**FarmMD**

|  |
| --- |
| A Project Report Presented to  The Faculty of the College of Engineering |
| San Jose State University In Partial Fulfillment Of the Requirements for the Degree **Master of Science in Software Engineering** |

|  |
| --- |
| By |
| Sachet Hegde, Shital Kadam, Vidya Khadsare, Pavani Vellal |
| May/2017 |

|  |
| --- |
| Copyright © 2017 |
| Sachet Hegde, Shital Kadam, Vidya Khadsare, Pavani Vellal |
| ALL RIGHTS RESERVED |

|  |
| --- |
| APPROVED |
|  |
| Prof. Rakesh Ranjan, Project Advisor |
|  |
|  |
| Prof. Dan Harkey, Director, MS Software Engineering |
|  |
|  |
| Xiao Su, Department Chair |

ABSTRACT

**FarmMD**

By Sachet Hegde, Shital Kadam, Vidya Khadsare, Pavani Vellal

Today the world is battling with the drought condition. To secure our future, we need ways to conserve the water in our daily life. Growers, be they agricultural, institutional or residential, consume a substantial amount of water and power resources every single day for irrigation. Efficient irrigation depends on upon many factors like weather conditions, soil pH, salinity, moisture, temperature and plant type. Close to 50% of water used outdoors is wasted due to inefficient resource management.

There are many existing systems which ensure efficient water and energy consumptions for irrigation, but these systems have many inherent problems. One issue is that most systems are hard-wired communications networks creating installation difficulties.

Growers also need systems that will have continual connectivity and eliminate any security breach. Another problem is no current system is designed to provide the number of the sensors required for adequate, instantaneous readings for moisture, pH, and salinity for agriculture, institutional and residential horticultural needs.

In this project, we aim to develop a sensor system which will provide a secure network while taking intelligent decisions to maintain the irrigation schedule that saves the water and energy. The end user can monitor functionalities of the system from anywhere. It provides the graphical representation of the resources used and saved over a period of time including details of the soil pH level, salinity, and moisture. Our work will help growers choose judicious watering schedules, suitable plants and fertilizers for the soil which leads to increase in harvest and greater awareness about their cultivation.

|  |
| --- |
| Acknowledgments |
| The authors are deeply indebted to Professor Rakesh Ranjan for his invaluable  comments and assistance in the preparation of this study |

Table of Contents

Contents

[Project Overview 10](#_Toc477882449)

[Introduction 10](#_Toc477882450)

[Proposed Areas of Study and Academic Contribution 10](#_Toc477882451)

[Current State of the Art 11](#_Toc477882452)

[Chapter 2. Project Architecture 13](#_Toc477882453)

[o **Introduction** 13](#_Toc477882454)

[o **Architecture Subsystems** 16](#_Toc477882455)

[ **Interactive mobile user interface Subsystem:** 16](#_Toc477882456)

[o **Login:** 16](#_Toc477882457)

[o **QR Code Scanning:** 16](#_Toc477882458)

[o **TextToSpeech:** 16](#_Toc477882460)

[o **Notification:** 16](#_Toc477882461)

[ **Interactive Web user interface Subsystem:** 17](#_Toc477882462)

[o **Login:** 17](#_Toc477882463)

[o **Sensor Node Topology:** 17](#_Toc477882464)

[o **Sensor Data Chart:** 17](#_Toc477882465)

[ **Backend Subsystem:** 17](#_Toc477882466)

[o **REST APIs:** 17](#_Toc477882467)

[o **Node-Red:** 17](#_Toc477882468)

[o **CouchDB:** 17](#_Toc477882469)

[ **Hardware Subsystem:** 18](#_Toc477882470)

[o **Sensors:** 18](#_Toc477882471)

[o **Arduino Due Board:** 18](#_Toc477882472)

[o **LinkLabs Router:** 18](#_Toc477882473)

[Chapter 3. Technology Descriptions 19](#_Toc477882474)

[Client Technologies 19](#_Toc477882475)

[Web Application: Web application is developed using AngularJS, ExpressJS, HTML, CSS, Bootstrap and NodeJS. 19](#_Toc477882476)

[Middle-Tier Technologies 19](#_Toc477882477)

[Data-Tier Technologies 19](#_Toc477882478)

[Hardware-Tier Technologies 19](#_Toc477882479)

[Different sensors are used including temperature, moisture, PH and salinity. 20](#_Toc477882480)

[Chapter 4. Project Design 21](#_Toc477882481)

[6.1. UML Diagrams 21](#_Toc477882482)

[6.1.1. Class Diagram 21](#_Toc477882483)

[6.1.2. Sequence Diagrams 22](#_Toc477882484)

[6.2. Mockups 23](#_Toc477882485)

[6.2.1. Web application 23](#_Toc477882486)

[6.2.2. Mobile application 26](#_Toc477882487)

[6.3. Database Entity Diagrams 30](#_Toc477882488)

[6.4. Hardware Block Diagrams 31](#_Toc477882489)

[6.5. System Block Diagrams 33](#_Toc477882490)

[Chapter 5. Project Implementation 35](#_Toc477882491)

[Project implementation scope 35](#_Toc477882492)

[Client Implementation 35](#_Toc477882493)

[Middle-Tier Implementation 36](#_Toc477882494)

[Data-Tier Implementation 37](#_Toc477882495)

[Implementation methodology 37](#_Toc477882496)

[Project implementation issues and resolutions 37](#_Toc477882497)

[Used tools and technologies 38](#_Toc477882498)

[Chapter 6. Testing and Verification 40](#_Toc477882499)

[Chapter 7. Performance and Benchmarks 44](#_Toc477882500)

[Chapter 8. Deployment, Operations, Maintenance 45](#_Toc477882501)

[Chapter 9. Summary, Conclusions, and Recommendations 47](#_Toc477882502)

[Summary 47](#_Toc477882503)

[Conclusions 47](#_Toc477882504)

[Recommendations for Further Research 47](#_Toc477882505)

[Glossary 48](#_Toc477882506)

[References 1](#_Toc477882507)

List of Figures

|  |  |
| --- | --- |
| Project Architecture | 5 |
| User Interaction with Mobile Application | 13 |
| Sensor Hub Interaction with link labs router | 13 |
| login Screen | 14 |
| dashboard | 14 |
| farm field map | 15 |
| Attribute details | 15 |
| login screen | 16 |
| Home Screen | 16 |
| Scan Screen | 17 |
| Registration Screen | 17 |
| Save Screen | 18 |
| Interact Screen | 18 |
| Weather Screen | 19 |
| Soil Health Screen | 19 |
| Arduino to Link-lab eval boards connections | 22 |
| Arduino to Link-lab eval boards connections | 23 |
| System Block Diagram | 24 |
| RESTful api | 25 |

# Project Overview

## Introduction

As another era of well-informed producers assumes liability for their own water utilize, they need "intelligent" system to oversee it. This product lives up to their desires by empowering versatile information applications for landowners and farming experts’ particular to the soil moisture, saltiness, and pH of their property. Our unique communication system won't associate with the landowner's network access. In this way, our system will be a protected other option to the home/business organize. The cloud will deal with the investigation and calculation execution and convey activities back to the center point. Huge amount of data is collected from every sensor node. The data is then transformed into a clean format and stored in a database as a key value pair. Goal is to analyze the data and build an intelligent system that can predict the soil health over different conditions.

## Proposed Areas of Study and Academic Contribution

Area of Study

Our project needed us to understand the IoT system architecture. We also needed to study the soil health parameters. To implement the idea of the project we had to understand how soil health parameters affect the growth of the crop. 2 Studying different hardware components was a very important part of this project. After considering different hardware we choose which are more beneficial for our project. We also studied data analytics in order to analyze real-time sensor data.

Academic Contribution

Our project contributes to the better analysis of soil health. It also contributes to saving water and energy consumption required in agriculture by providing irrigation automation.

