6TH SEM
SEC: CS-8

A Project Report on

"Predictive Analytics and Green House Automation using Internet of Things for Remote Monitoring and Alert Generation"

Submitted By

Shashi Kant : 1405425
Shreyansh Nayak : 1405430
Yash D. Rishivanshi : 1405450

Under the guidance of

Dr. Arup Abhinna Acharya

Assistant Professor, School of Computer Engineering

SCHOOL OF COMPUTER ENGINEERING
KIIT UNIVERSITY

SCHOOL OF COMPUTER ENGINEERING
KIIT UNIVERSITY
BHUBANESWAR - 24

CERTIFICATE

This is to certify that the project entitled "Predictive Analytics and Green House Automation using Internet of Things for Remote Monitoring and Alert Generation" is being carried out by Shashi Kant (1405425), Shreyansh Nayak (1405430) and Yash D. Rishivanshi (1405450), as a curriculum project for 6th semester in Computer Science & Engineering at School of Computer Engineering, KIIT University, Bhubaneswar during the academic year 2016-2017 under my supervision. The matter embodied in this project is original and has not been submitted for the award of any other degree.

Signature of Project Mentor
(Dr. Arup A. Acharya)

ACKNOWLEDGEMENT

Apart from our efforts, the success of this project depends largely on the encouragement and guidance of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take immense pleasure in thanking and warmly acknowledging the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our project mentor, ***Dr. Arup A. Acharya****, Assistant Professor, School of Computer Engineering, KIIT University, in bringing this report to a successful completion.*

We also express our sincere thanks to all our friends who have patiently extended all sorts of help for accomplishing this undertaking. Also, we would like to express our heartfelt thanks to each of our beloved parents for their blessings, for their help and wishes for the successful completion of this project.

Finally we extend our gratefulness to one and all that are directly or indirectly involved in the successful completion of this project work.

Shashi Kant (1405425)
Shreyansh Nayak (1405430)
Yash D. Rishivanshi (1405450)

ABSTRACT

As the name specifies **"*Predictive Analytics and Green House Automation using Internet of Things for Remote Monitoring and Alert Generation*"** is about modern agriculture techniques. It offers monitoring and managing crop production within a synthetic environment provided by green house. Internet of things helps us to connect with the greenhouse remotely, monitor and generate critical alerts to take proper action. The increasing demand of agricultural products over the years brings out the importance of greenhouse to maintain the balance of supply and demand. Classical farming method is obsolete and less reliable however it is the major crop production method followed in India. Tropical and geological variation plays a key role in crop production. The classical method of farming is dependent on tropical and geological location, hence making it less reliable and the need of the hour is to modify this method to cope up with the demand. Also, in certain harsh climatic condition the crop fails and we are forced to import our food from global market which lowers our GDP, net price of the agro-products touches skies and become unreachable for middle class. Greenhouse comes in handy when we need to grow crops away from their tropical and geological location by creating synthetic climate for the plant to grow and nurture. This alone doesn't solves the problem because the parameters which are needed to be monitored and maintained are large and become complex for a very large spread greenhouse. In current greenhouse with the help of agro-sensors we can equip ourselves with a way to monitor. But again to monitor a big greenhouse and controlling is manually is not possible around the clock. A chance of missing critical alerts and acting upon it becomes high when monitored manually and is the major reason of crop failure in greenhouse plantation. To increase the productivity and making the greenhouse reliable certain agro-sensors like temperature, nitrogen, carbon-dioxide, soil moisture, soil pH and ambient light are deployed for monitoring purpose. A micro-controller connected with these sensors redirect the captured data or the real time data to the Internet-of-Things (IoT) cloud server for analysis. On basic of the analysis the alerts are generated and filtered and then is forwarded to the user to do the needed and hence will increase the productivity. This microcontroller when connected to the internet does not only alert the user but gives a full power to observe the entire facility remotely. Having such power at the finger tips enables us to monitor a large compound with ease. This would lead to the scaling up mass production of the agricultural products in synthetic environment with a cost effective manner. The collected data over the time for different crops could be shared with the field expert to get their advice and reviews. All of it could be done remotely.

Keywords: Internet-of-Things (IoT), Cloud, Wireless Sensor Node (WSN), Greenhouse, Microcontroller Unit (MCU), GDP, Crop Failure, Greenhouse Monitoring, Farming

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Problem: Greenhouse plantation is adopted as an alternative to catch up with the supply and demand in agricultural industry because it offers plantation in any season at any location. But it alone is not sufficient to meet up with the demand.

Requirement & Analysis: It is required to deploy modern technology in the agricultural industry to increase the supply. Although few greenhouses could be found offshore in which they have adopted modern technology to monitor the plantation but they happens to be working manually.

In order to achieve maximum output, IoT could help us to monitor and automate certain vital process. By this project a way of predicting the plant growth as well as around the clock monitoring is achieved.

1.1 METHODOLOGY

Prototype Model, Why?

The project is dependent on off the shelf hardware components and sensors. So, in order to achieve our goals, keeping the user posted about what is done so far, Prototype model comes in handy. Moreover, all the user requirements are not clear in one go because of huge variation in geo-location and climatic parameter changes in different regions, therefore, prototype model helps us compare all those variables in several iteration to fulfill the user requirement.

The Prototyping Model is a Software Development Life-cycle Model (SDLC) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users.

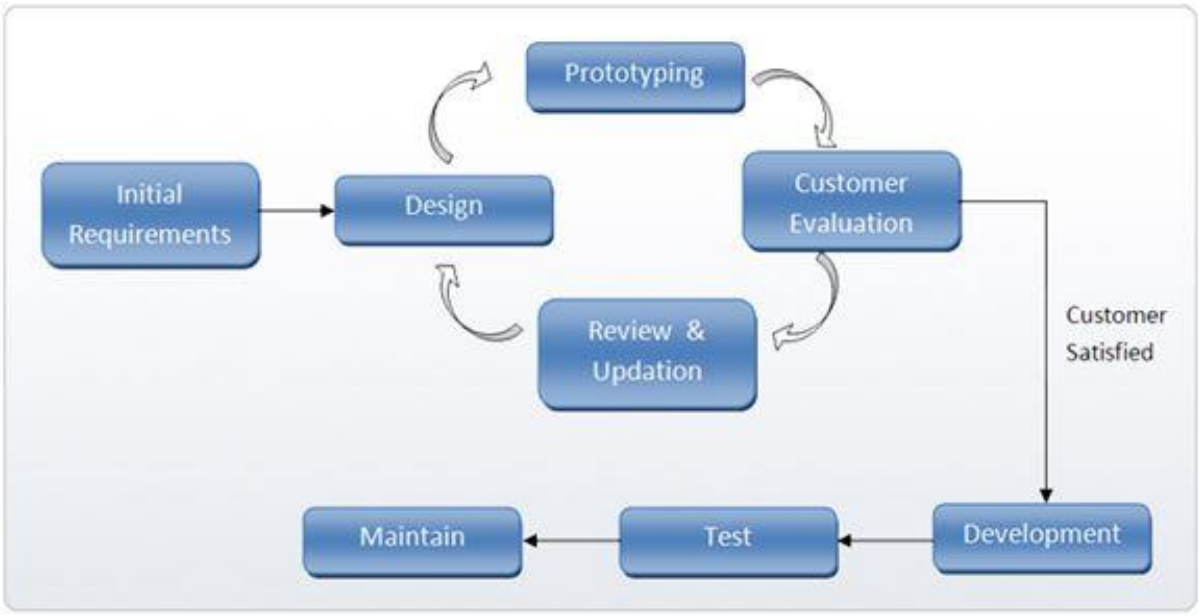

Fig 1.1: Prototype Model

1.2 PURPOSE

Agriculture is the backbone of India where more than 50% of our population is dependent on it. Still with the classical farming methods we are unable to catch up with the demand. So, in order to increase the productivity we must adopt advanced farming method. A few agriculture sensors could be attached to sense the climatic changes in order to monitor the facility around the clock. The data captured from the sensors will be stored online hence it can be provided for different research purpose to develop hybrid seeds to decrease the time period in which the crops/plants could grow in less time. Furthermore the microcontroller attached with sensors in the field will trigger real time alerts to report any fatal change in the climatic or soil condition. From the generated alert the concerned user could take designs to minimize any affect caused which will ultimately increase the productivity. Attaching the proposed idea in greenhouse plantation is even more effective as the planted crops in greenhouse needs closer monitoring. Data captured from the sensors can be directly forwarded to field experts to take their advice on use of fertilizers and pesticides. Hence this project will ensure minimum and effective use of fertilizers and pesticides to minimize any health effects. Best part is all of it could be done sitting remotely which enhances the user to survey even on a bigger farming land or greenhouse facility.

1.3 SCOPE

In order to automate the critical farming process using "SMART AGRICULTURE IN GREEN HOUSE USING IoT" along with software product an embedded sensor node is also offered to be deployed within the Greenhouse. This project aims to revolutionaries the farming industry by enabling us to monitor the climatic variables and different soil components, automating the irrigation system and controlling the supply of fertilizers and pesticides. To achieve all this, a web platform to monitor the farming and a wireless sensor node to be deployed within the field is being offered.

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

WSN (Wireless Sensor Network) – WSN is a wireless network consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions.

IoT (Internet of Things) – IoT is the interconnection of physical devices embedded with electronics like sensors etc.

GCS (Google Cloud Server) - Google Cloud Platform is a suite of public cloud computing services offered by Google. The Google Cloud platform includes a range of hosted applications and services for compute, storage and that run on Google in cloud.

MCU (Micro-Controller-Unit) - MCU is a small computer on a single integrated chip.

1.5 TECHNOLOGIES USED

GUI (Graphical User Interface) - JavaScript, CSS, HTML5, PHP, NodeJS, AngularJS, Python (Storing and retrieving data from the Database)

Backend - GCS (Google Cloud Server), MySQL, JavaSocket, LAMP (To store the data from sensors deployed within the field to a centralized Google cloud server for computing, storage and application development that run on Google hardware)

Embedded Systems - Arduino IDE, ArduPy, PySerial, UART, Parallel Programmer (A computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts)

Design Software - Eclipse Mars, MySQL WorkBench, Fritzing, Circuits IO (To store the data from sensors deployed within the field to a centralized Google clouds server)

Architecture - Ubuntu Server Edition, Windows Server 2016 (server-client architecture is used where server is Ubuntu client server and client is platform independent)

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

Many greenhouses could be found using sensors to monitor the growth of plant within the compound manually, but none have implemented IoT cloud database to monitor sensor data regarding growth rate with real time alert generation system for reliable monitoring around the clock. Moreover our project offers a way to store the data and compare it with the heuristic data and the stored data. The generated report could be forwarded to a field expert at any point of time to get their expert advice on fertilizers and pesticides to eliminate any crop failure and enhance the plant production.

2.2 USER CLASSES AND CHARACTERISTICS

As of now there are four user classes that are Admin, Maintenance, User and Field-Expert. The proposed IoT cloud platform will link all of them to a singular system. Role of admin is to manage database and schedule maintenance. Role of Maintenance team is to install and maintain the sensors and hardware interface at the facility. Field-Expert will provide their advice on soil type, fertilizers, pesticides and the climatic conditions to be maintained. Role of user is to use the platform for monitoring of their greenhouse facility.

