Phase 3:Project:
Development of a smart parking system:

- **Introduction**

Technological advances in the present increasingly provide convenience to us. The internet does not only connect one computer to another, but all types of devices that have internet protocols can be connected. The evolution of the internet has made the concept of the Internet of Things (IoT) no stranger to the world of technology. IoT connects almost all types of devices that support human life such as televisions, washing machines, refrigerators, and others so that they can be controlled remotely by users via smartphones. IoT is an effective solution to the time and cost problem of a job that is handled by humans.

IoT in Indonesia has a positive trend and is starting to be considered a lucrative business opportunity even though its ecosystem is not as strong as other sectors such as e-commerce. The development of IoT is supported by the launch of the industry 4.0 road map by the government where one of the priority sectors is IoT . It is estimated that there are 14.2 billion IoT devices in 2019 and are predicted to reach 25 billion in 2021. Technology initiatives that have the most influence on the development of the company according to more than 500 senior company executives, are IoT technologies with 33% outperforming robot technology, Artificial Intelligence (AI), and Augmented Reality (AR) based on Forbes Insight research.

IoT as enabling technology from an industrial point of view is divided into several IoT functionalities including wearable devices (Smart Wearable), home automation devices (Smart Home), environmental monitoring devices (Smart Environment), corporate infrastructure support devices (Smart Enterprise), and urban infrastructure support devices (Smart City) . IoT can monitor, manage and control equipment in real-time so that it supports the establishment of various Smart Initiatives in the Smart City concept that prioritizes responsive, reactive, and proactive services.

Smart Parking is one of the Smart Initiatives that tries to provide a solution to the classic problems surrounding parking lots in big cities. The growth in the number of vehicle populations has a direct impact on the problem of traffic density and parking areas. Parking has become one of the crucial things in road traffic, especially in urban areas, therefore parking problems are regulated in Law No. 14 of 1992 concerning Road Traffic and Transport. Parking problems if categorized can be divided into 3 categories, namely: parking lots, parking fees, and parking rules. Infrastructure in Indonesia, especially parking lots, generally does not compensate for the problem of increasing motorized vehicles. This problem raises new problems such as illegal parking which have an impact on public road congestion, for example, roads in the city of Semarang.

The ability of real-time data presented by IoT can be an opportunity to provide comfort in finding a parking space in the desired location so that by knowing the location of the parking before the consumer comes to the desired location the consumer can determine whether he will visit location A or B. Real-time information parking spaces and even direct bookings are very possible to do with IoT, thus establishing the concept of E-Parking. Commission B of the Semarang City DPRD urges the city government to immediately implement E-Parking , seeing that parking conditions in the city of Semarang are increasingly irregular because of the parking mafia.

- **Literature and method review**

This chapter contains a review of the literature and basic theories used in working on this study. Literature can be in the form of journal books, articles, or other forms related to the object being studied to support the completion of this study.

- *Internet of things*

The Internet of Things (IoT) began in 1991 when Mark Weiser described the vision for the future of the internet under the name Ubiquitous Computing. The word Internet of Things itself appeared in 1999 spearheaded by Kevin Ashton. The IoT concept entered a new era when it was published by the International Telecommunication Union (ITU) in 2005 through ITU Internet Reports 2005: The Internet of Things. ITU reports that IoT is all objects that can be connected not related to time or place through radio frequency identification technology (RFID), wireless sensor networks technology, intelligent embedded technology and nanotechnology.

IoT's definition of network connection to connect everything with the internet connects the protocol established through information sensing equipment to communicate information and communication to achieve intelligent recognition, connecting position, conversation, and administration. IoT architecture is composed of several layers of technology that support IoT. This IoT architecture layer illustrates various technologies that are connected and to connect scalability, modularity and manage IoT deployments in different scenarios. Figure provides an overview of the architecture and benefits of the broad benefits of implementing IoT from Smart Living to Smart Homes.

IoT can monitor, manage, and control equipment in real-time so that it supports the establishment of various Smart Initiatives in the Smart City concept that prioritizes responsive, reactive, and proactive services. The ability of real-time data presented by IoT can be an opportunity to provide support for various activities that require actual information or data.

