Software Design Specifications

[AppraiseChain]

Version: [xx.xx]

Project Code	F24-167
Supervisor	Muhammad Ali Shah Fatmi
Co Supervisor	-
Project Team	Taha Ali (21K-3867) Imran Ali (21K-3877) Taha Jawaid (21K-3881)
Submission Date	

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

Version	Name of Person	Date	Description of change
			[e.g. Document Created]
			[Added Non-functional requirements]
			[Added UseCase x.x.xx]

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

Name	Role
Muhammad Ali Shah Fatmi	Supervisor
-	Co Supervisor

Document Sign-Off

[Following table will contain sign-off details of the document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.
Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

Version	Sign-off Authority	Project Role	Signature	Sign-off Date

Document Information

Category	Information
Customer	FAST-NU
Project	Plantree AI
Document	Software Design Specification
Document Version	1.0
Status	Draft
Author(s)	Taha Ali Imran Ali Taha Jawaid
Approver(s)	Muhammad Ali Shah Fatmi
Issue Date	
Document Location	
Distribution	Advisor Project Coordinator’s Office (through Advisor)

Definition of Terms, Acronyms and Abbreviations

Term	Description
AI	Artificial Intelligence

Term	Description
IoT	Internet of Thing
ESP32	A microcontroller with integrated Wi-Fi and Bluetooth capabilities
DHT11	A digital sensor for measuring temperature and humidity
MQ135	A digital sensor to observe Air Quality of the specific location
HOD	Head of Department

Table of Contents

1 Introduction

1.1 Purpose of Document

The intention behind preparing this Software Design Specification (SDS) document is to develop an abstract and technical outline of the “Plantree”, an AI incorporated regional tree plantation predictor. Structural: It specifies the structural elements and their relationships in order to achieve functional and non-functional as well as technical necessities. This document is aimed at assisting the software architects, developers and testers realize systems implementations. The objectives of this document are:

- **To establish the system's architectural foundation:** Preliminary identification of major constructs and the way they are related on a conceptual level.
- **To describe key modules and workflows:** Informing the technical team of system features about how the features should be implemented.
- **To ensure design alignment with requirements:** Confirming that the proposed design is functional as well as friendly with the environment in accordance to the general goals and objectives of the project.
- **To enable traceability and scalability:** Both for meeting immediate needs of the current system and for future growth and maintenance.

1.2 Intended Audience

- Fast NU
- Jury
- Supervisor (Mr. Muhammad Ali Shah Fatmi)
- Students of Fast NU
- Our Team (Designer, Developer, Tester)
- Potential Users of this product

1.3 Document Convention

- Font Family = Arial

- Font Size = 18 for headings, 16 for sub-headings 14 for the rest of the content

1.4 Project Overview

The **"Plantree" AI-enhanced regional tree plantation predictor** is designed to address climate change challenges by optimizing tree plantation. The system uses IoT sensors and AI-driven algorithms to analyze environmental data and recommend plantation zones that yield maximum ecological benefits. The design integrates AI for predictions and analytics while providing an intuitive interface for organizations to interact with the system.

Key Design Goals

- **Optimization of tree plantation zones:** Identifying regions that minimize the impact of climate change through strategic tree planting.
- **Integration of IoT sensors and AI models:** Leveraging data-driven insights for informed decision-making.
- **Impact assessment and reporting:** Predicting and visualizing the climate benefits of suggested actions.
- **User-centric design:** Providing a seamless interface for stakeholders to interact with the system.

Included Functionalities:

- **IoT Sensor Data Collection:** Integration with MQ135, ESP32, and DHT11 to collect environmental data (e.g., air quality, temperature, humidity).
- **AI Prediction Engine:** Utilizing AI models like SvM, Random Forest, Neural Network and 1D CNN to analyze data and recommend plantation regions.
- **Impact Prediction Module:** Assessing the projected climate benefits of tree plantation activities.
- **Data Storage and Management:** Secure and efficient storage for collected data and prediction results.
- **Visualization and Reporting:** Providing interactive dashboards to represent data insights and predictions.

Excluded Functionalities:

- Manual verification of plantation zones by users or organizations.
- Predictive modeling for non-environmental factors such as economic or social parameters.

1.5 Scope

The design focuses on implementing a robust, scalable, and user-friendly system to enhance tree plantation efficiency. The scope includes:

Included Functionalities:

Environmental Data Collection:

- Collecting temperature, humidity, and air quality data using IoT sensors.

AI-Powered Predictions:

- Recommending optimal plantation zones.
- Predicting the ecological impact of plantation activities.

User-Friendly Interface:

- Providing stakeholders with access to reports and predictions.

Data Security and Management:

- Ensuring the secure storage of sensitive environmental and user data.

Visualization and Reports:

- Representing prediction results in graphical formats for better understanding.

Excluded Functionalities:

- Predictions beyond environmental impacts.
- Real-time alerts for environmental disasters or emergencies.

2 Design Considerations

This section outlines the primary considerations, assumptions, dependencies, and risks that influence the design of the **"Plantree"** AI system. Addressing these issues ensures a robust, scalable, and adaptable system architecture.

2.1 Assumptions and Dependencies

Assumptions:

- **IoT Sensor Reliability:** The IoT sensors (MQ135, ESP32, and DHT11) are assumed to provide accurate and consistent data for environmental monitoring.
- **AI Model Pre-training:** Pre-trained AI models (SvM, Random Forest, Neural Network and 1D CNN) are assumed to be available and can be fine-tuned for specific requirements without extensive training from scratch.
- **User Expertise:** It is assumed that end-users (e.g., environmental organizations) have basic technical proficiency to interact with the system's interface and understand recommendations.
- **Internet Connectivity:** The system is assumed to operate in regions with stable internet connectivity for data collection, processing, and visualization.
- **Geographical Data Availability:** Geographic and environmental datasets required for model validation and comparison are accessible through APIs or third-party sources.

Dependencies:

- **Third-Party APIs:** Dependence on external APIs for data validation and additional environmental insights (e.g., weather APIs, governmental data repositories).
- **Cloud Infrastructure:** The system depends on cloud-based storage and computation services to ensure scalability and secure data handling.
- **Sensor Hardware:** The performance of the system is dependent on the correct functioning and calibration of the IoT hardware.
- **AI Frameworks:** Dependence on AI libraries such as TensorFlow or PyTorch for model implementation.

2.2 Risks and Volatile Areas

Risks:

- **Sensor Failures:** IoT hardware malfunctions can lead to data inaccuracies, requiring fallback mechanisms for data validation and error correction.
- **Model Accuracy:** AI prediction models may produce unreliable recommendations if the training data lacks diversity or quality. Mitigating this risk requires continuous model refinement and revalidation.
- **Data Security:** The system deals with sensitive environmental and user data. A breach in the storage system could lead to misuse or loss of trust. Implementing strong encryption and secure access protocols is critical.
- **API Dependency:** Relying on third-party APIs for critical data introduces the risk of downtime or unexpected changes in API structures. A contingency plan for alternate data sources is necessary.
- **Scalability Challenges:** High system load during large-scale data processing or visualization tasks could impact system performance. Proper load-balancing techniques must be implemented.

