SCS 3214 / IS 3113: Group Project II - 2020

<u>Interim Report</u>

Project Title: Green Core - “The Smart Gardner”

Project Group Details

1. Group number: G27
2. Group members:

Name	*Reg. Number*	*Index Number*	*Email address*	*Mobile Phone*
(i) M.M.H Tharushika	2017/CS/176	17001765	hasinimanage@gmail.com	0772087177
(ii) H.U.K.M Pabasara	2017/IS/061	17020611	minurapabasara@gmail.com	0710310699
(iii) J.A.N.C Niroshana	2017/CS/119	17001196	chandimal.nipuna@gmail.com	0770495637
(iv) B.A Medawatta	2017/CS/107	17001072	binaramedawatta@gmail.com	0767322639
(v) K.C Gamage	2017/CS/047	17000475	kavigamage23@gmail.com	0716008664
(vi) K.S.A Ahamed	2017/IS/003	17020034	ibnusafina@gmail.com	0767018012

Details of Project Supervisor, Co-supervisor, Advisors and Clients

Project Supervisor (Academic Staff of UCSC):

Name of the supervisor: Dr Dinuni K Fernando

Signature of the supervisor:

Date:

Project Co-Supervisor (Assigned by Course Coordinator):

Name of the co-supervisor: Ms. Sithara Fernando

Signature of the co-supervisor:

Date:

The client of the Project

Name of the client: Dr Dinuni K Fernando

e-mail address of the contact person: dkf@ucsc.cmb.ac.lk

Project Details:

1. Project Title: Green Core - "The Smart Gardner"

People have busy lifestyles, most of the working crowd has very limited time at home, yet people tend to grow their own produce for sustainability. This project aims to develop a smart garden environment that allows the user to provide real-time gardening experience via a mobile phone. Our project aims on developing a Mobile Application that is capable of monitoring the water level of each individual unit, humidity levels, the intensity of light levels, etc of a real garden. Our Mobile Application with the help of sensors collects real-time sensor data to control and manage the garden.

2. The Goal and Objectives:

Our main goal is to develop a smart garden environment that-allows the user to provide a real-time gardening experience via a mobile phone.

Objectives

- People can get fresh vegetables and food from their garden with aid of a Mobile Application.
- Track the progress/status of a bed of plants.
- Adjust the growing environment remotely and in an automated way.
- Get real-time notifications of unit statuses via mobile App
- Administrators can collect, view data of all gardens via web application and inform users if there are any issues with the units.

3. Problem Definition

With busy lifestyles, people tend to stay at home leisurely for a limited time. However, with the current sustainable trend, people tend to grow their own produce. With this growing trend, people with limited time should have a feasible way of gardening with a minimum effort. So, there should be a convenient way of getting organic produce to the household with a minimum effort. Also, an agricultural revolution is going on these days because of the pandemic which occurred in the recent past. But after this is gone people may not have time to take care of their plants. For the above problem, we introduce a smart garden concept which is an automated way of gardening by using a mobile phone.

4. A brief introduction to the project

A smart Garden which allows the users to control their garden while they are away. The automatic gardening feature will check the environmental conditions and maintain the conditions required for the plants to grow. The manual method allows the users to control the plant conditions which will give full remote control of the garden to the user. This includes an IoT device, a Mobile Application and an admin portal.

The system mainly consists of two main deliverables.

1. Mobile Application and IoT Device

   System for the user consists of one or more IoT device units and a Mobile Application. The users can register to the system using the app and then connect the units for their account. A unit contains multiple sensors and actuators which are used to control the garden.

   Using the Mobile Application the users can control actions such as adding nutrients, watering the plants when the soil moisture level is low, changing lighting conditions when light intensity is low, etc. With the help of our Mobile Application users can get fresh vegetables and food from their garden.

   Plant watering system will automatically and manually water the plants when the soil moisture level is low and the surrounding temperature is high. If the user does not have sufficient knowledge of the plant, once the user starts cultivating the seeds, the app monitors the status of the plant and provides periodic data regarding the status of the plants.

   Through the chat module, users can send their issues about the units to the admins and ask for help if possible

2. Web Application

   Web application helps administrators to manage user accounts, devices and get statistics about them. Through this, admins can view data gathered from all the units. This feature helps to identify vulnerable units and help admins to fix issues related to individual devices

   Admins also can help users to fix issues related to their units through the chat module.

5. The scope of the project

Users (possible actors) of the system:

- Admins - Web Application
- Users - Mobile Application

Main functionalities of the system:

- Mobile Application
  - Control and get feedback of the IoT units
  - Manages IoT units
  - Manage the settings related to user accounts
  - Show notifications related to IoT devices
  - View the status of the garden (water levels, moisture levels)
  - Chat module to contact administrators regarding issues with the units
  - Receive planting tips
  - Chat with admins to resolve issues

- IoT
  - Plant watering system
  - Lighting system
  - Add nutritions

- Web Application
  - Analyze the data from all the gardens with web Application
  - Manage unit Data (Admin)
  - Monitor each IoT device's health and notify if something is wrong
  - Chat module to contact users regarding issues in their units
  - Manage the data related to the tips provided by the Mobile Application.
  - Chat with users to fix their issues

Out of Scope

- Train a machine learning model using the data gathered by the sensors to predict optimal conditions to grow specific plant types
- Monthly report generation for Mobile Application and web app
- Track progress of the plant growth and harvest

6. Technologies Used

- Mobile Application - React Native
- Web Application - React JS
- Web servers - Nodejs and ExpressJS
- IoT - Arduino Mega and ESP8266
- Database - MongoDB

Reasons for choosing,

React Native for the Mobile Application
- Easy to learn.
- Cross-platform Mobile Applications from a single codebase.
- Higher performance.
- Component-based development.
- Native look and feel.

React.js for the Web App
- UI can be segmented into components and write logic accordingly.
- Has a shallow learning curve.
- Can reuse components.
- The syntax is much like HTML with javascript.

Node.js with Express.js for the Backend servers
- Easy to scale the applications horizontally as well as vertically.
- Higher performance because of non-blocking I/O operations.
- Easy to learn because of the use of javascript.

Arduino Mega & ESP8266 for the IoT Device
- Production cost is low. we can print our own Arduino board by customising according to the requirement.
- Arduino is open source.
- Documentation is available for free.

MongoDB for the Database
- Due to the structuring (BSON format - key-value pair) way of the data in MongoDB, no complex joins are needed.
- Processing and analyzing big data is easy with NoSQL.
- Scalability and Availability requirements for sensor data storage.
- Easy to scale up when the system grows (long term).
- Clustering - can have different clusters for different regions (long term).

Reasons for not choosing Raspberry Pie for the IoT Device

- When it comes to production we can print our own Arduino board but we have to use Raspberry Pie without customizing it (since Raspberry Pi is not open source) so the production cost (cost per unit device) if we use Arduino is low compared to Raspberry Pi.

Reasons for not choosing Firebase as the Database

- Even though we have requirements of a NoSQL database we don’t need real-time processing of data.
- Firebase’s free quota has limitations than MongoDB

