UTEM EASY PARK
(UTEM PARKING RESERVATION WEB-BASED SYSTEM)

FATIN HANISAH BINTI ABD MALEK

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

DECLARATION

I hereby declare that this project report entitled

UTEM EASY PARK
(UTEM PARKING RESERVATION WEB-BASED SYSTEM)

is written by me and is my own effort and no

part has been plagiarizedwithout citations.

STUDENT: _____ Date : 20/09/2023

(FATIN HANISAH BT ABD MALEK)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR: _____ Date : 20/09/2023

(TS. ABDUL RAZAK BIN HUSSAIN)

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my supervisor TS. Abdul Razak Bin Hussain who had invaluable assistance and guidance throughout the completion of this project. His expertise and support have played a significant role in its success. I am truly grateful for his constant time, patience, and willingness to answer my questions and provide valuable insights.

I would also like to extend my deepest appreciation to my beloved parents. Their unwavering support, encouragement, and motivation have been instrumental in my journey towards completing this project. Their belief in my abilities and their continuous encouragement pushed me to overcome challenges and strive for excellence. Their presence and understanding during the ups and downs of this project were invaluable.

In addition, I would like to acknowledge the contributions of my parents, siblings, my supportive fiancé, housemates and my friends who guide me throughout along this beautiful journey, give valuable feedback and suggestions during the development of this UTeM EasyPark (UTeM Parking Reservation Web-Based System) project.

Lastly, I would like to express my gratitude to all the researchers, authors, and contributors whose work and publications I consulted throughout the course of this project. Their knowledge and insights have been vital in shaping my understanding of the subject matter and guiding my research.

I am deeply thankful to all those mentioned above, as well as anyone else who has directly or indirectly supported me in completing this project. Theirs assistance and encouragement have been invaluable, and I am truly grateful for your contributions to my success.

ABSTRACT

The "UTeM Parking Reservation Web-Based System" is a proposed technological solution designed to streamline parking reservation processes at Universiti Teknikal Malaysia Melaka (UTeM). The current manual system poses significant inefficiencies and challenges, prompting the need for a more organized and efficient approach. This web-based system aims to provide a convenient platform for UTeM's staff, lecturers, and related authorities to reserve parking spaces for specific events, minimizing the time and effort spent on this task. The system offers features such as tracking reservation details, including vehicle information, preferred location, and contact details. Moreover, it can monitor parking lot availability across various UTeM buildings and faculties. The existing manual system's limitations, including redundancy and lack of standardized processes, underscore the necessity of an automated, centralized reservation system. The proposed solution targets efficient parking space utilization, reduced congestion, increased user satisfaction, and improved data management. The key modules of the system include login, applicant, officer, parking reservation, approving, and reporting modules. The web-based system is expected to enhance the convenience and effectiveness of parking facilities at UTeM, benefiting both the applicant and the administrative staff while providing better data management through integration with Microsoft SQL Server. This thesis comprehensively explores the development and implementation of the UTeM Parking Reservation Web-Based System, addressing the identified problems and envisioning a more efficient parking reservation process.

ABSTRAK

"UTeM Parking Reservation Web-Based System" adalah cadangan penyelesaian teknologi yang direka untuk menyusun semula proses tempahan tempat letak kenderaan di Universiti Teknikal Malaysia Melaka (UTeM). Sistem manual semasa menimbulkan keberkesanan dan cabaran yang signifikan, memotivasi keperluan untuk pendekatan yang lebih teratur dan efisien. Sistem berasaskan web ini bertujuan untuk menyediakan platform yang mudah digunakan bagi kakitangan UTeM, pensyarah, dan pihak berkenaan untuk mereservasi tempat letak kenderaan untuk acara tertentu, meminimumkan masa dan usaha yang dihabiskan untuk tugas ini. Sistem ini menawarkan ciri-ciri seperti penjejakkan butiran tempahan, termasuk maklumat kenderaan, lokasi pilihan, dan maklumat hubungan. Selain itu, ia dapat memantau ketersediaan tempat letak kenderaan di pelbagai bangunan dan fakulti UTeM. Kelemahan sistem manual sedia ada, termasuk pengulangan dan kurangnya proses piawai, menandakan keperluan sistem tempahan terpusat yang berautomasi. Penyelesaian yang dicadangkan bertujuan untuk penggunaan ruang letak kenderaan yang efisyen, pengurangan kesesakan, peningkatan kepuasan pengguna, dan peningkatan pengurusan data. Modul utama sistem ini termasuk modul log masuk, pemohon, pegawai, tempahan tempat letak kenderaan, pengesahan, dan pelaporan. Dijangkakan bahawa sistem berasaskan web ini akan meningkatkan keselesaan dan keberkesanan kemudahan letak kenderaan di UTeM, memberi manfaat kepada pemohon dan kakitangan pentadbiran sambil menyediakan pengurusan data yang lebih baik melalui integrasi dengan Microsoft SQL Server. Tesis ini menyelidiki secara komprehensif pembangunan dan pelaksanaan "UTeM Parking Reservation Web-Based System", menangani masalah yang dikenal pasti dan membayangkan proses tempahan tempat letak kenderaan yang lebih efisien.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

UTeM Parking Reservation Web-Based System is used to ease the lecturers, staff, or authorities to reserve parking for special events around UTeM. This proposed system is also able to track all the process flows including details of reservations such as plate number of the car, type of car, color, preferable location, the number of parking lots intended, name of the applicant, phone number, date and time of reservation. Moreover, this system can trace the number of parking lots that is available in each building or faculty in UTeM.

UTeM Parking Reservation Web-Based System provides institutes staff with an easy way of reserving a parking space for their specific events using the web portal. It overcomes the problem of finding an available parking space around UTeM areas which unnecessarily consumes time. For example, during convocation day it will ease both parties which are the parents and staff who handle the events. The parents will know where to park the car, since there will be arrow signage to guide the parents. Moreover, it also helps to prevent traffic jams and spent more time finding a single parking slot.

By having this proposed system, the staff or person in charge of certain events need to apply or make a reservation through the system and fill in all details about parking reservations and submit the application. The approving staff will get a notification from the system about the reservation made, the approving staff has the authority to verify which are to accept or reject an application. If the application is accepted, then the system will notify Pejabat Keselamatan and provides parking reservation information. Then, it will give all the reservation information to an officer on duty to settle down their job by putting the cone or signage at the requested location. After the officer has completed his/ her task, they will update the status and images as proof to let the applicant knows.

1.2 PROBLEM STATEMENT

The current system is entirely manual and involves a lot of procedures which consumes a lot of time and makes it inefficient to be used. The applicant who wants to make a reservation must fill up a form with their details and the reason for the parking reservation, wait for approval, and require a letter of support from Ketua PTJ. Then, if the status of the parking reservation is approved, the applicant needs to show the form to Pegawai Operasi at Pejabat Keselamatan for the next process. Otherwise, if the status of the parking reservation is rejected for certain reasons, the applicant needs to make another reservation. The issue that might arise from the old system is redundant reservation. For example, if the other user wants to make the same parking reservation on the same date, day, and location it causes a lot of problems. Over that, all the processes using form or paper are highly likely to be lost, fragile, or torn.

Locating a single parking slot or lot for special events might be one of the most frustrating problems that UTeM societies are facing nowadays. For manual parking reservations, it requires a lot of time and energy, and the applicant needs to fill up the form manually about the reservation. The current system is entirely manual and involves a lot of procedures which consumes a lot of time and makes it inefficient to be used. The applicant who wants to make a reservation need to fill up a form with their details and the reason for the parking reservation, and needs to wait for the approval. In the real situation, Pejabat Keselamatan states the main problem with this manual system is unstandardized and they tend to receive parking requests in different reservations such as by informal call, message through WhatsApp, and verbal communication.

All the problems will cause many difficulties such as duplicate requests and overlapping on the same date, or location. Other than that, usually in the old system, the applicant does not know the status of the reservation request whether it has been approved or not. To make things clear, Pejabat Keselamatan needs to call the applicant manually by referring to the form application. Moreover, the other issue received is the old system's unable to produce an appropriate report. Having this report will ease Pejabat Keselamatan to monitor the summary of the parking reservation request for every month.

1.3 OBJECTIVE

A UTeM Parking Reservation Web-Based System aims to improve the efficiency and convenience of parking facilities by allowing individuals to reserve a parking spot in advance. This system can benefit both the applicant which is the UTeM's staff and Pegawai Keselamatan. Some of the key objectives of a UTeM Parking Reservation Web-Based System are:

- **Efficient use of parking space**: A reservation system ensures that parking spaces are utilized more efficiently, reducing the number of vacant parking spots, and minimizing the amount of time that customers spend looking for a parking space.
- **Reduced congestion**: With a reservation system in place, the applicant can plan their arrival time and park their vehicles more quickly, reducing congestion in the parking facility and surrounding areas.
- **Increased satisfaction**: A reservation system provides applicants with the convenience of knowing that they have a guaranteed parking spot, reducing their stress and anxiety and increasing their overall satisfaction with the parking experience.
- **Capability**: This system allows them to book a parking slot that can be made for special events which prevents lecturers, staff, or authorities from wasting too much time making a parking reservation.
- **Convenient**: UTeM Parking Reservation will ease administration to know available spaces of parking lots for each building or faculty. In addition, it also eases Pejabat Keselamatan to monitor the summary of the parking reservation request for every month which offers a more convenient life for the UTeM's society.
- **Better data management**: A parking reservation system can collect data on parking usage, and applicant behavior, which can be used to optimize operations and improve service.

1.4 PROJECT SCOPE

In this proposed UTeM Parking Reservation Web-Based System, the main target users are Staff of UTeM, Lecturers, PTJ (Pusat Tanggungjawab), and Pejabat Keselamatan. Besides that, below are a few modules that will be developed in this system:

- **Login Module**

  This module can be accessed by the applicant and the officer of Pejabat Keselamatan. Since the applicant or the officer are the staff of UTeM where each of the staff has their own unique id. To enter the system, they need to input their id, and the system will verify the id from the database. Therefore, if the entered id is correct, the system allows the user to enter the system.

