Sure-Park System Project

Version 1.4

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

**History**

|  |  |  |
| --- | --- | --- |
| **Ver.** | **Date** | **History** |
| 1.0 | 9, May, 2016 | Initial |
| 1.1 | 24, May, 2016 | Change functional requirement.  : Add exit gate  : Add entry/exit gate LEDs  : Remove lane IR sensors |
| 1.2 | 26, May, 2016 | Review with customer. Change QA priority. |
| 1.3 | 31, May, 2016 | Added use case diagram  Updated system context diagram by tony comment (initial presentation).  Added alternative post condition in use case scenario.  Added business constraints by tony comment. |
| 1.4 | 13, June, 2016 | 1st decomposition & 2nd decomposition |

Contents

[**1.** **Introduction** 5](#_Toc453617507)

[**2.** **Project Context** 5](#_Toc453617508)

[**2.1) Market Context** 5](#_Toc453617509)

[**2.2) Organizational Context** 6](#_Toc453617510)

[**2.3) Business Context** 6](#_Toc453617511)

[**2.4) Technical Context** 6](#_Toc453617512)

[**3.** **Functional Requirement** 7](#_Toc453617513)

[**4.** **Use Case Scenario** 8](#_Toc453617514)

[**4.1) UC01** Reserve parking spaces 8](#_Toc453617515)

[**4.2) UC02** Show up scenario 9](#_Toc453617516)

[**4.3) UC03 ‘**No show scenario and grace period’ 9](#_Toc453617517)

[**4.4) UC04 ‘**Get out the garage and charge scenario’ 10](#_Toc453617518)

[**4.5) UC05 ‘**Parking scenario’ 11](#_Toc453617519)

[**4.3) UC06 ‘**Monitoring scenario’ 11](#_Toc453617520)

[**4.3) UC07 ‘**Management scenario’ 11](#_Toc453617521)

[**5.** **Quality Attribute** 12](#_Toc453617522)

[**6.** **Quality Attribute Scenario** 12](#_Toc453617523)

[**6.1) QA01** 12](#_Toc453617524)

[**6.2) QA02** 13](#_Toc453617525)

[**6.3) QA03** 13](#_Toc453617526)

[**6.4) QA04** 14](#_Toc453617527)

[**6.5) QA05** 14](#_Toc453617528)

[**6.6) QA06** 15](#_Toc453617529)

[**6.7) QA07** 15](#_Toc453617530)

[**6.8) QA08** 15](#_Toc453617531)

[**7.** **Quality Attribute Utility** 16](#_Toc453617532)

[**8.** **Business Constraint** 16](#_Toc453617533)

[**9.** **Technical Constraint** 17](#_Toc453617534)

[**10.** **Overall Project Schedule** 17](#_Toc453617535)

[**11.** **Project Risk and Mitigation Plan** 18](#_Toc453617536)

[**12.** **Role & Responsibility** 18](#_Toc453617537)

[**13.** **1st Decomposition** 19](#_Toc453617538)

[**14.** **2nd** **Decomposition** 21](#_Toc453617539)

[**14.1) SurePark Manager** 21](#_Toc453617540)

[**14.2) Availability of Facility Controller** 22](#_Toc453617541)

[**14.2.1) How to check if the Facility Controller is alive** 22](#_Toc453617542)

[**14.2.2) How to check the slot status** 22](#_Toc453617543)

[**15.** **Detail Design** 22](#_Toc453617544)

[**15.1) Facility Controller to SurePark Manager Packet** 22](#_Toc453617545)

[**15.1.1) Packet Structure** 22](#_Toc453617546)

[**15.1.2) Detailed Packet Scenario** 23](#_Toc453617547)

# **Introduction**

The key requisites of the project are functions that:

* drivers can reserve a parking space by using a laptop or a phone.
* parking attendants can monitor parking facilities
* the system is initially built for a small parking facility and should be able to be applied to various sized parking facilities
* the system can provide basic statistics including average occupancy, peak usage hours, parking slot statistics, and revenue, which should be extensible in order to help developers to add more analysis algorithms

****

< Figure1. System context diagram >

This document is written with the words, ‘must’, ‘must not’ , ‘required’ , ‘shall’ , ‘shall not’ , ‘should’, ‘should not’, ‘recommended’ and ‘not recommended’ in a statement. These are the following meanings that:

* ‘must’, ‘shall’, or ‘required’ means the statement is an absolute requirement.
* ‘must not’, ‘shall not’ or ‘prohibited’ means the statement is an absolute prohibition.
* ‘should’ or ‘recommended’ means the full implications must be understood before choosing a different course.
* ‘should not’ or ‘not recommended’ means the full implications must be understood before choosing this course.

# **Project Context**

## **2.1) Market Context**

|  |  |
| --- | --- |
| **Who the customer/stakeholders are** | Garage owner, GTPS, Attendant, drivers, team members, team mentor, Smart phone company, App market, Credit card company, System installer, Project Manager |
| **Notions of quality** | Reduce driver frustration, more efficiently utilize the space, reducing liabilities, reducing operating costs for owner |
| **Functional expectations** | H/W control and reservation parking space, monitoring and managing parking facility |
| **Product packaging** | H/W devices, Server, Network device, DB, user manual and S/W. |
| **How quickly you must design and deliver new products** | We have various competitors, so we need to develop parking system in five weeks for preoccupying market. |

## **2.2) Organizational Context**

|  |  |
| --- | --- |
| **Structure** | The development team has 5 members.  Namjin Lee, he is a team leader.  Jack Oh, he is a software integration engineer.  Charles Park, he is a test engineer.  Joan Kim, she is a documentation manager.  Jaeheon Kim, he is an architect.  All members are involved software development. |
| **Culture** | Our team name is “Infinite Challenge”. It means that we have an “Infinite” passion and we love “Challenges”. |

## **2.3) Business Context**

|  |  |
| --- | --- |
| **Strategies** | We will focus on a successful deployment of the initial system and then we will extend markets into the global. |
| **Internal and external providers** | H/W parts company, Server provider |
| **Cost obligations and assets** | PC, Server, Development expense |
| **Profit model** | Maintenance fee/every month, Installation fee. |
| **Competition** | Other development team. |
| **Future direction** | GTPS would like to scale out the system to include larger parking lots and garages, and sell the system to other garage owners around the world if the solution is successful for them. |