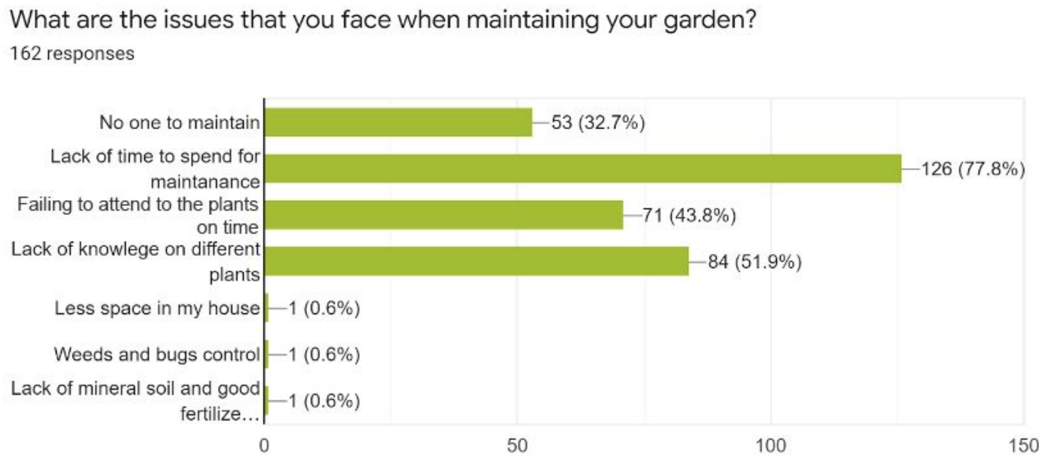
7. Feasibility Study

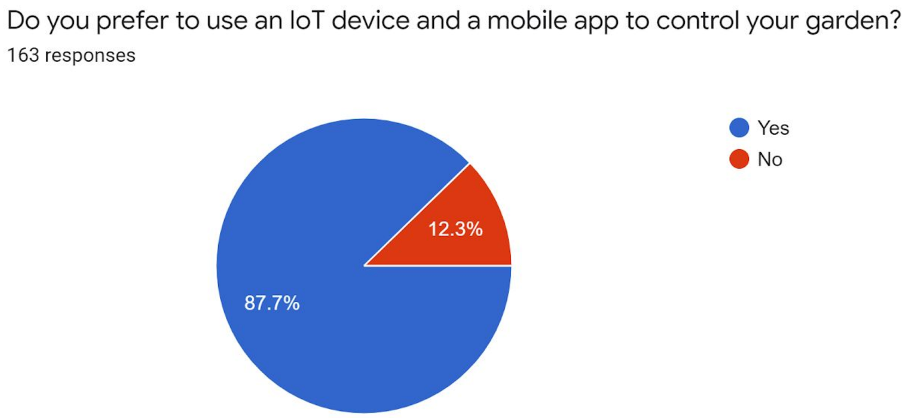
Operational feasibility

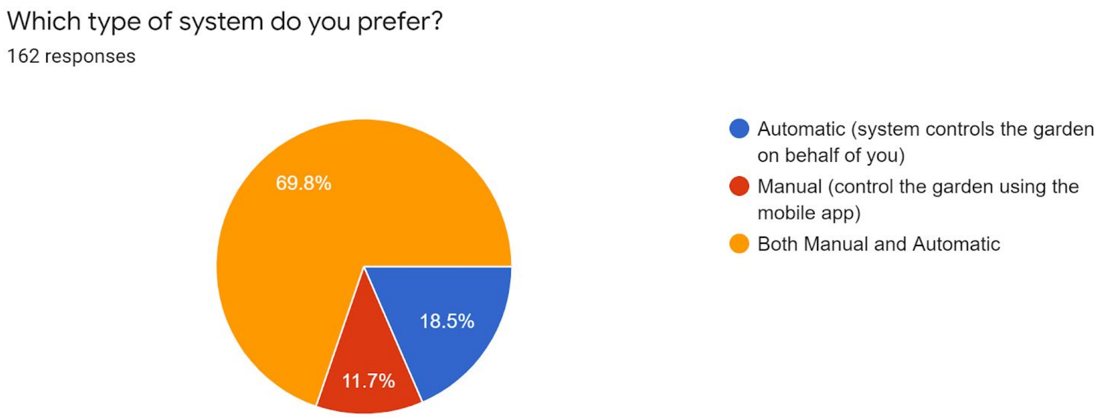
We have surveyed over 100 people from different areas of the island who have started farming in their home gardens after the pandemic started. We let them know about the idea of this system and asked whether they think it would be useful for them in their field. Over 85% of the participants of this survey gave positive feedback saying they are interested in such an idea and it would solve most of their problems.

As most of the fruits and vegetables we buy from the markets are heavily treated with pesticides and various types of chemicals, people are much willing to grow their own food. But the real issue occurs when it comes to time. As this system solves the time issue more people are willing to use this kind of solution.

Results of the survey

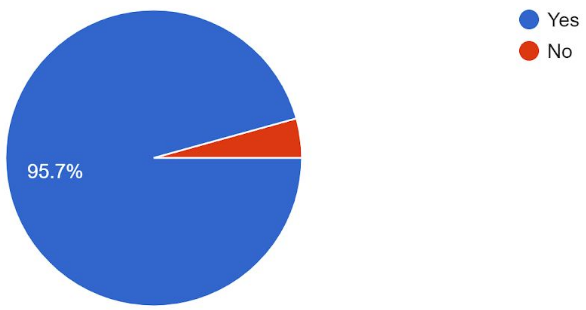






Is it comfortable for you to get the service of this type of application with the facilities you already have?(WiFi, Mobile devices etc)

163 responses



Technical Feasibility

The main product of Green Core is an IoT device built using NodeMCU and an Arduino and a Mobile Application developed by using React Native, JSX and for the backend, we will be using Node js and for the database, we will be using MongoDB. Most of these technologies are freely available or could be bought for a low cost.

For the administration, we are using another web app, which is hosted in a dedicated AWS server which is also available for free for a year. After a year we will have to pay and upgrade or look for another server to host our application. The dedicated hardware of 1GB RAM and 1 core CPU will be sufficient at the start. But as the user base grows, we will have to opt for a more powerful server. But for development purposes, these technologies are certainly sufficient.

Schedule Feasibility

Green Core should finish development by the end of August. And since we are following agile methodology for development, requirement gathering also happens continuously parallel to development.

It means more and more features may get added from time to time. But in the current context, considering the number of features that we have to implement as at now it will be possible. Also, it is worth noting that since we are using MERN stack, Arduino, node MCU, and AWS it makes development much easier and hassle-free which accelerates the development time to make it possible to be done by the deadlines.

Economic feasibility

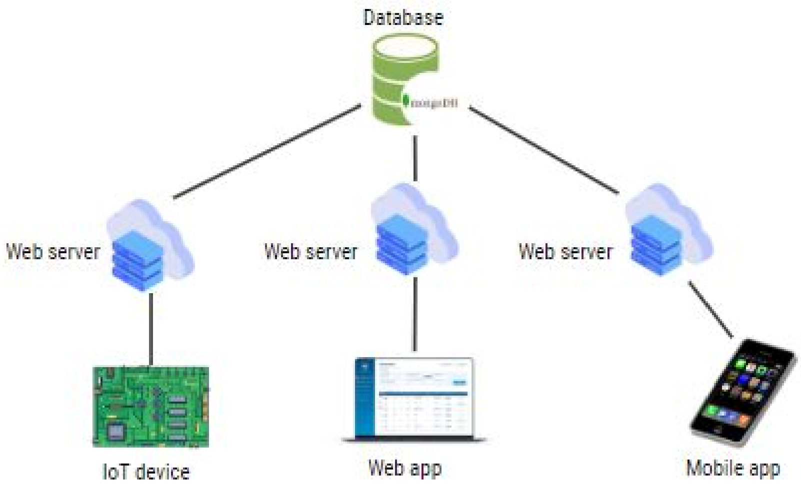
As we are developing this project with profitable motive the production cost should be much lower

- The web application is hosted in the AWS free-tier server, which is provided by AWS for free for one year and this bears zero cost for the first year of operation.
- As we use MongoDB for our database which is free and there are 25Million database operations provided for the free tier
- The web application should have a domain and we will have to pay a small annual subscription for the domain. ($12 per year)
- The development tools like visual studio code Arduino IDE can be used free of charge for the project.
- As we use NodeMcu and Arduino for the IoT units the cost will be around 15$ per unit
- Publishing an Android app requires a developer account which we will have to make a one-time payment for the account. ($25 one-off payment)
- Since it is developed only by a team of university students, development cost also can be factored as none

Therefore in the short-run, the development takes a very small cost ($37 + costs for units) (if we didn't factor the developers' cost). But in the long-run, as the system grows, we may have to bare a moderate cost for hosting fee in AWS as well as to allow more capacity in the MongoDB

With all these four feasibilities in check, we can safely assume that this is a feasible project in both the long and short run, thus we should proceed further with the development of this system.

8. Systems Architecture



- The system consists of a central database to which the Mobile Application, web app and the IoT device are connected using independent servers.
- Communication between the apps, devices and the servers are done using API calls.
- The IoT Device will collect data from the sensors and pass it to the database.
- The web app and the Mobile Application can retrieve data from the database.
- All the functions carried out by the user with the IoT device will be first updated in the database and then the action will be carried out in the IoT device.

9. Requirements Specification

You can find the complete requirement specification here https://drive.google.com/file/d/1RNIBIx4MRRw2TDgEsSO9oPC2uZHnhxWa/view?usp=sharing

Functional Requirements of the Mobile App

- Register User
- User Login
- View User Details
- Update User Details
- Change Password
- Forgot Password
- Navigation
- Link Units
- View All Units
- View Single Unit Details
- Control Actuators
- Send New Message
- View All Messages
- View Single Message
- Reply Message
- Alerts

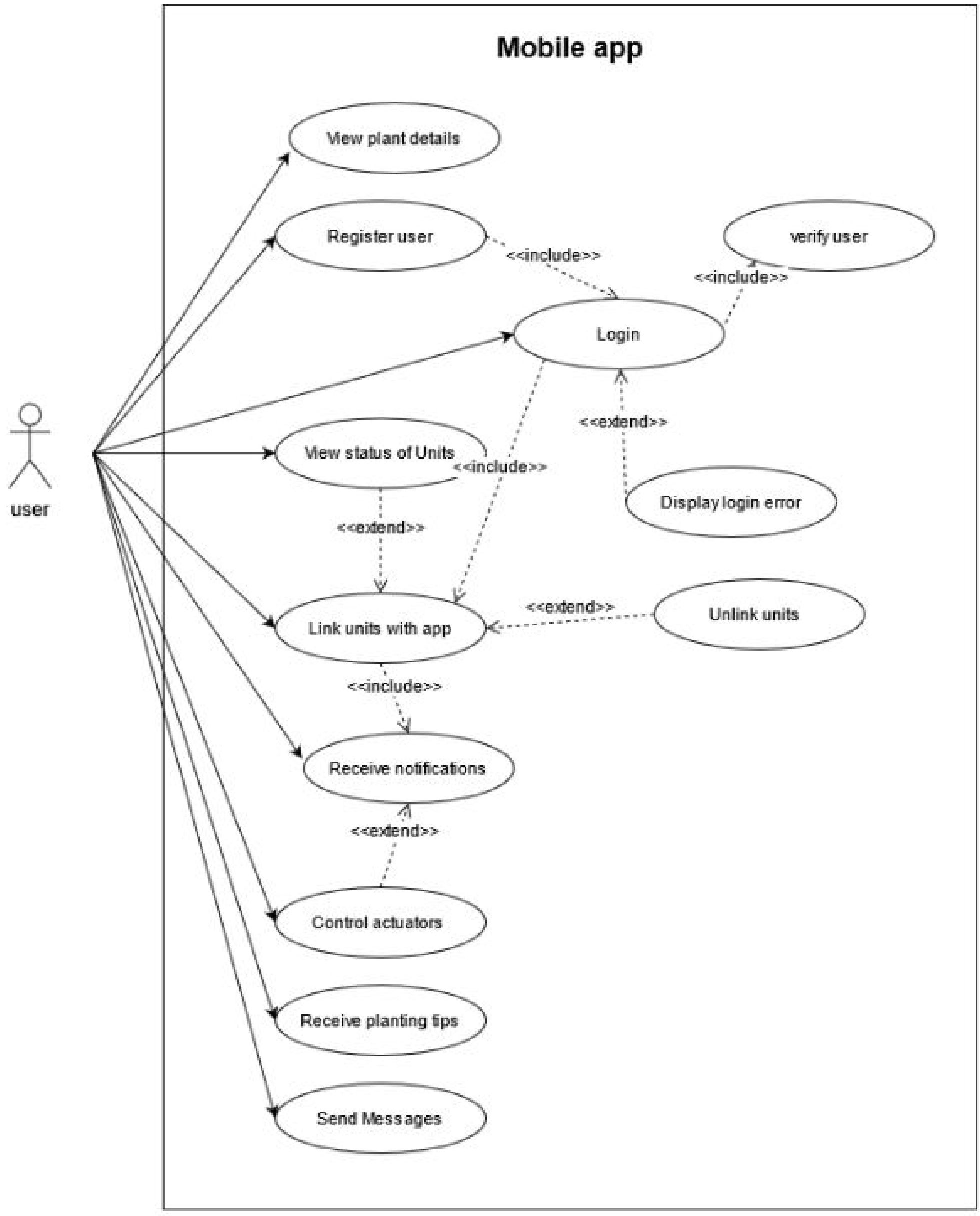
Functional Requirements of the Web App

- Register User (Admin)
- User (Admin) Login
- View User Details
- Update User (Admin) Details
- Change Password
- Forgot Password
- Navigation
- View Users
- Search Users
- View Single User
- View All Units
- Search Units
- View Single Unit
- View Vulnerable Units
- View All Messages
- View Single Message
- Send Message
- Reply Message
- View New Messages