- ***Object-oriented analysis and design method:***

The object-oriented development method is a software development approach where the fundamental abstraction in the system is object independent. The type of abstraction used is the same as in specifications, design, and development. An object is an entity that includes state, behavior, and identity. The structure and properties of similar objects are explained in the general class, where the instances and objects of the term can be exchanged. Object-Oriented Analysis and Design is a method that leads to object-oriented decomposition. The relation includes:

- ***Object-oriented programming (OOP).*** Object-Oriented Programming is an implementation method where the program is organized as a collection of interconnected objects. Each part represents an instance of some classes and all class parts of a hierarchy of classes are connected through inheritance relationships.

- ***object-oriented analysis (OOA).*** Object-Oriented Analysis is a method of analysis that observes requirements from the perspective of classes and objects encountered in a problem domain.

- ***Object-Oriented Design (OOD).*** Object-Oriented Design is a method that includes an object-oriented decomposition process and provides notation to describe the physical and logic as well as static and dynamic of the system being designed. The results from OOA will be presented as a model for starting OOD. The results of OOD can be used as a basis for implementing a system using the OOP method. The object-based software development methodology, Ripple, is directed to study what is involved in all software development both large and small, but can also be applied to the real world.

- ***Related works:***

Smart Parking is one of the Smart Initiatives that tries to provide a solution to the classic problems surrounding parking lots in big cities. Smart Parking is one of the topics that is growing quite popular and is often associated with the Internet of Things. The Internet of Things is the main actor behind the concept of Smart Cities. One of the things that make the Internet of Things discussed quite often is the

benefits provided. Table 1 provides information about some of the Internet of Things research that has been done in the Smart Parking topic.

Table: List of articles related to smart parking topic

No.	Author(s)	Title	Method
1.	Wang (2011)	A Reservation-based Smart Parking System	Reservation-based
2.	Pham et al (2015)	A Cloud-based Smart Parking System Based on Internet of Things Technologies	Cloud-based
3.	Fraifer (2016)	Smart Parking System Prototype Utilizing CCTV Nodes	CCTV
4.	Khanna (2016)	IoT based Smart Parking System	IoT-based

According to Table, the Smart Parking research conducted by Wang was applied by building a prototype of a parking system using the reservation method. In 2015, Pham proposed a parking system using cloud services so that it can provide better performance. Fraifer contributed to the topic of Smart Parking through his research which proposed the architecture of the Smart Parking system by using closed-circuit television (CCTV) devices. Then the research conducted by Khanna, built a Smart Parking using the internet of things integrated with cloud services. This thesis uses the four studies mentioned in Table 1 as a reference related to the development of smart parking systems, specifically the research of IoT based Smart Parking Systems by Khanna.

- ***Smart parking system architecture:***

Smart Parking System (SPS) is a system with an architecture consisting of several application platforms and integrated with embedded systems. The details of the SPS subsystem are explained as follows:

- ***Mobile apps:*** Mobile apps in SPS are mobile applications built in the Android OS, using the Android Studio application. Mobile apps are used by user actors. Mobile apps have parking slot booking and unlock parking slot functions.

- ***Web apps:*** Web apps in SPS are written in the Python programming language with the help of the Django web framework. Web apps provide APIs written in the PHP programming language and JavaScript as a communication bridge between server and client. Web apps are installed on one computer that acts as a server. Web apps are only used by admin actors with various functions such as booking parking slots, unlocking parking slots, adding user accounts, viewing history, adding parking slots, and removing parking slots.

- ***Embedded system:*** The embedded system built on SPS is a part of the system that has the main function as an indicator of booking as well as locking the parking slot. The SPS embedded system uses the Raspberry Pi device and the ESP8266 Wemos D1 Mini wifi module. In SPS communication, Raspberry Pi acts as a master and ESP8266 as a slave. Components of an SPS embedded system consistof :

- HC-SR04 Ultrasonic Sensors (2 pcs);

- ESP8266 Wemos D1 Mini (1pc);
- Buzzer (1pc);
- Resistor 1kΩ (1pc);
- RGB Led 5mm (1pc).

