VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

PROJECT PHASE - II REPORT

on

"Green House Automated System for Sustainable Agriculture using IoT and Machine Learning"

Submitted by

Srishreya	4SF21IS109
Yogesh Shivanand Patgar	4SF21IS125
P Samarth Shenoy	4SF22IS406
Sujan	4SF22IS410

In partial fulfillment of the requirements for the VI semester

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Under the Guidance of

Mrs. Masooda

Assistant Professor, Department of ISE

at

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

2024 - 25

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

Department of Information Science & Engineering

CERTIFICATE

This is to certify that the phase - II work of project entitled **"Green House Automated System for Sustainable Agriculture using IoT and Machine Learning"** has been carried out by **Srishreya (4SF21IS109), Yogesh Shivanand Patgar (4SF21IS125), P Samarth Shenoy (4SF22IS406) and Sujan (4SF22IS410)**, the bonafide students of Sahyadri College of Engineering and Management in partial fulfillment of the requirements for the VII semester of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2024 - 25. It is certified that all suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Project Guide	Project Convener	HOD
Mrs. Masooda	**Mr. Vasudeva Rao P V**	**Dr. Rithesh Pakkala P.**
Assistant Professor	Assistant Professor	Associate Professor & Head
Dept. of ISE, SCEM	Dept. of ISE, SCEM	Dept. of ISE, SCEM

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

Department of Information Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Project Phase - II Report titled **"Green House Automated System for Sustainable Agriculture using IoT and Machine Learning "** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Mrs. Masooda**, in partial fulfillment of the requirements for the VII semester of **Bachelor of Engineering** in **Information Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Srishreya (4SF21IS109)

Yogesh S Patgar (4SF21IS125)

P Samarth Shenoy (4SF22IS406)

Sujan (4SF22IS410)

Dept. of ISE, SCEM, Mangaluru

Abstract

In recent years, the Internet of Things (IoT) has revolutionized various sectors, including agriculture, by transforming traditional farming practices into more advanced, technology-driven approaches. This study explores the integration of IoT and machine learning in greenhouse automation, proposing a comprehensive IoT-based network framework aimed at sustainable and efficient resource management in greenhouse environments. The research highlights the significance of smart farming technologies, specifically in greenhouses, and provides a systematic analysis of high-quality research in this area, including sensors/devices and communication protocols. Furthermore, this study addresses the challenges and security issues in IoT-based smart greenhouse farming and suggests future research directions to enhance these systems.

In addition to IoT, the application of machine Learning (ML) in greenhouse systems offers a promising solution to global food insecurity, particularly in regions affected by adverse climatic conditions. This research presents a fully automated greenhouse system equipped with ML, utilizing around 10,000 plant images for real-time decision-making, disease detection, and monitoring of fruit ripeness stages. The implementation of neural network-based computer vision techniques enables the system to accurately track plant health, enhancing productivity and food security. The findings underscore the potential of combining IoT and ML technologies to significantly improve agricultural practices and ensure food security in various farming areas without extensive human intervention.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Project Phase-II Report on **"Green House Automated System for Sustainable Agriculture using IoT and Machine Learning"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi in partial fulfillment of the requirements for the VII semester of Bachelor of Engineering in Information Science & Engineering.

We are profoundly indebted to our guide, **Mrs. Masooda**, Assistant Professor, Department of Information Science & Engineering for innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We also thank **Mr. Vasudeva Rao P V**, Project Convener, Department of Information Science & Engineering for his constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Rithesh Pakkala P.**, Associate Professor and Head, Department of Information Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. S. S. Injaganeri**, Principal, Sahyadri College of Engineering & Management who has always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work.

Srishreya (4SF21IS109)
Yogesh S Patgar (4SF21IS125)
P Samarth Shenoy (4SF22IS406)
Sujan (4SF22IS410)
Dept. of ISE, SCEM, Mangaluru

Table of Contents

List of Figures

Chapter 1

Introduction

The growing global population, industrialization, and climate changes are decreasing arable land, thereby increasing the demand for sustainable food production. According to the United Nations Food and Agriculture Organization (FAO), the world population is projected to reach 9.73 billion by 2050, necessitating more cropland and water to meet food demands. Traditional farming methods, including conventional greenhouse practices, are challenged by labor shortages, water scarcity, and climate variability. Greenhouse farming, which allows for year-round cultivation under controlled conditions, has emerged as a viable solution. conventional greenhouses rely heavily on manual processes, which are labor-intensive and costly.

The integration of Internet of Things (IoT) technology into greenhouse farming offers a transformative solution by enhancing efficiency and reducing manual labor. IoT-enabled systems use smart devices and sensors to monitor and control farming variables such as soil moisture, air temperature, and plant health. These systems provide real-time data, enabling precise management of growing conditions and optimizing resource use. Additionally, IoT-based applications include advanced fertilization and irrigation techniques, disease control, and environmental monitoring, all of which contribute to increased productivity and reduced costs.

Machine Learning (ML) complements IoT by analyzing large datasets generated from sensors, predicting crop productivity, and identifying plant health issues. ML algorithms enable automated systems to make informed decisions, such as when to apply treatments, thus minimizing crop loss and enhancing yield. The combination of IoT and ML technologies in fully automated greenhouses allows for seamless monitoring, reduced human intervention, and consistent crop quality.

The integration of IoT and ML technologies in greenhouse farming is also a step towards achieving sustainability goals. These technologies help in conserving water and energy by optimizing irrigation schedules and lighting systems, respectively. For example, sensor data can inform irrigation systems to water plants only when necessary, reducing water wastage. Similarly, ML algorithms can analyze light patterns and adjust artificial lighting to ensure plants receive the optimal amount of light for photosynthesis while minimizing energy use. This not only reduces operational costs but also lessens the environmental impact of greenhouse farming. As the global community becomes increasingly aware of environmental issues, the adoption of such smart farming techniques can play a crucial role in promoting sustainable agricultural practices, ensuring food security, and meeting the nutritional needs of a growing population.