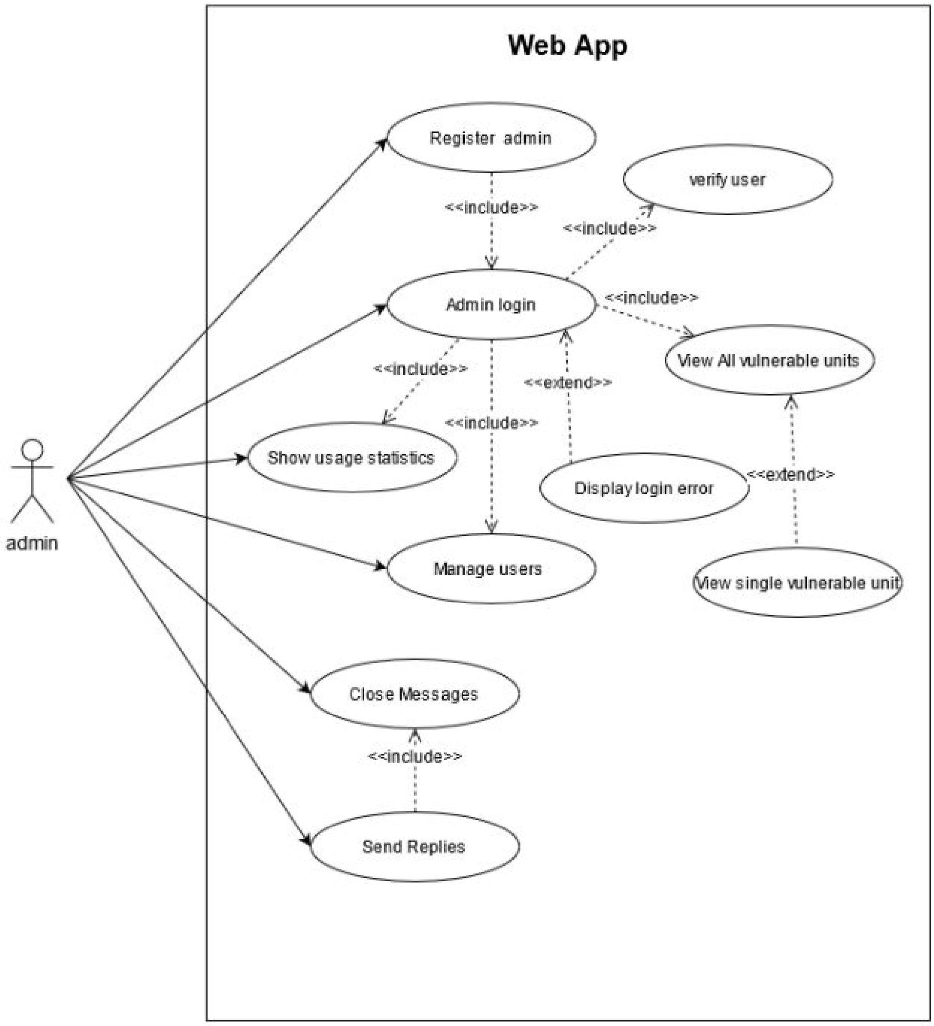
Functional Requirements of the IoT Device

- Watering Plants
- Adding Nutrients
- Change Lighting Conditions
- Activate the Buzzer
- Check sensor readings (ex: Soil Moisture, Humidity etc)
- Send sensor data to the user