2.3 OPERATING ENVIRONMENT

All the agricultural sensors will be attached with and embedded MCU (Arduino or ATMega 328 chip). The web interface for monitoring is platform independent. Backend of the system which stores the data in SQL server is needed to run on Ubuntu 14.04 Server edition or Windows Server 2016 on GCS.

2.4 USER DOCUMENTATION

User will get the help section on the portal which will guide them to operate their facility with ease and in some cases they can schedule a maintenance visit to evaluate the sensors and MCU. User will get all the required information as a "User's Guide" booklet which will specify all the do's and dont's.

2.5 SOFTWARE INTERFACES

To run the web-interface:

- Apache and Tomcat Server (For capturing data and generating alerts)
- HTML, PHP, JS, JSP and NodeJs (Storing and retrieving data from the DB)
- MySQL (To store the data from sensors deployed within the field to a centralized server)
- JavaSocket (For authenticating a device and receiving data using SSL)
- GCS to run Ubuntu 14.04 Server and Windows 2016 Server edition

2.6 HARDWARE INTERFACES

- Arduino Uno, Agriculture Sensors (Soil pH, Nitrogen, Carbon Mono-Oxide, Soil-Moisture, Ambient Light, Humidity, Soil-Temperature, Ambient-Temperature)
- PSU to supply power to the sensors
- Access Point to provide internet connectivity to the sensors.
- Since the data is in the cloud it can be monitored from any platform

2.7 ASSUMPTIONS

It is assumed that the client is having pre-installed water supply and power supply.

2.8 SCHEDULE FOLLOWED

Following is the tabular representation of the schedule followed by the team members to carry out the project up to an acceptable level.

Topic	Start Date	End Date
SRS Documentation	12-Jan-2017	24-Jan-2017
Hardware Prototype Building	14-Jan-2017	13-Feb-2017
Backend Design	25-Jan-2017	28-Feb-2017
Frontend Design	26-Jan-2017	15-Mar-2017
Testing	16-Mar-2017	25-Mar-2017
Report	25-Mar-2017	30-Mar-2017

Fig 2.1: Schedule Followed

3. PROJECT PLANNING

During the planning of this project, different farmers, technicians, citizens and the admin were interviewed and then the result was analyzed to determine the functional and non-functional requirements and the constraints.

3.1 QUESTIONNAIRE

ANALOG MODE: Questionnaire session with the client to get their requirements.

- Why adopt new methods of farming?
- What are the reasons behind the failure of supply vs. demand?
- What major problems one is facing at their greenhouse?
- How regularly you visit to see the farming site?
- What crops usually you want to cultivate?
- Would you like to monitor your facility remotely?
- Is your Greenhouse located in remote area?
- Is there any reliable source of power supply?
- What kind of irrigation system are you using and does it contain any foreign material unsuitable for crops you wish to grow?
- How often you consult to a field expert if things go south at your facility?
- Would you like to include the expert advice on pesticides and fertilizers?
- What connectivity is available at the farming site? Would you like to include remote GSM or Satellite connectivity?
- How technical are staffs/helpers currently involved at farming sites?
- Would you like to have a training program and hands on with the new system?
- Would you prefer scheduled maintenance?
- Would you like to have irrigation, mixing, pest-control, synthetic lightning and temperature controller to operate automatically?
- How often do you go out of the station? Who will be alternate point of contact for urgent response?

DIGITAL MODE: In order to get certain vital information regarding the climatic and soil condition of the facility which will support development in building a robust solution and ensure 100% up-time.

- Get samples of soil
- Test for soil pH, Nitrogen, phosphorous and Potassium content
- Getting average humidity, moisture and temperature value
- Checking turbidity of Ground water used for irrigation
- Based upon the Geo-location finding common pests for particular plants
- Gathering all the above data and sending to field expert to get advice on climate control and other parameters to be monitored for optimal yield

3.2 FUNCTIONAL REQUIREMENTS

❖ **R1: Registration:**

Description: Users have to register to the user web portal.

❖ **R2: Administration:**

Description: Admin approves and schedules a field insist to install the sensors.

❖ **R3: Farm Monitoring:**

Description: Field monitors sense or monitor the sensors attached to the soil.

❖ **R4: Farming Expert Guide:**

Description: Field experts give their feedback on soil types, pesticides, fertilizers.

❖ **R5: Maintenance:**

Description: Any error being generated in the sensor system is being reported to the maintenance team that works on it.

❖ **R6: User Portal:**

Description: The HTML based web page where the user logs in to access the system.

❖ **R7: Report Generation:**

Description: After the real time data of the climatic parameters is being recorded from the sensor then a report is generated.

❖ **R8: Database:**

Description: The recorded data of the sensors is stored in the database.

3.3 NON-FUNCTIONAL REQUIREMENTS

❖ **Performance Requirements:** The hardware sensor response time must be real time i.e., less than two seconds. The database created in the cloud must follow ACID property. Each sensor node should perform as an independent node and must have a separate table. Concurrent submission of data in the server and cloud database might not decrease the submission speed of data. The online portal must redirect the recent captured data to the server without refreshing the entire page and must be platform independent.

❖ **Safety Requirements:**

- ✓ Multiple nodes could submit their data to the same port for a particular node.
- ✓ Another server must not run on the same port as of the Apache Tomcat and NodeJs
- ✓ Entire web portal must follow encrypted cookie based session handling

❖ **Security Requirements:** The sensor node after connecting to the server must send encrypted token to validate and afterwards publish the data in database. The database must be replicated periodically and kept safe. In case of crash, the system should be able to backup and recover the data. User web portal must be free of DDoS attack (Distributed Denial of Service) and SQL injection.

3.4 CONSTRAINTS

- The application will be available only in English
- Only registered users can utilize the services of the application
- Valid user information is mandatory to create a profile so as to provide accurate data to the correct users
- Users who have the knowledge of accessing net facilities can only utilize the offered services

4. DESIGN

In this section, the interaction between the user and the system, message passing, control flow from one activity to another and the structure of the system are depicted using different UML diagrams.

4.1 UML DESIGN

4.1.1 USE-CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

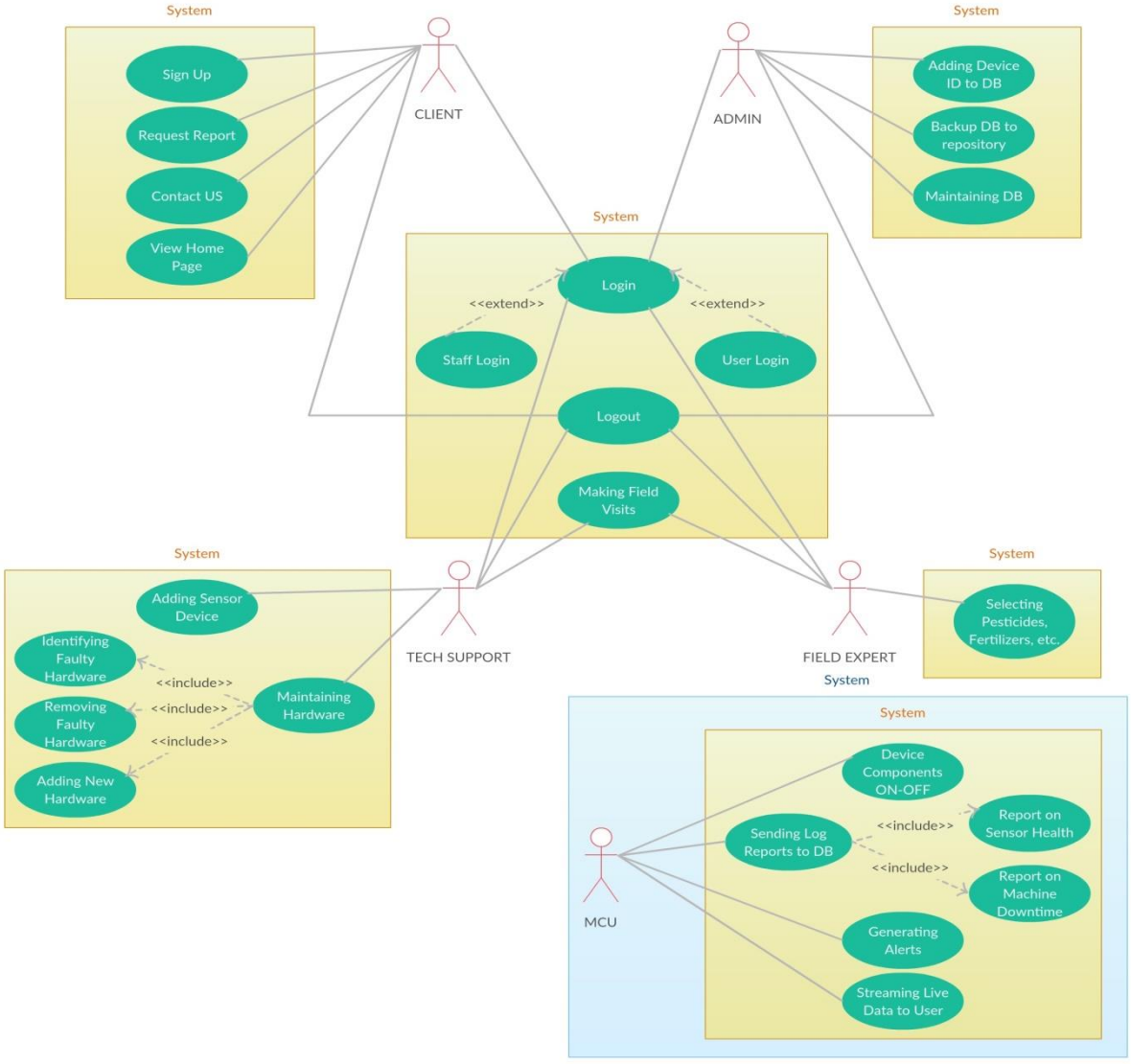

Fig 4.1: Use-Case Diagram

USE-CASE SCENARIOS:

A. Scenario for "Sign Up" Use-Case:-

Table 4.1: Use-Case Scenario for "Sign Up"

USE CASE NAME	Sign Up
PRIMARY ACTORS	Client
SUMMARY	Every unregistered user will need to sign up and create a profile by filling a form on the page. The profile is created after all the fields are validated.
PRE-CONDITION	▪ Net Connection is available ▪ Page should be displayed
FLOW OF EVENTS	1. User clicks on "Sign Up" button 2. User is re-directed to the sign-up form 3. User enters the details and then presses the "Submit" button 4. The details are validated 5. Upon successful validation, the profile is created and the user is re-directed to the "Login Page"
EXTENSIONS	3.1 User doesn't enter the details 3.2 User doesn't enter details correctly 4.1 Entered details are in-valid 5.1 Error message will be displayed

B. Scenario for "Login" Use-Case:-

Table 4.2: Use-Case Scenario for "Login"