Chapter 2

Problem Satement

2.1 Identifying the Problem

As the global population grows, the demand for food is increasing, while arable land is decreasing due to industrialization and climate change. Traditional greenhouse farming faces challenges like labor shortages, water scarcity, and inefficient resource use. To meet these demands, integrating technologies like the Internet of Things (IoT) and Machine Learning (ML) offers a solution. IoT can automate processes such as irrigation and temperature control, while ML can optimize crop health and yield predictions. Together, these technologies can improve efficiency, reduce costs, and promote sustainable farming practices, ensuring food security for a growing population.

2.2 Objectives of the Project

This project aims to develop a state-of-the-art IoT-based greenhouse model to address challenges in traditional greenhouse management by focusing on the following objectives:

1. **Real-Time Monitoring, Controlling, and Predictive Capabilities**: Design an IoT-enabled system that continuously monitors essential environmental parameters such as temperature, humidity, light intensity, and soil moisture. Implement automated control mechanisms to maintain optimal conditions and integrate predictive analytics for proactive decision-making.

2. **Addressing Challenges and Bridging Gaps**: Identify and mitigate key challenges such as inconsistent environmental conditions, inefficient resource utilization,

lack of real-time monitoring, and their adverse impact on crop productivity. Highlight future research directions to enhance IoT technology in greenhouse farming, ensuring scalability, reliability, and sustainability.

3. **Development of Machine Learning Models**: Build and train machine learning models to predict optimal environmental conditions for plant growth. Leverage these models to dynamically adjust greenhouse settings, thereby improving resource efficiency, crop yield, and overall system performance.

4. **User-Friendly Interaction for Farmers**: Develop an intuitive and accessible interface for farmers to interact with the automated system. Enable farmers to view real-time data, control greenhouse settings manually when necessary, and receive actionable recommendations for better decision-making.

Chapter 3

Architecture Diagram

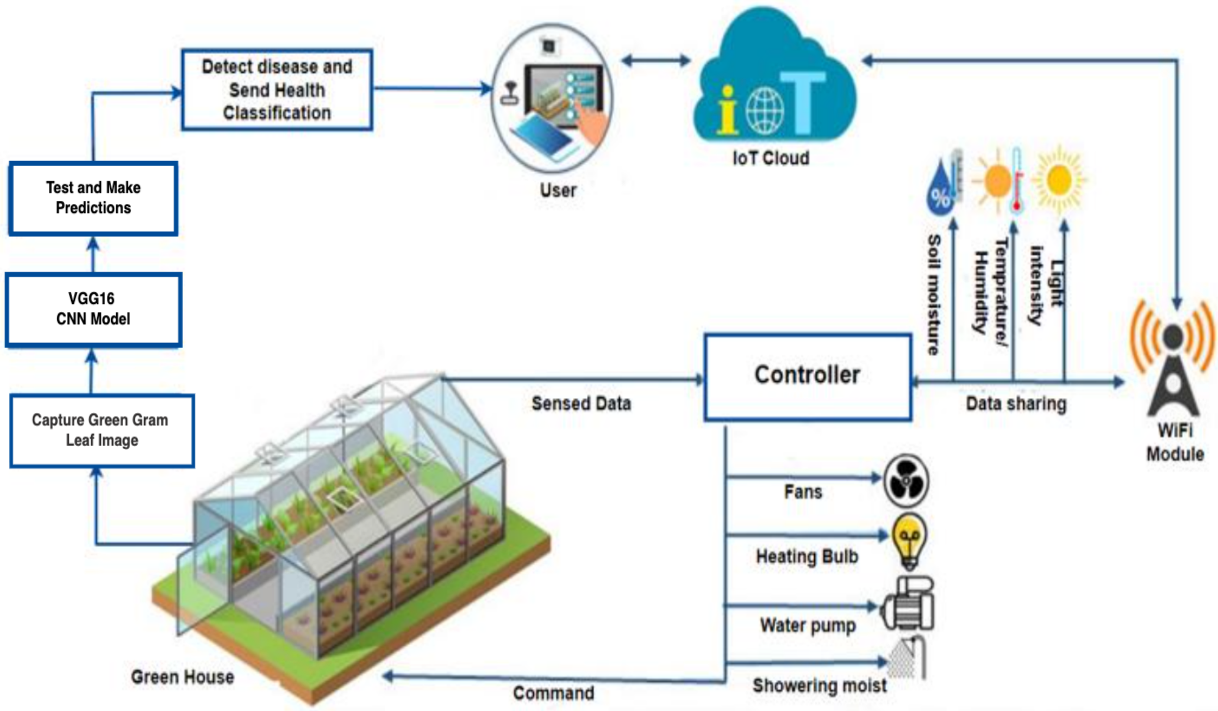

Figure 3.1: Architecture Diagram for Green House Automated System

This smart greenhouse system represents a cutting-edge integration of IoT (Internet of Things), artificial intelligence, and automation to optimize modern agricultural practices. The process begins with the capture of green gram leaf images, which are analyzed using a VGG16 CNN (Convolutional Neural Network) model to detect potential plant diseases. The AI-powered model processes these images to classify the health status of the plants and predict the presence of diseases. These predictions are then shared with the user through an IoT cloud platform, enabling timely decision-making and preventive measures.

Within the greenhouse, a range of sensors continuously monitor vital environmental parameters, including soil moisture, temperature, humidity, and light intensity. These sensors send real-time data to a central controller, which processes the information to automate various systems. For instance, based on the sensed conditions, the controller can activate fans for ventilation, heating bulbs for temperature regulation, water pumps for irrigation, or misting systems to maintain optimal moisture levels. This automation ensures that crops are grown under ideal conditions, reducing resource wastage and minimizing manual intervention.

The IoT cloud serves as a bridge between the greenhouse and the user, providing remote monitoring and control capabilities. This system not only enhances the efficiency of agricultural practices but also supports sustainability by optimizing resource usage and enabling early detection of diseases. By automating critical processes and providing real-time feedback, the smart greenhouse addresses challenges like food security, labor shortages, and environmental sustainability, making it a scalable and impactful solution for modern farming.

