

**MINISTRY OF EDUCATION AND TRAINING**

FPT UNIVERSITY

Capstone Project Document

**Warning system on parameters of the devices in the family**

|  |  |
| --- | --- |
| **Group 1** | |
| **Group members** | **Võ Phúc Hải – SE61582**  **Tôn Thất Minh Trí – SE61679**  **Nguyễn Phú Ngọc Trai – SE61060**  **Nguyễn Kim Cường – SE90180** |
| **Supervisor** | **Nguyễn Đức Lợi** |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | **APMS** |

-Ho Chi Minh City, **Sep 5th 2017**

Table ofContents

[A. Introduction 4](#_Toc497223523)

[Project Information 4](#_Toc497223524)

[Introduction 4](#_Toc497223525)

[Current Situation 4](#_Toc497223526)

[Proposed Solution 4](#_Toc497223527)

[Functional Requirements 5](#_Toc497223528)

[Roles and Responsibilities 5](#_Toc497223529)

[Conclusion 5](#_Toc497223530)

[B. Software Project Management Plan 7](#_Toc497223531)

[Problem Definition 7](#_Toc497223532)

[1.1 Name of this Capstone Project 7](#_Toc497223533)

[1.2 Problem Abstract 7](#_Toc497223534)

[1.3 Project Overview 7](#_Toc497223535)

[Project organization 10](#_Toc497223536)

[2.1 Software Process Model 10](#_Toc497223537)

[2.2 Roles and responsibilities 11](#_Toc497223538)

[2.3 Tools and Techniques 12](#_Toc497223539)

[Project management plan 12](#_Toc497223540)

[Coding convention 14](#_Toc497223541)

[C. Report No. 3 Software Requirement Specification. 14](#_Toc497223542)

[1. User requirement specification. 14](#_Toc497223543)

[2. System requirement specification. 14](#_Toc497223544)

[2.1 External interface requirement. 14](#_Toc497223545)

[2.2 System Overview Usescase 16](#_Toc497223546)

[2.3 List of usecase. 16](#_Toc497223547)

[3. Software System Attribute. 34](#_Toc497223548)

[3.1 Usability. 34](#_Toc497223549)

[D. Report No. 4 Software Design Description. 35](#_Toc497223550)

[1. Design Overview. 35](#_Toc497223551)

[2. System architectural design. 35](#_Toc497223552)

[2.1 Web service architecture design. 35](#_Toc497223553)

[3. Component diagram. 36](#_Toc497223554)

[4. Detail description. 36](#_Toc497223555)

[4.1 Class Diagram. 36](#_Toc497223556)

[5. Interface. 36](#_Toc497223557)

**List of Table**

Table 1: Definitions, Acronyms and Abbreviations 3

Table 2: General Roles and Responsibilities of Member 5

Table 3: Central processor 9

Table 4: Send/ receive data in a Zone. 9

Table 5: Detail Roles and responsibilities of Member 12

Table 6: Tools and Techniques use in project 12

Table 7: Project management plan 13

**List of Figures**

Figure 1: Iterative and Incremental development 11

**Definitions, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| **CCU** | Central Control Unit |
| **HTTP** | Hypertext Transport Protocol |
| **HTML** | Hypertext Markup Language |
| **IOT** | Internet of Thing |

Table 1: Definitions, Acronyms and Abbreviations

# Introduction

## Project Information

* Project name: **Warning system on parameters of the devices in the family**
* Project code: **APMS**
* Project type: **Internet of Thing (IOT)**
* Start date: **6th-Sep-2017**
* End date: **8th-Dec-2017**

## Introduction

The system will use sensors set up on the devices in house and sends parameters of these devices over the internet to report for users via mobile devices. Our system will auto check devices and send an error message to users when problems, errors occur from electrical appliances in house.

## Current Situation

The most garden dendrobium is operating by people. They can be the farmer or the boss. In this way they follow their experience to take care the dendrobium orchid. The traditional base on human experience. They do not have any machine to measure exactly parameter of environment like humidity and temperature.

There are many other third party have implemented a device fit our system require, but the price is high and all of the vendor are far away from Viet Nam. Some of them does not support or even sell their product outside their country. So we decided to research some embedded system and come up with do-it-yourself solution.

There are some Viet Nam companies which have provided IoT agriculture devices such as : Bach Khoa, HACHI… They also provide IoT system for clean agriculture with the product reached Viet GAP or further is Global GAP.

## Proposed Solution

Many modern garden in develop country they apply IOT in to their farm. The result is quantity is double. The cost for water, worker and energy was reduce very much.

This solution is provide a system to tracking the garden. This is applied IOT which can control from anywhere have the Internet.

The mobile application also provides the recommendation for user. How to use fertilizer suitable with the weather and the stage in time life of orchid.

This solution use industrial device with high quality and reliability. We provide a HMI monitor to control at the garden and an application for android mobile.

## Functional Requirements

Functional requirements of the system are listed as below:

* Tracking component:
* Define temperature and humidity.
* Compare the current situation with the best situation for orchid.
* HMI component:
* Display the temperature and humidity.
* Control the water provider.
* Notification mobile application component:
* Speaker.
* Water provider component:
* Automation operate.
* Control by mobile application and HMI.

## Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full name** | **Role** | **Position** | **Contact** |
| 1 | Nguyễn Đức Lợi | Project manager | Supervisor | [loind@fpt.edu.vn](mailto:loind@fpt.edu.vn) |
| 2 | Nguyễn Quang Huy | Developer | Leader | [Huynqse61525@fpt.edu.vn](mailto:Huynqse61525@fpt.edu.vn) |
| 3 | Nguyễn Tuấn Sang | Developer | Member | [Sangntse61499@fpt.edu.vn](mailto:Sangntse61499@fpt.edu.vn) |
| 4 | Mai Thanh Hiếu | Developer | Member | [Hieumtse61559@fpt.edu.vn](mailto:Hieumtse61559@fpt.edu.vn) |
| 5 | Huỳnh Quang Đạt | Developer | Member | [Dathqse61783@fpt.edu.vn](mailto:Dathqse61783@fpt.edu.vn) |
| 6 | Trần Hồng Sơn | Developer | Member | [Sontqse61479@fpt.edu.vn](mailto:Sontqse61479@fpt.edu.vn) |

Table 2: General Roles and Responsibilities of Member

## Conclusion

* Research Configure and commands a Arduino, **ESP8226 NodeMCU**.
* Researching how to communicate HMI and Raspberry PI 3.
* Communication technique: How about communicate interface; SPI, I2C, UART
* Researching how to pulse width modulation for Pumper.
* Researching software architecture of Raspberry Pi.
* Researching how to operate ESP8226 NodeMCU with sensor.
* Design and implement UI of HMI.
* Design and Implement the Voltage regulator circuit, the display and alarm circuit.
* Use software in design PCB, Schematic such as OrCAD, Eagle, Proteus.
* C , C++ or Java and Android Studio, Python.
* Documentation.

# Software Project Management Plan

## Problem Definition

### 1.1 Name of this Capstone Project

* Official name: Dendrobium orchid care solution.
* Vietnamese name: Giải pháp chăm sóc lan rừng dendro.
* Abbreviation: OCS.

### 1.2 Problem Abstract

* In the present day orchid growing is very popular but still has many difficulties in doing so. Orchid growers have to spend a lot of effort, time and cost but not sure to have a good result as expected. Therefore, there should be a system that can help the orchid grower solve those problems.
* Orchid growers will determine the status of plants difficulty such as how much water should be planted, how many stools...
* They cannot know how the Orchid status when they go away.

### 1.3 Project Overview

#### 1.3.1 Current Situation

By research other systems, we found some problem current situation below :

* **Integrate with IoT hardware** : Our team did not have too much experience with hardware so that we much research about hardware, mainboard circuit, connecting device with the internet.
* **Orchid information :** The information about the orchid is very important in this project. So we take a lot of time to collect information.
* **Server balancing :** Cause of the cloud server play as collector of system. Then we need to calculate balancing for server to avoid crashing.
* **Human mistake on central controller :** Smart technology is still new knowledge with Vietnamese. Therefor we need to have implement manual instruction in our product for usability requirement.
* **Orchid have long time life:** In fact the orchid needs 6 to 12 months to complete a cycle of time life. So in 4 months of this project we need to analyzing the best for test case.

#### 1.3.2 The proposed system

* After research about Internet of Thing (IoT) concept, we found out that a control center application install into a hardware with network module is capable of interaction between our system and sensors devices. In this document, we use a control center run on Android operation system which have wifi network module. We using flatform CloudMQTT to support user activate and control their control center with a mobile application.
* This system will be include sensors that read humidity and temperature and send data to server . From the results obtained we can develop a processor that providers solutions for orchids growers have the most appreciate way to irrigate or fertilize the orchids without wasting much time , energy and effort wastage.
* We colllect information of orchid form many resource. We build a database to store. This database can maintain and access to provide data for mobile application.
* We build a mobile application. This application access to database and compare with the information from the realtime to give the suitable recommendation.