## Current State of the Art

FarmMD is a one of a kind Multi-Sensor Intelligent Agricultural IOT System that is capable of monitoring the soil health and the atmosphere around it to be able to provide just enough, just in time water to the crop and maintain a conducive environment for a good yield. It harnessing the power of sensor networks, IOT and Artificial Intelligence to revolutionize the way growers approach agriculture. It enables the growers to get insights into their crop and help them to make informed decisions to improve yield. The system uses artificial intelligence and machine learning to monitor the health of the crop and enable the growers to effectively water the plants and hence save the precious water resources. It also provides information about the soil health and the 3 effectiveness of the fertilizers used, by using sensors such as the pH, salinity and moisture sensors. The system leverages the power of LPWAN to communicate data from multiple sensors over a wide area to the cloud with a low bit rate hence enabling longer battery life of the sensor clusters. The data from the system is extrapolated in a graph on the web application to be able to identify patterns that were once only a mystery to the user. We also provide the capability for other 3rd party system to collaborate with our system and obtain the intelligence from our system. The system shall also be designed to recommend the amount of water required for the crops, advisable pH levels, suitable crops to grow, based on a number of parameters such as soil, atmospheric conditions and user's behavioral patterns. Our project stands out from a lot of other systems that are already available in many ways such as follows: It uses a unique communication system that will not connect to the land owner’s internet service. Therefore, our system will be a secure alternative to the home/business network making it easy to install and use the system. The system is equipped with multiple different sensors to precariously monitor all the factors that affect farming. 4 Performs analytics on the historical and streaming data to provide suggestions for choosing crop suitable for land which is a major issue in more of the farm lands as they are unaware of these details. The system comes with an easy to use Mobile and Web application that uses sensor data visualization to provide information gathered and analyzed by the system. The application is made very easy to set up as it registers the system and makes connection with the mobile just by scanning a QR code. The Link-labs hub that we are using in our project uses Low-power WAN Technology which is specially designed for Industrial IOT Device to Device communication. It has a wider range of connectivity that spread across miles with lower power consumption and is more economical to use than mobile networks. Focuses on water conditions. Large amount of data receives from multiple systems can improve our capability to predict drought.

# Project Architecture

* + **Introduction**

FarmMD is an IOT based project, it contains the hardware as well as software components. Below diagram depicts the high-level architecture of the project.



*Figure 1: System Architecture*

FarmMD is a system for customers including farmers, industrial growers, backyard gardeners and many more. In the process of the system setup growers will have to install the sensor nodes in the various parts of the farm. These nodes to obtain the soil health data. This data can be accessed from the FarmMD web application as well as mobile application. System architecture is composed of the hardware setup, cloud server, web application, and mobile application.

The project architecture contains following components:

* **Hardware Setup**
  + Hardware setup includes the sensor nodes, link lab’s evaluation board, gateway, and Arduino due board. Gateway is designed to access the user’s Wi-Fi system.
  + There are many sensors in the single node including temperature, moisture, PH, salinity. These sensors will be placed under the soil to gather the soil health data.
  + Soil health data will be collected and saved into the Cloudant non-relational database. To leverage this process Arduino due microcontroller is used. All the sensors are connected to the Arduino due board through LinkLabs’ evaluation board.
* **Cloud Server Setup**
  + We have implemented the Node-Red Flow on the IBM cloud. The sensor data is fetched from LinkLabs’ cloud database into the Node-Red flow.
  + Real-time weather data is obtained from IBM’s weather service. This weather data is paired with the sensor node’s location so system can provide the weather data to user.
  + The database used for the system is also hosted on the IBM’s cloud platform. We are using the Cloudant platform which uses the CouchDB. CouchDB is nonrelation, document based database. This database contains the user credentials, sensor node id, and sensor real-time data.
* **Web Application**
  + We have hosted FarmMD web application on the IBM’s cloud platform.
  + This web application communicates to cloud server using REST APIs.
  + Web application provides the login and registration for the system user.
  + Web application provides the live streaming sensor data to the end user.
  + Web application contains the sensor node topology, it provides the pictorial setup of the sensor installation in the farm.
* **Mobile Application**
  + FarmMD Mobile application is developed using the android platform.
  + Mobile application is designed using Android Material Design.
  + Login module authenticates the user with the session management.
  + User must provide the device id while login into the system.
  + Mobile application communicates with cloud server using REST APIs.
  + The mobile application provides the live streaming sensor data to the end user.
  + User needs to scan the sensor node’s QRCode to register the sensor node into the system. Sensor node’s id and the location of the node will be registered into the system.
  + User gets the feature of communication. User can ask the question about the sensor data or the weather and mobile application will provide the real-time answer.

* + **Architecture Subsystems**

FarmMD architecture can be further divided into different subsystems. There are mainly four subsystems for the FarmMD. Each subsystem is explained in detail in the following section.

* **Interactive mobile user interface Subsystem:**
  + **Login:**

Login feature allows existing user to login into the. User must register using the web application. Registration will require the device id and user credentials. After successful registration user can login into the mobile application.

* + **QR Code Scanning:**

User will get one device which have unique id. Each device can have multiple sensor nodes Every sensor node is associated with unique QRCode. QRCode is embed unique id. User can use mobile application to scan QRCode and register it to the database.

* + **SpeechToText:**

This is very useful feature to allow user to interact with mobile application by using speech. User can ask question related to soil health and weather data and. Mobile application will get that input as speech and convert into text and sent it to the system for further processing. Also, this text will be printed on the screen.

* + **TextToSpeech:**

After the system is ready with the processed answer this feature will be used. It will convert text answer coming from the backend to speech. User can hear the answer for the asked question.

* + **Notification:**

This feature provides the notifications to the user of any events in the system. This includes notifications about hardware failures and sensor values running out of bound.

* **Interactive Web user interface Subsystem:**
  + **Login:**

Login feature allows existing user to login into the web application. New user can register and create a new account. Registration will require the device id and user credentials. After successful registration user can login into the web application.

* + **Sensor Node Topology:**

Sensor node topology shows the sensor node network. It shows the connection between all the sensor node and sensor hub.

* + **Sensor Data Chart:**

Sensor data chart shows historical data in form of graph. It shows time on X axis and Humidity, temperature, salinity and moisture values on Y axis. User can get the real-time sensor data.

* **Backend Subsystem:**
  + **REST APIs:**

We created the web server and REST API’s. These REST APIs are used as a communication mechanism between mobile / web application and the database which provide the sensor data.

* + **Node-Red:**

Node-Red performs analytics on the data coming from in the sensors. When user makes a request via web/mobile application it is send to Node-Red first. Node-Red performs analytics and sends the results back to web/mobile application.

* + **CouchDB:**

We are using CouchDB which is hosted on the Cloudant platform. We have created the database server which contains multiple databases for the system. The database contains the real time sensor data, and user data.

* **Hardware Subsystem:**
  + **Sensors:**

Sensors used in the system are Ph, temperature, moisture and salinity. Which can measure the soil health reading.

* + **Arduino Due Board:**

All sensors are connected to the Arduino Due Board through LinkLabs evaluation board.

* + **LinkLabs Router:**

LinkLabs router sends the real-time sensor data to Link Lab cloud database.

# Technology Descriptions

FarmMD development includes various technologies and development tools. Technologies can be highly divided into the four segments. Following is the high-level description of the used technologies.

## 

## Client Technologies

## Web Application: Web application is developed using AngularJS, ExpressJS, HTML, CSS, Bootstrap and NodeJS.

* **Mobile Application**: To develop the mobile application we have used Android technology including Material Design. IBM’s conversion API is used to develop the chat bot for the FarmMD.

## 

## Middle-Tier Technologies

* Node-Red Flow is designed to serve as the middle layer between client layer and Link Lab’s Cloud database.
* NodeJS serve as the middle tire for the web application.
* CISCO IoT Framework is used to give the pictorial represent the different sensor nodes.
* IBM’s cloud platform is used to host the middle-tire.

## 

## Data-Tier Technologies

* FarmMD used the non-relational, document based database called as CouchDB.
* We have used IBM’s Cloudant database technology which used the CouchDB technology.

## Hardware-Tier Technologies

* The Link Labs starter kit is used which includes an evaluation board, a router that uses the LoRa technology.
* Arduino Duo is used as the microcontroller.

## Different sensors are used including temperature, moisture, PH and salinity.

# Project Design

## 6.1. UML Diagrams

This section provides the UML diagrams representing the system design.

