B. Report No.2 Task Plan

1. Problem Definition

1.1. Name of this Capstone Project

- Design and implement the model of smart garden

1.2. Problem Abstract

With gardening, tracking information about temperature, humidity, light, PH degree and soil moisture is need to do usually. Gardener have to combine many indexes and have a better plan to process them. Easily if you have a garden with one kind of tree, but you have variety trees in garden and you don’t have enough time to handling each case. Gardener have another problem when they working on large garden. It also takes too much time and cost for tracking indexes by manually from different areas in garden. Changes in the weather is also a difficult problem, for example gardener doesn’t allow flower to bloom too early so they limit water for the tree but it’s rain and they don’t have any immediately solutions.

So we have a suggestion for Smart Garden Model to solve above problems. We provide a system to measure indexes of trees from sensors in automatically and regularly save time and costs for gardener. We also provide an immediately response system with changing of the weather to limit gardeners’ losses. Besides that, we have other advantages of this system to help gardener who can control their garden from anywhere, anytime on mobile application. Gardening is lighter with a detail schedule which is planned out daily, weekly, or monthly. A webserver is used for processing received indexes from microcontroller. Based on that, it give a response to control actuators through microcontroller or remind a plan for gardener.

1.3. Project Overview

1.3.1 Current Situation

There are the problems encountered in this project

+ Knowledge of garden: lack of information about trees and their indexes. One tree has many variety indexes in different unit such as light is lux, soil moisture is present or temperature is degree. It is complex to combine those indexes together to compute range standard suitable values for each area or each tree.

+ Endurance in severe weather: the system use many kind of sensors which are communicate with environment (such as water, humidity, or chemical fertilizers) directly. With our budget as a student, we just doing this project with some cheaper sensors so maybe it won’t have a long durable time.

+ The variety of model: gardener can plant kind of tree in an area or split them in each potted plant. There are 2 device relate with soil is pH sensors and soil moisture. We need a solution to get indexes from 2 device in a same soil. If there are many potted plant and many kind of trees, so the system must equipped sensors for each kind. That make cost rise.

Lack of efficacy device: there are less of pH sensors for soil in the market. We just found a sensor which measure pH degree by manually. Because we can not plug sensor in the soil 24/24, it is make sensor damage earlier. In addition, this sensor has a lower price than other pH sensor so it can give not true value.

Lack of energy and connectivity: There are individually module sensors which are used in this system. They don’t have power bank, battery or power recharge themselves so we need to power for them. Beside that there are connect to microcontroller via wired. That make complex to deploy this system in the garden with many electric wires.

Facial recognition system builds in security systems laptop or cell phone, door in the bank and smart home. They have relatively high security.

1.3.1.1 Advantages

- Can be applied in many fields.

- Convenient, social acceptability and inexpensive technique of identification

1.3.1.2 Disadvantages

- The relative angle of the target’s face influences the recognition score profoundly. When a face is enrolled in the recognition software, usually multiple angles are used (profile, frontal and 45-degree are common). Anything less than a frontal view affects the algorithm’s capability to generate a template for the face

- Other conditions where face recognition does not work well include poor lighting, sunglasses, long hair, or other objects partially covering the subject’s face, and low resolution images

- Another serious disadvantage is that many systems are less effective if facial expressions vary. Even a big smile can render the system less effective.

- Face recognition systems can’t tell the difference between identical twins.

- Cannot alert to owner through cellphone or email.

1.3.2 The Proposed System

According to the technology researches in first week, we decide to focus on how to processing sensors’ indexes, managing devices, giving protocol module with sever and user (gardener) with server. We also pay attention how to make a plan for each tree base on received indexes, how to control a life cycle of tree such as when are they need more light for growing or delay bloom flower with warm water v.v. …). Above solutions will help gardener not take too much time and cost in gardening.

According above purpose, we chosen a microcontroller which is…. This microcontroller have…. We can learn more about… We want to challenge…. We want to make a deep research on….

We also chosen some cheaper sensors which just get approximate values for sending to webserver. The important thing is we can processing all indexes and combine them together to have a better plan or response automatically.

The system using one high resolution color camera and use this to capture user face image. Then compare that image with trained data to detect this user is allow to access or not. In case user is accepted, solenoid lock will be opened, door auto open. In some specific case such as camera does not work or visit of guest, user can use alternative way to access house by provide phone number which is registered in the system, system will auto generate passcode and send it to phone number. User will use this passcode to unlock and this passcode will be valid for one hour. In case, user use unlock by facial recognition if this user is not existed in database his face will be captured and log into database. Besides, the system has a small LCD to show the face of user so that he can adjust his face to right position for capture. If use has highest role in system they can access system management after they recognize their face successfully. For backup, one battery will be attached to system to using in case main power supplier drops down. The system provides us SMS and internet alarm to pre-define user. So that the house owner can easier monitor and call police in case someone want to destroy. Besides, this system still available disadvantage. It can confuse between twins and in it can not distinguish between real person and his image