Chapter 4

Design Diagrams

4.1 Class Diagram

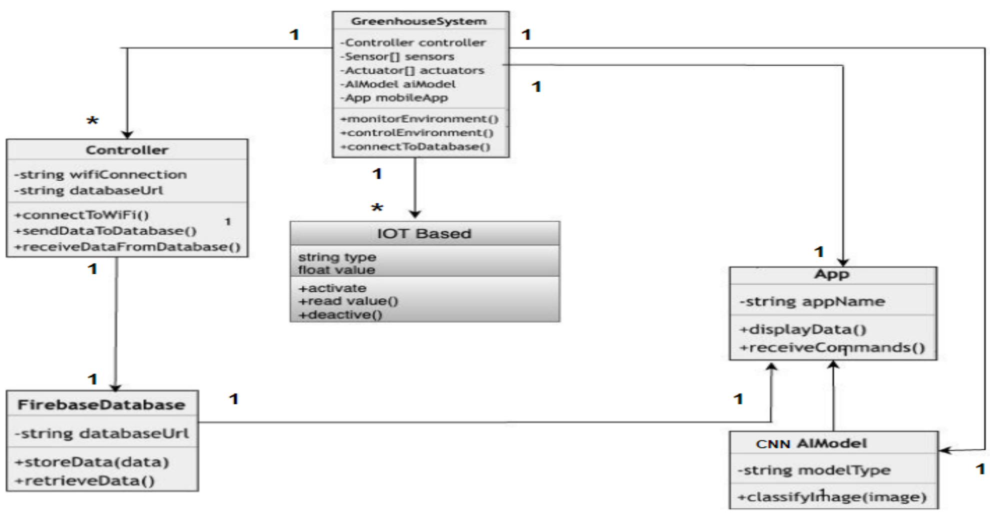

Figure 4.1: Class Diagram for Green House Automated System

Components

1. **GreenhouseSystem**: This class serves as the central hub of the system, managing the various components and their interactions. It has the following attributes:
   - **controller**: This component handles the core functionalities like monitoring the environment, controlling the environment, connecting to the database, and managing the WiFi connection.
   - **Sensor**: These are responsible for gathering data from the greenhouse environment, such as temperature, humidity, light intensity, etc.
   - **Actuators**: These devices act upon the environment based on the Controller's instructions. They might control things like fans, heaters, lights, or irrigation systems.
   - **AI Model:** This component likely uses image classification techniques (CNN) to analyze images of the plants and provide insights for decision-making.
2. **Sensor**: Abstract class for various sensors with methods like `readValue()`. Specific sensors include:
   - **TemperatureSensor**: `readTemperature()`
   - **HumiditySensor**: `readHumidity()`
   - **MoistureSensor**: `readMoisture()`
3. **Controller**: Connects to sensors and actuators, manages data transmission with methods like `connectToWiFi()` and `sendDataToDatabase()`.
4. **Actuator**: Controls devices with `activate()` and `deactivate()`. Examples include:
   - **Fan**: `coolEnvironment()`
   - **SolenoidValve**: `waterPlants()`
   - **LED**: `provideHeat()`, `provideLight()`
5. **FirebaseDatabase**: Handles data storage with `storeData()` and `retrieveData()`.
6. **App**: User interface for displaying data and receiving commands with methods `displayData()` and `receiveCommands()`.
7. **AIModel**: Classifies plant health from images using `classifyImage()`.

4.2 State Diagram

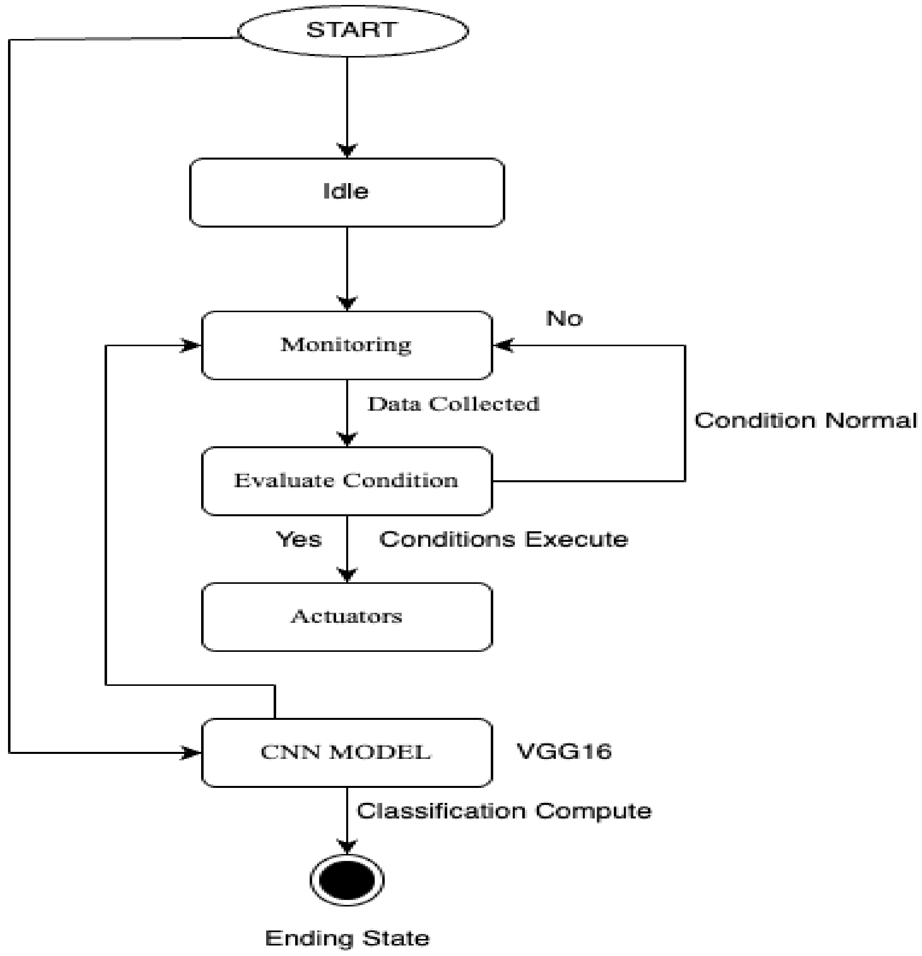

Figure 4.2: State Diagram for Green House Automated System

- **SystemIdle**: The initial state where the system is in standby mode, ready to capture data or retrieve saved results.
- **CapturingData**: The system actively collects image and sensor data needed for analysis.
- **DataCaptured**: Marks the successful collection of data, indicating readiness for the next step.
- **PreprocessingData**: The data is preprocessed to format it appropriately for disease classification.
- **DataPreprocessed**: Confirms that the preprocessing is complete, and the data is ready for the model.

