Final Presentation

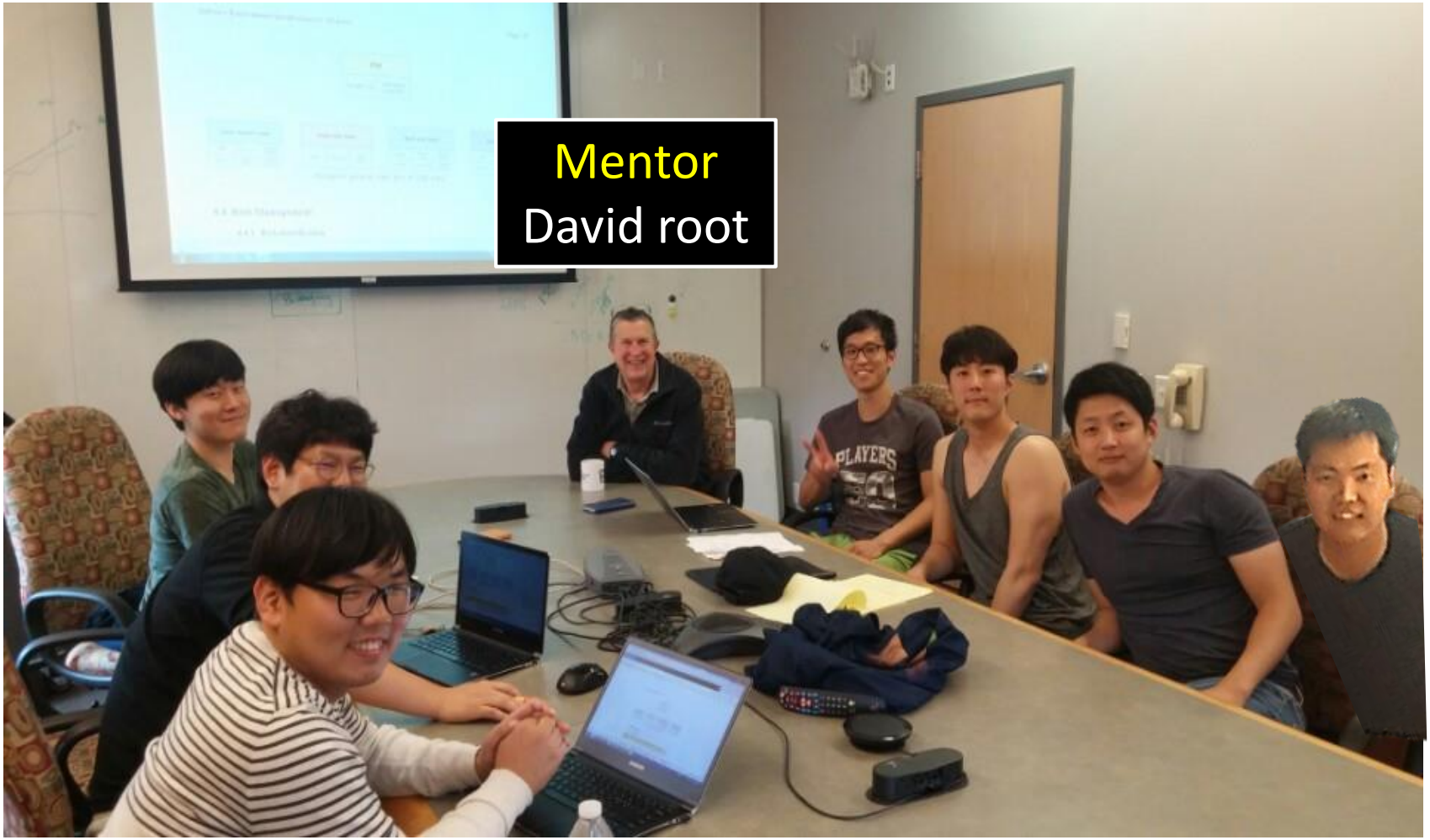
Sure-Park System

< Team C >
Dongho Lee (Leader), Kyeongseok Yang, Woojin Han, Myoungki Hong,
Sunghoon Byun, Daesoon Kim, Dongjae Kim

Contents

Contents

Introduction of team c



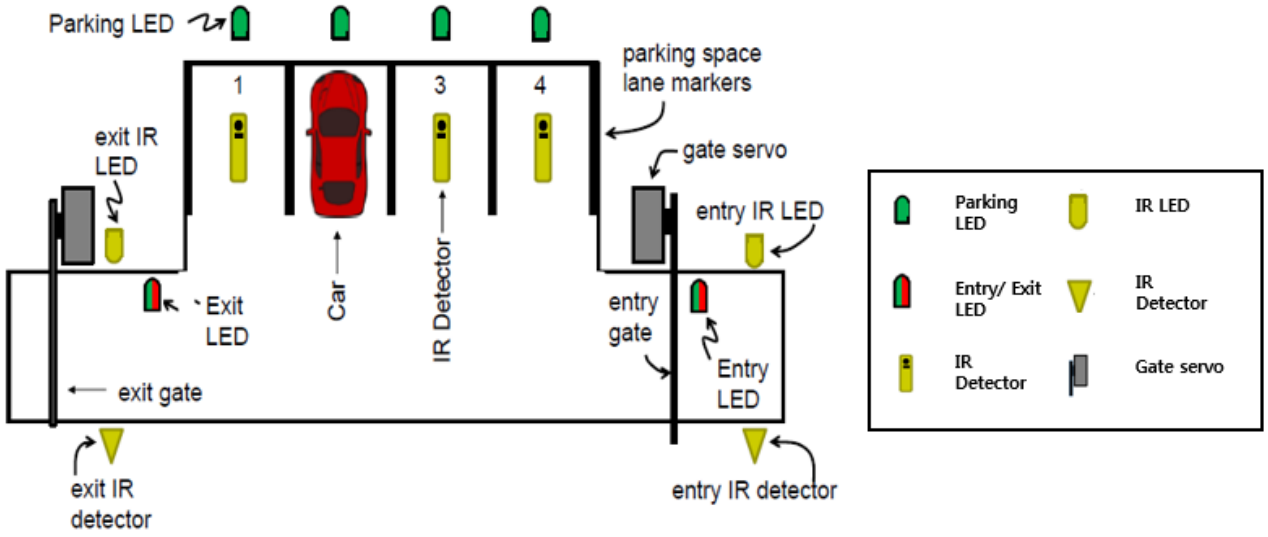
Dongho Lee (Leader), Kyeongseok Yang, Woojin Han, Myoungki Hong, Sunghoon Byun, Daesoon Kim, Dongjae Kim

Contents

Project Overview

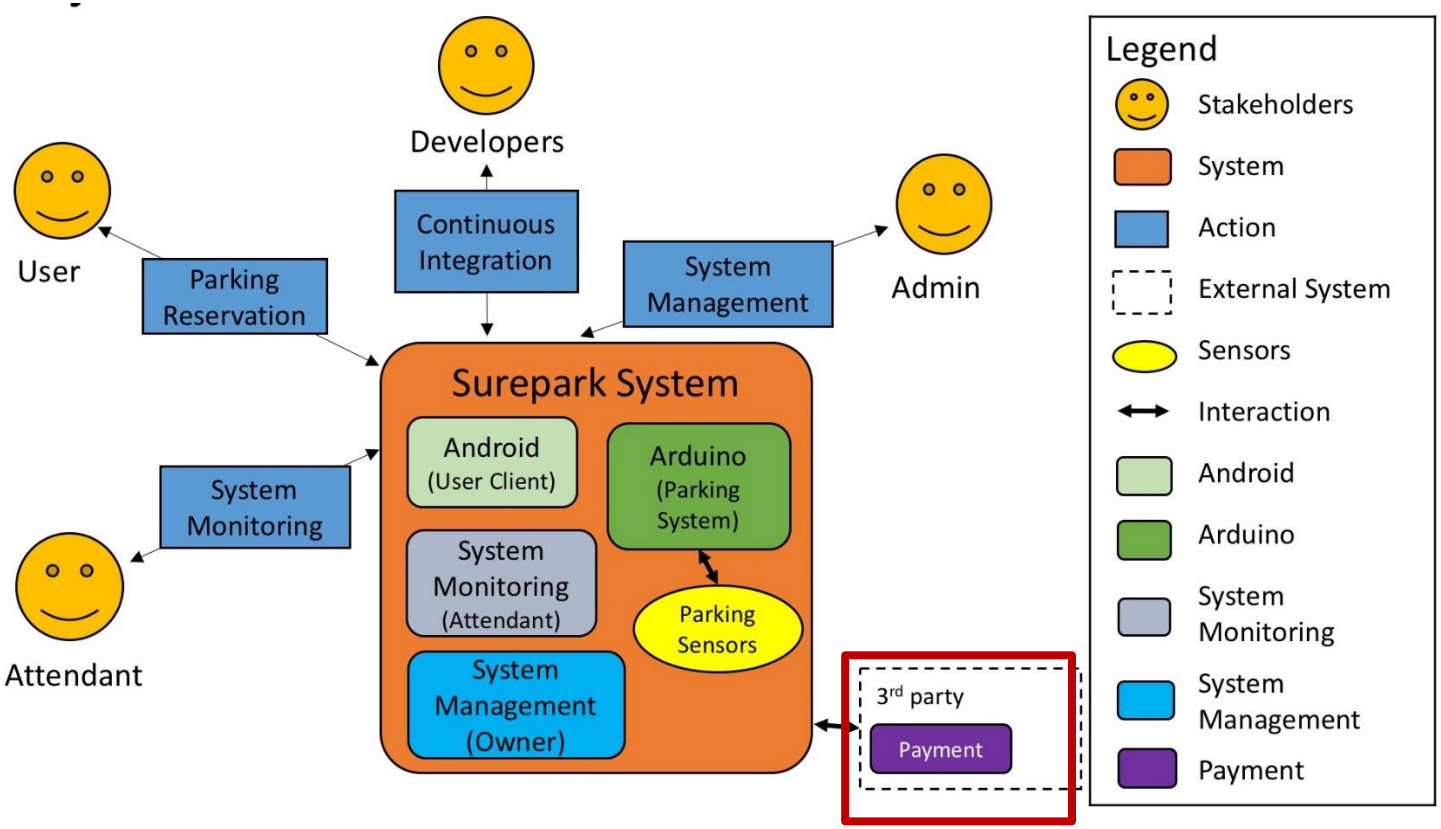
- Problems
  - Drivers don’t know where a vacant slot is
  - They tend to wander on the facility to find a vacant slot.
  - It needs many employees to manage parking facilities.

Waste of time and manpower!!