### 6.1.1. Class Diagram

../../../../Downloads/Class%20Diagram_295%20(1).p

Figure 2 System Class Diagram

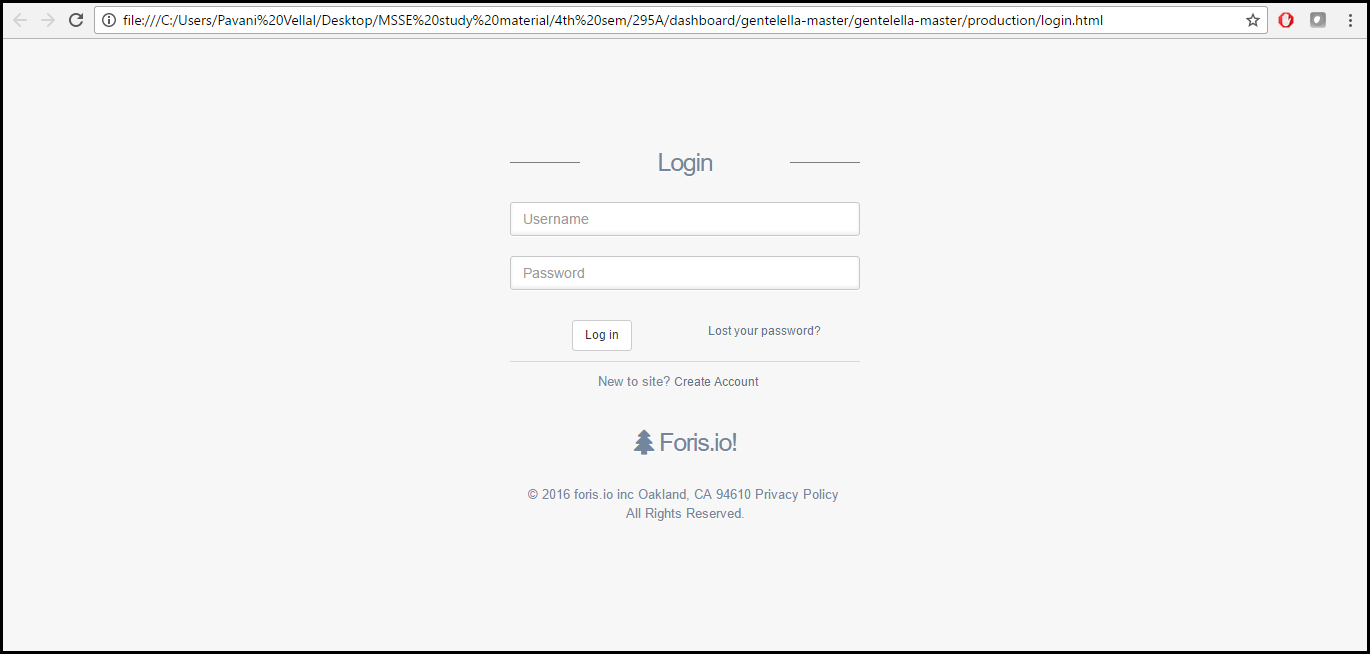
### ../../../../Downloads/Sequence%20Diagram%20(2).p6.1.2. Sequence Diagrams

../../../../Downloads/Copy%20of%20Sequence%20DiagramFigure 3 User Interaction with Mobile Application

## 6.2. Mockups

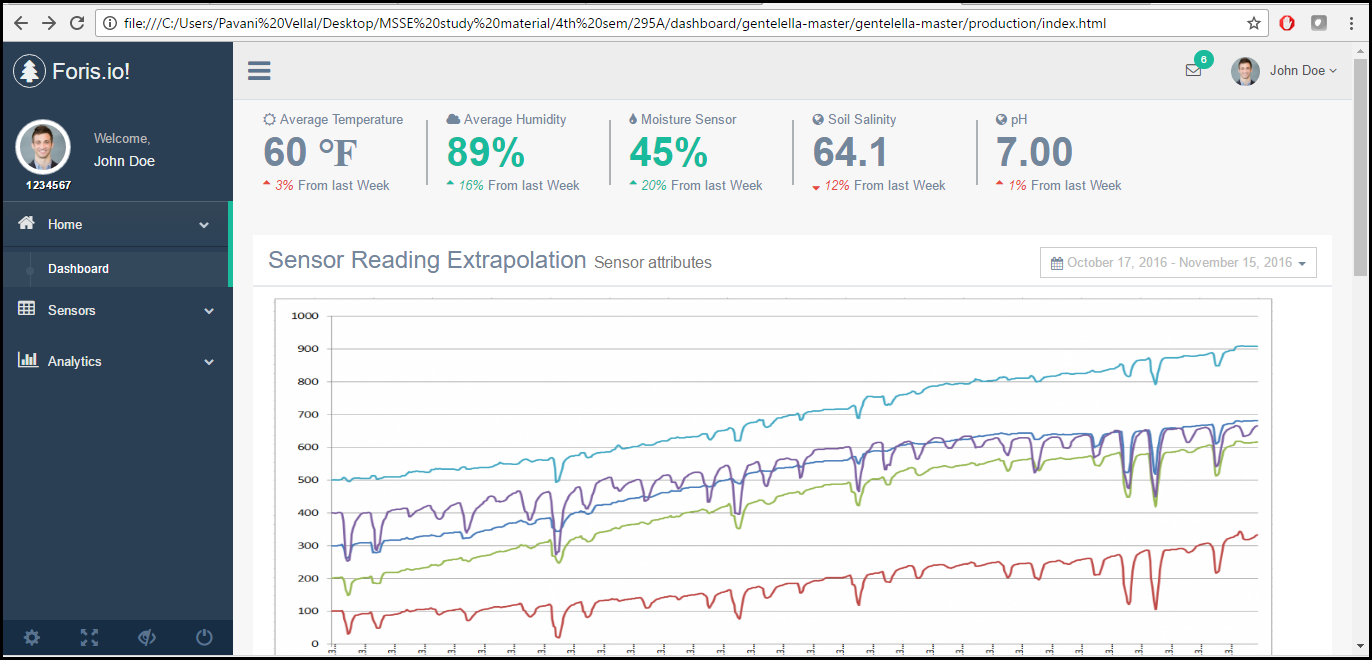
### 6.2.1. Web application

Login Screen

 login screen

Dashboard

The main graph in the dashboard shows a graph of the variation of different sensor data over time

