SAMSUNG

SOFTWARE REQUIREMENT SPECIFICATION (Version 0.9)

Autonomous Sprinkler System

Software that controls sprinklers autonomously.

August 2022

DUE DATES

	Scope of AD	Due Date	Length Limit
Interim Report	Chapters 1~4	8/25(Th), 9pm	30± Pages
Pre-Final Report	Chapter 5	9/04(Sun), 9pm	50± Pages
Final Report	Chapters 6~7	9/26(Mon), 9pm	70± Pages

CEP INSTRUCTOR

Soo Dong Kim, Ph.D.
Professor, School of Software
Soongsil University, Seoul, Korea
sdkim777@gmail.com 010-7392-2220

삼성전자 첨단기술연구소

TABLE OF CONTENTS

Autonomous Sprinkler System

1. Purpose of the Document

The purpose of this document is to specify the requirement for developing the target system in this CEP. The requirement will become the basis for designing the software architecture of a target system, which is required as a fulfillment to acquire the Samsung Associate Architect (AA) certification.

2. Comprehensive Evaluation Project (CEP)

2.1. About Associate Architect (AA) Program

Associate Architect Program of Samsung Electronics is to provide participants with two sets of software architecture design proficiency.

- **Body of Knowledge (BoK) on Software Architecture**

  This set includes the fundamental theories and methods for designing SW architecture

- **Skillset for Designing SW Architecture**

  This set includes the practical skill for applying architecture design methods to a given SRS.

CEP is designed to fulfill the second set of AA program through an individual design project.

2.2. How is the CEP problem prepared?

The CEP problem is prepared by the instructor, based on the following principles.

- **Principle 1. Utilizing the Whole BOK of Software Architecture Design**
  - Utilizing Architecture Styles
  - Designing Architecture for multiple Views
  - Designing Architecture for Non-Functional Requirements
- **Principle 2. Handling the Complexity of Industrial Systems**

  The target system to design in CEP is an industry-level complex software system, i.e., not an academic problem appeared in books or literature.

- **Principle 3. Solution Not Available in Public**

  CEP problem is not a reproduction of already existing exercise problem in books, and hence the architecture design solution for the CEP problem is not available in public.

2.3. Architecture Design Reports in CEP

Each participant designs and submits the design of software architecture for the target system in incremental manner.

- Interim Report
  This report includes the context analysis model and the skeleton architecture design of the target system.
- Prefinal Report
  This report includes the architecture design for multiple views of the system; functional view, information view, behavior view, and deployment view.
- Final Report
  This report includes the architecture design for non-functional requirements and the validation of the architecture design.

❑ **Weight Distribution of CEP Reports**

Interim Report	Prefinal Report	Final Report	TOTAL
30 points	30 points	40 points	**100 points**

2.4. Template for Architecture Design

A template for designing the software architecture in CEP is provided and hence participants can utilize the template in specifying the architecture design. The template is devised to be consistent with the architecture design methodology provided by the instructor.

❑ **Filename**

AA-2022-A4, CEP, AD Template.docx

3. Overview of the Target System

3.1. What is Smart Farming?

- **Definition of Smart Farming**

  Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology – including big data, the cloud and the internet of things (IoT) – for tracking, monitoring, automating and analyzing operatio

  It refers to the use of IT technologies to increase the productivity and efficiency of agricultural systems. The key technology of smart farming includes sensor fusion, data analytics, robotics, AI, and internet of things (IoT), which together provide the capability of tracking, monitoring, analyzing, and automating farming operations.

- **Benefits of Smart Farming**

  Smart Farming allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc. For example, solar powered and mobile operated enable the cost saving of electricity, and drones and robots perform labor-intensive farming tasks in efficient ways.

