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** 7](#_Toc454208902)

[**4.** **Facility Controller** 9](#_Toc454208903)

[**5.** **Web Service** 9](#_Toc454208904)

[**6.** **Facility Controller to SurePark Manager** 9](#_Toc454208905)

[**6.1.** **How to check if Facility Controller is alive.** 9](#_Toc454208906)

[**7.** **SurePark Manager to Web Service** 10](#_Toc454208907)

[**8.** **Detail Design** 10](#_Toc454208908)

[**8.1.** **FC-SM packet** 10](#_Toc454208909)

[**8.2.** **SM-WS packet** 10](#_Toc454208910)

[**8.3.** **SM-WS DB Schema** 10](#_Toc454208911)

[**8.4.** **SurePark Manager Detailed Design** 10](#_Toc454208912)

[**8.5.** **Facility Controller Detailed Design** 10](#_Toc454208913)

[**8.6.** **Web Service Detailed Design** 10](#_Toc454208914)

[**9.** **Future Works** 10](#_Toc454208915)

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

# **System Context**



< Figure1. System Context >

# **1st Decomposition**

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



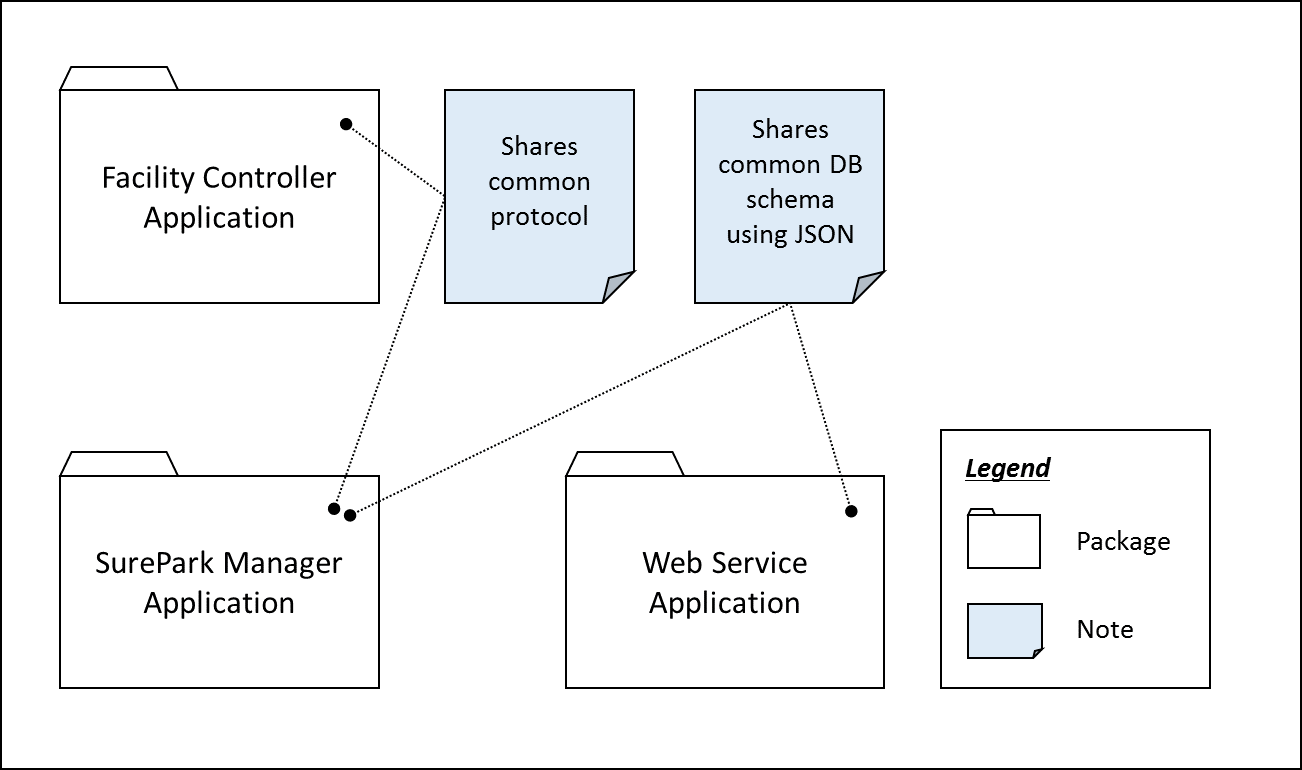
< Figure 2. Physical view of 1st decomposition >

