## D. Report No. 4 Software Design Description

### Design Overview

- This document describes the technical and user interface design of MSSC 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 and how users interact to systems. Each screen includes functions, types of input/output, and event handling.
* Section 6: describe a fully attributed ERD.
* Section 7: describe algorithms.

### System Architectural Design

Figure 1: System Architecture Design Overview

**Web Server**

**View**

**Web API**

**Controller**

**Web API**

**Model**

**Web Services**

**Database**



**Android Client**

**User Interface**

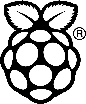
HTTP Request

HTTP Response

**Main Controller**

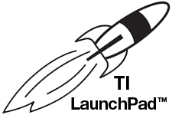
**Raspberry Pi 3**

**Mode B**



**CC1310**

**Launchpad**



**RF**

**Core**

Request

HTTP

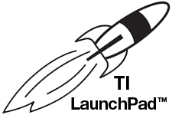
HTTP

Response

**Sub Controller**

**CC1310**

**Launchpad**



**RF**

**Core**

**End Devices**

**Sensors I2C**

**Sensors Analog**

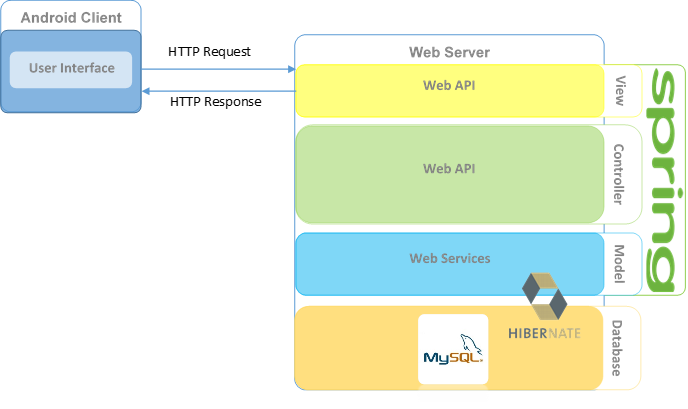
DIO

SCL

SDA

ADC

#### Software architecture description



There are two parts in software architecture of this system: Android application and Web Server.

+ Android application is developed by JAVA language programing with Android Studio. This application is used to show all user interface, request and response data with web server via Internet.

+ Web server is developed under Spring MVC – Hibernate architecture style. We choose this architecture because of following advantages:

- Web application contains Web services with MVC architecture, we can separate business code with Controller and View. So we can use the business code in web service without repeat the code.

- We can organize the code better for maintainability, extensibility, reusability so we can expand the scope to other kind of illnesses such as flu, fever...

- In scope of 3-members team, MVC architecture make it easier to split the big project into small modules and make it easier to assign each module for members in our team.

This project follows MVC architecture with following components:

Controller is the part of the web server that acts like event handler to handles user interaction.

+ Web Controller is the parts of the controller that acts like event handler for web and server communication via HTTP protocol.

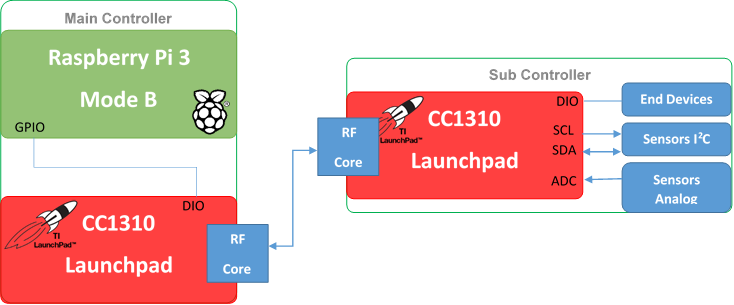
Model is the parts of the web server that acts like a data transfer object between the system and database, handle business of system.

+ Business is the parts of the model that do business processing to solve domain problems.

Business Component is the parts of the business that do business of system.

Business Objects is the parts of the business that define some objects to use in business component.

#### Hardware architecture description



The hardware architecture contains two parts: Main Controller and Sub Controller.