- **Applicant Module**

  This module can be accessed by the applicant. They can edit their personal information, which the system will retrieve from the database to display in the system. The system allows the applicant to make parking reservations and cancel the reservation. They are also able to receive notification once their application has been approved.

- **The Officer Module**

  This module can be accessed by the officer of Pejabat Keselamatan. They can edit their personal information, which the system will retrieve from the database to display in the system. The system allows the officer to verify the parking reservation request, they can accept or reject the application for certain reasons. The status of an application will be displayed in the system. The officer will get a notification once a parking request is made by the applicant.

- **Parking Reservation Module**

  This module can be accessed by both users applicant and officers. For the applicant, they can make reservations by fill up all the information regarding parking reservations. In addition, the officer is only able to view parking reservation requests and update the status of requests.

- **Approving Module**

  This module can be accessed by the officer, which can view the parking reservation request and update the status of the request.

- **Reporting Module**

  This module can be accessed by the officer who can view the summary of the reservation request report monthly.

1.5 PROJECT SIGNIFICANCE/ EXPECTED OUTPUT

UTeM Parking Reservation is a Web Based System. It allows the applicant, which is the staff of UTeM to reserve parking around UTeM for special events and the system will notify Pejabat Keselamatan about the parking reservation. The officer of Pejabat Keselamatan needs to approve the parking reservation request and update the status and upload photos of parking. The system will respond back to the applicant where their parking reservation has been approved. This UTeM Parking Reservation data will be stored in Microsoft SQL Server.

CHAPTER 2

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 INTRODUCTION

The literature review is a review of a case study. It includes all the facts and findings related to the case study. Research about a case study should include a review of the relevant literature of facts and findings. The review will be the guidelines to develop a good system. In this chapter, it explains the methodology to be used in the project development. The methodology consists of several phases as guidelines that need to be achieved.

2.2 FACTS AND FINDINGS (BASED ON TOPIC)

Fact is a statement or assertion of verified information about something that case or has happened. Moreover, finding is the act of determining the properties of something usually by research or calculation. For this project, facts and findings will be supported by the review of the existing system and technique that is related and already used by others.

2.2.1 Domain

The domain for this project is the parking reservation system. This proposed system needs to be developed because it can ease specifically ease the lecturers, staff, or authorities to reserve parking for special events around UTeM. This proposed system is also able to track all the process flows including details of reservations such as plate number of the car, type of car, color, preferable location, the number of parking lots intended, name of the applicant, phone number, date, and time of reservation. For this purpose, system the appropriate domain is UTeM Management, Staff since they will use the system to make parking reservations around UTeM.

2.2.2 Existing System

There are several existing systems related to parking reservations. The existing system is used to A parking reservation system is a software application or platform that allows users to reserve parking spaces in advance. It helps streamline the process of parking management by providing a convenient way for users to book parking spots and ensuring efficient utilization of available parking spaces.

2.2.2.1 The Development of Mobile Application for Parking Lot Management at the University Campus

This project is a campus parking mobile application project for academic purposes. The aim of this project is to develop a parking system that replaces the current old fashion system that can improve effectiveness when looking for parking spaces, to show the availability of parking spaces, and remind users of their vehicle location. The current parking system on the campus is not effective enough. Therefore, by implementing this project will improve the effectiveness and reduce the time needed to find a parking slot. Due to the different obstacles faced, Quick Response (QR) is the best-suited method to implement into the system. This system will improve student time management and reduce empty parking space time wasting. Other than that, the student data can be gathered from this system which can be further implemented or perform behavior analysis on the user.

Nowadays, it's challenging to locate a parking spot, especially during peak hours, in places like university campuses. The most challenging problem on a university campus is finding a place where all the staff and students can study or work. (Shang, Lin, et al., 2007; Balsas, 2003; Alshuwaikhat et al., 2008). Due to the limited amount of parking spaces available to students, faculty, and staff at the university, it has become more difficult to find open spots. Finding a parking space is therefore always difficult when there are too many cars in a parking lot with little accessible space.

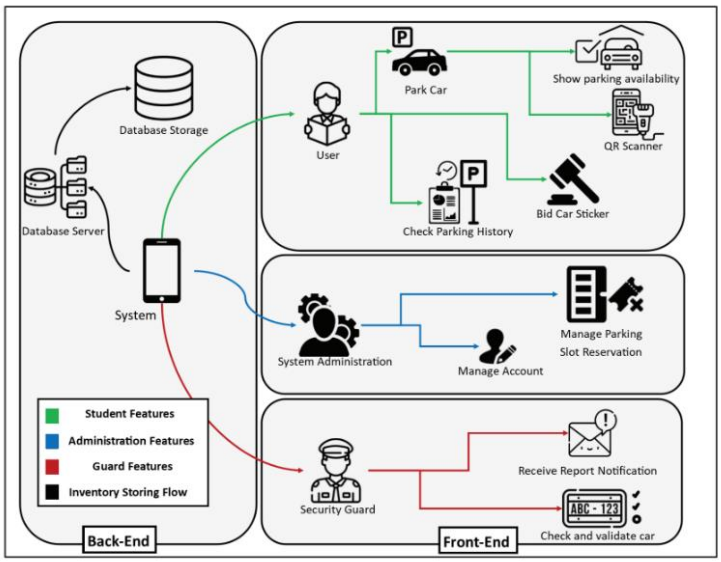


Figure 2.2.2.1 Proposed system framework

2.2.2.2 Smart Parking Management System

Parking lots have always been crucial because they enable drivers to leave their vehicles in a secure manner while going about their daily business. In general, the assistance offered, and the information provided by the smart parking system are highly helpful in aiding the motorist in finding available space. It is now simpler to pay parking fines thanks to the adoption of new technology. To assist in the detection of autos, sensors are utilized. Information regarding parked vehicles is required for creating a smart parking management system, so it is extremely vital. It is simple to gather the data from the sensor so that the system may use it. The driver will also receive the data.

Parking Guidance and Information System (PGIS) can be divided into two distinct features, according to Iyaka Beni. This can be used to monitor specific parking spaces or the entire building (Iyaka Beni, n.d.). The categories are primarily used in large parking lots. The five advantages of the Smart Parking System are essentially the same as those of the Parking Guidance and Information System (PGIS). Making decisions is where there is the most commonality. The system's information aids drivers in making decisions about how to go to their destinations as well as in discovering open parking spaces in the parking lot. Variable message signs were utilized in the Parking Guidance and Information System (PGIS) to instruct drivers on what to do when looking for vacant.

Figure 2.2.2.1 Parking Space Available according to the level

Figure 2.2 shows the placement of various sensors in separate parking spaces to detect the presence of cars. LED lights and sensors are linked together, and then the sensors are placed in each parking space in the parking lot. This is used to indicate that a

parking space has been occupied. Then, according to the different sensors, before outputting the available light spots, the occupancy of each light spot is counted. From the information obtained, the execution process will analyse the information to display accurate information for the driver to view.

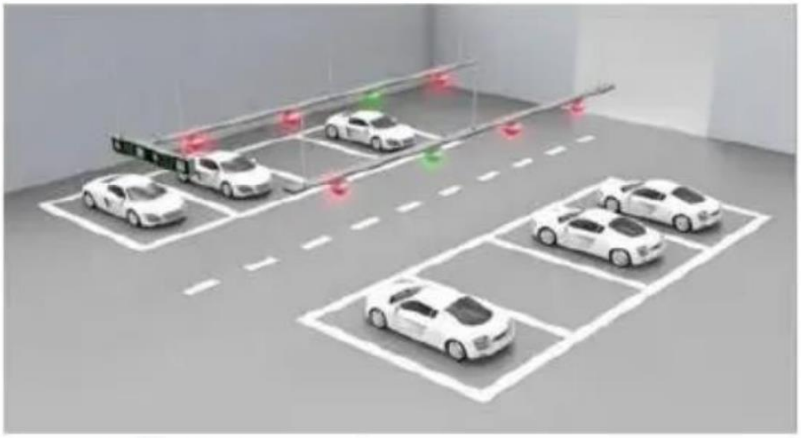


Figure 2.2.2.2 LED Sensor Detection (Iyaka Beni, n.d.)

2.2.2.3 Electronic Parking

According to Iyaka Beni, this is a parking system that allows the driver to select or request the availability of parking spaces. If there are vacancies, they can reserve a parking space to ensure that there will be no parking problems when they arrive at their destination. The electronic parking system allows drivers to reserve parking spaces through various methods such as telephone and online. The advantage of using an electronic parking system is that the driver does not need to find all the trouble of finding a parking space. (Iyaka Beni, n.d.).

Figure 2.2.2.2 Online Reservation System

2.2.3 Technique

Several methodologies may be relevant and linked when looking at different Parking Reservation System approaches. Here are a few instances:

Reservation based on a time slot: Customers are given specified time windows to reserve parking in this method. Customers can select their preferred time slots depending on availability when the system separates the available parking spaces into set time intervals (for example, hourly or daily). This method can be used effectively in situations when customers have predictable parking times because it is reasonably simple to put into practice. Nevertheless, it might not be the best option when clients require greater flexibility or when the length of parking is unknown.

Parking rates can be changed according to demand and supply with dynamic pricing. To calculate the best pricing for various time periods, the system employs algorithms to analyze real-time data such as parking occupancy rates and historical patterns. To control demand, prices are higher during peak hours, while discounts may be provided during off-peak hours to draw customers. By using dynamic pricing, parking resources may be distributed effectively, and income can be maximized. Dynamic pricing algorithms can be difficult to implement and manage, and there may be questions about their fairness and ability to satisfy customers.