Mobile App Use Case Diagram



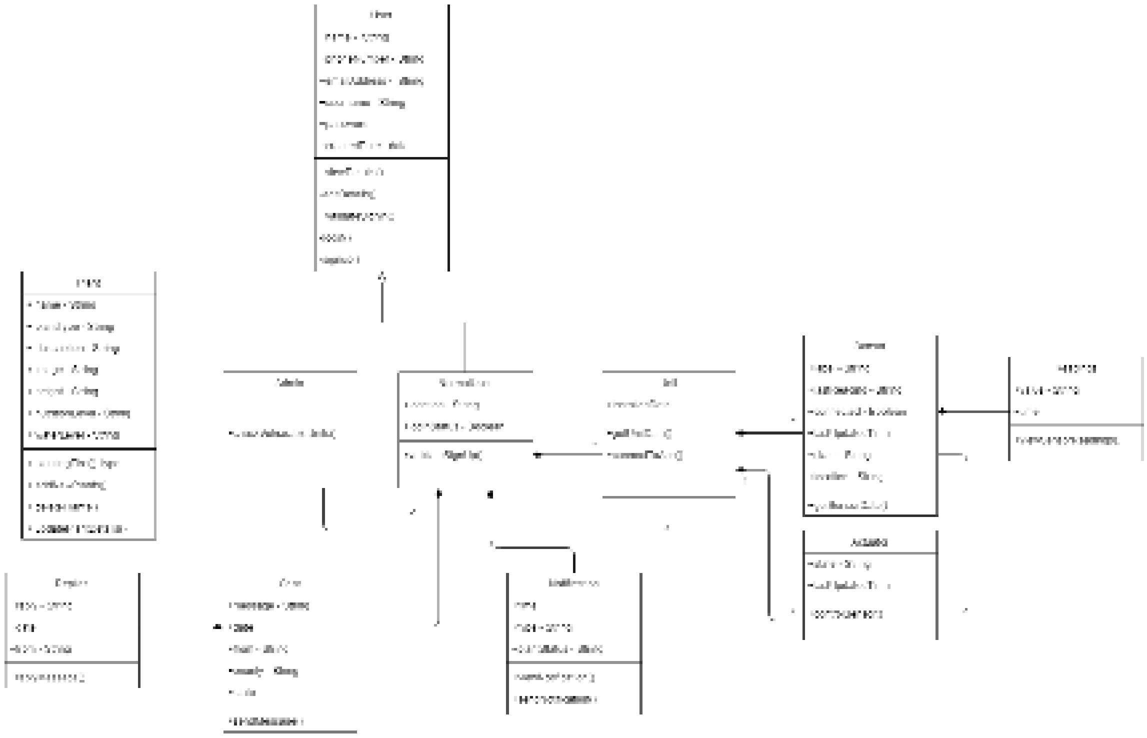
Web App Use Case Diagram



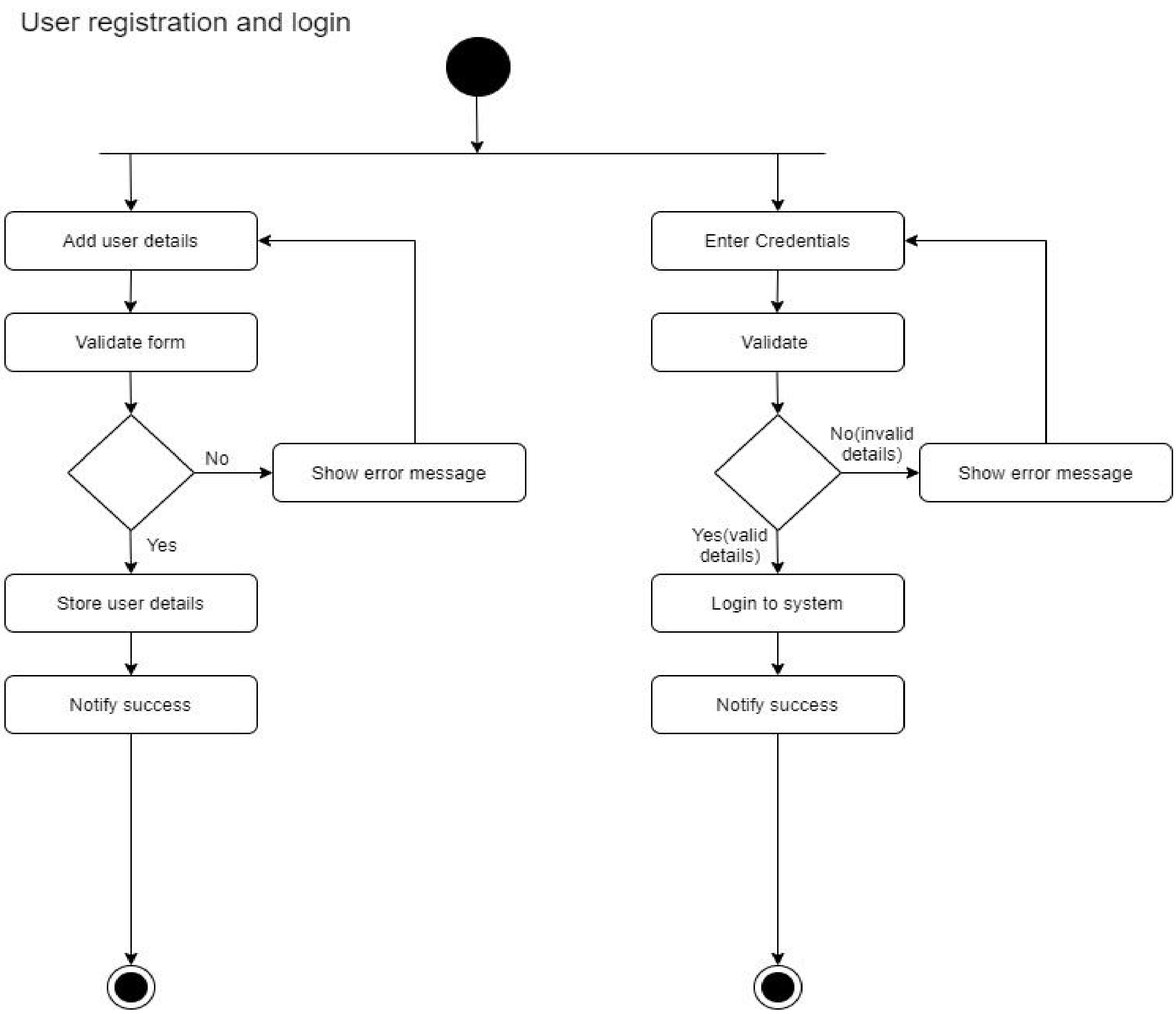
10.Design Specification

Class Diagram

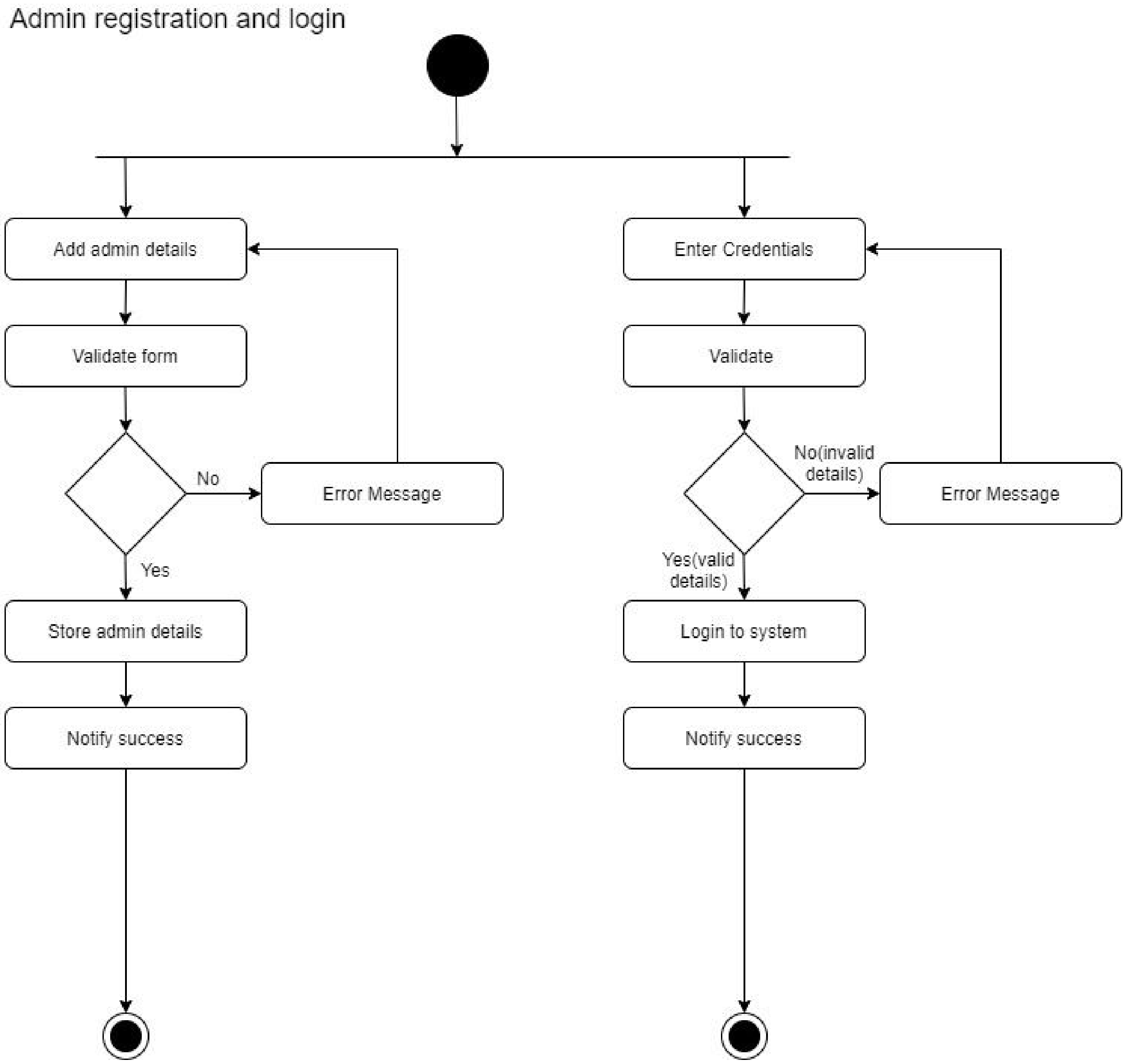
https://drive.google.com/file/d/1Z5BIZyDVVxbIrcTP3Bn35S1-5tVtpGdK/view?usp=sharing



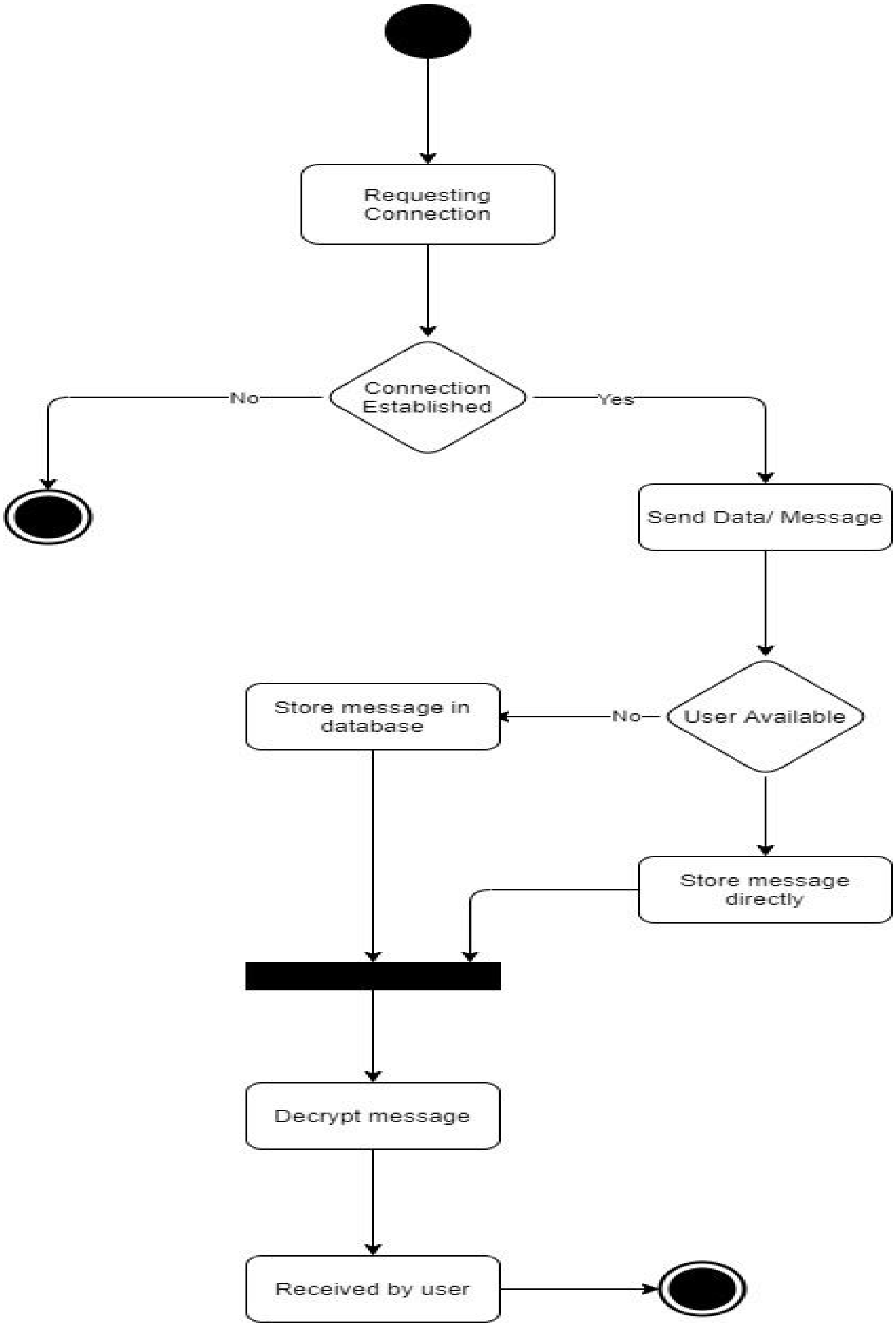
Activity diagram for user registration and login- Mobile App