#### 1.3.3 The Boundaries of the system

The system apply with farm have electric devices and internet that users want to control their farm by smart phone, tablet. The system include:

* Allow the user control devices by control panel screen ( Smart phone, tablet, HMI).
* Allow the user check temperature , humidity.
* Allow the user control devices by groups and schedule them active automatically.
* Give the recommendation about the fertilizer for the user.

The system does not include :

* The mobile application running on other platform except Android OS.
* The feature to maintain hardware.
* The IoT private protocol to against hacker attack.

#### 1.3.4 Future Plans

* There are no perfect solutions to problems, as well as there are no perfect systems. With the inexperience of our team members and the time constrains, our proposed solution and project contains many issues. Below are the problems encountered in this project:
* **Hardware Knowledge**: We are inexperienced with hardware. All the hardware components chosen to be used in this project is based on our familiar with them, or based on the shortest time we need to learn how to use them. So they are only the most appropriate, not the best choice for the project.
* **Security**: Currently, there is few possible problems encountered with RF, as RF is vulnerable to replay attack.
* **Server crash**: All the needed data for the app is stored in the server. So if server crash, all the devices cannot working properly.
* Our future plan is try to solve these problems one by one.
* We will also try to working to implement a sensor to read data , handle situation with suitable, specific options.

#### 1.3.5 Development Environment

##### 1.3.5.6 Hardware requirement



|  |  |
| --- | --- |
| * Component | * Hardware |
| * Mainboard | * Raspberry Pi 3 |
| * Communication | * USB Cable |
| * Sensor Devices | * SHT10 version 2 |
| * Motors | * Pumper |
| * Receive/Transfer data | * RF 315Mhz |
| * Wi-Fi module | * ESP8266 nodeMCU |

Table 3: Central processor

|  |  |
| --- | --- |
| * Component | * Hardware |
| * Mainboard | * Arduino Mini |
| * Sensor Devices | * SHT10 version 2 |
| * Receive/Transfer data | * RF 315Mhz |
| * Notification | * LED |

Table 4: Send/ receive data in a Zone.

##### 1.3.5.7 Software requirement

* Windows XP/7/8/10: operating system for developing and deploying.
* Arduino IDE: Programming main source code.
* Visual Studio: Programing Desktop Application.
* Proteus 8: Used to drawing PCB board.
* Draw.io: Used to create models and diagrams.
* C/C++: used for embedded module.
* C#: windows application.
* Python : receive / send data .

## Project organization

### 2.1 Software Process Model

* This project is developed under Iterative and incremental development model. We apply customized Iterative and incremental development model to capable with current situation in our team. We choose this model because of the following reasons:
* We are still inexperienced and by develop the system through iterations (repeated cycles) and incrementally (in small portions of time), we can learn from our mistakes and apply that knowledge on the next iteration.
* We are researching and developing the system at the same time, so using this model allow us more flexibility to adapt to changes.
* Working with embedded system hides a lot of problems that are unknown in the planning phase until it is too late. With Iterative and incremental development model, we test the system in small portion at a time, therefore reduce risk and build a feature rich and robust system.
* 

Figure 1: Iterative and Incremental development

### 2.2 Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| * No | * Full name | * Team Role | * Responsibilities |
| * 1 | * Nguyễn Đức Lợi | * Supervisor, Project Manager | * Specify user requirement * Advisor for ideas and solutions * Give out techniques and business analysis support |
| * 2 | * Nguyễn Quang Huy | * Team Leader, BA Developer, Tester | * Create test plan * Clarifying requirements * Prepare document * Coding * Testing |
| * 3 | * Nguyễn Tuấn Sang | * Team Leader, BA, Developer, Tester | * Create test plan * Diving task for team member * Managing process * Clarifying requirements * Prepare document * Testing |
| * 4 | * Trần Hồng Sơn | * BA, Developer, Tester | * Create test plan * Clarifying requirements * Prepare document * Coding * Testing |
| * 5 | * Mai Thanh Hiếu | * BA, Developer, Tester | * Create test plan * Clarifying requirements * Prepare document * Coding * Testing |
| * 6 | * Huỳnh Quang Đạt | * Developer, Tester | * Create test plan * Clarifying requirements * Prepare document * Coding * Testing |

Table 5: Detail Roles and responsibilities of Member

### 2.3 Tools and Techniques

|  |  |
| --- | --- |
| * Tools | |
| * Operating System | * Windows 7/8/10 & Windows Server 2012 R2 |
| * Developing tool | * Arduino IDE, Visual Studio, * Notepad++, Sublime Text 3 |
| * Source Control | * GitHub |
| * Managing Database | * CloudMQTT |
| * Communication tool | * Facebook, Skype, Slack |
| * Model and Diagrams tool | * Draw.io , Software Ideas Modeler, Fritzing, Creately.com |
| * Testing | * Postman |
| * Programing language | |
| * Arduino | * C++ |
| * Application | * C#, JSON |
| * Webserver | * PHP, HTML, JSON, CakePHP 3.4 |

Table 6: Tools and Techniques use in project

## Project management plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | Scope | Evaluation | Activities | Estimated Duration | Assign Responsibilities |
| Initial Iteration | Initial team workplace and identify project scope | A working team environment | * Set up Git Repository with Gitflow * Set up Slack * Set up Trello | 5 days | SangNT,  HieuMT,  HuyNQ,  SonTH,  DatHQ |
| Iteration 1 | Identify boundaries of the system, planning software and hardware. Create a proof-of-concept prototype. | Report 1, Report 2 and a proof-ofconcept prototype | * Introduction document * Software and Hardware Project Management Plan document * Proof-ofconcept prototype | 15 days | SangNT,  HieuMT,  HuyNQ,  SonTH,  DatHQ |
| Iteration 2 | Produce an architectural prototype | Report 3, Report 4 and an architectural prototype | * Software and Hardware Requirement Specification document * Software and Hardware Design Description document * Architectural prototype | 15 days | SangNT,  HieuMT,  HuyNQ,  SonTH,  DatHQ |
| Iteration 3 | Build the product (up to beta release) | Report 5 and a working product (beta release) | * System Implementation and Test document * PCB * Application * Web API Server | 15 days | SangNT,  HieuMT,  HuyNQ,  SonTH,  DatHQ |
| Iteration 4 | Finish the product (full product release) | Report 6 and the completed product | * Software and Hardware User’s Manual document * Product demonstration model | 15 days | SangNT,  HieuMT,  HuyNQ,  SonTH,  DatHQ |
| Final Iteration | Prepare for Demo Day | Final Documentation, Presentation Slide | * Final Document * Mini Document * CD contains all source code * Presentation Slide | 5 days | SangNT,  HieuMT,  HuyNQ,  SonTH,  DatHQ |

Table 7: Project management plan

## Coding convention

* C/C++ Convention
  + C/C++: Using to develop program and solve algorithm on hardware.

Summary:

* Naming Convention:
* Using Pascal case for class name.
* Using Camel case for function, variable’s name.
* The #define and global variable’s name must uppercase and separate by underscore. Ex: GLOBAL\_VARIABLE.
* Commenting Convention:
* Place the comment on the separate line with function.
* Place the comment at the end of the line, which has calculation formula.
* More details about coding conventions for C/C++ language by Google: https://google.github.io/styleguide/cppguide.html

# Report No. 3 Software Requirement Specification.

## User requirement specification.

User is a person who has accessed in system with key account. User can do the following functions :

* View garden area :
* Adjust humidity.
* Adjust temperature.
* Add plant.
* Edit plant.
* Delete plant.
* View history.
* Export log.
* Delete garden area.
* Add garden area.

## System requirement specification.

### External interface requirement.

#### User interface

The user use English language for all control mobile application.

#### Hardware interface.

Server:

Raspberry pi 3 model B :

* + RAM: 1GB.
  + CPU: 1.2 Ghz 64-bit, quad-core ARM Cortex-A53.
  + On-board wireless LAN.

Mobile smart phone:

* Chipset : Qualcomm Snapdragon 800 4 core 64-bit.
* CPU: Quad-core 2.3Ghz.
* Disk storage: 16GB.
* RAM: 2GB.
* Wi-Fi connection: Wi-Fi 802.11 a/b/g/n/ac, Wi-Fi Direct, DLNA.
* 3G band: HSDPA 850 / 900 / 1900/ 2100 MHz .