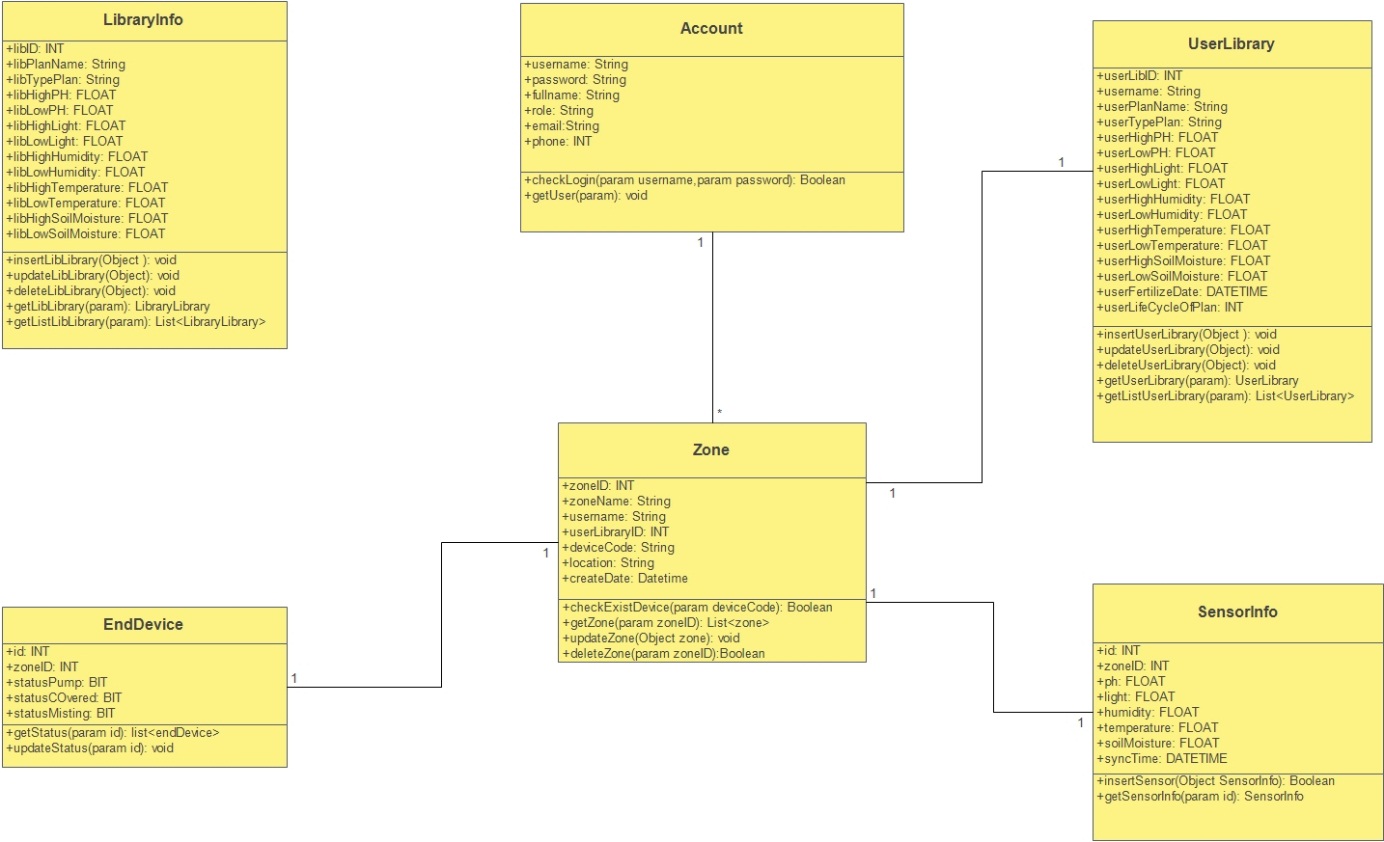
+ Main Controller includes Raspberry Pi 3 and CC1310 Launchpad. We use UART to communicate data between them. Raspberry Pi 3 connect with Web server via HTTP request and HTTP response. CC1310 send and receive data to Sub Controller via RF which created by RF core in each CC1310 at nodes.

+ Sub Controller, that at each node in garden, has a CC1310 Launchpad which to send and receive data to Main Controller via RF. In addition, CC1310 connect with end devices such as water pump, pressure pump, and traction motor via DIO pins. I2C sensors wired with CC1310 via SCK and SDA pins. Signals from analog sensors will processed with Analog Digital Converter 12-bit inside CC1310.