- **ModelClassification**: A deep learning model analyzes the data to detect the presence of disease.
- **ClassificationComplete**: Signals that the classification process has finished, and the results are ready.
- **SavingResults**: The system saves the classification outcome locally for access and future reference.
- **ResultsSaved**: Verifies that the results have been stored successfully in the local database.
- **ViewingResults**: The user accesses and views previously saved classification results.

4.3 Use Case Diagram

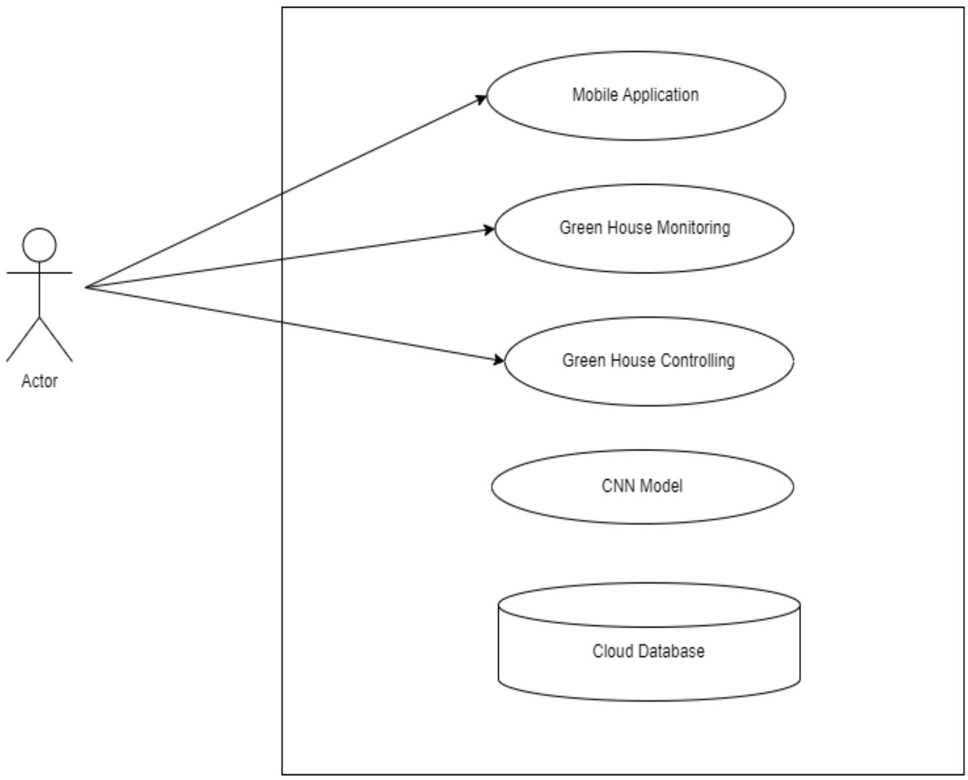

Figure 4.3: Use Case Diagram for Green House Automated System

The diagram illustrates the interactions between an Actor and various Use Cases within a greenhouse monitoring and control system.

- **Actors**

  1. Represents the user who interacts with the system. This could be a farmer, technician, or any other person responsible for managing the greenhouse.
  2. IoT Devices are Sensors and actuators (e.g., cooling fan, LED lights, solenoid valve) installed in the greenhouse to monitor and manage environmental conditions.

- **Use Cases**

  1. **Mobile Application:** This use case indicates that the system provides a mobile application interface through which the user can interact with the greenhouse. This could include features like monitoring sensor data, controlling environmental parameters, and receiving notifications.
  2. **Green House Monitoring:** This use case represents the system's ability to monitor various parameters within the greenhouse, such as temperature, humidity, light intensity, and soil moisture. Data collected from sensors is likely stored and analyzed in the cloud database.
  3. **Green House Controlling:** This use case signifies the system's capability to control different aspects of the greenhouse environment, such as adjusting temperature, humidity, and lighting conditions. This control might be automated based on predefined rules or manual adjustments made through the mobile application.
  4. **CNN Model:** This use case suggests that the system employs a Convolutional Neural Network (CNN) model for image recognition or other data analysis tasks. This could be used for tasks like plant disease detection, crop yield prediction, or automated irrigation control.
  5. **Cloud Database:** This use case represents the cloud-based storage and management of data collected from the greenhouse sensors and other system components. This data could be used for historical analysis, trend identification, and predictive modeling.

- **Overall Functionality:**

  The system, as depicted in the use case diagram, enables users to:

1. **Monitor Greenhouse Conditions:** The system collects data from various sensors and displays it on the mobile application, providing real-time insights into the greenhouse environment.
2. **Control Greenhouse Environment:**Users can adjust parameters like temperature, humidity, and lighting through the mobile application, either manually or by setting automated schedules.
3. **Analyze and Predict:**The CNN model can process sensor data and images to detect potential issues, predict crop yields, or optimize resource usage.
4. **Store and Access Data:**The cloud database stores historical data, allowing for trend analysis, performance evaluation, and informed decision-making.