Context Diagram

- Android app for reservation & check & cancel
- Arduino for a smart parking facility
- Monitoring & management system for attendant and owner



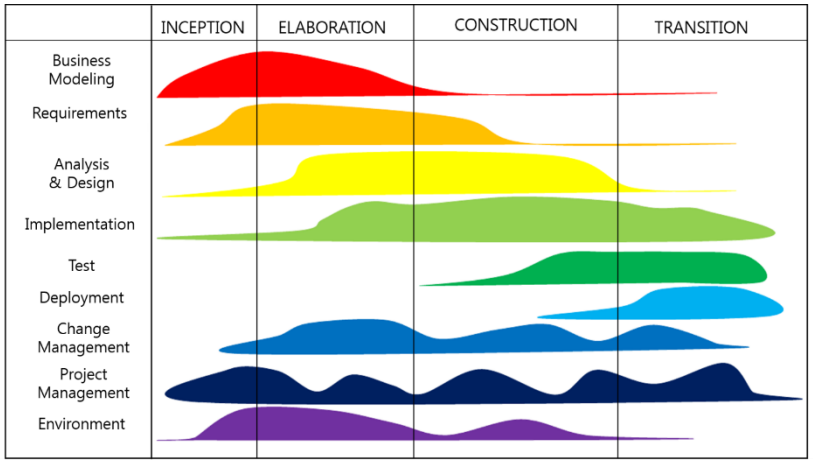
Contents

Role Assignment

Roles		
Member ID	Name	ACDM Role
DH	Dongho Lee	Managing engineer
DS	Daesoon Kim	Quality process engineer
MK	Myoungki Hong	Production engineers
SH	Sunghoon Byun	Requirements engineer
KS	Kyeongseok Yang	Production engineers
WJ	Woojin Han	Chief architect
DJ	Dongjae Kim	Production engineers

Planning

- Development process
  - Based on openUP



Tailored openUP process of our team

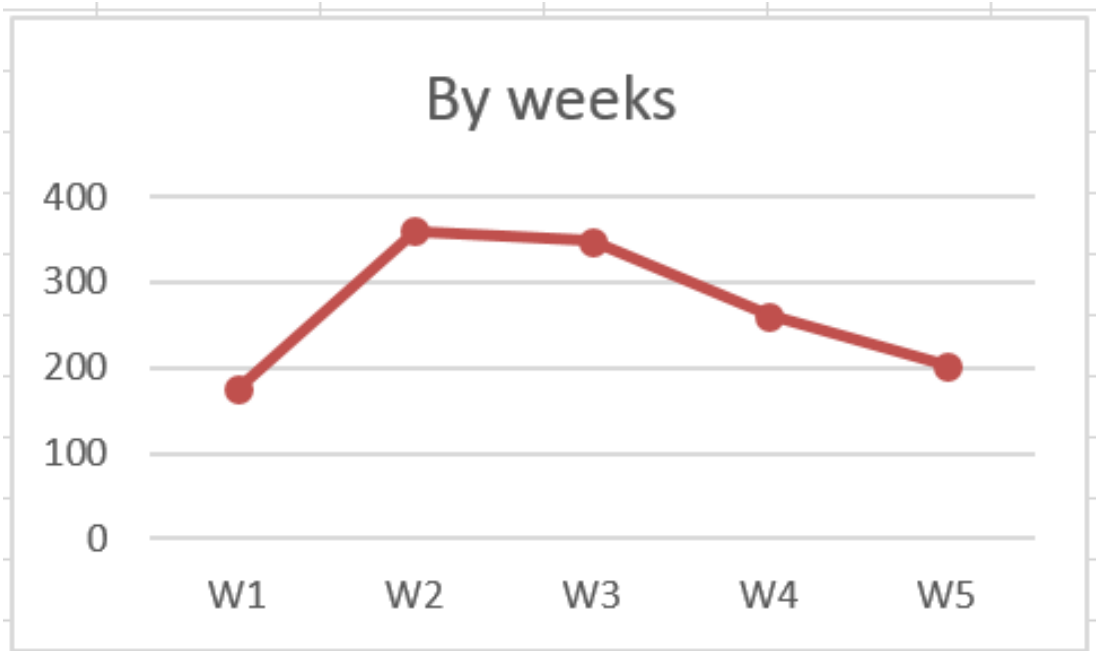
- Project plan

Work flow	Tasks	June					July																	
		week1					week2					week3					week4					week5				
		27	28	29	30	1	4	5	6	7	8	11	12	13	14	15	18	19	20	21	22	25	26	27	28	29
Planning	Determine problem statement, find out architecture drivers, estimate efforts, assign roles create WBS, set milestones, define test plan																									
Requirement analysis	Project context, Risks Description, Functional Requirements, Quality Attributes, Constraints, Write test cases																									
Design	System structure design, Architectural design, Detailed design (physical, static, dynamic view), Design test cases																									
Implementation	Android, Web server, Web application, Arduino, DB																									
Testing	Unit test, Integration, Blackbox test, Requirement testing																									
Documentation	Weekly documentation (ADS, final document, presentation), Regular meeting (with/without mentor), Change request (appendix)																									
Project Management	Project plan update, time logs																									
Meeting	Weekly documentation (ADS, final document, presentation), Regular meeting (with/without mentor), Change request (appendix)																									

Time Log

- By weeks

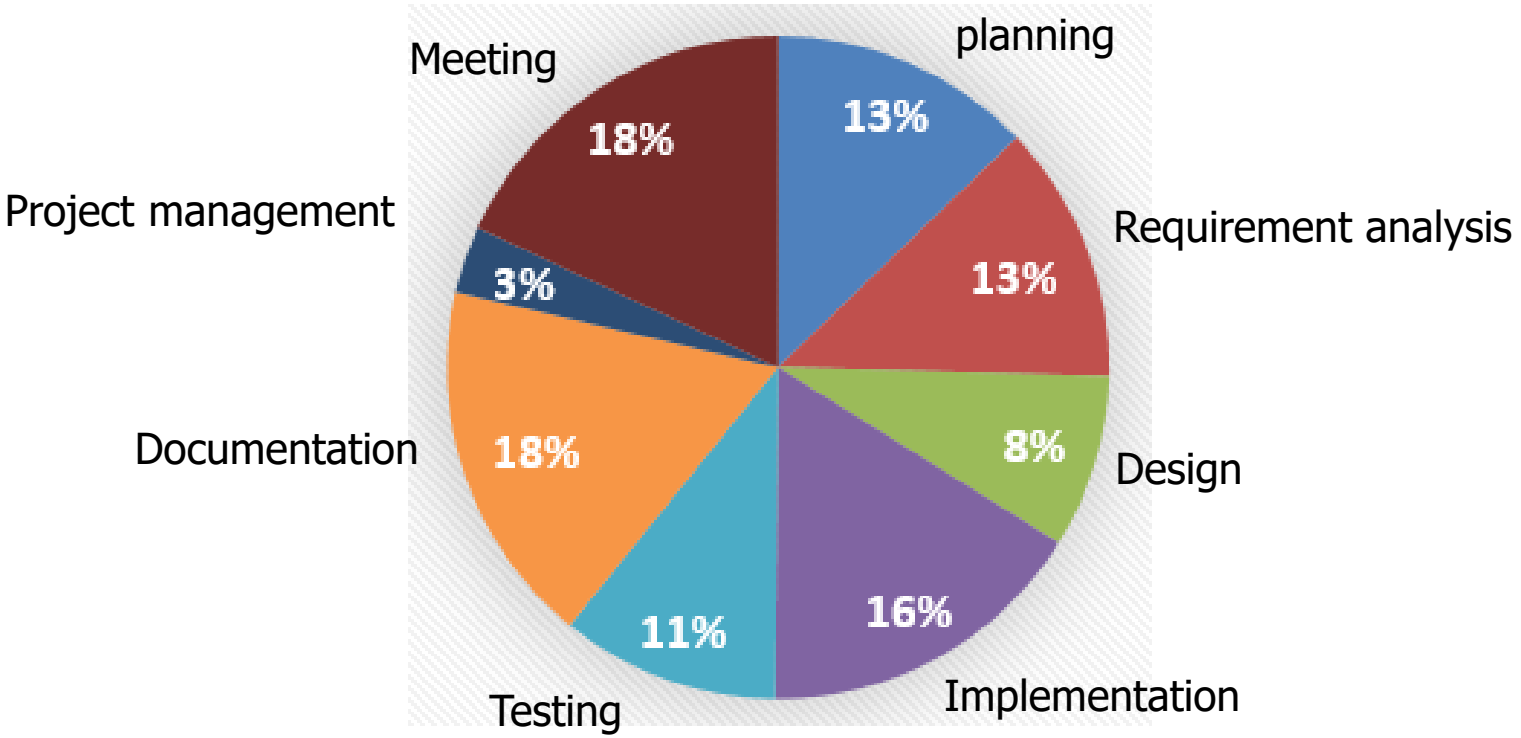
by weeks					
W1	W2	W3	W4	W5	SUM
175	360.5	348.5	261.5	203.5	1349



Time Log

- By process

By process								
Planning	Requirement analysis	Desig	Implementation	Testin	Documentation	Project Management	Meeting	SUM
173	170	114	219.5	144	240.5	44	244	1349



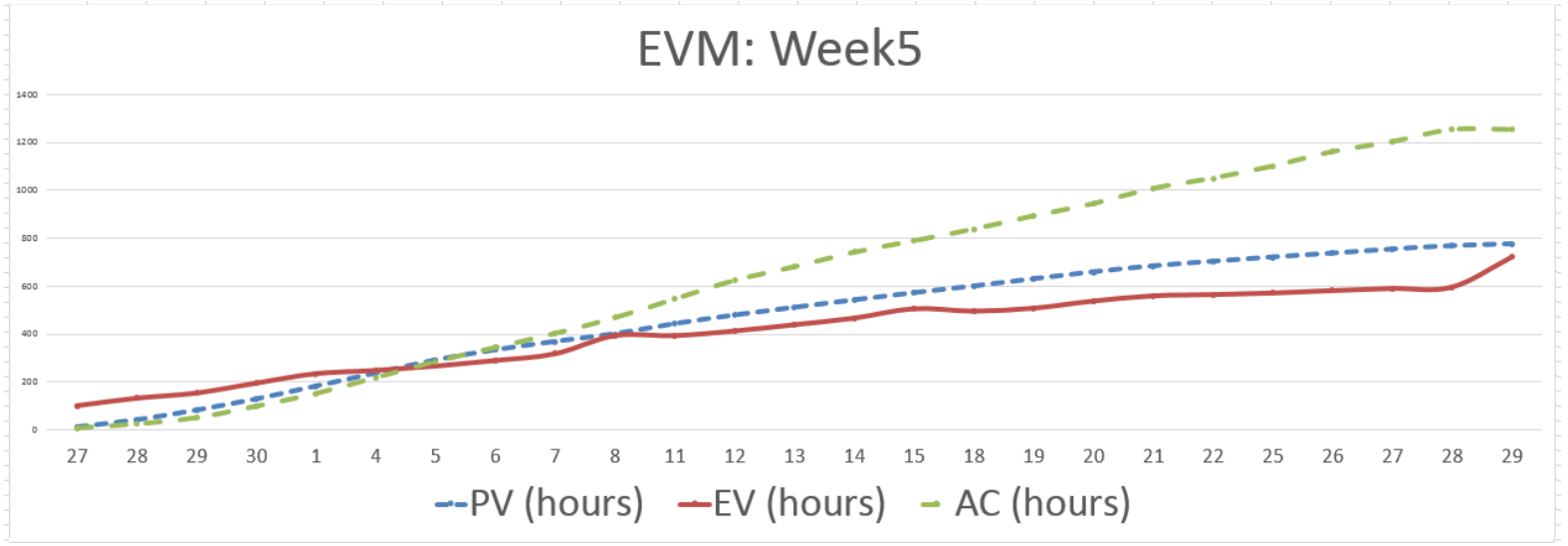
Time Log

- By members

Name	Dongho Lee	Daesoon Kim	Myoungki Hong	Sunghoon Byun	Kyeongseok Yang	Woojin Han	Dongjae Kim
Total Time spent	202.5	176	196	194	192	200.5	188