ESP8266 NodeMCU CP2012:

* Wi-Fi connection: Wi-Fi 802.11 b/g.
* UART connection : CP2012.
* 5VDC MicroUSB.
* GPIO 3.3V.

* + 1. **Software interface.**

Server 3rd party:

CloudMQTT.com.

Json API.

* + 1. **Communication protocol.**

Uses MQTT protocol for communication between :

CloudMQTT.com and ESP8266 NodeMCU.

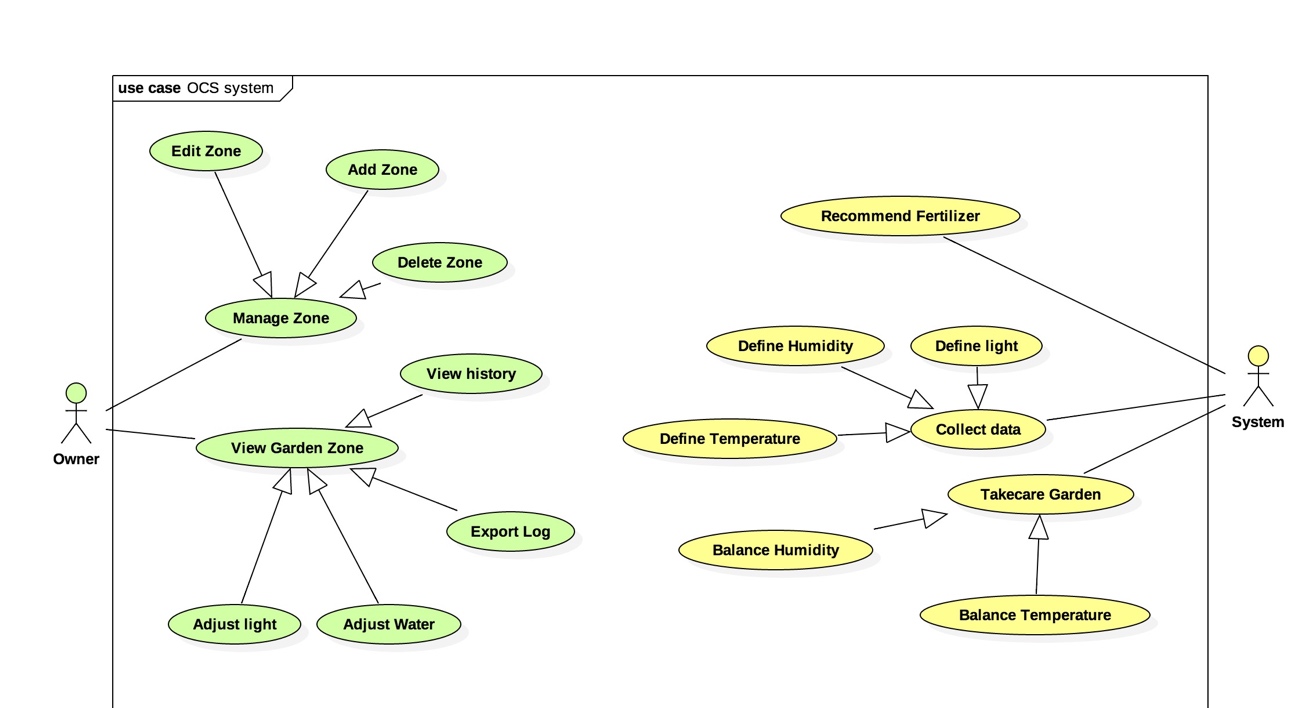
CloudMQTT.com and Raspberry Pi3.

CloudMQTT.com and Mobile application.

Uses HTTP protocol for communication between :

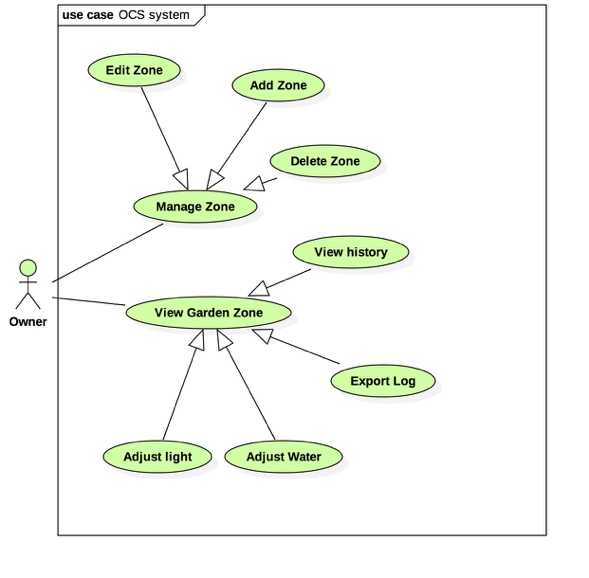
CloudMQTT.com and Mobile application.