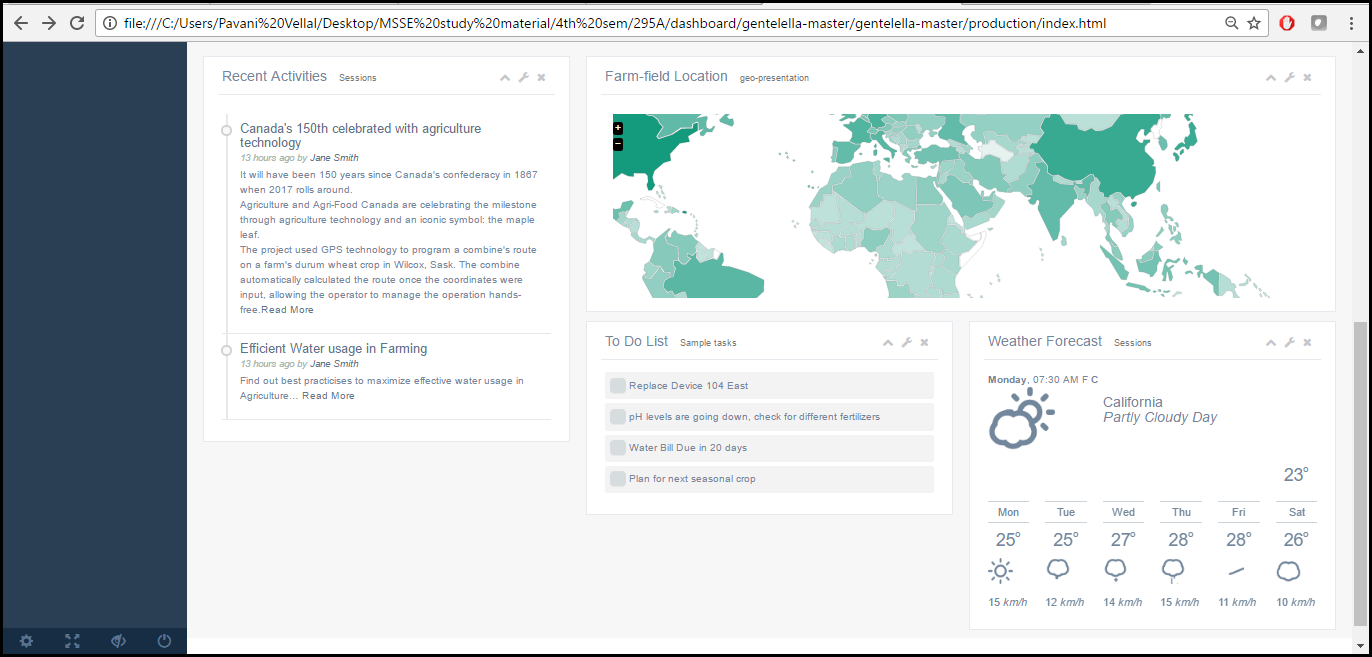


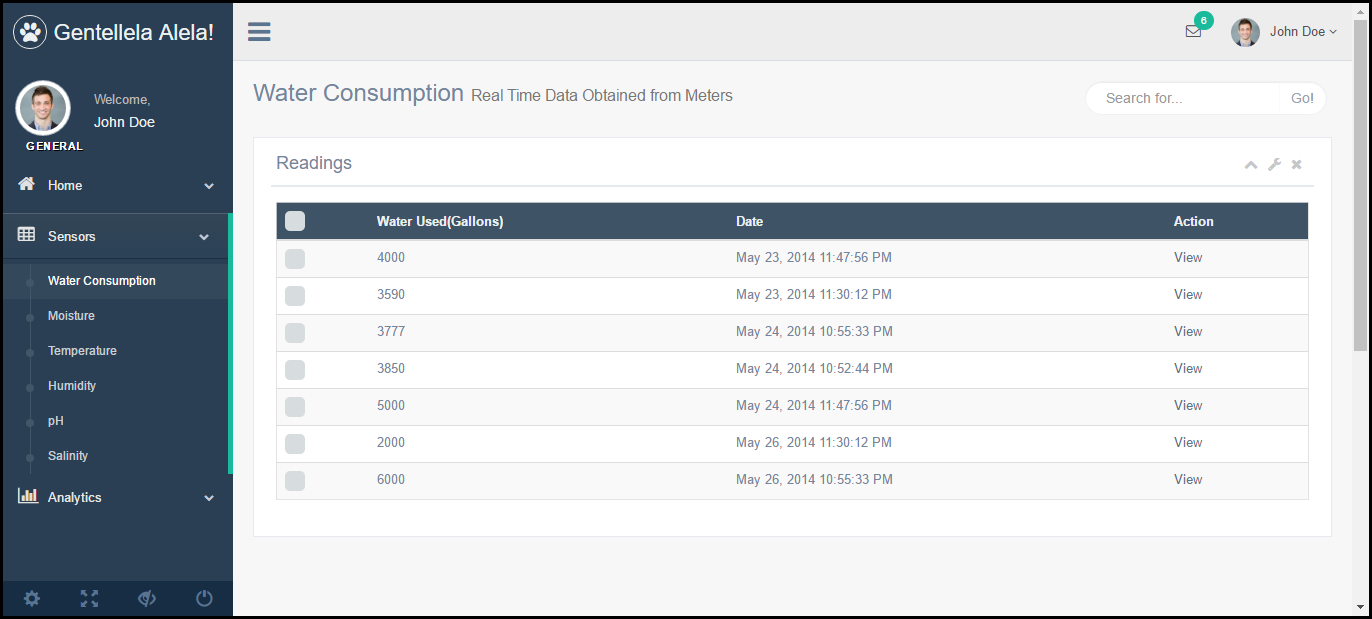
The farm field location indicates al the areas where the sensor is places.

The Recent activities gives new updates on efficient farming and suggestion to improve farming practices

The To-Do List indicates the list of things to be done for the system

And the weather forecast for the week is shown as it is an integral part in deciding whether or not to water crops.



Each Attribute Reading in Detail

### 6.2.2. Mobile application

Mobile application provides user the sensor data. User needs to download the application and install it on the android mobile device. User needs to login into the system using the credentials provided while purchasing the system. If user forgets the password, he can retrieve it using the “Forgot Password” option. Recovered password will be emailed to the user.

On the home screen user will get two options to choose

1. Register Sensor

2. Interact

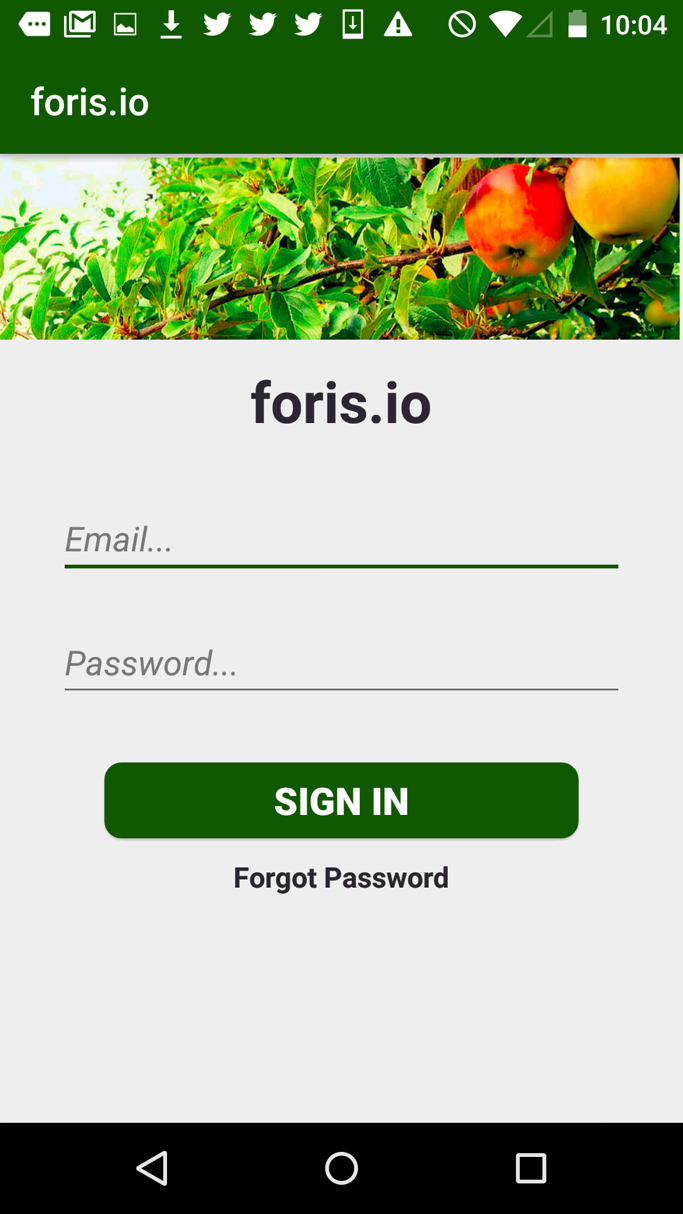
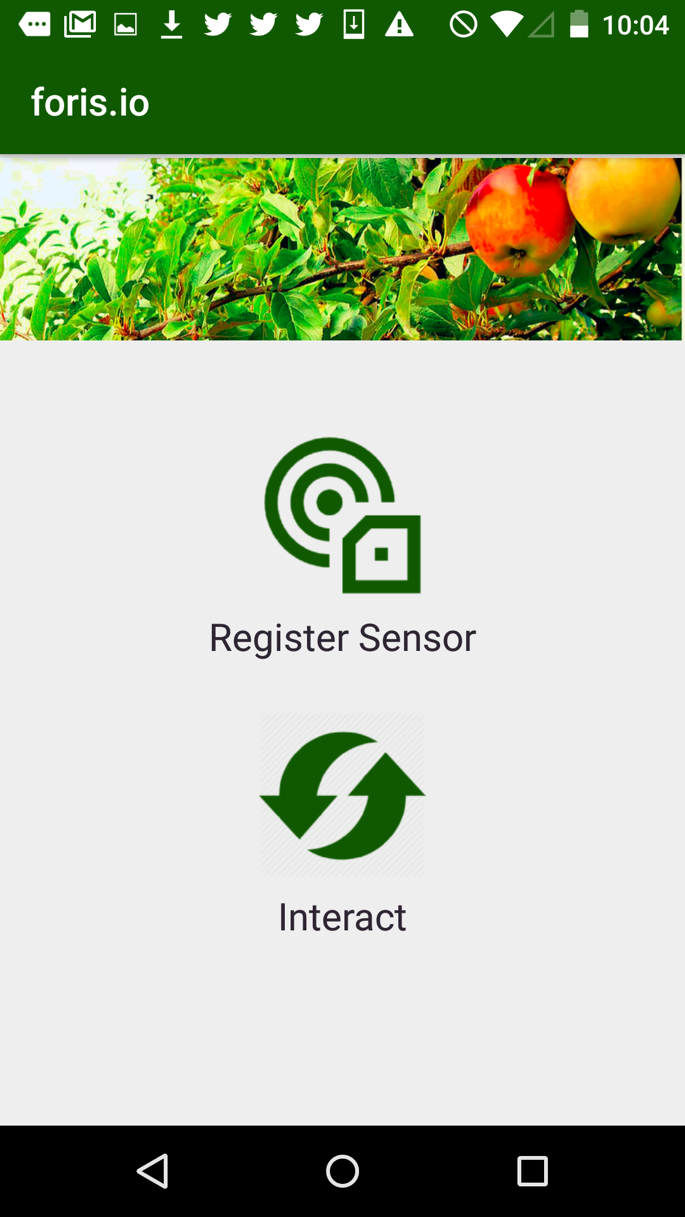


Figure 4 Login Screen Figure 5 Home Screen

When user wants to register a new sensor choose the “Register Sensor” option. User can then scan the QRCode by tapping the “SCAN” button. It will redirect to the scanner application then user can scan the code of the sensor. Once the scanning is to register it use the “REGISTER” button. Then the sensor will be successfully register in the cloud account. It contains the sensor id, latitude and longitude of the sensor location.