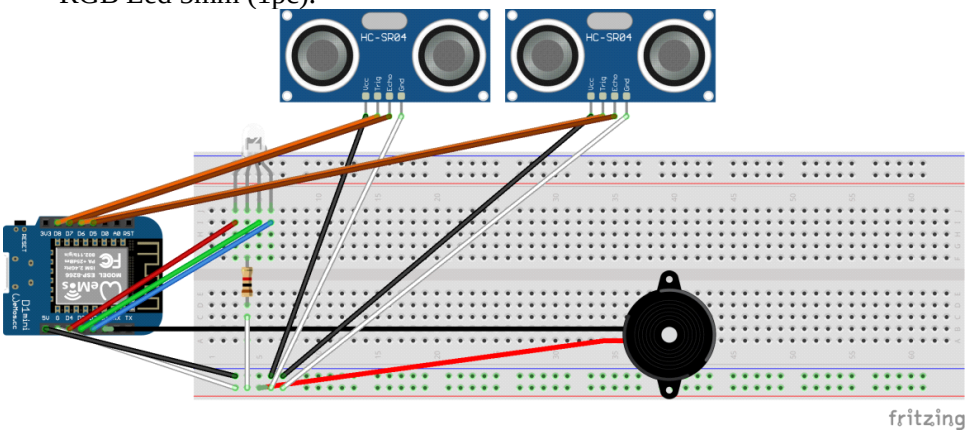

Figure SPS embedded systems scheme

In Figure an image of the SPS embedded system scheme is built from the components written earlier. Led has a function as a slot indicator light in a locked state or not. The buzzer has a function as a sound indicator when the slot is locked but some vehicles occupy the slot. Two ultrasonic sensors work as a vehicle detector in the parking slot by measuring the distance.

SPS has two parking data storage with the details of one main database with the MySQL engine installed on the localhost server computer that has the SPS web apps installed, and one small database with the MariaDB engine on the Raspberry Pi. A small database is built to store temporary data on the Raspberry Pi, if the SPS web apps server is down, so the small database only stores parking slot data. While the main database stores all information such as history, parking slots, user accounts, and bookings.SPS has 2 actors involved namely user and admin. Users are system users through mobile apps that can book parking slots and unlock parking slots. Admin is a system user through web apps that can

add parking slots, delete parking slots, add user accounts, view history, book parking slots, and unlock parking slots. The complete Smart Parking System process flow is illustrated in Figure

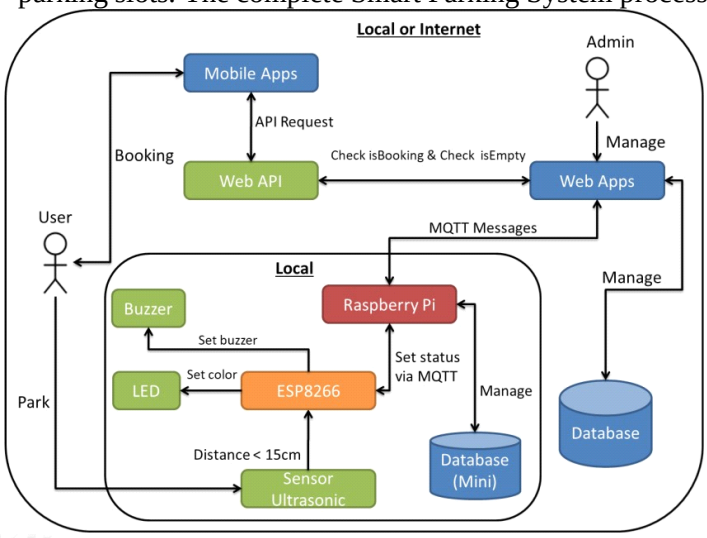

Figure Smart Parking System Process Flow

- ***Development phases:***

Smart Parking System (SPS) has two applications namely web apps and mobile apps that are integrated with embedded systems. Each application is written in a different programming language but with the same paradigm of object-based programming, so the use of the Object-Oriented Analysis and Design (OOAD) method will facilitate the process of software development and documentation. The artifact implementation and SPS phase are adjusted to the conditions in the software development process, created by referring to the Ripple methodology described by Docherty. The phases and artifacts are explained as follows:

- ***Genesis:*** The genesis phase is the stage to find what is needed by the customer. In SPS software, the genesis phase produces three artifacts, namely a glossary that contains terms used in development to reduce redundancies, and a test plan that will be carried out during the testing phase.

- ***Requirements:*** The requirements phase is the stage to translate what is needed by the customer into a complete and unambiguous description, by modeling it in the form of documents based on business needs and system requirements. In SPS software, the requirements phase produces several artifacts based on both business and system including, a list of actors, use case lists, use case details, activity diagrams, use case diagrams, use case surveys, and user interface sketches.

- ***Analysis:*** The analysis phase decomposes a set of complex requirements into important elements and their relationships that form the basis of the solutions made. Docherty explained in modeling the analysis, there are two models, namely static models using class diagrams and dynamic using communication diagrams. In SPS software, the analysis phase produces two artifacts, namely class diagram and communication diagram.

