

Design and Development of Mobile Application for Parking Reservation System

Ray Adderley JM. Gining^{1*}, S.S.M. Fauzi¹, Muhammad Nabil Fikri Jamaluddin¹, Muhammad Akram Bin Marizalee¹, Tajul Rosli Razak¹, Anis Afiqah Sharip²

¹Faculty of Computer Science and Mathematics, Universiti Teknologi MARA, Perlis Branch, Arau Campus, Malaysia

²Faculty of Computer Science and Mathematics, Universiti Teknologi MARA, Melaka Branch, Jasin Campus, Malaysia

Corresponding author: ray_adderley@uitm.edu.my

Received Date: 31 October 2019

Accepted Date: 11 November 2019

ABSTRACT

Every emerging country face the issue of public parking congestion in the urban area. Ignoring the issue would breed a series of problems; from traffic disruption to violating the laws related to traffic. In Malaysia, the City Council has made an effort to mitigate the problem by using a coupon - which largely handled manually by the user by purchasing and paying the coupon through a machine. However, the procedure of using a coupon is considered inefficient due to challenges such as; extensive time, coupon exploitation and material waste. Therefore, to mitigate the problem surfaced in the current procedure, a mobile application is developed to handle the parking reservation process. The reservation process included the function to view the parking's availability thus presented the user the advantage of reserving parking earlier in the case of parking usage are at the peak. This paper highlighted the requirement identification and design and development phases.

Keywords: *Urbanization, city management, public parking, mobile applications, mobile technology*

INTRODUCTION

Urbanization is not only referring to a modern phenomenon but also population growth in an urban area. The matter of economic crisis causes people to change their directions of living (Nations & Introduction, 2010), consequently, they moved from rural to the urban area for job opportunities. A high-density population in the urban area comes with a different perspective (Brenner, 2013), blooming business activities and searing infrastructure. These positive impacts come with drawbacks - such as parking congestion. Parking congestion or the limitation of parking spaces for public populace is a well-known issue in the crisis of urbanization (Liu et al., 2012).

In the context of parking spaces, in Perlis, a smaller state in Peninsular Malaysia also confronted with the parking congestion problem. The City Council implemented the use of parking coupon for the parking space management. Although, it is considered to be inefficient since the public can break the parking rules and nonchalantly abusing it (Fisman & Miguel, 2007) - such as using the 30 minutes coupon over its time limit. These have become a norm and if this issue were neglected, it will affect the economy as well as the rights of the public. Furthermore, the process to find an empty parking space is frustrating because the drivers need to take a detour and wait for a parked car to release a parking space (Yan, Yang, Rawat, & Olariu, 2010).

Thus, there is a need for research to find a solution to solve or at least mitigate the issue. With the emergence of mobile technologies, which allow a mobile device to communicate with each other (Rico, Sancho, Cendon, & Camus, 2013), the application of this technology can contribute towards exploring the solutions to solve the parking management issue.

The objective of this paper is to develop a prototype of Android mobile application which serves as the means for parking reservation management. The prototype would be able to reduce the time consumed in reserving empty parking spaces for people to use, mitigating the issues of the current coupon purchase procedure.

This paper is organized as follows; the first section introduces the motivation and objective of the paper. Some related works described in the second section, followed by the discussion about the methodology in the third section. In the fourth section, the result and discussion are presented and the paper is concluded in the final section.

RELATED WORKS

Manual Coupon

The first approach is the manual method which is used by the City Council in Perlis. The City Council produces a physical coupon which is sold to the public for parking reservation. It needs to be displayed clearly on a parked car for City Council staff observational purposes - usually on the car front dashboard. This manual method had issues with user ethics, parking management, and resources wastes. Buying the coupon designates the right for the buyer to use a parking space in the covered area. Although, there are cases which the buyer uses a used coupon to pay for their parking use, which indicates the problem of user ethics. However, the City Council combated this issue by sending their staff to check for cars that parked illegally by manually checking the detail of the coupon displayed on the car dashboard. Nonetheless, the solution is poor as it is time-consuming and wasting labour resources. Another issue with resources is the coupon itself, which uses paper as its material and needed to be mass-produced. On the side of the parking user, aside for ethical issues, it is time-consuming for them to search for empty parking to use, and they also need to use a physical object to scratch open the coupon every time they needed to apply it.