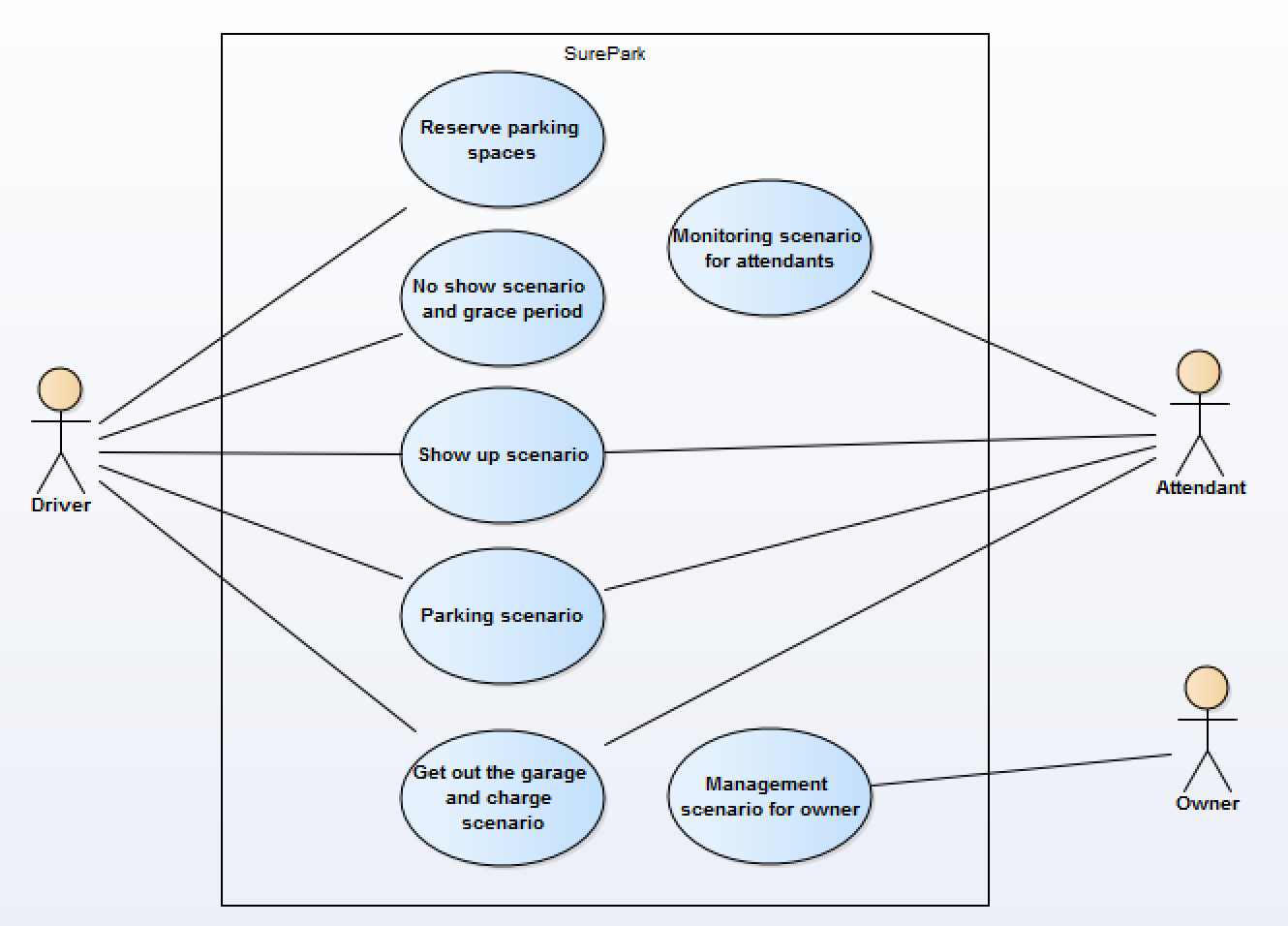
## **2.4) Technical Context**

|  |  |
| --- | --- |
| **Languages** | JAVA, C, C++ |
| **Tools** | Eclipse, Arduino IDE, JDK |
| **Operating system and hardware platform** | Arduino, Windows, Mac OSX |
| **Implementation frameworks** | Arduino |

# **Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Requirement** | **Description** |
| FR01 | The system must detect cars in parking space. | Facilities control |
| FR02 | The system must open and close the entry/exit gates. |
| FR03 | The system must change the entry/exit LED color and turn on/off stall LEDs. |
| FR04 | The system must detect presence of a car when cars arrive at the entry/exit gates. |
| FR05 | The system shall allow drivers to reserve parking spaces.  Reservations will be made via a mobile app, a laptop, or a desktop app for drivers. | Reservation system for drivers |
| FR06 | For reservation, drivers must sign up the system so that the system can prevent from unauthorized users. |
| FR07 | The system must provide available number of parking slots to drivers. |
| FR08 | Drivers must provide the day and time they would like to park, and credit card information (payment information) after logging in system. |
| FR09 | The system must return a confirmation information to the driver if reservation is succeed. |
| FR10 | The system must check the confirmation information to verify the deriver's information and reservation. | When drivers come up an entry gate |
| FR11 | “Grace period” must be configurable. | No-show process |
| FR12 | If the driver does not show up at the start of their reservation time, the system must operate the "grace period". |
| FR13 | If the driver doesn't show up within the grace period, the system must cancel the reservation. |
| FR14 | The system must calculate the total parking fee by hour and it shall charge on their credit card. | Charge system |
| FR15 | The system must show which parking spots are occupied and which are opened. | Monitoring system for attendants |
| FR16 | The system must show how long the car has occupied the particular parking space. |
| FR17 | The system must show the status when a driver parks in the wrong parking space and must automatically reassign parking spaces and correlate associated reservations. |
| FR18 | The system must show the facility usage and revenue.  The facility usage must include average occupancy, peak usage hours, parking slot statistics. | Management system for owner |
| FR19 | The system shall extend analysis algorithms or applications without disrupting operations. | Management system for owner  Extend system |
| FR20 | The system must provide login system for preventing unauthorized users. | System security |
| FR21 | The system must not allow anyone to view facility data (reservations, credit cards, etc.) except owner. |