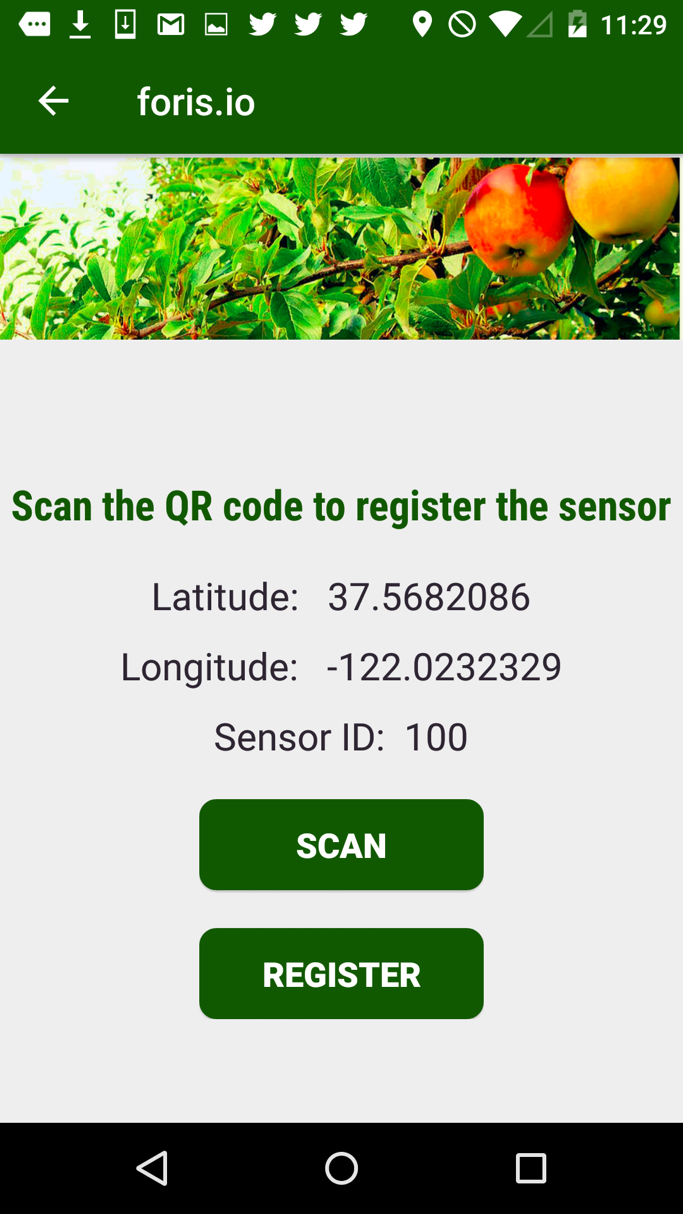
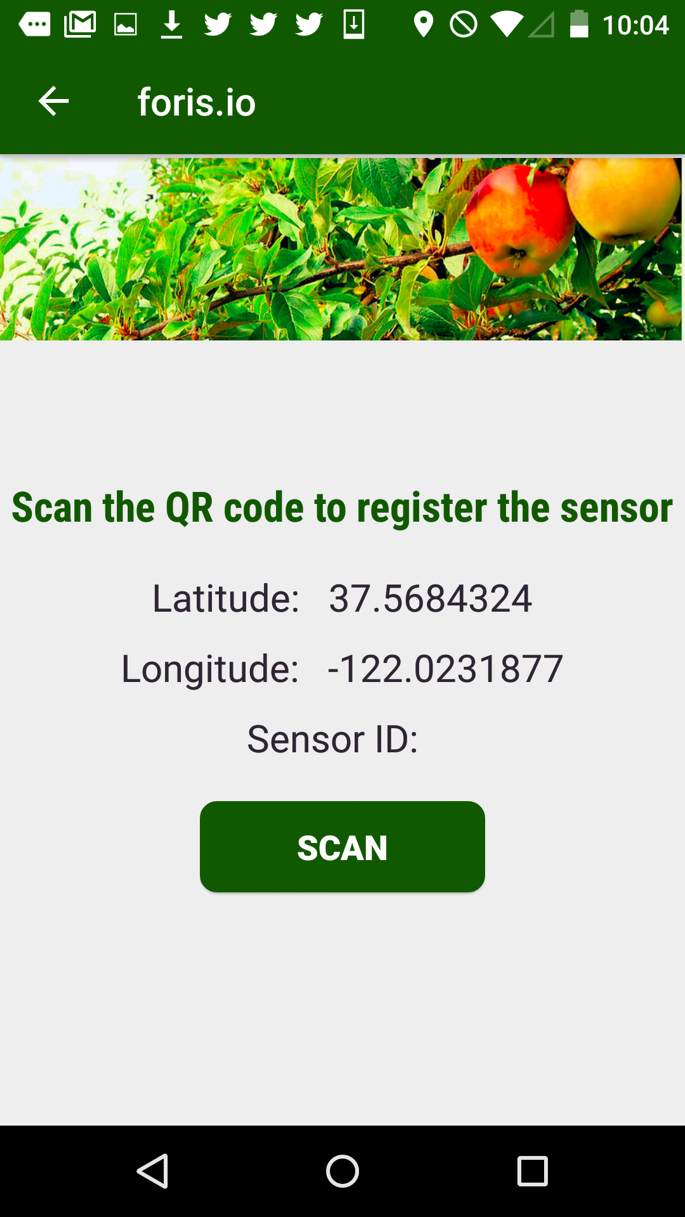
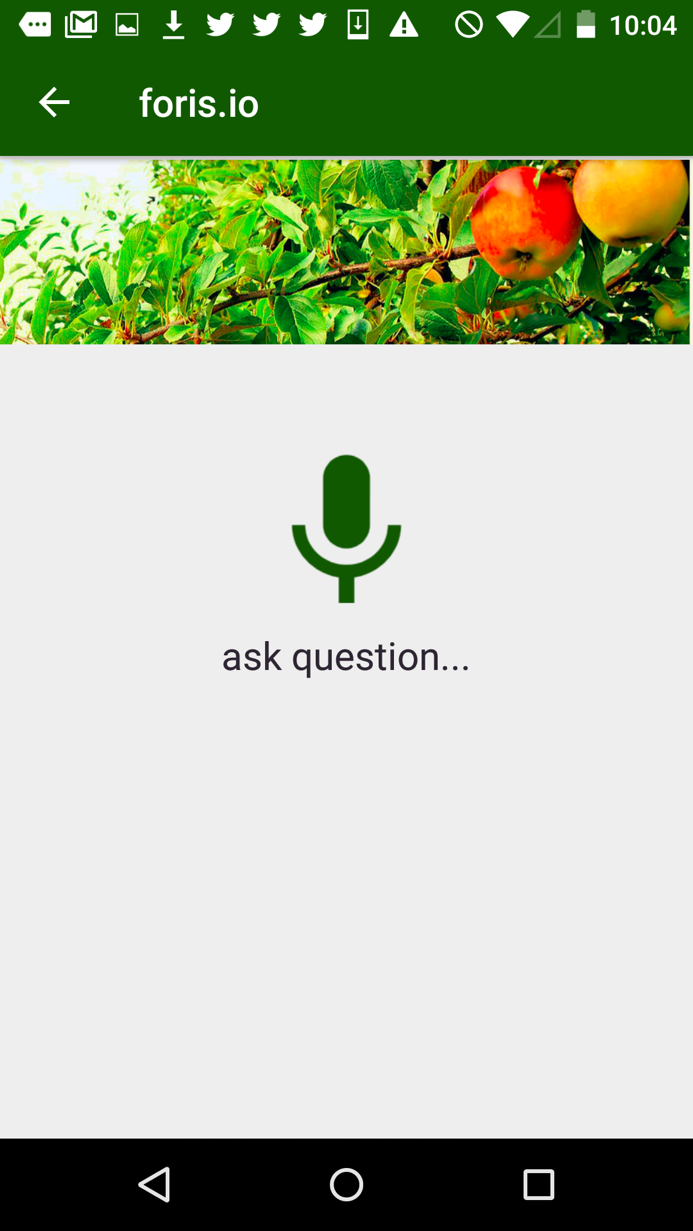


Figure 6 Scan Screen Figure 7 Registration Screen

A notification will be displayed as the toast message which contains the sensor id embedded in the QRCode.