### 2.2 System Overview Usescase

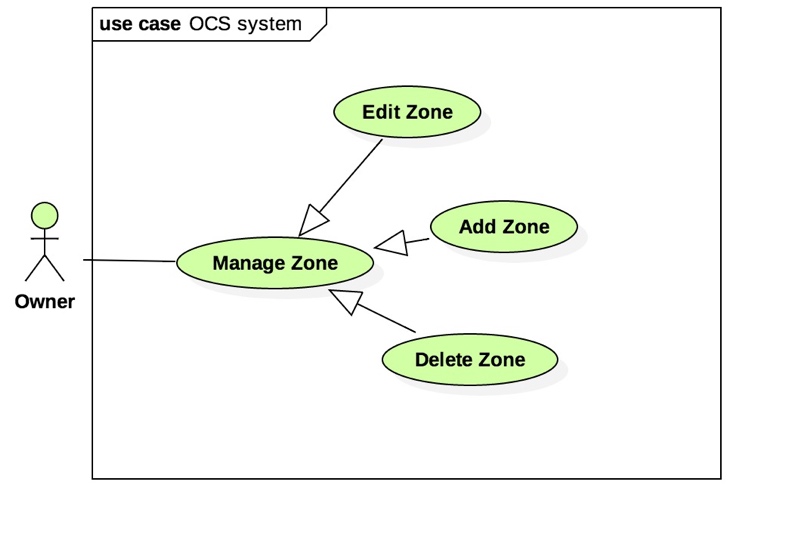


### 2.3 List of usecase.

#### <Owner> Overview usecase.

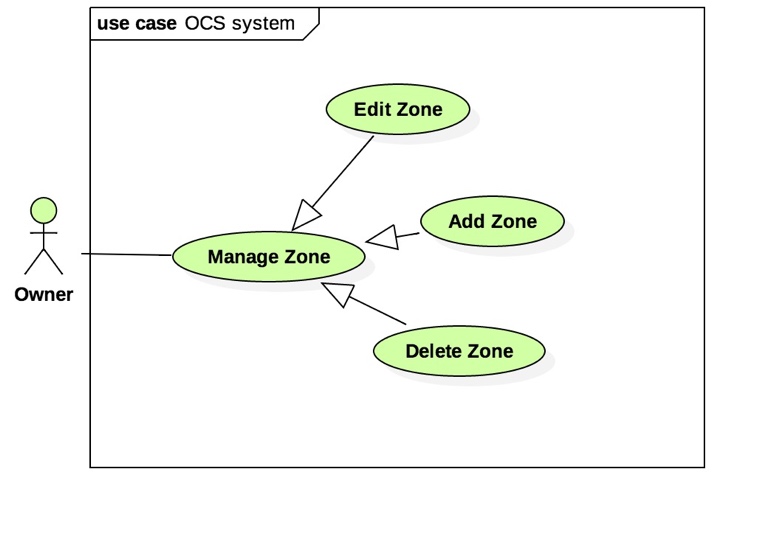


##### 2.3.1.1<Owner> Add Garden Zone.



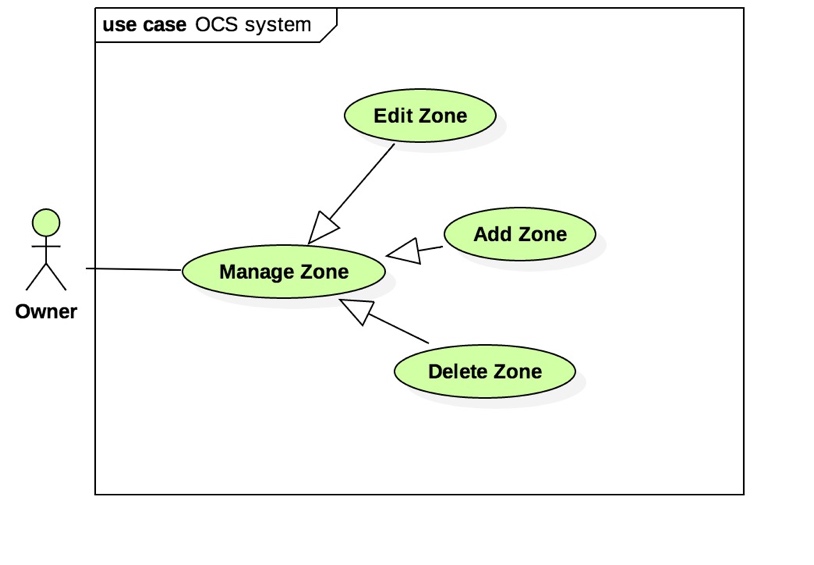
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.01** | | | |
| **Usecase No.** | UC\_A.01 | **Usecase Version** | 1.0 |
| **Usecase Name** | Add Garden Zone. | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner.   **Summary:**   * Owner can add new plant zone information in system.   **Goal:**   * Owner can add new plant zone information.   **Triggers:**   * Owner sends new information to system.   **Precondition:**   * N/A.   **Post Condition:**   * **Success:** system add new garden zone information in storage. * **Fail:** System show error message “ Can not add garden zone”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Owner goes to add new garden zone view | System require new garden zone information. | | **2** | Owner choose option for garden zone. |  | | **3** | Owner sends new garden zone information with selections. | System validates command and add new garden zone in storage. |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** |  |  | | **2** |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.1.2<Owner> Delete Garden Zone.



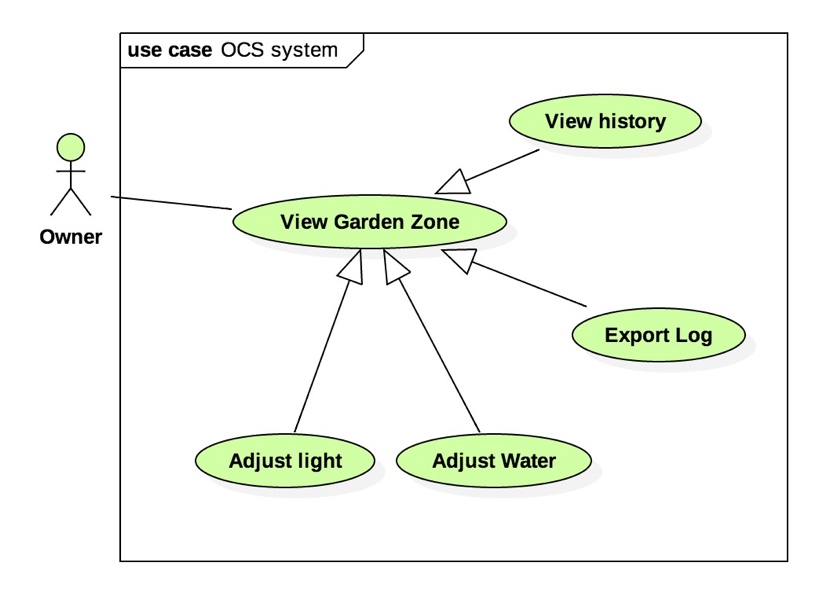
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.02** | | | |
| **Usecase No.** | UC\_A.02 | **Usecase Version** | 1.0 |
| **Usecase Name** | Delete Garden Zone. | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner   **Summary:**   * Owner can delete garden zone in system.   **Goal:**   * Owner can delete garden zone.   **Triggers:**   * Owner sends command delete garden zone to system.   **Precondition:**   * Owner zone is chosen available.   **Post Condition:**   * **Success:** system delete garden zone in storage. * **Fail:** system show error message “Can not delete garden zone”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User goes to garden zone view and choose delete | System require delete garden zone current. | | **2** | User confirm to delete current garden zone. | System confirm this command and show pop-up “ delete successful”. | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.1.3<Owner> Edit Garden Zone.



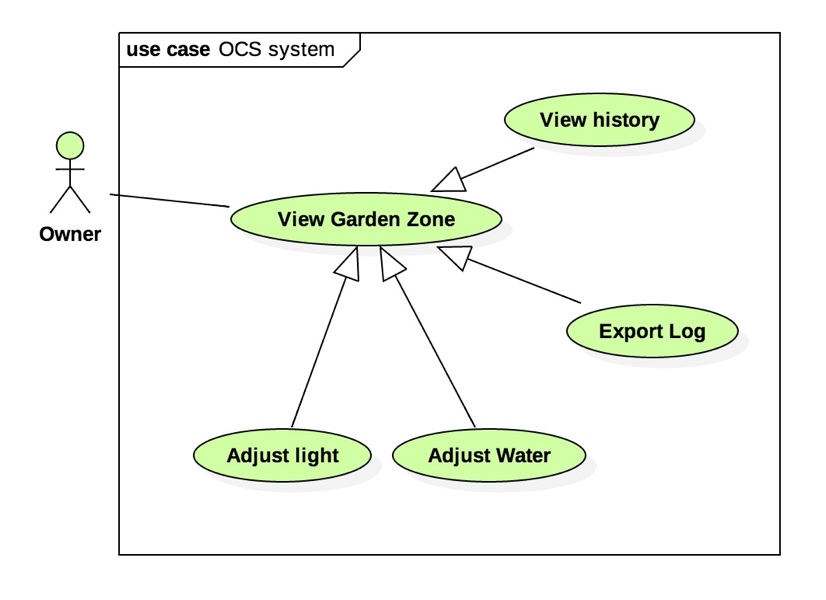
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.03** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | Edit Garden Zone | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner   **Summary:**   * Owner can edit garden zone information in system.   **Goal:**   * Owner can edit garden zone information.   **Triggers:**   * Owner send view edit zone command.   **Precondition:**   * Garden zone must be created before.   **Post Condition:**   * **Success:** show garden zone information in screen. * **Fail:** system show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User goes to menu of plant zone. | System show plant zone list. | | **2** | User |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.1.4<Owner> Adjust water.



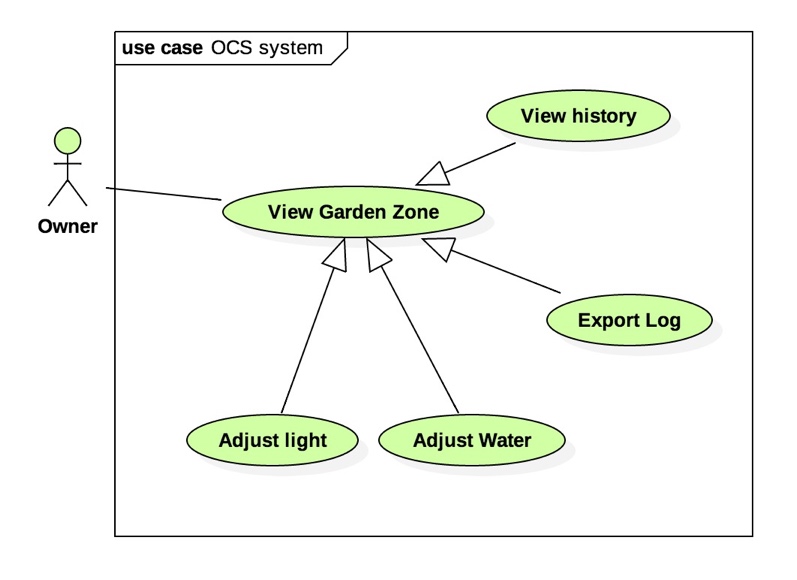
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.04** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | Adjust water | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner   **Summary:**   * Owner can adjust water to control the humidity in a specific garden zone.   **Goal:**   * Owner can adjust water to control the humidity in a specific garden zone.   **Triggers:**   * Owner sends a adjust water command to system.   **Precondition:**   * The plant in the garden zone is too dry or wet.   **Post Condition:**   * **Success:** system response the humidity is good. * **Fail:** the humidity is no change.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.1.5<Owner> Adjust Light.