## **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**

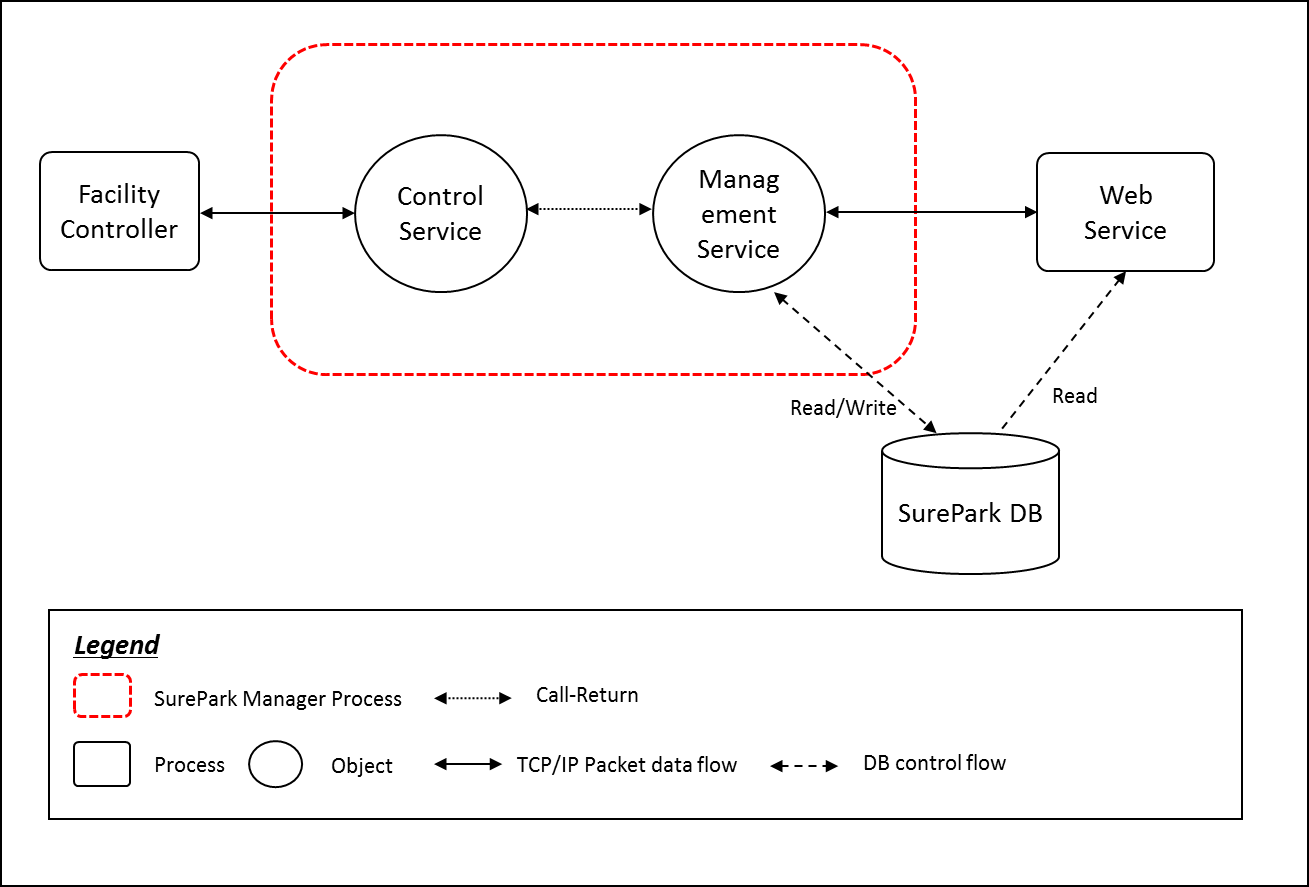
Modifiability is one of the most important QAs of the SurePark system. An engineer needs to scale up the system within a week. We have divided the whole system into 5 parts according to their responsibilities, and applied client-server and repository pattern to connect each parts.