# **Use Case Scenario**



< Figure2. Use case diagram >

## **4.1) UC01** Reserve parking spaces

|  |  |
| --- | --- |
| **ID: UC01** | **Description** |
| **Title** | (FR05 ~ FR09) Reserve parking spaces |
| **Stakeholders** | A driver who would like to reserve a parking space. |
| **Preconditions** | The driver must satisfy with FR06 |
| **Main success scenario** | 1) The Sure-Park system allows an authorized driver to reserve a parking slot (FR05).  2) The system shows available number of parking slots to the driver (FR07).  3) If a parking slot is available, the driver needs to input the day and time they would like to park, and credit card information (FR08).  4) If all information is ok, the system provides confirmation information to drivers (FR09). |
| **Post conditions** | The reservation was confirmed. |
| **Alternate scenario** | 3a) No available parking spaces  1) The system closes the reservation.  4a) Some information is invalid  1) The system displays which information is failed.  2) Repeat steps 3-4 until all information is valid. |
| **Alternative post conditions** | 3a) Reservation failed.  4a) Reservation failed. |

## **4.2) UC02** Show up scenario

|  |  |
| --- | --- |
| **ID: UC02** | **Description** |
| **Title** | (FR02-FR04,FR10) Show up scenario |
| **Stakeholders** | A driver who has made a reservation, An attendant who confirms the reservation |
| **Preconditions** | UC01, The entry gate LED is red. |
| **Main success scenario** | 1. A driver comes to the gate. 2. The system detects the presence of a car at the gate (FR04). 3. A driver provides confirmation information to attendant. 4. The attendant inputs confirmation information to the system. 5. The system verifies the driver’s information and confirms the reservation (FR10). 6. The system lifts the entry gate and allows the driver to enter the facility (FR02). 7. The system changes the entry gate LED from red to green (FR03). 8. After the driver passed the gate, the system close the entry gate (FR02) and the entry LED turns red (FR03). |
| **Post conditions** | The car entered into the garage. |
| **Alternate scenario** | 5a) Invalid confirmation information,   1. The system does not allow the driver to enter the garage. |
| **Alternative post conditions** | 5a) A driver cannot access the garage. |

## **4.3) UC03 ‘**No show scenario and grace period’

|  |  |
| --- | --- |
| **ID: UC03** | **Description** |
| **Title** | (FR11-FR13) No show scenario and grace period |
| **Stakeholders** | A driver who has made a reservation |
| **Preconditions** | UC01, “Grace period” has been configured (FR11). |
| **Main success scenario** | 1) If the driver does not show up at the start of their reservation time, the parking spot will be held for a “grace period” after the start of the reservation (FR12).  2) If the driver doesn’t show up within the grace period, the parking spot is released. Drivers are not charged for “no-show”, but they lose their reservation (FR13). |
| **Post conditions** | The reservation has canceled. |
| **Alternate scenario** | 1a) If the driver shows up at the garage  1) process UC02 scenario  2a) If the driver shows up within the grace period,  1) process UC02 scenario |
| **Alternative post conditions** | 1a) UC02  2a) UC02 |

## **4.4) UC04 ‘**Get out the garage and charge scenario’

|  |  |
| --- | --- |
| **ID: UC04** | **Description** |
| **Title** | (FR01 ~ FR04, FR14) Get out the garage and charge scenario |
| **Stakeholders** | A driver who has made a reservation. Attendants who check reservation. |
| **Preconditions** | UC02, The exit gate LED is red (FR03). |
| **Main success scenario** | 1. A driver leaves the parking slot (FR01). 2. The system saves the time of departure from the parking slot. 3. The driver drives toward the exit gate. 4. The system detects the car at the exit gate (FR04). 5. The system opens the exit gate to allow the driver to leave the facility (FR02). 6. The system changes the exit gate LED from red to green (FR03). 7. The driver leaves the parking garage. 8. The system closes the exit gate (FR02) and the exit gate LED turns red (FR03). 9. The system calculates the parking fee by the hour (time of entry, to time of departure from the parking slot, time > 30 min : full charge, time <= 30min : half charge) and charges it automatically on driver’s credit card (FR14). 10. The system updates the parking status, makes the parking slot free. |
| **Post conditions** | The parking fee has charged and parking status has updated. |
| **Alternate scenario** | None. |
| **Alternative post conditions** | None |

## **4.5) UC05 ‘**Parking scenario’

|  |  |
| --- | --- |
| **ID: UC05** | **Description** |
| **Title** | (FR01, FR03, FR17) Parking scenario |
| **Stakeholders** | Attendant, A driver who is parking his/her car. |
| **Preconditions** | UC02 |
| **Main success scenario** | 1) The system is illuminating the stall LED at assigned parking space (FR03).  2) A car parked at the designated parking spot (FR01).  3) The stall LED at the parking space will be turned off (FR03). |
| **Post conditions** | The car is parked correctly. |
| **Alternate scenario** | 2a) If a car parked at other spot,   1. The system will automatically reassign parking spaces and correlate associated reservations (FR17). |
| **Alternative post conditions** | 2a) Reallocation of other space. |

## **4.3) UC06 ‘**Monitoring scenario’

|  |  |
| --- | --- |
| **ID: UC06** | **Description** |
| **Title** | (FR15, FR16) Monitoring scenario for attendants. |
| **Stakeholders** | Attendants |
| **Preconditions** | Attendants must log in the system. |
| **Main success scenario** | 1) Attendants select the monitoring menu.  2) The system shows which parking spots are open and occupied (FR15). Also, it will show how long a car has occupied a particular parking spot (FR16). |
| **Post conditions** | Display a parking status. |
| **Alternate scenario** | None. |
| **Alternative post conditions** | None |

## **4.3) UC07 ‘**Management scenario’

|  |  |
| --- | --- |
| **ID: UC07** | **Description** |
| **Title** | (FR18) Management scenario for owner |
| **Stakeholders** | Owner |
| **Preconditions** | Owner must log in the system. |
| **Main success scenario** | 1) Owner selects the management menu which shows parking statistics and revenue.  2) The system shows which basic statistics on facility usage to include average occupancy, peak usage hours, parking slot statistics (e g. how much time cars were parked in parking slots) and revenue (FR18). |
| **Post conditions** | Display a basic statistics and revenue. |
| **Alternate scenario** | None. |
| **Alternative post conditions** | None. |