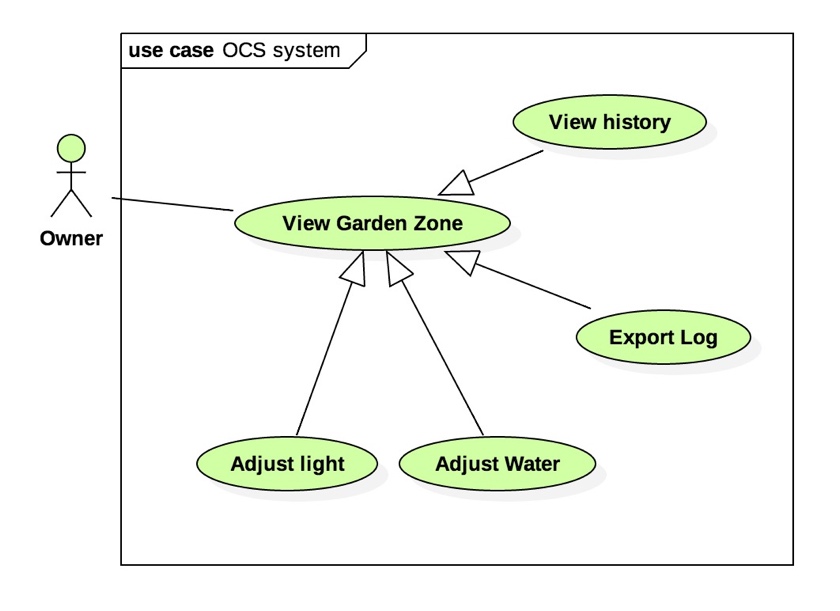
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.05** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | Adjust light. | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner.   **Summary:**   * Owner can adjust light in a the specific garden zone.   **Goal:**   * Owner can adjust light in a the specific garden zone.   **Triggers:**   * Owner send adjust light command to the system.   **Precondition:**   * The light in the garden is not suitable.   **Post Condition:**   * **Success:** system response the light is changed. * **Fail:** system not response the light is not changed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.1.6<Owner> Export log.



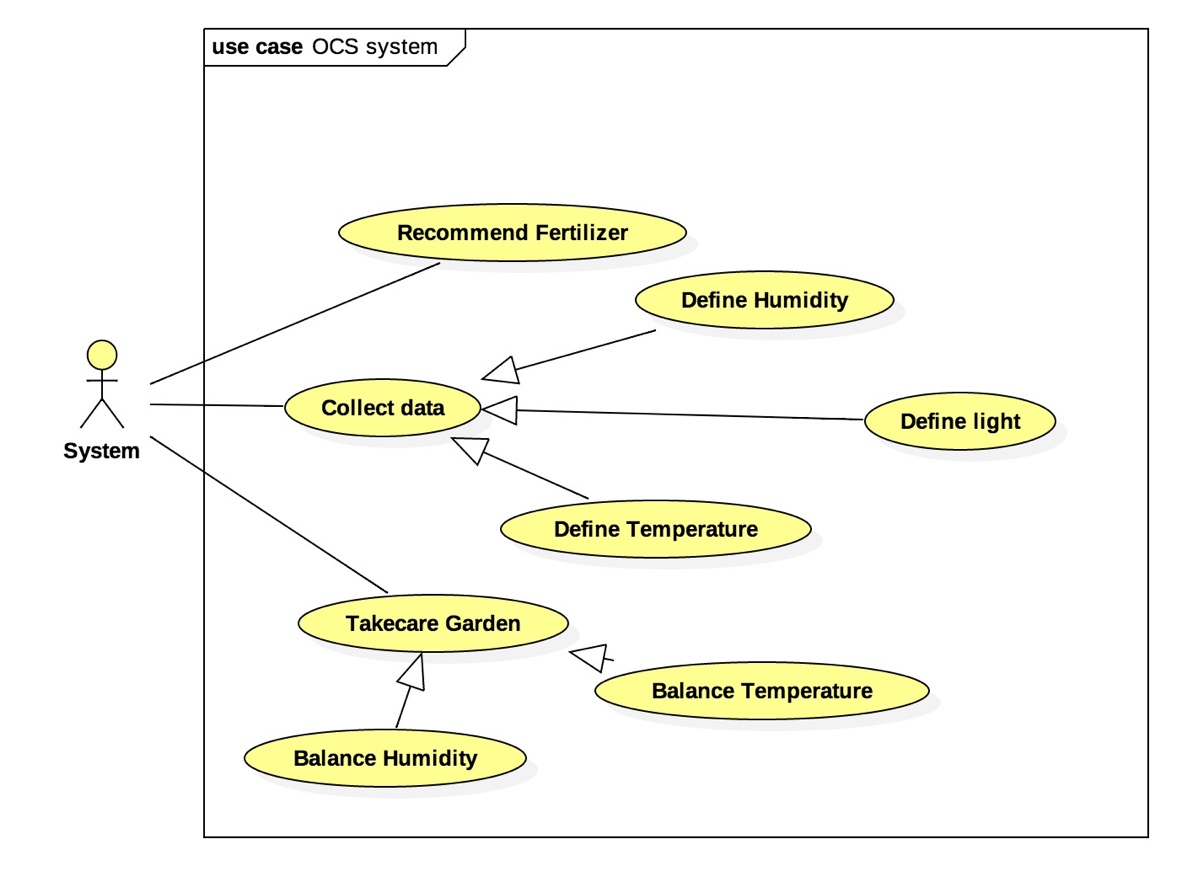
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.06** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | Export file log. | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner**.**   **Summary:**   * Owner can export a log file to storage information about their harvest.   **Goal:**   * Owner can export a log file to storage information about their harvest.   **Triggers:**   * Owner send export command to the system..   **Precondition:**   * The garden zone is created before and data is collected.   **Post Condition:**   * **Success:** File report is created. * **Fail:** File report is not created.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.1.7<Owner> View history plant zone.

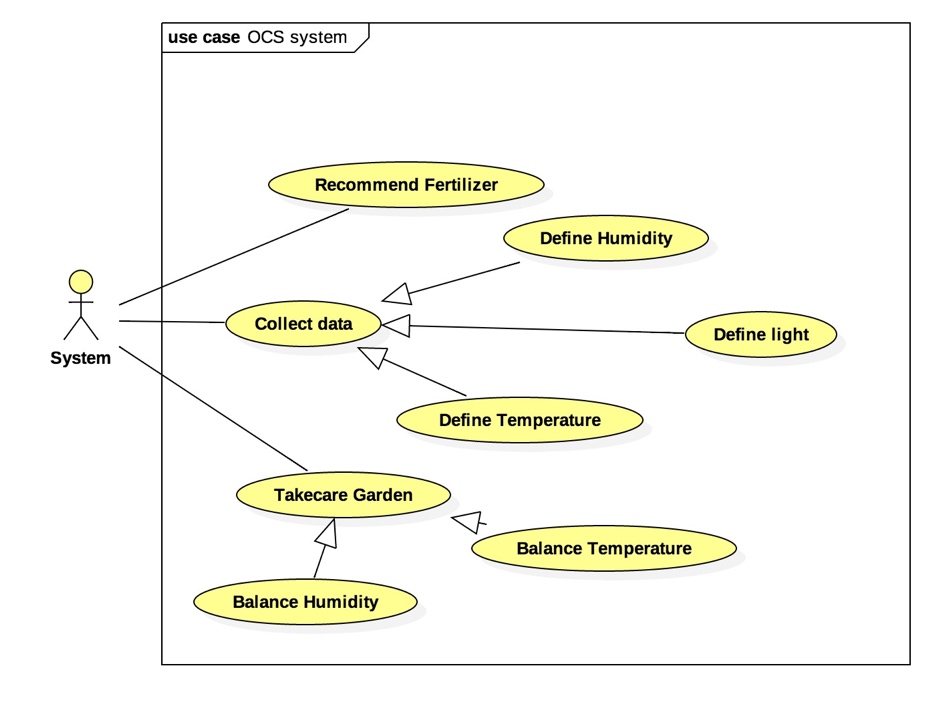


|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_A.07** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | View history plant zone | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * Owner.   **Summary:**   * Owner can explore history of garden zone.   **Goal:**   * Owner can explore history of garden zone.   **Triggers:**   * Owner send view history command to the system.   **Precondition:**   * The garden zone is available and data is created.   **Post Condition:**   * **Success:** graph and data is show in history view. * **Fail:** the system response “no data or garden zone”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