4.4 Sequence Diagram

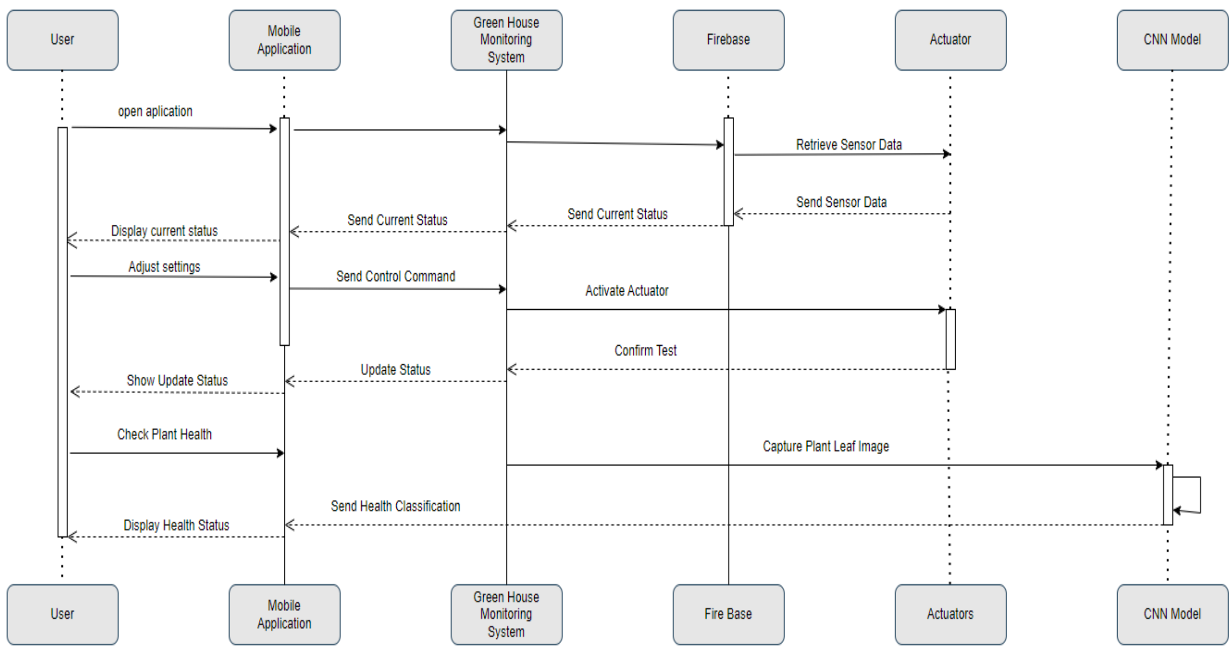

Figure 4.4: Sequence Diagram for Green House Automated System

The sequence diagram illustrates the interactions between different components within a greenhouse monitoring and control system.

1. **Participants:**

- **User:** Represents the user who interacts with the system through a mobile application.

- **Mobile Application:**The software running on the user's mobile device.
- **Greenhouse Monitoring System:**The core system responsible for monitoring and controlling the greenhouse environment.
- **Firebase:**A cloud-based platform used for real-time data exchange and storage.
- **Actuator:** Devices that control physical components in the greenhouse, such as fans, heaters, and lights.
- **CNN Model:**A Convolutional Neural Network used for image analysis and classification tasks.

2. **Interactions:**

- **User Opens Application:**The user initiates the sequence by opening the mobile application.
- **Display Current Status:** The mobile application requests the current status of the greenhouse from the monitoring system.
- **Send Current Status:** The monitoring system sends the current status (temperature, humidity, light intensity, etc.) to the mobile application.
- **Adjust Settings:** The user adjusts settings on the mobile application, such as desired temperature or humidity levels.
- **Send Control Command:** The mobile application sends the updated settings to the monitoring system.
- **Activate Actuator:**The monitoring system sends control commands to the actuators to adjust the greenhouse environment.
- **Show Update Status:** The monitoring system sends an updated status to the mobile application, reflecting the changes made.
- **Check Plant Health:**The user initiates a plant health check.
- **Capture Plant Leaf Image:** The mobile application captures an image of a plant leaf.
- **Send Leaf Image:**The mobile application sends the image to the CNN model for analysis.

- **Send Health Classification:** The CNN model analyzes the image and sends a classification (healthy or unhealthy) to the mobile application.
- **Display Health Status:**The mobile application displays the plant health classification to the user.

3. **Overall Functionality:**

The system, as depicted in the sequence diagram, enables users to:

- **Monitor Greenhouse Conditions:**The user can view real-time data on the mobile application, including temperature, humidity, light intensity, and plant health.
- **Control Greenhouse Environment:** The user can adjust settings for temperature, humidity, and lighting through the mobile application.
- **Perform Plant Health Checks:** The user can capture images of plant leaves and receive automated health assessments from the CNN model.

4.5 Activity Diagram

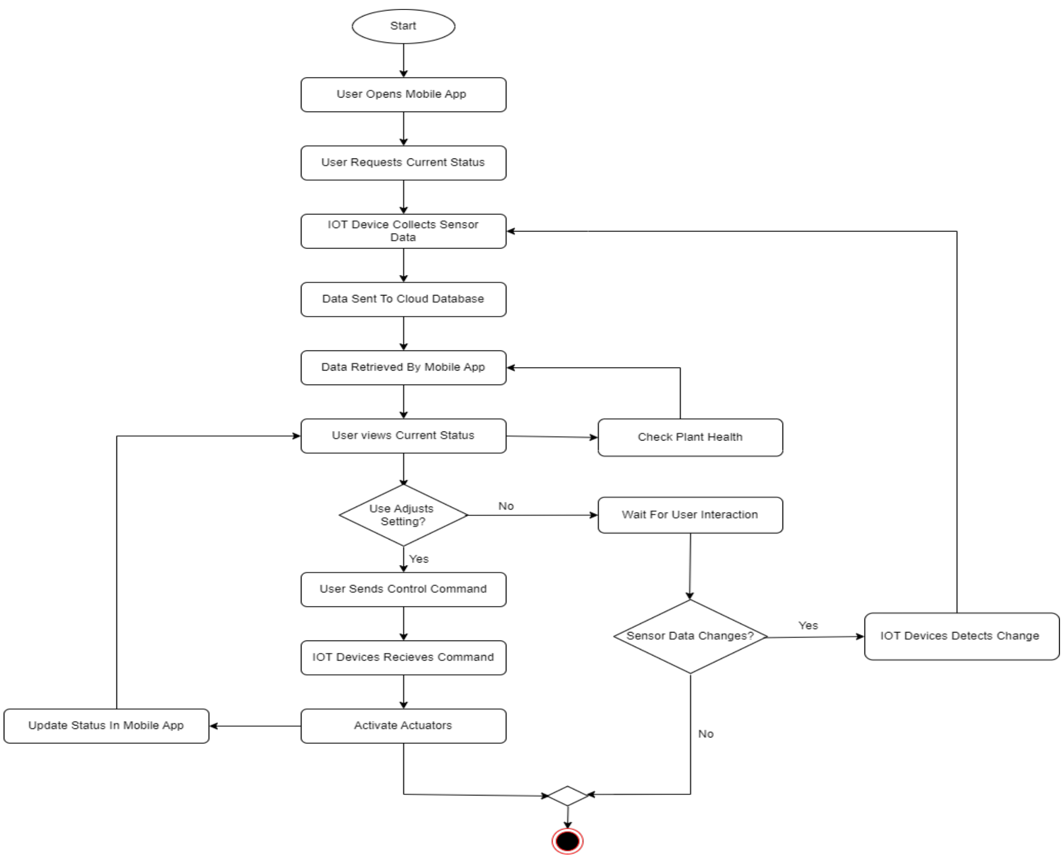

Figure 4.5: Activity Diagram for Green House Automated System

The flowchart depicts the interactions between various components within a greenhouse monitoring and control system.