USE CASE NAME	Login
PRIMARY ACTORS	Client, Admin, Tech Support, Field Expert
SUMMARY	Every registered user will need to login to use the services of the system. The system will validate the login credentials.
PRE-CONDITION	▪ Net Connection is available ▪ Page should be displayed ▪ Profile of user should exist
FLOW OF EVENTS	1. User clicks on the "Login" button 2. User is re-directed to the "Login Page" 3. User enters the log-in credentials and then presses the "LOGIN" button 4. The details are validated 5. Upon successful validation, the user is re-directed to his/her "Dashboard Page" (User Home Page)
EXTENSIONS	3.1 User doesn't enter the details 3.2 User doesn't enter details correctly 4.1 Entered details are in-valid 5.1 Error message is displayed

C. Scenario for "View Homepage" Use-Case:-

Table 4.3: Use-Case Scenario for "View Homepage"

USE CASE NAME	View Homepage
PRIMARY ACTORS	Client, Admin, Tech Support, Field Expert
SUMMARY	Every user will need to view the homepage to use our system.
PRE-CONDITION	▪ Net Connection is available ▪ Page should be displayed ▪ Page should be available
FLOW OF EVENTS	1. User sends a request to view the homepage by providing the "Page URL" 2. The page is fetched from the database 3. The user is directed to the "Home Page"
EXTENSIONS	1.1 User doesn't enter any URL 1.2 User doesn't enter the correct URL 3.1 Error message will be displayed

D. Scenario for "Contact Us" Use-Case:-

Table 4.4: Use-Case Scenario for "Contact Us"

USE CASE NAME	Contact Us
PRIMARY ACTORS	Client
SUMMARY	A user can contact us to solve any query they might have regarding the system or the services offered by the system.
PRE-CONDITION	▪ Net Connection is available ▪ Page should be displayed ▪ Page should be available
FLOW OF EVENTS	1. User clicks on the "Contact Us" button 2. User is re-directed to the "Contact Us Page" 3. User fills up the query form and then presses the "Submit" button 4. The details are then saved onto the database 5. Confirmation message is then displayed
EXTENSIONS	3.1 User doesn't enter the details 3.2 User doesn't enter details correctly 5.1 Error message will be displayed

E. Scenario for "Update DB" Use-Case:-

Table 4.5: Use-Case Scenario for "Update DB"

USE CASE NAME	Update DB
PRIMARY ACTORS	Admin
SUMMARY	The admin can update and maintain the database by logging into to the system first.
PRE-CONDITION	▪ Net Connection is available ▪ Page should be displayed ▪ Page should be available ▪ Admin profile should be available
FLOW OF EVENTS	1. Admin clicks on the "Login" button 2. Admin is re-directed to the "Login Page" 3. Admin fills up the login credentials and then presses the "Login" button 4. The details are then validated 5. Upon successful validation the admin is re-directed to his/her dashboard 6. Admin tries to update the database 7. The controller then saves the data onto the database
EXTENSIONS	3.1 User doesn't enter the details 3.2 User doesn't enter details correctly 4.1 Entered details are in-valid 5.1 Error message is displayed

F. Scenario for "Request Report" Use-Case:-

Table 4.6: Use-Case Scenario for "Request Report"

USE CASE NAME	Request Report
PRIMARY ACTORS	Client
SUMMARY	Every registered user can request for reports after successfully logging into the system.
PRE-CONDITION	▪ Net Connection is available ▪ Page should be displayed ▪ Profile of user should exist
FLOW OF EVENTS	1. User clicks on the "Login" button 2. User is re-directed to the "Login Page" 3. User enters the log-in credentials and then presses the "LOGIN" button 4. The details are validated 5. Upon successful validation, the user is re-directed to his/her "Dashboard Page" (User Home Page) 6. User can then request for a report 7. The controller then generates a report for the user
EXTENSIONS	3.1 User doesn't enter the details 3.2 User doesn't enter details correctly 4.1 Entered details are in-valid 5.1 Error message will be displayed

4.1.2 CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

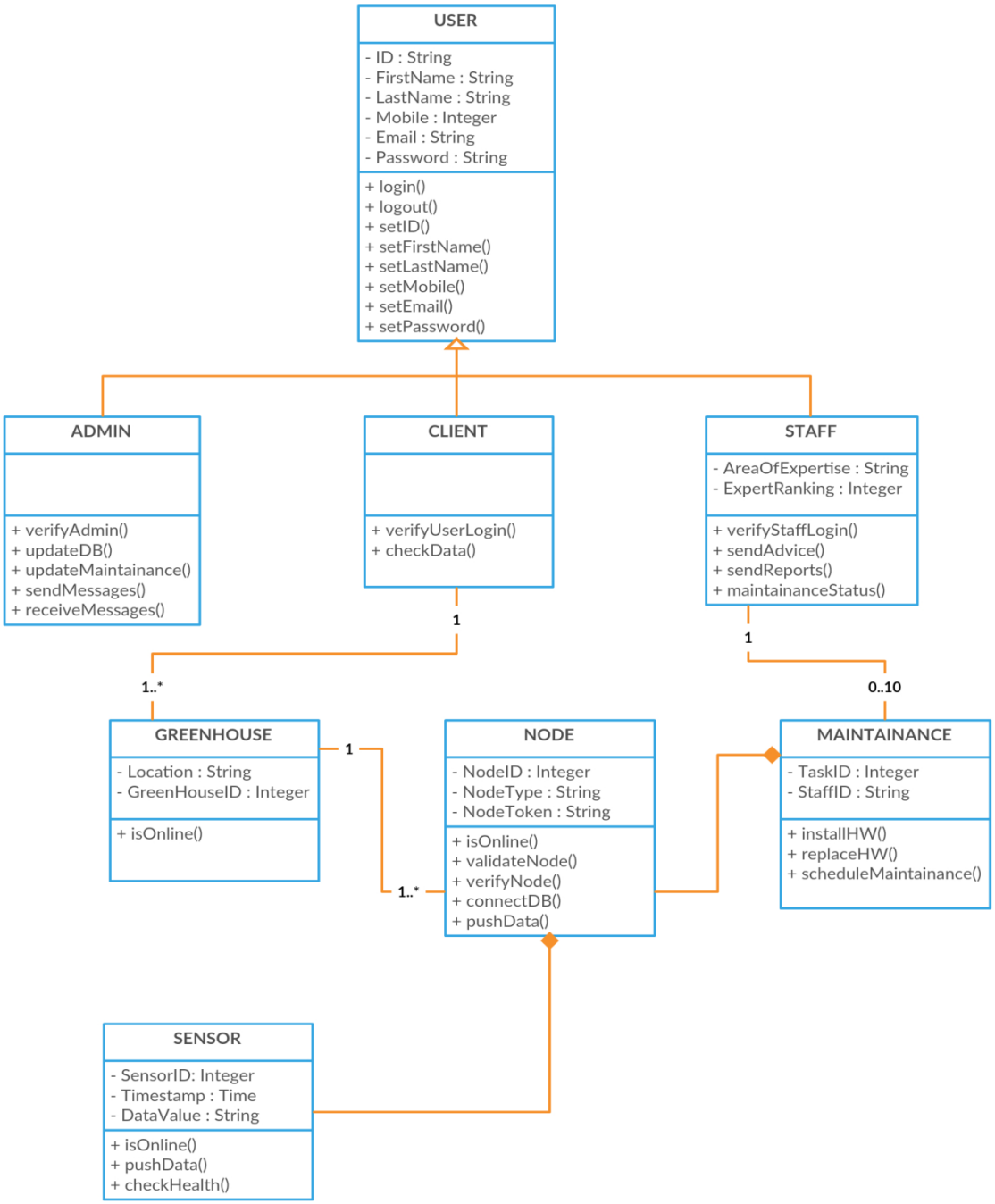

Fig 4.2: Class Diagram

4.1.3 SEQUENCE DIAGRAMS

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A. Sequence Diagram for "Sign Up" Use-Case:-

This sequence diagram shows the registration of new user. Here the user first requests for the form. Then the controller sends for the verification of the form to the admin. After the verification has been done, the account is successfully created.

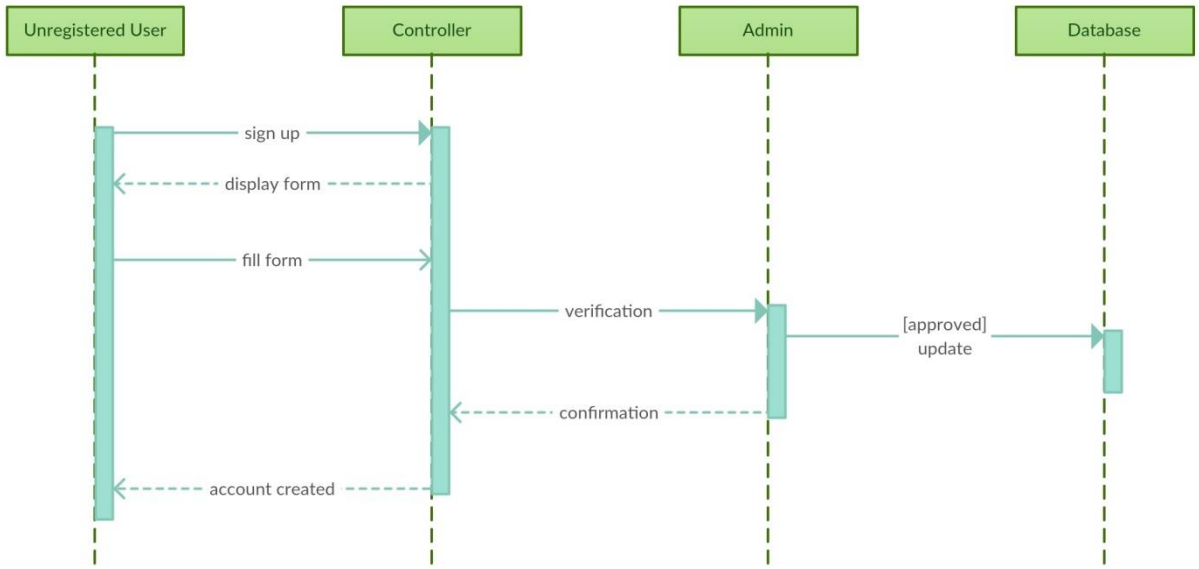

Fig 4.3: Sequence Diagram for "Sign Up"

B. Sequence Diagram for "Login/Logout" Use-Case:-

This sequence diagram shows how a user can login and logout. He first requests for login, and then the controller sends for the verification of the details entered by the user. Then the database verifies it. If verification is successful, the user is directed to his/her dashboard.