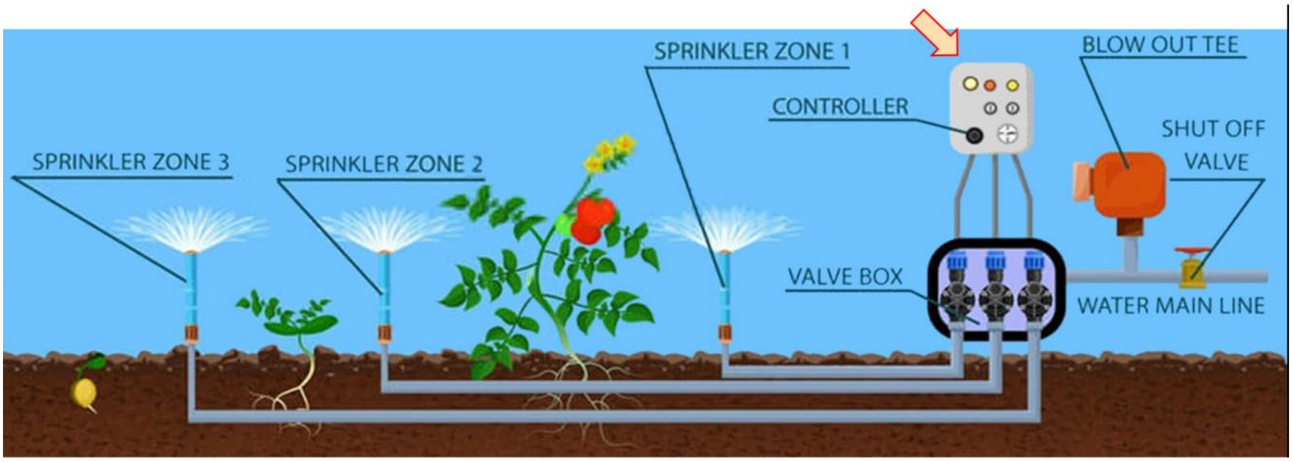
3.2. What is Sprinkler System?

- **Sprinkler**

  Sprinkler for farms and gardens is a hardware device used to spray water. Sprinklers are used to water plants or grass, or to put out fires in buildings.

- **Sprinkler System**

  Sprinkler System is an integrated set of devices and a software controller to provide a cost-efficient watering capability. A typical configuration of the devices is shown in the following figure.



As shown in the figure, there is a flow of water from the meter to the sprinklers at end points. The supply of water is programmed and controlled by the 'Controller' device. The water is distributed through a network that may consist of pumps, valves, pipes, and sprinklers. Irrigation sprinklers can be used for residential, industrial, and agricultural usage.

- **Water Values and Sprinkler Heads**

  Water values are used to open or close the water flow to sprinklers as shown in the following figure.

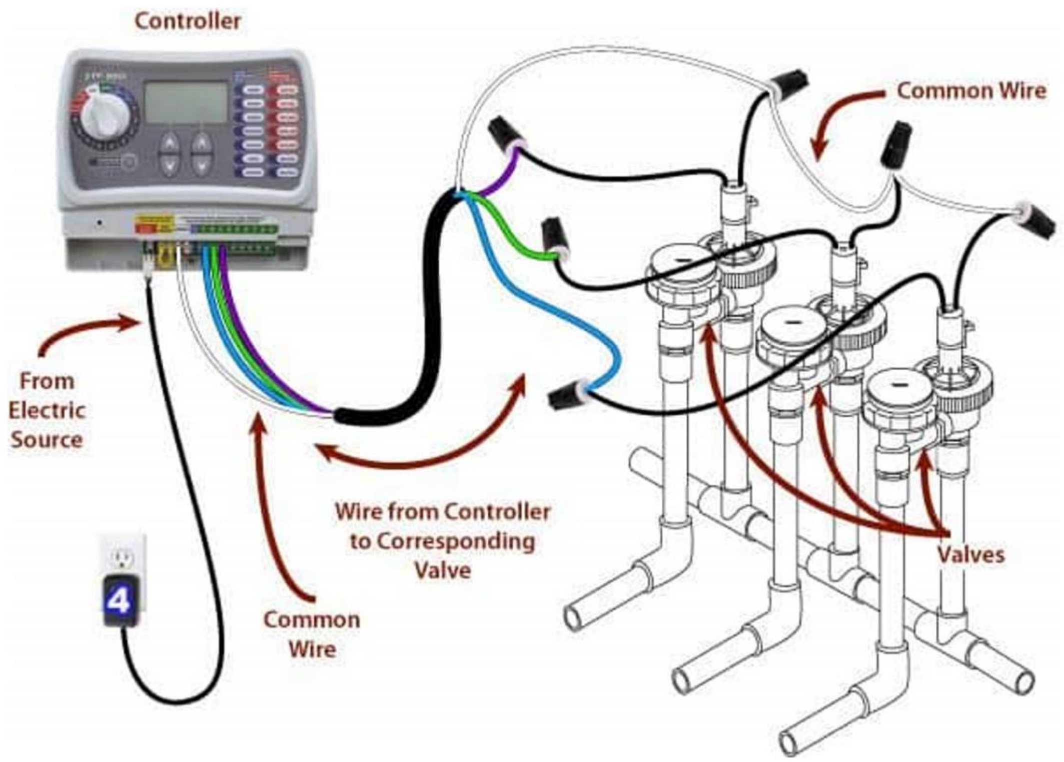
The figure shows a set of 3 water values, components of a water value, and two types of sprinkler heads. A water value houses a DC-powered motor inside, which is to turn on or off the water flow. The sprinkler controller unit supplies the DC electricity to the motors according to the present schedule.