Padiyo Mobile Apps



Figure 1: Padiyo Mobile Apps

PADIYO mobile application allows car owners to pay their parking fees via the apps. The primary issue of searching for coupon machine is eliminated. Even though the user is in an unfamiliar area, the information about the parking spaces will be identified automatically by the apps - which includes the supported local council, the area, and charges of the parking space they wanted to use. Additional features such as a reminder for parking validity is also included. The use of e-Wallet technology making sure it is easy for the user to convert their money into tokens which are used to the parking payment. (“Padiyo | PARK PAY HASSLE-FREE,” n.d.).

Parking.sg Mobile Apps

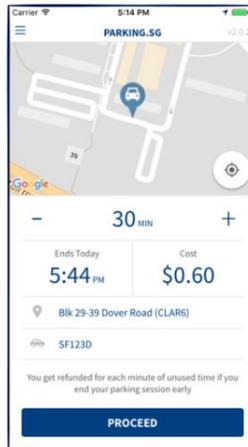


Figure 2: Parking.sg Mobile Apps

The third related work is Parking.sg, a mobile app which provides the features for its user to pay for their short-term parking charges. This app can be used in conjunction with the existing coupon-based public car parking spaces - it serves as a better alternative for the user for parking charges payment. Information such as vehicle number, the parking space, parking duration, and payment using debit or credit card are needed before using the parking space. Parking period extension is also possible through using the app.

Comparison of Related Works

Table 1: Related works comparison

	Physical object	Internet Connectivity	Login System	Law enforcing monitoring	Parking allocation availability
Coupon Parking	✓	X	X	✓	X
Padiyo	✓	✓	✓	✓	X
Parking.SG	✓	✓	X	✓	X

least efficient since it only provides two of the listed features. The prototype for this project, My Public Parking (MPP) mobile application, would not resolve all the problem faced by these related works. Rather, it focuses to mitigate the problem faced by the Perlis City Council concerning with the public

parking spaces. The Perlis City Council is currently using manual coupon parking which is described in the related works. MPP provides the feature of a parking reservation and a real-time monitoring feature that shows the availability of the parking spaces. Some limitation to MPP includes; authority was unable to use it for monitoring and payment purposes yet. Nevertheless, the objective of this prototype is to provide a means for parking users to plan their journey so that it can save them time in searching for parking at their destination, which in the same time could save material resources from producing physical coupon results to wastes.

METHODOLOGY

The figure below shows the summarization of the applied research methodology. The first phase focuses on gathering information to develop the prototype - Requirement Identification. Followed by the second phase that based on the previous requirement, a design is created and the prototype is built based on the design. In this section, only two of the methodology are discussed; requirement identification and design and development.

