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](#_Toc454208900)

[**2.** **1st Decomposition** 5](#_Toc454208901)

[**3.** **SurePark Manager** 8](#_Toc454208902)

[**4.** **Facility Controller** 10](#_Toc454208903)

[**5.** **Web Service** 10](#_Toc454208904)

[**6.** **Facility Controller to SurePark Manager** 10](#_Toc454208905)

[**6.1.** **How to check if Facility Controller is alive.** 10](#_Toc454208906)

[**7.** **SurePark Manager to Web Service** 11](#_Toc454208907)

[**8.** **Detail Design** 11](#_Toc454208908)

[**8.1.** **FC-SM packet** 11](#_Toc454208909)

[**8.2.** **SM-WS packet** 11](#_Toc454208910)

[**8.3.** **SM-WS DB Schema** 11](#_Toc454208911)

[**8.4.** **SurePark Manager Detailed Design** 11](#_Toc454208912)

[**8.5.** **Facility Controller Detailed Design** 11](#_Toc454208913)

[**8.6.** **Web Service Detailed Design** 11](#_Toc454208914)

[**9.** **Future Works** 11](#_Toc454208915)

[**10.** **Lesson Learned** 11](#_Toc454208916)

# **System Context**



< Figure1. System Context >

# **1st Decomposition**

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



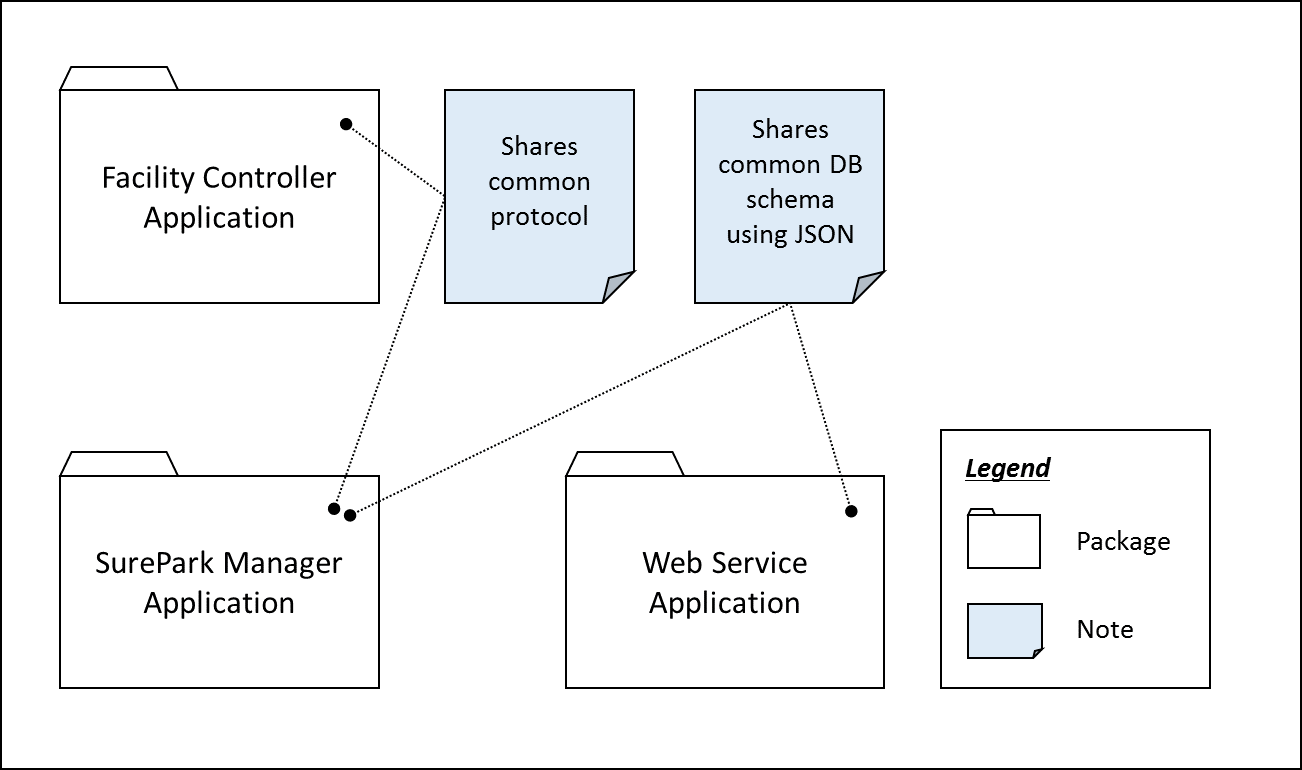
< Figure 2. Physical view of 1st decomposition >