1. **Start:**
   - The starting point of the system.
2. **User Opens Mobile App:**
   - The user initiates the sequence by opening the mobile application.
3. **User Requests Current Status:**
   - The user requests the current status of the greenhouse environment from the mobile application.
4. **IoT Device Collects Sensor Data:**
   - IoT devices (sensors) deployed in the greenhouse collect data on parameters like temperature, humidity, light intensity, and soil moisture.
5. **Data Sent to Cloud Database:**
   - The collected sensor data is sent to a cloud database for storage and analysis.
6. **Data Retrieved by Mobile App:**
   - The mobile application retrieves the latest data from the cloud database.
7. **User Views Current Status:**
   - The mobile application displays the retrieved data to the user, providing a visual representation of the greenhouse environment.
8. **Check Plant Health:**
   - The user can initiate a plant health check.
9. **User Adjusts Settings?:**
   - The user can adjust settings for temperature, humidity, and lighting through the mobile application.
10. **Wait for User Interaction:**

- If the user does not adjust settings, the system waits for further user input.

11. **User Sends Control Command:**
    - If the user adjusts settings, the mobile application sends a control command to the monitoring system.
12. **IoT Devices Receive Command:**
    - The monitoring system receives the control command and sends it to the appropriate IoT devices.
13. **Activate Actuators:**
    - The IoT devices (actuators) receive the control command and adjust the greenhouse environment accordingly (e.g., turning on/off lights, adjusting fans, etc.).
14. **Sensor Data Changes?:**
    - The system continuously monitors sensor data for changes.
15. **IoT Devices Detect Change:**
    - If a significant change in sensor data is detected, the system triggers an alert or takes corrective actions.
16. **Update Status in Mobile App:**
    - The mobile application displays updated status information to the user, reflecting any changes in the greenhouse environment.

Overall Functionality:

The system, as depicted in the flowchart, enables users to:

- **Monitor Greenhouse Conditions:** The user can view real-time data on the mobile application, including temperature, humidity, light intensity, and plant health.
- **Control Greenhouse Environment:**The user can adjust settings for temperature, humidity, and lighting through the mobile application.
- **Receive Alerts:**The system can send alerts to the user if any critical parameters exceed predefined thresholds.

4.6 Data Flow Diagram

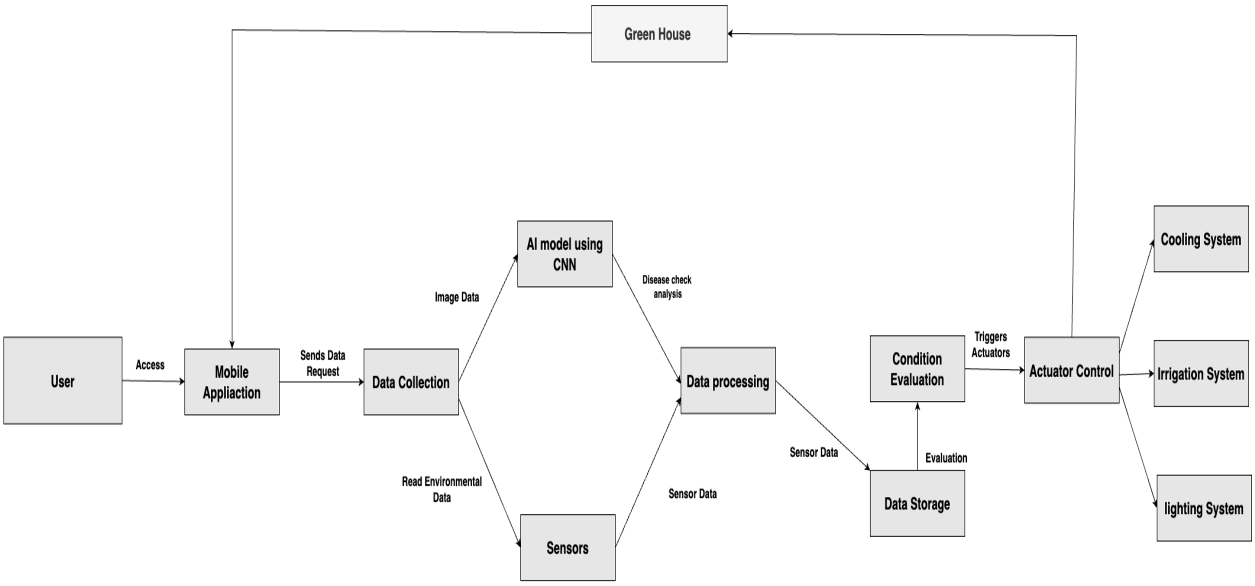

Figure 4.6: Data Flow Diagram for Green House Automated System

The diagram depicts a greenhouse monitoring and control system that utilizes an AI model for plant disease detection and environmental control. Here's a breakdown of the components and their interactions:

- **User:**: Represents the user who interacts with the system through a mobile application.
- **Mobile Application:**: The software running on the user's mobile device, providing a user interface to interact with the system.
- **Data Collection:** This component encompasses various sensors deployed in the greenhouse to collect data on environmental parameters (temperature, humidity, light intensity, etc.) and plant images.
- **Sensors:**: These are the physical devices that gather data from the greenhouse environment.
- **Data Processing:**This component handles the processing of collected data, including image processing and data analysis.
- **Data Storage:**: The processed data is stored in a database for analysis and historical reference.
- **Condition Evaluation:**This component analyzes the collected data and compares it against predefined thresholds or models to determine if any conditions need to be adjusted.