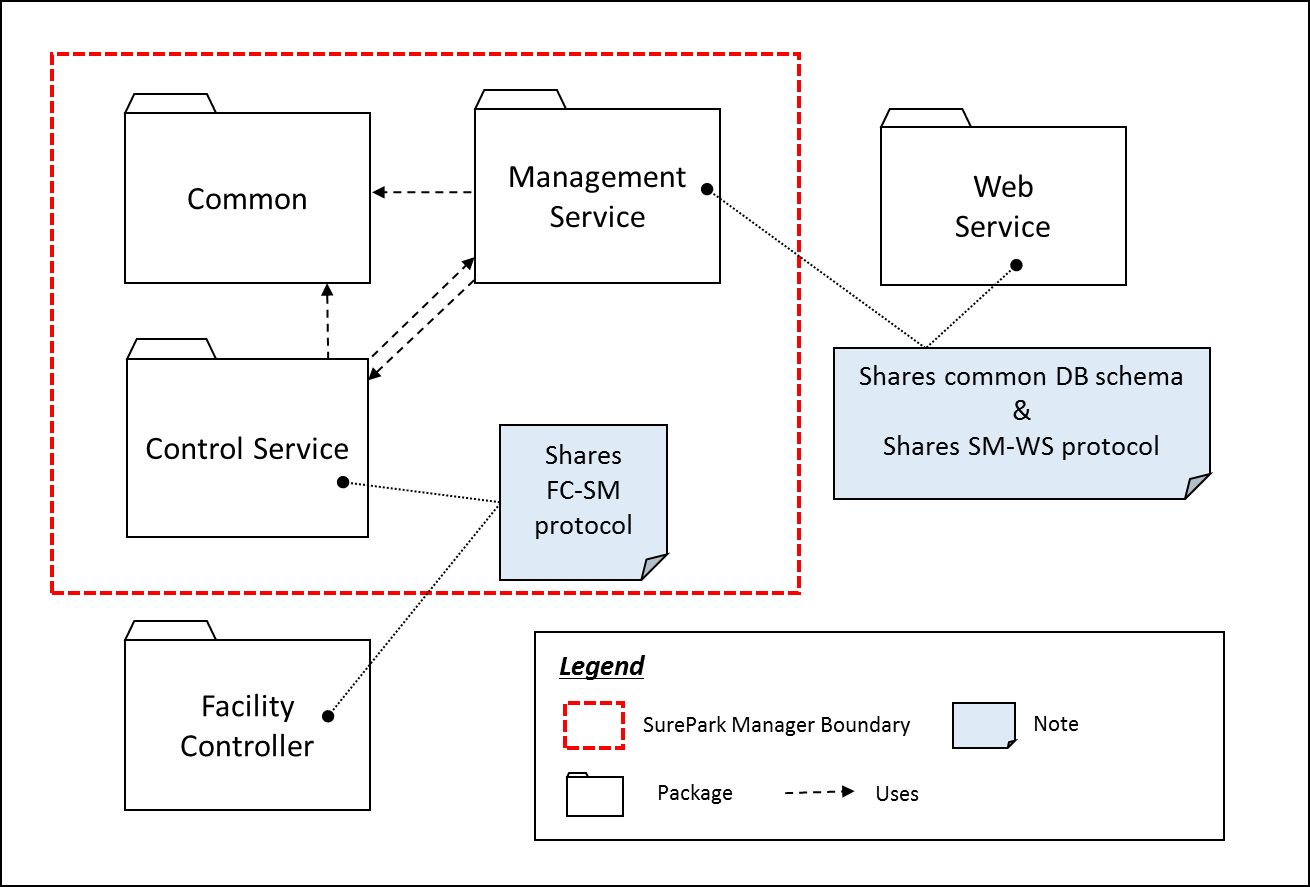
< Figure 5. Architectural patterns of SurePark System >