- **Sprinkler Controller**

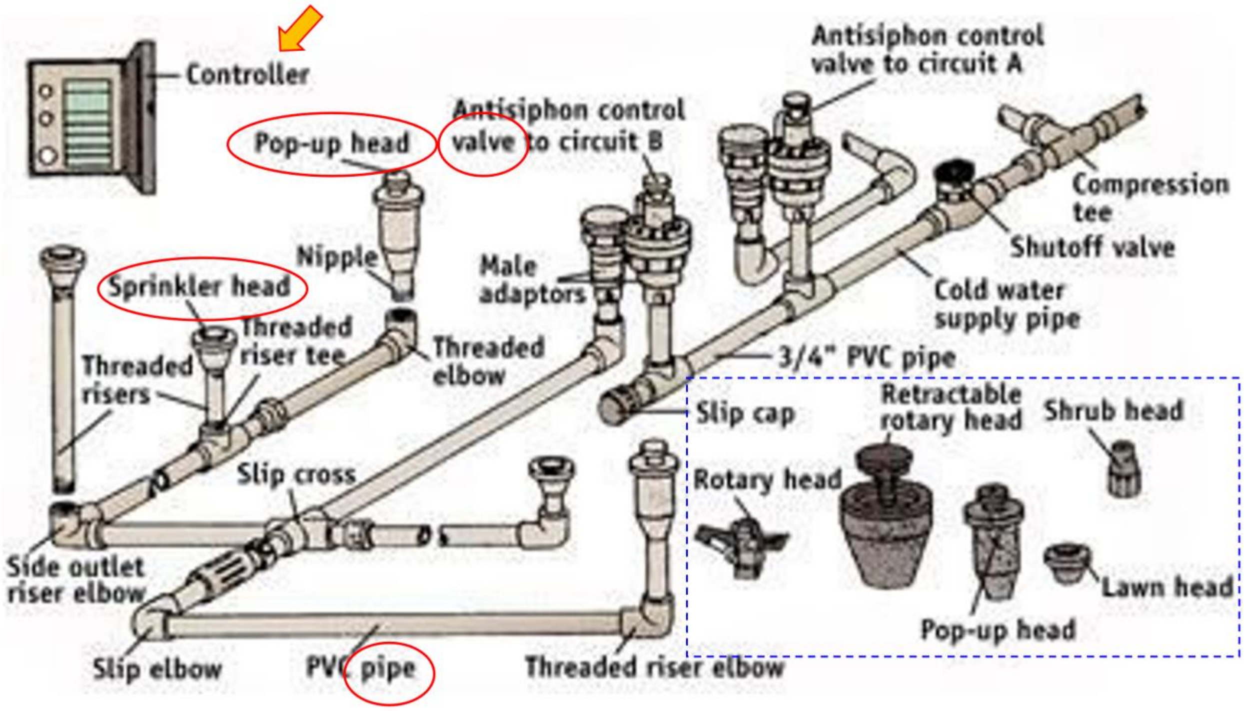
Like thermostats, a sprinkler controller is a programmable device that is used to set the watering schedule. The following figure shows an example of the controller.