### Component Diagram

### Detailed Description

#### Class Diagram



#### Class Diagram Explanation

##### Library Info

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| libID | Int | Private | Unique identifier of a plant in library |
| libPlantName | String | Private | Plant of name |
| libTypePlant | String | Private | Type of plant |
| libHighPH | Float | Private | High threshold of pH degree |
| libLowPH | Float | Private | Low threshold of pH degree |
| libHighLight | Float | Private | High threshold of light intensity |
| libLowLight | Float | Private | Low threshold of light intensity |
| libHighHumidity | Float | Private | High threshold of humidity |
| libLowHumidity | Float | Private | Low threshold of humidity |
| libHighTemperature | Float | Private | High threshold of temperature |
| libLowTemperature | Float | Private | Low threshold of temperature |
| libHighSoilMoisture | Float | Private | High threshold of soil moisture |
| libLowSoilMoisture | Float | Private | Low threshold of soil moisture |
| Method | **Type** | **Visibility** | **Description** |
| insertLibLibrary | void | Private | Insert new plant to Library Info |
| updateLibLibrary | void | Private | Update a plant in Library Info |
| deleteLibLibrary | void | Private | Delete a plant in Library Info |
| getLibLibrary | Library | Private | Get ID of a plant in Library Info |
| getListLibLibrary | List | Private | Get List of plant from Library Info |

##### Account

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| username | String | Private | Unique username for each user |
| password | String | Private | Password of User |
| fullname | String | Private | Full Name of User |
| role | String | Private | Role of User |
| email | String | Private | Email of User |
| phone | Integer | Private | Phone of User |
| Method | **Type** | **Visibility** | **Description** |
| checkLogin | boolean | Private | Check username and password |
| getUser | void | Private | Get username of user |

##### User Library

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| userLibID | Int | Private | Unique identifier of a plant which user define in user library |
| username | String | Private | User who define a plant |
| userPlantName | String | Private | Name of plant which user define |
| userTypePlan | String | Private | Type of plant which user define |
| userHighPH | Float | Private | High threshold of pH degree |
| userLowPH | Float | Private | Low threshold of pH degree |
| userHighLight | Float | Private | High threshold of light intensity |
| userLowLight | Float | Private | Low threshold of light intensity |
| userHighHumidity | Float | Private | High threshold of humidity |
| userLowHumidity | Float | Private | Low threshold of humidity |
| userHighTemperature | Float | Private | High threshold of temperature |
| userLowTemperature | Float | Private | Low threshold of temperature |
| userHighSoilMoisture | Float | Private | High threshold of soil moisture |
| userLowSoilMoisture | Float | Private | Low threshold of soil moisture |
| userFertilizeDate | Datetime | Private | Date of Fertilize Date which user define |
| userLifeCycleOfPlant | Int | Private | Life Cycle of a plant |
| Method | **Type** | **Visibility** | **Description** |
| insertUserLibrary | void | Private | Insert new plant to Library Info |
| updateUserLibrary | void | Private | Update a plant in Library Info |
| deleteUserLibrary | void | Private | Delete a plant in Library Info |
| getUserLibrary | Library | Private | Get ID of a plant in Library Info |
| getListUserLibrary | List | Private | Get List of plant from Library Info |

##### Zone

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| zoneID | Int | Private | Unique ID for each zone |
| zoneName | String | Private | Zone Name |
| username | String | Private | Username of user who define a zone |
| userLibraryID | Int | Private | ID of plant in a zone |
| deviceCode | String | Private | Device code to identify devices in zone |
| location | String | Private | Location of zone |
| createDate | Datetime | Private |  |
| Method | **Type** | **Visibility** | **Description** |
| checkExistDevice | boolean | Private | Check end devices exist in system |
| getZone | List<zone> | Private | Get ZONE |
| updateZone | void | Private | Update information of plant in zone |
| deleteZone | Boolean | Private | Delete zone |

##### Sensor Info

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Int | Private | Unique ID for set of sensors |
| zoneID | Int | Private | Zone ID include sensors |
| ph | Float | Private | pH degree (reach realtime) |
| light | Float | Private | light intensity (reach realtime) |
| humidity | Float | Private | Humidity (reach realtime) |
| temperature | Float | Private | Temperature degree (reach realtime) |
| soilMoisture | Float | Private | Soil Moisture (reach realtime) |
| syncTime | Datetime | Private | Synchonization Date when get indexes from sensors |
| Method | **Type** | **Visibility** | **Description** |
| insertSensor | boolean | Private | Insert new sensor |
| getSensorInfo | List<zone> | Private | Get indexes of sensors from Sub C |

##### End Devices

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| id | Int | Private | ID of set end devices in one node |
| zoneID | Int | Private | Zone ID include end devices |
| statusPump | Bit | Private | Full Name of User |
| statusCOvered | Bit | Private | Role of User |
| statusMisting | Bit | Private | Email of User |
| Method | **Type** | **Visibility** | **Description** |
| checkLogin | boolean | Private | Check username and password |
| getUser | void | Private | Get username of user |

#### Interaction Diagram

### Interface

#### Component interface

#### User Interface Design

### Database Design

#### Entity relationship diagram (ERD)

#### Data Dictionary

### Algorithms

#### Document Breakdown

##### Definition

##### Define Problem

##### Complexity

##### Flowchart