## **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**

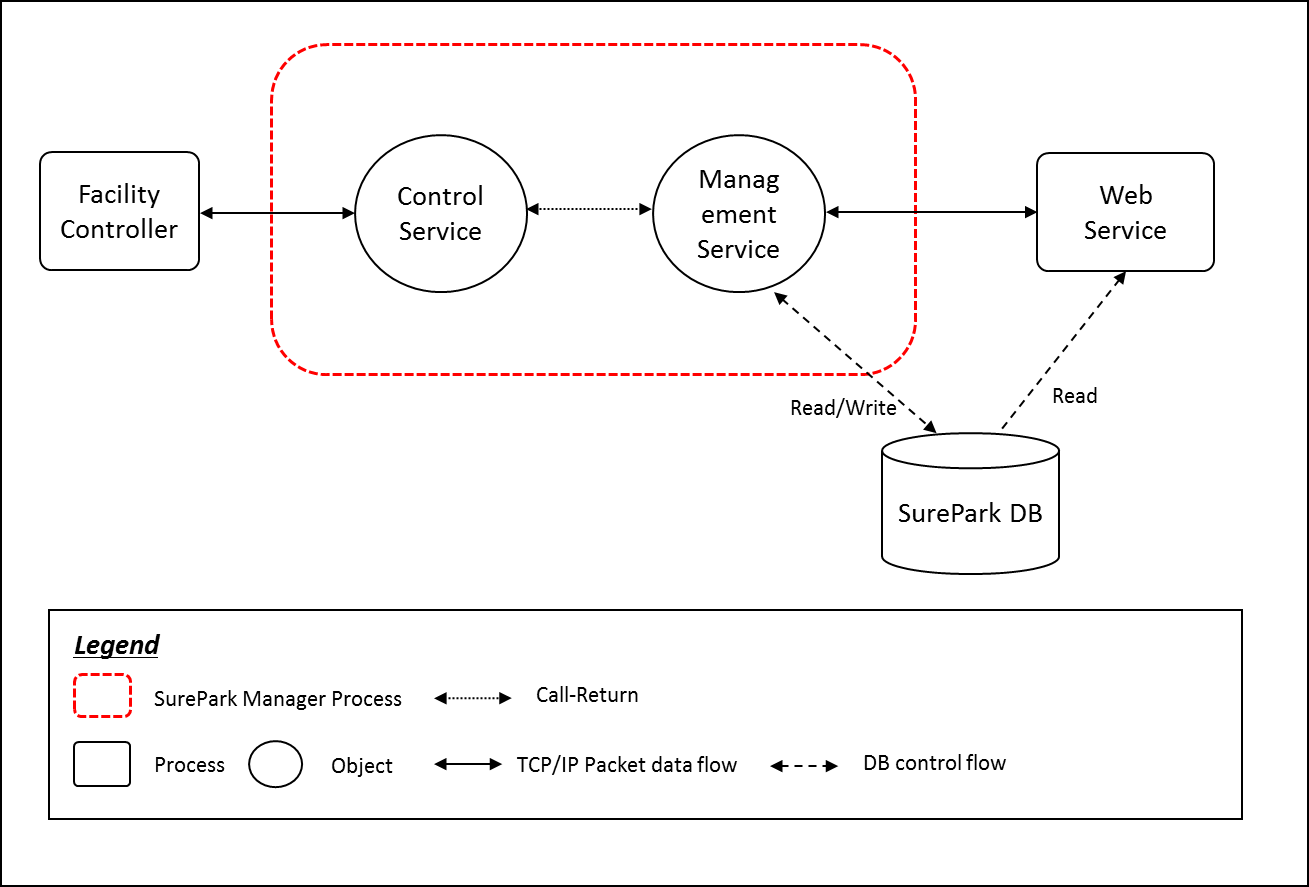
Modifiability is one of the most important QAs of the SurePark system. An engineer needs to scale up the system within a week. We have divided the whole system into 5 parts according to their responsibilities, and applied client-server and repository pattern to connect each parts.