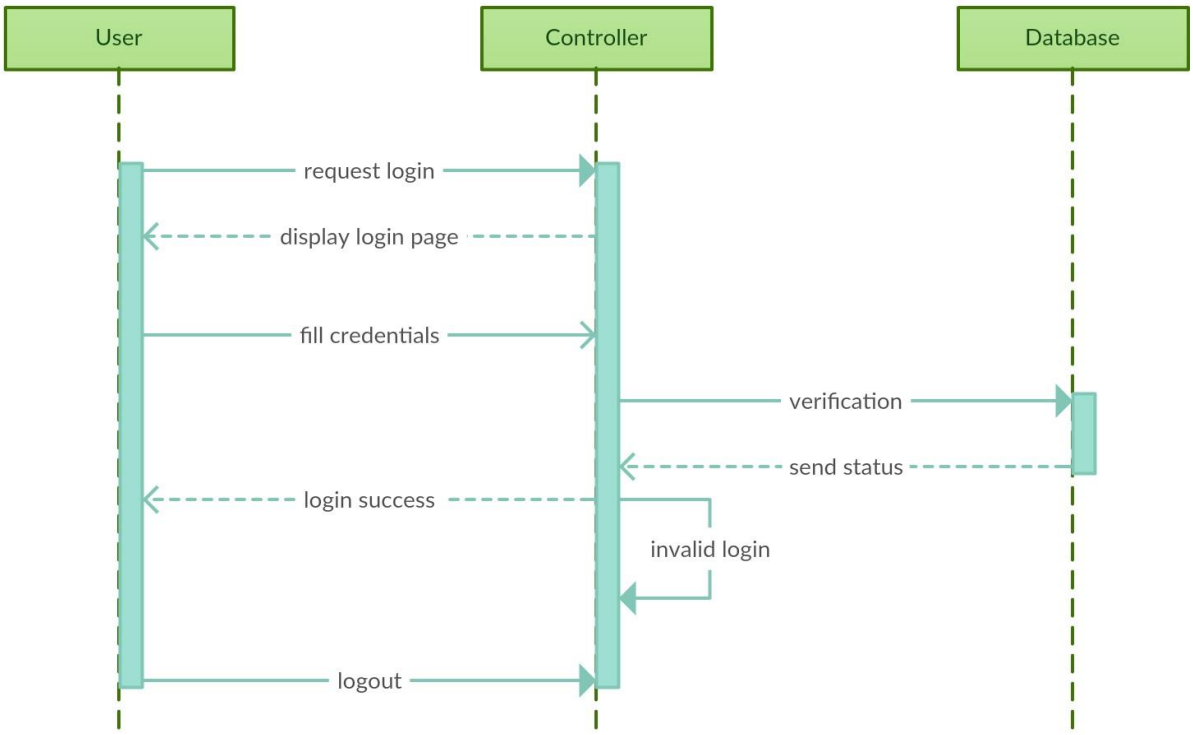

Fig 4.4: Sequence Diagram for "Login/Logout"

C. Sequence Diagram for "View Homepage" Use-Case:-

This sequence diagram represents the process through which a user can view the homepage. The user requests the home page. The Controller then sends the request to database. The controller then displays the homepage successfully.

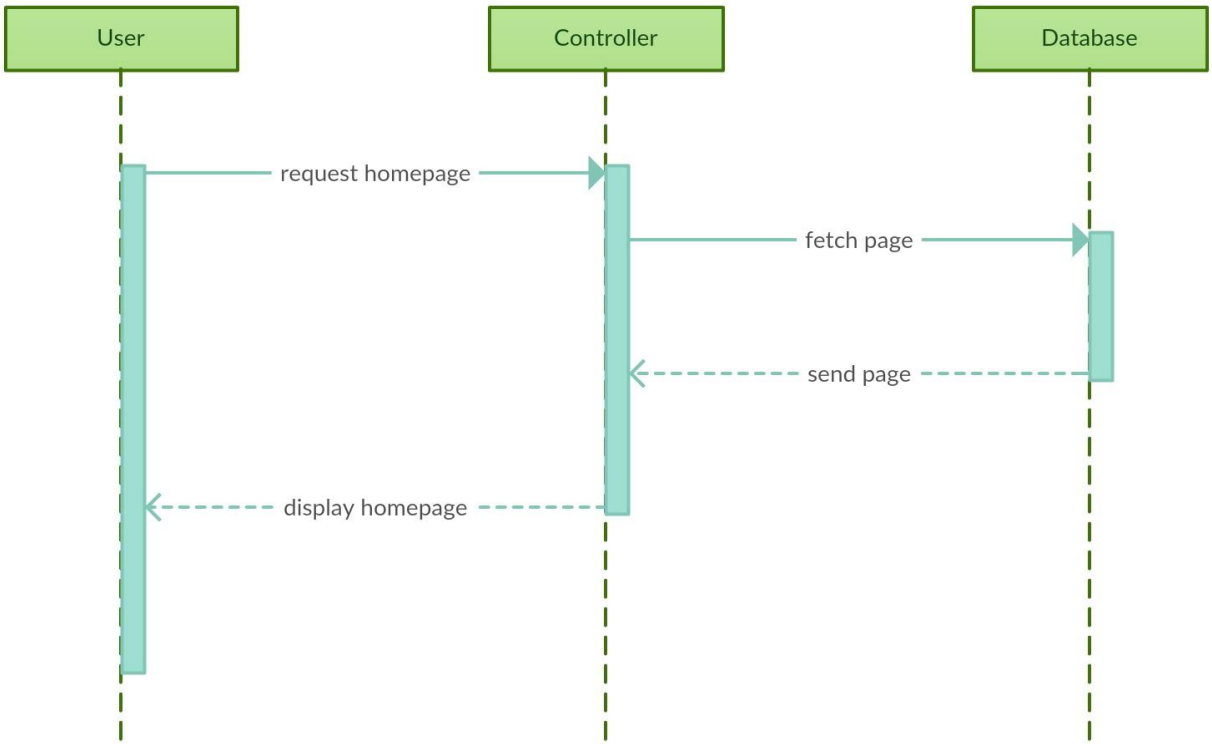

Fig 4.5: Sequence Diagram for "View Homepage"

D. Sequence Diagram for "Contact Us" Use-Case:-

This sequence diagram represents the process through which a user gets to contact us. The user requests the page where he could contact us. The Controller then saves the filled form to the database.

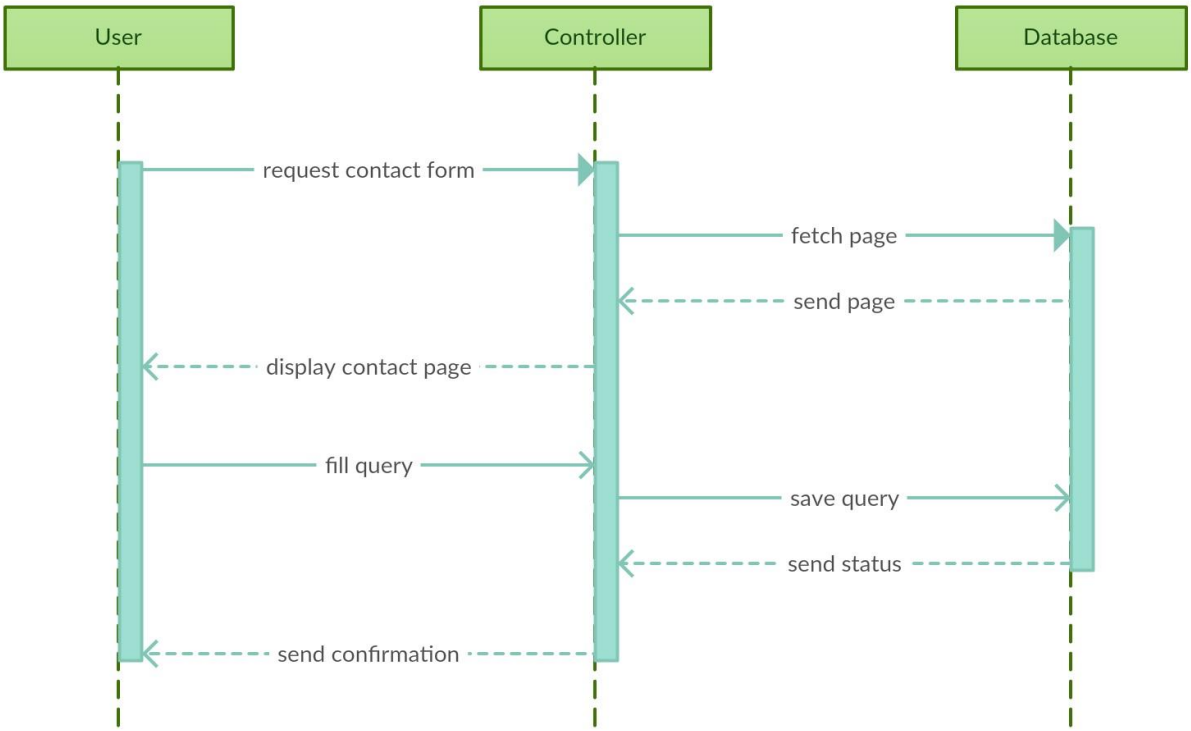

Fig 4.6: Sequence Diagram for "Contact Us"

E. Sequence Diagram for "Update DB" Use-Case:-

This sequence diagram represents the process through which the admin updates and maintains the database. The admin logs into his account first to update database. The Controller then records the details into the database.

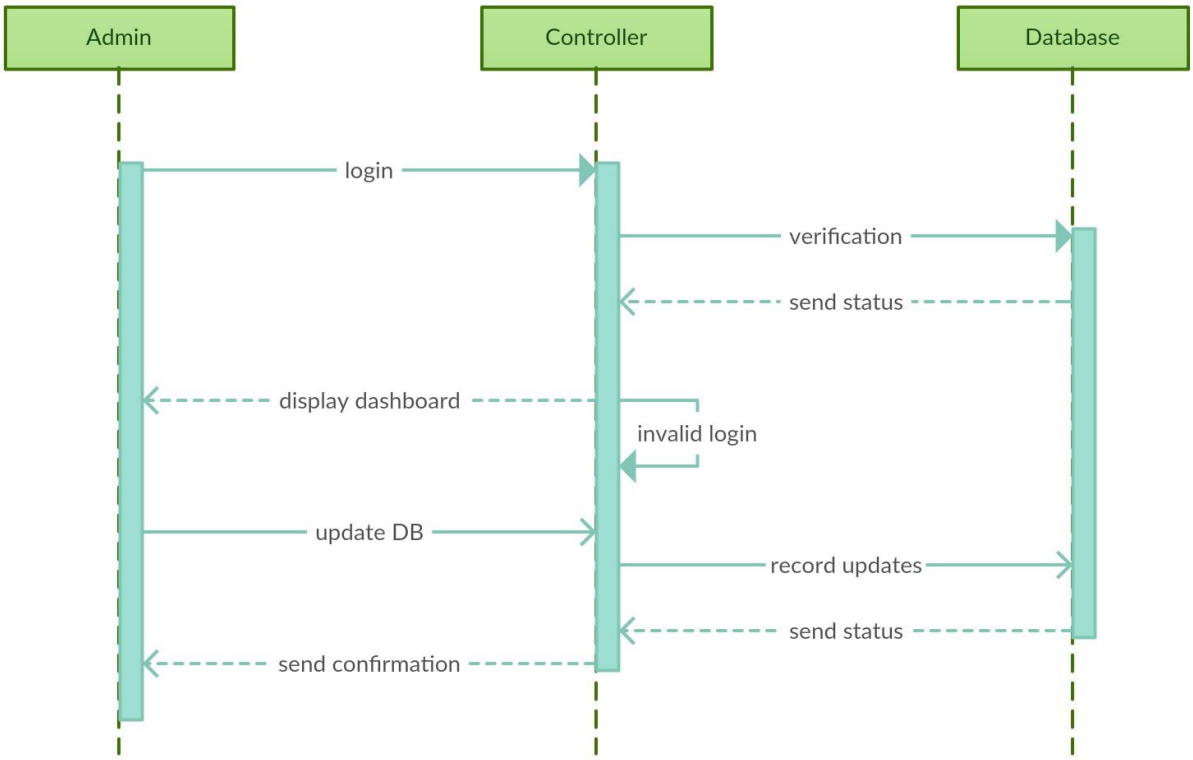

Fig 4.7: Sequence Diagram for "Update DB"

F. Sequence Diagram for "Request Report" Use-Case:-

This sequence diagram represents the process through which the user request for report. The user logs into his account first to request a report. The Controller then fetches the report from the database and displays it to the user.