2.3 PROJECT METHODOLOGY

According to Wikipedia, methodology the free encyclopedia, methodology can be defined as: (http://en.wikipedia.org/wiki/Methodology)

1. "the analysis of the principles of methods, rules, and postulates employed by a discipline"
2. "the systematic study of methods that are, can be, or have been applied within a discipline" or
3. "a particular procedure or set of procedures".

A methodology is a set of instructions and rules for different stages of the software lifecycle. The technique establishes the framework for a software product's development. There are various methodologies, including:

- Modelling a waterfall
- Spiral
- Incremental Model
- Prototyping

To create a high-quality product and guarantee a smooth development process, it is crucial to employ the appropriate technique. The approach picked must be appropriate for the project being produced. Following considerable investigation and analysis of various approaches, the application will be developed using the conventional waterfall paradigm. The project requirements and their implementation are well understood; hence the waterfall methodology was chosen.

2.3.1 Database Life Cycle

The database life cycle model has six basic steps to be implemented. Figure 6 below shows what are the steps in DBLC and the flow of each step. The DBLC model will be used in developing this project, especially in database development because it will guide the developer to develop the project. Since this project involved a database application, it is the most suitable model that can be used to be implemented.

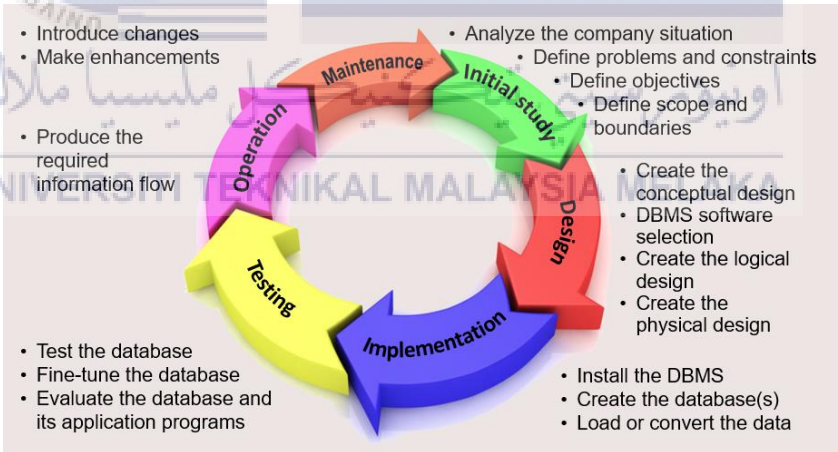


Figure 2.3.1.1 Database Life Cycle Model

2.3.1.1 Database Initial Study

- Analyze the company's situation.
- Define problems and constraints.
- Define objectives.
- Define scope and boundaries.

The initial stage of the database life cycle model is the database initial study. Before analyzing the issues with the corporate system at this level, one must be aware of the business scenario. Constraints in the company must be identified, and the best solutions must be sought in order to address the difficulties and constraints. It is necessary to take note of or pay more attention to the company's goal of establishing a good new system. Therefore, the system's objectives must be specified. When developing a system, it's important to understand the system's scope and bounds, including its users, functions, hardware, and software.

2.3.1.2 Database Design

- Create a conceptual design.
- DBMS software selection
- Create a logical design.
- Create a physical design.

The second level of the Database Life Cycle Model is database design. This level is the most critical DBL phase. Makes sure the final product meets requirements and focuses on data requirements. First, create a conceptual design to know the relationship between entities. At this level also, suitable software must be selected to develop the system. In that case, DBMS software must be chosen because it offers the best requirement to implement the system. Since there are various types of data models that can be used in conceptual design, the designer needs to choose one data model to transform it to its equivalent logical representation. Then, a physical design must be created that includes the coding in DBMS.

2.3.1.3 Implementation and Loading

- Install the DBMS.
- Create the Databases.
- Load or convert the data.

The third level is implementation and loading, in this level, DBMS should be installed in the computer. Follow the instructions step by step to install the DBMS so that it will operate properly in the future. Then, we databases must create databases regarding the system. After creating the databases, can load and convert the data to DBMS language because only DBMS understand programming language. At this level, performance, security, backup and recovery, integrity, company standards, and concurrency controls must be created.

2.3.1.4 Testing and Evaluation

- Test the Database.
- Fine-tune the database.
- Evaluate the database and its application programs.

Testing and evaluation is the fourth level in Database Life Cycle. At this level, test the database to see whether it works correctly or not. For example, must insert, alter, delete, and save data from DBMS. Then, test and fine-tune the database for performance, integrity, concurrent access, and security constraints. Testing and evaluation must be done in parallel with application programming. Lastly, need to evaluate the database and its application programs. If tests fail, must fine-tune based on reference manuals and modified the physical design over again.

2.3.1.5 Operation

- Produce the required information flow.

Operation is the fifth level in Database Life Cycle. At this level, the user must produce the required information flow and starts the process of system evaluation. This step can help users to use the system to manage the business for the company. For example, users can use the order function in the system. Ready to settle unforeseen problems that we may be faced.

2.3.1.5 Maintenance and Evaluation

- Introduce Changes.
- Make enhancement.

The last level of the Database Life Cycle Model is maintenance and evaluation. With the changes, the system will offer better service because the changes can update the function, data, interface, and so on. Enhancements can be made such as to the functions, according to the current situations and requirements that are needed by the user. At this level, it is required to do preventative maintenance, corrective maintenance, adaptive maintenance, and assignment of access permissions.

2.4 PROJECT REQUIREMENTS

During the development of this project, there are several requirements that need to be paid attention to complete this project successfully. The requirement can be the base or main

requirement which is important in this project and by-product requirement. The requirements that are used during the development of this project will be explained below.

2.4.1 Software Requirements

a. Development Tools

- Microsoft Visual Studio 2022

Platform to do the web-based. Easier to code and design interface for the web-based system.

b. Adobe Photoshop CS4

- Software used for editing images for website interface design.

c. PHP (PHP Hypertext Processor)

- PHP is a programming language that allows web developers to create dynamic content that interacts with databases.

d. Draw.io

- Draw.io will be used to design diagram and model of the project.

e. Microsoft Word

- Microsoft Word will be used to write the report of UTeM Parking Reservation system's development.

f. WAMP Server 2.0

- WAMP Server is a form of mini-server that can run on almost any Windows operating system. WAMP includes Apache 2, PHP 5 and MySQL (phpMyAdmin) in its package.

g. Database Management System

- MySQL
- Relational database management system (RDBMS). The program runs as a server providing multi-user access to several databases.

2.4.2 Hardware Requirements

The hardware required is very minimum and can meet client server needs. Perhaps the hardware for the client and server is much better than these minimum requirements. The hardware requirements are as below:

- Laptop

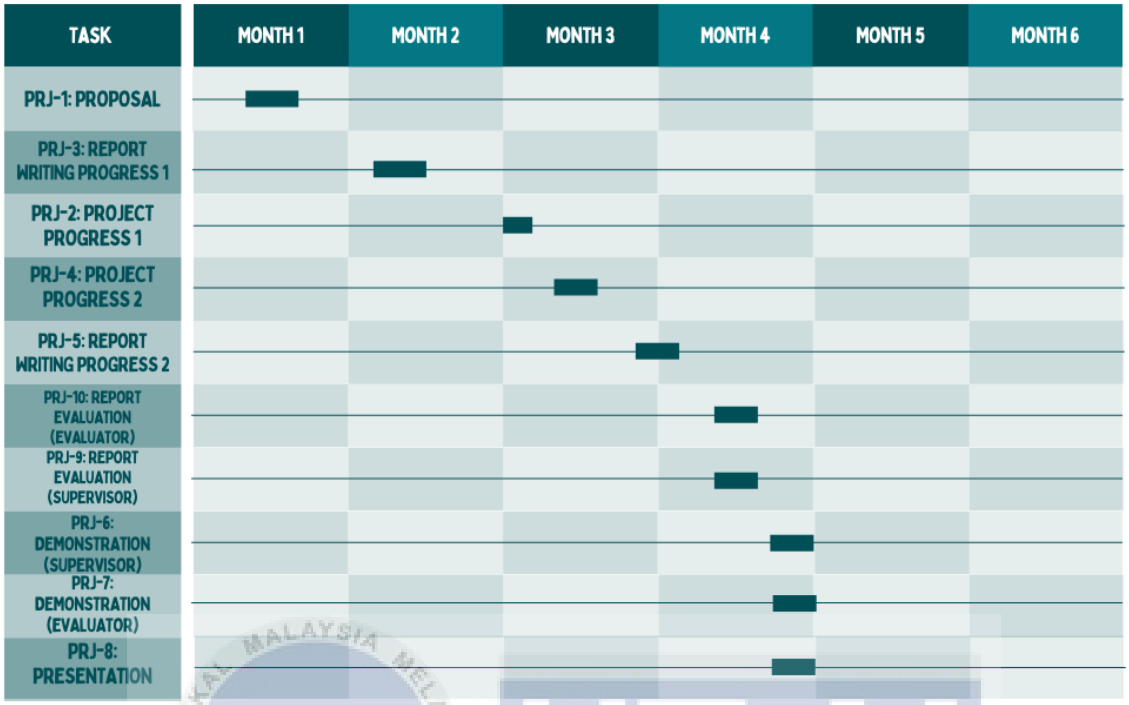
Acer Nitro 5

1. **Processor**: 4-core Intel Core i5-7300HQ, 2.50 GHz, 6 MB Cache
2. **Memory**: 8 GB DDR4 SDRAM
3. **Storage**: 1 TB Serial ATA SSD
4. **Graphics**: NVIDIA GeForce GTX 1050 Ti, 4 GB GDDR5
5. **Display**: 15.6″, FHD (1920 x 1080)

2.4.3 Other Requirements

- Internet : Maxis Home Fibre 300mbps

2.5 PROJECT SCHEDULE AND MILESTONE



2.6 CONCLUSION

In conclusion, this chapter describes the literature review and methodology that will be applied to the project. The literature review is about the facts and findings related to the project and is collected from the internet and archival collections. Besides that, project methodology describes the approach or methodology that will be used to develop this project. All the findings and topics in this chapter are completed. The next chapter is chapter three which will cover project analysis. Chapter 3 contains problem analysis, requirement analysis, data requirement, functional requirement, non-functional requirement, and another requirement that think might be suitable with this project.