- ***Design:*** The design phase is the stage of finding solutions to existing problems. The design phase determines certain technology choices such as programming languages, protocols, and the database management system used. Design can be divided into two main activities, namely system design, and subsystem design. System design describes the system into logical and physical components with particular attention paid to network topology. In SPS software, the design phase produces several artifacts, namely, deployment diagrams, class diagrams, sequence diagrams, and database schemes.

- ***ClassSpecification:*** The class specification is a clear and unambiguous description in showing how the behavior of a component in software and how it should be used. In SPS software, the class specification phase produces an artifact that is a comment written in the application source code so that it can provide a clear description and function of a software component.

- ***Implementation:*** Writing code fragments that will work together to form subsystems which can then turn collaborating into systems as a whole, is an understanding of the implementation phase. In the SPS software, the implementation phase produces three artifacts including, source code, database implementation, and user interface implementation.

- ***Testing:*** The testing phase is the phase where when the software has been completed, testing is needed to test whether the system requirements are following its main purpose. In SPS software, the testing phase produces a test report artifact.

- **Result and discussion:**

In this chapter, the Smart Parking Systems development process will be briefly explained by showing some of the diagrams made followed by software test results.

- ***Use Cases:***

Use case diagrams in the Smart Parking System application both web apps and mobile apps consist of 2 external actors namely user and admin who interact with 10 use cases in the system. User actors interact with 3 use cases, while admin actors with all existing use cases. Use case diagrams for the Smart Parking System application are shown in Figure 3.

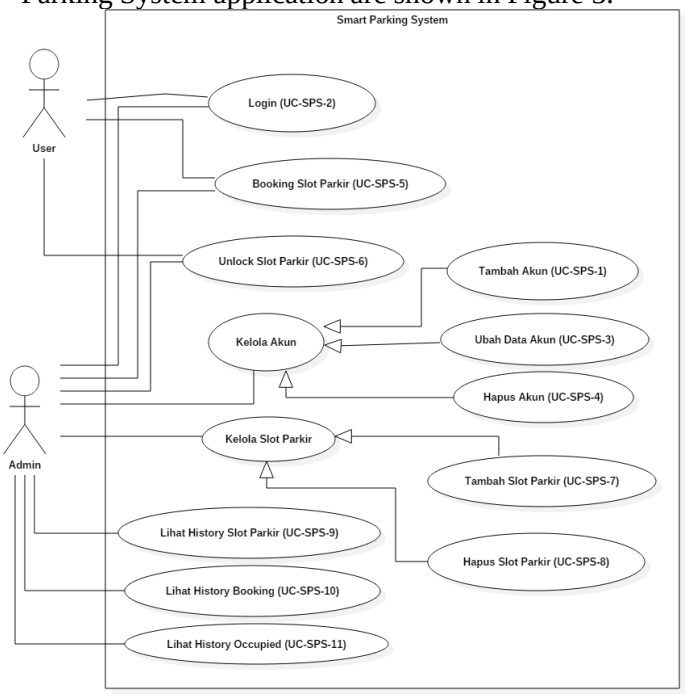

Figure Smart parking system use case diagram

- ***Class diagram:***

Smart Parking System software requirements that have been described in the requirements phase describe the external communication between the actor and the system. At this stage, an analysis of the internal communication of the system is based on the use case that has been made to determine what classes are needed by the system along with the attributes and operations needed for each class. Class Diagram Design Smart Parking System for web apps and mobile apps is shown in Figure 4. Smart Parking System in its implementation uses the Django framework, to facilitate understanding of existing classes, in its depiction the naming model is used according to the Django structure, namely Model-View -Template.