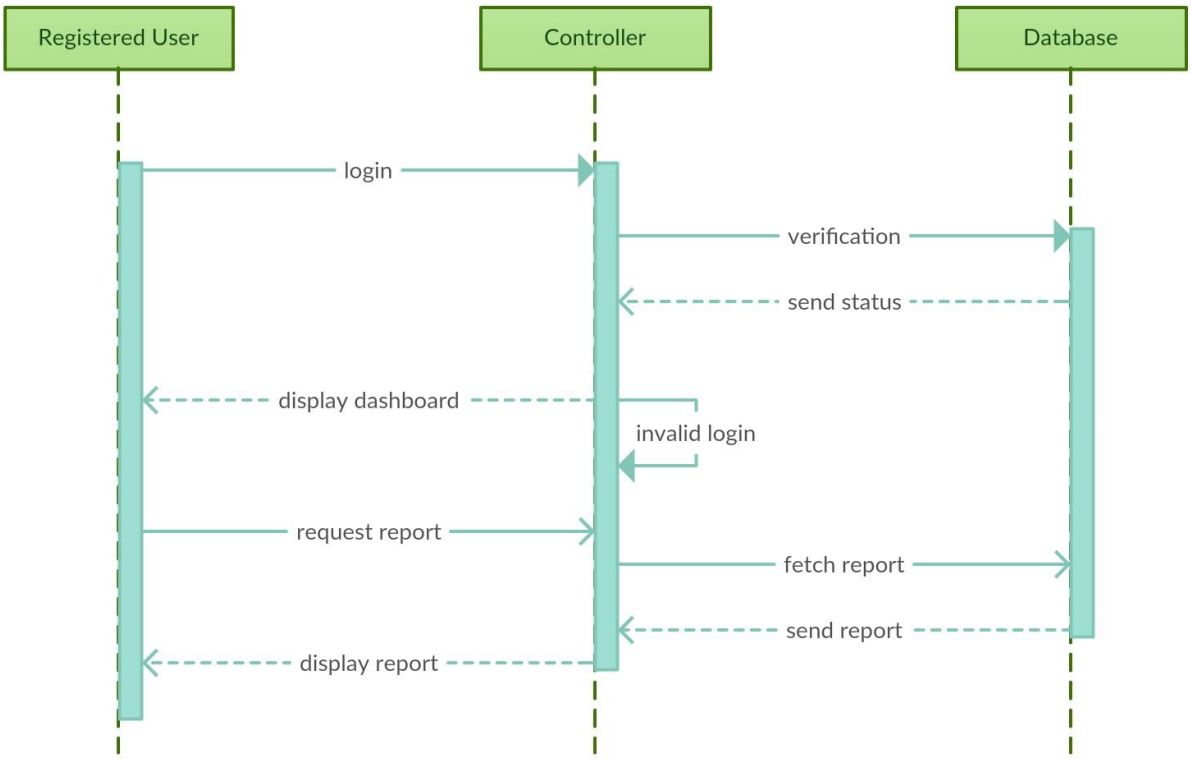

Fig 4.8: Sequence Diagram for "Request Report"

G. Sequence Diagram for "Send Log Report" Use-Case:-

This sequence diagram represents the process through which the MCU sends log report to the database. The MCU sends the log report. The Controller then records the details into the database.

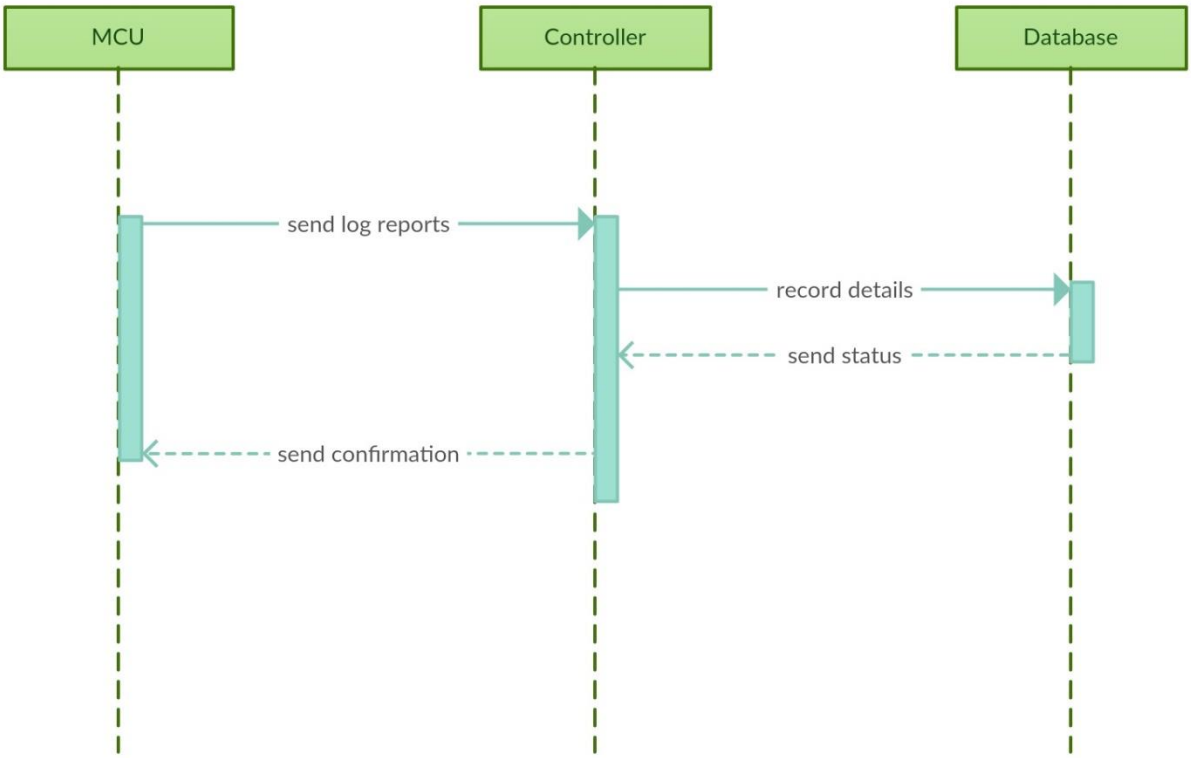

Fig 4.9: Sequence Diagram for "Send Log Report"

H. Sequence Diagram for "Stream Live Data" Use-Case:-

This sequence diagram represents the process through which the MCU streams live data to the registered user. The MCU requests for the live data. The Controller then fetches the data from the database and forwards it to the MCU. The MCU then streams the data to the user.

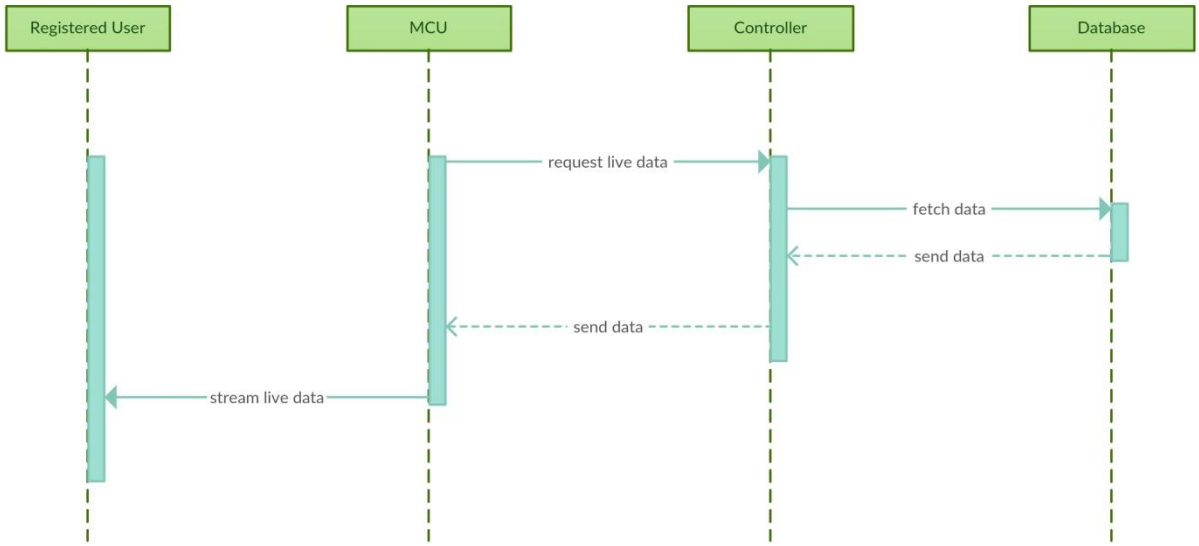

Fig 4.10: Sequence Diagram for "Stream Live Data"

4.1.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modeling where they are used for detailing the processes involved in business activities.