Volatile Areas:

- **Technology Evolution:** Rapid advancements in AI models and IoT technology could make existing components obsolete. The system design must allow for modular updates without affecting core functionality.
- **Environmental Policy Changes:** Shifts in environmental regulations or plantation priorities could require redesigning the recommendation logic to comply with new rules.
- **New Functional Requirements:** Stakeholders may request additional features such as economic impact analysis or biodiversity evaluation, which could necessitate significant architectural changes.
- **Data Sources:** Any changes to the availability or structure of external data sources (e.g., APIs or governmental datasets) could require modifications in data-fetching and validation modules.

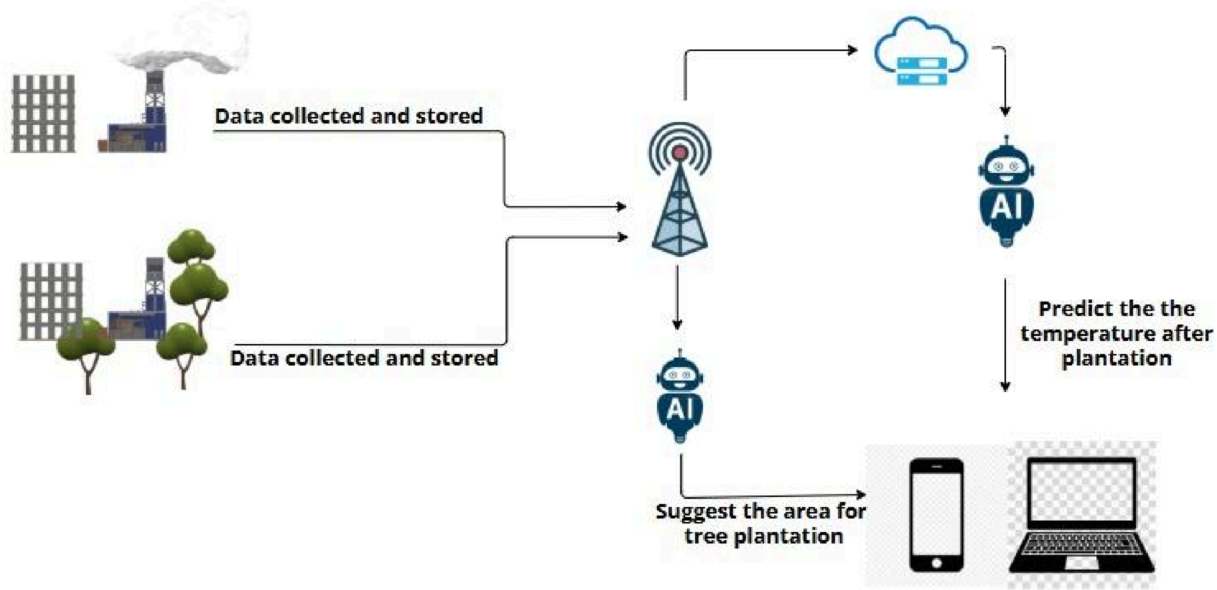
3 System Architecture

This section outlines the high-level system architecture for the **"Plantree"** AI project. It provides a breakdown of the system's components, their relationships, and their interactions to achieve the desired functionalities.

3.1 System Level Architecture

The **"Plantree"** system is divided into several key subsystems and components to facilitate modular design, scalability, and ease of maintenance.

1. **System Decomposition into Elements:**
   - **IoT Sensor Integration Module:**
     Collects environmental data (air quality, humidity, temperature) from sensors such as MQ135, DHT11, and ESP32.
   - **AI Prediction Engine:**
     Processes data using pre-trained AI models (SvM, Random Forest, Neural Network and 1D CNN) to recommend optimal tree plantation regions.
   - **Impact Prediction Module:**
     Analyzes and predicts the environmental impact of tree plantations based on collected data and AI outputs.
   - **Data Storage and Management System:**
     Stores sensor data, AI predictions, and user inputs securely using a relational or NoSQL database.
   - **User Interface (UI):**
     A front-end interface for interacting with the system, including dashboards for data visualization, recommendations, and reports.
   - **Reporting and Visualization Module:**
     Generates visual and textual representations of recommendations, predicted impacts, and data trends for users.



3.2 Software Architecture

The software architecture is designed using a layered approach, ensuring separation of concerns and modular interaction.

1. **User Interface Layer:**
   - Provides a user-friendly interface for accessing the system.
   - Includes features for data input, recommendations display, and report generation.
   - Technologies: React.js or Angular (frontend), RESTful API for backend communication.
2. **Middle Tier:**
   - Contains the business logic for processing data and making predictions.
   - Modules include:
     - **AI Model Handler:** Processes incoming data using AI models.
     - **Impact Analyzer:** Predicts environmental benefits of plantation.
     - **Error Management:** Handles exceptions and errors in the system.
   - Technologies: Python-based AI frameworks (e.g., TensorFlow, PyTorch), Flask/Django for APIs.
3. **Data Access Layer:**

- Interacts with the database to store and retrieve data.
- Supports efficient querying and secure access to data.
- Technologies: SQL databases (e.g., PostgreSQL) or NoSQL (e.g., MongoDB).

4 Design Strategy

The design strategy for the **"Plantree"** system is based on modularity, scalability, and ease of future enhancements. This section discusses the strategic decisions and trade-offs made during the design of the system's high-level structure, focusing on extensibility, reusability, and efficient operation.

4.1 Future System Extension or Enhancement

The system has been designed with scalability and future enhancements in mind:

1. **Modular Design:**
    - Each subsystem (e.g., IoT sensor integration, AI engine, data management) operates as an independent module.
    - New sensors, AI models, or analysis modules can be integrated without altering the core architecture.
2. **API-Based Communication:**
    - External APIs (e.g., environmental or geographic data sources) have been abstracted for easy replacement or addition of new APIs.
    - RESTful APIs ensure a standardized interface for future integrations.
3. **Cloud-Based Infrastructure:**
    - Cloud-based deployment facilitates scaling up for higher workloads or new computationally intensive models.
4. **AI Model Flexibility:**
    - The architecture supports the addition of new AI models (e.g., newer versions of SvM, Random Forest, Neural Network and 1D CNNr) as technology evolves.

4.2 System Reuse

The design ensures that various components of the system can be reused across similar projects or contexts:

1. **Reusable Modules:**
    - The IoT sensor integration module can be adapted for other environmental monitoring projects.
    - The AI engine is generic enough to be applied to other predictive tasks, such as agricultural optimization or climate forecasting.
2. **Standardized Data Processing Pipeline:**
    - The data collection, preprocessing, and analysis pipeline can be reused across systems requiring sensor data integration and AI-driven insights.
3. **Modular UI Design:**
    - The user interface is built with reusable components (e.g., charts, data input forms) that can be repurposed for other systems.

4.3 User Interface Paradigms

The user interface has been designed with usability and responsiveness as primary goals:

Web-Based Interface:

- Built with responsive web design to ensure accessibility across devices (e.g., desktops, tablets, smartphones).
- Technologies like React.js or Angular allow for a modern, intuitive, and visually appealing interface.

User Roles and Customization:

- Separate views for different user roles (e.g., administrators, HODs, employees) ensure that each user gets a tailored experience.
- Dashboards with interactive charts and maps make insights easily digestible.