Tracking

- EVM Chart



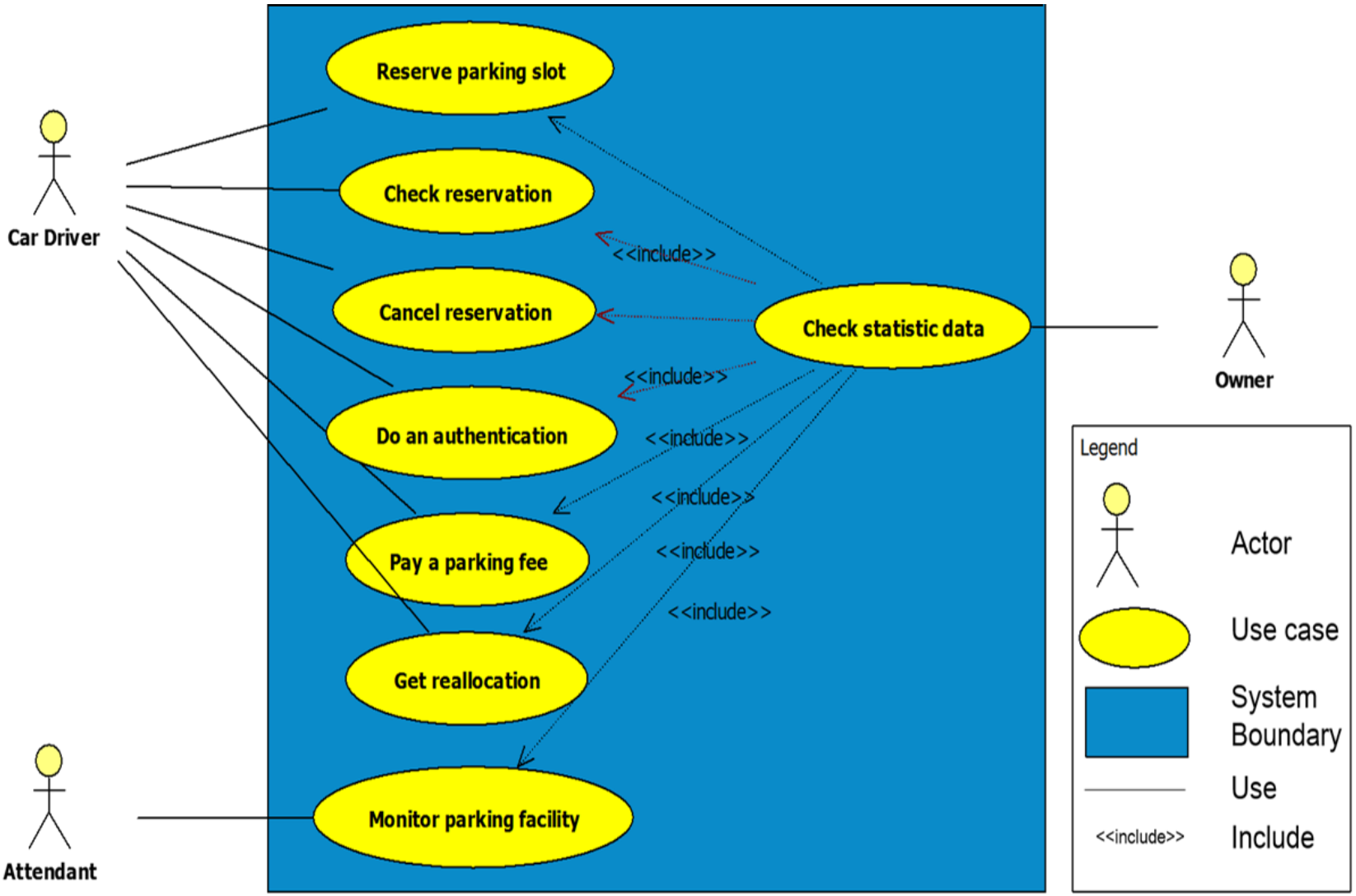
Contents

Customer's needs

- A system that enables users to reserve parking slots using a laptop or phone
- A system that enables a parking attendant to monitor the parking facility
- A system that provides owner with facility usage data that include average usage, peak usage hours, revenue
- A system that allows drivers to reserve parking slots in any of the parking facilities owned by customer and it can be reused and scaled for another parking facility

Functional Requirements

- Use case diagram



Quality Attributes

- Scalability
  - In our system, scalability means that the **system can accept more Arduinos** on condition that the **system performance should not be lowered**, or should be slightly lowered.
- Security
  - In our system, **security means the prevention of malicious** or accidental actions, such as hacking information. It falls within confidentiality part
- Availability
  - In our system, availability means **the proportion of time** in which the system performs its functionality **normally**

Constraints

- Technical constraint

Consideration	Technical constraints
Hardware constraints	Arduino Mega 2560 Arduino Wi-Fi Shield Sensor modules - 4 IR Parking LED (Parking detector) - 4 LED (Parking led) - 2 Gate servo (Gate open/close) - 1 Entry IR LED, 1 Entry IR detector - 1 Exit IR LED, 1 Exit IR detector - 1 Entry LED, 1 Exit LED - 1 Car Pre-made parking facility model (supported by CMU)
Software constraints	- Java Language (Server side) - C/C++ Language (Arduino)
Environmental constraints	- Local sever (laptop) - Wi-Fi Router which is shared with the class - No internet connection - All the system elements should communicate each other through Wi-Fi connection

Constraints

- Business constraint

Consideration	Business Constraints
Stakeholders' demands	Owner: Owner should be able to check daily and monthly statistics of the system (such as daily peak time usage, monthly revenues, parking slot st atistics, and average occupancy). Attendant: Attendant should be able to monitor the parking facility (occup ied slots) by using our system. Car Drivers: Car Drivers should be able to make a parking reservation by using our system.
Available devices limitations	Our team should use the provided hardware, sensors by CMU.
Time limitation	We have only 5 weeks for 'Sure-park' project
Legal restrictions	Private information should be secured.
Policy	Arduino device cannot be taken out of the classroom. Therefore, our team is able to test this only in the classroom.

High priority architectural drivers

- Architectural drivers

Priority	ADS	Description
1st	Time Limitation (Business Constraint)	Our team should complete the project within the limited short period. And also, we should accomplish the required quality.
2nd	Scalability (Quality Attribute)	Our System should operate well when it is sold to the bigger fa cilities, or it should be easily extended larger. It should be scal ed out to a bigger parking lot.
3rd	Security (Quality Attribute)	We should make the information not be stolen by the malicious users. For example, we should protect our customer's credit ca rd information from hacking.

Design Decisions

- General decisions
  - Decide to use android application for reservation
  - Not make login function for car drivers
  - Login the system with ID and password by owner
  - Assign to car drivers parking slot sequentially
  - Assumption on the payment function
  - Directly implement Push Server

Design Decisions

- Scalability-related decisions
  - Server-client pattern
    - Can add multiple Arduino to the server.
    - Servers can be added to support scalability or availability
  - Middleware (RESTful)
    - Makes us convenient to manage **many connections** through only one component
    - Makes communication between **heterogeneous type devices** easier

Design Decisions

- Security-related decisions
  - HTTPs for security of communication
    - Should pay if we use HTTPs
  - AES256 and SHA256 algorithm
    - Protect data in preparation for hacking
  - Authentication step
    - Authentication Number using to enter the parking facility.
  - Login function in the management system
    - The only owner should be able to check statistical data

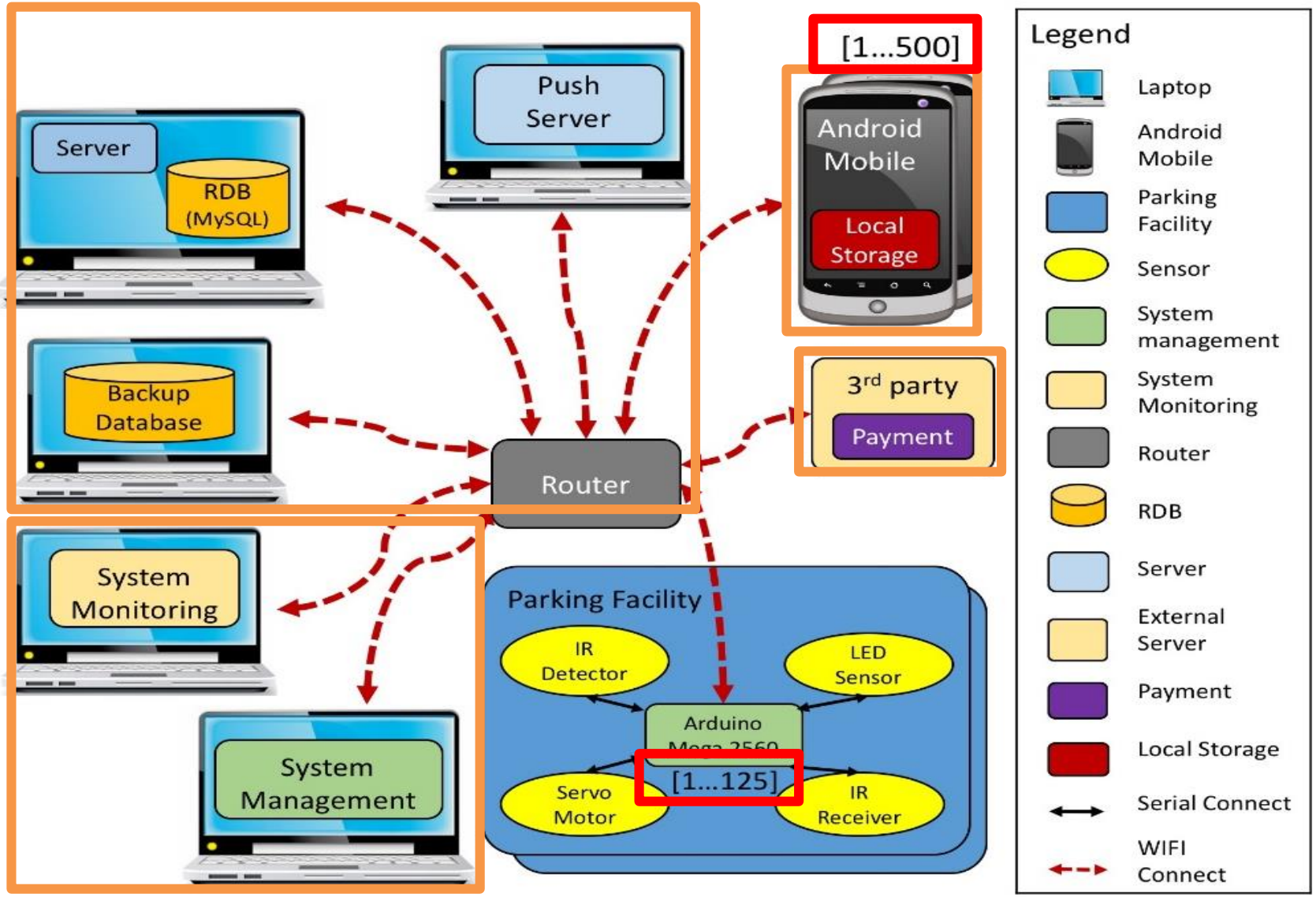
Design Decisions

- Availability-related decisions
  - Backup database
    - Restore fast whenever database does not work
  - ‘Shared preferences’ in Android
    - XML-based local storage
  - Heartbeat pattern
    - Arduino sends its status to server every 10 minutes.
  - Ping& echo pattern
    - Attendant’s monitoring system tries to require server to return parking slot status every two seconds