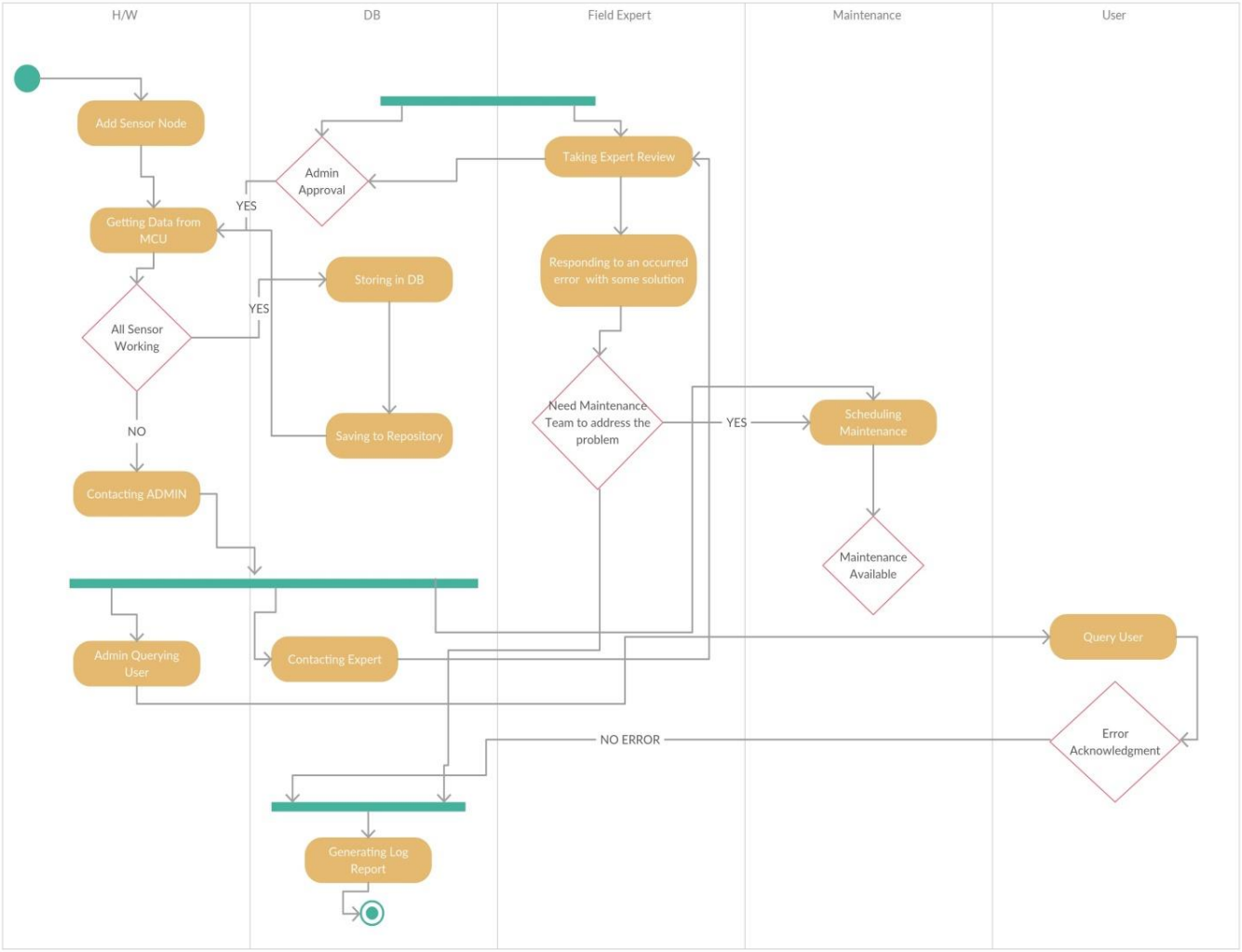

Fig 4.11: Activity Diagram

4.2 DATABASE DESIGN

4.2.1 ER - DIAGRAM

An entity-relationship model is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities and the relationships that can exist among them.

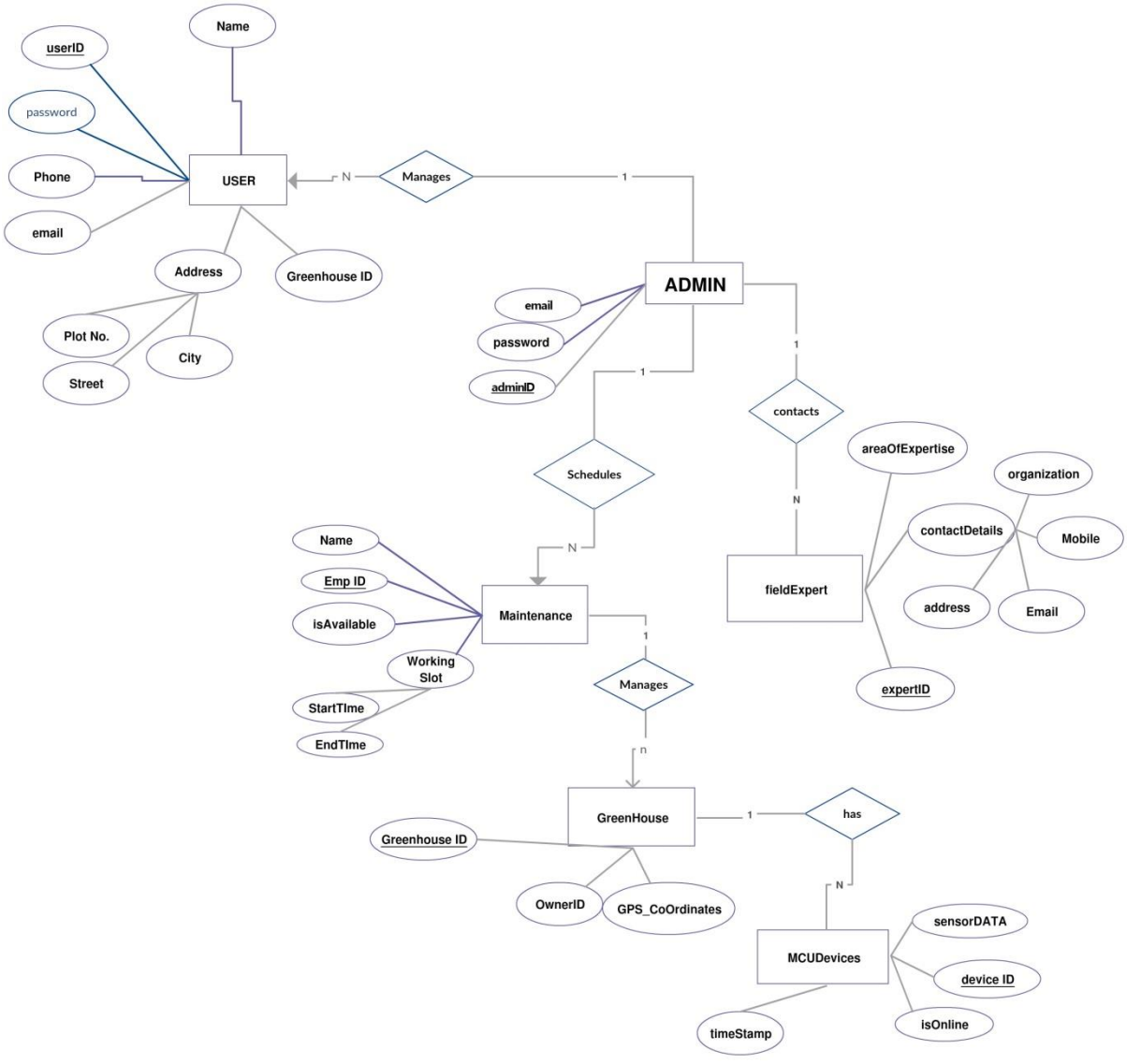

Fig 4.12: ER Diagram

4.3 GUI DESIGN

The front end is designed using JavaScript, CSS and HTML. It consists of one Login Page, New User Registration Page, Home Page, Dashboard and Logout page.

4.3.1 HOMEPAGE

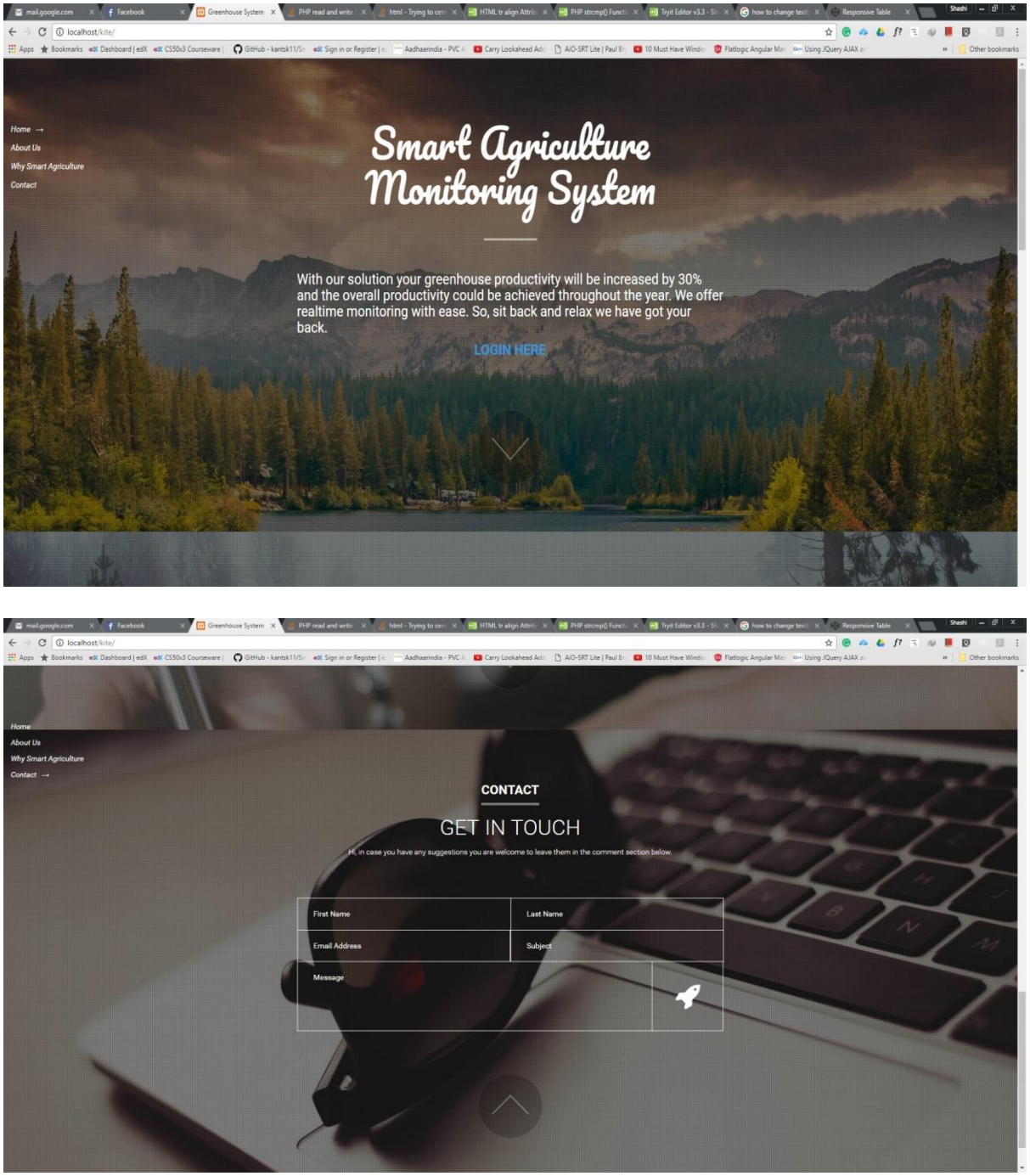

Fig 4.13: Homepage

4.3.2 LOGIN PAGE

Users who have earlier registered could login to the page using their email-ID and password as login credentials.

