Sure-Park Project

Architectural Design Document

Version 1.0

|  |  |
| --- | --- |
| **Team** | **Infinite Challenge(Team 3)** |
| **Members** | Namjin Lee, Jack Oh, Charles Park, Joan Kim, Jaeheon Kim |

**History**

|  |  |  |
| --- | --- | --- |
| **Ver.** | **Date** | **History** |
| 1.0 | 19, June, 2016 | Initial |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[**1.** **System Context** 4](#_Toc454318342)

[**2.** **1st Decomposition** 6](#_Toc454318343)

[**2.1.** **Physical View of 1st Decomposition** 6](#_Toc454318344)

[**2.2.** **Dynamic View of 1st Decomposition** 6](#_Toc454318345)

[**2.3.** **Static View of 1st Decomposition** 7](#_Toc454318346)

[**2.4.** **Entity Catalog** 7](#_Toc454318347)

[**2.5.** **Rationale** 8](#_Toc454318348)

[**3.** **2nd Decomposition** 8](#_Toc454318351)

[**3.1.** **SurePark Manager** 8](#_Toc454318352)

[**3.1.1.** **Dynamic view of SurePark Manager** 8](#_Toc454318353)

[**3.1.2.** **Static view of SurePark Manager** 10](#_Toc454318354)

[**3.1.3.** **Entity Catalog** 10](#_Toc454318355)

[**3.1.4.** **Rationale** 10](#_Toc454318356)

[**3.2.** **Facility Controller** 11](#_Toc454318357)

[**3.2.1.** **Dynamic view of SurePark Manager** 11](#_Toc454318358)

[**3.2.2.** **Dynamic view of SurePark Manager** 12](#_Toc454318359)

[**3.2.3.** **Entity Catalog** 12](#_Toc454318360)

[**3.2.4.** **Rationale** 12](#_Toc454318361)

[**3.3.** **Web Service** 13](#_Toc454318362)

[**3.3.1.** **Dynamic view of Web Service** 13](#_Toc454318363)

[**3.3.2.** **Static view of Web Service** 14](#_Toc454318364)

[**3.3.3.** **Entity Catalog** 14](#_Toc454318365)

[**3.3.4.** **Rationale** 14](#_Toc454318366)

[**3.4.** **Facility Controller(FC) to SurePark Manager(SM)** 15](#_Toc454318367)

[**3.4.1.** **How to check if Facility Controller is alive** 15](#_Toc454318368)

[**3.5.** **SurePark Manager to Web Service** 16](#_Toc454318369)

[**3.5.1.** **How to resist attack from an attacker?** 16](#_Toc454318370)

[**4.** **Detail Design** 16](#_Toc454318371)

[**4.1.** **FC-SM packet** 16](#_Toc454318372)

[**4.1.1.** **Packet Structure** 16](#_Toc454318373)

[**4.1.2.** **Detailed Packet Scenario** 18](#_Toc454318374)

[**4.2.** **SM-WS packet** 20](#_Toc454318375)

[**4.3.** **SM-WS DB Schema** 23](#_Toc454318376)

[**4.4.** **SurePark Manager Detailed Design** 24](#_Toc454318377)

[**16.1.** **Facility Controller Detailed Design** 28](#_Toc454318378)

[**16.2.** **Web Service Detailed Design** 30](#_Toc454318379)

[**17.** **Future Works** 32](#_Toc454318380)

[**10. Lesson Learned** 32](#_Toc454318381)

# **System Context**

우리의 목표는 안정되고, 확장이 쉽고, 보안이 훌륭한 Sure Park System을 개발하는 것이다. Sure Park System은 H/W Part로 Parking Garage를 제어할 수 있어야 하고, 주차 예약, 주차 감시 및 관리 시스템을 제공해야 한다.