Using the control knob and buttons, the schedule for watering on different watering zones can be set. Then, the controller controls the water values to supply water according to the schedule.

The controller has wire-connections to water valves and controls the values using DC electricity, as shown in the following figure.



Each water valve has a pipe that streams water to sprinkler heads as shown in the following figure.



In the figure, the controller is shown on the upper-left corner, (water) control valves are shown in the middle, and pipes transmit water from the values to the sprinkler heads.

- **Types of Sprinkler Heads**

  There are different types of sprinkler heads for different watering needs. Some of the common head types are shown in the lower-right corner of the figure above. Sprinkler heads have common characteristics.

  - Sprinkler heads commonly operate at a residential water pressure of 30 to 50 pounds per square inch (PSI). Sprayers and rotary heads work best the water pressure is less than 40 PSI.
  - A rotary sprinkler head is most effective on lawns greater than 30 square feet.
  - Most sprinkler heads are made of plastic and a few are manufactured with brass nozzles. These sprinkler heads are as durable as older brass and zinc models.

  Common types of sprinkler types include the followings.

3.3. Limitations of Sprinkler System

There are limitations with the conventional sprinkler system.

- **Limited Precision on Watering Schedule**

  It is not trivial to precisely determine the amount to water, the timing to start watering, duration of watering, and the direction of watering.

- **Difficulty to Reflect Evolving Weather Conditions**

  The watering schedule is usually set for an anticipated average weather condition for a region and for a season. However, the weather may expectedly be hotter or colder, more rain or more snow than the anticipated. On these occasions, the preset watering schedule would no longer be appropriate. The results can be *underwatering* or *overwatering*.

  - Underwatering
    The drought in summer would result in underwatering, which will threat the healthiness of plants.

  - Overwatering
    The occasion of heavy rains would result in overwatering, which will cause the high cost of water and negative effects of plants such as rotten roots.

- **Not Reflecting the Plant-specific Conditions**

  Each type of plants responds differently to the amount of water supplied. It would be ideal to supply the amount of water that is the most appropriate for the target plants. The current Sprinkler controllers do not consider the plant-specific conditions. They are set to a single uniform schedule regardless of plant conditions. The results can be un-even growths or unhealthy conditions of plants.

3.4. What is Autonomous System?

3.4.1. Autonomic Computing

Autonomic computing is the capability of a system that can manage itself without the intervention of human users through adaptive technologies. It is the next higher level of technology to the automation technology. Autonomic computing is characterized by self-* capability as in the following list.

- Self-regulation
- Self-learning
- Self-awareness
  - Self-inspection, Self-decision
- Self-organization

- Self-creation
  - Self-assembly, Self-replication
- Self-management
  - Self-governance
- Self-description
  - self-explanation, Self-representation

- **Difference between Automatic and Autonomous**

  The difference between automatic and autonomous is the degree of human intervention. For example, an automated car does not have the level of intelligence or independence that an autonomous car has. A driverless and self-driving car is operated in the autonomous manner.

  The capability of autonomous systems provides a higher productivity, safety, and convenience to human beings than automatic systems.

3.4.2. Adoption of Autonomous Computing Technology

Autonomous technology is rapidly spreading beyond the automobile and transportation sectors into healthcare, logistics and manufacturing. The development of autonomous systems becomes feasible due to the technology maturity of sensors, IoT devices, big data analysis, and machine learning.

There are a number of systems that are built with autonomic computing principle. Some of them are listed here.

- Self-driving Car and Truck
- Delivery Drone such as Amazon Prime Air
- Smart Factory
- Autonomous Robots for Manufacturing Plants
- Mail and Marine Transport
- Autonomous Air Space Control

The following is an example of adopting autonomic computing in space programs.

In the figure, a space probe that is able to deal with the surface of a planet to collect samples. If the ground is too hard for digging, the problem will device to move to an area that looks software.