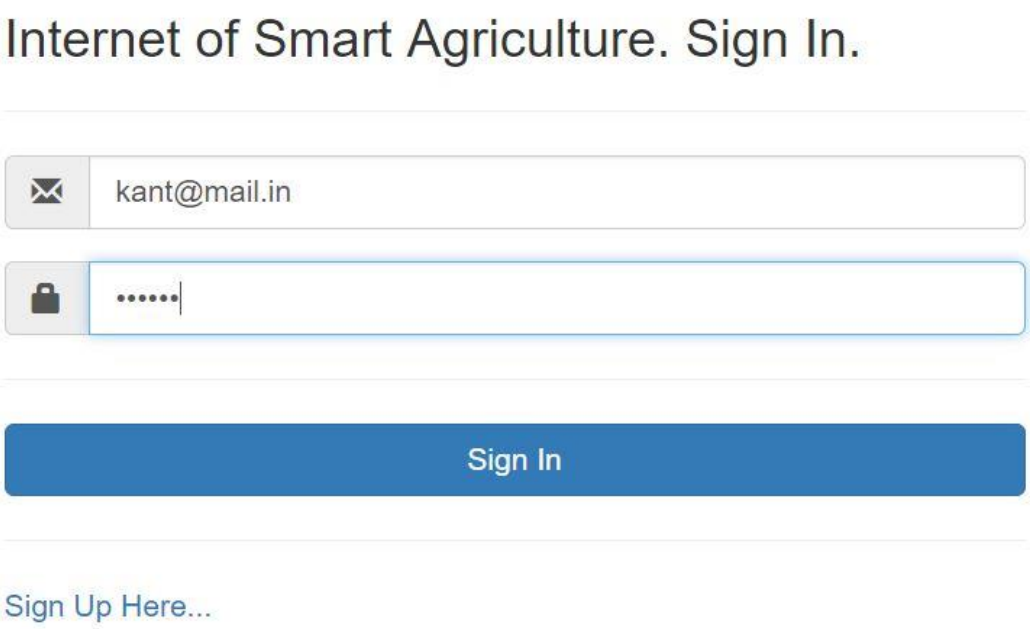

Fig 4.14: Login Page

4.3.3 REGISTRATION PAGE

New user registration page requires email-ID, name and password to sign up.

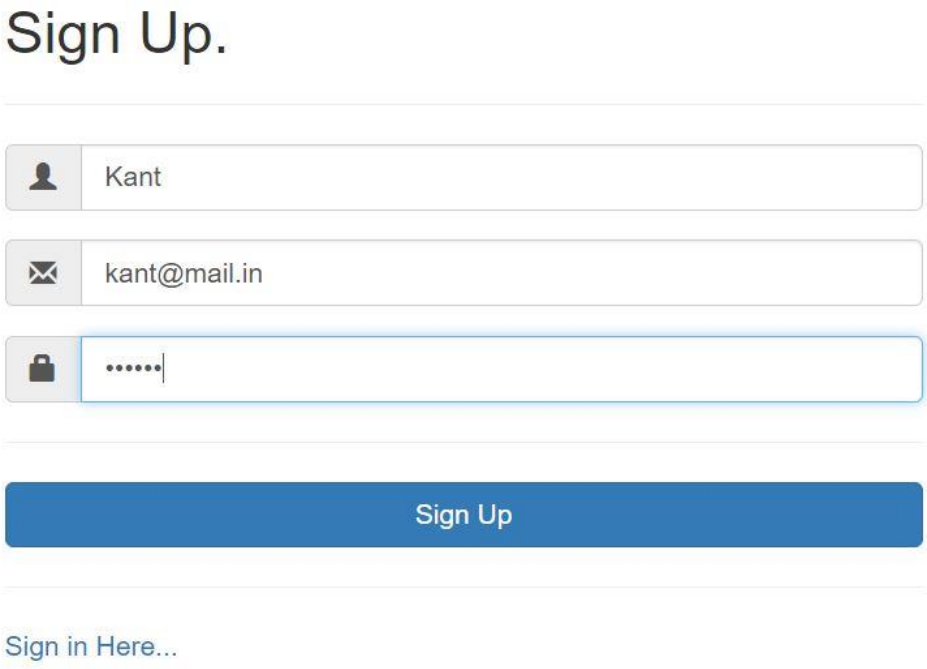

Fig 4.15: Registration Page

4.3.4 DASHBOARD (USER HOMEPAGE)

To monitor the greenhouse, dashboard is used. It shows the real time data fetched directly from the field.

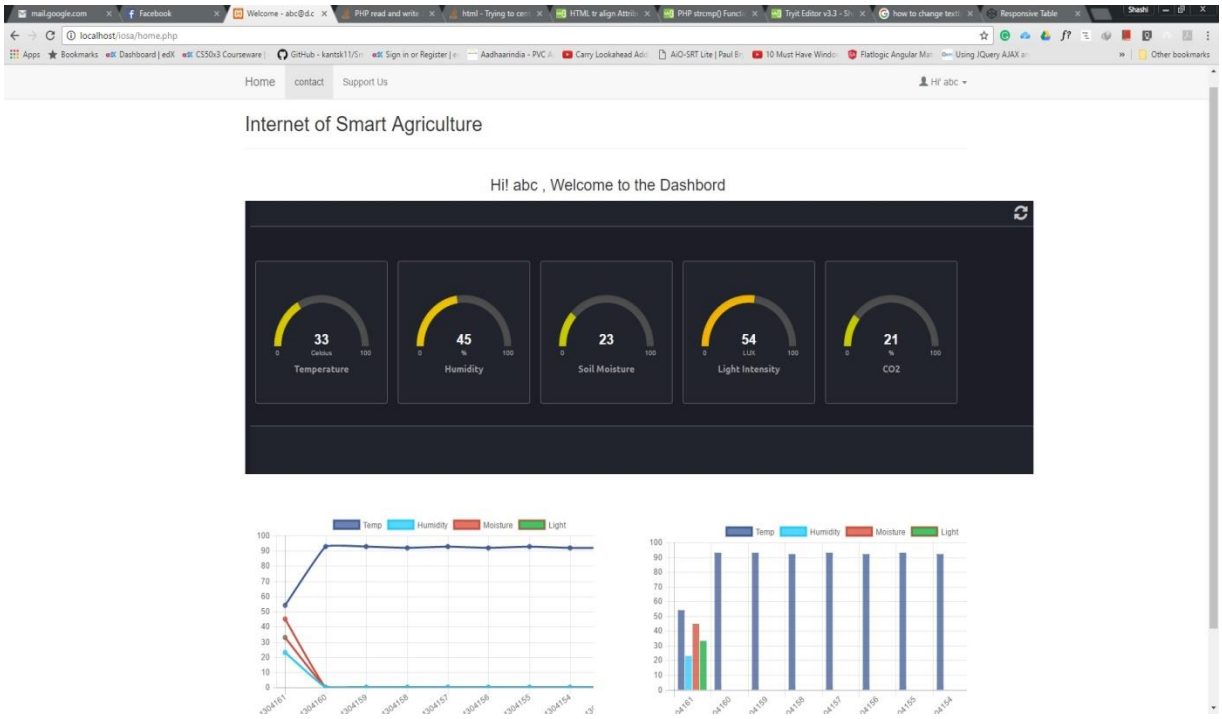

Fig 4.16: Dashboard

4.4 TEST CASES DESIGN

A. Test Cases for "Login":-

Following table depicts the test cases for manual testing of the User Login page to ensure optimal performance and security in the system.

Table 4.7: Test Cases for "Login"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT
1	Login_All_Blank	"LOGIN" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter username and password
2	Login_PWD_Blank	"LOGIN" button is active	Enter the username and leave the password field blank	Must prompt the user to enter password
3	Login_User_Blank	"LOGIN" button is active	Enter the password and leave the username field blank	Must prompt the user to enter username
4	Login_Invalid_PWD	"LOGIN" button is active	Enter the correct username and incorrect password	Must display an error message and prompt the user to enter valid password
5	Login_Invalid_User	"LOGIN" button is active	Enter the correct password and incorrect username	Must display an error message and prompt the user to enter valid username
6	Login_Invalid_All	"LOGIN" button is active	Enter incorrect password and incorrect username	Must display an error message and prompt the user to enter valid username and password
7	Login_Valid_All	"LOGIN" button is active	Enter correct password and correct username	Login Successful and user is re-directed to his/her dashboard

B. Test Cases for "Sign Up":-

Following table depicts the test cases for manual testing of the Sign Up page to ensure optimal performance and security in the system.

Table 4.8: Test Cases for "Sign Up"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT
1	Signup_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter all details
2	Signup_Name_Blank	"SUBMIT" button is active	Enter all the details and leave the name field blank	Must prompt the user to enter name
3	Signup_Email_Blank	"SUBMIT" button is active	Enter all the details and leave the email field blank	Must prompt the user to enter email
4	Signup_Address_Blank	"SUBMIT" button is active	Enter all the details and leave the address field blank	Must prompt the user to enter address
5	Signup_Phone_Blank	"SUBMIT" button is active	Enter all the details and leave the phone field blank	Must prompt the user to enter phone number
6	Signup_Existing_Email_All	"SUBMIT" button is active	Enter all the details correctly	Must display a message telling the user that the email id already exists
7	Signup_Valid_All	"SUBMIT" button is active	Enter all the details correctly	Sign up successful and the user is re-directed to the login page

C. Test Cases for "Contact Us":-

Following table depicts the test cases for manual testing of the Contact Us page to ensure optimal performance and security in the system.

Table 4.9: Test Cases for "Contact Us"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT
1	Contact_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to fill form
2	Contact_Name_Blank	"SUBMIT" button is active	Enter all the fields and leave the name field blank	Must prompt the user to enter name
3	Contact_Email_Blank	"SUBMIT" button is active	Enter all the fields and leave the email field blank	Must prompt the user to enter email
4	Contact_Phone_Blank	"SUBMIT" button is active	Enter all the fields and leave the phone field blank	Must prompt the user to enter phone number
5	Contact_Query_Blank	"SUBMIT" button is active	Enter all the fields and leave the query field blank	Must prompt the user to enter query
6	Contact_Valid_All	"SUBMIT" button is active	Enter the all fields correctly	Display query submitted and re-direct to contact us page

5. CODE SNIPPETS

5.1 HARDWARE PROGRAMMING

The following is a code snippet of MCU for collecting sensor data, appending all the data as a single byte code and pushing that to server.

Fig 5.1: Code Snippet for Arduino to program MCU to create a connection with the server in cloud

5.2 SERVER SIDE PROGRAMMING

The following is a code snippet of Server running in cloud for collecting sensor data from MCU and pushing this data in MySql database.

```
import java.net.Socket;
import java.util.Arrays;
import java.io.*;
import java.sql.DriverManager;
import java.sql.Connection;

public class wsn{
    public static void main(String[] args)throws IOException  {
        System.out.println("start");
        ServerSocket ss=new ServerSocket(4000);

        while(true){
            java.util.Date date= new java.util.Date();
             Socket s=ss.accept();
             DataInputStream din=new DataInputStream(s.getInputStream());
             //System.out.println(din.readUTF());
             BufferedReader reader=new BufferedReader(new InputStreamReader(din));

             String msg=reader.readLine();

             System.out.println("MSG:  "+msg);
             String[] msgArray = msg.trim().split("\\s*,\\s*");
             System.out.println();
             System.out.println(Arrays.toString(msg.getBytes()));
             for (String object: msgArray) {
                 System.out.println(object);
                }

             try{
                //Writer output = null;
                PrintWriter file = new PrintWriter("log.txt");

                file.close();
            }

            finally{

                if(reader != null)
                    reader.close();
            }
```

Fig 5.2: Code Snippet for server running JavaSocket to get data from MCU

6. TESTING

PHASE-1:

Following tables depict the test results and their remarks on the basis of phase – 1 manual testing of various test case designs.