# **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**

# **Web Service**

# **Facility Controller to SurePark Manager**

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

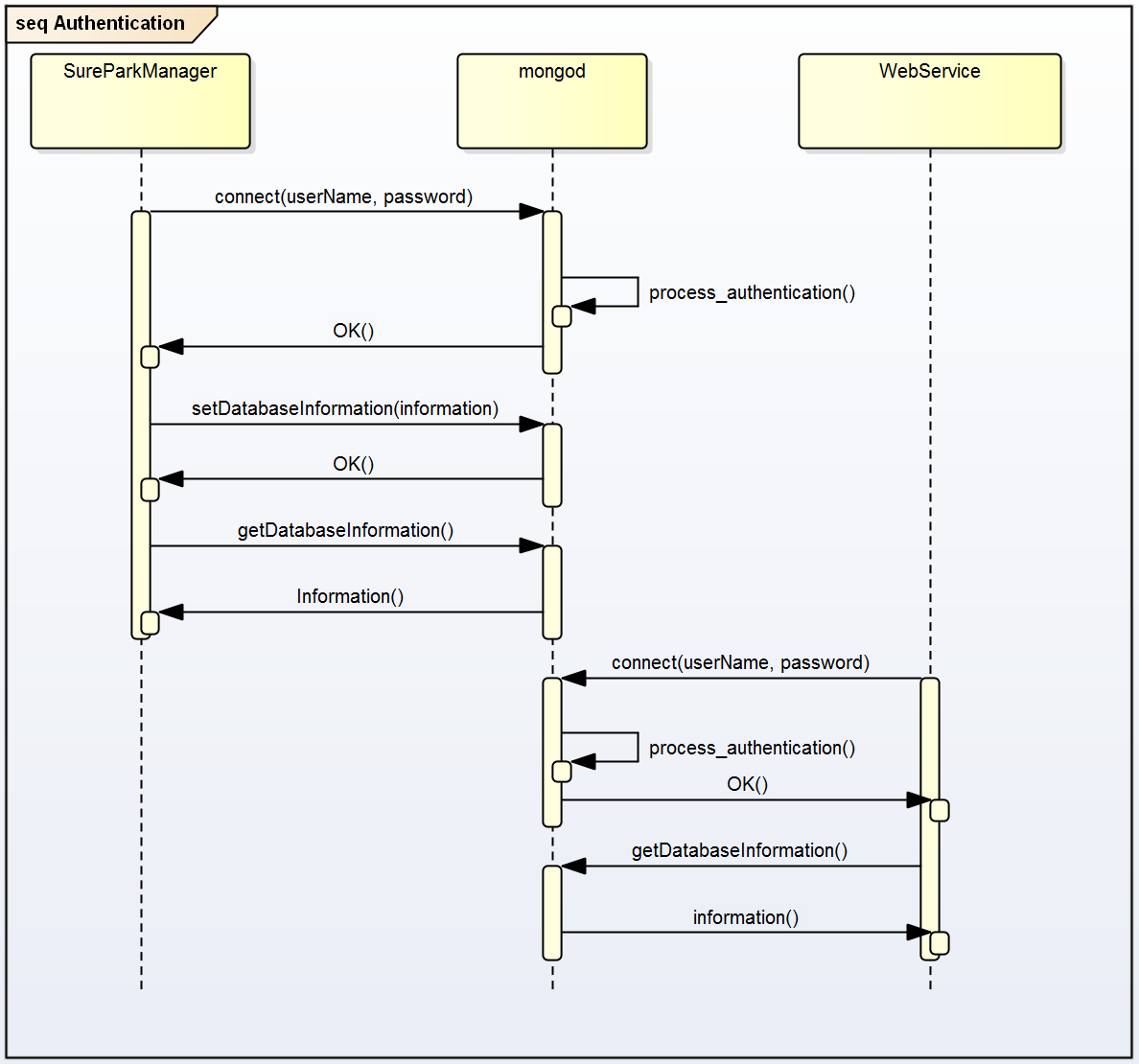
|  |  |
| --- | --- |
| **Availability(QA02)** | **Perspective: Dynamic** |
| Architectural Pattern | Client-Server structure with heartbeat tactic. |
| Rationale | Facility Controller send a packet every 10 seconds to SurePark Manager. If Controller Service doesn’t get this packet within 20 seconds,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



# **SurePark Manager to Web Service**

## **How to resist attack from an attacker?**

|  |  |
| --- | --- |
| **Security(QA03)** | **Perspective: Dynamic** |
| Architectural Pattern | Authenticate actors |
| Rationale | SurePark Manager and WebService use MongoDB for database. It supports a number of authentication mechanisms that clients can use to verify their identity. Only verified user can access database |



# **Detail Design**

## **FC-SM packet**

## **SM-WS packet**

Communication between SureParkManger and WebService use SMWS packet. The packet structure is used JSON for sending, and the all packet data is encapsulated to string. The communication use request-response mechanism. WebService request services, and SureParkManager do services and response services.

And the communication use notify mechanism also. SureParkManager notify events to WebService, but no response for this mechanism.

1. Request packet structure

|  |  |
| --- | --- |
| **Request command** | **Data** |
| String | JSON Object |
| JSON Object | |

1. Response packet structure

|  |  |
| --- | --- |
| **Response command** | **Acknowledge** |
| String | String |
| JSON Object | |

1. Notify packet structure

|  |  |
| --- | --- |
| **Request command** | **Data** |
| String | JSON Object |
| JSON Object | |