System Views

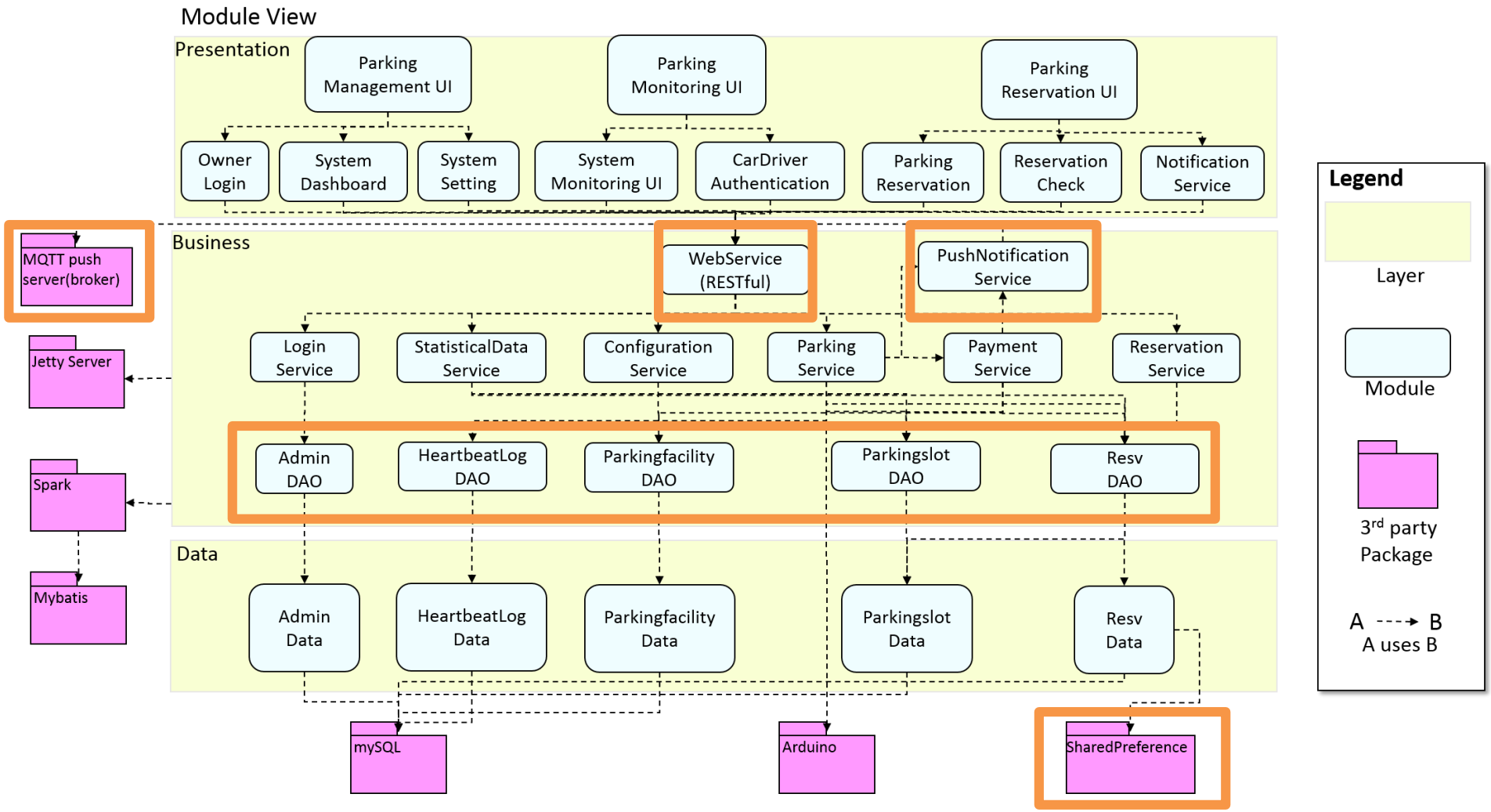
- Physical view
  - This view shows the physical environments in the parking system
- Static view
  - The “module view” and “Class diagram” show how the key elements of the software are mapped to modules and subsystems.
- Dynamic view
  - The “C&C view” and “Sequence diagram” show the flow of system.