Localization Support:

- The system is designed to support multiple languages for wider accessibility.

4.4 Data Management (Storage, Distribution, Persistence)

Data Storage:

The system uses a hybrid database strategy:

- **Relational Database (e.g., PostgreSQL):** For structured data such as user accounts, recommendations, and reports.
- **NoSQL Database (e.g., MongoDB):** For unstructured or semi-structured sensor data.

Data Distribution:

- Data collected from IoT sensors is transmitted securely to the cloud via MQTT or HTTP protocols.
- API endpoints allow external systems to fetch or send data.

Data Persistence and Backup:

- Regular data backups are scheduled to prevent data loss.
- Historical data is retained for longitudinal analysis and trend forecasting.

Data Security:

- End-to-end encryption ensures secure data transmission.
- Role-based access control (RBAC) limits unauthorized access to sensitive data.

4.5 Concurrency and Synchronization

IoT Sensor Data Collection:

- Sensors operate in parallel to collect real-time data.
- A message queue (e.g., RabbitMQ) ensures proper synchronization of incoming data streams.

AI Model Execution:

- Concurrent processing of AI models is supported to handle multiple data inputs simultaneously.
- Task scheduling ensures that resources are optimally allocated for high-priority tasks.

User Requests and Server Load Balancing:

- The system uses load balancing to handle multiple user requests concurrently.
- Sessions are managed to ensure smooth performance under high traffic conditions.

Synchronization:

- Consistency in data updates is maintained through database transaction mechanisms.
- Timestamps are used to ensure proper sequencing of sensor data and analysis results.

4.6 Trade-Offs

Modularity vs. Performance:

Modular design may introduce slight performance overhead due to inter-module communication but ensures long-term maintainability.

Cloud Dependency:

Reliance on cloud infrastructure improves scalability but requires consideration of operational costs and internet dependency.

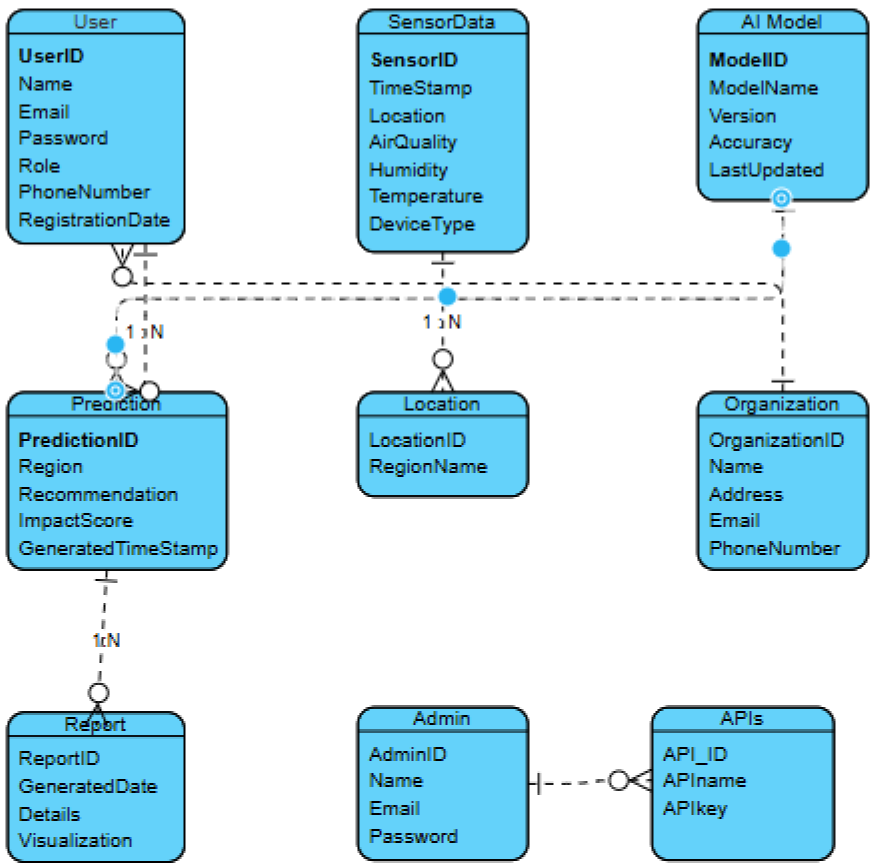
Reusability vs. Customization:

While generic modules are reusable, certain functionalities may require customization for specific project requirements.

5 Detailed System Design

5.1 Database Design

5.1.1 ER Diagram



5.1.2 Data Dictionary

5.1.2.1 Data 1

Activity	
Name	Activity
Alias	Activity

Where-used/how-used	When an employee logs their activity, it will be automatically tracked. The time spent working and time spent inactive will be stored in the database.					
Content description	This data table is used to track and store productive statistics for employees involved in tree plantation monitoring activities.					
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
id	unique identifier	Varchar	15	NO	None	PK
emp_id	Employee's id	Int	11	NO	None	Fk
active_item	Time spent working	Int	15	NO	None	
passive_item	Time spent working	Varchar	15	NO	None	
date	Date	date	8	NO	None	

5.1.2.2 Data 2

Performance						
Name	Performance					
Alias	AI Model Evaluation					
Where-used/how-used	This table stores performance statistics for the AI model used in predicting optimal tree plantation locations and their respective impact.					
Content description	The table contains data on AI predictions, model accuracy, and suggested locations to measure how well the system is performing.					
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
id	unique identifier	Varchar	15	NO	None	PK
model_name	Name Of AI Model	Varchar	15	NO	None	

accuracy	Accuracy of the AI predictions	Decimal	256	5,2	None	
predicted_area	Suggested plantation area	Varchar	256	NO	None	
impact_score	Predicted environmental impact	Varchar	256	YES	None	
date	Date	date	8	NO	None	

5.1.2.3 Data 3

Sensor Data	
Name	Sensor Data
Alias	IoT Readings
Where-used/how-used	This table stores real-time data captured by IoT sensors like MQ135, DHT11, and ESP32. The data is used to analyze temperature, humidity, and air quality for optimal tree plantation predictions.
Content description	The data table records environmental parameters that form the basis for AI model predictions regarding tree plantation locations.

Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
id	unique identifier	Int	11	NO	None	PK
temperature	Temperature reading from DHT11 sensor	Varchar	15	NO	None	
humidity	Humidity reading from DHT11 sensor	Varchar	1	YES	NULL	
air_quality	Air quality reading from MQ135 sensor	Varchar	8	NO	None	
location	Location details	Varchar	10	NO	None	
Time_stamp	Timestamp	DateTime	19	NO	None	

5.1.2.4 Data 4

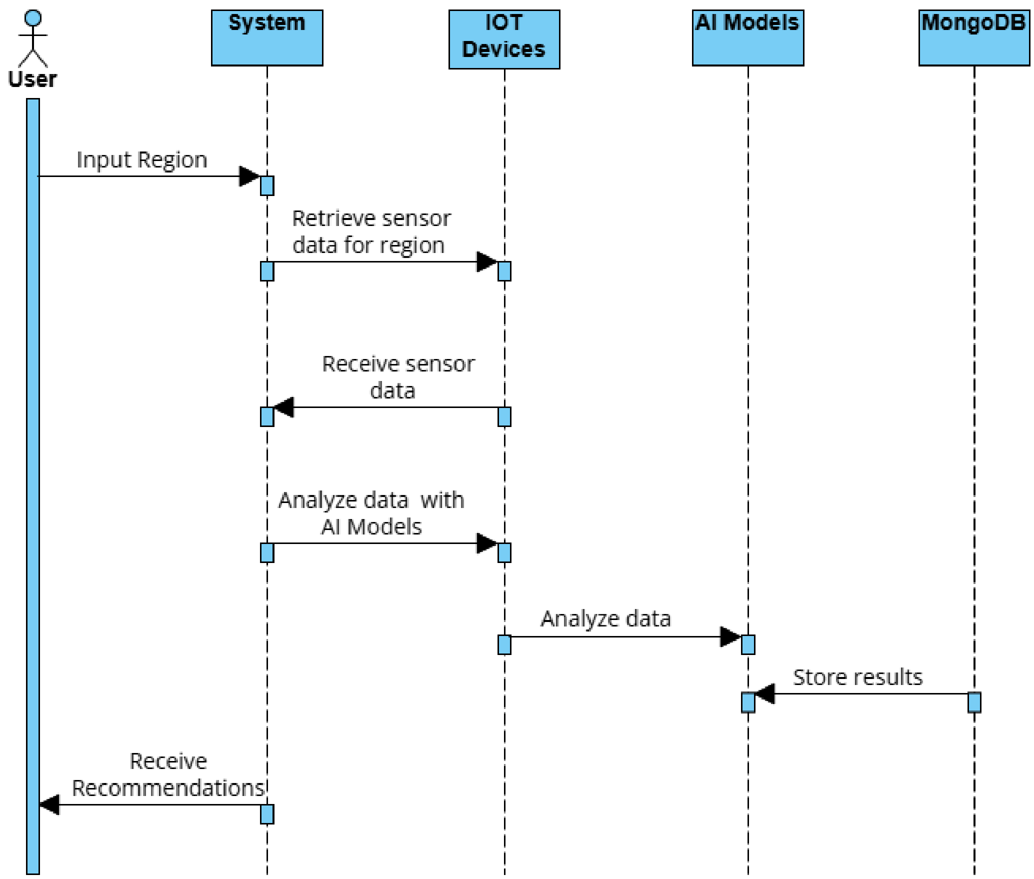
Recommendation	
Name	Recommendation
Alias	Tree Plantation Suggestion
Where-used/how-used	Stores AI-generated recommendations for optimal tree plantation locations based on collected environmental data.
Content description	The table highlights areas best suited for plantation along with potential environmental benefits derived from the action.

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
id	unique identifier	Int	11	NO	None	PK
remmended_area	Area suggested for plantation	Varchar	15	NO	None	
tree_type	Type of tree recommended	Varchar	15	NO	None	
impact_score	Predicted impact of the recommendation	Varchar	15	NO	None	
timestamp	Time	DateTime	8	NO	None	