CHAPTER 3:

ANALYSIS

3.1 INTRODUCTION

The analysis phase describes and gathers information about the current application being developed as well as identifies high-level requirements by detailing the functional and non-functional requirements of the application. For problem analysis, the problems of the current system will be explained in detail and a flowchart is used to show the flow of the system. This is because the current system did not use a computerized system and Flowchart is suitable to show the current reservation process of parking around UTeM. Requirements analysis will cover the topic of data requirement which is the data that should be the system input and output and the data that the system should store internally.

Then for functional requirements, Data Flow Diagram (DFD) will be used to show the movement of data between external entities and processes and between processes and data stores. The DFD portrays the system in terms of its component pieces, with all interfaces among the component indicated. The non-Functional requirement then will cover the system performs its intended functions. In other requirements, there are three subs which are software requirement, hardware requirement, and network requirements justification. It will be carried out by investigating the application scenario and extracting the required analysis information as described in the following sub-chapter.

3.2 PROBLEM ANALYSIS

For manual parking reservations, it requires a lot of time and energy, and the applicant needs to fill up the form manually about the reservation. The current system is entirely manual and involves a lot of procedures which consumes a lot of time and makes it inefficient to be used. The applicant who wants to make a reservation need to fill up a form with their details and the reason for the parking reservation and needs to wait for approval. In the real situation, Pejabat Keselamatan states the main problem with this manual system is unstandardized and they tend to receive parking requests in different reservations such as by informal call, message through WhatsApp, and verbal communication. This matter will cause many difficulties such as duplicate requests and overlapping on the same date, or location. Other

than that, usually in the old system, the applicant does not know the status of the reservation request whether it has been approved or not. To make things clear, Pejabat Keselamatan needs to call the applicant manually by referring to the form application. Moreover, the other issue received is the old system's unable to produce an appropriate report. Having this report will ease Pejabat Keselamatan to monitor the summary of the parking reservation request for every month.

3.3 REQUIREMENT ANALYSIS

The requirement is a singular documented need of what a particular product or service should be or should be. In systems engineering or software engineering, it is most frequently used formally. For a system to be valuable and useful to a user, a declaration identifying an essential attribute, capacity, trait, or quality of the system must be made. The requirement analysis was the main topic of this section. The functional requirement and non-functional requirement are two crucial components of requirement analysis. A good functional system that satisfies the requirements would be constructed by comprehending and emphasizing these requirements.

3.3.1 Data Requirement

A data dictionary is used to record all those pieces of information about a system (textual or numeric) that cannot be recorded on diagrams. It is an underlying structure that links the different views of the system presented by different types of diagrams. Data requirements described what is the system input and output and the data that the system will be stored. All the information that will be used and manipulated in this system will be illustrated below. In the data requirement, all the information used in the system will be explained in detail so that the user can easily understand the data type that is used.

applicant

Column	Type	Null	Default	Comments
applicant_id *(Primary)*	int(11)	No		
applicant_email	varchar(50)	No		
applicant_password	varchar(255)	No		
applicant_name	varchar(100)	No		
applicant_phoneno	varchar(15)	No		
applicant_position	varchar(50)	No		
PTJ_id	int(30)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	applicant_id	20	A	No	
fk_faculty	BTREE	No	No	PTJ_id	4	A	No	

approver

Column	Type	Null	Default	Comments
approver_id *(Primary)*	int(11)	No		
approver_email	varchar(50)	No		

Column	Type	Null	Default	Comments
approver_password	varchar(255)	No		
approver_name	varchar(100)	No		
approver_phoneno	int(15)	No		
approver_position	varchar(50)	No		
address	varchar(300)	No		
image	longblob	No		
PTJ_id	int(30)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	approver_id	6	A	No	
fk_ptj_id	BTREE	No	No	PTJ_id	6	A	No	

parking

Column	Type	Null	Default	Comments
parking_id *(Primary)*	int(11)	No		
parkingcode	varchar(300)	No		
parking_position	varchar(300)	No		
parking_type	varchar(100)	No		
PTJ_id	int(30)	No		

Column	Type	Null	Default	Comments
availability	varchar(100)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	parking_id	251	A	No	
unique_parkingcode	BTREE	Yes	No	parkingcode	251	A	No	
fk_parking_ptj	BTREE	No	No	PTJ_id	4	A	No	

ptj

Column	Type	Null	Default	Comments
PTJ_id *(Primary)*	int(30)	No		
PTJ_fullname	varchar(300)	No		
PTJ_initialname	varchar(300)	No		
PTJ_address	varchar(300)	No		
PTJ_image	longblob	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	PTJ_id	12	A	No	

reservation

Column	Type	Null	Default	Comments
reservation_id *(Primary)*	int(11)	No		
applicant_id	int(11)	No		
purpose	varchar(300)	No		
plateno	varchar(300)	No		
cartype	varchar(300)	No		
PTJ_id	int(30)	No		
location	varchar(300)	No		
startdate	date	No		
enddate	date	No		
parking_id	int(11)	No		
quantity	int(11)	No		
status	varchar(300)	No	pending	
special_request_id	varchar(50)	No		
approver_id	int(11)	No		
reason_status	varchar(300)	No		
duration	int(30)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	reservation_id	2	A	No	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
fk_parking	BTREE	No	No	parking_id	2	A	No	
fk_ptjid	BTREE	No	No	PTJ_id	2	A	No	
fk_approver	BTREE	No	No	applicant_id	2	A	No	
approver_id	BTREE	No	No	approver_id	2	A	No	

special_request

Column	Type	Null	Default	Comments
special_request_id *(Primary)*	varchar(50)	No		
request_name	varchar(300)	No		
request_type	varchar(100)	No		
details	varchar(300)	No		

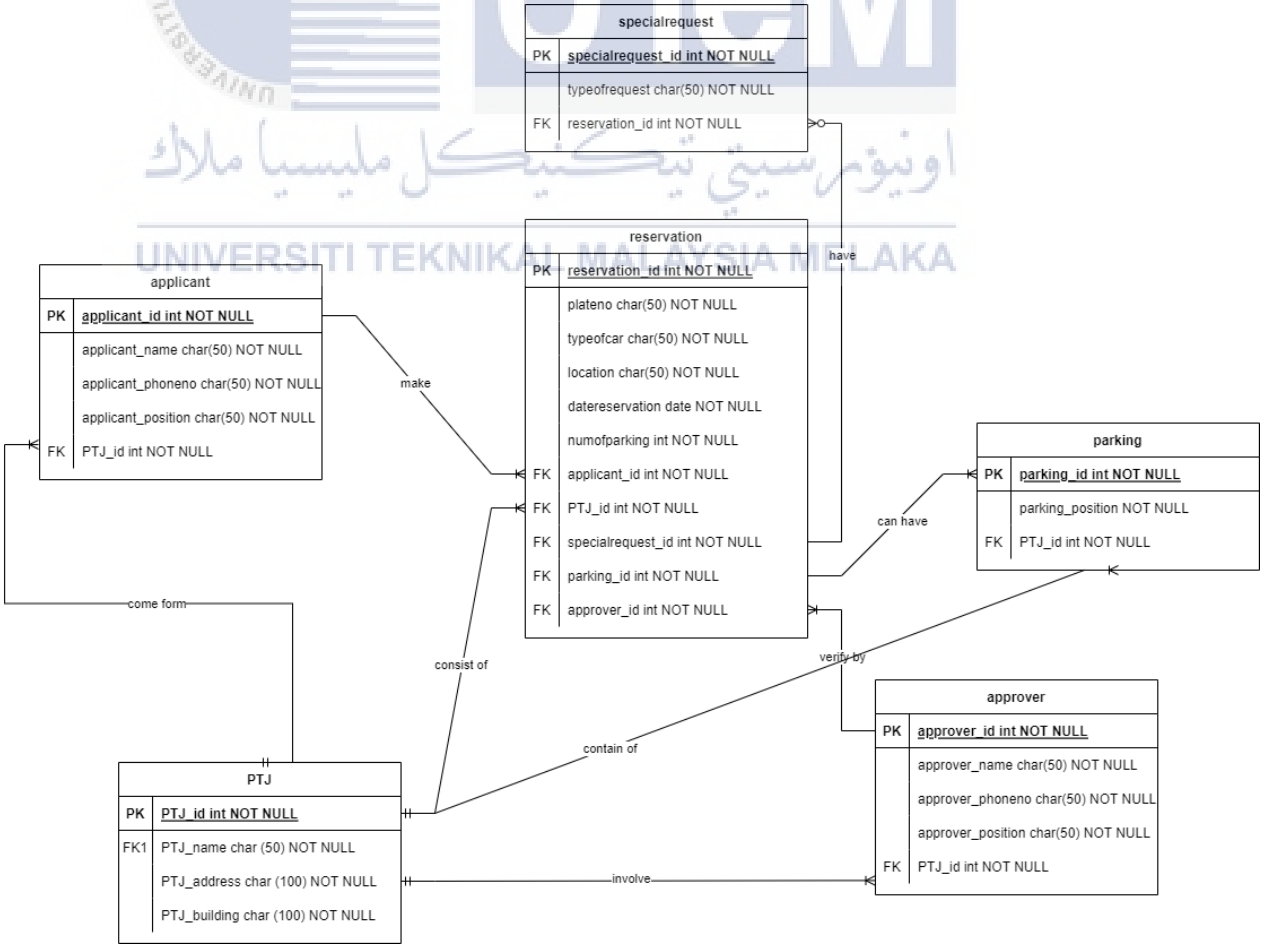
Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	special_request_id	0	A	No	

3.3.2 Functional Requirement

Functional requirements are related directly to a process the system must perform or the information it needs. Besides that, it also shows the statements of services the system should provide, how the system should react to inputs, and how the system should behave in situations.