1.3.3 Boundaries of the System

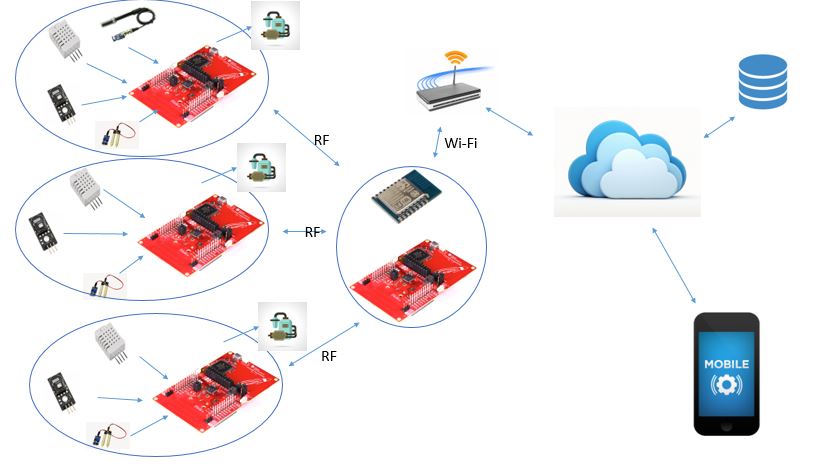
****

Figure 1: Boundaries of the System

1.3.4 Development Environment

1.3.4.1 Hardware requirements

1. CC1310 Launchpad

2. Light Sensor BH1750 FVI

3. Temperature and Humility Sensor DHT22

4. Soil Humility Sensor LM393

5. pH Sensor

6. Water pumps

7. The traction motor

8. Nebulizer

9. ESP8266 V7

PA\\*EFORMAT 1

1.3.4.2 Software requirements

1. Spring Tool Suite
2. Android Studio
3. IAR Workbench
4. Source Tree
5. Slake
6. Trello
7. Microsoft SQL Server/ MySQL
8. Star UML/ Software Ideas Modeler

2. Project organization

2.1 Software Process Model

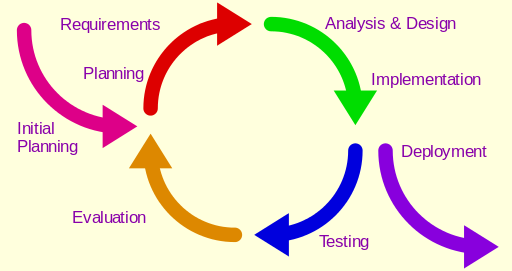


Figure 2: Incremental Model

For more information:

* <https://en.wikipedia.org/wiki/Iterative_and_incremental_development>

2.2 Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Nguyễn Đức Lợi | Supervisor  Project manager | * Defining business * Supporting in technique issues * Controlling the development process |
| 2 | Phan Thành Sang | Team leader  Business Analysis  Developer  Tester | * Managing process * Clarifying requirements * Preparing documents and reports * Creating task plan * Reviewing documents and reports * Committing all individual works. * Design and implement hardware. * Design and implement Database. * Researching components, document for implementing. * Supporting each other. |
| 3 | Phạm Hoàng Chinh | Developer  Tester | * Implement document and reports * Review documents and reports * Committing all individual works * Researching components, document for implementing. * Implement chart. |
| 4 | Huỳnh Hữu Nghị | Developer  Tester | * Implement document and reports * Review documents and reports * Design and implement mobile application. |
| 5 | Lê Văn Pháp | Developer  Tester | * Implement document and reports * Review documents and reports * Design and implement web server. |

Table 2: Roles and responsibilities

MAT 1

2.3 Tools and Techniques

|  |  |  |
| --- | --- | --- |
| No | Tools/Techniques | Name/Version |
| 1 | For Hardware | IAR Workbench 7.4 |
| 2 | For Hardware | PCB Editor 16.6 |
| 3 | For Software | Spring Tool Suite |
| 4 | For Software | Android Studio 2.1 |