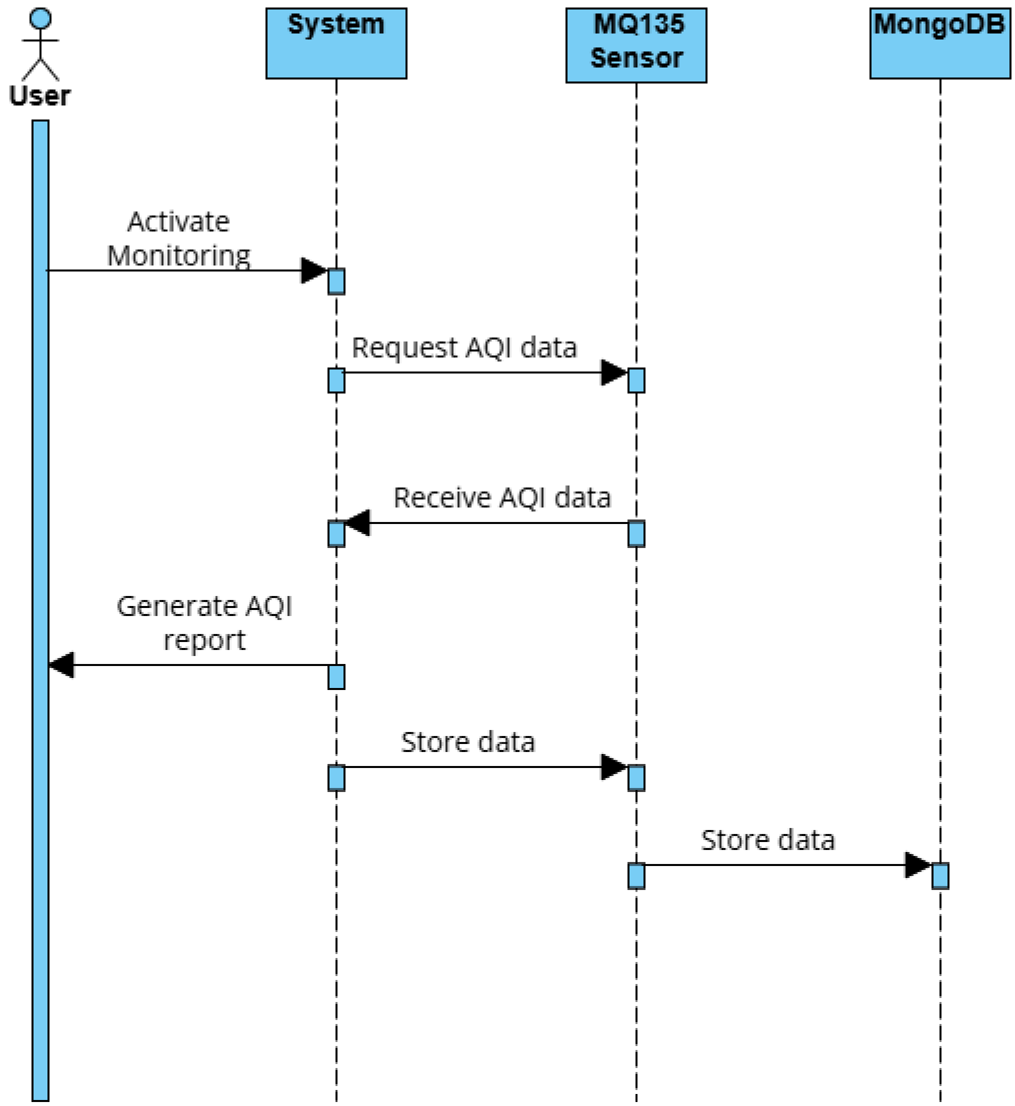
5.2 Application Design

5.2.1 Sequence Diagram

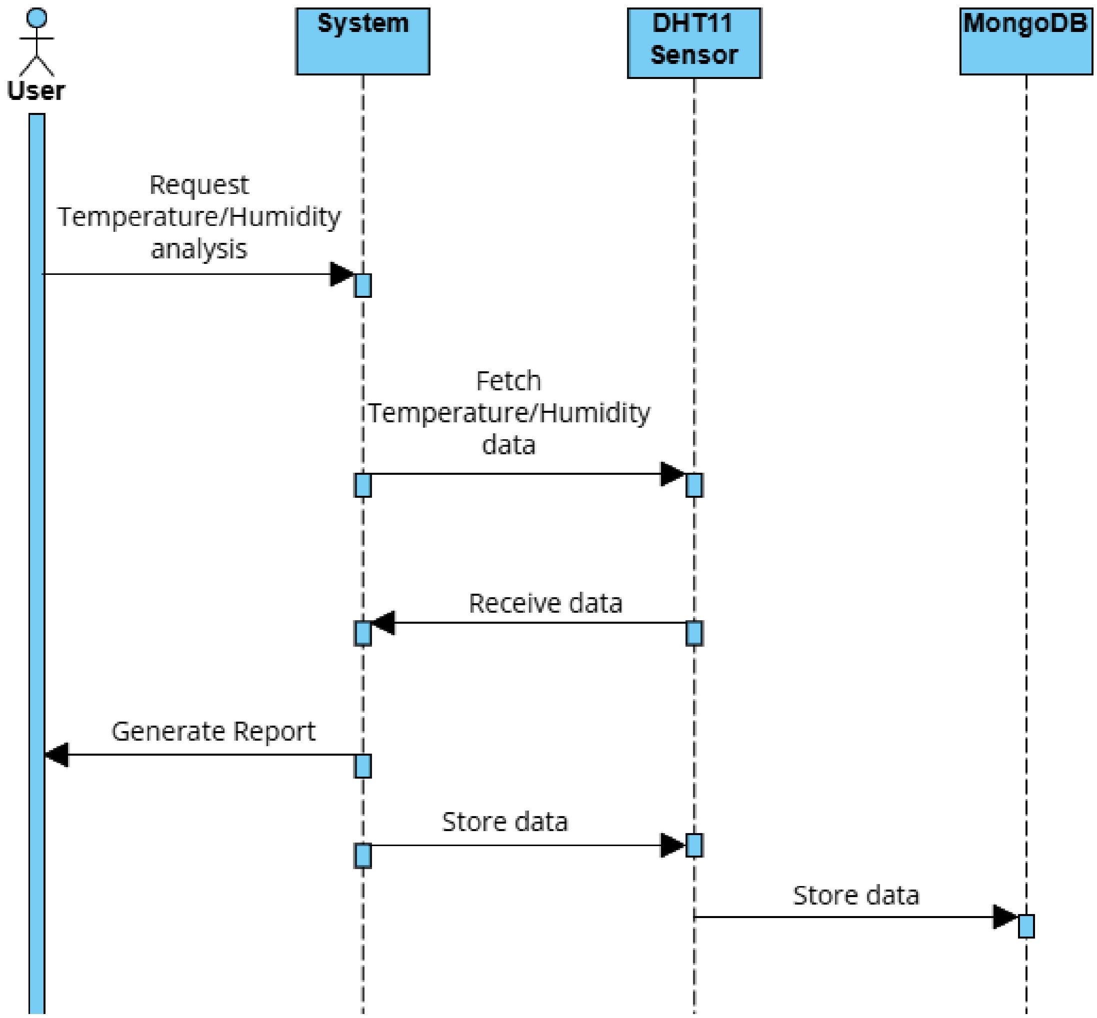
5.2.1.1 Data Fetching