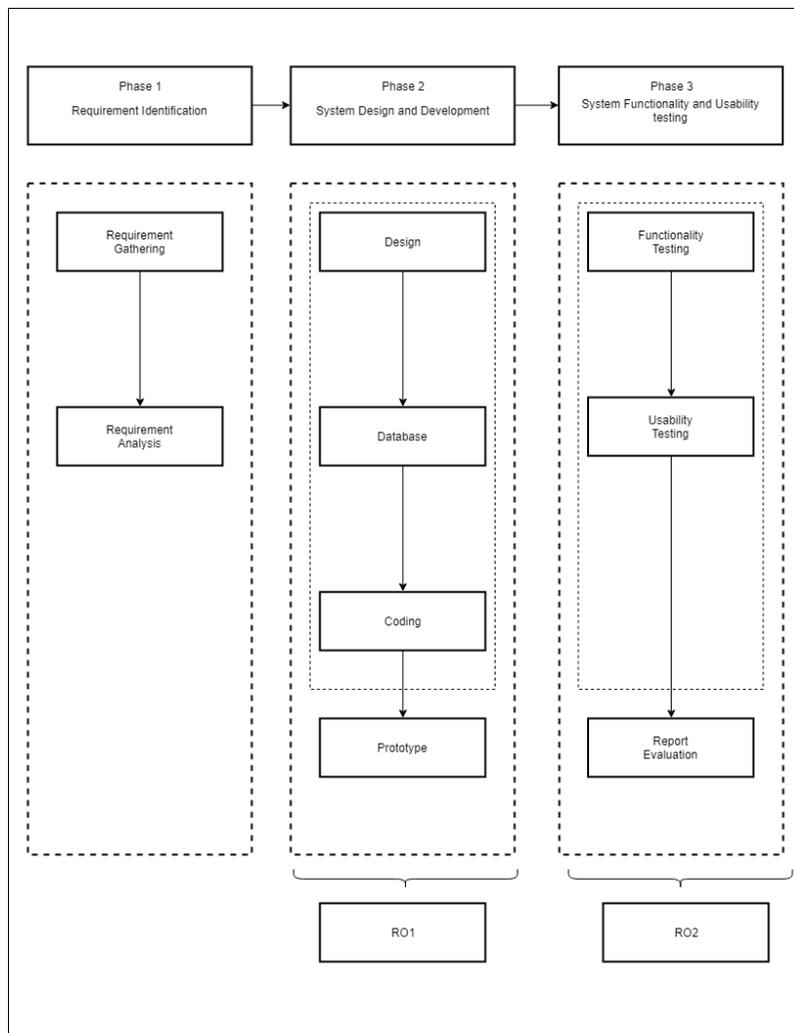


Figure 3: Methodology (based on Prototyping Model)

Phase 1: Requirement Identification

Requirement identification phases were introductory research in discovering the needed details which act as an affirmation of the requirements before developing the application.

There are two sub-phases in this phase; 1) requirement gathering where the data of the user requirements were collected by conducting two main activities; interviews and observation, and 2) requirement analysis, where the gathered requirement is translated into useful diagrams and informative tables that mainly assist the developer to visualise, document, specify and construct the artefacts of the application to be developed.

Phase 2: Design and Development

The second phase is a crucial phase which focuses on the design and the development of the system based on the collected requirements on the previous phase.

This phase contained two sub-phases; 1) system design - designing the interface by using Android Studio, the database by using NoSQL with the support of Google Firebase, and 2) system development - which mostly consists of the coding process to implement the functionality indicated in the first phase, thus producing a complete mobile application prototype.

FINDINGS AND DISCUSSION

The results of the interview indicated the target users highly preferred a platform that supports mobility, thus a mobile application is a better choice as it fulfils the user requirement. Other than that, the mobile apps requirements are further derived from conducting interviews with the target users - random citizen with cars, and observation. The interview consists of questions that need to be answered, with some suggestions from the users. Meanwhile, the observation is carried out by monitoring and recording the detailed process of parking in Arau, Perlis parking spaces. Through the review of the information gathered from the interview and observation session, the requirement for mobile application is determined. The requirements are divided into categories based on these priorities:

- i. M – Mandatory requirements (something the system must do)
- ii. D – Desirable requirements (something the system preferably should do)
- iii. O – Optional requirements (something the system may do)

The table below shows the requirement list obtained from the two performed activities.

Table 2: Requirement List

A. FUNCTIONAL REQUIREMENTS			
No	Requirement ID	Requirement Description	Priority
	MPP_01	Authentication System (User)	
1.	MPP_01_01	New user register as user by providing the information.	M
2.	MPP_01_02	User must enter email and password to login.	M
3.	MPP_01_03	Error message will alert user when login using invalid email or password, or both.	D
4.	MPP_01_04	Invalid email format message appears when register wrong format.	D
5.	MPP_01_05	Invalid password format message appears when the user registers the	D

		wrong format.	
6.	MP_01_06	User will be redirected to the user's profile after logged in or registered.	D

These requirements are analysed and translated into modelling tools diagrams; Unified Modelling Language (UML) diagram which consists of the Class Diagram, Use Case Diagram and Activity Diagram.