3. Project Management Plan

3.1 Software development life cycle

|  |  |
| --- | --- |
| **Increment 0** | |
| Description | - Collect requirement  - Research the same system  - Research tools and technique  - Clarify requirement  - Define test approach  - Create introduction report |
| Deliverables | - Introduction report  - Research report  - Research hardware (sensor, MCU…) |
| Resource Needed | - 7 man-days |
| Dependencies and Constraints | - N/A |
| Risk | - Missing requirement  - Unclear scope of project  - Lack of member share of understand |
| **Increment 1** | |
| Description | - Create Software project management plan report  - Create Software Requirement Specification report  - Define requirement  - Define performance measures  - Design test case  - Research giant.  - Research hardware, sensor  - Connect sensor with MCU.  - Research RF connection.  - Research connect MCU to Internet  - Research Web server  - Research mobile application |
| Deliverables | - Software project management plan  - Software Requirement Specification  - Connect and transfer data with hardware.  - Connect and transfer data with internet  - Connect and transfer data with mobile application  - Test report  - Performance measures report |
| Resource Needed | - 14 man-days |
| Dependencies and Constraints | - Depend on introduction report, research reports, hardware device |
| Risk | - Lack of Experience.  - Late for shipping hardware  - Measure is not exactly  - Test case cannot cover all cases |
| **Increment 2** | |
| Description | - Create Software Design Description report  - Define requirement  - Define performance measures  - Design test case  - Design user interface  - Implement transfer data from sensor to MCU and MCU to server  - Implement control with algorism in web server and sent to mobile application  - Implement Web API and hardware get API, controlling water supplying system.  - Testing hardware transfer |
| Deliverables | - Software Design Description report  - Mobile application  - Web server and Web API  - Test report  - Performance measures report |
| Resource Needed | - 21 mans-day |
| Dependencies and Constraints | - Depend on introduction report, research reports, hardware |
| Risk | - Lack of Experience.  - Measure is not exactly  - Hardware work not good  - Test case cannot cover all cases |
| **Increment 3** | |
| Description | - Create System Implementation & Test report  - Define requirement  - Define performance measures  - Design test case  - Implement schematic of board  - Implement hardware connects together  - Implement transfer data with wireless communication  - Implement transfer data from mobile application to server  - Implement control water supplying  - Test transfer data  - Test control signal from server to MCU |
| Deliverables | - Create System Implementation & Test report  - Board for hardware  - Web server for control and transfer and save data  - Mobile Application for control and show information  - Test report  - Performance measures report |
| Resource Needed | - 21 mans-day |
| Dependencies and Constraints | - Depend on introduction report, research reports, hardware |
| Risk | - Lack of Experience.  - Measure is not exactly  - Hardware is not exactly  - Test case cannot cover all cases |
| **Increment 4** | |
| Description | - Create installation guide  - Create User guide  - Integration test  - Implement modeling garden. |
| Deliverables | - Installation guide  - User guide  - Integration test report  - Modeling garden |
| Resource Needed | - 7 mans-day |
| Dependencies and Constraints | - All functions and reports are completed |
| Risk | * Lack of experience * Hard to grow plant * Depend on weather |

Table 3: Software Development Life Cycle Detail

3.2 Increment Detail

3.2.1 Increment 0

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 4: Increment 0 Detail

3.2.2 Increment 1

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 5: Increment 1 Detail

3.2.3 Increment 2

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 6: Increment 2 Detail

3.2.4 Increment 3

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 7: Increment 3 Detail

3.2.5 Increment 4

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |

Table 8: Increment 4 Detail

3.3. All Meeting Minutes

<https://github.com/sangphanthanh/CP_SmartGarden/tree/master/Document>

4. Coding Convention

4.1. Local Variables

Declare variables in local scope and initialization should be used instead of declaration and assignment.

<http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Local_Variables>

4.2. General Naming Rules

Function names, variable names, and filenames should be descriptive; eschew abbreviation. Give as descriptive a name as possible, within reason. Do not worry about saving horizontal space as it is far more important to make your code immediately understandable by a new reader. Do not use abbreviations that are ambiguous or unfamiliar to readers outside your project, and do not abbreviate by deleting letters within a word.

<http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#General_Naming_Rules>

4.3. File Names

Filenames should be all lowercase and can include underscores (\_) or dashes (-). Follow the convention that your project uses. If there is no consistent local pattern to follow, prefer "\_".

<http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#File_Names>

4.4. Variable Names

The names of variables and data members are all lowercase, with underscores between words. Data members of classes (but not structs) additionally have trailing underscores.

<http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Variable_Names>

4.5. Function Names

Regular functions have mixed case; accessors and mutators match the name of the variable. Functions should start with a capital letter and have a capital letter for each new word. No underscores

<http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Function_Names>

4.6. Type Names

Type names start with a capital letter and have a capital letter for each new word, with no underscores. The names of all types — classes, structs, typedefs, and enums — have the same naming convention. Type names should start with a capital letter and have a capital letter for each new word. No underscores.

<http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Type_Names>

4.7. Line Length

Each line of text in your code should be at most 80 characters long. We recognize that this rule is controversial, but so much existing code already adheres to it, and we feel that consistency is important.

[http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Line\_Length](http://google-styleguide.googlecode.com/svn/trunk/cppguide.html%23Line_Length)

4.8. Class Comments

Every class definition should have an accompanying comment that describes what it is for and how it should be used.

[http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Class\_Comments](http://google-styleguide.googlecode.com/svn/trunk/cppguide.html%23Class_Comments)

4.9. Function Comments

Declaration comments describe use of the function; comments at the definition of a function describe operation.

[http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Function\_Comments](http://google-styleguide.googlecode.com/svn/trunk/cppguide.html%23Function_Comments)

4.10. Variable Comments

In general, the actual name of the variable should be descriptive enough to give a good idea of what the variable is used for. In certain cases, more comments are required.

[http://google-styleguide.googlecode.com/svn/trunk/cppguide.html#Variable\_Comments](http://google-styleguide.googlecode.com/svn/trunk/cppguide.html%23Variable_Comments)