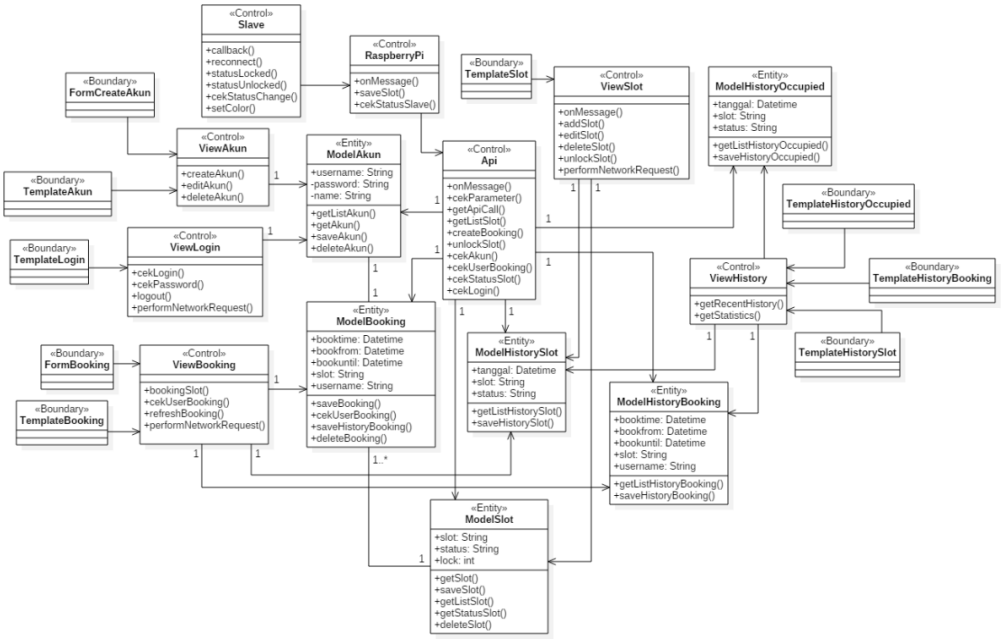

Figure Smart parking system class diagram design

- ***Deployment diagram:***

The Smart Parking System as a whole is built on three subsystems that run in parallel namely web applications, mobile applications, and embedded systems. The concept of IoT on the Smart Parking System is applied to the communication lane between the application and the embedded system which enables the embedded system to be controlled remotely through the application. Applications with different platforms require subsystems to communicate through networks with special protocols. The lightweight communication protocol used in the Smart Parking System is MQTT. The web application has two execution environments namely API and the webserver.

The API functions as a communication bridge between mobile apps and embedded systems, and the webserver functions as a controller that manages data. Web applications can only be accessed by the admin. Mobile applications have their network protocol components that allow users to carry out activities in applications with internet networks. Through the network protocol, mobile applications make requests to the webserver API. The main database in the Smart Parking System, SPSDB, was built with a MySQL engine that was embedded on a web server computer. The main database is supported by a mini database that is a mini DB that was built using the MariaDB engine embedded in RaspberryPi to cope if a connection is lost between the web apps and the embedded system so that the embedded system can continue to run only with the local network. The embedded system in the Smart Parking System is built with a combination of RaspberryPi as a master and the ESP8266 wifi module as a slave.

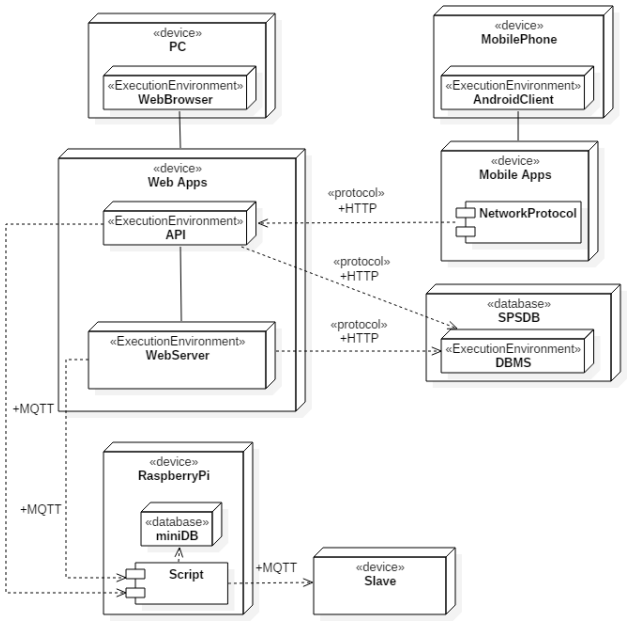

Figure Smart parking system deployment diagram

- ***Testing result:***

The testing phase or the testing phase is the phase when the software that is made has been integrated and completed into one whole system so that further testing is needed to test the feasibility of the system if it is following the requirements specified earlier. In the testing phase, the results of software testing are given using the black box testing method. Blackbox testing is carried out by one examiner with a test case which is described in Table 2. Conclusion of the test results on the Smart Parking System application both web apps and mobile apps for all given test cases successfully run or received.