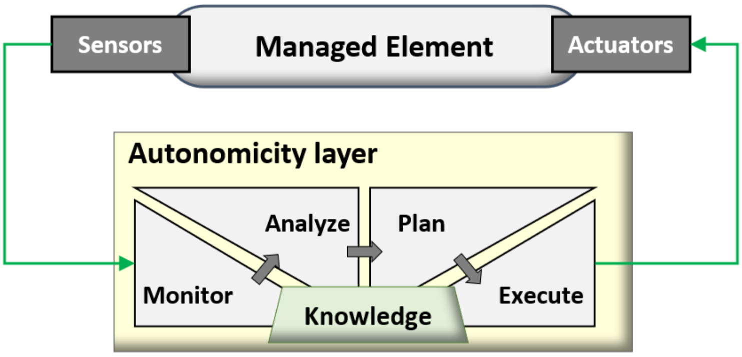
3.4.3. Benefits of Autonomic Computing

The key benefits of autonomic computing can be the followings.

- Optimization of Functionality and Process Control
- Increased Productivity
- Reduced Operation Cost
- Adaptability to Changes

3.4.4. MAPE-K Model for Autonomic Computing

The MAPE-K model is adopted as the foundation for building autonomous systems. It have 4 phases in its control loop.



There are 4 phases in MAPE-K model, and the sequence of running the 4 phases repeats.

- **Phase of Monitor**

  This phase is to self-acquire the environmental contexts using sensors. IoT devices, and cameras.

- **Phase of Analyze**

  This phase is to self-determine the situation using acquired contexts.

- **Phase of Plan**

  This phase is to self-devise a plan that is suited to the determined situation.

- **Phase of Execute**

  This phase is to self-execute the devised action plan, and then the environment would be changed.

The core of the MAPE-K is the knowledge base that is used to govern the four operations. It maintains the following information.

- Meta-data of the environment, i.e. managed elements
- Rules and Policies for performing the four operations; M, A, P and E.
- Governs the behavior of the four operations.

Hence, it is the most essential element of autonomous systems. It should always be maintained at the most optimal level.

3.5. What is Autonomous Sprinkler System?

Autonomous Sprinkler is a relatively new concept that provides autonomous management capability for watering on farms and gardens. It is considered as a key element of smart farming and smart agriculture. Autonomous sprinkler system can effectively handle the limitations of the conventional Sprinkler system. Hence, it controls watering for specific soils, plants, status of plants, evolving weather condition, and many other factors.

With autonomous sprinkler, users do not need to explicitly set or change the watering schedule. It operates completely without the intervention of users.

3.5.1. Hardware Support for Autonomous Operations

The autonomous capability of the system is enabled by interoperating new types of hardware devices and a control software. The system requires a minimal hardware support as the followings.

- Water-proof *Cameras* to be installed on the target farm/garden
  To capture images of sprinkler operations, plant trunk, leaves, and flowers.
- *Thermometers* measuring the temperatures of Atmosphere and Ground/Soil
- *Hygrometers* measuring the humidity of Atmosphere and Ground/Soil

Examples of WIFI-based camera and sensor are shown in the following figure.

Water-proof Garden Camera

Moisture Sensor for Soil

Thermometer & Humidity Sensors

Thermometer for Soil

There is no requirement for additional actuators beyond the conventional actuators. However, the deployment of smart actuators with WIFI-connectivity is desirable to enable the true Autonomicity of the system.

- Smart Water Value IoT devices providing API for setting the values
- *Smart Sprinkler IoT devices* providing API for setting the directions and water level of sprinklers

3.5.2. Key Features of Autonomous Sprinkler System

The key features of autonomous sprinkler system are listed below.

- **Monitoring the Environmental Conditions**

  The system monitors the conditions of the plants and soils on target watering zones by acquiring contexts through sensors and cameras.

- **Analyzing the Contexts to Determine Situations**

  The system analyzes the set of collected contexts and determines the current situation of the watering zones.

- **Planning Watering Actions**

  The system devises an action plan that is the most appropriate for the determined situation. The plan will be represented as a watering schedule that is tailed for the situation.