3.3.2.1 Entity Relationship Diagram (ERD)



3.3.3 Non-Functional Requirement

Non-functional requirement refers to behavioral properties that the system must have, such as performance and usability. It also can describe the variety of characteristics of the system such as operational, performance, security, and cultural and political. There are several main classifications of this non-functional requirement such as product requirements, organizational requirements, and external requirements. Product requirements explain the requirements which specify that the delivered product must behave in a particular way, for example, execution speed, reliability, and more. Organizational requirements describe requirements that are a consequence of organizational policies and procedures like process standards used, implementation requirements, and more. Lastly, the external requirements explain the requirements which arise from factors that are external to the system and its development process such as interoperability requirements and legislative requirements.

Requirement Category	Non-Functional Type (Process/ Data)	Example
Accuracy	Both	Process: All required fields for reservation form must be entered. Data: Parking Reservation must be made at least 3 days before date of use.
Availability	Process	UTeM EasyPark can be access 24 hours in a day and 7 days per week including weekends.
Backup and Recovery	Both	Process: Parking reservations can be made available after unplanned system downtime within 1 working day. Data: Data for parking reservation will be made daily.
Error-handling	Process	The popup message will appear for any required field not filled.
Performance	Process	System responses should be not

		more than 3 second for all the system process.
Security	Both	Process: Only lecturer/ staff UTeM can login to the system. Data: Parking information, reservation information only can be manage and access by Pejabat Keselamatan staff that in charge.

3.3.4 Other Requirements

The UTeM Easy Park System is a software solution aimed at streamlining parking management within the Universiti Teknikal Malaysia Melaka (UTeM) campus. This document outlines the software requirements necessary for the development and successful implementation of the UTeM Easy Park System. The system relies on a combination of various software components, including Microsoft Windows 11, XAMPP, Google Chrome, Microsoft Word, and Draw.io.

Operating System:

The UTeM Easy Park System is designed to be compatible with Microsoft Windows 11 as the primary operating system. It should support all the necessary functionalities and features offered by the latest version of Windows.

Web Server:

XAMPP will serve as the web server for hosting and deploying the UTeM Easy Park System. It provides a complete environment that includes Apache, MySQL, PHP, and Perl, ensuring compatibility with the system's backend requirements.

Web Browsers:

The system should be accessible through Google Chrome, one of the most popular web browsers. Compatibility testing should be conducted to ensure the system's smooth functioning on the latest stable version of Google Chrome.

Document Processing:

Microsoft Word will be used for generating and managing system-related documentation, including user manuals, installation guides, and system requirements. It is crucial to have Microsoft Word installed to facilitate efficient documentation and collaboration among the development team.

Diagramming Tool:

Draw.io, a popular diagramming and visual representation tool, will be utilized to create system architecture diagrams, flowcharts, and other visual representations related to the UTeM Easy Park System. It allows for clear communication of system design and requirements among team members.

Additional Considerations:

The system should be developed using a programming language that is compatible with the chosen web server and supported by Windows 11.

Database management and communication with the web server will require the appropriate database management system compatible with XAMPP, such as MySQL or MariaDB.

Version control software, such as Git, should be employed to track changes, collaborate effectively, and maintain code integrity throughout the development process.

Security measures, such as SSL/TLS encryption, should be implemented to ensure secure communication between the system and its users.

The UTeM Easy Park System relies on various software components, including Microsoft Windows 11, XAMPP, Google Chrome, Microsoft Word, and Draw.io. These software requirements are essential for the successful development and deployment of the system, ensuring compatibility, functionality, and efficient documentation.

3.4 CONCLUSION

This chapter explains about the analysis that covers problem analysis for the current system, data requirement, functional requirement, non-functional requirement, and other requirements. Problem analysis for the current system is analyzed and from that requirements for the new system to be developed can be determined. Data requirement is for determining what data should be stored in the database. Functional requirements specify the functions of the system, and how it records, computes, transforms, and transmits data. Non-functional requirements specify how well the functions of the new system perform its intended functions. Other requirements contain three categories which are software, hardware, and network requirements. The justifications for the usage of each of them in this system are explained in this chapter. The following chapter will cover more about the design of the system. First, it will describe high-level design which contains system architecture, user interface design, and conceptual and logical database design. Second, it will describe system architecture which contains software design and physical database design.

CHAPTER 4

DESIGN

4.1 INTRODUCTION

This chapter focuses on the detailed design of the UTeM Easypark, a Parking Reservation System. Building upon the preliminary design analysis, this chapter delves into the specifics of the system's architecture, components, and functionality. By presenting a comprehensive design plan, the chapter aims to provide a clear roadmap for implementing the system effectively and efficiently.

1. System Architecture: The detailed design outlines the overall architecture of the UTeM Easypark system. It defines the system's structure, including its various modules, components, and interactions. The architecture ensures the system's scalability, reliability, and performance by organizing the different elements coherently and optimally.

2. User Interface Design: A crucial aspect of the detailed design is the user interface (UI) design. It involves designing intuitive and user-friendly interfaces that allow users to interact with the system seamlessly. The chapter explores the UI components, layout, navigation, and visual design principles to provide an engaging and efficient user experience.

3. Database Design: Efficient data management is essential for the functioning of any system. The chapter addresses the database design aspects of UTeM Easypark, including defining the database schema, tables, relationships, and indexing strategies. It ensures proper data organization, integrity, and security, facilitating smooth data storage and retrieval.

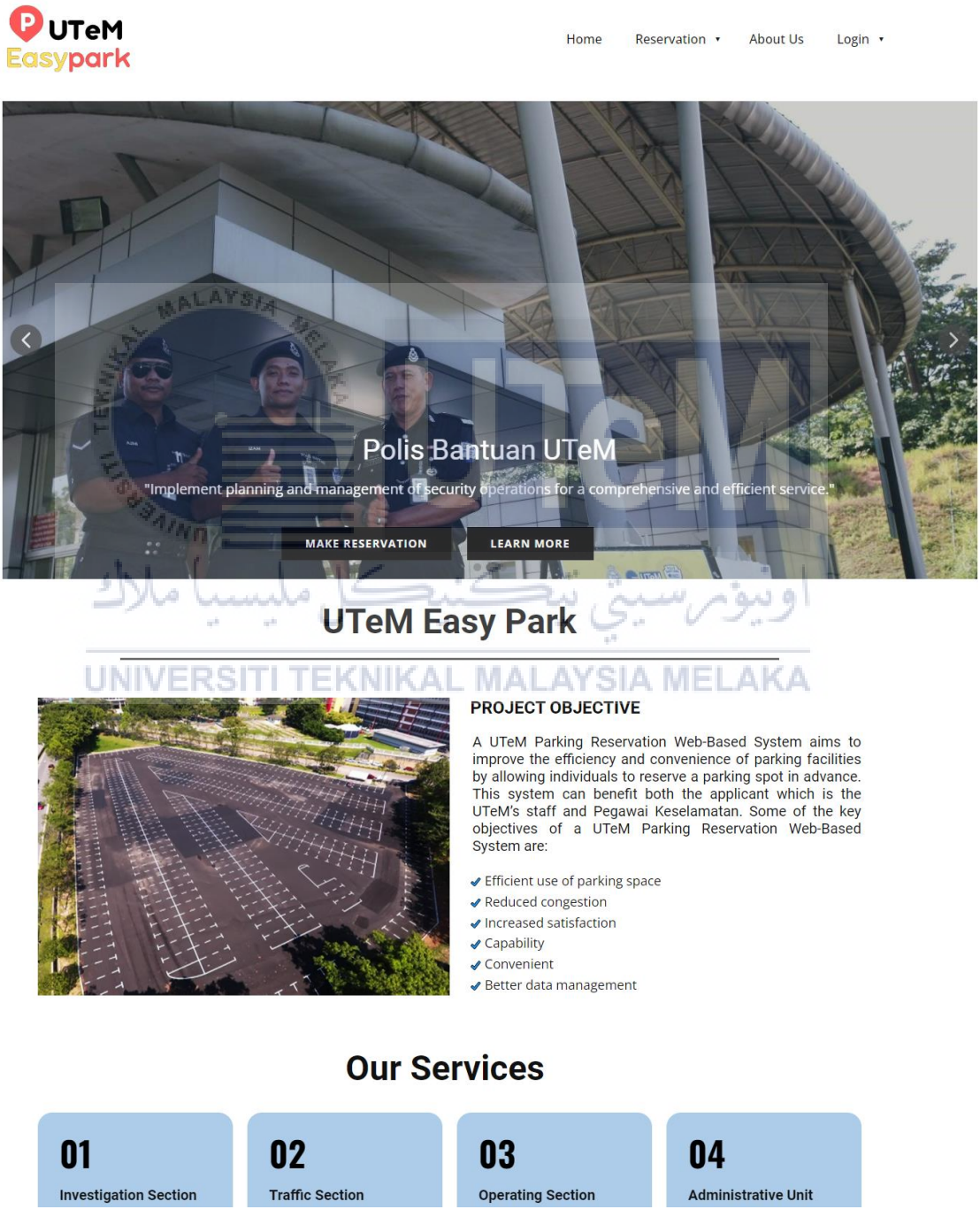
4. Functional Modules and Features: The detailed design describes the various functional modules and features of UTeM Easypark in depth. It elaborates on each module's purpose, functionality, and interdependencies. Examples of functional modules could include user login, parking reservation, administrative controls, and reporting.