	MPP_02	User Profile	
7.	MPP_02_01	User is able to edit their personal information.	M
8.	MPP_02_02	User's information should be displayed at the user's profile page.	M
9.	MPP_02_03	User is able to upload or change their profile picture.	D
10.	MPP_02_04	The application should be able to detect the current user when they close the application without signing out.	D
11.	MPP_02_05	There is a back button in the edit profile section to go to the previous page.	D
12.	MPP_02_06	User is able to change their email and password for authentication.	D
	MPP_03	Browse Parking	
13.	MPP_03_01	User is able to reserve the button or parking slot.	M
14.	MPP_03_02	User is able to check out the button from disable.	M
15.	MPP_03_03	User is able to select parking area.	D
16.	MPP_03_04	Alert message should notify the user before they confirm the reserve or the checkout button.	D
17.	MPP_03_05	Another user is able to notify the current reserve slot user for their next booking.	D
18.	MPP_03_05	There is a timer showing the current time left for the current slot.	D
19.	MPP_03_06	The slot will alert the user that time is running out.	D
	MPP_04	Parking History	
20.	MPP_04_01	User is able to view their reserve history.	M
21.	MPP_04_02	The history should sort to the latest reserve parking history.	M
22.	MPP_04_03	User is able to delete the history.	D
23.	MPP_04_04	User is able to search the specific reserve history.	O
24.	MPP_04_05	History section should display understandable time of reserved parking.	O

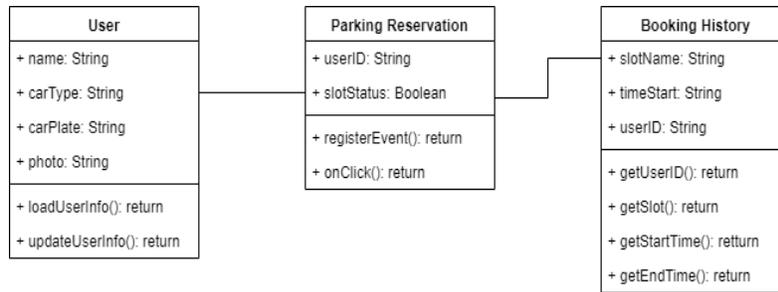


Figure 4: Class Diagram

The figure above shows the class diagram of the system. The diagram showed a static view of the system. The class diagrams are widely used to directly mapped with object-oriented languages, it is used during the system construction period.

The figure below shows the use case diagram of the system. The diagram shows two actors, first the Arau Communities which consist of people who uses public parking, and the admin actor who manages the data for the mobile application. The cases shown in the diagram are the requirement summarised into a simple case. The detailed activity for each case are translated in the activity diagram.

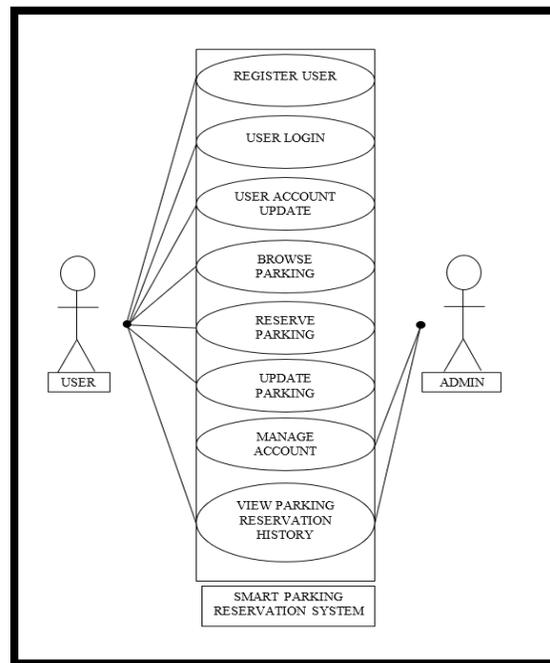


Figure 5: Use Case Diagram

The Activity Diagram shows the detailed process for both actors interaction with the system. For the sake of simplicity, only one Activity Diagram is included in this paper which is the registration process below.