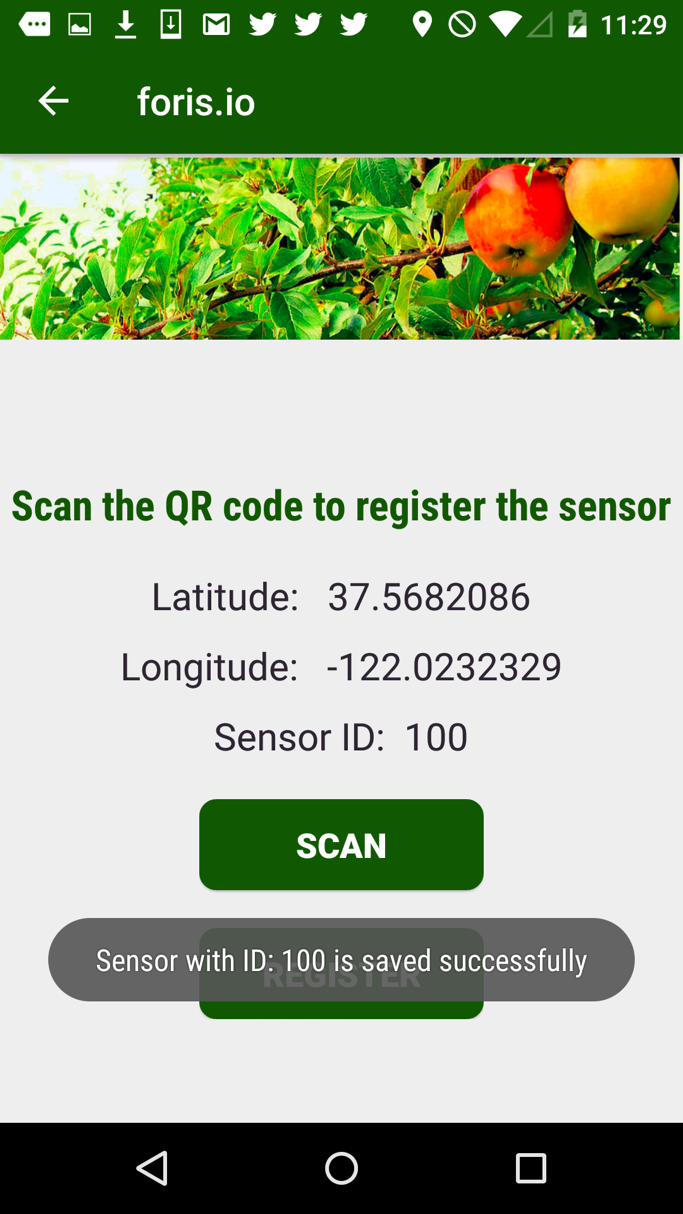
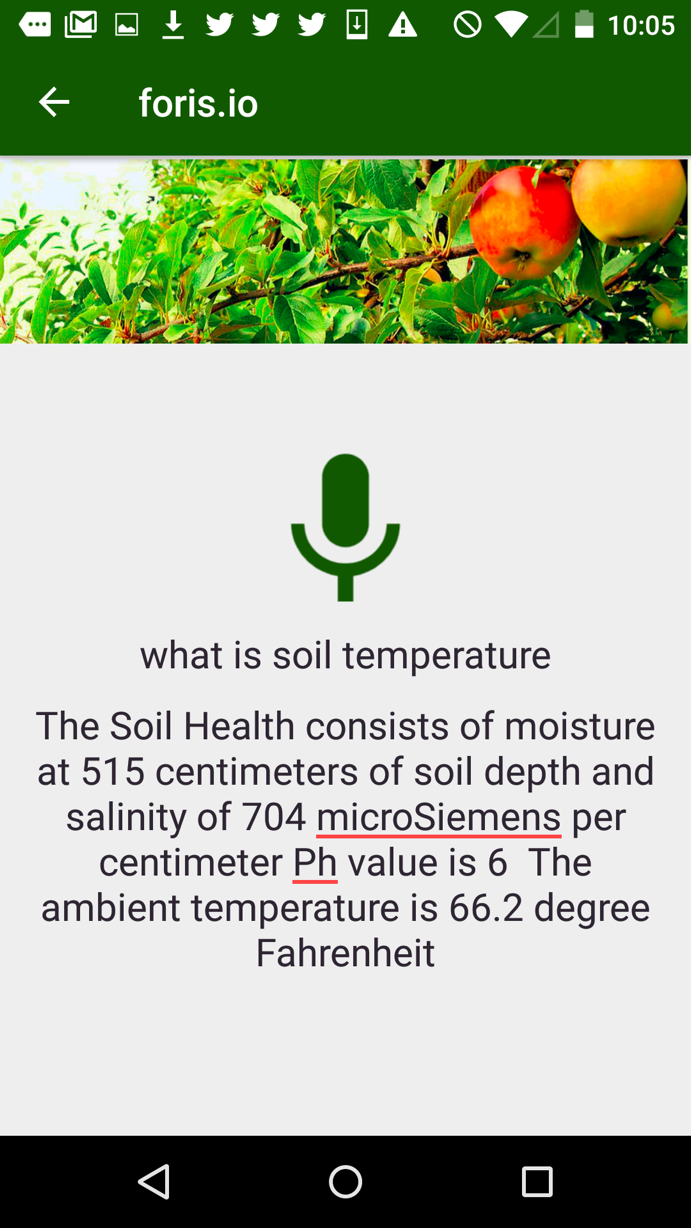
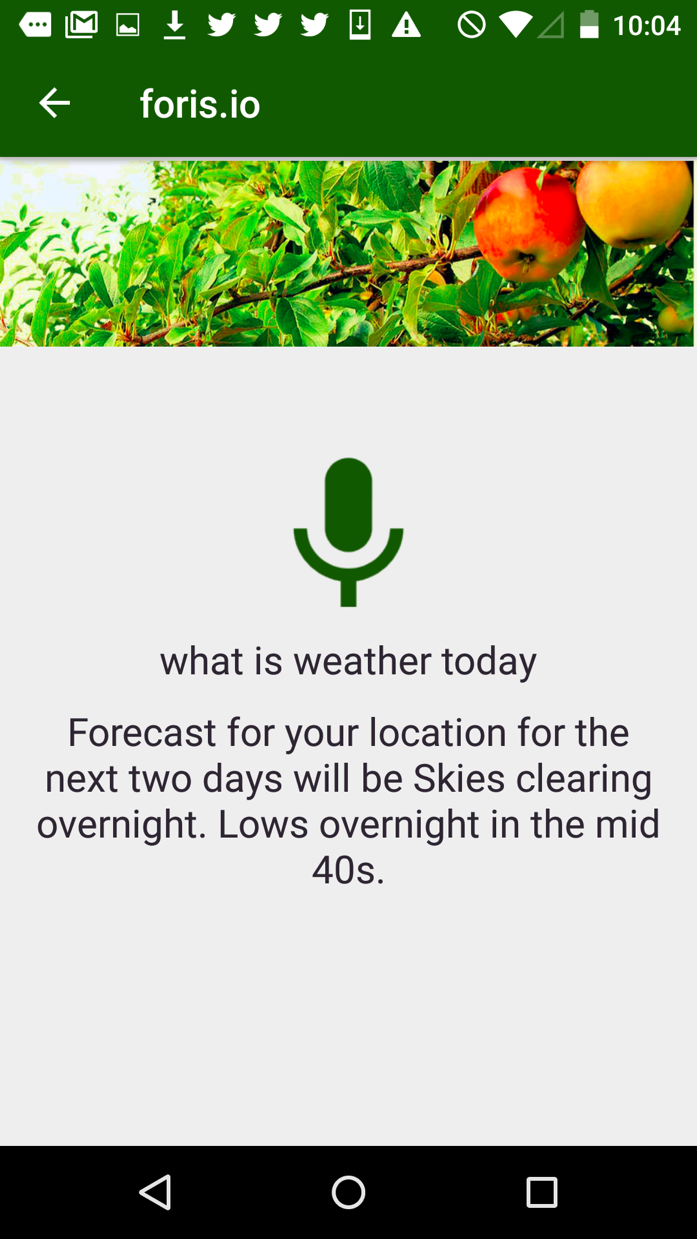


Figure 8 Save Screen Figure 9 Interact Screen

To get the soil health data user can use the “Interact” option. User can ask the question about the soil health or the weather. Application will give the answer according to the question. The appropriate answer will be displayed on the screen.