4.2 HIGH LEVEL DESIGN

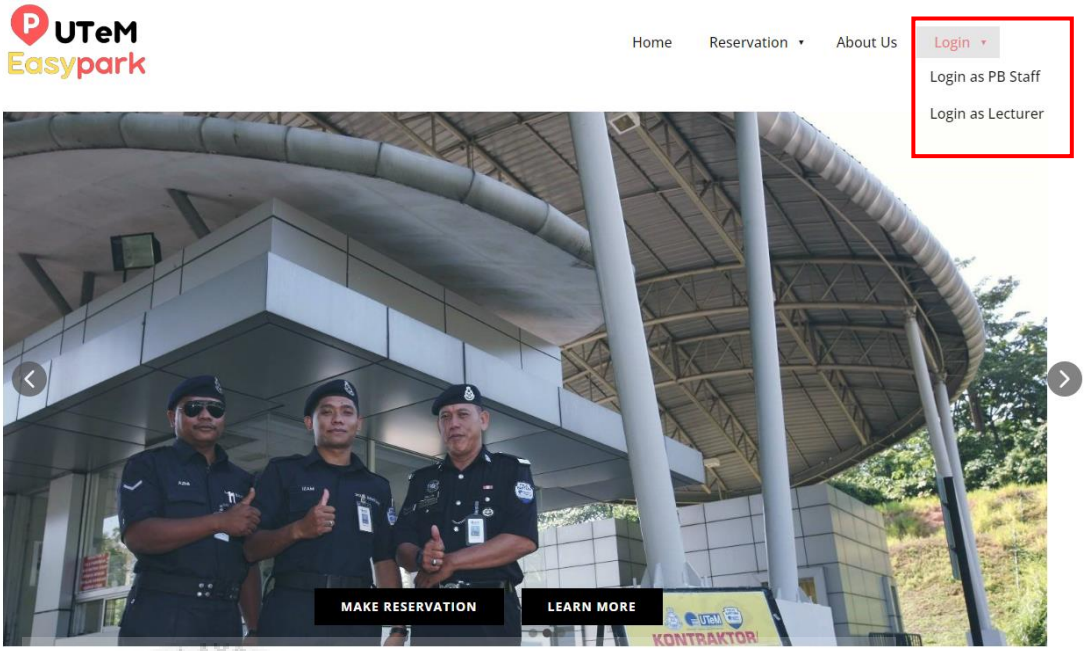
4.2.1 System Architecture

4.2.2 User Interface Design

- Homepage of UTeM Easypark

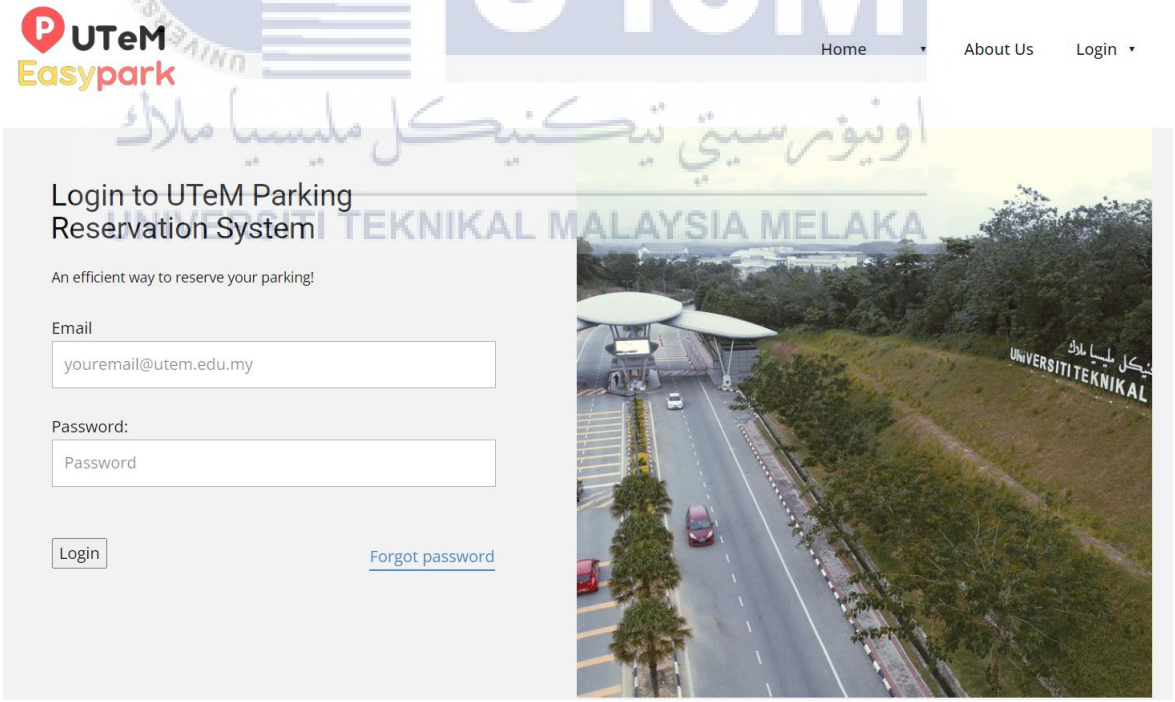


- Login section

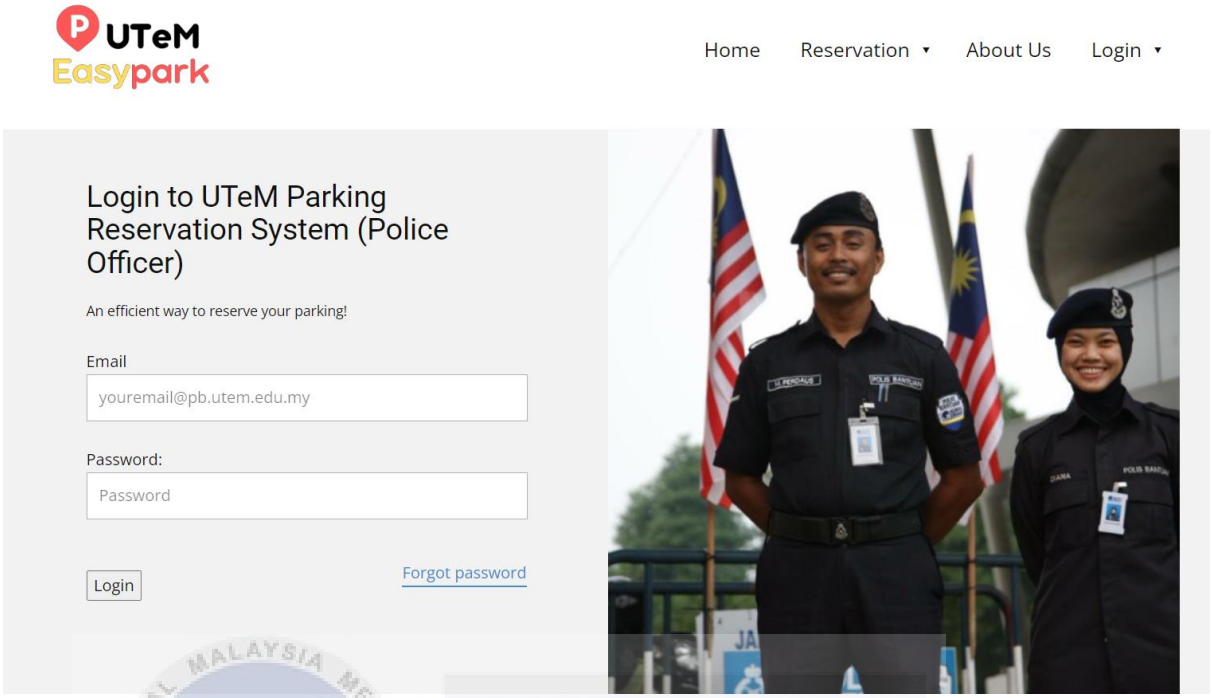


UTeM Easy Park

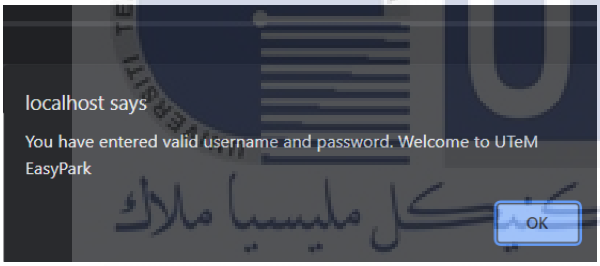
- Lecturer/ Staff login page



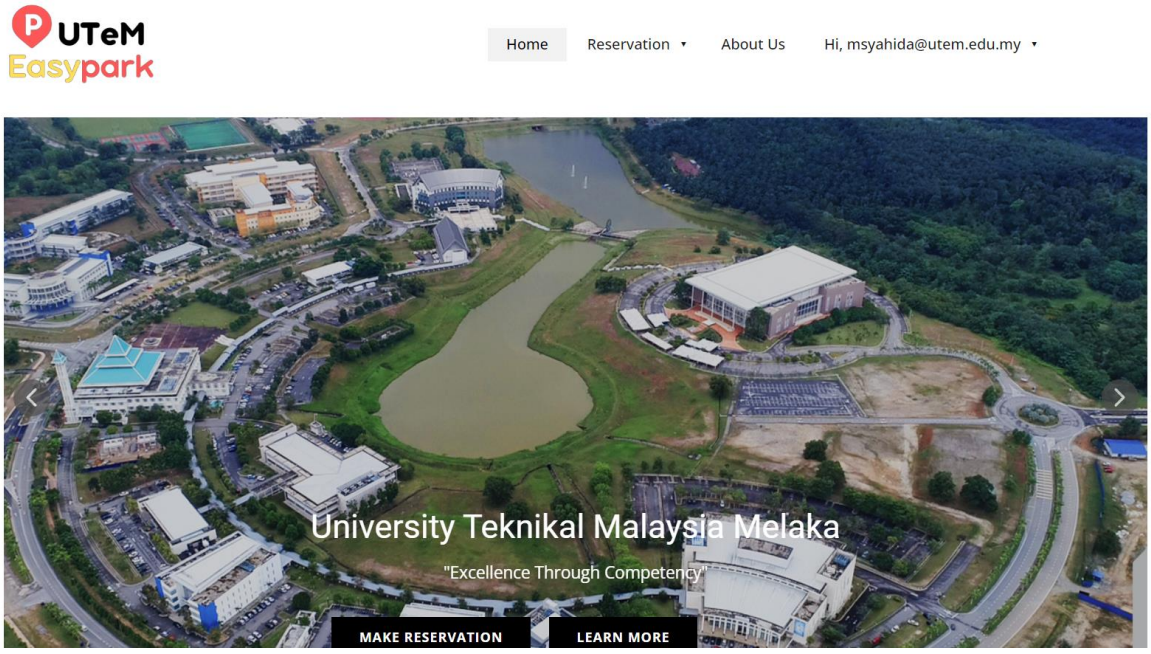
- Polis Bantuan login page



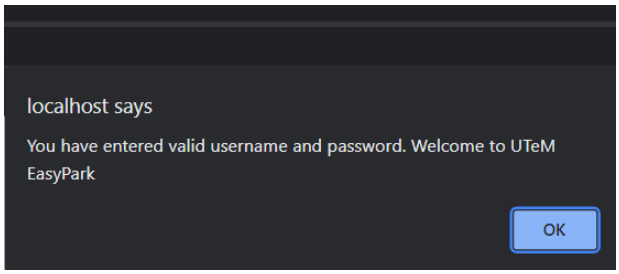
- Successful login (Applicant, read as: lecturer/staff)

