|  |
| --- |
| LGE – CMU SW Architect course 2016 |
| Architecture Drivers Specification |
| TEAM 2 – NOT YET |
|  |
| **Wonhong Kwon (wonhong.kwon@lge.com)** |
| **2016-06-20** |

|  |
| --- |
| This document describes architecture driver specification of the studio project called Sure-Park System. Architecture driver specification includes function requirements, constraints, and quality attributes. |

Table of Contents

[0. Document Revision 1](#_Toc453289194)

[1. High-level Functional Requirements 2](#_Toc453289195)

[**Make reservation** 3](#_Toc453289196)

[**Park a car** 3](#_Toc453289197)

[**Leave a facility** 3](#_Toc453289198)

[**For attendant** 3](#_Toc453289199)

[**For owner** 3](#_Toc453289200)

[2. Constraints 4](#_Toc453289201)

[**Technical Constraints** 4](#_Toc453289202)

[**Business Constraints** 5](#_Toc453289203)

[3. Quality Attributes 5](#_Toc453289204)

[**List of QAs** 5](#_Toc453289205)

[**Quality Attribute Scenario** 6](#_Toc453289206)

[4. Reference 6](#_Toc453289207)

0. Document Revision

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Author |
| 0.1 | 2016.06.07 | Initial version | Wonhong Kwon |
| 0.2 | 2016.06.09 | Refine QAs | Wonhong Kwon |
| 1.0 | 2016.06.13 | Version 1.0 release | Wonhong Kwon |
| 1.1 | 2016.06.15 | Refine availability | Wonhong Kwon |
| 1.2 | 2016.06.18 | Refine technical constraint | Wonhong Kwon |
| 1.3 | 2016.06.20 | Add QA priority and refine some QAS | Wonhong Kwon |

1. High-level Functional Requirements

This section describes which functional requirements can affect the architecture design of our system. Some of requirements derived from Project Description [1] and discussion with customer are not described here because of their low degree of impact on the architecture.

## **Make reservation**

**[FR-001]** System shall provide an application (it can be mobile, laptop, or desktop app.) to allow a driver to make reservation of a parking slot.

**[FR-002]** System shall provide the availability of parking spaces in a garage to a driver.

**[FR-003]** System shall ask a driver about identifying information, the day and time, the payment information and reserve a parking slot using this information.

**[FR-004]** System shall provide confirmation information to a driver when reservation is done.

## **Park a car**

**[FR-011]** System shall ask a driver about confirmation information to identify the driver.

**[FR-012]** System shall verify confirmation information and confirm driver’s reservation.

**[FR-013]** System shall hold the reservation for a “grace-period” even if a driver doesn’t show up at the start of his/her reservation time.

**[FR-014]** System shall release the reservation if a driver doesn’t show up within the “grace-period”. – This is called “no-show”.

## **Leave a facility**

**[FR-021]** System shall charge on driver's credit card for the duration of stay (half an hour basis).

## **For attendant**

**[FR-031]** System shall provide an application (it can be mobile, laptop, or desktop app.) to allow an attendant to check the system’s status.

**[FR-032]** System shall show which parking spots are open and which are occupied.

**[FR-033]** System shall show how long a car has occupied a particular spot.

**[FR-034]** System shall show when a driver parks in the wrong parking space and automatically reassign parking spaces and correlate associated reservations.

## **For owner**

**[FR-041]** System shall provide an application (it can be mobile, laptop, or desktop app.) to allow an owner to check the system’s statistics.

**[FR-042]** System shall have the following statistics.

* Average occupancy
* Peak usage hours
* Parking slot statistics (how much time cars were parked in parking slots)
* Revenue

**[FR-043]** System shall configure “grace-period” and parking fee as owner’s preference.

**[FR-044]** System shall support that parking fee can be changed daily basis. And, even if the fee is changed, driver which is already entered into the parking facility will be charged with the fee when he/she entered.

2. Constraints

## **Technical Constraints**

**Use dedicated HW type**

Use Arduino 2560 and dedicated type of sensors, indicators, and gate actuators since HW for the facility has already been established and customer doesn’t want to change them.

**No changes on underlying HW**

Customers won’t change underlying HW even if they would like to market the system to other garage owners around the world.

## **Business Constraints**

**Schedule & Resource**

We just have 5 weeks to finish the project, but we only have 6 developers.

**Low Cost**

Reduce the # of people required to operate the facilities reducing operation cost

**Facility Usage Limitation**

Only a car can be moving in the facility at a time

**No cash**

Only credit card is supported as payment method

**Reservation rule**

Driver can reserve a slot 3 hours in advance only. Driver cannot enter into the facility w/o reservation. Driver can only reserve a slot on online. (No offline reservation)

3. Quality Attributes

This section describes a bunch of quality attributes derived from the discussion with stakeholders and defines quality attribute scenario to indicate how the system reacts to satisfy those quality attributes.

## **List of QAs and their priority (Importance, Difficulty)**

Scalability (M, M): Easy to install new parking slots to an existing parking facility or set-up a new parking facility

Security (H, M): Prevent unauthorized users from accessing the information related to operating facility

Availability (H, H): Allow drivers to reserve parking spaces and use parking facilities even if the system or the facility has some problems

Configurability (M, L): Easy to change the parking fee or the grace-period

Extensibility (L, M): Add new analysis algorithm to the system without disrupting operations

## **Quality Attribute Scenario**

**Easy to install new parking slots to an existing parking facility or set-up a new parking facility**

A new facility controller is introduced; the system is able to recognize this new HW without having to re-start the system or changing existing processing components.

**Prevent unauthorized users from accessing the information related to operating facility**

When a malicious attacker who doesn’t have any authorization tries to access the information, the system doesn’t allow it and notifies it to an operator within 30 seconds when the system detects an attack.

**Allow drivers to reserve parking spaces and use parking facilities even if the system or the facility has some problems**

Power suddenly went out in a facility; the system is able to detect the power outage and calls an attendant of the facility in a minute to take care of the facility manually. Also, the system is able to call local power supplier to fix the facility’s power outage. During the repair of the facility, the system doesn’t allow any reservation on the facility.

A facility controller is out of order; the system is able to detect it and call an attendant of the facility in a minute to take care of the facility manually and fix/replace the controller to run the facility again. During the repair of the facility, the system doesn’t allow any reservation on the facility.

The system is down when driver is making reservation of a parking slot; the back-up system is able to detect this fault and works as usual without having any person’s intervention in a minute.

**Easy to change the parking fee or the grace-period**

When an owner wants to change the parking fee (an hour basis), the system is able to change it without having to re-compile applications or re-start the system.

**Add new analysis algorithm to the system without disrupting operations**

When an owner wants to see new statistic, a developer is able to add new analysis algorithm to the system without having re-compile a statistic application or re-start the system.

4. Reference

[1] Sure-Park: Parking Garage Management System Project Description – V2