# **Quality Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Quality Attribute** | **Priority** | **Description** |
| QA01 | Scalability | MID | Installers should complete setup and tests for a new facility controller in an hour. |
| QA02 | Availability | HIGH | Sure park system’s software detects software failure. In this case, Sure Park system’s software notify attendants in 30 seconds and restart in 1mins. |
| QA03 | Security | HIGH | Users log in the system and get the permission to access the authorized data and information. The unauthorized user tries to access the data and information which are permitted only attendants and owner. The system prevents all unauthorized access. |
| QA04 | Extensibility | MID | Developer wants to add new algorithm application to Sure Park software. The system needs to be updated without disrupting operations. New algorithm can be implemented and tested within 1 week. |
| QA05 | Performance | MID | When driver wants to get an empty parking slot, system must provide it in 5 sec. |
| QA06 | Usability | HIGH | The owner wants to check basic statistics on facility usages. The owner can show statistic report in 3 step after login. |
| QA07 | Interoperability | MID | When a driver enters and goes out the parking garage, facility controller and Sure park system must communicate without communication loss. |
| QA08 | Modifiability | HIGH | The developers want to design scale up/out the system. The new system is implemented and tested in a week. |

# **Quality Attribute Scenario**

## **6.1) QA01**

|  |  |
| --- | --- |
| **Title** | Scale out to other parking facilities |
| **ID** | QA01 |
| **Quality Attribute** | Scalability |
| **Scenario** | Installers should complete setup for a new facility controller in an hour. |
| **Source of stimulus** | Installer |
| **Stimulus** | Installer wants a new facility controller in the parking garage. |
| **Artifact** | The system |
| **Environment** | New owner wants to install a new system or existing owner wants to extend the current system. |
| **Response** | New facility controller will be installed completely |
| **Response measure** | 1 hour for installing |

## **6.2) QA02**

|  |  |
| --- | --- |
| **Title** | Detect malfunction of the Sure park system’s software |
| **ID** | QA02 |
| **Quality Attribute** | Availability |
| **Scenario** | Sure park system’s software detects software failure. In this case, Sure Park system’s software notify attendants in 30 seconds and restart in 1mins. |
| **Source of stimulus** | Sure park system’s software |
| **Stimulus** | Malfunction of Sure park system |
| **Artifact** | Sure Park system’s software |
| **Environment** | During normal operation |
| **Response** | The system software fault is detected, the system logs the fault and notifies attendants. |
| **Response measure** | The system should notify attendants of the fault in 30 seconds. |

## **6.3) QA03**

|  |  |
| --- | --- |
| **Title** | Protect data and information from unauthorized access |
| **ID** | QA03 |
| **Quality Attribute** | Security |
| **Scenario** | Users log in the system and get the permission to access the authorized data and information. The unauthorized user tries to access the data and information which are permitted only attendants and owner. The system prevents all unauthorized access. |
| **Source of stimulus** | Unauthorized user, unauthorized system |
| **Stimulus** | Unauthorized attempts to display data and access system service. |
| **Artifact** | The system |
| **Environment** | Normal operation (run time) |
| **Response** | The data and information are protected from unauthorized access. |
| **Response measure** | How many unauthorized accesses are protected? 100% |

## **6.4) QA04**

|  |  |
| --- | --- |
| **Title** | Add more analysis algorithms or analysis applications |
| **ID** | QA04 |
| **Quality Attribute** | Extensibility |
| **Scenario** | Developer wants to add new algorithm application to Sure Park software. The system needs to be updated without disrupting operations. New algorithm can be implemented and tested within 1 week. |
| **Source of stimulus** | Developers |
| **Stimulus** | Add new algorithm to the system |
| **Artifact** | Sure Park system’s software |
| **Environment** | Normal operation (run time) |
| **Response** | New algorithm should be added without disrupting operations. |
| **Response measure** | New algorithm can be implemented and tested within 1 week. |

## **6.5) QA05**

|  |  |
| --- | --- |
| **Title** | Retrieve an available parking slot ASAP. |
| **ID** | QA05 |
| **Quality Attribute** | Performance |
| **Scenario** | When driver wants to get an empty parking slot, system must provide it in 5 sec. |
| **Source of stimulus** | Driver |
| **Stimulus** | Request reservation. |
| **Artifact** | Sure Park system |
| **Environment** | Normal operation (run time) |
| **Response** | Retrieve about parking slot status. |
| **Response measure** | The system must return the parking slot status in 5 sec. |

## **6.6) QA06**

|  |  |
| --- | --- |
| **Title** | Obtain basic statistics on facility usage. |
| **ID** | QA06 |
| **Quality Attribute** | Usability |
| **Scenario** | The owner wants to check basic statistics on facility usages. The owner can show statistic report in 3 step after login. |
| **Source of stimulus** | The owner |
| **Stimulus** | Check statistics on facility usage. |
| **Artifact** | Sure Park system |
| **Environment** | Normal operation (Run time) |
| **Response** | Display basic statistics |
| **Response measure** | Statistics report can be show in 3 steps after the owner log in. |

## **6.7) QA07**

|  |  |
| --- | --- |
| **Title** | Communicate between facility controller and Sure Park system. |
| **ID** | QA07 |
| **Quality Attribute** | Interoperability |
| **Scenario** | When a driver enters and goes out the parking garage, facility controller and Sure park system must communicate without communication loss. |
| **Source of stimulus** | Facility controller and Sure park system |
| **Stimulus** | Exchange updated status or commands. |
| **Artifact** | Facility controller and Sure park system |
| **Environment** | Normal operation (run time) |
| **Response** | Communication success. |
| **Response measure** | Communication success rate : 100%(100 times communication try and 100 times success.) |

## **6.8) QA08**

|  |  |
| --- | --- |
| **Title** | Scale up/out to parking facilities |
| **ID** | QA08 |
| **Quality Attribute** | Modifiability |
| **Scenario** | The developers want to design scale up/out the system. The new system is implemented and tested in a week. |
| **Source of stimulus** | Developers |
| **Stimulus** | The developers who would like scale up/out. |
| **Artifact** | The source code |
| **Environment** | Design process |
| **Response** | Complete the implementation and testing. |
| **Response measure** | Complete the implementation and testing in a week. |

