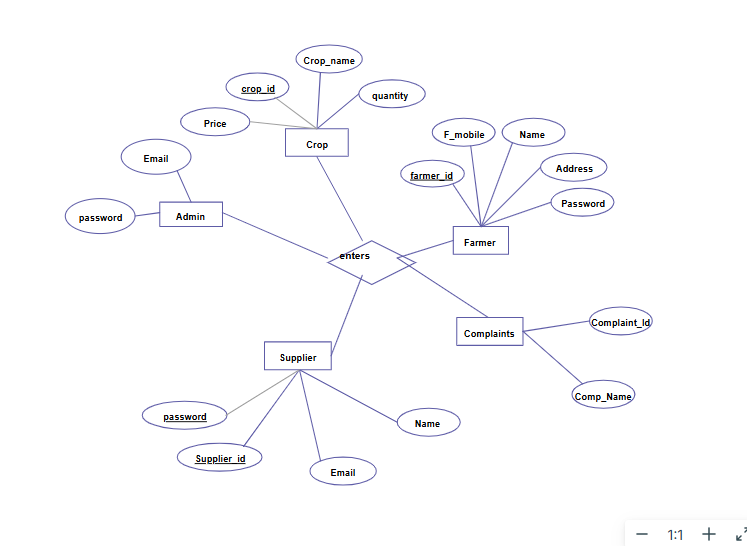
1. **Farmer**:
   * **View Crop Management Dashboard**: View the status of crops, growth, irrigation needs, etc.
   * **Register Crop Types**: Register different types of crops (e.g., vegetables, grains).
   * **Track Crop Growth**: Monitor growth stages, water needs, and fertilizers.
   * **Manage Livestock**: Track livestock health, production, and feeding.
   * **Place Order for Supplies**: Order seeds, fertilizers, tools, or machinery from the supplier.
   * **Request Expert Consultation**: Request advice from the Agricultural Consultant.
2. **Agricultural Consultant**:
   * **Provide Crop Management Advice**: Offer recommendations for optimizing crop growth.
   * **Offer Livestock Management Advice**: Give advice on managing livestock health, feeding, and production.
   * **Generate Reports**: Generate reports on farm performance, crop yields, etc.
3. **Admin**:
   * **Create and Manage User Accounts**: Add, update, and remove users in the system (farmers, consultants, suppliers).
   * **Manage System Settings**: Oversee the configuration of the AMS, including resources and policies.
   * **Generate System Reports**: Generate reports on system performance, crop statistics, and farm outputs.
4. **Supplier**:
   * **View Orders**: Check the orders placed by farmers for supplies.
   * **Supply Products**: Fulfill orders for seeds, fertilizers, and equipment.
5. **Government/Regulatory Agency**:
   * **Monitor Crop and Livestock Data**: View data on crops, livestock, and agricultural production to ensure compliance.
   * **Receive Reports for Subsidies and Compliance**: Receive reports from the system for subsidies, certifications, and regulations.

**Step 3: Create the UML Use Case Diagram**

Now that we have the actors and use cases, we can create the UML **Use Case Diagram** to visualize the system’s functionality.

**Key Elements in the Diagram**:

* **Actors** are represented by stick figures.
* **Use cases** are represented by ovals.
* **Associations** are represented by lines connecting actors and use cases.
* **System boundary** is shown as a rectangle around the use cases to indicate what is part of the system.

**UML Use Case Diagram for Agriculture Management System**

#### Step 4: ****Capture Use Case Scenarios****

Each **use case** needs to be described in detail, including its flow, actors, and system responses. Let’s capture the **Use Case Scenarios** for a few of the identified use cases.

##### 1. **Use Case Scenario: View Crop Management Dashboard**

* **Actor**: Farmer
* **Goal**: To view the current status of crops in the field.
* **Preconditions**: The farmer has logged into the system and has registered crop data.
* **Basic Flow**:
  1. The farmer logs into the system.
  2. The farmer selects the "Crop Management Dashboard" from the main menu.
  3. The system displays the dashboard with the current crop status, growth stages, irrigation status, and any alerts (e.g., water shortages).
  4. The farmer can interact with the dashboard to drill down into specific crops for detailed information.
* **Postconditions**: The farmer successfully views the crop dashboard.
* **Exceptions**: If there is no crop data registered, the system will display a message: "No crop data available."

##### 2. **Use Case Scenario: Track Crop Growth**

* **Actor**: Farmer
* **Goal**: To track the growth stages and conditions of crops.
* **Preconditions**: The farmer has crops registered in the system and has access to crop data.
* **Basic Flow**:
  1. The farmer selects a crop from the "Track Crop Growth" section.
  2. The system displays the current growth stage (e.g., germination, flowering, etc.), irrigation needs, and any fertilizer recommendations.
  3. The farmer can update the status (e.g., watering completed, fertilizer applied).
  4. The system logs the updates and adjusts recommendations accordingly.
* **Postconditions**: The crop growth information is updated, and any recommendations are provided.
* **Exceptions**: If there is insufficient data, the system will prompt the user to update the crop details.

##### 3. **Use Case Scenario: Place Order for Supplies**

* **Actor**: Farmer
* **Goal**: To place an order for supplies (seeds, fertilizers, tools).
* **Preconditions**: The farmer is logged into the system and has access to a catalog of supplies.
* **Basic Flow**:
  1. The farmer navigates to the "Order Supplies" section.
  2. The farmer selects the items needed (e.g., seeds, fertilizers, tools).
  3. The system shows the available items, their quantities, and prices.
  4. The farmer reviews the order and places it.
  5. The system confirms the order and sends a notification to the supplier.
* **Postconditions**: The order is successfully placed, and the system notifies the supplier.
* **Exceptions**: If an item is out of stock, the system will notify the farmer and suggest alternatives.

##### 4. **Use Case Scenario: Provide Crop Management Advice**

* **Actor**: Agricultural Consultant
* **Goal**: To provide advice on crop management to the farmer.
* **Preconditions**: The farmer has requested advice on crop management.
* **Basic Flow**:
  1. The agricultural consultant logs into the system and selects "Provide Advice."
  2. The system presents available crop information from the farmer (e.g., crop type, growth stage, health).
  3. The consultant reviews the data and provides recommendations (e.g., irrigation, pest control, fertilizers).
  4. The system sends the advice to the farmer.
* **Postconditions**: The farmer receives expert advice.
* **Exceptions**: If the consultant cannot provide advice, the system will suggest an alternative.

#### Conclusion

This practical exercise illustrates the process of creating a **UML Use Case Diagram** and capturing detailed **Use Case Scenarios** for an **Agriculture Management System**. By following these steps, you can effectively model the key features of any system and ensure that all user interactions are clearly defined and documented.