*Figure 10 Weather Screen Figure 11 Soil Health Screen*

## 6.3. Database Entity Diagrams

This is non-relational database. So, the Database Entity diagram is not applicable to our system. Following is the document structure for the system.

{

"User" : "",

"FirstName" : "",

"LastName" : "",

"password" : "",

"ContactNumber" : "",

"GateWayID" : "",

"Application ID" : "",

"Sensor" : [{

"SensorID" : "",

"Name" : "",

"Latitude" : "",

"Longitude": "",

"Status" : "",

"Data" :

[

{

value : "",

timeStamp : ""

},

{

value : "",

timeStamp : ""

},

{

value : "",

timeStamp : ""

}

],

"ErrorNotification" : [{

"Inactive" : "",

"OutOfRange" : ""

}]

}]

}

## 6.4. Hardware Block Diagrams

**Arduino to Link-lab eval boards connections**

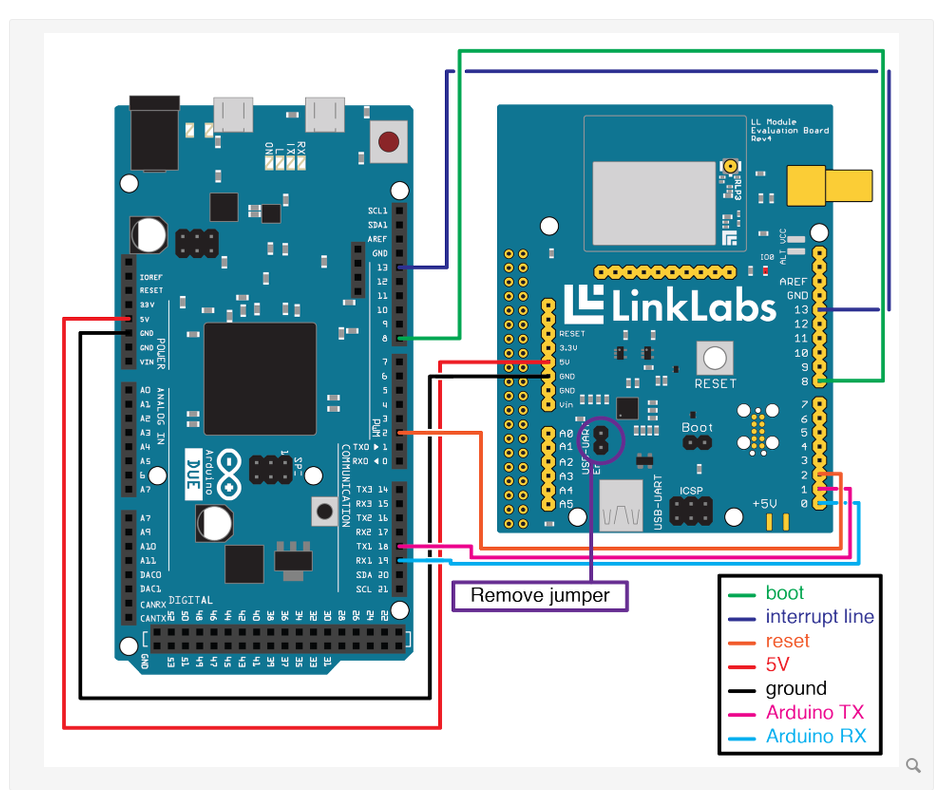


Figure 12 Arduino to Link-lab eval boards connections

**Arduino to sensors connections**



Figure 13 Arduino to sensors connections

## 6.5. System Block Diagrams

C:\Users\Pavani Vellal\Desktop\MSSE study material\4th sem\295A\system block diagram (3).pngThe following block diagram shows the design needed to deploy FarmMD System.

Figure 15 System Block Diagram

The database repository for our project is IBM Cloudant, one of the fully managed NoSQL database-as-a-service. It is built upon the CouchDB and it is also one of the open source services. It is easy to integrate because of the compatibility it has with the RESTful APIs.

# Project Implementation

The FarmMD project being an IOT project has several components such as the hardware, middleware and the application software that make up the system. In order to plan, implement and deploy such a system we have followed the agile development methodology.

## Project implementation scope

The system has several interrelated components that come together to form a robust architecture. FarmMD is an agricultural IOT system that has several tiers such as the Hardware, Middleware and the Application Front-End.

## Client Implementation

* Hardware

The hardware consists of sensors and gateway that connects to WiFi. These are built together to collect data from the atmosphere and soil, and sent to the database.

**Sensors**: The sensors are used to determine atmospheric conditions. Below are some of the sensors used in our system.

* + Temperature Sensor
  + Moisture Sensor
  + Humidity sensor
  + Salinity sensor

**Gateway:** LoRa Nano-Gateway is a WAN solution that provides kilo-meters of range while consuming very low power. This makes it very suitable for IOT. This is used to set up a network. The LoRa network consists of devices, gateway, network and an application.

* Mobile App

The Mobile application for the project is developed in Android OS. The mobile application is used to register a new device by reading QRCode. It is also used to view alerts from the system and take voice commands to perform action. Through the mobile application, the system settings can be customized as needed.

* Web Application

The Web Application provides the user interface for the User and Administrators to interact with the system. It is developed using, Node.js, Express.js and Angular.js. It provides user options to customize the system based on the user’s requirement. It also displays the sensor readings as graphs and charts by fetching the sensor data stored in the Cloudant database and visualizes the data as charts. This helps give the user a better understanding of the data.

The application also provides notifications and warnings according to the settings defined to the registered recipients.

* API

The system also provides API for fetching data and analyzed decisions from the system. This can be used by 3rd party devices to fetch data and watering decisions.

* Watering Schedules from Predictive Analytics

The system determines the ideal watering schedules for crops based on the sensor data collected. Doing so helps effective usage of water resources as well as improve crop yield.

This is done using R and the result of these operations are consumed in the Node.js web application which is then provided as an API for usage in other devices.

## Middle-Tier Implementation

The middleware tier is responsible for collecting data from gateway and sending it to the Cloudant database.

Using CURL library in C, an application is designed to send sensor data to the Cloudant noSQL Database directly from the gateway. This has been developed from scratch as there was no documentation available to send data from the LoRa-Nano gateway directly to the cloud.

## Data-Tier Implementation

* Cloudant noSQL Database

This is a cloud database by IBM where the system’s data resides. This data is used in the web and mobile app to retrieve sensor readings as well as user details.

## Implementation methodology

We have followed the Agile Software Development methodology for all the phases of our project development. We have built frequent working versions of our system and worked closely with each other in a team to clearly understand the requirements and deliver suitable solutions for it. We communicate on Slack and have weekly scrum meetings to discuss the progress of our wok. We also maintain to-do lists to be able to track our progress.

## Project implementation issues and resolutions

The working of this project posed multiple challenges on us. Below are some of the challenges that define our project.

* As it is an IOT project, we were supposed to learn to work with hardware and controllers, it was a steep learning curve which eventually taught us a lot about working with low level languages and interfacing with hardware.
* Decision on using suitable gateway technology for our requirement involved us experiment with various different hardware and understand the way they are working.
* Using CURL library in C, application is designed to send data to the Cloudant noSQL Database directly from the gateway. This has been developed from scratch as there was no documentation available to send data from the LoRa- Nano gateway directly to the cloud.
* Working on the web application involved a lot of learning as the application was to be in Angular.js and Node.js.
* As we are working with very new technology, we have not had the privilege of getting much from the internet. We are worked hard to understand everything on our own and with the help and guidance of our project guide.
* Finding the right data to determine accurate watering schedules has been a challenge as there is a lot of research to be done in getting the right data and then processing the data to get results suitable to us.

## Used tools and technologies

We have used a number of emerging technologies in our project. Below are some of the major categories of technologies used.

Hardware and Middleware:

* LoRaWAN : It is a Low Power Wide Area Network (LPWAN) specification for battery operated things in a network. LoRaWAN provides secure bi-directional communication, mobility and localization services. The LoRaWAN specification provides seamless interoperability among smart Things without the need of complex local installations.[15]
* LoRa Nano-Gateway: Is a WAN solution that provides kilo-meters of range while consuming very low power. This makes it very suitable for IOT. This is used to set up a network. The LoRa network consists of devices, gateway, network and an application.
* Sensors such as the temperature, humidity, moisture and salinity.
* Node red: Tool for integrating devices and APIs by IBM.
* R : R programing language is used in predicting effective watering schedules for the crops.