# **Quality Attribute Utility**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Quality Attribute** | **Description** | **Difficulty**  **(D)** | **Priority**  **(P)** | **Total Score**  **(D x P)** |
| QA01 | Scalability | Scale out to other parking facilities | 3 | 3 | 9 |
| QA02 | Availability | Detect malfunction of the Sure park system’s software | 9 | 9 | 81 |
| QA03 | Security | Protect data and information from unauthorized access | 3 | 9 | 27 |
| QA04 | Extensibility | Add more analysis algorithms or analysis applications | 3 | 3 | 9 |
| QA05 | Performance | Retrieve an available parking slot ASAP. | 1 | 3 | 3 |
| QA06 | Usability | Obtain basic statistics on facility usage. | 1 | 9 | 9 |
| QA07 | Interoperability | Communicate between facility controller and Sure Park system. | 3 | 3 | 9 |
| QA08 | Modifiability | Scale up/out to parking facilities | 9 | 9 | 81 |

\* highest value is the highest.

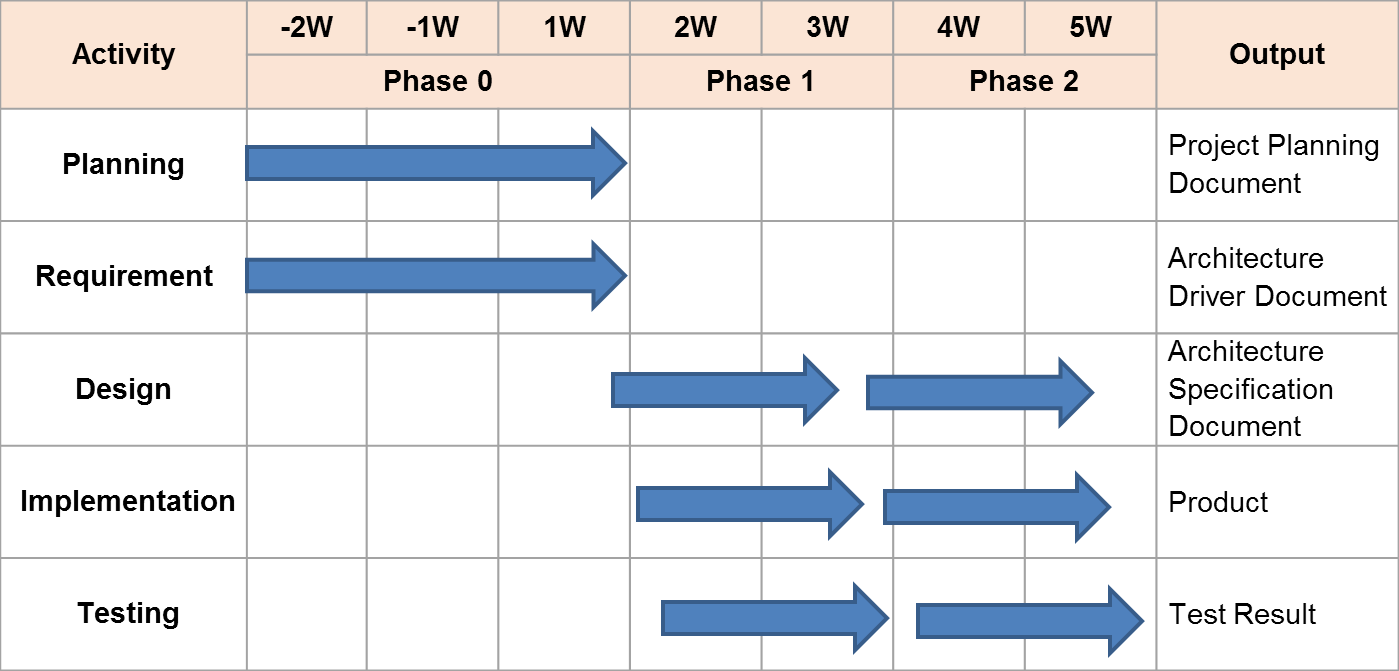
# **Business Constraint**

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Constraint** | **Description** |
| BC01 | Reducing complain | GTPS wants to reduce driver frustration when customers find available parking slots and reserve them. |
| BC02 | Increasing profits | More efficient space utilization is needed. |
| BC03 | Reducing liabilities | It is needed to reduce traffic congestion and the chance for accidents inside the parking facilities. |
| BC04 | Reducing operating costs | It is required to utilize personnel efficiently and reduce the number of employee. |
| BC05 | Applying other garage | GTPS would like to market the system to other garage owners around the world. |
| BC06 | Delivery | The system should be delivered in 5 weeks. |
| BC07 | Availability of workforce | The team is consists of 5 members. Java expert is only 1 person. |
| BC08 | Access the garage | Only a car can get in/out the garage. |
| BC09 | Parking | A driver who made a reservation can park a car. |
| BC10 | Charge | The system will charge a check by 30 minutes. |
| BC11 | Reservation | A driver can make a reservation within 3 hours. |

# **Technical Constraint**

|  |  |  |
| --- | --- | --- |
| **ID** | **Technical Constraint** | **Description** |
| TC01 | H/W System | Wi-Fi enabled Arduino(mega 2560)  - Flash Memory: 256KB of which 8KB used by bootloader  - SRAM: 8KB  - EEPROM: 4KB  - Clock Speed: 16MHz |
| TC02 | Programming language | For development Arduino: C  For server and application: Java |
| TC03 | Network | Wi-Fi  Wi-Fi configuration |
| TC04 | Facility parts | Scalable parts and Arduino are the same. |

# **Overall Project Schedule**



Phase 0:

Output: Project planning document,

Architecture Driver document,

Phase 1:

Output: Architecture design document(draft)

Arduino design & prototype implementation

Controller design & prototype implementation

UI design & prototype implementation

Protocol design

DB design

Unit test result

Time log

Phase 2:

Output: Architecture design document(final)

Sure park system (final product)