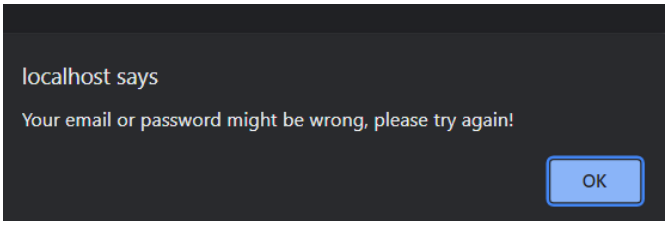


- Homepage for Applicant

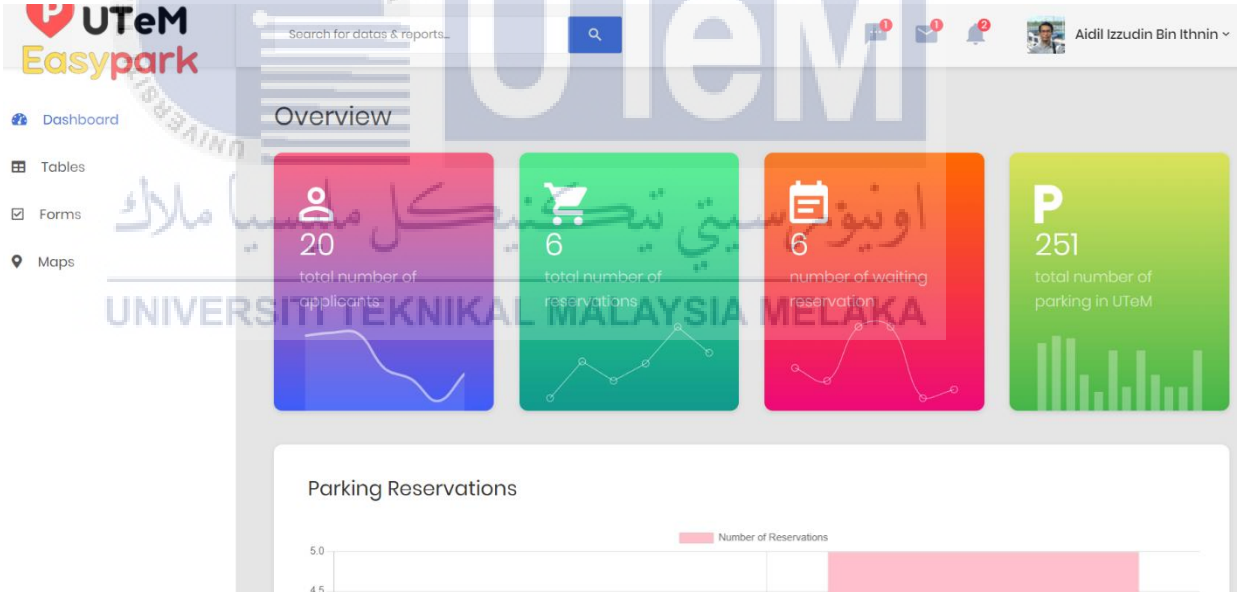


- Successful login and wrong password message box (Approver, read as: Polis Bantuan)

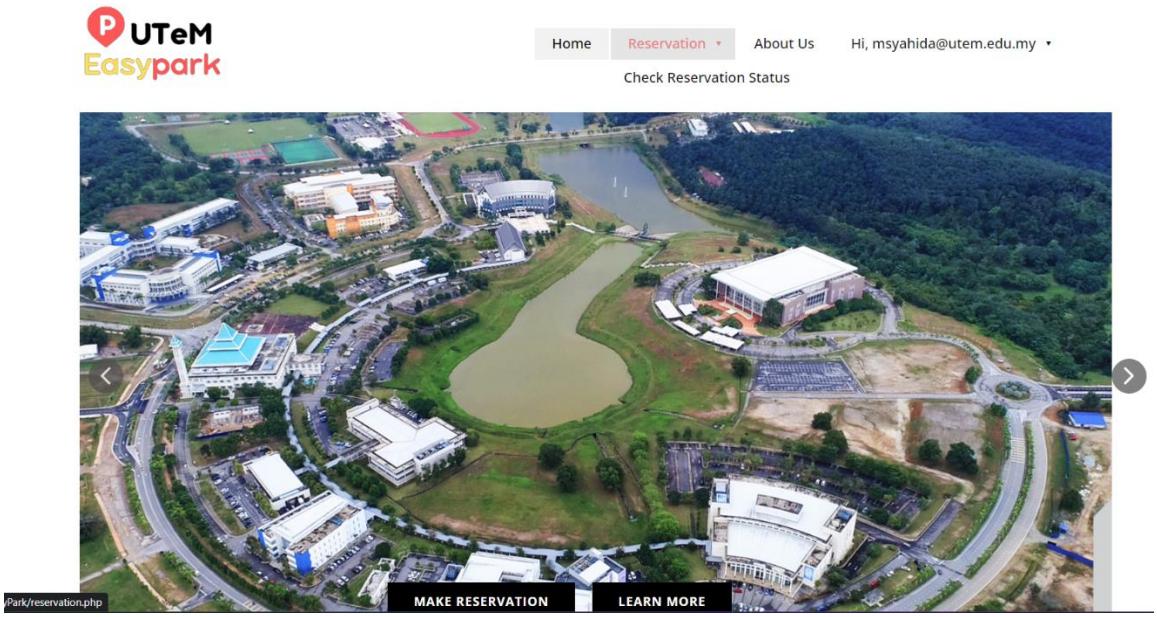


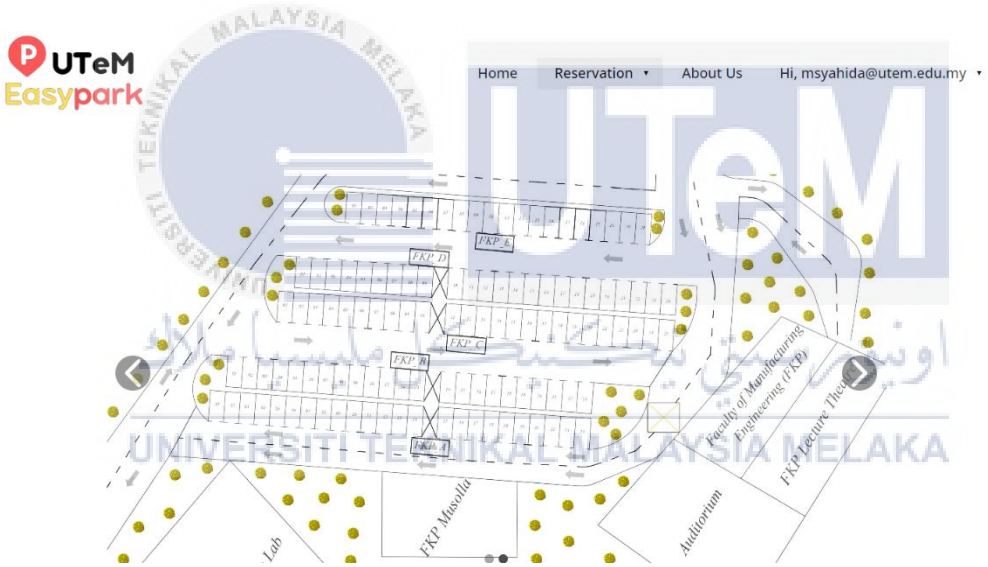


- Homepage for admin



- Make reservation





Make a reservation

A UTeM Parking Reservation Web-Based System aims to improve the efficiency and convenience of parking facilities by allowing individuals to reserve a parking spot in advance.