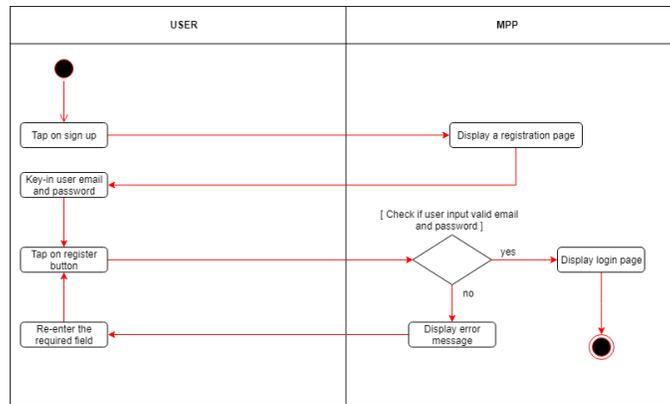


Figure 6: Activity Diagram for Registration Process

In the design and development phase, the activity started with user interface design. User interface (UI) design is the process of making interfaces in software or computerized devices with focused looks or style. Therefore, UI for mobile application is the graphical and usually touch-sensitive display on a mobile device. In addition to this matter, mobile devices have smaller screen size and touch-screen control. Hence, mobile UI needs to be simple and less in text field usage.

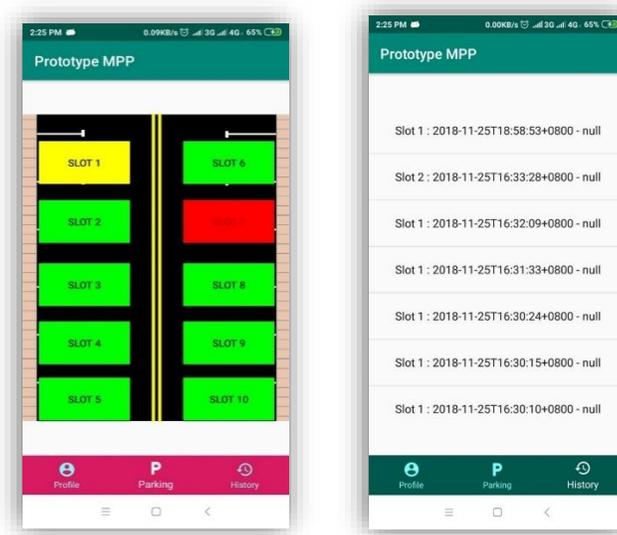


Figure 7: Parking Browser and Parking History page

Two figures above are the parking browser and parking history page. In the parking browser page, the user can view the empty or occupied parking. Green color indicates empty parking, while red is reserved by other users. Yellow color indicates the current user themselves. In the parking history, user’s reservation history is shown. In this interface, the setting to show the parking history starts with the parking slot number. Then, it is followed by the date with a format of “YYYY-MM-DD”. Next is the time the user starts parking which is in the format of 24 hour “HH:MM:SS” as the display, and the time for parking checkout follows right after. The value null means, the car is not checked out yet.

For the database of the mobile application, Firebase will be used. Firebase provides a realtime database and backend as a service. The service provides the application developers with an Application Programming Interface (API) that allows the application data to be synchronized across clients and stored on firebase's cloud service.

Moreover, the firebase is a NoSQL database, a big JSON object where its user can store whatever they want inside it. Moreover, providing easier access and storing, firebase JSON uses no schema for the database, no tables or columns, but with only the combinations of key or value or both. The data sample for the mobile apps shown in JSON format is shown in the figure 9.