- **Actuators:** These are devices that control physical components in the greenhouse, such as fans, heaters, lights, and irrigation systems.
- **AI Model (CNN):**A Convolutional Neural Network (CNN) is used to analyze plant images and detect diseases.
- **Cooling System:**A system that regulates the temperature within the greenhouse.
- **Irrigation System:** A system that controls the watering of plants.
- **Lighting System:** A system that controls the lighting conditions within the greenhouse.

Flow of Operation:

- **User Interaction:** The user interacts with the mobile application to monitor the greenhouse environment and control settings.
- **Data Collection:** Sensors continuously collect data on environmental parameters and plant images.
- **Data Processing:**Collected data is processed and analyzed. Plant images are sent to the CNN model for disease detection.
- **Condition Evaluation:**The system evaluates the environmental conditions and plant health status based on the processed data.
- **Actuator Control:**If necessary, the system sends commands to the actuators to adjust temperature, humidity, lighting, and irrigation.
- **Data Storage:** Processed data is stored in the database for future analysis and historical reference.
- **User Feedback:** The system provides feedback to the user through the mobile application, including real-time data, alerts, and recommendations.

4.7 Modular Diagram

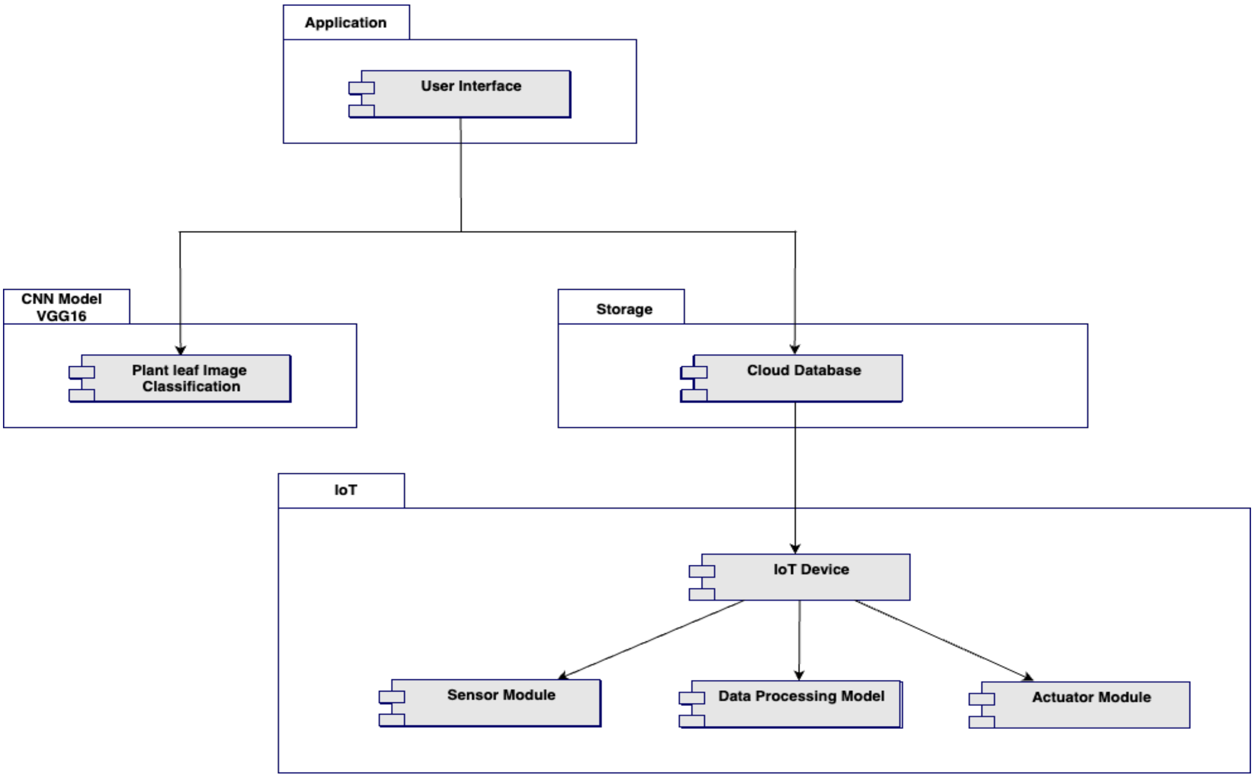

Figure 4.7: Modular Diagram for Green House Automated System

This diagram appears to represent a modular system integrating various components for plant leaf classification and IoT functionalities. Below is an explanation of each module and its possible information for a report:

1. **Application:**

   - Role:

     - The UI serves as the point of interaction for users to input data or retrieve results.
     - Features provided by the UI (e.g., upload images, visualize classification results, control IoT devices).
     - User experience (UX) considerations.

2. **CNN Model:**

   - VGG16 (Convolutional Neural Network):

     - A pretrained deep learning model used for plant leaf image classification.

- Plant Leaf Image Classification:
  - Functionality: Processes uploaded plant leaf images to classify them (e.g., identifying plant species or detecting diseases).
  - Details about the CNN architecture (VGG16 specifics: layers, weights, etc.).
  - Training dataset (e.g., leaf dataset used for training and testing).
  - Accuracy and performance metrics (e.g., precision, recall, F1-score).

3. **Storage:**

- Cloud Database:
  - Centralized storage for data generated and used by the system, such as images, classification results, and IoT sensor data.
  - Database type (e.g., SQL or NoSQL) and structure.
  - Data management strategies (e.g., data security, scalability, and backup).
  - Integration with other modules (e.g., IoT devices and the CNN model).

4. **IoT:**

- IoT Device:
  - The central hardware for IoT data collection, processing, and actuation.
  - Hardware specifications (e.g., microcontroller, connectivity modules).
  - Software running on the IoT device (e.g., firmware details).
- Sensor Module:
  - Collects environmental data (e.g., temperature, humidity, or soil moisture).
  - Types of sensors used and their capabilities.
  - Methods for data collection and transmission to the IoT device.
- Data Processing Model:
  - Processes sensor data locally on the IoT device or sends it to the cloud for analysis.
  - Algorithms or techniques used for data processing.
  - Decision-making strategies based on processed data (e.g., thresholds).