#### <System> Usecase overview.

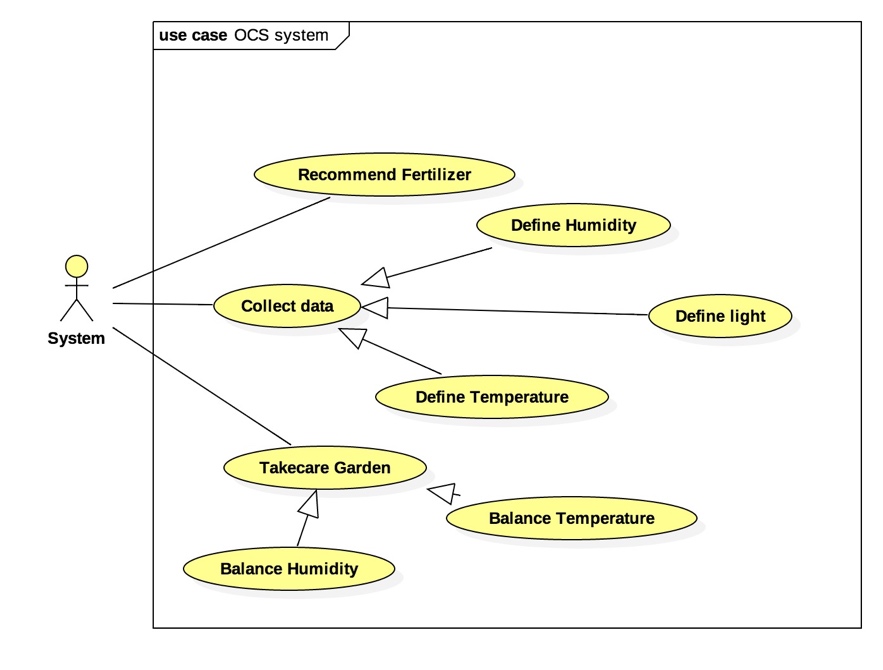


##### 2.3.2.1<System> Recommend Fertilizer.



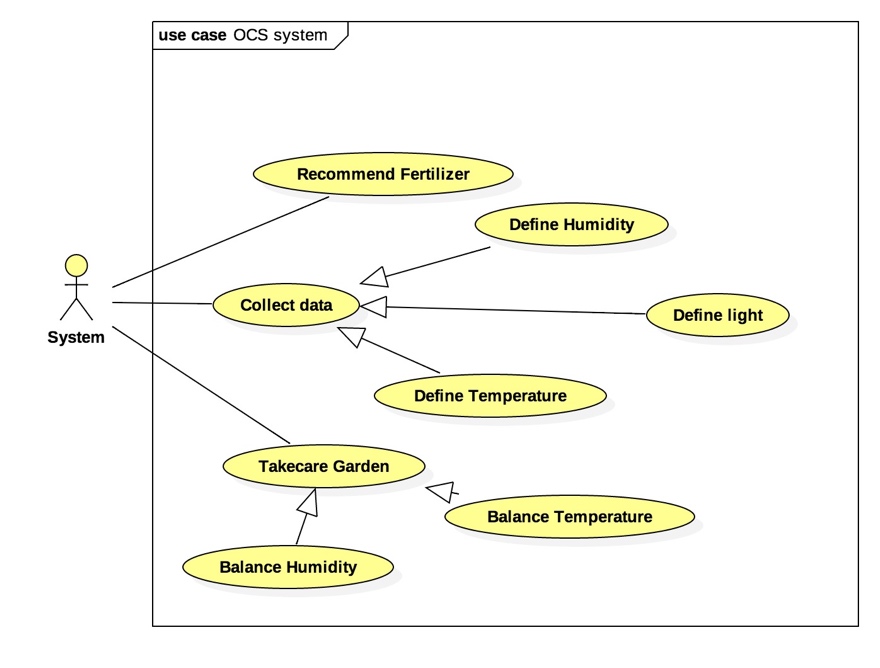
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S.0** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | System recommend fertilizer | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Normal |
| **Actor:**   * System**.**   **Summary:**   * System give owner recommendation for the garden.   **Goal:**   * Owner can fertilize exactly the fertilizer in time.   **Triggers:**   * The garden timeline is coming.   **Precondition:**   * Timeline is coming.   **Post Condition:**   * **Success:** the notification is show and notify to owner. * **Fail:** nothing to show.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.2.2<System> Define Humidity.



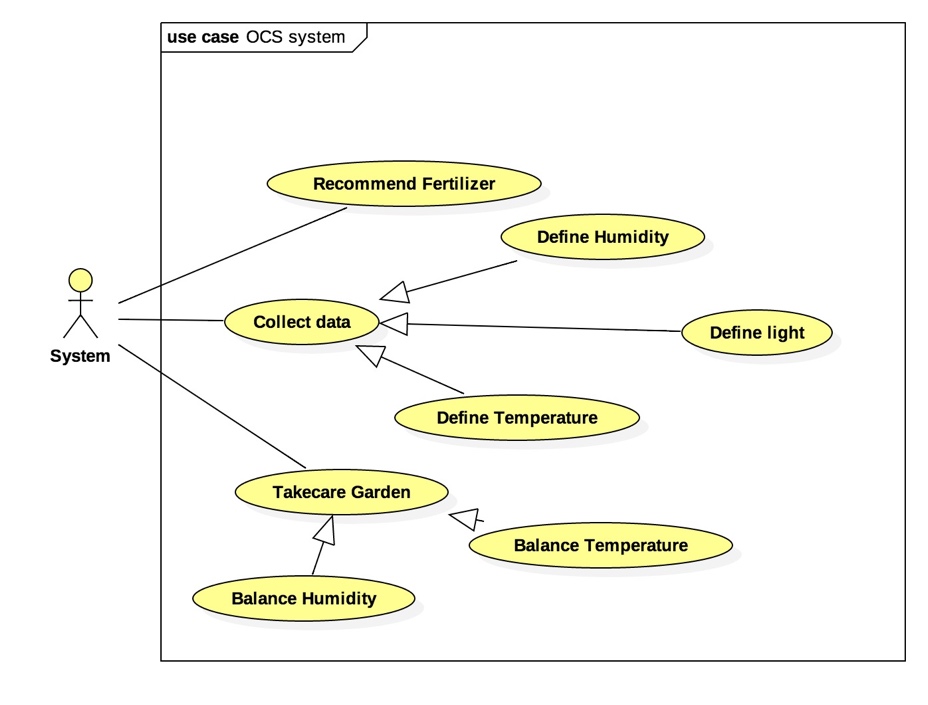
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S.0** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | System define humidity. | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | Highl |
| **Actor:**   * System**.**   **Summary:**   * System define the humidity of the garden zone.   **Goal:**   * System define the humidity for data collector and report use case.   **Triggers:**   * The system receive the data from sensor.   **Precondition:**   * The sensor is working fine.   **Post Condition:**   * **Success:** The data is recorded by the system and show in screen. * **Fail:** The system show “ Humidity sensor is failure”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.2.3<System> Define Temperature.



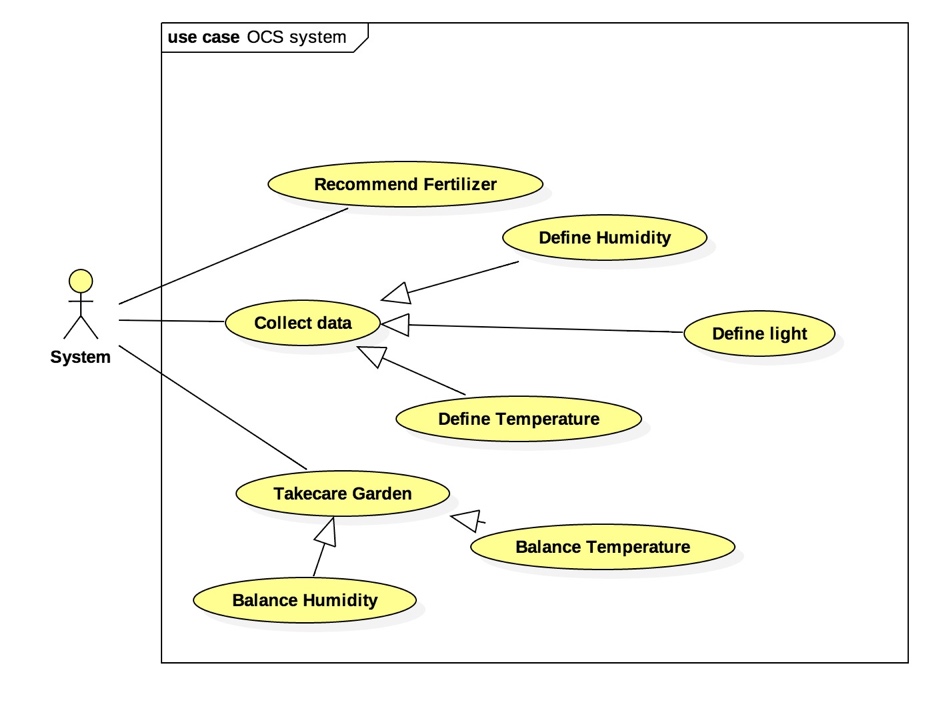
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S.0** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | System define temperature | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | High |
| **Actor:**   * System**.**   **Summary:**   * System define the temperature of the garden zone.   **Goal:**   * System define the temperature of the garden zone for graph and report use case.   **Triggers:**   * The system receive the data from sensor.   **Precondition:**   * The sensor is working fine..   **Post Condition:**   * **Success:** The data is recorded by the system and show in screen. * **Fail:** The system show “Temperature sensor is failure”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.2.4<System> Define light.