Integration Test result

Time log

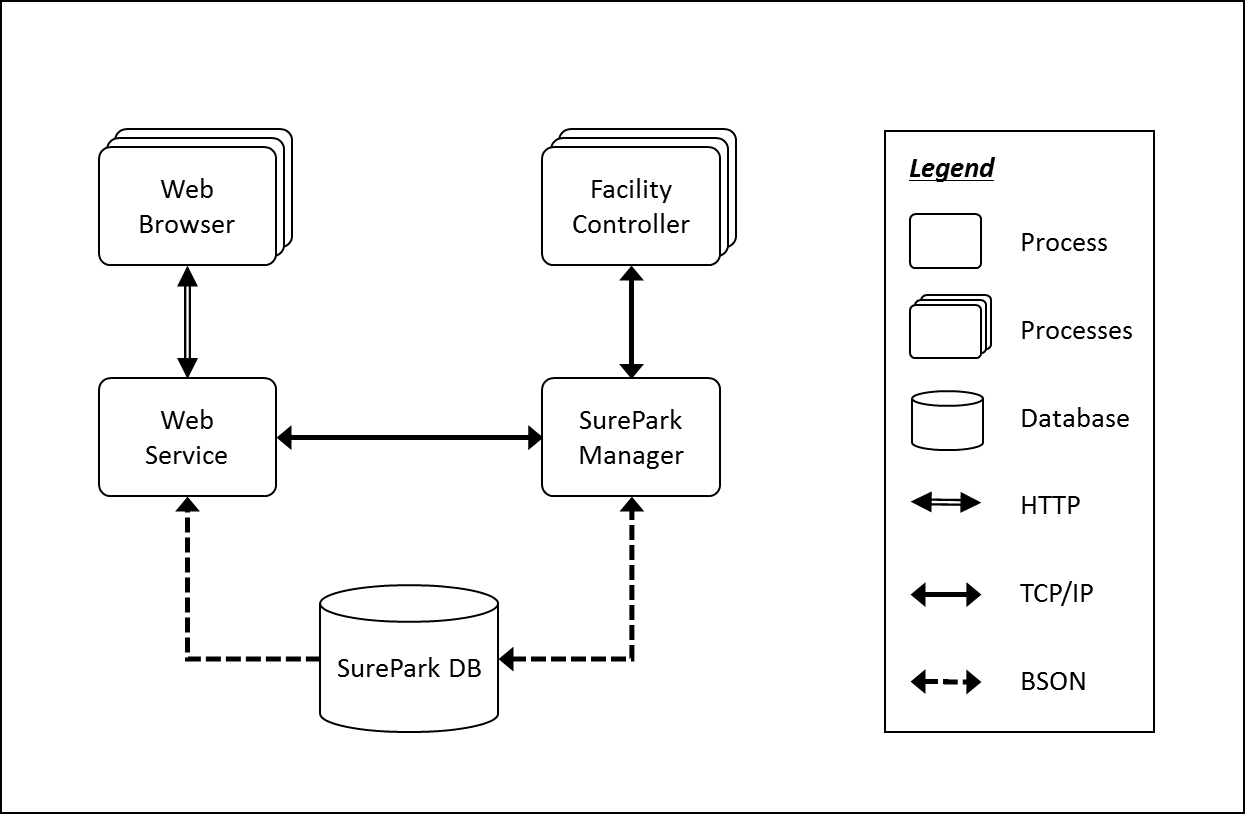
# **Project Risk and Mitigation Plan**

|  |  |  |
| --- | --- | --- |
| RISK | Priority | Mitigation Plan |
| Low experience of JAVA development | Low | We will be familiar with JAVA before arrived at CMU. |
| No experience of Arduino development | Low | We will be familiar with Arduino before arrived at CMU. |
| Not familiar with architectural patterns | High | We discuss various architectural patterns with mentor. |
| Short term for development | High | We will make a plan well and manage it perfectly. |
| Difficult to test big scaled system | Mid | We will design architecture considering testability. |

