



**National Institute of Business Management  
School of Computing and Engineering**

**Diploma in Computer Science with AI**

**Batch 25.1**

**Semester 2**

**Object Oriented Programming**

**Group no : 11**

**Index no : KIC-DCSAI-251-F-011**

**Index no : KIC-DCSAI-251-F-036**

**Index no : KIC-DCSAI-251-F-041**

## 1. Real – World Problem

Farming isn't easy. You're juggling soil moisture, nutrients, and the weather—none of them ever really play along. That's why the AgroVision Smart Farming System stands out.

With it, farmers get a clear picture of what's happening in their fields, right as it happens, and they can actually do something about it. Want to change up irrigation or give a certain area more fertilizer? Just hop onto the dashboard and make it happen, zone by zone. The automation takes so much of the guesswork out, so instead of crossing your fingers, you're making smart calls as things shift.

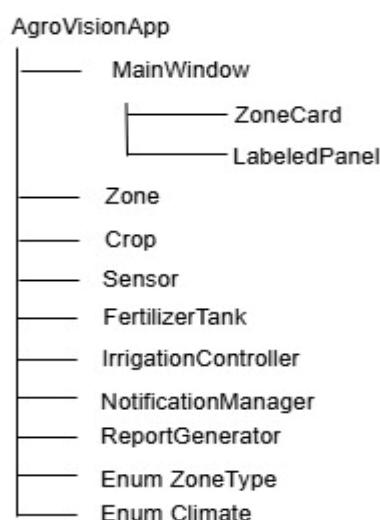
AgroVision really proves how IoT and AI can change the game for farming. You're not stuck doing things the old way anymore. You finally have control—and you can see the difference.

## 2. Objective

- Automatically apply water and fertilizer based on crop requirements and environmental conditions.
- Offer an intuitive user interface for monitoring multiple farm zones.
- Produce reports and alerts to aid in decision-making.
- Real-time crop growth and maintenance simulation
- 

## 3. Class Diagram

The diagram below shows how the main classes of the AgroVision system relate to one another.



## 4. OOP Concepts

- Encapsulation :

Getters and setters are used to encapsulate attributes in classes such as **Zone**, **Crop**, **Sensor**, and **FertilizerTank**.

Each zone's behavior is encapsulated by UI elements such as **ZoneCard**.

- Inheritance :

**JFrame** is extended by **MainWindow**.

**JPanel** is extended by **ZoneCard** and **LabeledPanel**.

- Polymorphism :

Polymorphism is used by swing components through **JComponent** references.

Any **ZoneCard** object can use the **updateUIFromZone()** method.

- Abstraction :

**IrrigationController** and **NotificationManager** Complex simulation logic is hidden by the manager.

User interaction is limited to the GUI.

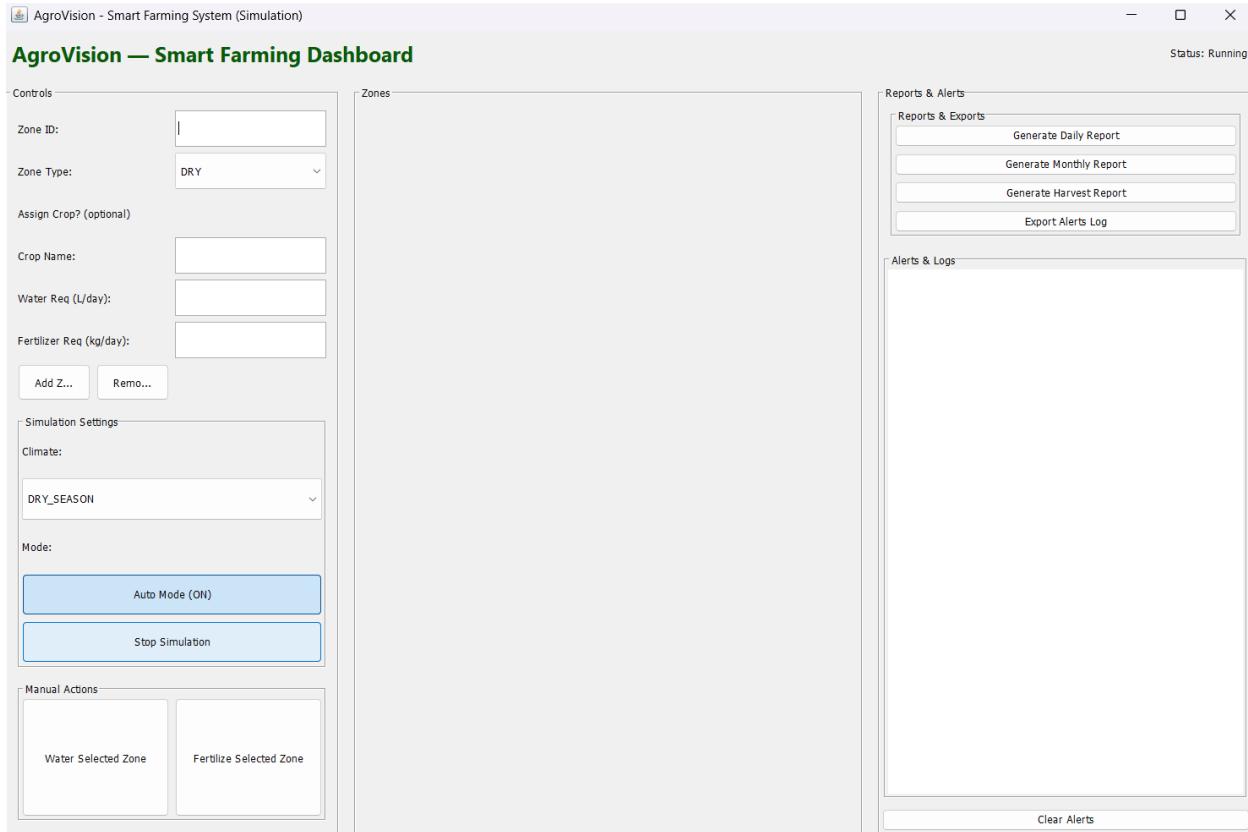
- Composition :