A. Testing for "Login":-

Table 6.1: Phase – 1 Testing for "Login"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Login_All_Blank	"LOGIN" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter username and password	Error message displayed	Successful
2	Login_PWD_Blank	"LOGIN" button is active	Enter the username and leave the password field blank	Must prompt the user to enter password	Error message was not displayed	Un-successful
3	Login_User_Blank	"LOGIN" button is active	Enter the password and leave the username field blank	Must prompt the user to enter username	Error message was not displayed	Un-successful
4	Login_Invalid_PWD	"LOGIN" button is active	Enter the correct username and incorrect password	Must display an error message and prompt the user to enter valid password	Error message was not displayed	Un-successful
5	Login_Invalid_User	"LOGIN" button is active	Enter the correct password and incorrect username	Must display an error message and prompt the user to enter valid username	Error message was not displayed	Un-successful
6	Login_Invalid_All	"LOGIN" button is active	Enter incorrect password and incorrect username	Must display an error message and prompt the user to enter valid username and password	Error message was not displayed	Un-successful
7	Login_Valid_All	"LOGIN" button is active	Enter correct password and correct username	Login Successful and user is re-directed to his/her dashboard	Successful login	Successful

B. Testing for "Sign Up":-

Table 6.2: Phase – 1 Testing for "Sign Up"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Signup_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter all details	Error message displayed	Successful
2	Signup_Name_Blank	"SUBMIT" button is active	Enter all the details and leave the name field blank	Must prompt the user to enter name	Error message was not displayed	Un-successful
3	Signup_Email_Blank	"SUBMIT" button is active	Enter all the details and leave the email field blank	Must prompt the user to enter email	Error message was not displayed	Un-successful
4	Signup_Address_Blank	"SUBMIT" button is active	Enter all the details and leave the address field blank	Must prompt the user to enter address	Error message was not displayed	Un-successful
5	Signup_Phone_Blank	"SUBMIT" button is active	Enter all the details and leave the phone field blank	Must prompt the user to enter phone number	Error message was not displayed	Un-successful
6	Signup_Existing_Email_All	"SUBMIT" button is active	Enter all the details correctly	Must display a message telling the user that the email id already exists	Error message was not displayed	Un-successful
7	Signup_Valid_All	"SUBMIT" button is active	Enter all the details correctly	Sign up successful and the user is re-directed to the login page	Sign up successful	Successful

C. Testing for "Contact Us":-

Table 6.3: Phase – 1 Testing for "Contact Us"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Contact_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to fill form	Error message displayed	Successful
2	Contact_Name_Blank	"SUBMIT" button is active	Enter all the fields and leave the name field blank	Must prompt the user to enter name	Error message was not displayed	Un-successful
3	Contact_Email_Blank	"SUBMIT" button is active	Enter all the fields and leave the email field blank	Must prompt the user to enter email	Error message was not displayed	Un-successful
4	Contact_Phone_Blank	"SUBMIT" button is active	Enter all the fields and leave the phone field blank	Must prompt the user to enter phone number	Error message was not displayed	Un-successful
5	Contact_Query_Blank	"SUBMIT" button is active	Enter all the fields and leave the query field blank	Must prompt the user to enter query	Error message was not displayed	Un-successful
6	Contact_Valid_All	"SUBMIT" button is active	Enter the all fields correctly	Display query submitted and re-direct to contact us page	Query sub-mission successful	Successful

PHASE-2:

Following tables depict the test results and their remarks on the basis of phase – 2 manual testing of various test case designs.

A. Testing for "Login":-

Table 6.4: Phase – 2 Testing for "Login"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Login_All_Blank	"LOGIN" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter username and password	Error message displayed	Successful
2	Login_PWD_Blank	"LOGIN" button is active	Enter the username and leave the password field blank	Must prompt the user to enter password	Error message displayed	Successful
3	Login_User_Blank	"LOGIN" button is active	Enter the password and leave the username field blank	Must prompt the user to enter username	Error message displayed	Successful
4	Login_Invalid_PWD	"LOGIN" button is active	Enter the correct username and incorrect password	Must display an error message and prompt the user to enter valid password	Error message displayed	Successful
5	Login_Invalid_User	"LOGIN" button is active	Enter the correct password and incorrect username	Must display an error message and prompt the user to enter valid username	Error message displayed	Successful
6	Login_Invalid_All	"LOGIN" button is active	Enter incorrect password and incorrect username	Must display an error message and prompt the user to enter valid username and password	Error message was not displayed	Un-successful
7	Login_Valid_All	"LOGIN" button is active	Enter correct password and correct username	Login Successful and user is re-directed to his/her dashboard	Successful login	Successful

B. Testing for "Sign Up":-

Table 6.5: Phase – 2 Testing for "Sign Up"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Signup_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter all details	Error message displayed	Successful
2	Signup_Name_Blank	"SUBMIT" button is active	Enter all the details and leave the name field blank	Must prompt the user to enter name	Error message displayed	Successful
3	Signup_Email_Blank	"SUBMIT" button is active	Enter all the details and leave the email field blank	Must prompt the user to enter email	Error message displayed	Successful
4	Signup_Address_Blank	"SUBMIT" button is active	Enter all the details and leave the address field blank	Must prompt the user to enter address	Error message displayed	Successful
5	Signup_Phone_Blank	"SUBMIT" button is active	Enter all the details and leave the phone field blank	Must prompt the user to enter phone number	Error message was not displayed	Un-successful
6	Signup_Existing_Email_All	"SUBMIT" button is active	Enter all the details correctly	Must display a message telling the user that the email id already exists	Error message was not displayed	Un-successful
7	Signup_Valid_All	"SUBMIT" button is active	Enter all the details correctly	Sign up successful and the user is re-directed to the login page	Sign up successful	Successful

C. Testing for "Contact Us":-

Table 6.6: Phase – 2 Testing for "Contact Us"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Contact_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to fill form	Error message displayed	Successful
2	Contact_Name_Blank	"SUBMIT" button is active	Enter all the fields and leave the name field blank	Must prompt the user to enter name	Error message displayed	Successful
3	Contact_Email_Blank	"SUBMIT" button is active	Enter all the fields and leave the email field blank	Must prompt the user to enter email	Error message displayed	Successful
4	Contact_Phone_Blank	"SUBMIT" button is active	Enter all the fields and leave the phone field blank	Must prompt the user to enter phone number	Error message was not displayed	Un-successful
5	Contact_Query_Blank	"SUBMIT" button is active	Enter all the fields and leave the query field blank	Must prompt the user to enter query	Error message was not displayed	Un-successful
6	Contact_Valid_All	"SUBMIT" button is active	Enter the all fields correctly	Display query submitted and re-direct to contact us page	Query sub-mission successful	Successful

PHASE-3:

Following tables depict the test results and their remarks on the basis of phase – 3 manual testing of various test case designs.

A. Testing for "Login":-

Table 6.7: Phase – 3 Testing for "Login"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Login_All_Blank	"LOGIN" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter username and password	Error message displayed	Successful
2	Login_PWD_Blank	"LOGIN" button is active	Enter the username and leave the password field blank	Must prompt the user to enter password	Error message displayed	Successful
3	Login_User_Blank	"LOGIN" button is active	Enter the password and leave the username field blank	Must prompt the user to enter username	Error message displayed	Successful
4	Login_Invalid_PWD	"LOGIN" button is active	Enter the correct username and incorrect password	Must display an error message and prompt the user to enter valid password	Error message displayed	Successful
5	Login_Invalid_User	"LOGIN" button is active	Enter the correct password and incorrect username	Must display an error message and prompt the user to enter valid username	Error message displayed	Successful
6	Login_Invalid_All	"LOGIN" button is active	Enter incorrect password and incorrect username	Must display an error message and prompt the user to enter valid username and password	Error message displayed	Successful
7	Login_Valid_All	"LOGIN" button is active	Enter correct password and correct username	Login Successful and user is re-directed to his/her dashboard	Successful login	Successful

B. Testing for "Sign Up":-

Table 6.8: Phase – 3 Testing for "Sign Up"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Signup_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to enter all details	Error message displayed	Successful
2	Signup_Name_Blank	"SUBMIT" button is active	Enter all the details and leave the name field blank	Must prompt the user to enter name	Error message displayed	Successful
3	Signup_Email_Blank	"SUBMIT" button is active	Enter all the details and leave the email field blank	Must prompt the user to enter email	Error message displayed	Successful
4	Signup_Address_Blank	"SUBMIT" button is active	Enter all the details and leave the address field blank	Must prompt the user to enter address	Error message displayed	Successful
5	Signup_Phone_Blank	"SUBMIT" button is active	Enter all the details and leave the phone field blank	Must prompt the user to enter phone number	Error message displayed	Successful
6	Signup_Existing_Email_All	"SUBMIT" button is active	Enter all the details correctly	Must display a message telling the user that the email id already exists	Error message displayed	Successful
7	Signup_Valid_All	"SUBMIT" button is active	Enter all the details correctly	Sign up successful and the user is re-directed to the login page	Sign up successful	Successful

C. Testing for "Contact Us":-

Table 6.9: Phase – 3 Testing for "Contact Us"

SL. NO	TEST CASE NAME	PRE – CONDITION	TEST PROCEDURE	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1	Contact_All_Blank	"SUBMIT" button is active	All the fields are kept blank	Must display an error message and prompt the user to fill form	Error message displayed	Successful
2	Contact_Name_Blank	"SUBMIT" button is active	Enter all the fields and leave the name field blank	Must prompt the user to enter name	Error message displayed	Successful
3	Contact_Email_Blank	"SUBMIT" button is active	Enter all the fields and leave the email field blank	Must prompt the user to enter email	Error message displayed	Successful
4	Contact_Phone_Blank	"SUBMIT" button is active	Enter all the fields and leave the phone field blank	Must prompt the user to enter phone number	Error message displayed	Successful
5	Contact_Query_Blank	"SUBMIT" button is active	Enter all the fields and leave the query field blank	Must prompt the user to enter query	Error message displayed	Successful
6	Contact_Valid_All	"SUBMIT" button is active	Enter the all fields correctly	Display query submitted and re-direct to contact us page	Query sub-mission successful	Successful

7. FUTURE WORKS

Although the project carried out was complete and promising a few more add-ons like embedding the system to navigate all the features using a smart phone would come in handy. Moreover, centralized database could be conceptualized for different agricultural products for research purposes. Also, predictive analysis for the plant growth or failure along with alert generation would be implemented in this project.

8. CONCLUSION

Through this project, we have proposed a framework to predict the expected growth rate of the plant and the time when the plant is fully matured. The entire process is divided into three major parts information gathering, analyzing and predicting, which will help us in scaling up mass production of the agricultural products and catching up with the demand in synthetic environment with a cost effective approach.

9. REFERENCES

1. Farmers Portal, (Govt. of India): http://farmer.gov.in/

2. Ethics Agrotech Private Limited: http://www.ethicsagrotech.in/

3. Agri Farming: http://www.agrifarming.in/

4. M. Outanoute, A. Lachhab, A. Ed-dahhak, A. Selmani, M. Guerbaoui and B. Bouchikhi, "A neural network dynamic model for temperature and relative humidity control under greenhouse", (RAWSN), pp. 6-11, May. 2015.

5. M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1–1, 2016.

6. D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, Jan. 2015.