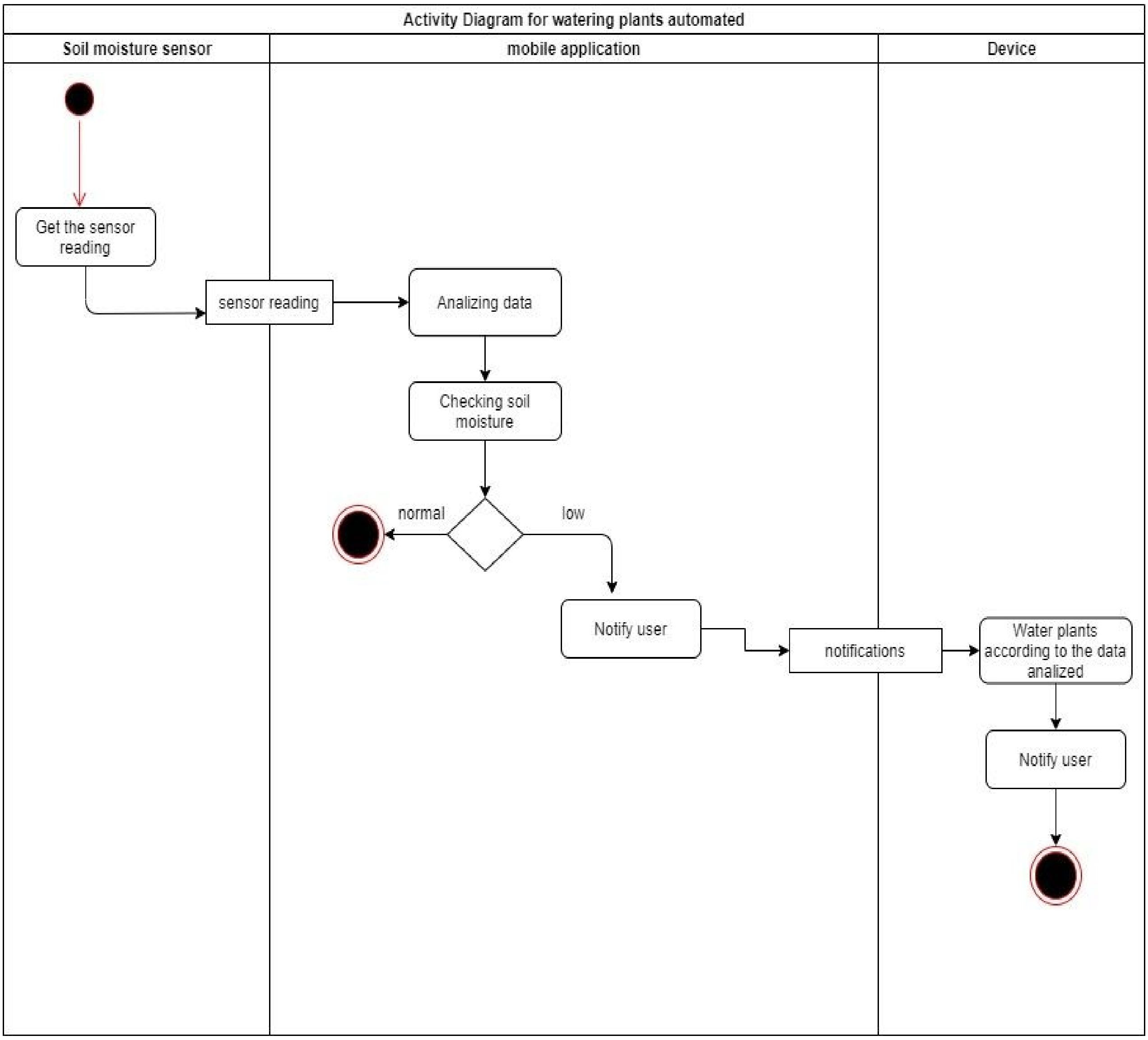
Activity diagram for user registration and login- Web App



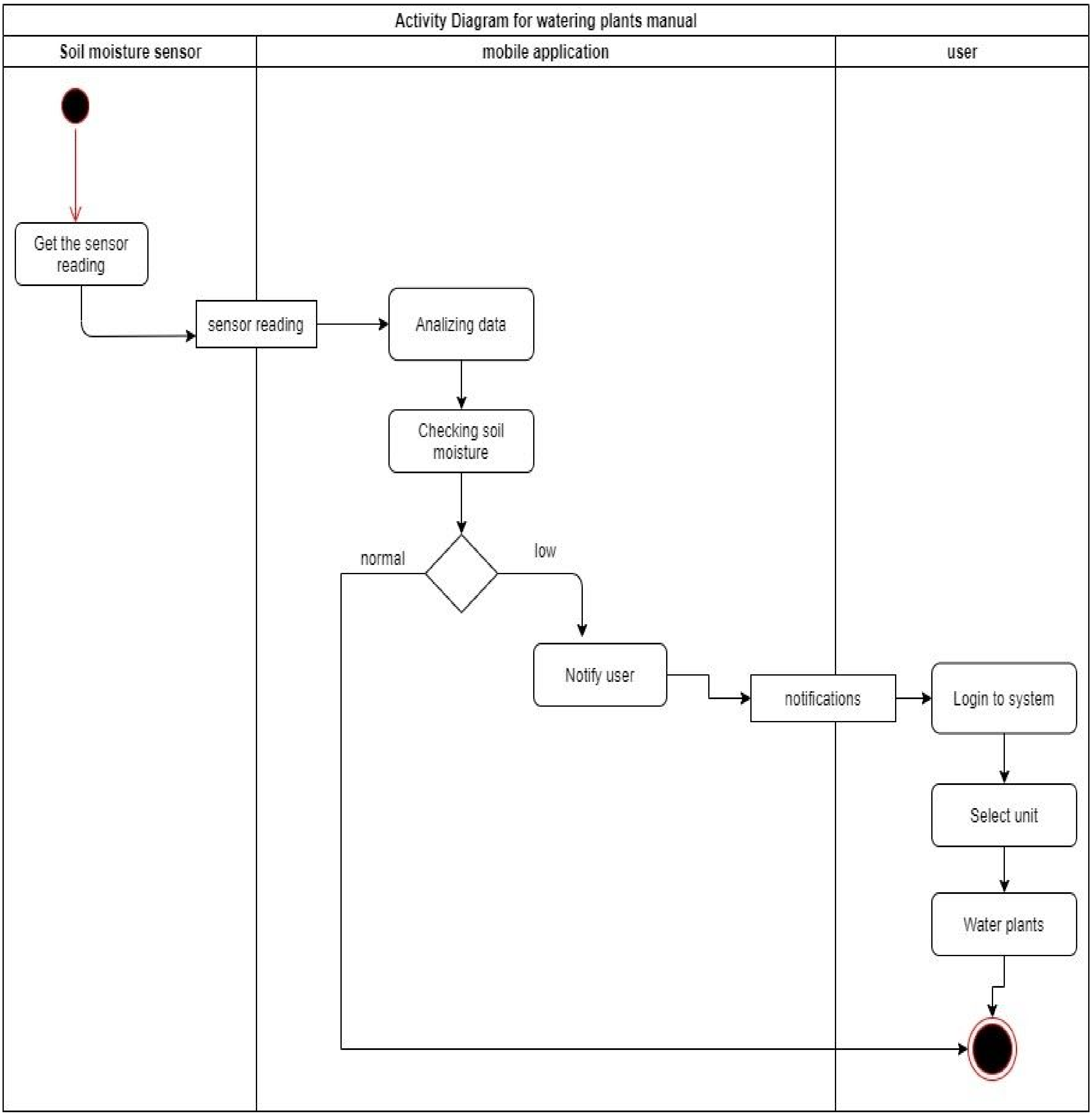
Activity diagram for chat module



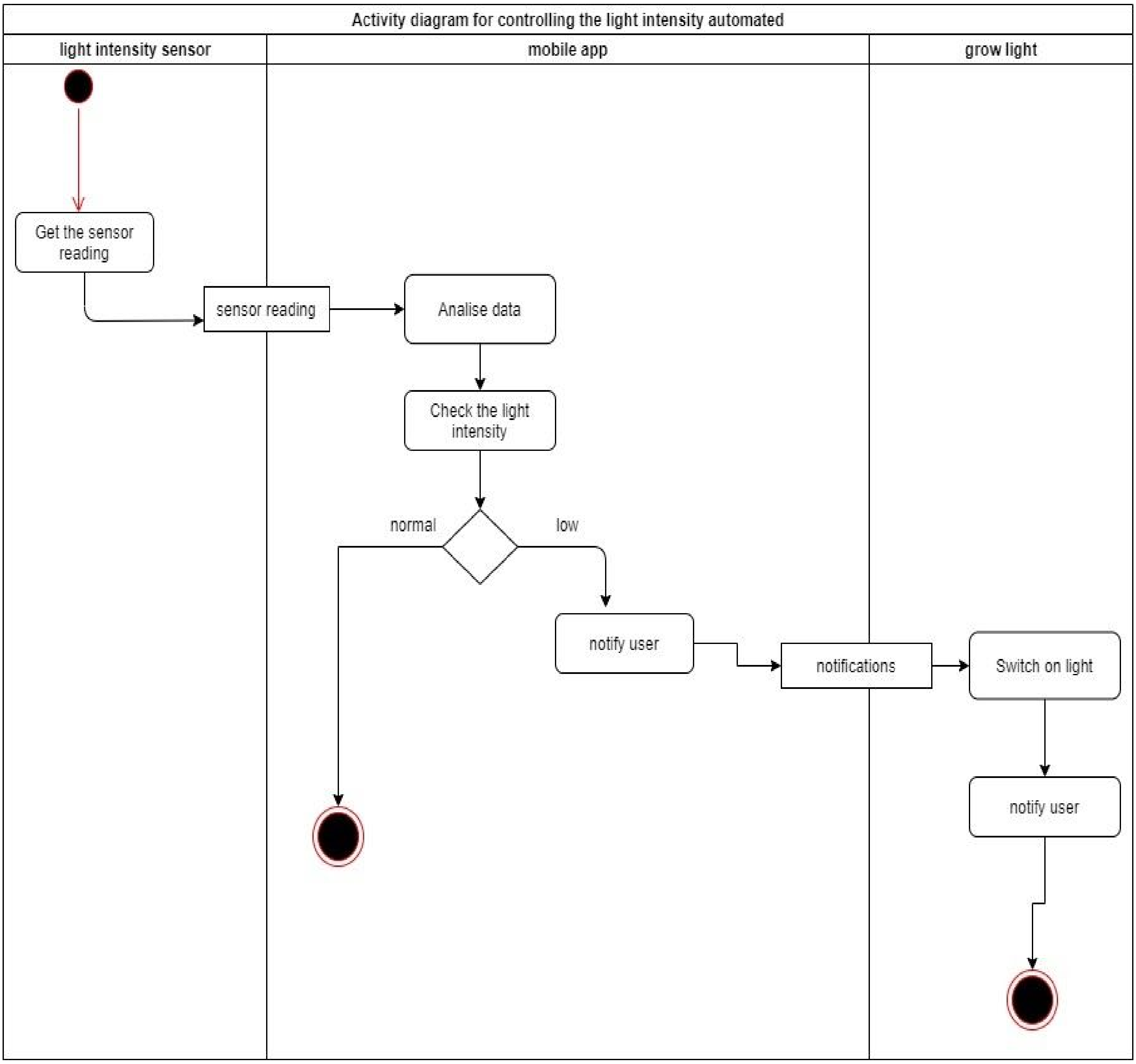
Activity diagram for watering plants automated