- **Executing the Plan**

  The system executes the action plan, i.e., the watering schedule by itself. That is, the right intensity of watering is applied to different watering zones.

- **Updating the Knowledgebase of the System**

  The system evaluates the result of executing the plans and updates its knowledgebase by referring to the evaluation results. That is, the knowledgebase is updated by reflecting the positive effect or negative effect of executed plans.

3.5.3. Benefits of Autonomous Sprinkler System

The system provides the following benefits.

- **Healthier Plants**

  The system optimizes the watering schedule for the current situations of plants and soils. Hence, it could eliminate the problems of underwatering and overwatering. And, the plants are supplied the optimal level of water on time. As the result, the plants can stay healthy and higher harvest productivity is achieved.

- **Reduced Cost of Water**

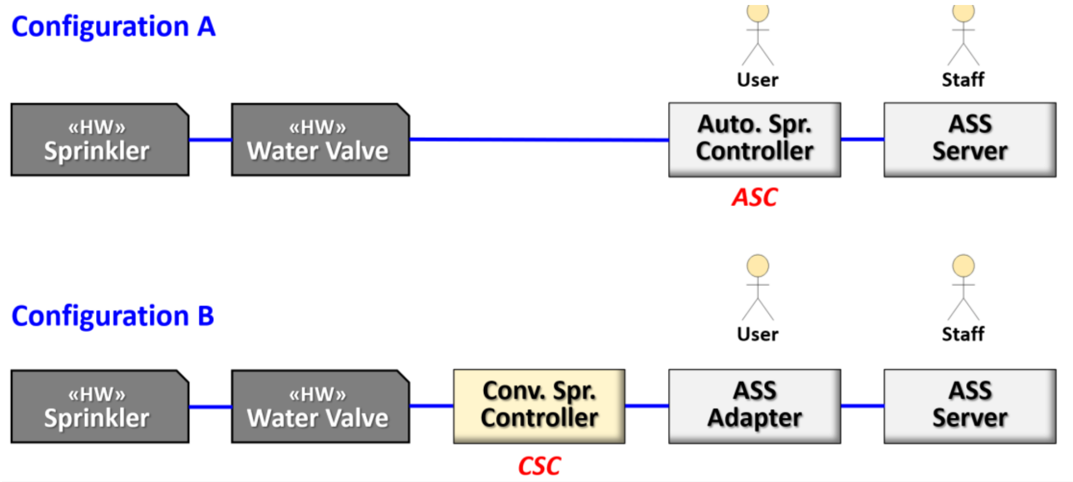
  By eliminating the problem of overwatering while keeping the plants healthy, the cost of water is reduced to the minimal.

- **Removing the Burden for Manually Settings**

  Since the system sets the watering schedule by itself, users do not have the burden of setting and changing the schedule manually.

3.6. Deployment of the Target System

The target system can be deployed in two different ways as shown in the following figure.



3.6.1. Configuration A (using Autonomous Sprinkler Controller)

This is the most ideal and optimal configuration since a specially designed *Autonomous Sprinkler Controller (ASC)* is used instead of *Conventional Sprinkler Controller (CSC)*. The ASC is installed in a house or a farm, and it is connected to the water values as shown in the figure above.

The *ASC* unit has the autonomous capability, and it communicates with the server for storing the acquired contexts and downloading the up-to-date knowledge used in ASC. The ASC unit acquires environmental contexts from the sensors and the environmental images form the cameras using WIFI network. After analyzing the contexts, it determines the optimal watering schedule and controls the water values to apply the watering schedule.

The back-end server, *ASS Server*, is to store the data repository of the collected contexts and the applied schedules and to perform advanced analytics on the system operations. Based on the analytics results, the staffs on the server side may develop and deploy an enhanced version of *ASC* elements such as the key knowledgebase.

3.6.2. Configuration B (using Conventional Sprinkler Controller)

This configuration is to utilize the existing sprinkler controller, i.e., *Conventional Sprinkler Controller (CSC)*, but to enable the autonomous control of watering. There is an issue of interoperability between the target system and *CSC*, and the issue is resolved by deploying an *Autonomous Sprinkler System (ASS)* Adapter, as shown in figure above.