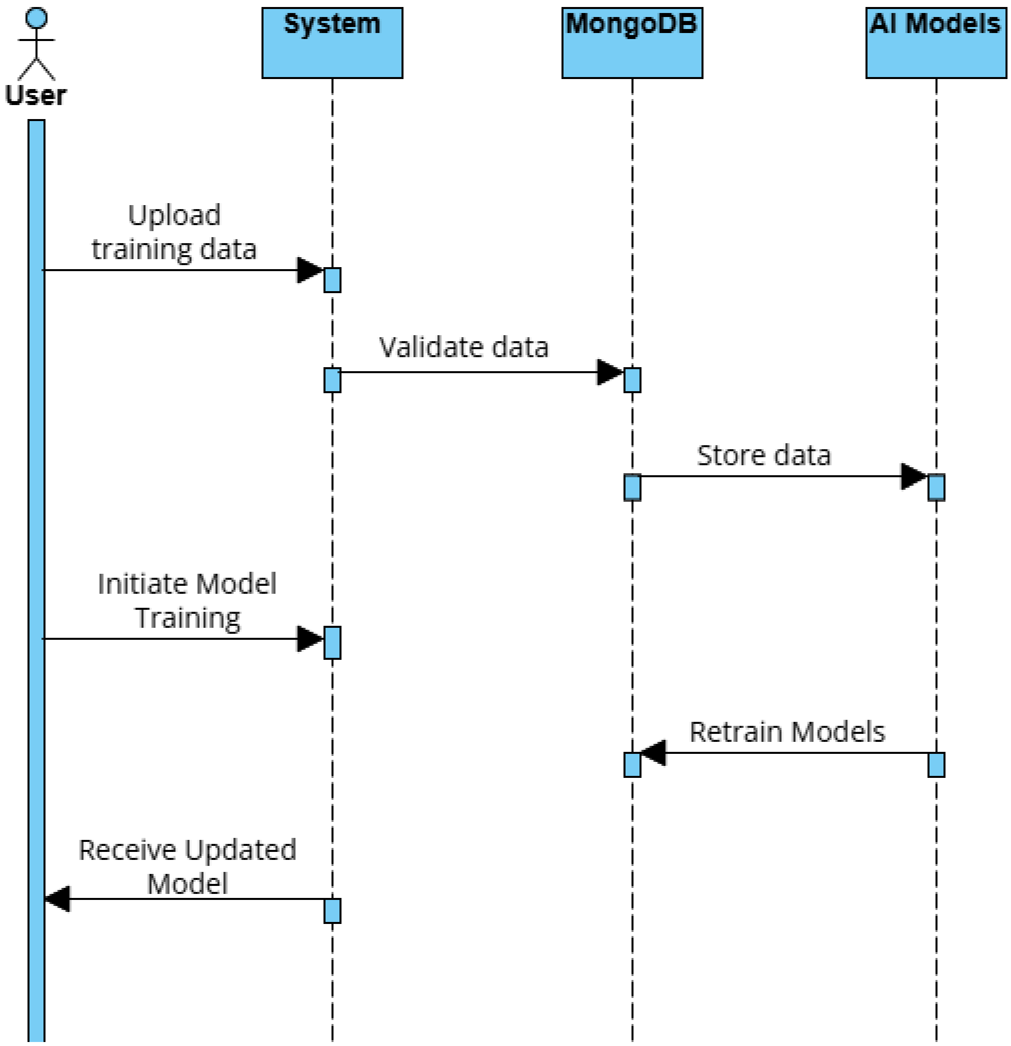
5.2.1.2 Monitor Air Quality



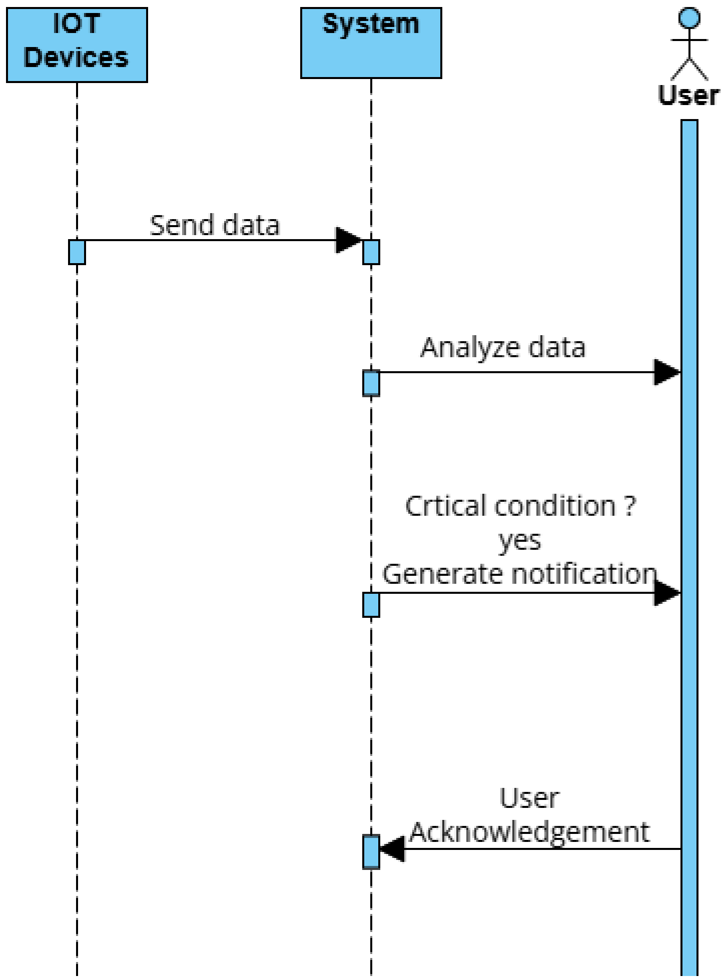
5.2.1.3 Monitor Temperature and Humidity