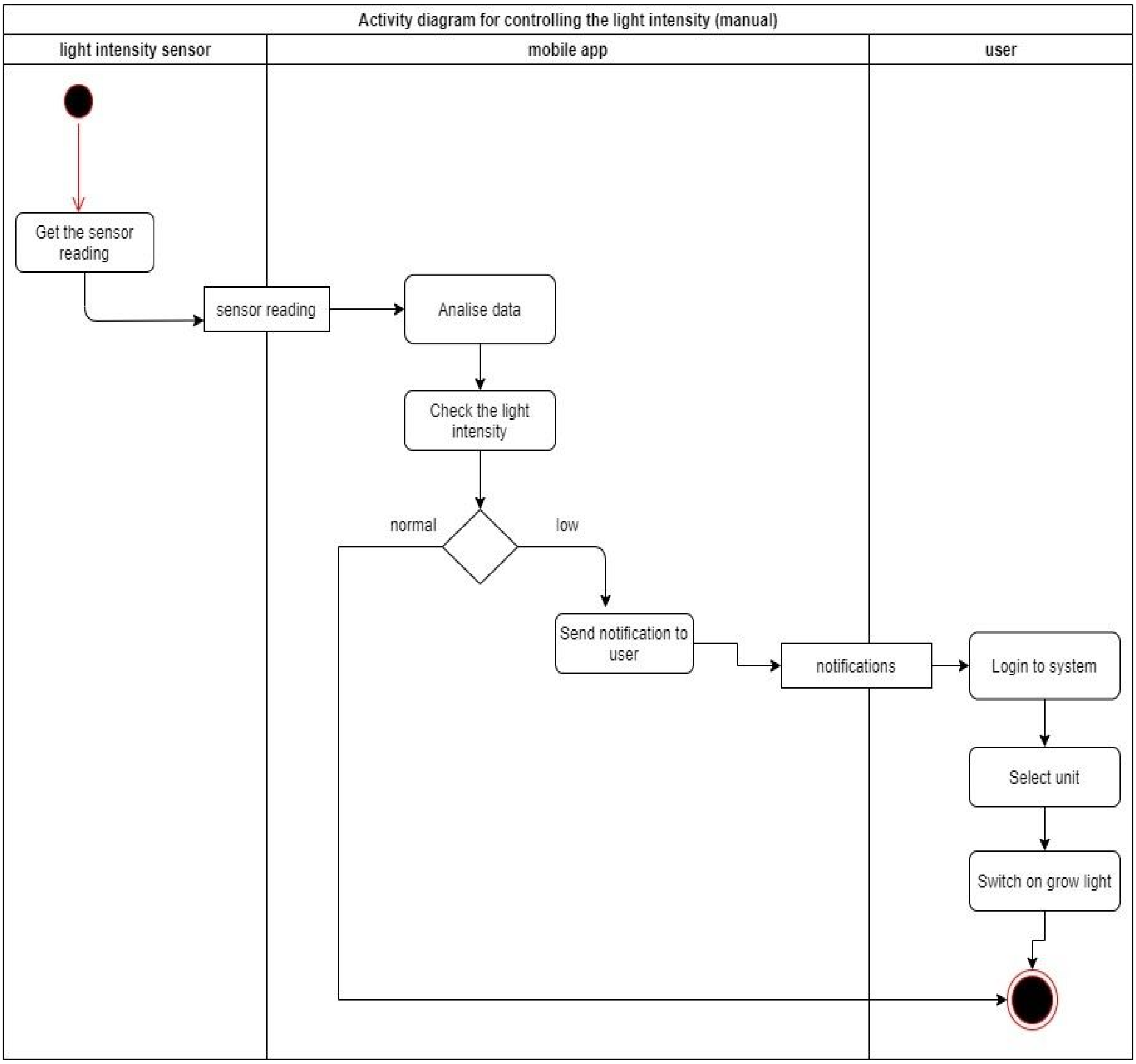
Activity diagram for watering plants manual



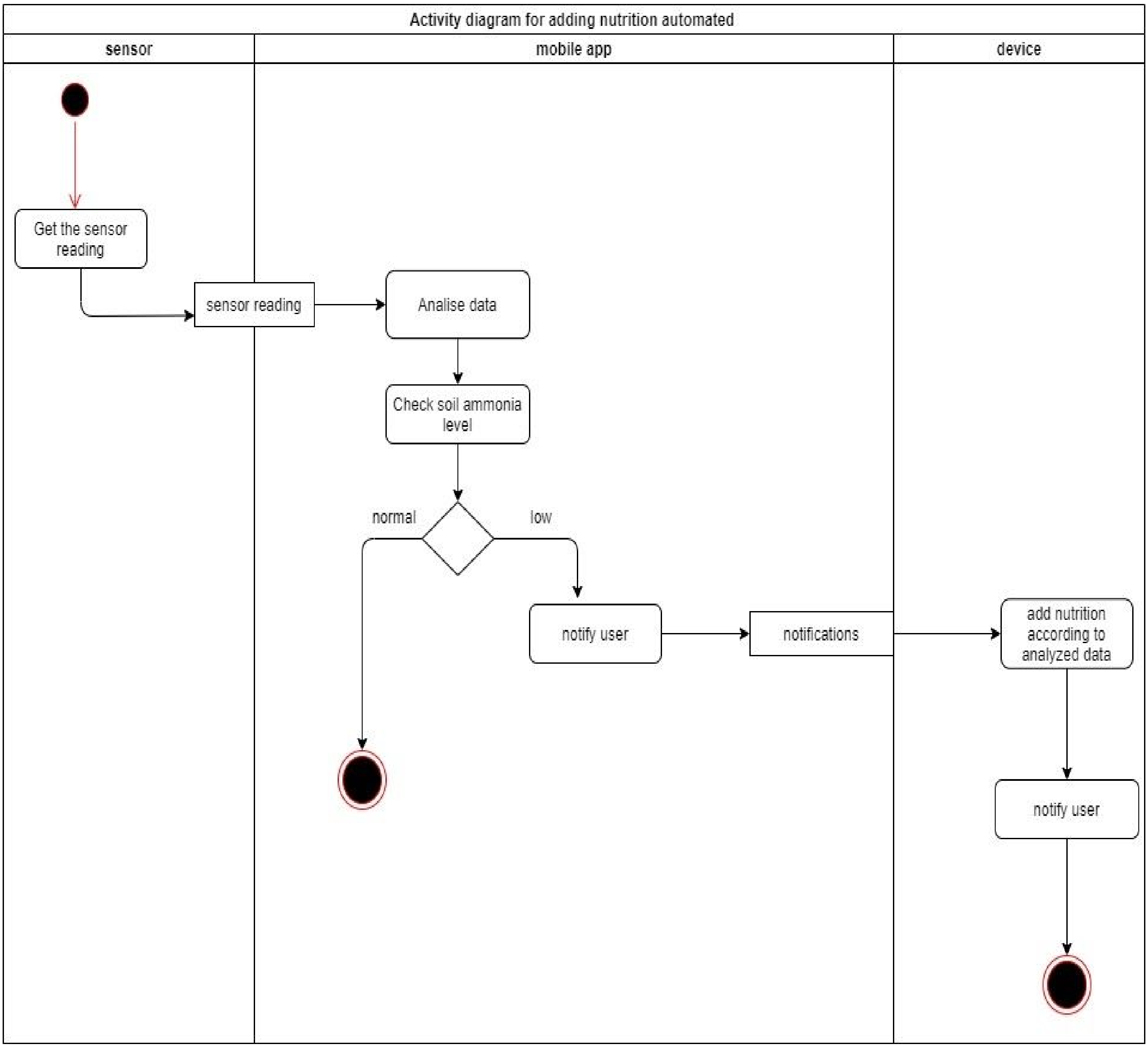
Activity diagram for controlling light intensity automated



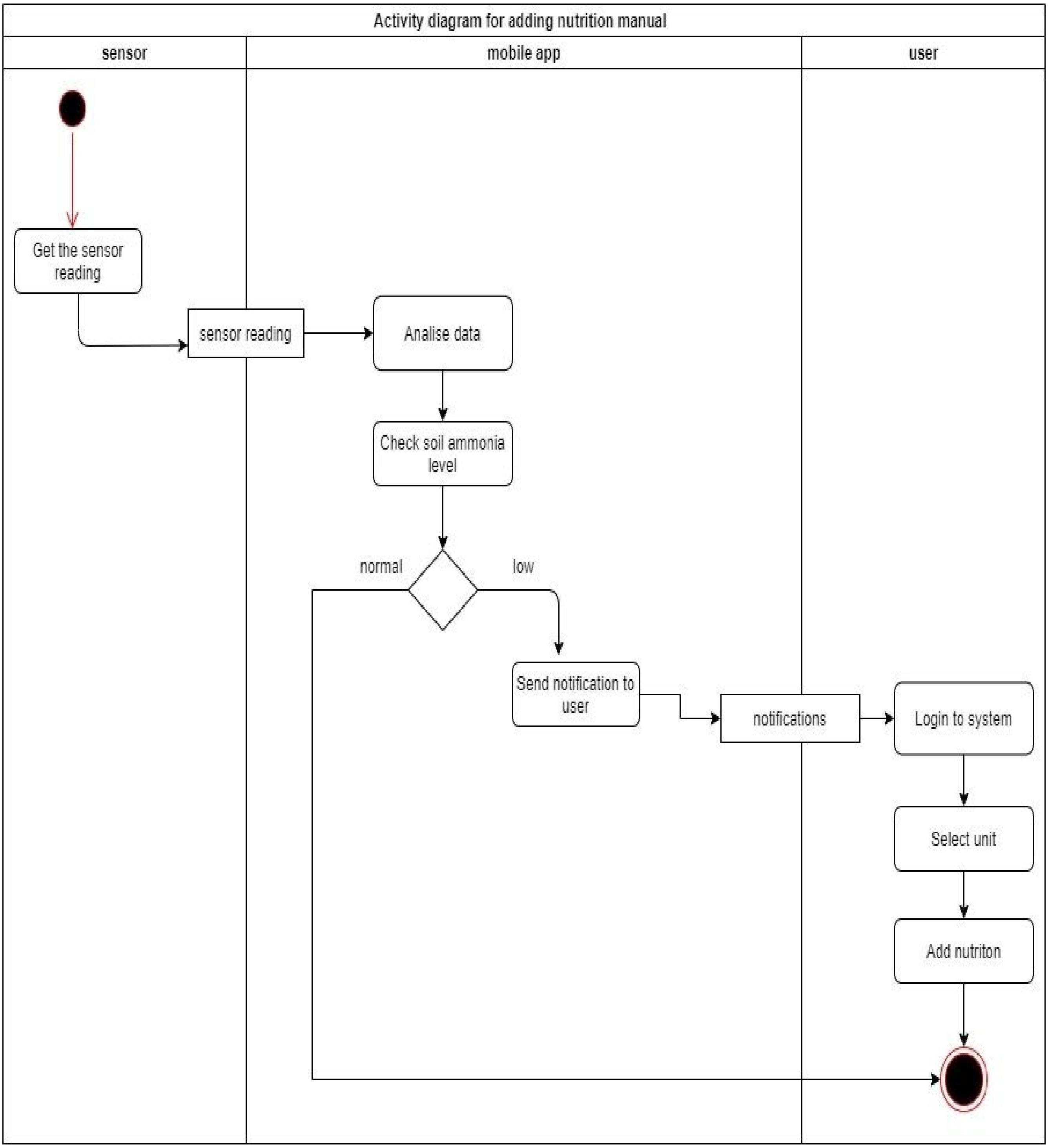
Activity diagram for controlling light intensity manual