Physical View



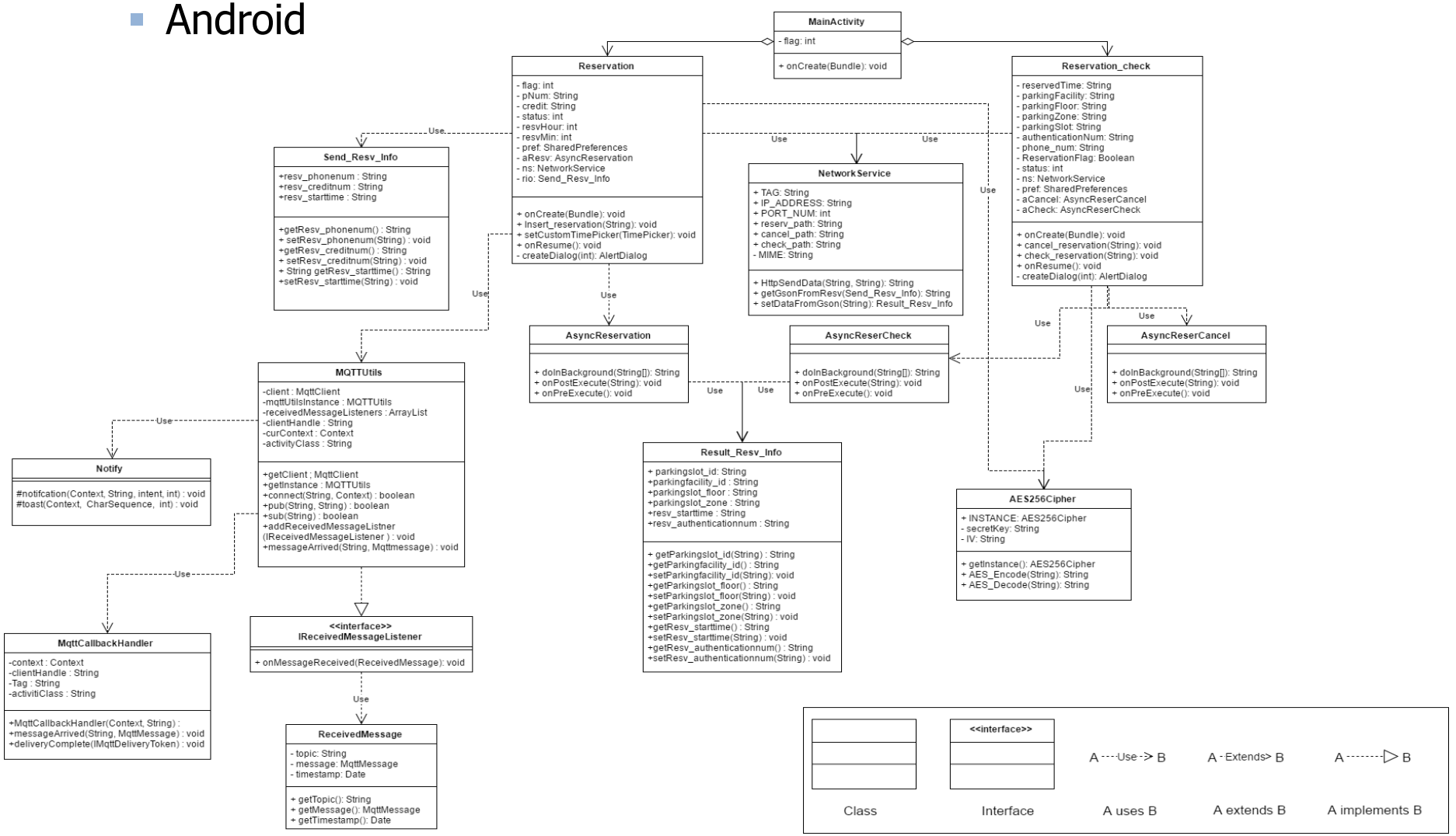
Static View

- Module View



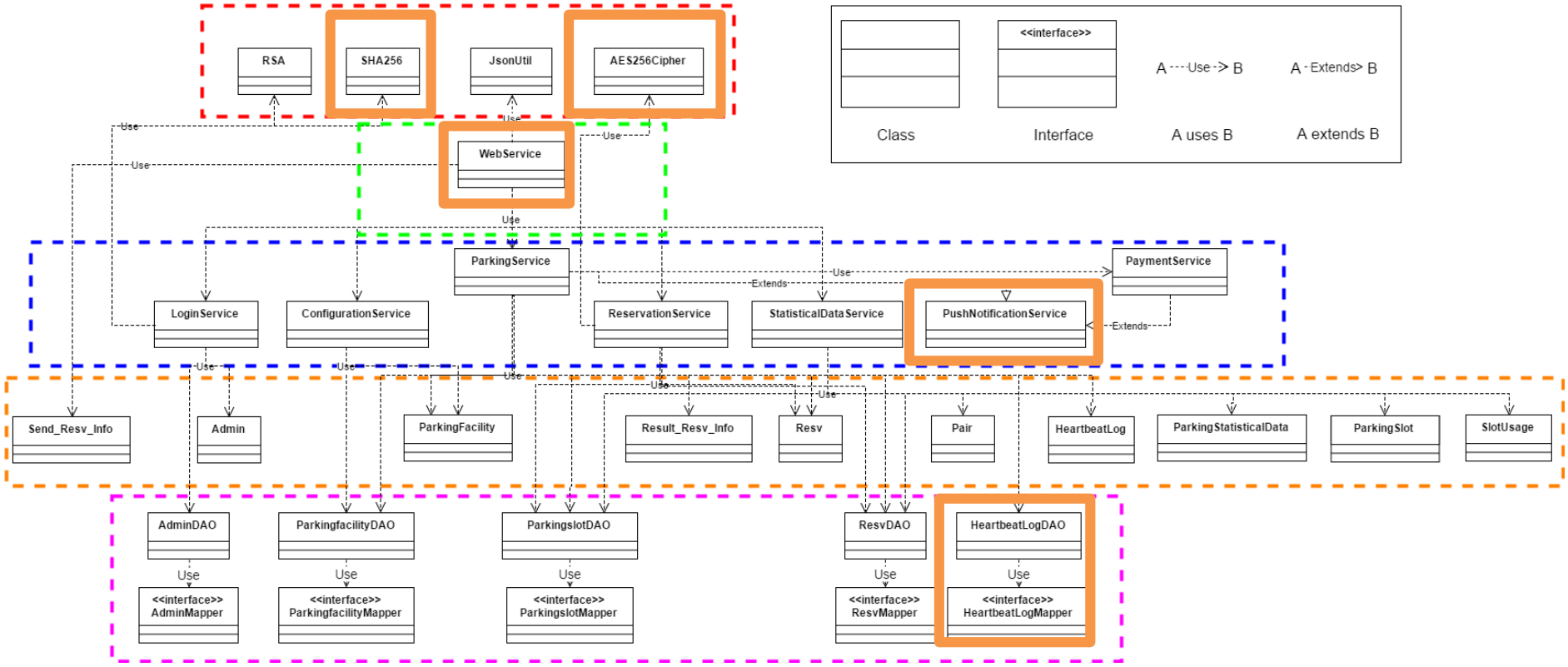
Static View

- Class Diagram
  - Android



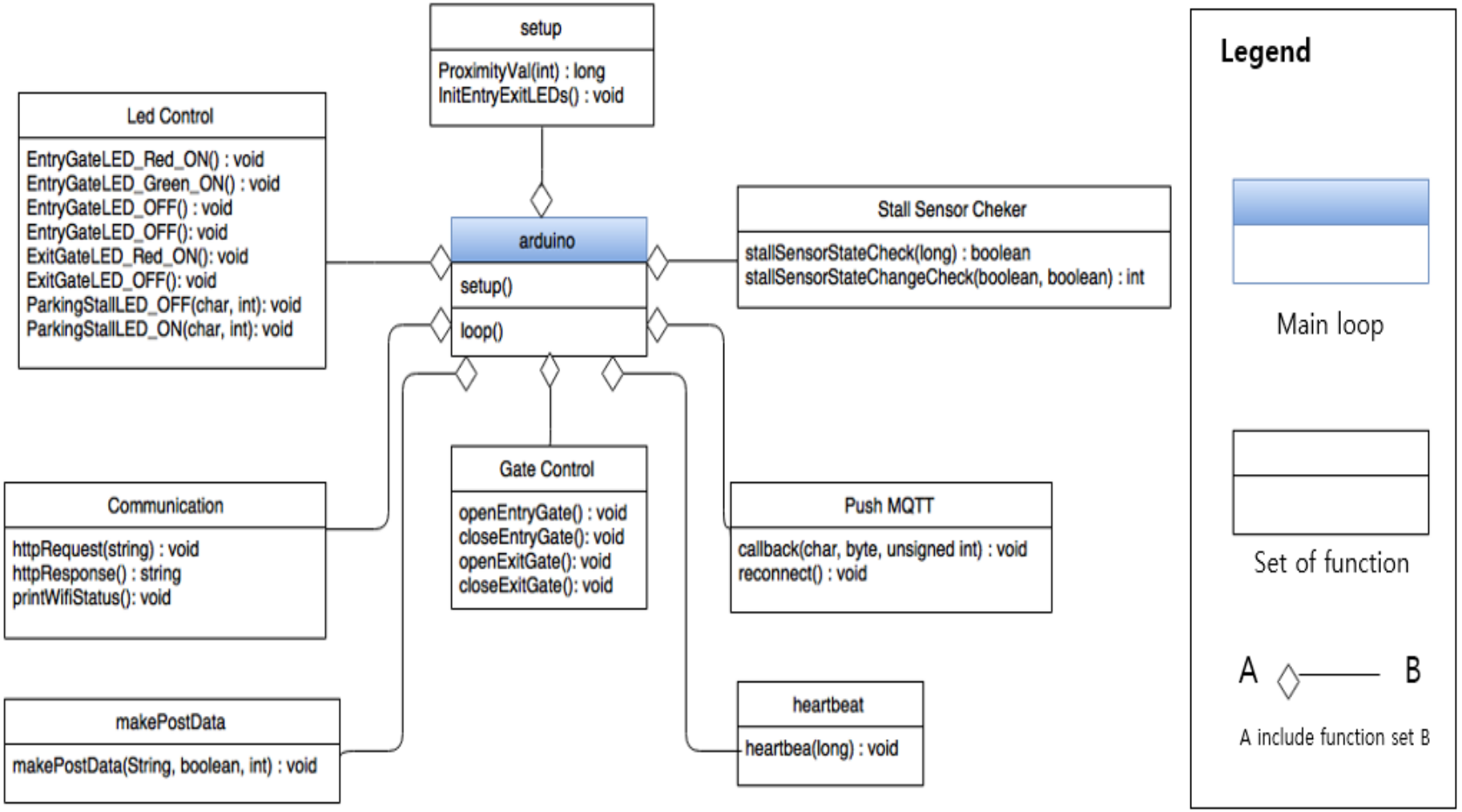
Static View

- Class Diagram
  - Server



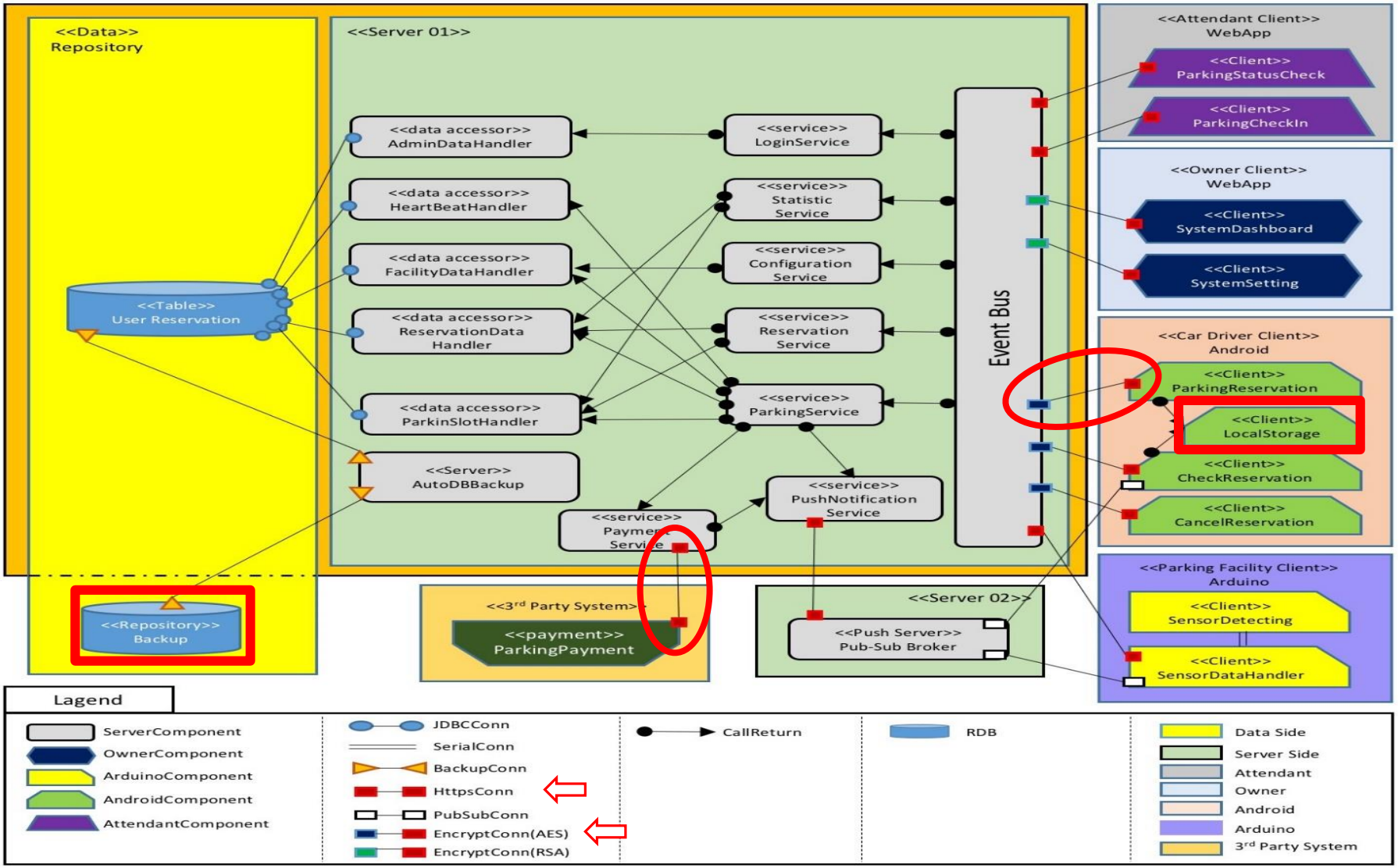
Static View

- Diagram
  - Arduino



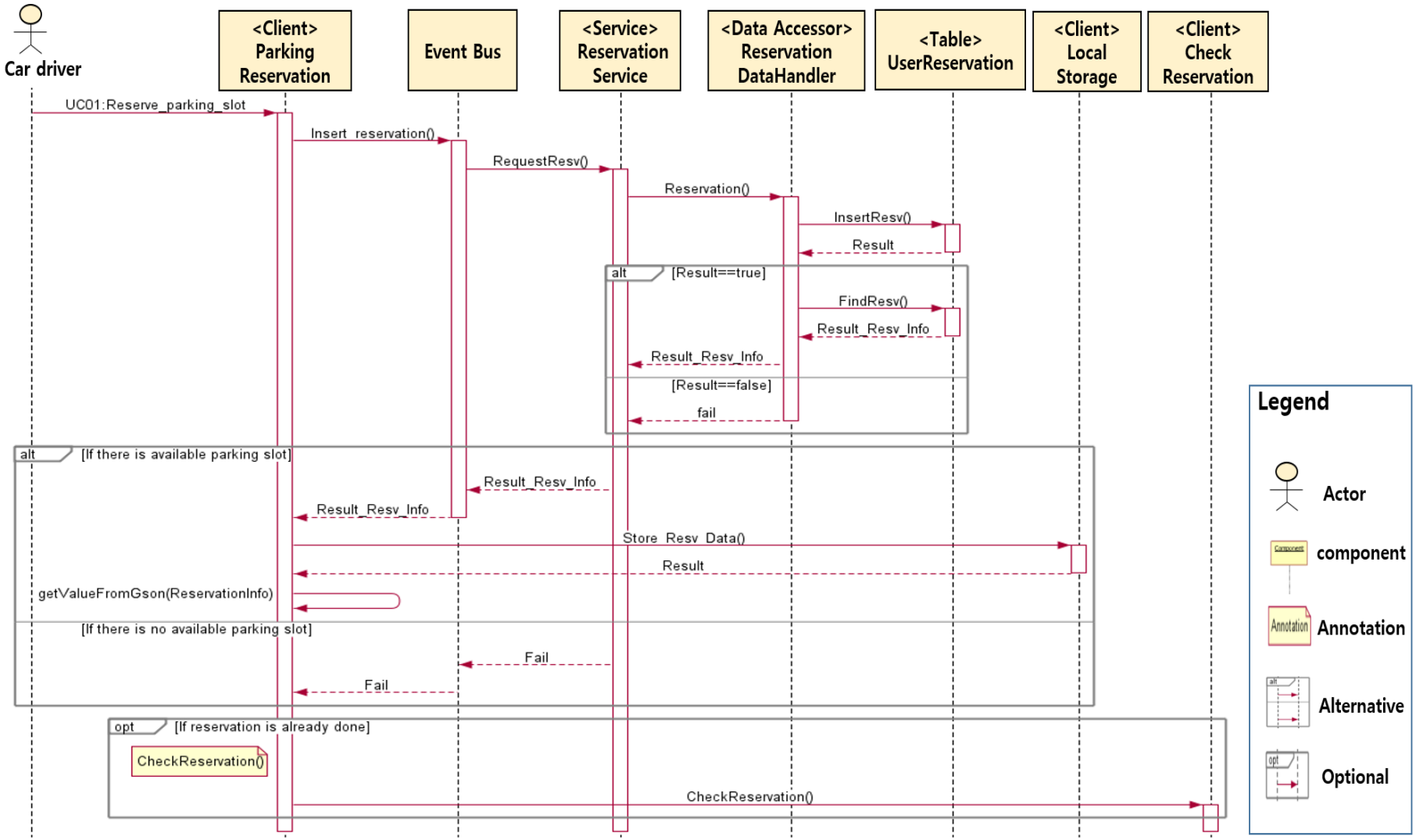
Dynamic View

- Component & Connector View



Dynamic View

- Sequence Diagram



Contents

Testing Plan

- Testing objectives
  - To detect & modify defects in the software
  - To check whether software meets the specified requirements or not
  - To lower the risks of project failure