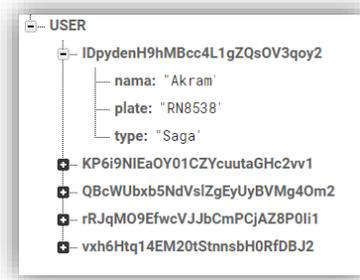


Figure 8: Firebase User table content

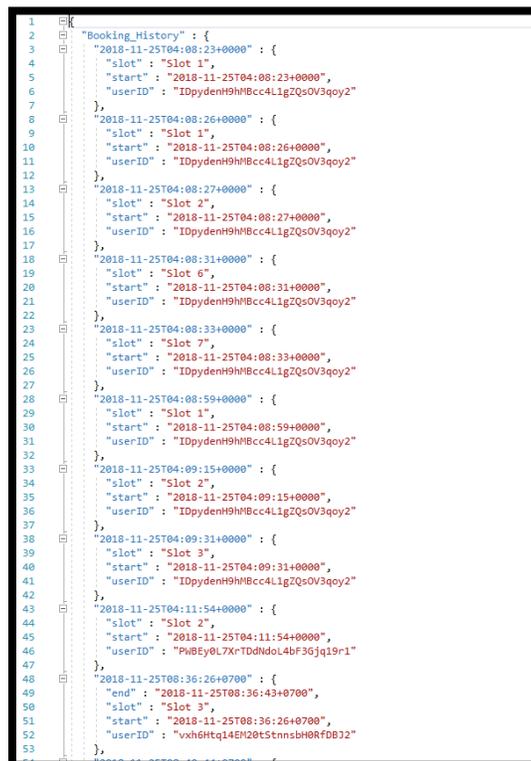


Figure 9: JSON data sample

Android Studio is the software used mostly for the development process. The software support C++, Java and Kotlin programming languages. In this prototype development, the programming languages used is Java.

In this paper, the development process highlighted is the User-sign in functionality. User authentication used for the system is provided by firebase API, integrated with the Android Studio codes. By using their API, the user needs to enter valid email and password where the minimum length is 6 characters.

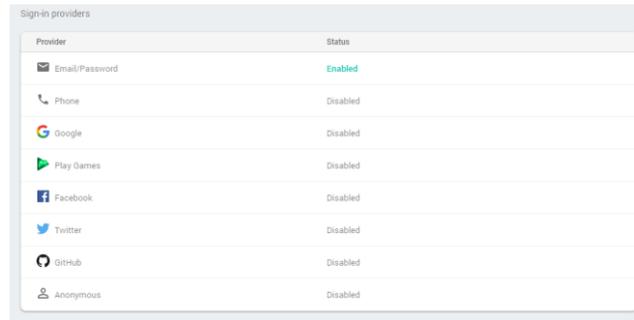


Figure 10: Firebase sign-in method API

Figure below shows the registration process. Besides the email icon, the user also needs to key-in their valid email. Then, below the email is where the user keys-in the password. In addition, for security purposes, each password character typed will be replaced with symbol “*”.

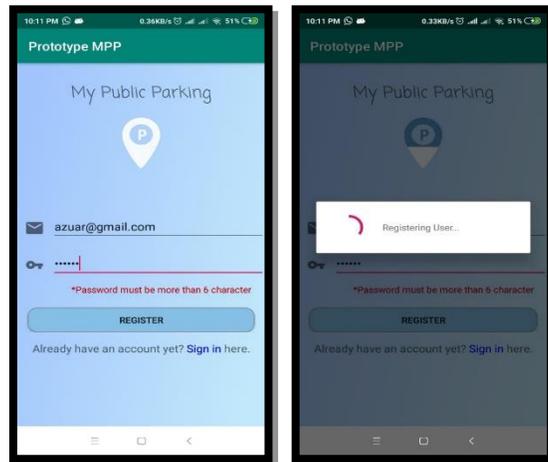
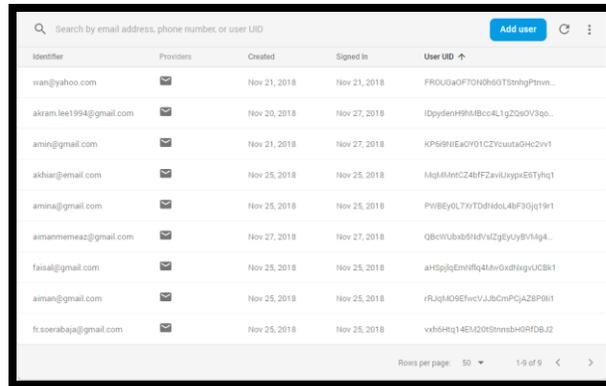