The Sure Park system



< Figure1. System Context >

# **1st Decomposition**

## **Physical View of 1st Decomposition**



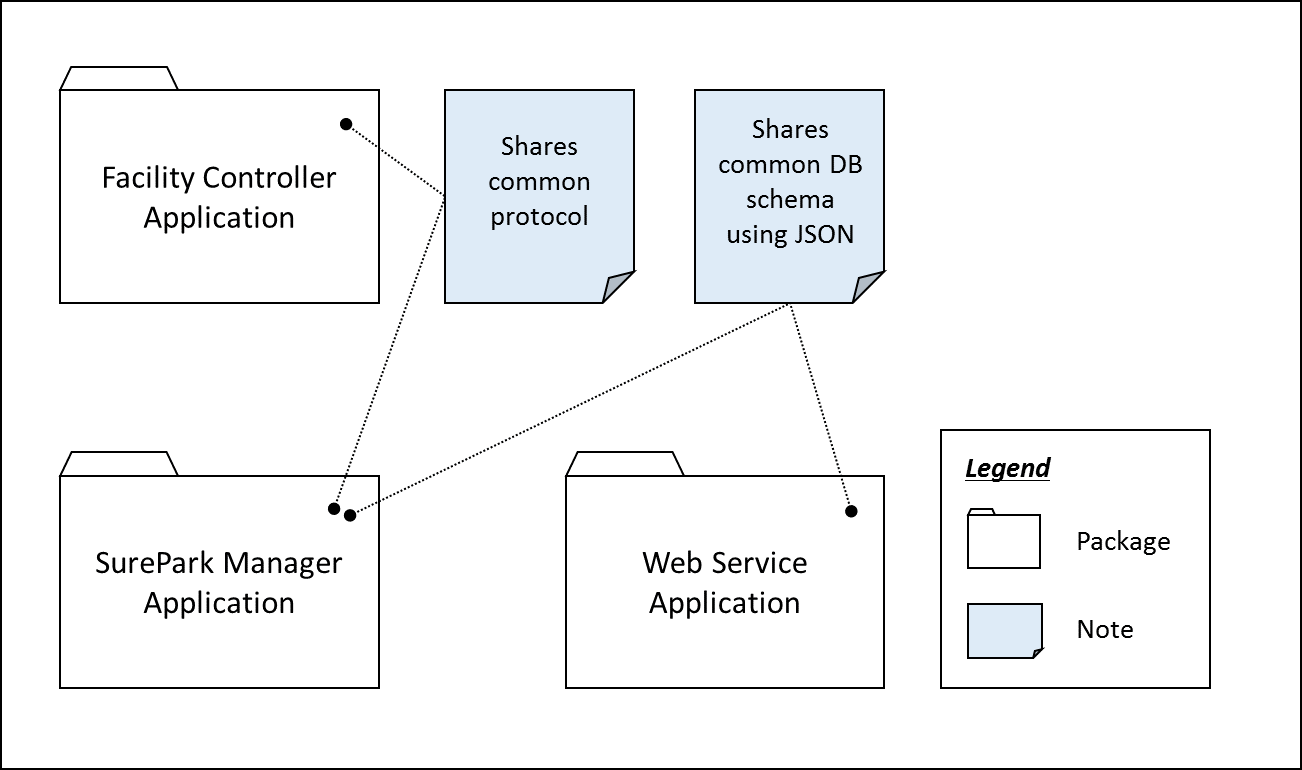
< Figure 2. Physical view of 1st decomposition >

## **Dynamic View of 1st Decomposition**



< Figure 3. Dynamic view of 1st decomposition >

## **Static View of 1st Decomposition**



< Figure 4. Static view of 1st decomposition >

## **Entity Catalog**

|  |  |
| --- | --- |
| **Entity** | **Description** |
| Web Browser | Users, attendants and owner can access their own UI through the web browser provided by the web server. |
| Web Service | Provides users with the functions of sign-up, log in, reservation, monitoring facilities and/or showing parking statistics based on data retrieved from SurePark DB.  Sends information to SurePark Manager for DB updates. |
| Facility Controller | Controls parking facilities; get the status of parking slots, turn on/off LEDs, detect a car at the gates and open/close the gates.  Receives data from SurePark Manager to control LEDs and/or gates.  Sends data to SurePark Manager to update the status of parking slots. |
| SurePark Manager | Handles show-up and no-show scenarios based on DB information.  Updates SurePark DB when a user has signed up, a reservation has been made or facility status has been changed. |
| SurePark DB | Keeps all of the data about users, garages and reservations.  Only can be updated by SurePark Manager. |

## **Rationale**

Modifiability (QA08) is one of the most important QAs of the SurePark system. An engineer needs to scale up the system within a week. We have divided the whole system into 5 parts according to their responsibilities, and applied client-server and repository pattern to connect each parts.