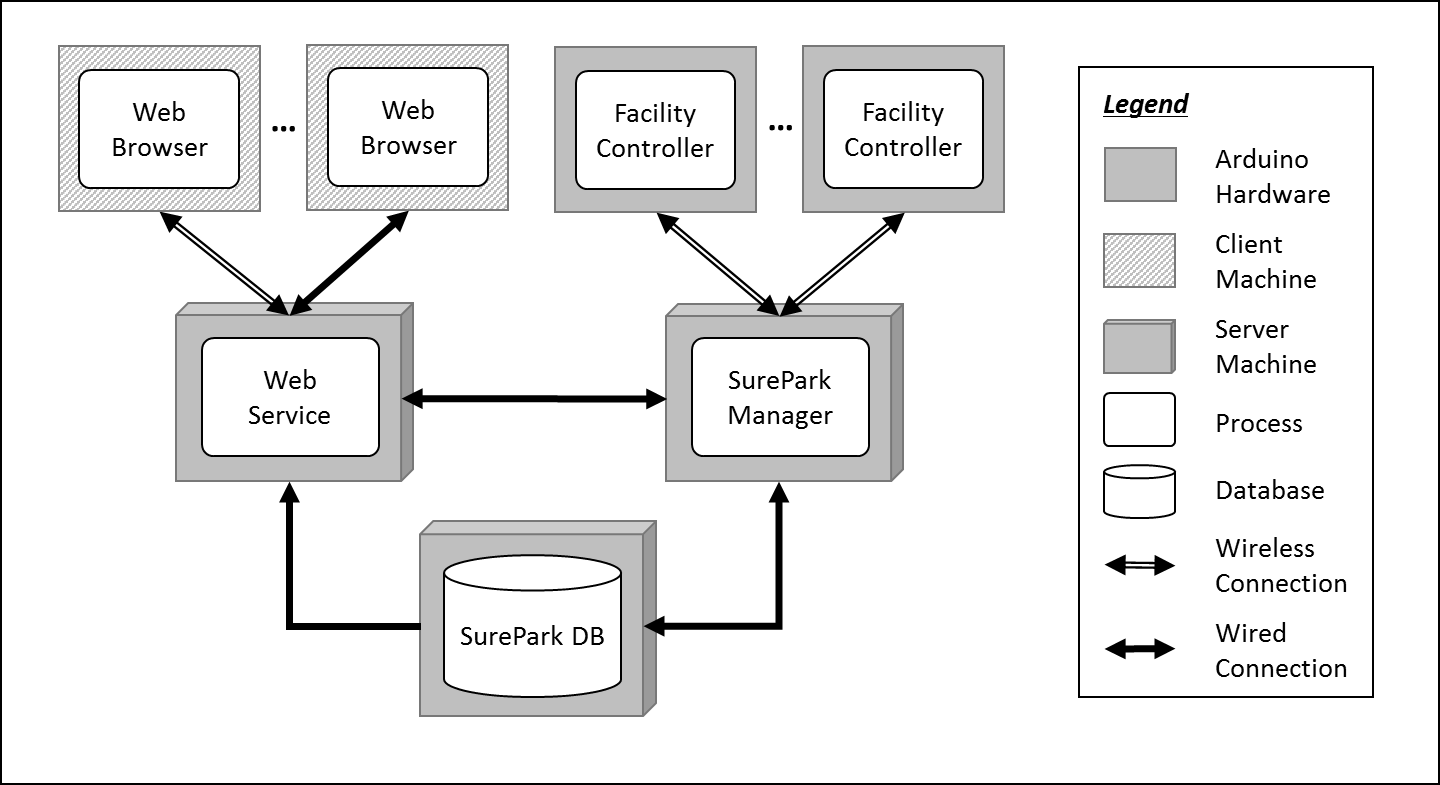
# **Role & Responsibility**

|  |  |  |
| --- | --- | --- |
| Role | Assign | Responsibility |
| Team leader | Namjin Lee | Check time log and risk management |
| Architect | Jaeheon Kim | Design system architecture |
| Integration | Jack Oh | Integrate all artifacts(source code, documents …etc). |
| Test | Charles Park | Test and delivery. |
| Documentation | Joan Kim | Create document artifacts. |
| Development | All | We all develop the parking system. |