The ASS Adapter is placed between the Server and the CSC, and it has an interface for adopting specific types of conventional sprinkler system. By implementing the specified interface, the interoperability between the autonomous sprinkler system and a CSC is established.

For the simplicity of the design in this CEP, we consider only the configuration A.

4. Functional Requirements

The functionality of the system is classified into the following functional categories.

4.1. User Profile Management

This functionality is to manage the profiles of users who operates the Autonomous Sprinkler system at a garden or a farm. It includes the identification, contact, and login information of users.

4.2. Staff Profile Management

This functionality is to manage the profiles of staffs who operates the server system. It includes the identification, contact, and login information of users. Note that staffs do not operate the sprinkler systems but manage the service-side functionality for archiving the contexts and updating the knowledgebase.

4.3. Hardware Configuration Registration

This functionality is to register various types of hardware devices connected to the target system.

- ❑ **Registering Sprinkler Controller**
- ❑ **Registering Water Values and their connections**
- ❑ **Registering Sprinklers and their connections**
- ❑ **Various types of Sensors**
- ❑ **Water-proof Cameras**
- ❑ **(Optional) Actuators and IoT Devices**

4.4. Registration of Layout and Managed Objects

This functionality is to specify the layout of a farm or a garden to be managed. The layout information is utilized by the system in locating the locations of watering zones, sprinkler heads, sensors, cameras, and actuators.

There can be various types of farms or garden, and hence the system should be designed to accommodate the variability. For example, a house may have a front-yard garden and a back-yard garden, and another house may only have a back-yard garden.

- **Registration of Locations**

  The locations of managed plants, sensors, cameras, sprinkler heads, and water values should precisely be specified.

- **Registration of Plants**

  This functionality is to register plants to be managed, i.e., managed objects. The type of plant, name of plant, location, characteristics, and any other description should be specified for autonomic management.

4.5. Monitoring of the Environment

This functionality is to acquire contexts of the environment using sensors and cameras. The acquisition of contexts enables the detailed monitoring of the plants.

4.6. Analyzing the Situation through Contexts

This functionality is to analyze the acquired contexts and determine the situation of each managed object such as plants. The analytics on the contexts can be performed in various ways, but advanced analytic methods should be utilized to yield a high accuracy of context analysis.

The determined situation should be specified in a well-formed format for capturing the comprehensive information of the situation. The representation format of situations should be proposed by the development team.

4.7. Planning the Actuation

This functionality is to devise a plan, i.e., watering schedule, that is the most appropriate for the determined situation. An actuation plan consists of one or more actions, and each action can be a watering action or a sprinkler calibration action.

A watering action is specified as a tuple of (Watering Zone, Water Value, Start Time, Duration, Strength of Water Stream, and other options).

A sprinkler calibration action is specified as a tuple of (Sprinkler ID, new height, left-most angle, right-most angle, water strength, and other options).

Advanced recommendation models for generating the plans should be utilized.

4.8. Executing the Actuation

This functionality is to execute the generated plan, i.e., performing watering according to the generated schedule. As the result, plants and soils are provided with an appropriate level of water.

A sprinkler calibration action can be performed automatically for IoT-type sprinklers. For conventional sprinklers, a user has to manually adjust the sprinkler as directed by the system.

4.9. Managing the Knowledgebase

This functionality is to manage the knowledgebase of the autonomous sprinkler system.

- **Building the Initial Knowledgebase**

  This is to build the initial knowledgebase to enable the normal operations.

- **Updating the Knowledgebase**

  This is to update the knowledgebase by reflecting the results of autonomously generated watering schedule.

4.10. Report Generation

This functionality is to generate various reports on the system operations. A set of pre-specified reports are periodically generated. Reports can also be generated on demand.

5. Non-Functional Requirements

There can be several non-functional requirements that are essential in the target system. For CEP, we consider only 2 NFR items.

5.1. NFR-1. Accuracy of Determining the Situation

Design the system to provide a high accuracy of determining the situation of various managed objects such as plants and flowers. The system provides a self-monitoring capability using sensors and cameras. Then, the system analyzes the acquired contexts to determine situation of each managed object.