**Sensor** and **FertilizerTank** are located in **Zone**.

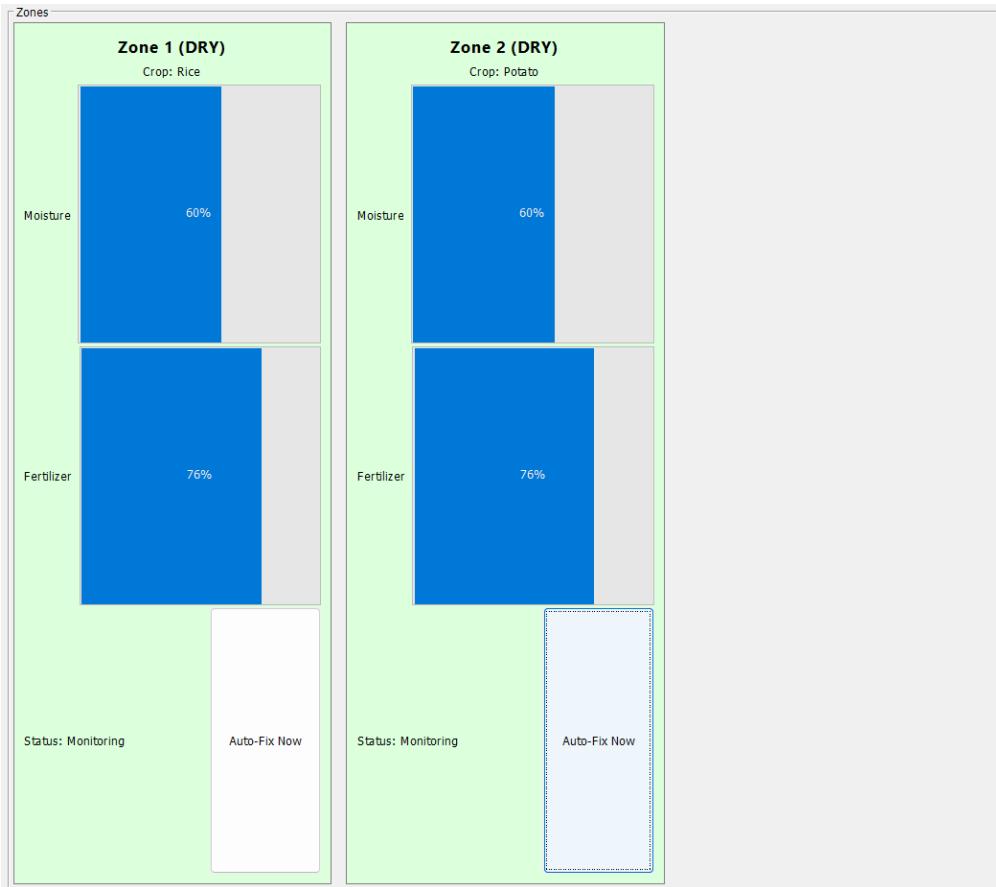
There are several **ZoneCards** in the **MainWindow**.

## 5. Screenshots

### • Dashboard



## • Zone Card



## • Reports & Alerts

Alerts & Logs
[13:41:23] Zone 1 manually fixed (reset levels)
[13:41:24] Zone 2 manually fixed (reset levels)
[13:42:52] Zone 1 auto watered 10.0L
[13:42:52] Zone 2 auto watered 10.0L
[13:43:04] Zone 2 manually fixed (reset levels)
[13:43:05] Zone 1 manually fixed (reset levels)
[13:43:09] Zone 1 manually watered

## **6. Features**

- Zones can be added and removed dynamically.
- Assign crops to areas that need fertilizer and water.
- Both manual and automated fertilization and irrigation.
- Simulated real-time sensors for fertilizer and moisture levels.
- Zone-specific color-coded status:
  - I. Green: Nutritious
  - II. Yellow: Moderate
  - III. Red: Vital
- Create harvest, monthly, and daily reports.
- Export logs and alerts.
- Reset fertilizer and moisture levels with auto-fix zones.

## **7. Conclusion**

In conclusion, AgroVision's interactive GUI and OOP principles allow it to provide a complete smart farming simulation. It helps farmers or students monitor and manage multiple zones, ensuring the best use of water and fertilizer. Future projects can expand the system to incorporate real IoT devices for farm monitoring.