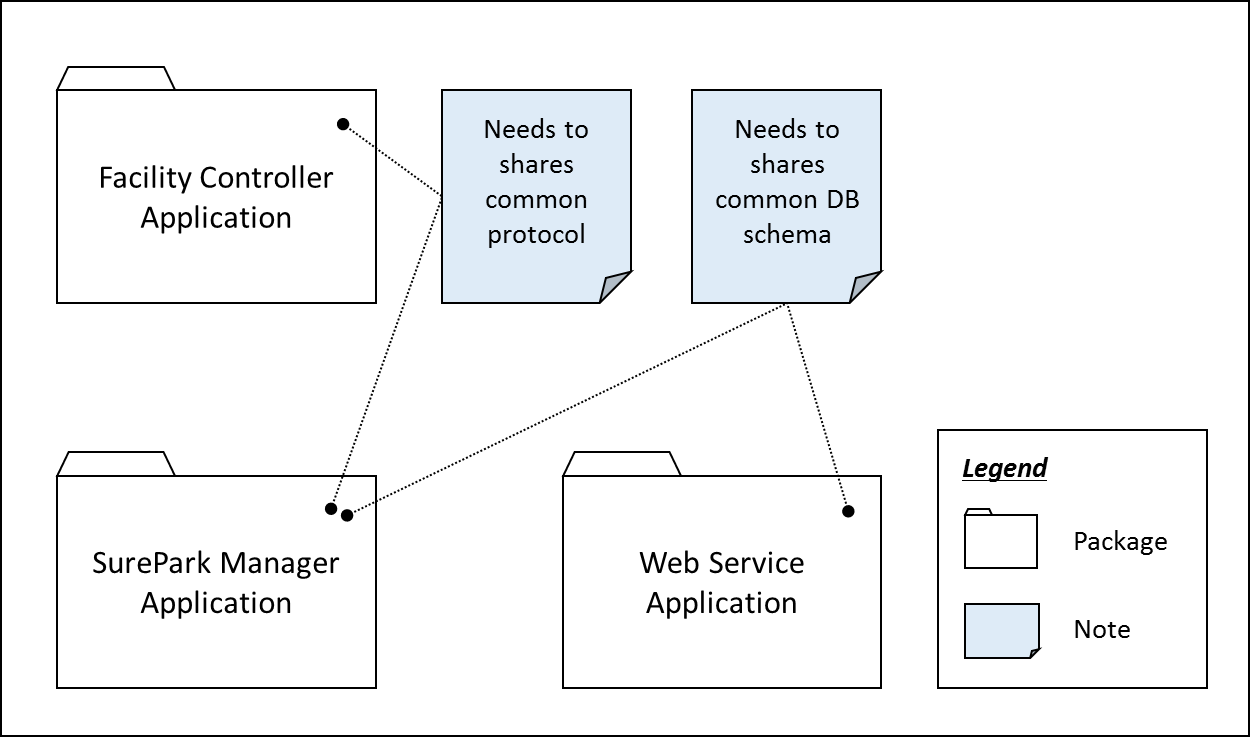
# **1st Decomposition**



< Figure3. Dynamic view of 1st decomposition >



< Figure4. Allocation view of 1st decomposition>



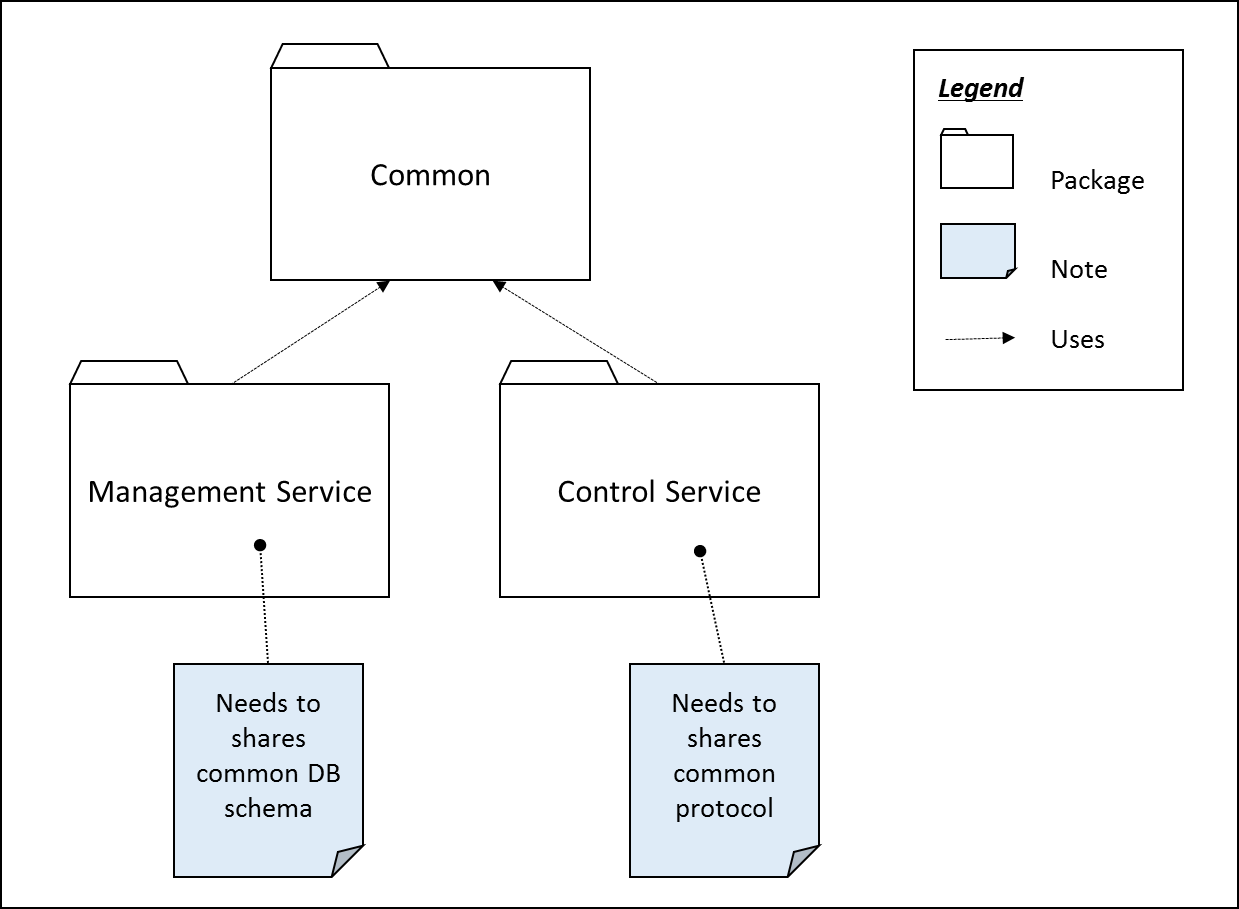
< Figure5. Static view of 1st decomposition >

|  |  |
| --- | --- |
| **Modifiability(QA08)** | **Perspective: Dynamic** |
| Architectural Pattern | Client-Server pattern with Facility Controller and SurePark Manager.  Repository pattern with SurePark DB |
| 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.    < Figure6. Architectural patterns of SurePark System > |

|  |  |
| --- | --- |
| **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. |

# **2nd** **Decomposition**

## **14.1) SurePark Manager**



< Figure7. Static view of SurePark Manager >

|  |  |
| --- | --- |
| **Entity** | **Description** |
| common | Common package consists of configuration and message between management service and control service. |
| Management Service | Management service manages DB. |
| Control Service | Control Service controls to communicate with the facility controller. |

## **14.2) Availability of Facility Controller**

### **14.2.1) How to check if the Facility Controller is alive**

|  |  |
| --- | --- |
| **Availability(QA02)** | **Perspective: Dynamic** |
| Architectural Pattern | Client-Server structure with heartbeat tactic. |
| Rationale | Facility Controller send a packet every 5 seconds to Controller Service. If Controller Service doesn’t get this packet until 15 seconds, the Controller Service notify it to Manager Service for alarming to attendant. |

### **14.2.2) How to check the slot status**

|  |  |
| --- | --- |
| **Availability(QA02)** | **Perspective: Static** |
| Architectural Pattern | Client-Server structure through heartbeat packet |
| Rationale | The Facility Controller can check whether a slot’s IR sensor is broken or not through sensitivity value. So that the Facility Controller send a heartbeat packet with all stall status.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | $ | Facility Id | S | Slot 0 | Slot 1 | ... | Slot N | \n |   \* value 0 means a slot is opened.  \* value 1 means a slot is occupied.  \* value 2 means a slot is broken.  Ex1) $0001S1001\n (a Slot 0 and a slot 3 are occupied.)  Ex2) $0001S0020\n (a Slot 2 is broken.) |

# **Detail Design**

## **15.1) Facility Controller to SurePark Manager Packet**

### **15.1.1) Packet Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Start Symbol** | **Facility Id** | **Code** | **Value** | **End Symbol** |
| 1byte($) | 4byte | 1byte | Variable length | 1byte(\n) |

Start Symbol: Start point of valid packet.

Arduino Id: Assigned the Arduino Id.

Code: Indicate what kind of packet is. I means “Information”.

S means “Slot Status”. G means “Entry Gate”. L means “LED”.

Value: It depends on “Code”. Please refer to “detailed packet scenario”.

End Symbol: End point of valid packet.

### **15.1.2) Detailed Packet Scenario**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Send Information | After connection, the SurePark Manager has to send information to the Facility Controller.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | $ | Facility Id | I | Slot No. | \n |   Ex) $0001I4\n (Garage 1 consists of 4 stalls.) |
| Send Slot Status | Basically, the Facility Controller has to send the slot status to a SurePark every 3 seconds. And if slot status is changed, the Facility Controller send it again regardless of under 3 minutes.  The value 0 means a slot is opened, 1 means a slot is occupied. 2 means a slot is broken.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | $ | Facility Id | S | Slot 0 | Slot 1 | ... | Slot N | \n |   Ex1) $0001S1001\n (Slot 0 and slot 3 are occupied.)  Ex2) $0001S0000\n (All slots are opened.) |
| Open Entry Gate | |  |  |  |  |  | | --- | --- | --- | --- | --- | | $ | Facility Id | G | 1 | \n |   Ex1) $0001G1\n (request to open the entry gate.) |
| Turn on Slot LED | |  |  |  |  |  | | --- | --- | --- | --- | --- | | $ | Facility Id | L | Slot No. | \n |   Ex1) $0001L1\n (Slot 1's LED has to get "Green".)  Ex2) $0001L0\n (Slot 0's LED has to get "Green".) |