5.2.1.4 AI Model Training

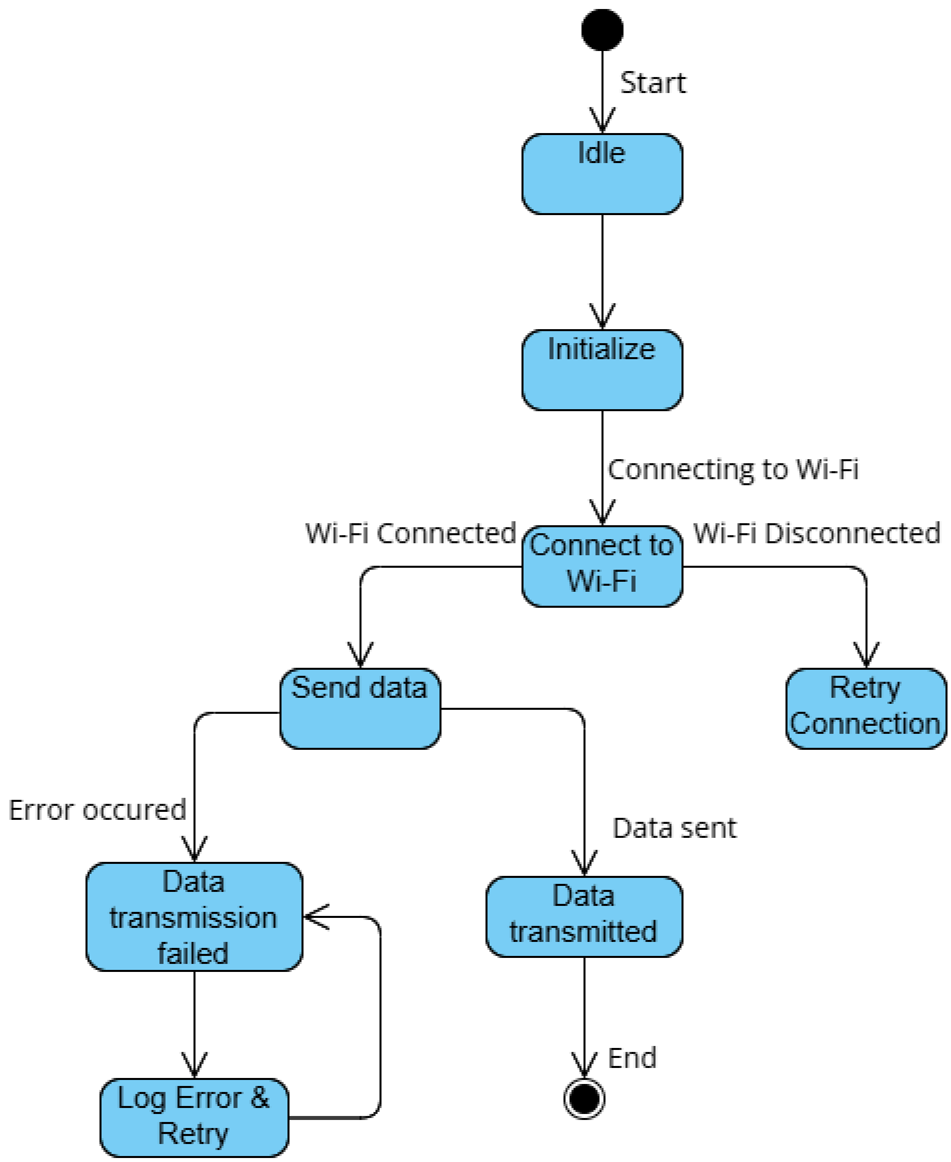


5.2.1.5 Critical Environmental Alerts

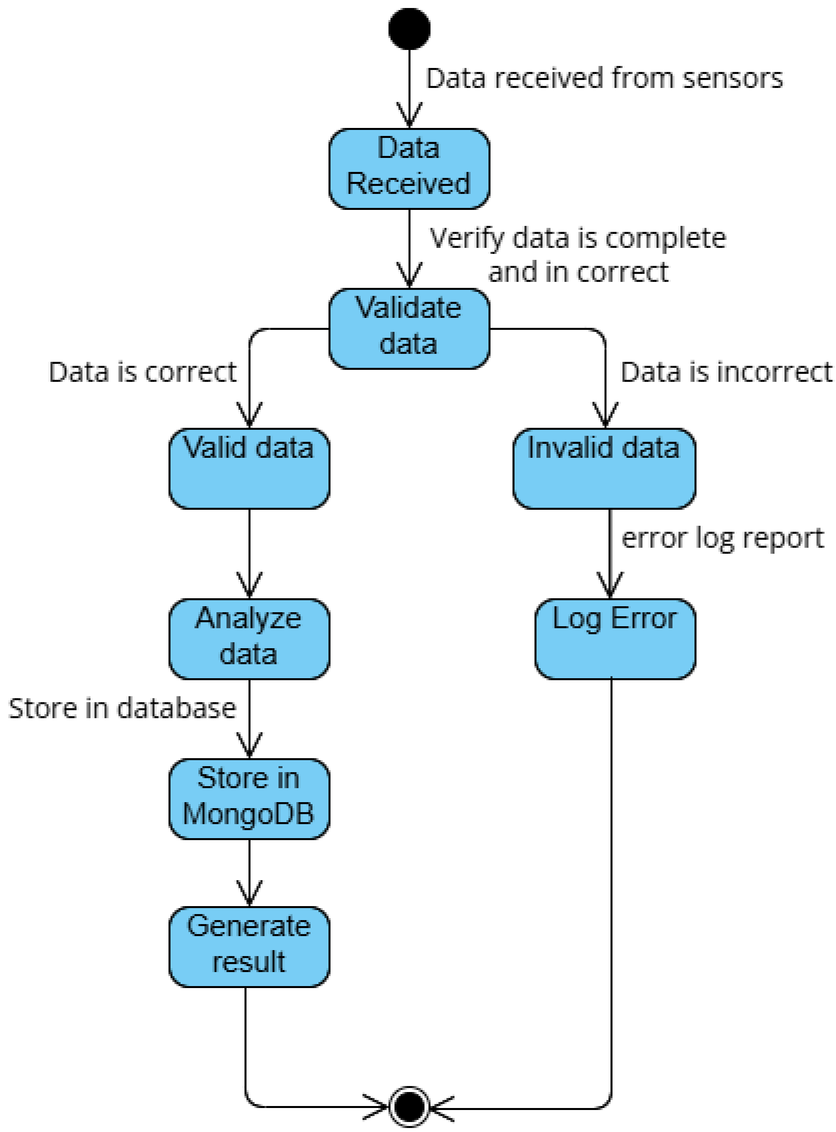


5.2.2 State Diagram

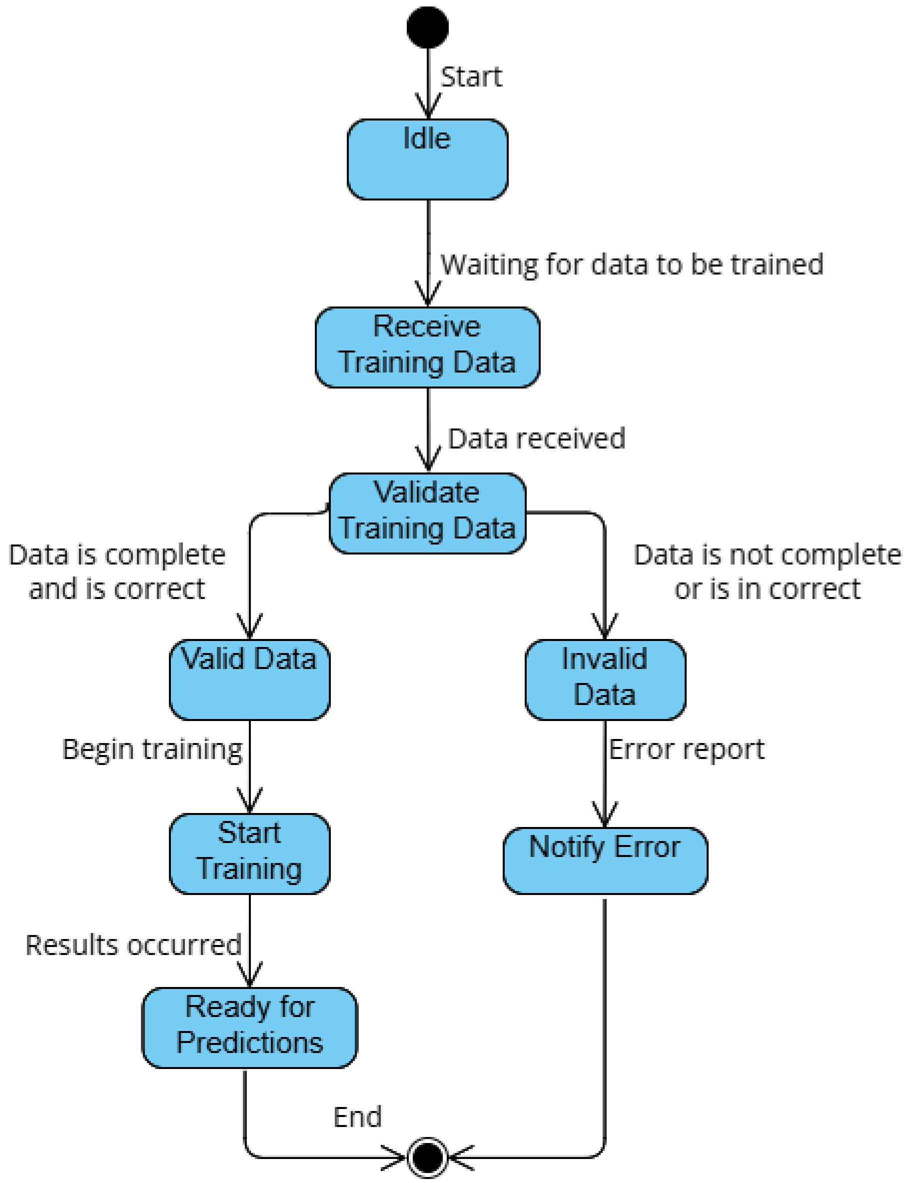
5.2.2.1 IoT Device State Diagram



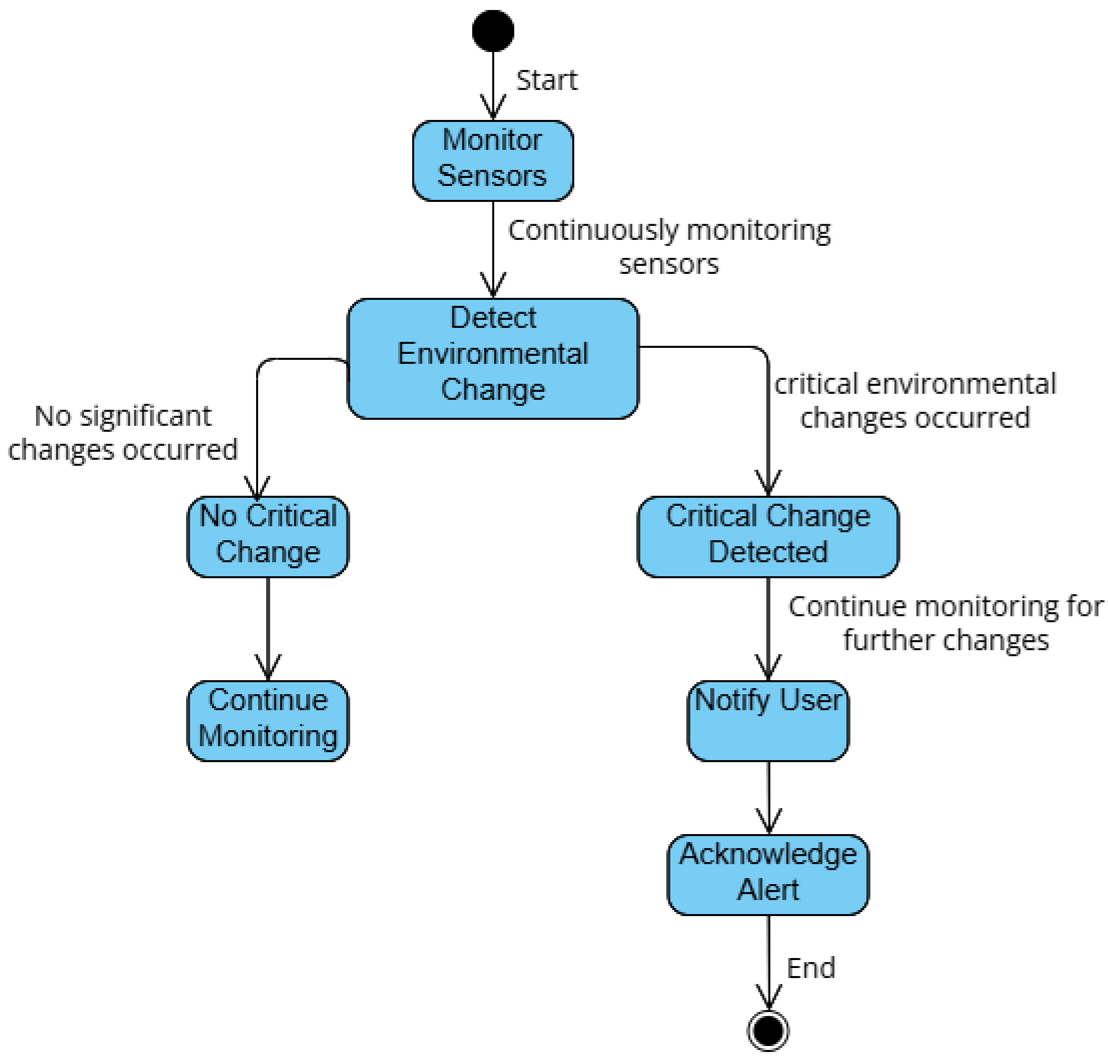
5.2.2.2 Data Processing State Diagram



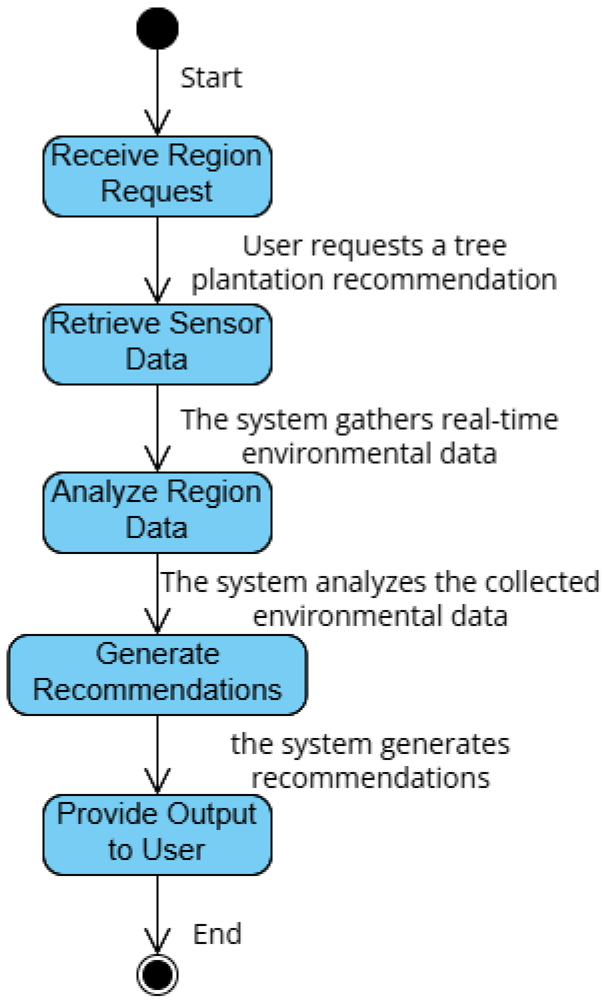
5.2.2.3 AI Model Training State Diagram



5.2.2.4 Environmental Alert State Diagram



5.2.2.5 Tree Plantation Recommendation State Diagram



6 References

Reference 01:

[1] B. K. Kaginalkar et al., "Review of urban computing in air quality management as smart city service: An integrated IoT, AI, and cloud technology perspective," Frontiers in Environmental Science, vol. 10, 2022.

Reference 02:

[2] K. L. Bowler et al., "Urban greening to cool towns and cities: A systematic review of the empirical evidence," Landscape and Urban Planning, vol. 182, pp. 12-24, 2019.

Reference 03:

[3] T. K. Patil et al., "An IoT Based Air Pollution Monitoring System for Smart Cities," Journal of Urban Technology, 2021.

Reference 04:

[4] P. Zhang et al., "Artificial Intelligence in Green Building," Energy and Buildings, vol. 247, 2021.

Reference 05:

[5] S. Lee et al., "Urban Vegetation Mapping from Aerial Imagery Using Explainable AI," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, 2021.

7 Appendices

This glossary provides definitions for terms and abbreviations used in this document.

Artificial Intelligence (AI):
The simulation of human intelligence in machines programmed to think and learn. AI is used in this project to analyze environmental data and provide insights for tree plantations.

Internet of Things (IoT):
A system of interconnected devices that communicate over the internet to collect and exchange data. In this project, IoT devices like ESP32 collect temperature, humidity, and AQI data for analysis.

ESP32 Microcontroller:
A low-power system-on-chip microcontroller with integrated Wi-Fi and Bluetooth, used to collect and transmit environmental data from sensors to the server.

DHT11 Sensor:
A digital sensor designed to measure temperature and humidity with moderate accuracy. It is connected to the ESP32 for environmental data collection.

MQ135 Sensor:
A sensor that measures air quality by detecting gases such as CO2, NH3, and NOx. It provides analog signals processed by the ESP32 microcontroller.

Predictive Analytics:
The process of using statistical algorithms and machine learning techniques to predict future environmental conditions, such as the impact of tree plantation on air quality and temperature.

GRU (Gated Recurrent Unit):
A type of recurrent neural network (RNN) used for processing sequential data, such as time-series measurements of temperature, humidity, and AQI..

Air Quality Index (AQI):
A measure of air pollution that indicates how clean or polluted the air is and its potential health impacts. MQ135 sensors provide the AQI used in plantation analysis.

Secure Wi-Fi Network:
Wi-Fi with WPA2 encryption ensures secure communication between IoT devices and the server, protecting sensitive data.

MongoDB:
A NoSQL database used to store structured environmental data collected by sensors, along with predictions and reports generated by the AI models.

Environmental Sensors:
Devices such as DHT11 and MQ135 that collect real-time data on temperature, humidity, and air quality, enabling location-specific recommendations.

Tree Plantation Impact Prediction:
The use of AI models to forecast the long-term benefits of tree plantation on environmental factors, such as air quality improvement and temperature regulation.

Weatherproof Enclosures:
Enclosures rated IP65 to protect IoT devices from environmental elements such as dust, rain, and high humidity, ensuring uninterrupted operation.

Data Timestamping:
The process of recording the exact time a data point is collected, ensuring chronological accuracy for analysis.

Billion Tree Tsunami:
A large-scale tree plantation campaign initiated in Pakistan, one of the potential beneficiaries of this project's data-driven recommendations.

MTJ (Maulana Tariq Jameel):
A foundation engaged in tree plantation campaigns, serving as a primary stakeholder for