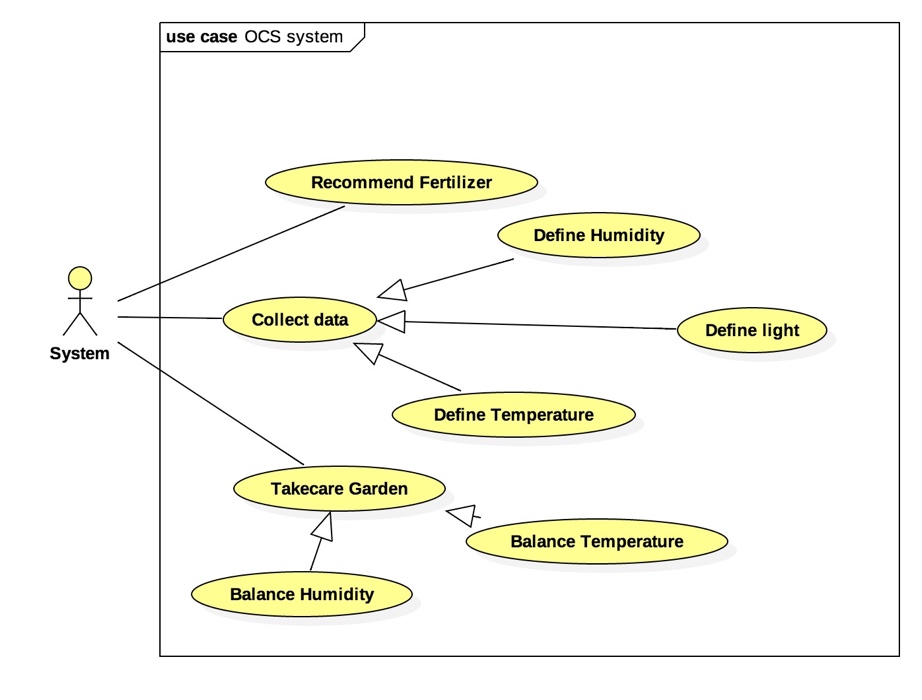
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S.0** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | System define light | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | High |
| **Actor:**   * System**.**   **Summary:**   * System define the light of the garden.   **Goal:**   * System define the light of the garden for graph and report use case.   **Triggers:**   * The system receive the data from sensor.   **Precondition:**   * The sensor is working fine.   **Post Condition:**   * **Success:** the data is recorded by the system and show in screen. * **Fail:** the system show “ Light sensor is failure”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.2.5 <System> Balance Humidity.



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S.0** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | System balance humidity | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | High |
| **Actor:**   * System**.**   **Summary:**   * System can auto balance humidity in the garden zone.   **Goal:**   * System an auto balance humidity in the garden zone.   **Triggers:**   * The data receive from the sensor is out of range.   **Precondition:**   * The sensor is working fine.   **Post Condition:**   * **Success:** the pumper is working to pump the water. * **Fail:** the system show “ the garden zone is not suitable”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

##### 2.3.2.6 <System> Balance Temperature.



|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S.0** | | | |
| **Usecase No.** | UC\_A.0 | **Usecase Version** | 1.0 |
| **Usecase Name** | System balance temperature | | |
| **Author** | SANGNT | | |
| **Date** | 20-10-2017 | **Priority** | High |
| **Actor:**   * System**.**   **Summary:**   * System can auto balance temperature in the garden zone.   **Goal:**   * System an auto balance temperature in the garden zone.   **Triggers:**   * The data receive from the sensor is out of range.   **Precondition:**   * The sensor is working fine.   **Post Condition:**   * **Success:** the pumper is working to pump the water. * **Fail:** the system show “ the garden zone is not suitable”.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  |  | | **2** |  |  | | **3** |  |  |   **Alternative scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | |  |  |  | |  |  |  |   **Relationship:** N/A.  **Business role:** | | | |
|  | | | |

## Software System Attribute.

### 3.1 Usability.

UI mobile application mobile is scalable with each monitor of smart phone or tablet:

* Font size: follow Google Material design guideline for typography.
* Font style: sans-serif font.
* Color: white.
* Background: green.
* User can be trained in one day to use all system functions.

UI HMI is scalable with each monitor of HMI:

* Font size : 14.
* Font style : Times new roman.
* Color : brown- silver.
* Background : black-silver.
* User can trained in one day to use all system function.

### 3.2 Reliability.

* + - * Owner can control the system from mobile with 100% accuracy.
      * Wireless control can reach 99% accuracy.
      * The sensor is working fine with 99% accuracy.

### 3.3 Availability.

System replies in maximum 5 seconds.

System controls under user confirmation.

### 3.4 Security.

User only access their garden with their garden account.

### 3.5 Maintainability.

The system is divided into modules separate for easy maintain.

### 3.6 Portability.

User can use the application on devices running Android 5.0 or later.

Web application can be run on Chrome browsers version 42 or later.

### 3.7 Performance.

The system can handle 10 requests at once time.

The garden zone system can handle 3 requests at once time.

Since user give command to HMI or Mobile, it cost maximum 5 seconds until it handle command and control device.

## Data structure.

**MÔ HÌNH JSON DATA BỔ SUNG SAU**

# Report No. 4 Software Design Description.

## Design Overview.

This document describes the technical and user interface design of VASH System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.

The database design describes the relationships between entities and details of each entity.

Document overview:

 Section 2: gives an overall description of the system architecture design.

 Section 3: gives component diagrams that describe the connection and  integration of the system.

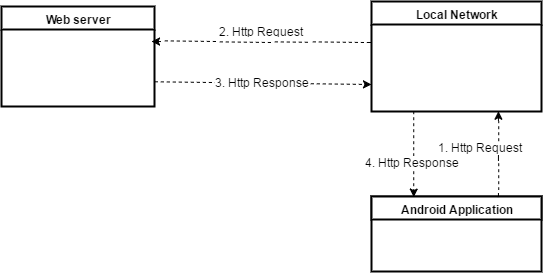
 Section 4: gives the detail design description which includes class diagram,  class explanation, and sequence diagram to details the application functions.

 Section 5: describe screens design.

 Section 6: describe a fully attributed ERD.

 Section 7: describe algorithms.

## System architectural design.



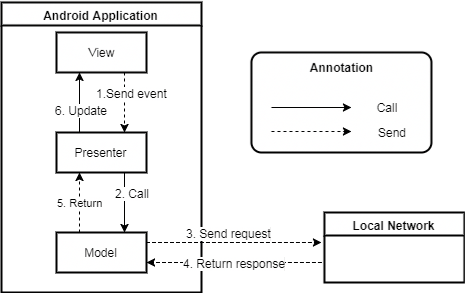
### Web service architecture design.