< Figure 5. Architectural patterns of SurePark System >



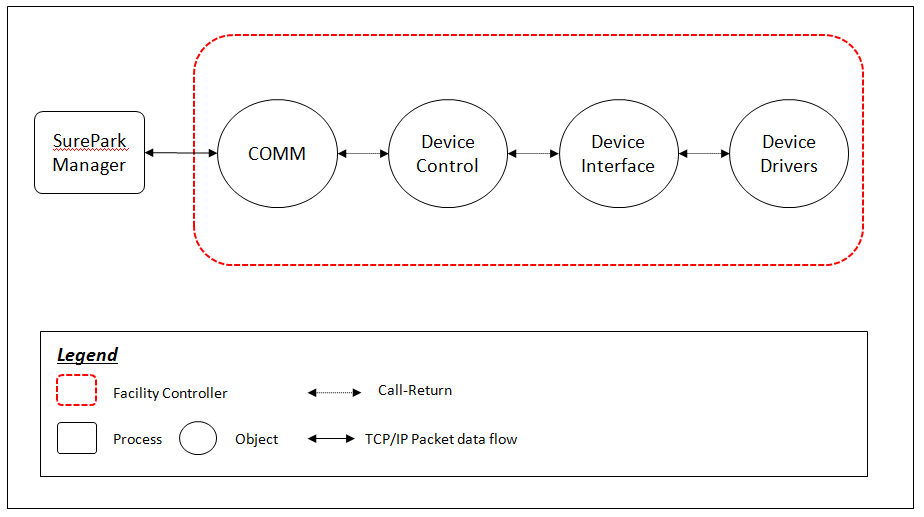
# **2nd Decomposition**

## **SurePark Manager**

### **Dynamic view of SurePark Manager**

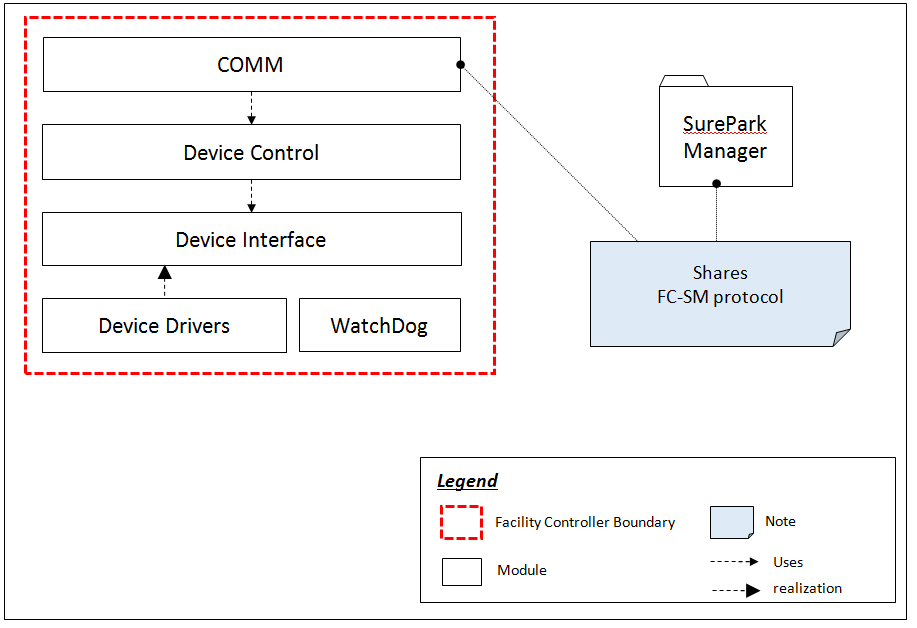
## **Facility Controller**

### **Dynamic view of SurePark Manager**



< Figure?. Dynamic view of Facility Controller >

### **Dynamic view of SurePark Manager**



< Figure?. Static view of Facility Controller >

### **Entity Catalog**

|  |  |
| --- | --- |
| **Entity** | **Description** |
| Comm | Communication module between surepark manager and facility controller |
| Device Control | Module for Non feed/back device control(exit/entry gate LED, exit/entry servo on/off)  This module observes state of devices.  When the state changed, then send message to superpark manager |
| device interface | Set states of exit/entry gate LED, exit/entry servo and stall LEDs by deviceControl  Get states of stallsensors and exit/entry sensors when observer request |
| Device driver | Control ON/OFF or OPEN/CLOSE(exit/entry gate LED, exit/entry servo and stall LEDs)  Sensing value of stallsensors and exit/entry sensors |
| WatchDog | A watchdog timer is used to detect and recover from mcu malfunctions. During normal operation, the mcu regularly restarts the watchdog timer to prevent it from "timing out". |

### **Rationale**

The structure is layered type. We use interfaces so that new devices is be added. thus, We can reduce dependency. Developers can design scale up/out the system easy. (Modifiability(QA08))

## **Web Service**

### **Dynamic view of Web Service**

# **Detail Design**

## **FC-SM packet**

### **Packet Structure**

## **Facility Controller Detailed Design**

### **Design pattern of Facility Controller**



#### Rationale

For easy adding or removing devices, we use observer pattern. Device driver is sensing every 0.5sec and Observer is polling state of sensor. When state of sensor changed, Observer notify comm..

Finally, Comm. send message to surepark manager.

#### Interface design

1->3step : generalization to on/off

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 step (device name : count) | 2 step | 3 step |
| non feed back device (OUTPUT) | Stall LED : 4 | ON/OFF | ON/OFF |
| Extit LED : 1 | ON/OFF | ON/OFF |
| Entry LED : 1 | ON/OFF | ON/OFF |
| Exit Servor : 1 | OPEN/CLOSE | ON/OFF |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 step (device name : count) | 2 step | 3 step | 4 step | 5 step |
| feed back device (INPUT) | Stall Sensor : 4 | value : 0~256 | IF value\_average - offset value >= 50  then no car, else car is | Parked/No car | HIGH/LOW |
| Entry Gate Sensor : 1 | HIGH/LOW | HIGH/LOW | HIGH/LOW | HIGH/LOW |
| Exit Gate Sensor : 1 | HIGH/LOW | HIGH/LOW | HIGH/LOW | HIGH/LOW |

1->5step : generalization to HIGH/LOW

### **Detailed Parking Scenario : Initial and idle state**



### **Detailed Parking Scenario : Parking and Get Out car**