Figure 11: Registration process

Next, after the user inserts their email and password and tapping on the register button, a progress dialog will appear to notify the user that the system is registering them as a user. Hence, the user now has the authentication to use the system as showed in figure below – interface is provided by the Firebase API.



Identifier	Providers	Created	Signed In	User UID
wan@yahoo.com		Nov 21, 2018	Nov 21, 2018	FROU0aOF7ONh60T5tnhgPm...
akram.lee1994@gmail.com		Nov 20, 2018	Nov 27, 2018	IDPpydenH8HmBoc4L1gZ0aOv3qo...
amin@gmail.com		Nov 21, 2018	Nov 27, 2018	KP69HIEaOY01CZYcousaHic2vv1
akhsa@email.com		Nov 25, 2018	Nov 25, 2018	MqjMteC24bFZavUxyx5E5Yhq1
amina@gmail.com		Nov 25, 2018	Nov 25, 2018	PWBEy0L7xTDDhdol4bF3jg19r1
aminamemeaz@gmail.com		Nov 27, 2018	Nov 27, 2018	QBcWUxb68hdvaZygyUy9VAg4...
faisal@gmail.com		Nov 25, 2018	Nov 25, 2018	aHSpjgEmfHq4kxvDxgUCBk1
amin@gmail.com		Nov 25, 2018	Nov 25, 2018	rRujkI09EhewVJbCmPcJAZ2P8h1
fr.soerabaja@gmail.com		Nov 25, 2018	Nov 25, 2018	vsh6Hq145X205tmbsH9RF06J2

Figure 12: Firebase API authenticated users

CONCLUSION AND RECOMMENDATION

This paper described a study on the design and development of the mobile application for parking reservation system. It has two phases which is on requirement identification and design and development. These two phases consist of several activities that are suitable for an application development. Modeling tools and software used are emphasised in the result and discussion. The phases implemented in the methodology could be applied in any project involving a development process.

REFERENCES

- Brenner, N. (2013). Theses on Urbanization. *Public Culture*, 25(1 69), 85–114. <https://doi.org/10.1215/08992363-1890477>
- Fisman, R., & Miguel, E. (2007). Corruption, Norms, and Legal Enforcement: Evidence from Diplomatic Parking Tickets. *Journal of Political Economy*, 115(6), 1020–1048. <https://doi.org/10.1086/527495>
- Liu, Y., Wang, W., Ding, C., Guo, H., Guo, W., Yao, L., ... Tan, H. (2012). Metropolis parking problems and management planning solutions for traffic operation effectiveness. *Mathematical Problems in Engineering*, 2012. <https://doi.org/10.1155/2012/678952>
- Nations, U., & Introduction, I. (2010). *Economic and Social Council*, 03074(July), 1–8. <https://doi.org/10.1017/S0020818300006640>
- Padiyo, (n.d.). Park pay hassle free. Retrieved May 4, 2018, from <https://padiyo.net/site/>
- Rico, J., Sancho, J., Cendon, B., & Camus, M. (2013). Parking easier by using context information of a smart city: Enabling fast search and management of parking resources. *Proceedings - 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013*, 1380–1385. <https://doi.org/10.1109/WAINA.2013.150>
- Yan, G., Yang, W., Rawat, D. B., & Olariu, S. (2010). *Smart Parking: A Secure and Intelligent Parking System*, (April 2011), 18–30.