Activity diagram for adding nutrition automated

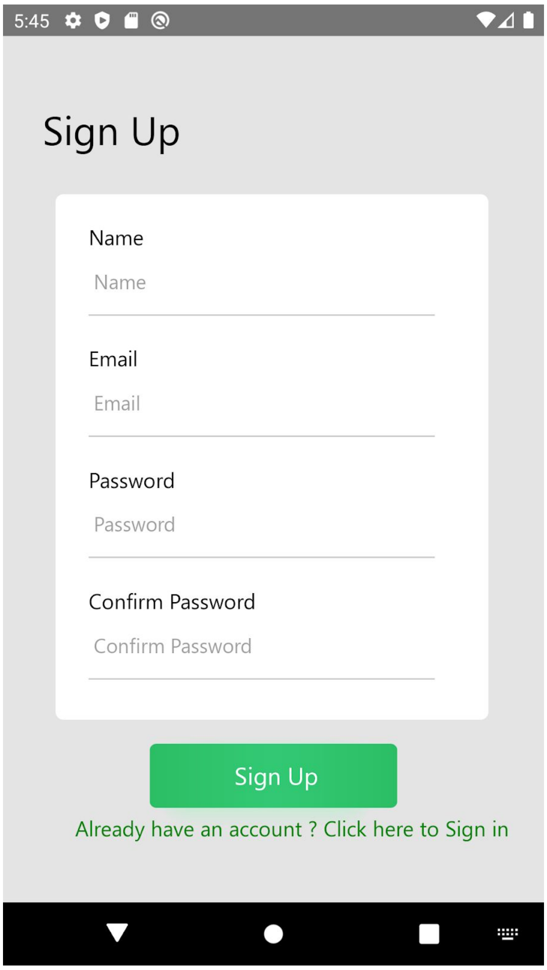


Activity diagram for adding nutrition manual

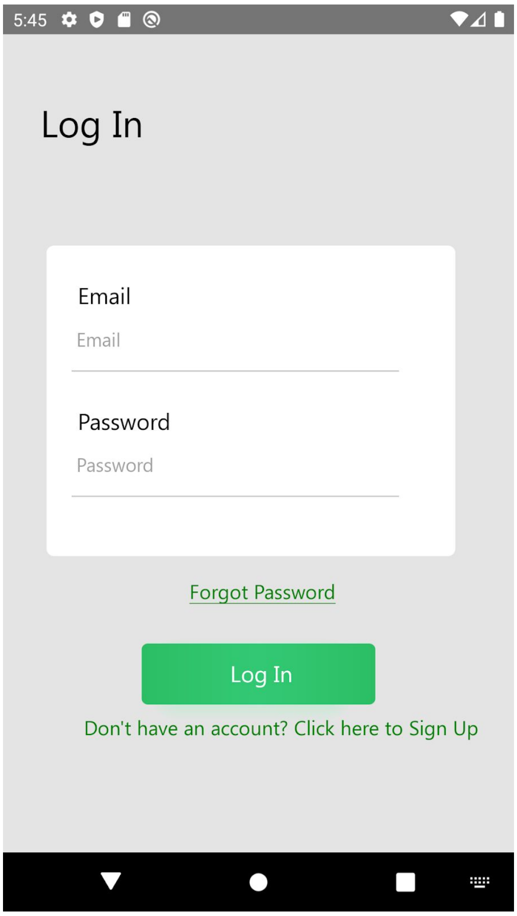


11.User Interfaces

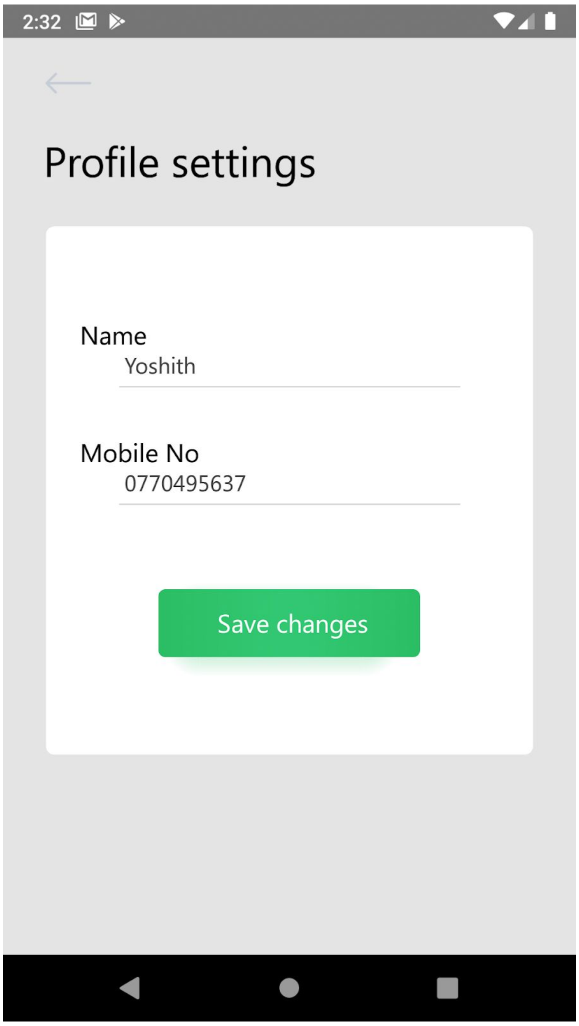
Sign Up screen

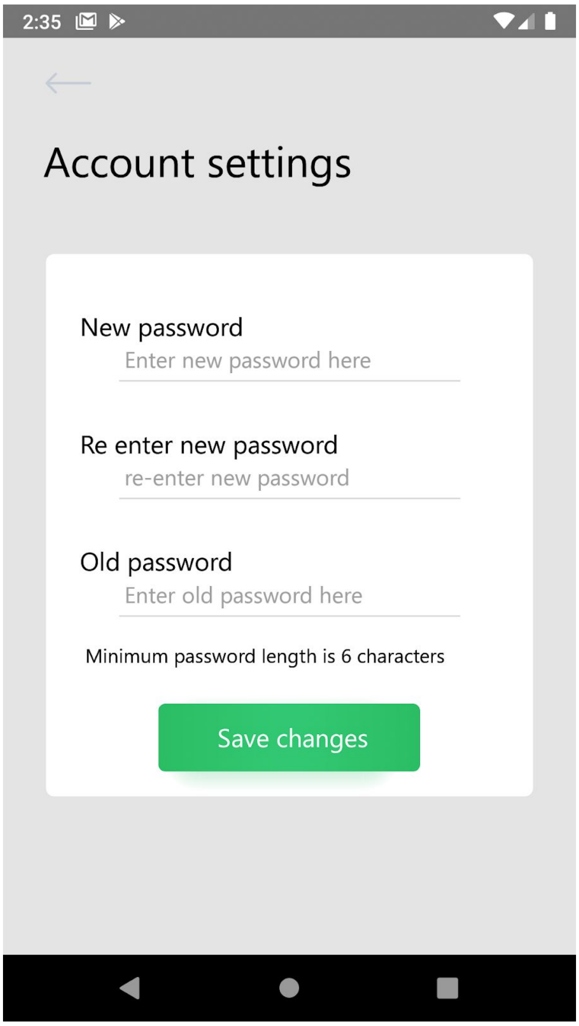


Login screen



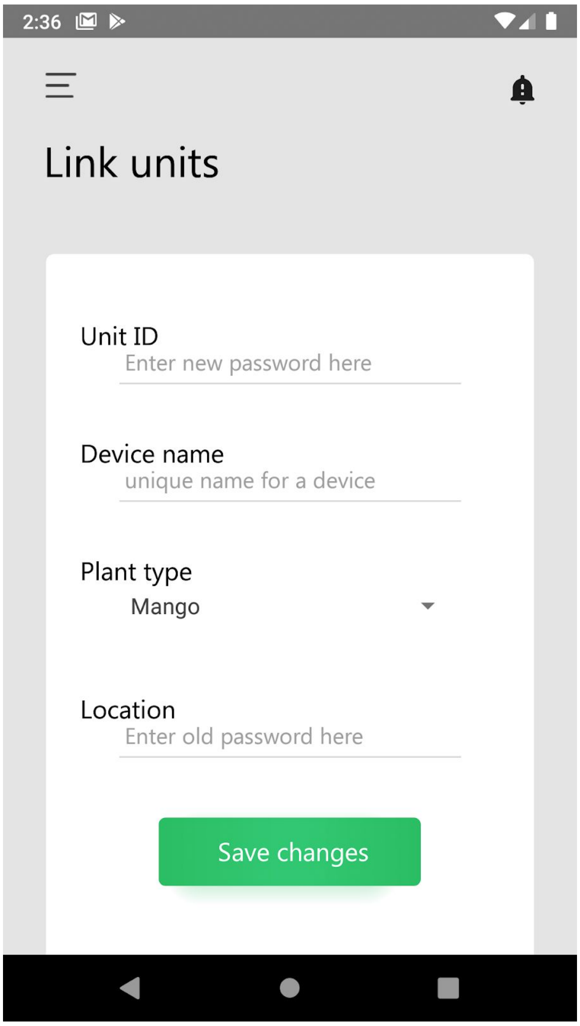
Splash screen





Profile Settings screen

Account Settings screen

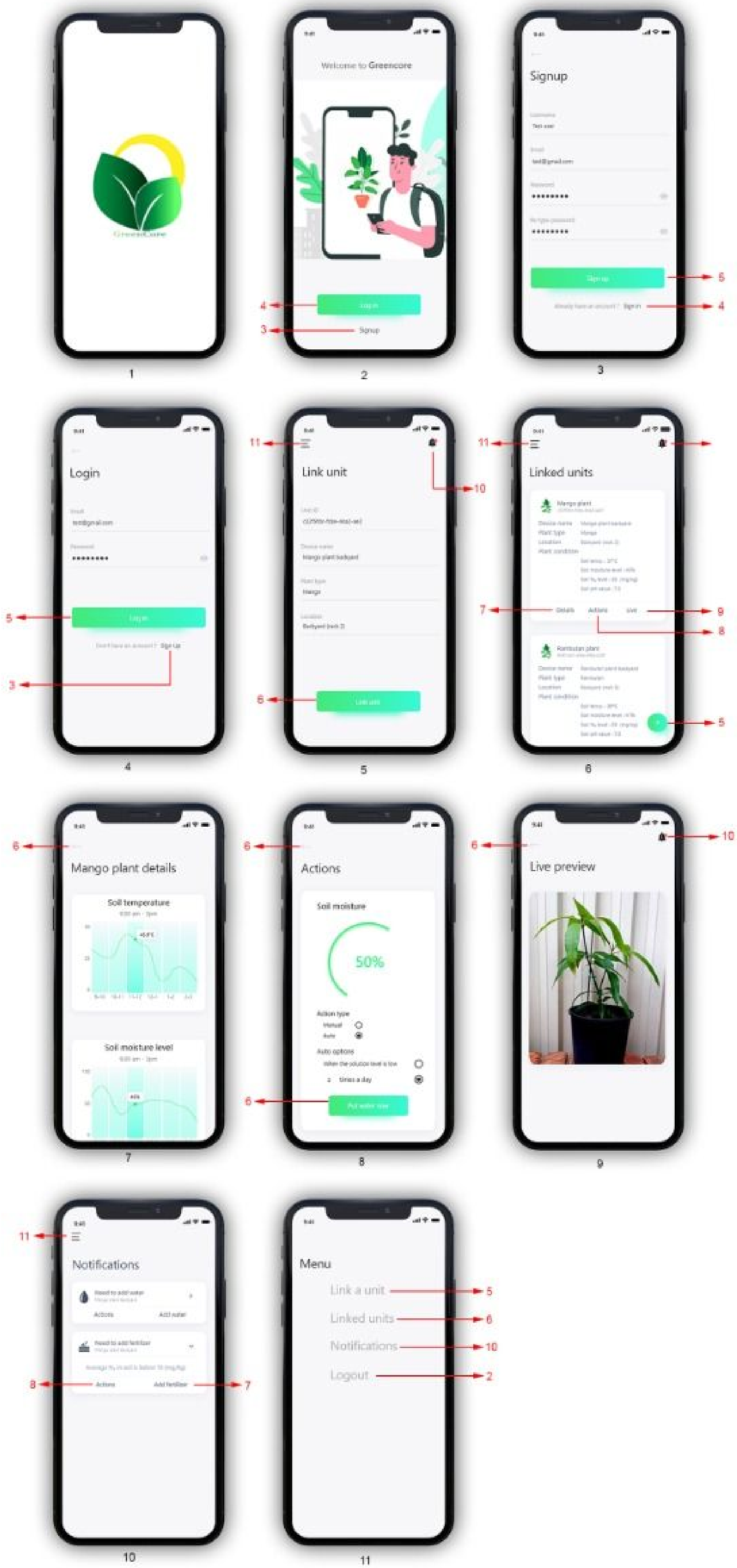


Link Units screen

Plant Details screen

- Mobile App has used a Green friendly theme.
- Comfortable for the user's eye and it is suitable with the concept of Green-core app.

UI Flow Diagram - Mobile App



Welcome to Greencore
Signup
Login
Link unit
Linked units
Mango plant details
Soil temperature
Soil moisture level
Actions
Soil moisture
50%
Live preview
Notifications
Menu
Link a unit
Linked units
Notifications
Logout

12.Main deliverables of the system

1. Complete working Mobile Application and Web Application and source code
2. Complete Software Requirement Specification for both Mobile app and the Web app
3. Complete Software Requirement Specification(SRS)
4. User manual for the IoT device
5. License of the software
    - React JS - standard MIT License
    - Node Js - MIT license
    - ExpressJs - MIT License
    - Arduino Mega - Licensed under a Creative Commons Attribution Share-Alike license
    - ESP8266 - GNU Lesser General Public License v2.1
    - MongoDB - Server Side Public License (SSPL) v1.0

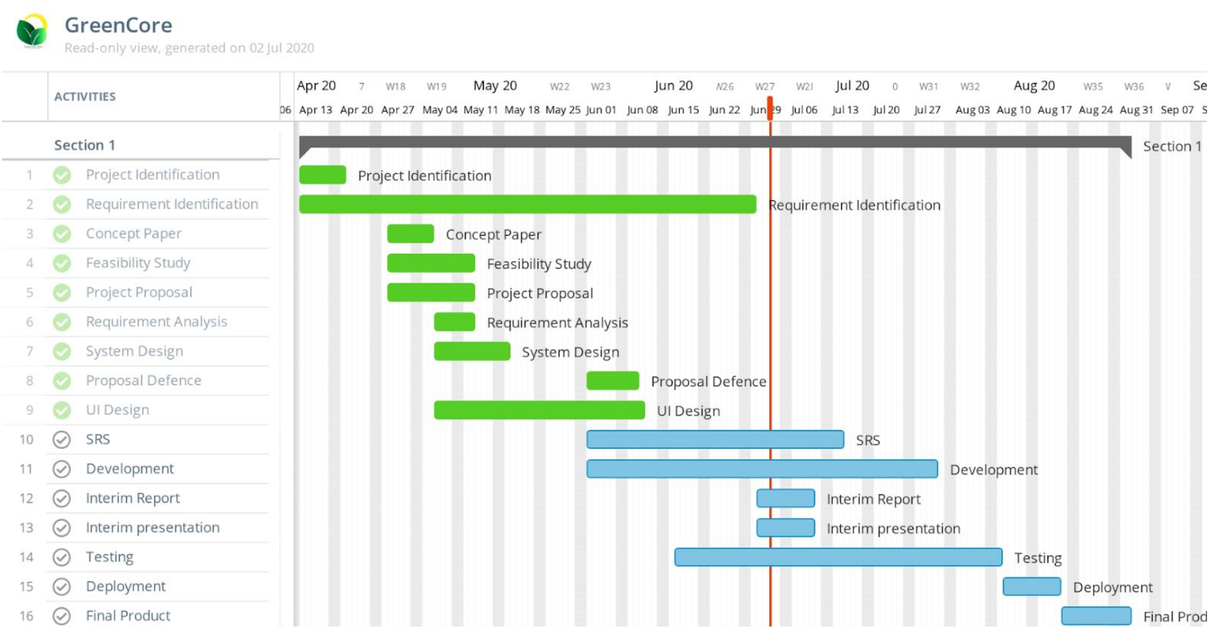
13.The Project Plan

Start Date : 21/04/2020
End Date : 30/08/2020

Main 5 milestones,

1. System Design (both Mobile and Web)
2. Mobile Application
3. Web Application
4. IoT Device