- Testing Strategy
  - Code review
  - Code coverage testing
  - Integration testing
  - Quality attribute testing

Use case testing

TEST INFORMATION					
Test Case ID:	TC_FR_01	**Related Use Case**	**UC 01**		
Used entities	Car driver, Android, Server, Parking reservation application (Android), and Database,				
Test Description	Test the reservation function in Android application				
Test Priority	High				
Tester's name	Dongho Lee	**Date Tested**	July 22th	**Test result**	**pass**
Test Scenario	Car driver enters the valid reservation time, phone number, credit card number, parking facility ID, and then the driver can reserve the parking slot and check whether the reservation trial is successful or not.				

#	Pre-conditions:	#	Test Data
1	A car driver entered the reservation time, credit card number (16 ciphers) and phone number (9~12 ciphers).	1	Reservation_time = 22:50
2	Server should be connected to Android.	2	Credit_num = 1234567812345678
3	Both Android and server must use the same Wi-Fi router.	3	Phone_num = 4124252315

TEST STEPS				
#	Step	Expected Results	Actual Result	Status
1	A car driver enters reservation time, phone number, credit card number, parking facility ID. Also, he Clicks reservation button.	Reservation screen on Android should show the dialog with result, such as "reservation is available".	The dialog of reservation result was shown with result, such as "reservation is available".	**pass**
2	The car driver clicks "OK" button in the dialog screen.	Android start communicating request of reservation with server.	Android communicated request of reservation with the serve by using HTTP.	**pass**
3	Server receives the request of reservation and if the reservation is available, then server return this information to Android.	Result of reservation should be received, and reservation information should be stored in local storage (shared preference).	Android received the reservation information, and stored reservation information in local storage (shared preference).	**pass**
4	Android screen is switched to reservation check screen.	Reservation check screen should show reservation information.	The car driver checked the reservation information on reservation check screen.	**pass**
Post-Condition:	The car driver received success information of reservation from sever, and store this information in Android local storage (shared preference). Also, he checked his reservation information on Android reservation check screen.			

Code coverage testing

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
spark_web	77.7 %	3,634	1,045	4,679
src/main/java	77.7 %	3,634	1,045	4,679
com.spark_web.dao	55.9 %	964	759	1,723
ResvDAO.java	61.4 %	602	379	981
ParkingslotDAO.java	47.9 %	135	147	282
AdminDAO.java	48.1 %	103	111	214
ParkingfacilityDAO.java	51.4 %	75	71	146
HeartbeatLogDAO.java	49.0 %	49	51	100
com.spark_web.service	92.6 %	1,622	130	1,752
ParkingSerivice.java	91.0 %	666	66	732
PushNotificationService.java	64.1 %	93	52	145
LoginService.java	94.0 %	94	6	100
ReservationService.java	95.2 %	119	6	125
ConfigurationService.java	100.0 %	49	0	49
PaymentService.java	100.0 %	13	0	13
StaticalDataService.java	100.0 %	588	0	588
com.spark_web.util	73.8 %	237	84	321
AesUtil.java	53.6 %	81	70	151
SHA256.java	84.6 %	44	8	52
AES256Cipher.java	97.0 %	97	3	100
JsonUtil.java	83.3 %	15	3	18
com.spark_web.domain	84.9 %	298	53	351
ParkingSlot.java	66.7 %	30	15	45
Send_Resv_Info.java	50.0 %	12	12	24
Admin.java	70.8 %	17	7	24
Resv.java	90.4 %	66	7	73
Pair.java	73.9 %	17	6	23
Result_Resv_Info.java	86.7 %	39	6	45
HeartbeatLog.java	100.0 %	31	0	31
ParkingFacility.java	100.0 %	31	0	31
ParkingStaticalData.java	100.0 %	38	0	38
SlotUsage.java	100.0 %	17	0	17
spark_web	96.4 %	513	19	532
WebService.java	96.4 %	513	19	532

Quality attributes testing

- Security

Name	Design decisions description
Authentication number	We perform an authentication step. If we perform an authentication step by usi ng only credit card number or phone number, there may be security risks. Ther efore, we give car drivers authentication number when car drivers make a reser vation. Car drivers should show their given number to enter the parking facility.
AES256	When server and clients communicate, we transmit data which are encrypted w ith AES256. Therefore, we can protect these data. Also, server stores client's da ta encrypted with AES256. Therefore, it is less risky although this data is hacke d by malicious users.
SHA256	When the owner tries to log in the owner's parking management system, we tr ansmits owner's data by encrypting this data with random key. In other words, we use token. Therefore, we can protect the owner's information.

Quality attributes testing

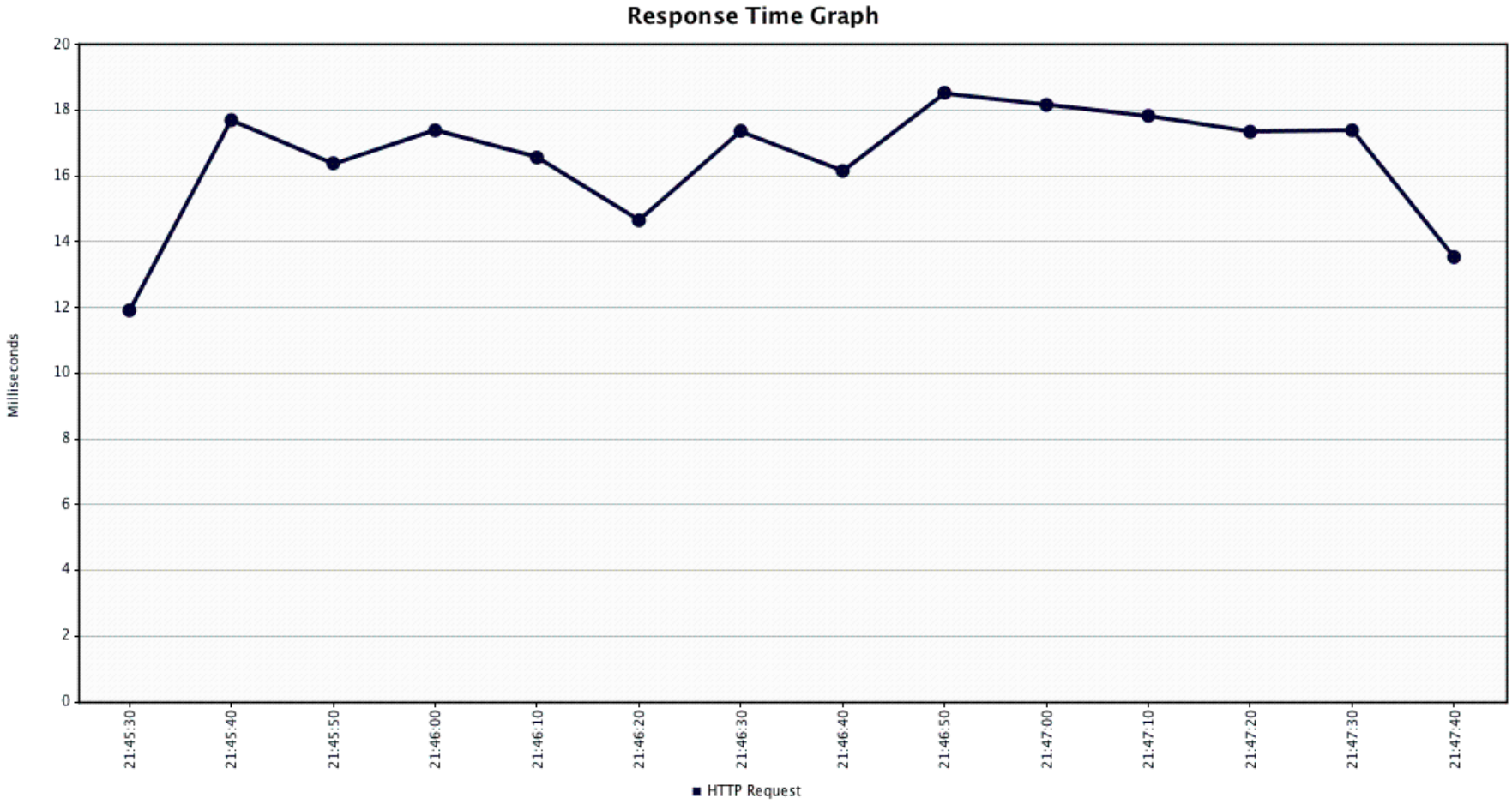
- Availability

Name	Design decisions description
Backup DB	We made a backup database. Because we stores information of main database in to backup database periodically, we can restore fast through backup DB whenever database does not work.
Shared Preferences	We used 'shared preferences' in Android. It is a XML-based local storage. Even though Android cannot communicate with server, Android can get data from its local storage
Heartbeat Pattern	We used heartbeat pattern. Whenever malfunction of Arduino occurs, we should fix it as soon as possible. Therefore, we should check periodically whether Arduino works normally or not. To do this, we used heartbeat pattern. Arduino sends its status to server every 30 minutes. The systems that are directly related to human safety should be checked every seconds, but our system has no such a safety-critical characteristics. Therefore, we decided that 30 minutes is acceptable to our system.
Ping & echo Pattern	We used ping & echo pattern. Its purpose is to fix server as soon as possible in case it undergoes failure. Attendant's parking monitoring system checks the status of server every two seconds. In detail, attendant tries to require server to return parking slot status every two seconds. At this time, if there is no response from server, we can decide that there some problems with our server