Table Blackbox testing results for smart parking system

Test Case ID	Use Case	Test Case	Results
TC-SPS-1	Access to the Add New User Account page by pressing the Create Account button		Accepted
TC-SPS-2		Fill out the new account data form withinvalid input then press the submit button	Accepted
TC-SPS-3	Add account		

	Fill out the new account data form withvalid input then press the submit button	
Accepted		
TC-SPS-4	Do not fill out the form then press thesubmit button	
TC-SPS-5	Fill in the username that already exists inthe database	
Accepted		
Accepted		
TC-SPS-6	Access to the SPS web login page	Accepted
TC-SPS-7	Fill in a valid password then press the loginbutton	
Accepted		
TC-SPS-8		
	Login	
	Fill in the invalid password then press thelogin button	
Accepted		
TC-SPS-9	Access to the SPS mobile login page	Accepted
TC-SPS-10	Fill in the login data that is valid usernameand password then press the login button	
Accepted		
TC-SPS-11		
	Fill in the login data that is an invalid username and password then press the loginbutton	
Accepted		
TC-SPS-12		

	Access to the manage users page	AcceptedAccess the account edit page by pressing
	TC-SPS-13	
Changeaccount		
the edit button on the account you want tochange		
Accepted		
	TC-SPS-14	Fill out the changes in account data thenpress the save button
Accepted		
	TC-SPS-15	Access to the manage users page Accepted
	TC-SPS-16	Press the delete button on the account youwant to delete
Accepted		
	TC-SPS-17	
Delete account		
Press confirm on the modal warning foraccount deletion		
Accepted		
	TC-SPS-18	Pressing the cancel button on the modalwarning for account deletion
Accepted		
	TC-SPS-19	Access to the SPS web booking page Accepted
	TC-SPS-20	Fill in the booking detail form with invaliddata then press the submit button
	TC-SPS-21	Fill in the booking detail form with validdata then press the submit button
	TC-SPS-22	Do not fill in the booking detail form thenpress the submit button
		Fill in the parking slot input on the booking
AcceptedAcceptedAccepted		

TC-SPS-23

TC-SPS-24

Parking slotbooking

detail form with the slot being booked and fill in other input data then press the submitbutton
Access to the SPS mobile booking page bypressing the booking button on the slot youwant to book

Accepted

Accepted

TC-SPS-25	Fill in the booking detail form with invaliddata then press the submit button
TC-SPS-26	Fill in the booking detail form with validdata then press the submit button
TC-SPS-27	Do not fill in the booking detail form thenpress the submit button
TC-SPS-28	Press the booking button on the parking slotthat is being booked Pressing the booking button on a particular

AcceptedAcceptedAcceptedAccepted

TC-SPS-29

TC-SPS-30TC-SPS-31

TC-SPS-32

TC-SPS-33TC-SPS-34

Add parkingslot

parking slot when the user is having anactive booking

Press the unlock button on the parking slotwith the booking status you want to unlockPress the unlock button on the parking slotwith the parking slot status unlocked Pressing the unlock button on the parkingslot with the booking status but does notbelong to the user
Access to the parking slot addition page bypressing the Create Slot button on the web parking slot list page
Fill out the form for adding a new parking slot with invalid data then press the save button

Accepted

AcceptedAccepted

Accepted

AcceptedAccepted

TC-SPS-35		Fill in the form for adding a new parking slot with valid data then press the save button	Accepted
TC-SPS-36	Delete parking slot	Press the delete button on the parking slot you want to delete on the web parking slot list page	Accepted
TC-SPS-37		Press the confirm button on the modal warning for deleting the parking slot	Accepted
TC-SPS-38		Pressing the cancel button on the modal warning deleting the parking slot	Accepted
TC-SPS-39	Parking slot View History	Access the history slot page by pressing the history slot button on the historic main page	Accepted
TC-SPS-40	Booking view History	Access the booking history page by pressing the booking history button on the main history page	Accepted
TC-SPS-41	Occupied view History	Access to the occupied history page by pressing the occupied history button on the main history page	Accepted

- **Conclusion:**

In this paper, we showed a summary development process of the Smart Parking System based on the Internet of Things using the Object-Oriented Analysis and Design (OOAD) method. Smart Parking System's software development using OOAD provides convenience in the process because of the framework stated that gives us ease by showing each step in developing software. Smart Parking System development was tested by black-box testing resulting in a perfect score for 41 test cases given. By this study, we are hoping that Smart Cities' topic will be one of the best solutions to various problems in real life that we face today.