1. Commands

|  |  |
| --- | --- |
| **Command** | **Data** |
| newGarage | When new garage is installed, Web Service add to the garage’s information to SureParkManager. SureParkManager update garages database.  - Request packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | garageName | String | | garageNumber | String | | slotNum | String | | slotStatus | JSON Object | | gracePeriod | String | | parkingFee | String | | garageIP | String | | isAvailable | String |   Ex) {"newGarage": {"garageName":"AAAAA", "garageId":”1001”, "slotNumber":”4”, "slotStatus":["Open","Open","Open","Open"],  "gracePeriod":"90", "parkingFee":"30", "garageIP":"127.0.0.1", "isAvailable":true}}  - Response packet   |  |  |  | | --- | --- | --- | | **Data Field** | **Data type** | **Data** | | Acknowledge | String | “OK” or “FAIL” |   Ex) {"newGarage": "OK"} or {"newGarage": "FAIL"} |
| newReservation | When a driver makes a reservation, WebService send new reservation to SureParkManger. SureParkManager update reservations database.  - Request packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | userID | String | | cardInfo | String | | confirmInformation | String | | gracePeriod | String | | parkingFee | String | | usingGarage | String |   {"newReservation": {"userID":"jack", "cardInfo":"1111-\*\*\*\*-\*\*\*\*-4444", "confirmInformation":"A1234", "gracePeriod":"90", "parkingFee":"30", "usingGarage":"Sure-Park"}}  - Response packet   |  |  |  | | --- | --- | --- | | **Data Field** | **Data type** | **Data** | | Acknowledge | String | “OK” or “FAIL” |   Ex) {" newReservation ": "OK"} or {" newReservation ":"FAIL"} |
| cancelReservation | When a driver cancels a reservation, WebService send cancel reservation to SureParkManger. SureParkManager update reservations database.  - Request packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | userID | String | | confirmInformation | String | | usingGarage | String |   Ex) {"cancelReservation": {"userID":"jack", "usingGarage":"Sure-Park", "confirmInformation":"A1234"}}  - Response packet   |  |  |  | | --- | --- | --- | | **Data Field** | **Data type** | **Data** | | Acknowledge | String | “OK” or “FAIL” |   Ex) {" cancelReservation ": "OK"} or {" cancelReservation ":"FAIL"} |
| newUser | When new user signed up, Web Service send new user sign up information to SureParkManager. SureParkManager update users database.  - Request packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | userID | String | | userPassword | String | | userType | String | | userName | String | | userEmail | String | | displayName | String |   Ex) {"newUser": {"userID":"myID", "userPassword":"myPassword", "userType":"user", "userName":"myName", "userEmail": “myEmail@email.com", "displayName":"MY NAME"}}  - Response packet   |  |  |  | | --- | --- | --- | | **Data Field** | **Data type** | **Data** | | Acknowledge | String | “OK” or “FAIL” |   Ex) {" newUser ": "OK"} or {" newUser ":"FAIL"} |
| parkingCar | When a driver who has reservation is show up with confirm information, Web Service send information to SureParkManager. SureParkManager update reservation database, and control facilities.  - Request   |  |  | | --- | --- | | **Data Field** | **Data type** | | userID | String | | usingGarage | String | | confirmInformation | String |   Ex) {"parkingCar": {"userID":"myID", "usingGarage":"Sure-Park", "confirmInformation":"A1234"}}  - Response packet   |  |  |  | | --- | --- | --- | | **Data Field** | **Data type** | **Data** | | Acknowledge | String | “OK” or “FAIL” |   Ex) {" parkingCar ": "OK"} or {" parkingCar ":"FAIL"} |
| updateSlotStatus | When slot status is changed, Web Service need to update slot status. SureParkManger notify updateSlotStatus event to WebService.  - Notify packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | garageName | String | | garageNumber | String |   Ex) {"updateSlotStatus":{"garageName":"Sure-Park", "garageNumber":"1001"}} |
| wrongParking | When facility controller detect a driver parked wrong slot, , Web Service need to update slot status. SureParkManger notify wrongParking event to WebService.  - Notify packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | garageName | String | | garageNumber | String |   Ex) {" wrongParking ":{"garageName":"Sure-Park", "garageNumber":"1001"}} |
| detectFailure | When SureParkManager detect facilities failure, Web Service need to notice failure to attendants. SureParkManger notify detectFailure event to WebService.  - Notify packet   |  |  | | --- | --- | | **Data Field** | **Data type** | | garageName | String | | garageNumber | String |   Ex) {" detectFailure ":{"garageName":"Sure-Park", "garageNumber":"1001"}} |

## **SM-WS DB Schema**

## **SurePark Manager Detailed Design**

## **Facility Controller Detailed Design**

## **Web Service Detailed Design**

# **Future Works**

# **Lesson Learned**