The accuracy of determining the situation is an essential factor for the overall quality of the system. This is because the watering schedule is devised based on the determined situation. If the situation is correctly determined, the watering schedule for the situation would be useless and may create problems on farms and garden.

Examples of determined situations can be the following.

- Healthy Condition of Plant 'A'
- Plant 'B' excessively dried
- Soil on watering zone 'X' is too wet due to overwatering.
- Temperature of watering zone 'Y' is too high, demanding more watering
- Two plants of same type 'C' and 'D' are in different health conditions, due to uneven spraying of water.

5.2. NFR-2. High Reliability of the Server

Design the system to provide a high reliability for the Autonomous Sprinkler System. The reliability is essential since the server is accessed by all client instances of autonomous sprinkler system. Any malfunctioning on the server will have impacts on the client instances.

The reliability is determined by the degree of avoiding the system failures, being tolerant on faults, and the efficiency of recovering the software and its datasets upon a failure. That is, the reliability of the system should be met by addressing its sub-quality factors.

- Maturity
- Fault-Tolerance
- Recoverability

Any effective methods for high reliability should be explored and applied to the development.

6. Guidelines for Conducting CEP

6.1. Guidelines for Designing the Architecture

Apply the following guidelines for writing CEP Reports.

- **Conformance to the given SRS**

  The submitted AD should conform to the given SRS.

- **Conformance to UML Standards**

  The submitted AD should conform to the notational and usage standards of UML.

- **Consistency among various Artifacts in AD**

  There should be a high consistency among various artifacts (such as diagrams) in the submitted AD.

- **Comprehensibility of Textual Description**

  The textual elaboration of the architecture design should be written in accurate, precise, and condensed way. Hence, the understandability of the AD becomes high. The textual description can be written in English, Korean, or their mixture.

- **Readability of Figures and Tables**

  The figures and tables should be easily readable by applying good formats, right font size and special effects on them. For example, a use case diagram with 100 use cases should be well structured and enlarged if needed.

- **Reasonable Details of Machine Learning design**

  The submitted AD would include a design for managing machine learning models. The description of the machine learning model generation should be written in reasonably details. The description typically includes machine learning algorithms utilized, training sets used, the details for designing the model generation components.

- **Originality of the AD**

  The submitted AD should be an individual work. Any same or highly similar solutions would get a score penalty.

6.2. Guidelines for Submitting Reports

Apply the following guidelines for writing CEP Reports.

- **Due Dates for Submission**

  The due dates and times for each CEP report are specified. The CEP reports should be submitted by the due. Late submissions of CEP reports are not accepted.

- **Format of the CEP Report**

  Use the word processor, MS Word, for formatting your CEP reports. Submit the word files, not the PDF files.

- **Submission**

  Submit your CEP report to the course manager, not to the instructor.

6.3. Evaluation Form for CEP Reports

The following form is used to evaluate the CEP report.

Evaluation of CEP Report

Name: 홍길동

Criteria	Max	Earned
Ch.1, Introduction	2	2
Ch.2, (A1) Architectural Requirement Refinement	2	2
Ch.3, (A2) System Context Analysis		
System Boundary Context	3	3
Functional Context	5	5
Information Context	5	5
Behavior Context	5	5
Ch.4, (A3) Skeleton Architecture Design		
Justification of Candidate Architecture Styles	6	6
Integrating Selected Styles into Architecture	2	2
Ch.5, (A4) Architecture with Views		
Applying Functional Viewpoint	8	8
Applying Information Viewpoint	8	8
Applying Behavior Viewpoint	8	8
Applying Deployment Viewpoint	6	6
Ch.6, (A5) Architecture with Quality-driven Design		
Design for NFR #1, Applying Process	8	8
Design for NFR #1, Quality Delivered	7	7
Design for NFR #2, Applying Process	8	8
Design for NFR #2, Quality Delivered	7	7
Ch.7, (A6) Architecture Evaluation		
Intermediate Steps of applying Evaluation	6	6
Correctness of the Evaluation	4	4
CEP Score	100	100

- **Strength**
  -
- **Weakness**
  -