- Actuator Module:
  - Executes actions based on IoT data (e.g., watering plants or controlling environmental conditions).
  - Types of actuators used (e.g., motors)
  - Control mechanisms (e.g., relay systems, direct software commands).

4.8 ER Diagram

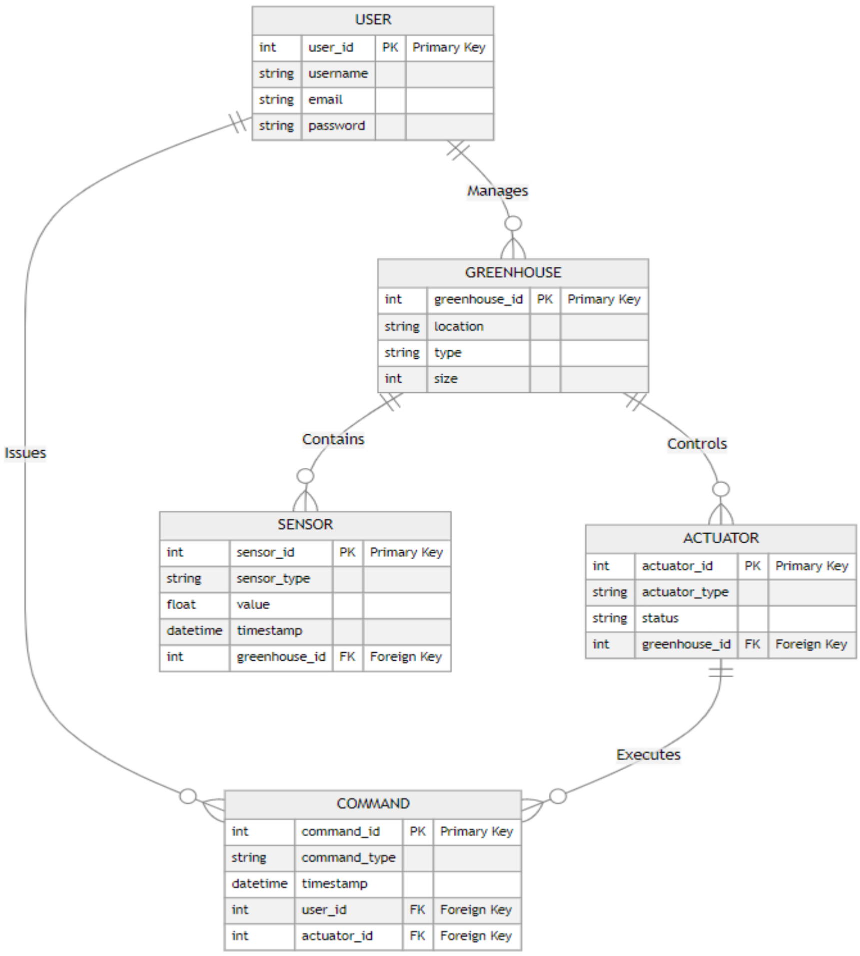

Figure 4.8: ER Diagram for Green House Automated System

1. **User :**

- **Attributes:**
  - **user_id:** A unique identifier for the user (Primary Key).
  - **username:** The user's login name or identification.
  - **email:** The user's contact email address.

- **Relationships:**
  - **Manages:** A user can manage one or more greenhouses.
- **Information about user:**
  - Role of the user in the system.
  - Authentication and authorization details.
  - User data privacy and security measures.

2. **GREENHOUSE:**

- **Attributes:**
  - **greenhouse_id:** A unique identifier for the greenhouse (Primary Key).
  - **location:** Describes the physical location of the greenhouse.
  - **type:** Specifies the type of greenhouse (e.g., hydroponic, conventional).
  - **size:**Indicates the size of the greenhouse (e.g., square meters).
- **Relationships:**
  - **Managed by USER:** Each greenhouse is managed by a user.
  - **Contains SENSOR:** A greenhouse contains one or more sensors
  - **Controls ACTUATOR:** Actuators are installed to control greenhouse conditions.
- **Information about GREENHOUSE:**
  - Description of greenhouse types and management responsibilities.
  - Integration of IoT devices (sensors and actuators) in the greenhouse environment.

3. **SENSOR:**

- **Attributes:**
  - **sensor_id:** A unique identifier for the sensor (Primary Key).
  - **sensor_type:** The type of sensor (e.g., temperature, humidity, soil moisture).
  - **value:** The measured value captured by the sensor.
  - **timestamp:** The date and time of the recorded sensor value.

    - **greenhouse_id:** Foreign Key linking the sensor to a specific greenhouse.

4. **Relationship:**
    - **Belongs to GREENHOUSE:**
        - **Belongs to GREENHOUSE:** A sensor is linked to a specific greenhouse.
    - **Information of Sensor:**
        - List of sensor types and their roles (e.g., monitoring environmental conditions).
        - Data processing methods for sensor readings (e.g., data storage, threshold
5. **ACTUATOR:**
    - **Attributes:**
        - **actuator_id:** A unique identifier for the greenhouse (Primary Key).
        - **actuator_type:** The type of actuator (e.g., irrigation system, ventilation control).
        - **status:** The current state of the actuator (e.g., active, inactive).
        - **greenhouse_id:** Foreign Key linking the actuator to a specific greenhouse.
    - **Relationships:**
        - **Controlled by GREENHOUSE:** An actuator is associated with a greenhouse.
    - **Information of Actuator:**
        - Types of actuators and their functionality (e.g., watering plants, adjusting temperature).
        - Actuation mechanisms and how commands are executed.
6. **COMMAND:**
    - **Attributes:**
        - **command_id:** A unique identifier for the command (Primary Key).

  - **command_type:** The type of command issued (e.g., "Activate irrigation").
  - **timestamp:** The date and time the command was issued.
  - **user_id:** Foreign Key linking the command to the user who issued it.
  - **actuator_id:** Foreign Key linking the command to the specific actuator.

- **Relationships:**
  - **Issued by USER:** Commands are created by users.
  - **Executed by ACTUATOR:** Commands are sent to actuators to perform actions.

- **Information of Command:**
  - Types of commands that can be issued and their effects.
  - The workflow from command creation to execution.
  - Log and tracking of command history for auditing.