**IBM Cloudant.**

* 1. **Control center - HMI application architecture description.**

**Node RED architecture**

* 1. **Android application architecture description.**

**.** 

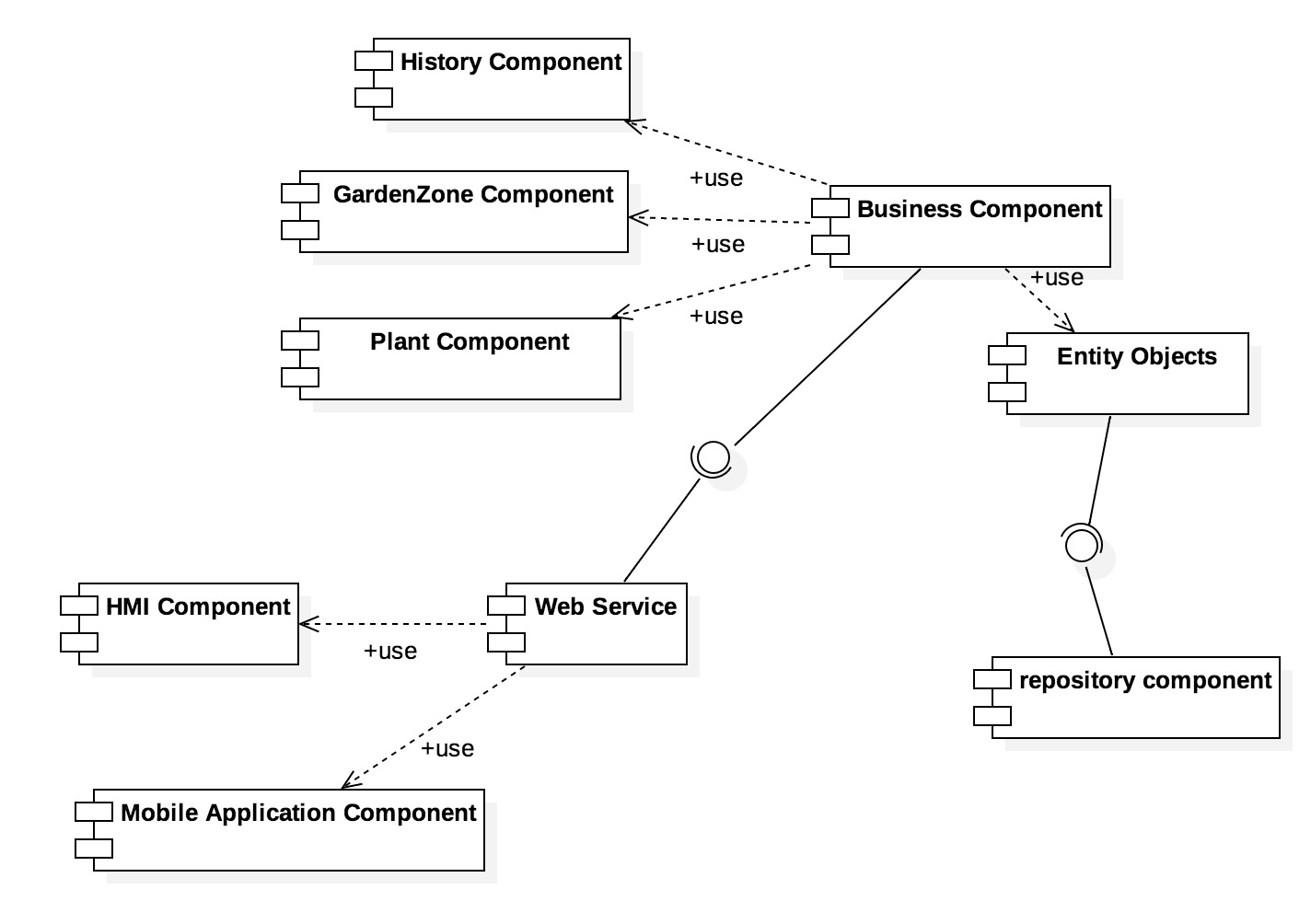
In Android application, the system is developed under MVP architecture style. We choose this architecture for Android Application because of following advantages:  The Model-View-Presenter pattern solves both of these issues by breaking the connection that the View has with the Model and creating only one class Presenter, that handles everything related to the presentation of the View  Presenter is also a single class that is easy to unit test. Android application follows MVP architecture with following components:

* **Model**: the data layer. Responsible for handling the business logic and communication with the network and database layers.
* **View**: the UI layer, displays the data and notifies the Presenter about user actions.
* **Presenter**: retrieves the data from the Model, applies the UI logic and manages the state of the View, decides what to display and reacts to user input notifications from the View.

**Reference**:

MVP Architect: *https://medium.com/@cervonefrancesco/model-view-presenter-android- guidelines-94970b430ddf https://pure.tue.nl/ws/files/48628529/Lou\_2016.pdf*

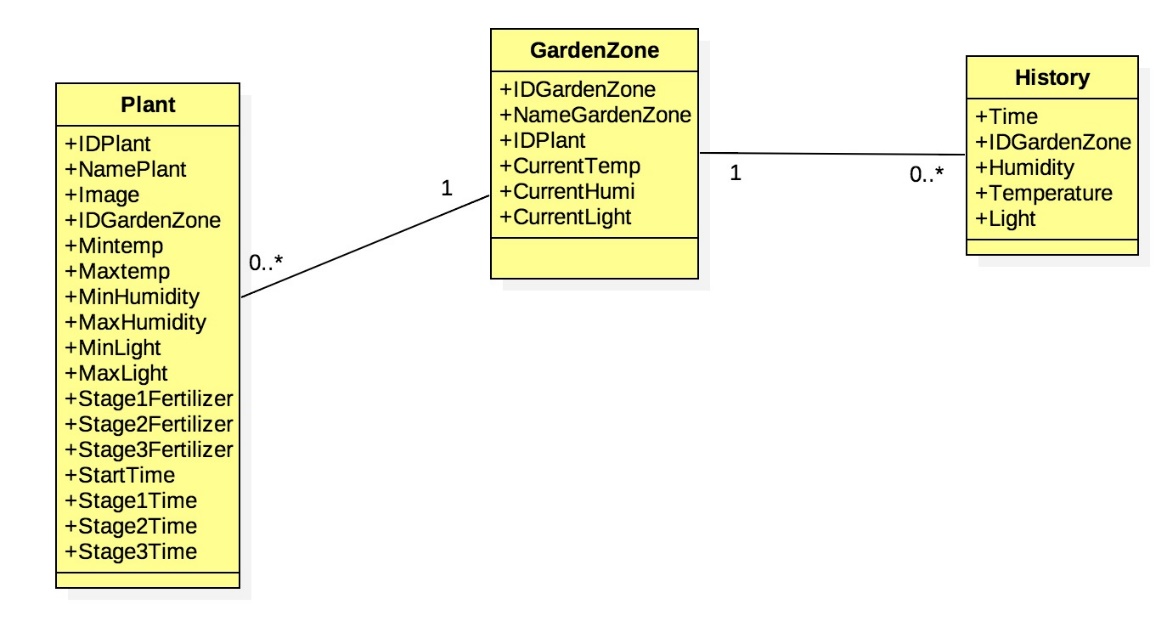
## Component diagram.



|  |  |
| --- | --- |
| **Component Name** | **Description** |
| History component | Component to handle history activities in the system. |
| GardenZone component | Component to handle plant Garden Zone activities in the system. |
| Plant component | Component to handle plant activities in the system |
| Business component | Common object to handle domain business operations for each objects. |
| Entity objects | Common object to handle transfer data from system to database. |
| Repository component | Component to handle interaction between database and system. |
| Web service | Provide API for mobile application to interact with the system. |
| HMI component | Provide user interface for owner |
| Mobile Application component | Mobile application package |

## Detail description.

### Class Diagram.



|  |  |  |
| --- | --- | --- |
| Class Name | Mapping column with Conceptual diagram | Description |
| Plant |  | Contain the plant information. |
| Garden Zone |  | Contain the garden zone information. |
| History |  | Contain the history of garden zone. |

* 1. **Class Diagram Explanation.**
     1. **Plant.**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | | | |
| **Attribute** | **Type** | **Visibility** | **Description** |
| IDPlant | UUID | Private | Unique identifier of plant |
| NamePlant | String | Private | Plant full name |
| Image | Image | Private | Image of plant |
| IDGardenZone | String | Public | Garden zone |
| MinTemp | Float | Public | Minimum temperature |
| MaxTemp | Float | Public | Maximum temperature |
| MinHumidity | Float | Public | Minimum Humidity |
| MaxHumidity | Float | Public | Maximum Humidity |
| MinLight | Float | Public | Minimum Light |
| MaxLight | Float | Public | Maximun Light |
| Stage1Fertilizer | String | Public | Recommendation Stage 1 Fertilizer |
| Stage2Fertilizer | String | Public | Recommendation Stage 2 Fertilizer |
| Stage3Fertilizer | String | Public | Recommendation Stage 3 Fertilizer |
| StartTime | Date | Public | Date time plant was grown |
| Stage1Time | Int | Public | Amount of time belong to stage 1 |
| Stage2Time | Int | Public | Amount of time belong to stage 2 |
| Stage3Time | Int | Pulic | Amount of time belong to stage3 |
| Method | | | |
| **Medthod** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

* + 1. **Garden Zone.**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUTE** | | | |
| **Attribute** | **Type** | **Visibility** | **Description** |
| IDGardenZone | UUID | Private | Unique identifier of Garden Zone |
| NameGardenZone | String | Private | Name of garden zone. |
| IDPlant | UUID | Private | Unique identifier of Plant. |
| CurrentTemp | Float | Public | Current temperature |
| CurentLight | Float | Public | Current light |
| CurrentHumi | Float | Public | Current Humidity |
| **METHOD** | | | |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

* + 1. **History.**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUTE** | | | |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Time | UUID | Private | Unique identifier of time. |
| IDGardenZone | UUID | Private | Unique identifier of Garden zone. |
| Humidity | Float | Public | Parameter of humidity |
| Temperature | Float | Public | Parameter of temperature |
| Light | Float | Public | Parameter of light |
| **METHOD** | | | |
| Getter | Attribute type | Public | Get attribute value |
| Setter | Void | Public | Set value of attribute |

* 1. **Interaction Diagram.**
     1. **Sequence diagram.**
     2. **Activity diagram.**

## Interface.

# Report 5. System Implementation & Test.

## In

# Report 6. Software User’s Manual

## Installation Guide.