5. Deployment

and will be completed parallelly. A milestone will consist of several sprints



14.References

- Scrum, “What is Scrum?.” Scrum.org. https://www.scrum.org/resources/what-is-scrum (accessed May. 10, 2020).
- React Native, “A framework for building native apps using React.” ReactNative.dev. https://reactnative.dev/(accessed May. 10, 2020).
- Click and Grow, “The Click & Grow Smart Indoor Gardens are the most advanced and easiest indoor gardening solutions.” Click and Grow. https://asia.clickandgrow.com (accessed May. 10, 2020).
- Jeffrey L Whitten and Lonnie D Bentley, “Feasibility Analysis and the System Proposal,” in System Analysis and Design Methods, Pard Duchont, NY, USA, 2007, ch. 11, sec. x, pp. 412–437.
- “IEEE REFERENCE GUIDE,” Dec. 11, 2018. [Online]. Available: https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf. [Accessed: May. 9, 2020].
- “The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game,” Jul. 2013. [Online]. Available: https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf. [Accessed: May. 9, 2020].

15.Declaration

We as members of the project titled Green Core, certify that we will carry out this project according to guidelines provided by the coordinators and supervisors of the course as well as we will not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university. To the best of our knowledge and belief, the project work will not contain any material previously published or written by another person or ourselves except where due reference is made in the text of appropriate places.

Name	***Signature***
(i) M.M.H Tharushika	
(ii) H.U.K.M Pabasara	
(iii) J.A.N.C Niroshana	
(iv) B.A Medawatta	
(v) K.C Gamage	
(vi) K.S.A Ahamed	