Client Applications:

* Android: Is used for developing the mobile application.
* Node.js and Express.js: For developing the backend of the Web Application and for building REST APIs.
* Angular.js, Bootstrap, HTML, CSS, JQuery: For Developing the front end of the Web Application
* Canvas.js: For developing the graphs of the application
* NEXT framework by Cisco is used to show the sensor topology in the web application.
* RESTFul services and JSON: For sending and receiving data to and from the cloud.
* Cloudant NoSQL database: Database that is used to store the sensor data and the system specific information.
* Git and Github for maintaining and collaborate in groups and for tracking.

# Testing and Verification

**Test Strategy**:

Following types of testing will be performed on this system:

a. Unit testing: This is going to be basic type of testing in which we will test each smallest function of the system to verify expected results.

b. Integration Testing: Once we consolidate front end, middleware, backend and the database together, we will perform this testing to make sure system is working end-to-end

c. Performance Testing: This testing will verify if system is working according to the performance criteria’s mentioned earlier.

**Testing Process**:

1. Test plan will be created
2. Testing tools we are going to use:

a. Selendroid framework for Android mobile app testing.

b. Mocha for web application development.

1. Test cases will be written and executed. 4
2. Test results will be document

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01 | Hardware SetUp | Test case to check if hardware setup and data transfer is correct. | 1.Setup the hardware according to instructions in ReadMe.  2.Check if data is sent properly to the cloud. | FarmMD should have proper hardware setup and should send the appropriate data to the cloud. |  |
| 02 | Mobile App installation | Test case to check if mobile application has been installed correctly. | 1.Install mobile application on Android device.  2. Click on the FarmMD icon.  3. Application should open without any errors. | Application should get open and home screen should be displayed. |  |
| 03 | Web App installation | Test case to check if web application has been installed correctly. | 1.Access the link to open the web application  2. Application should open without any errors. | Application should get open and home screen should be displayed. |  |
| 04 | Mobile App Login | Test case to check if mobile application Login functionality is working. | 1.Open the mobile application.  2.Enter credentials which is username and password.  3.If its correct, home screen should be displayed. | Login should be successful if credentials are correct else error message should appear. |  |
| 05 | Web App Login | Test case to check if web application Login functionality is working. | 1.Open the web application.  2.Enter credentials which is username and password.  3.If its correct, home screen should be displayed. | Login should be successful if credentials are correct else error message should appear. |  |
| 06 | Web application Graph | Test case to check if the graph is correctly showing expected information in web application. | 1.Open the web application.  2.Open each type of graph.  3.Check if all the graphs are opening correctly and displaying the correct information. | All graphs should open and display the correct information. |  |
| 07 | Web application Weather data | Test case to check if weather widget is correctly showing expected information in web application. | 1.Open the web application.  2.Observe the data in weather widget.  3.Check if its displaying the correct information. | Weather widget should display the correct information. |  |
| 08 | Web application Twitter  API | Test case to check if Twitter API is displaying correct data in web application. | 1.Open the web application.  2.Observe the twitter in feeds widget.  3.Check if its displaying the correct information. | Twitter widget should display the correct information. |  |
| 09 | Web application Sensor  Topology | Test case to check if sensor topology is displayed correctly. | 1.Open the web application.  2.Open the sensor topology widget.  3.Check if its displaying the correct information. | Check if sensor information is correctly displayed in sensor topology. |  |
| 10 | Sensor Registration with Mobile App | Test case to check if sensor is registering correctly in Mobile Application | 1.Open the mobile application.  2.Use QR code scanning feature to scan the QR code and register the sensor. | Check if sensor information is correctly stored in cloud DB. |  |

# Performance and Benchmarks

**Performance:**

Performance is a trivial factor in any project. In future, we may deal with peta bytes of data as the sensor keeps pushing the data for every regular interval of time. We have implemented most optimal algorithms to send and receive the data. In every step, we made sure to have the optimal time and space complexity to serialize and reserialize the data.

We are hosting our web app in multiple servers to make sure load balancing is achieved, meantime we make sure they ate synchronized. Every cloud hosting providers provide default load balancing. We have also implemented in a similar way. Since our user base is going to be more down the line we made sure it’s available so huge number of visitors.

**Benchmarks:**

We always keep our benchmark as tech giants in the industry. When it comes to deal with data, companies like google can be set as a benchmark. As we have mentioned before, while dealing with huge amount of data we need to make sure to use as less resources as possible. Adding the additional hardware like storage and compute may end up with more cost.

# Deployment, Operations, Maintenance

Describe any deployment strategies, operational needs, and maintenance required for your project.

1. Deployment

* The Web Application that is used to interface with the system is deployed on the IBM Bluemix Cloud.
* The API to fetch the data from the cloud is available on IBM Bluemix Cloud Platform.
* The Cloudant database is a cloud database available on the IBM Bluemix Cloud Platform.
* We have 3 different stages for Developing, Staging and Production hence always securing our production.
* We have set up the system for continuous delivery by using continuous integration tools in IBM Bluemix with Git.

1. Operations

One of the major aspects of our projects is the Analytics involved in it. Several data intensive operations are performed on the data collected from the sensors and the data from other external operations to make accurate decisions on saving water and optimizing agricultural yield. We are currently working on developing suitable formulas for operations on data.

1. Maintenance

In the development of a large project, it is essential to plan the maintenance of the software well in advance. Below are some of the ways in which we are currently maintaining our software and intend to make delta improvements in the future.

* The application code is maintained in a common group github repository which helps keep a track of our changes and the progress that we have made.
* Continuous integration and deployment of application from github reduces efforts in updating changes in multiple locations.
* The common database being using for the application is available on the cloud accessible to all developers.
* We have maintained good coding standards in order to improve its understandability.
* The reports that we have worked on time to time helps in effectively documenting the system.
* We have designed the code to increase efficiency and reusability.

# Summary, Conclusions, and Recommendations

## Summary

‘FarmIoT’ system is providing a unique IoT solution the problem to saving water and energy using ‘LinkLab’ technology. This system also supports huge variety of sensors to provide understanding about soil health. Any grower can access this system via a web application or a mobile application.

## Conclusions

This project was an opportunity to work with fast paced startup in the bay area. As a team, we have learned a lot together starting from requirement gathering till testing phase. This system gave us a good understanding about IoT platform, data analytics and machine learning, web application and mobile frameworks.

## Recommendations for Further Research

* This system can be enhanced to work in the areas where there are network problems.
* Data analytics and machine learning part can be enhanced to give more number of recommendations to the users.
* More number of exciting features can be included in mobile and web application.

Glossary

[1].**IOT** (Internet of things) is a system of interconnected computing devices with unique identifiers that have the ability to connect to the network and send data without human intervention.

**[2]. Artificial Intelligence** is the man made system that is designed to behave in a manner that can be called intelligent.

**[3]. Machine Learning** is a type of Artificial intelligence that provides computers the ability to learn without being explicitly programmed. [3]

**[4]. Linklabs** is a device that provides low-power, wide area network technologies to power Internet of Things.[4]

**[5]. Arduino Due** is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU [5]

**[6]. LPWAN** is a WAN technology specialized for IOT device connectivity with low-bandwidth.

[15]. “Richard Viel,” *LoRa*. [Online]. Available: https://www.lora- alliance.org/What-Is-LoRa/Technology. [Accessed: 06-Mar-2017].

References

1. D. B. E. and D. A. W., Purdue University, “2015 PRECISION AGRICULTURAL SERVICES DEALERSHIP SURVEY RESULTS,” SPONSORED BY CROPLIFE MAGAZINE AND THE CENTER FOR FOOD AND AGRICULTURAL BUSINESS, pp. 19–20, Aug. 2015.

http://www.link-labs.com/symphony/

1. M. H. Anisi, G. Abdul-Salaam, and A. H. Abdullah, “A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture,” Precision Agric Precision Agriculture, vol. 16, no. 2, pp. 216–238, Nov. 2014.
2. “What is machine learning? - Definition from WhatIs.com,” WhatIs.com. [Online]. Available: http://whatis.techtarget.com/definition/machine-learning. [Accessed: 12-Dec-2016].
3. @L. L. I., “Learn More About Link Labs,” Link Labs, Jun-2015. [Online]. Available: https://www.link-labs.com/about/. [Accessed: 12-Dec-2016].
4. “Arduino - ArduinoBoardDue,” Arduino - ArduinoBoardDue. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardDue. [Accessed: 12- Dec-2016].
5. “foris-io,” foris-io. [Online]. Available: http://www.foris.io/. [Accessed: 12-Dec- 2016].
6. “Application of Internet of Things in Facility Agriculture,” 2016 ASABE International Meeting, 2016.
7. “Richard Viel,” LoRa. [Online]. Available: https://www.lora-alliance.org/What- Is-LoRa/Technology. [Accessed: 06-Mar-2017].