Quality attributes testing

- Scalability

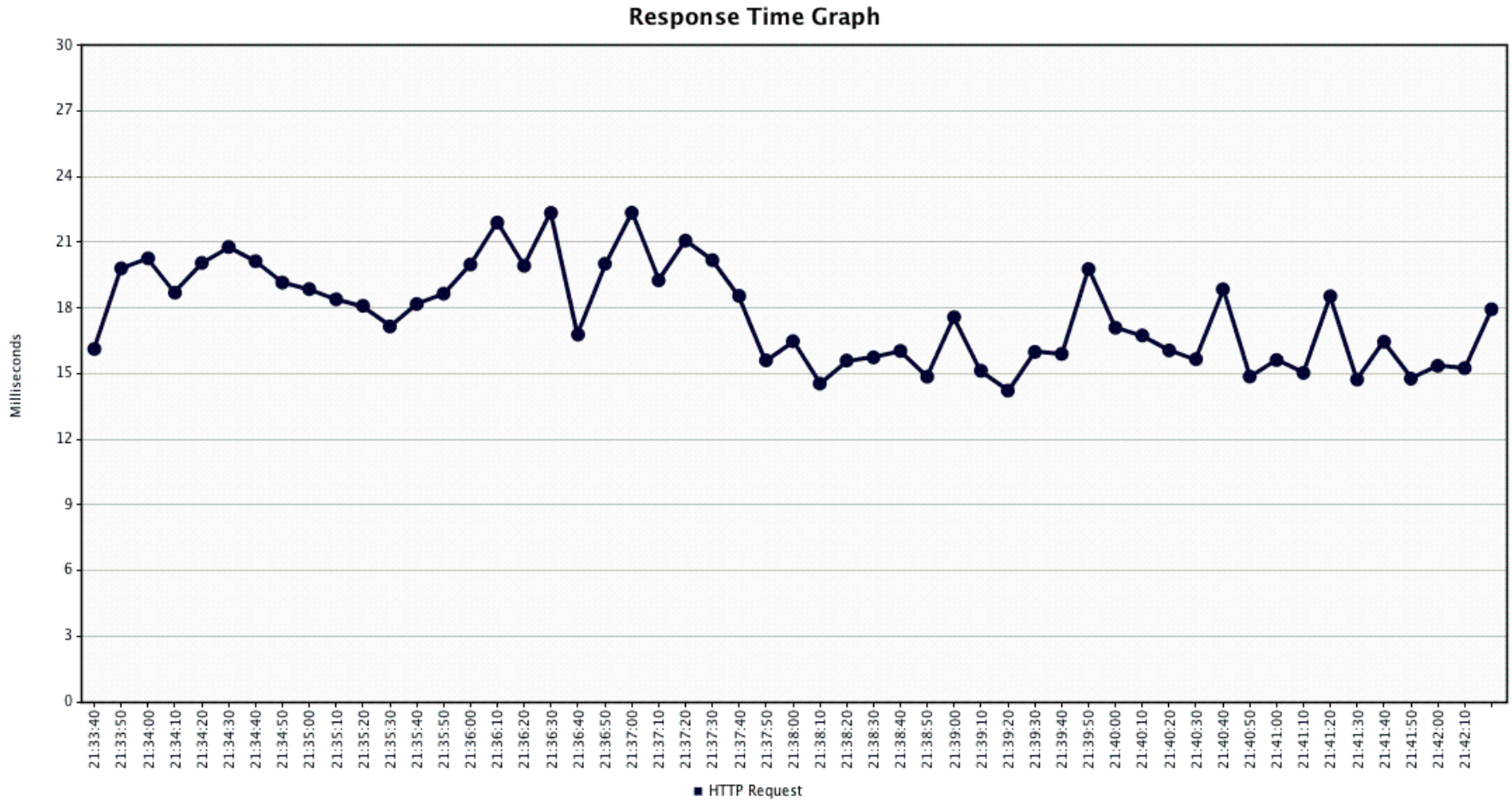
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	75000	16	4	3097	33.81	0.00%	547.7/sec	92.00	172.0
TOTAL	75000	16	4	3097	33.81	0.00%	547.7/sec	92.00	172.0



Quality attributes testing

- Scalability

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	300000	17	3	3029	32.48	0.00%	561.3/sec	198.97	363.0
TOTAL	300000	17	3	3029	32.48	0.00%	561.3/sec	198.97	363.0



Contents

- Introduction of team C
- Project Description
- Project Plan
- Architectural Driver Specification
- Test Plan
- Lessons Learned
  - Team perspective
  - Technical perspective
  - Architecture perspective
- Demonstration
- Q n A

Lessons Learned

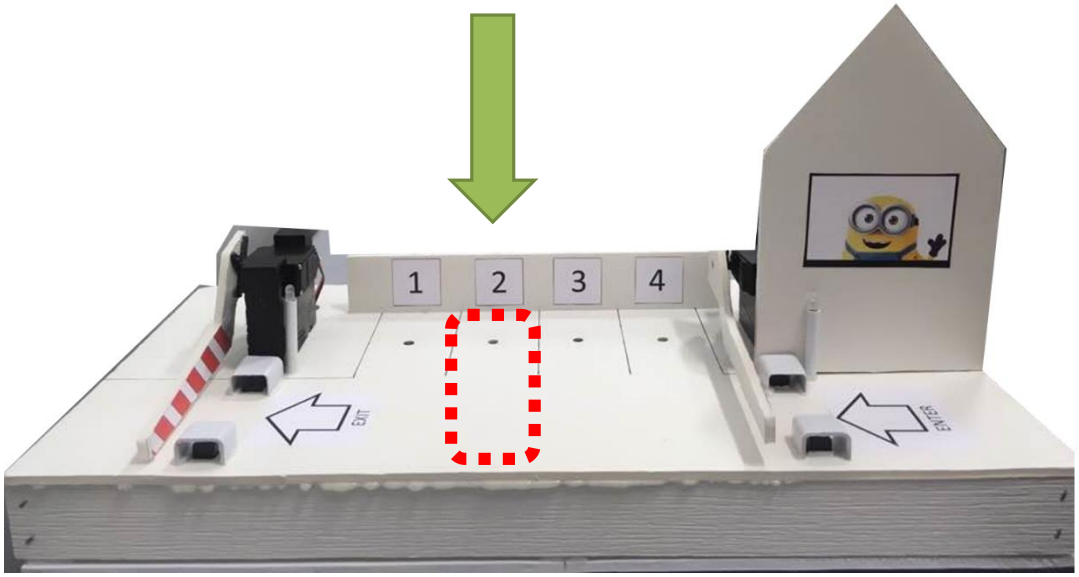
- Team perspective
  - Time management problems
  - Preliminary knowledge & Ability of members
- Technical perspective
  - Limited network
  - Environment configuration problems
  - Development
- Architecture perspective
  - Requirement analysis
  - Design architecture considering QA
  - Design decision problems

Contents

Surepark cartoon

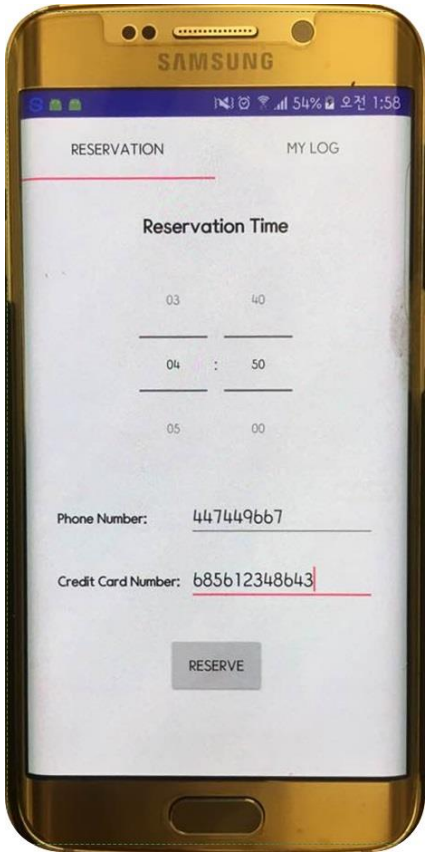
NEXT

reservation



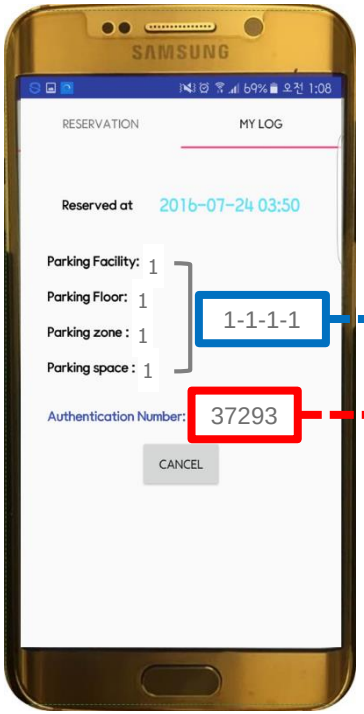
NEXT


Let's find a parking slot and reserve a slot with smart phone.


SAMSUNG
54% 오전 1:58
RESERVATION
MY LOG
Reservation Time
03
40
04
:
50
05
00
Phone Number:
447449667
Credit Card Number:
685612348643
RESERVE

NEXT


Oh! My reservation number is "37293". Thank you~!
your parking slot is 1-1-1-1


SAMSUNG
RESERVATION
MY LOG
Reserved at 2016-07-24 03:50
Parking Facility: 1
Parking Floor: 1
Parking zone : 1
Parking space : 1
1-1-1-1
Authentication Number:
37293
CANCEL

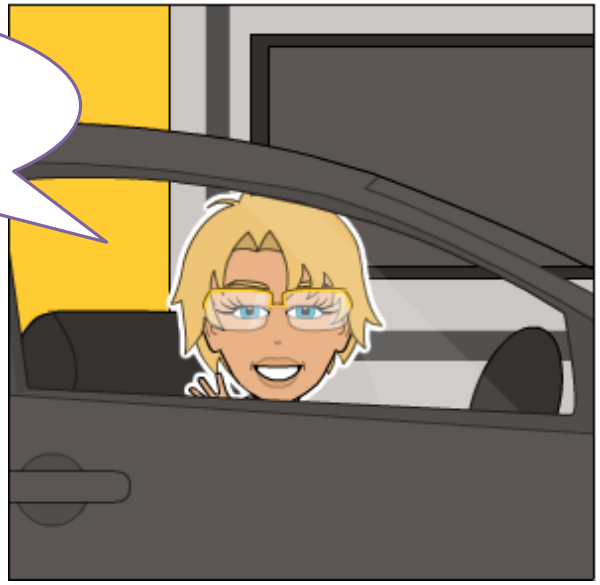
NEXT

surepark

http://wilkinsonbrothers.com/portfolio/parking-map-illustration/#prettyPhoto

NEXT

My reservation number is 37293



Good!
Your reservation is approved.
Your parking slot is 1-1-1-1.
Entry gate is opened!