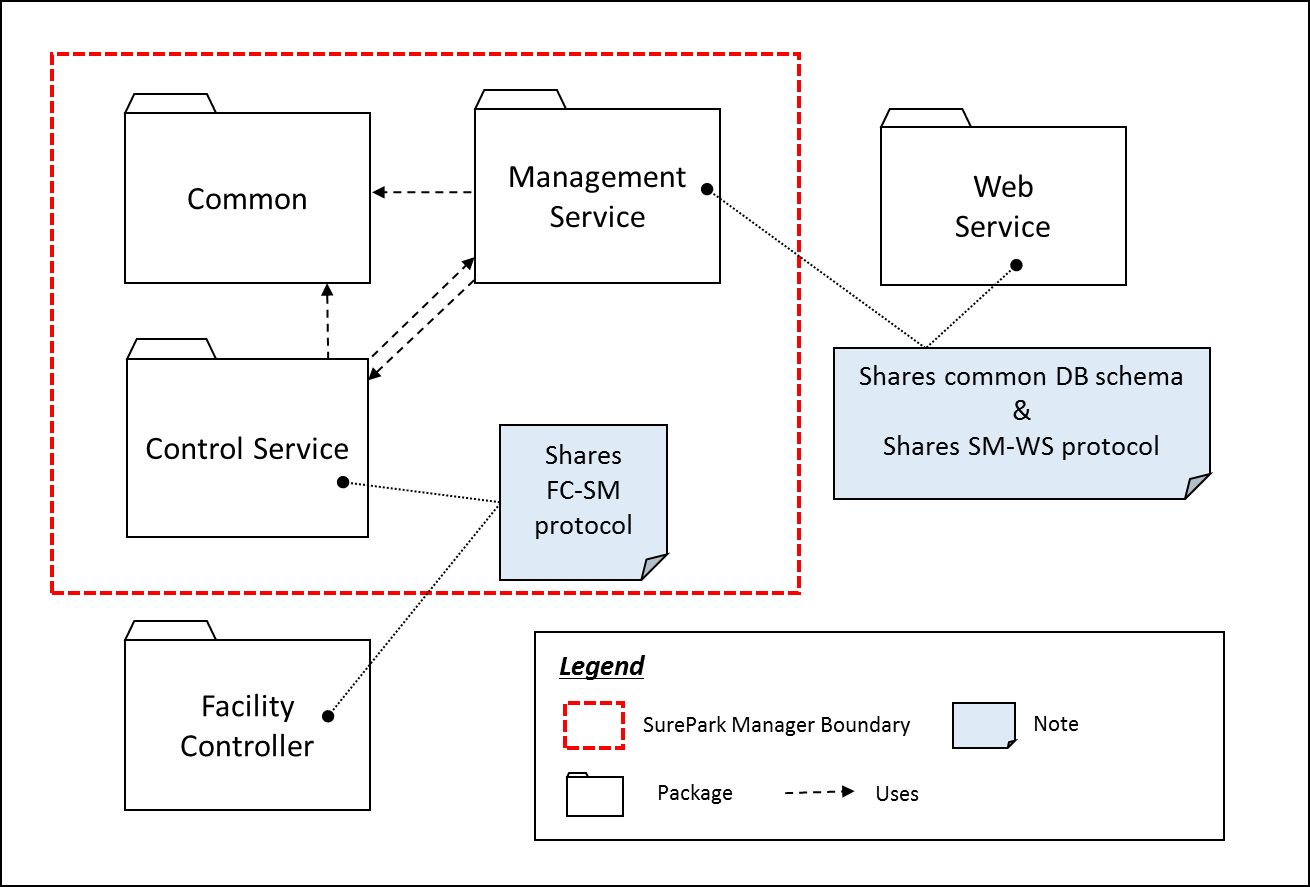
< Figure 5. Architectural patterns of SurePark System >



# **SurePark Manager**



< Figure6. Dynamic view of SurePark Manager >



< Figure6. Static view of SurePark Manager >

|  |  |
| --- | --- |
| **Modifiability(QA08)** | **Perspective: Static** |
| Architectural Pattern | Increase Cohesion, Reduce Coupling |
| Rationale | In SurePark Manager, we’ve divided three parts, Common, Control Service and Management Service. Control Service have a responsibility of communication with the Facility Controller. And it also have managed Facilities. The Management Service have a responsibility of communication with the Web Service and also manage DB.  Common is common source between Control Service and Management Service. Control service and Management service call each interface class for reducing coupling. |

|  |  |
| --- | --- |
| **Entity** | **Description** |
| Common |  |
| Control Service |  |
| Management Service |  |

# **Facility Controller**



< Figure?. Static view of Facility Controller >

|  |  |
| --- | --- |
| **Modifiability(QA08)** | **Perspective: Static** |
| Architectural Pattern | Layered |
| Rationale | we’ve divided that use interfaces between DeviceControl/Observer and Devices.  So, We reduce dependency. Finally, Developers can design scale up/out the system easy. |

|  |  |
| --- | --- |
| **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) |
| Observer | This module observes state of devices.  When the state changed, then send message to superpark manager |
| Non f/b device interface | Set states of exit/entry gate LED, exit/entry servo and stall LEDs by deviceControl |
| f/b device interface | Get states of stallsensors and exit/entry sensors when observer request |
| Non f/b device | Control ON/OFF or OPEN/CLOSE(exit/entry gate LED, exit/entry servo and stall LEDs) |
| f/b device | Sensing value of stallsensors and exit/entry sensors |

# **Web Service**

# **Facility Controller to SurePark Manager**

## **How to check if Facility Controller is alive.**

|  |  |
| --- | --- |
| **Availability(QA02)** | **Perspective: Dynamic** |
| Architectural Pattern | Client-Server structure with heartbeat tactic. |
| Rationale | Facility Controller send a packet every 10 seconds to SurePark Manager. If Controller Service doesn’t get this packet within 20 seconds, the SurePark Manager notify it to SurePark WebService for alarming to attendant for fixing the system. |



# **SurePark Manager to Web Service**

# **Detail Design**

## **FC-SM packet**

## **SM-WS packet**

## **SM-WS DB Schema**

## **SurePark Manager Detailed Design**

## **Facility Controller Detailed Design**



< Figure?. Sequence Diagram of Facility Controller when initial/idle state >



< Figure?. Sequence Diagram of Facility Controller when parking/Get out >

## **Web Service Detailed Design**

# **Future Works**

# **Lesson Learned**