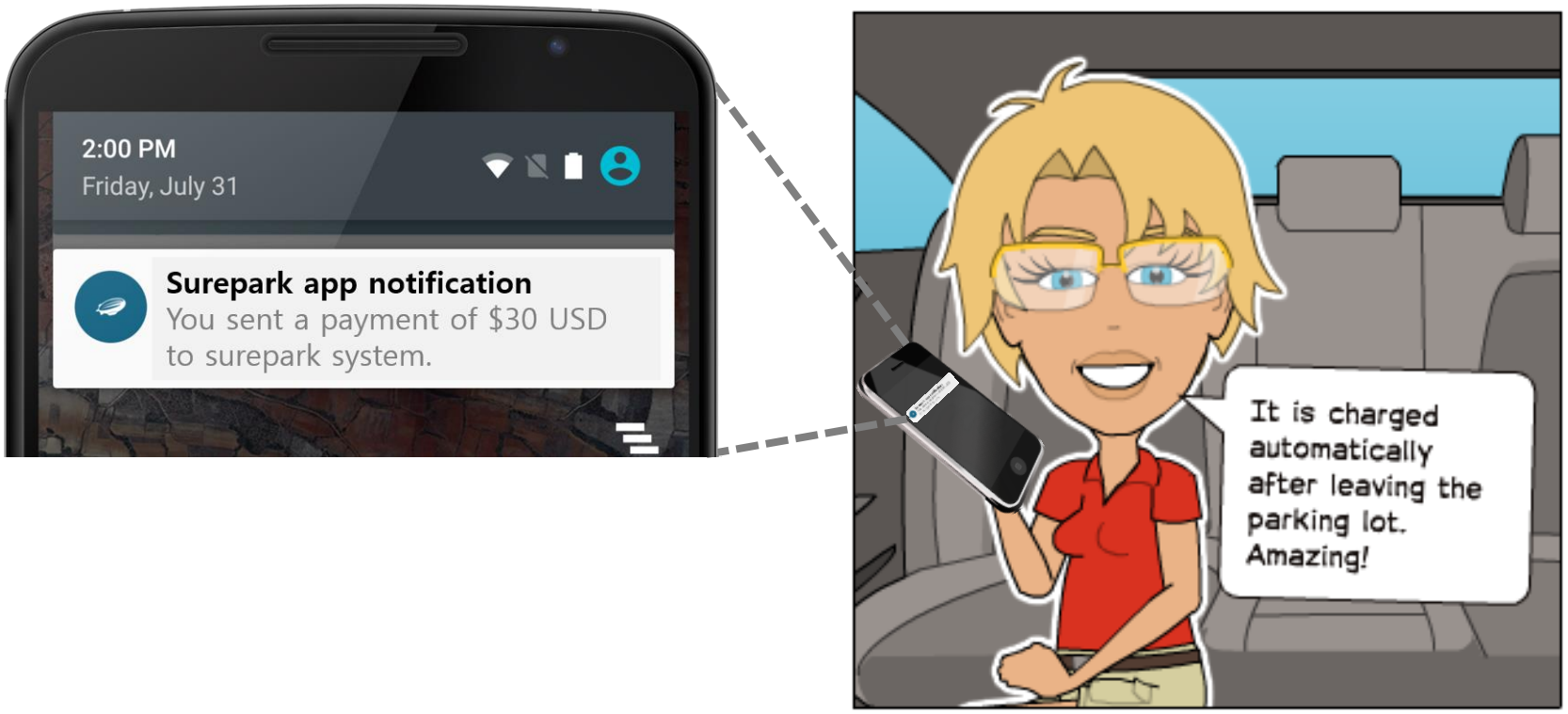
NEXT

5 hours later …

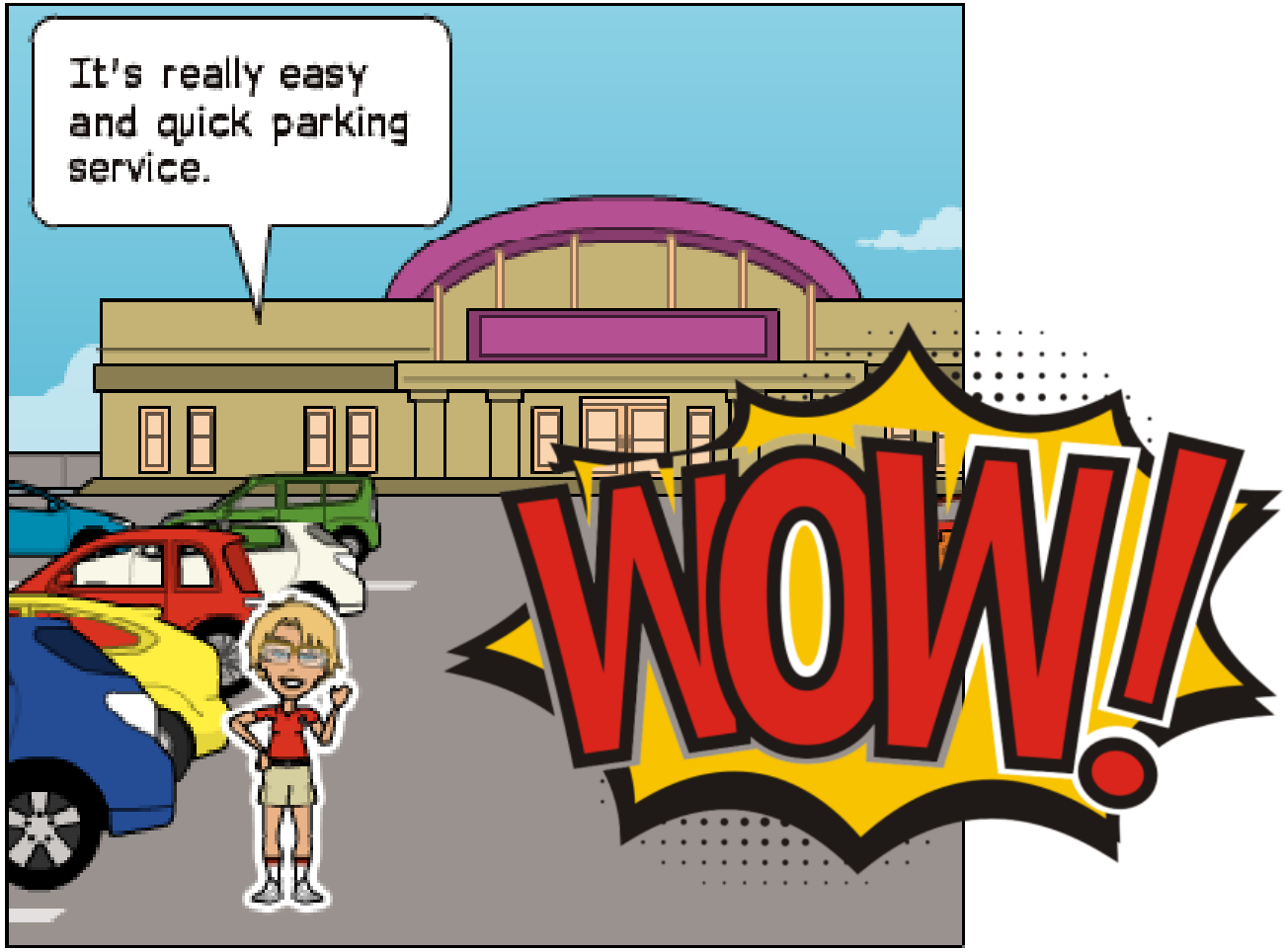
NEXT


It is time to go back home.
What I have to do is ...
Just go through the exit gate!

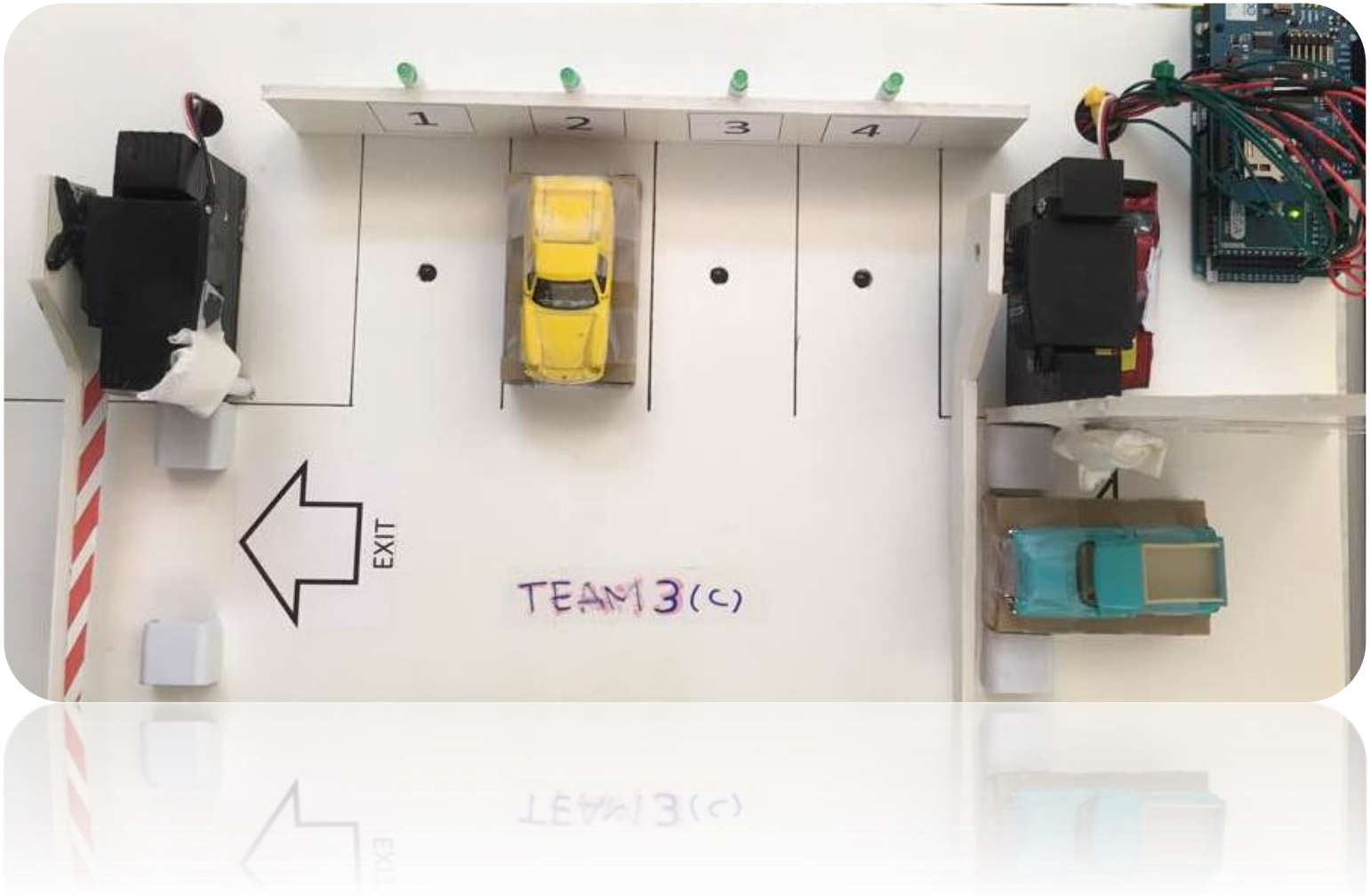
NEXT


2:00 PM
Friday, July 31
Surepark app notification
You sent a payment of $30 USD to surepark system.
It is charged automatically after leaving the parking lot. Amazing!

NEXT



Real Demo!



Q & A

Thank you