Name

DR. SYAHIDA BINTI MOHTAR

Phone

+6062702482

Email

msyahida@utem.edu.my

Positon

PENSYARAH

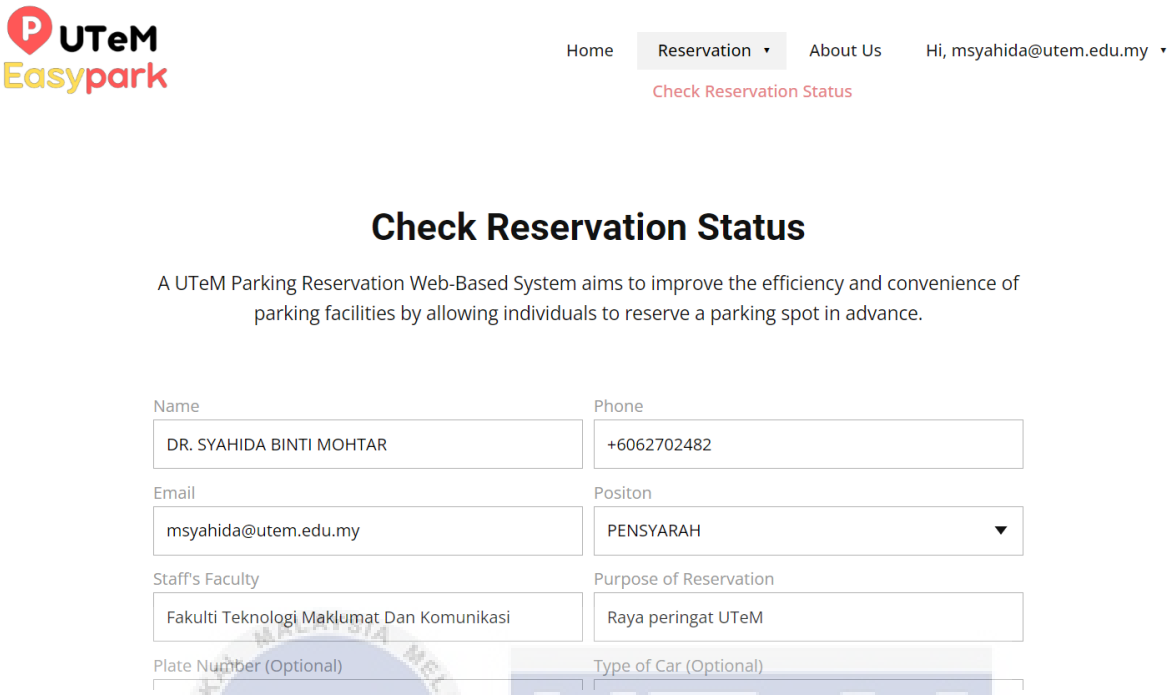
Staff's Faculty

Fakulti Teknologi Maklumat Dan Komunikasi

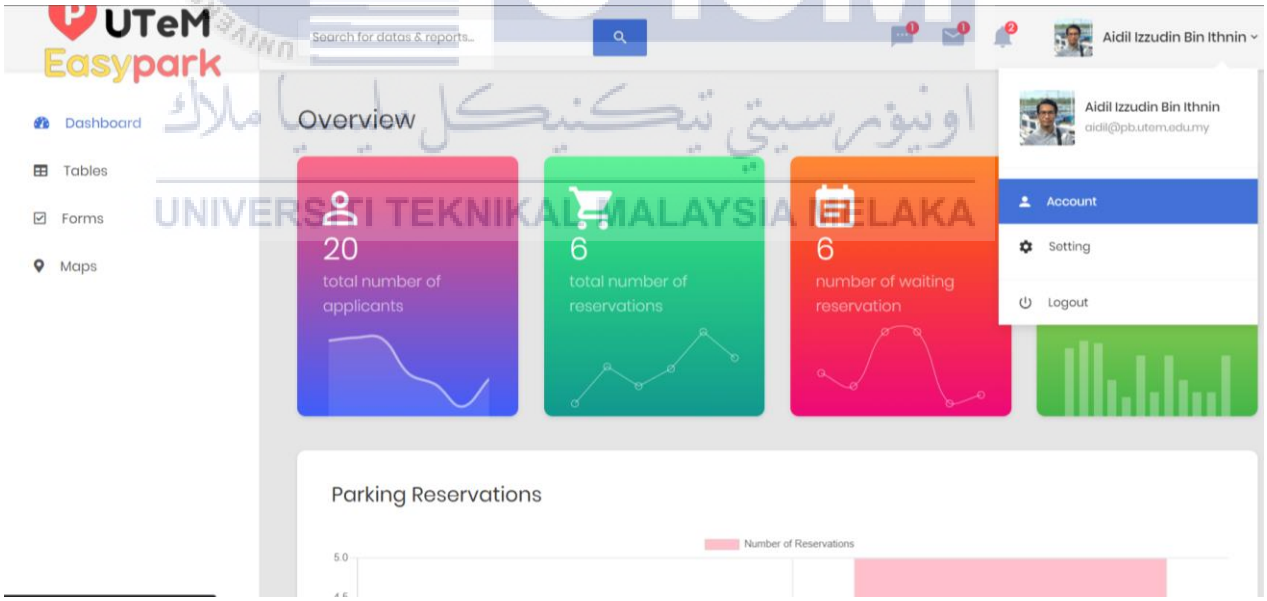
Purpose of Reservation

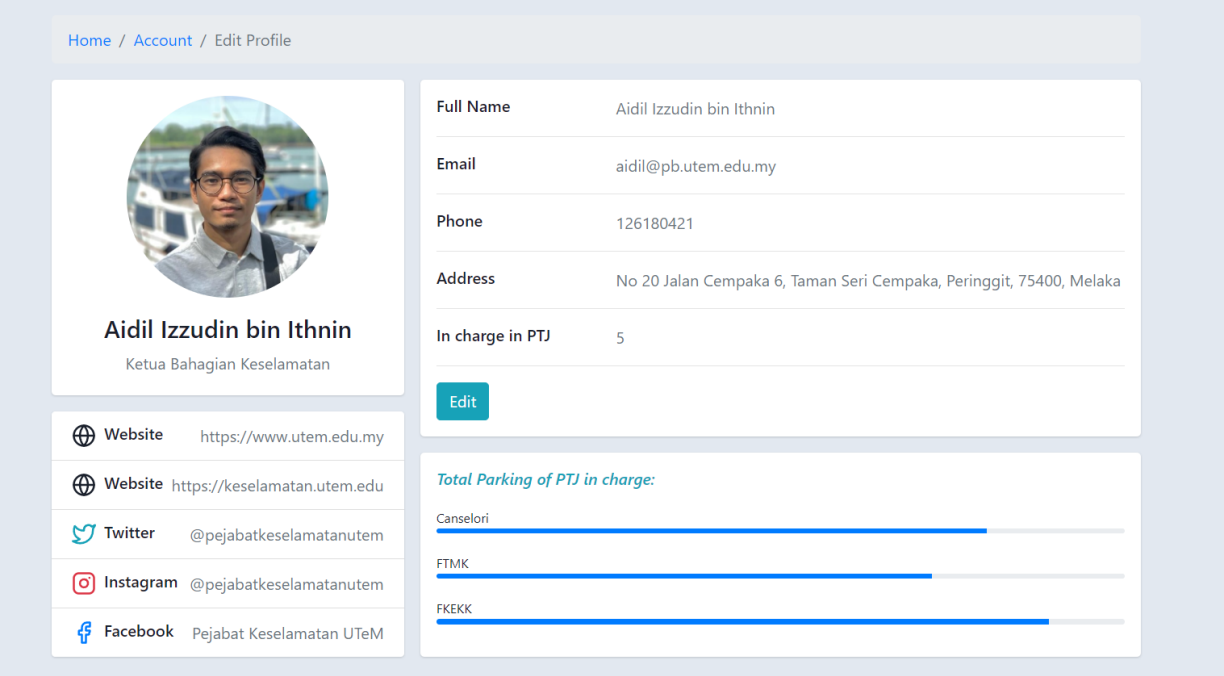
e.g: Hari Raya UTeM

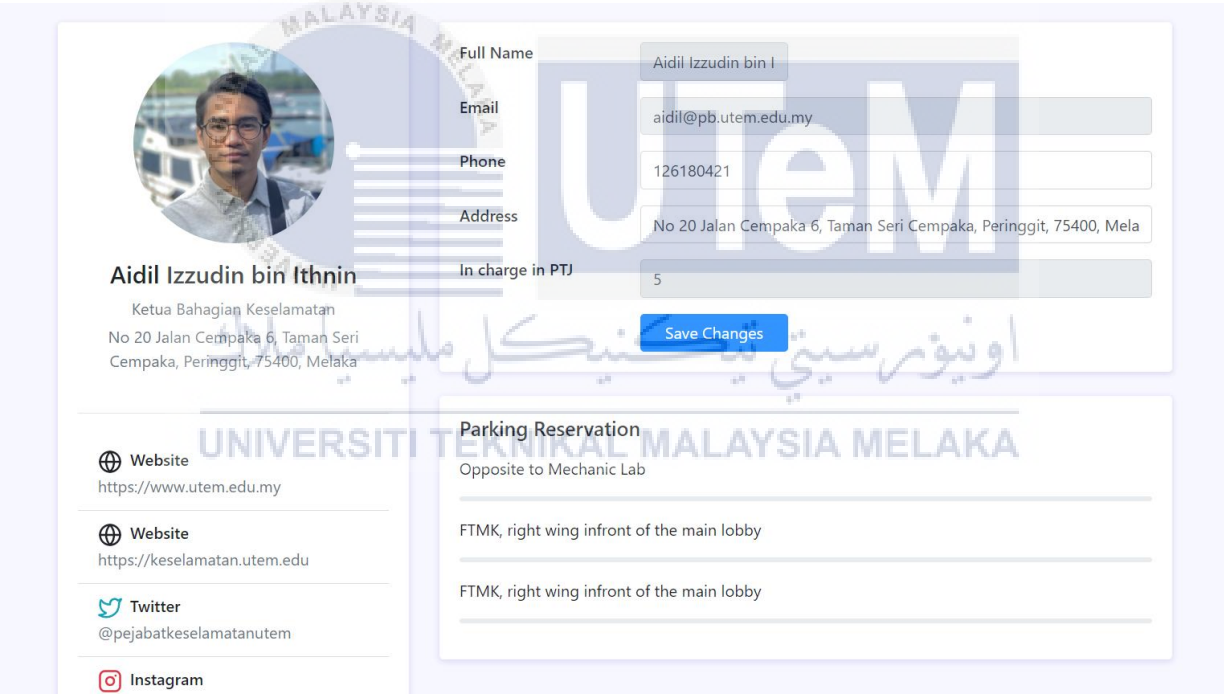
- Check Reservation Status



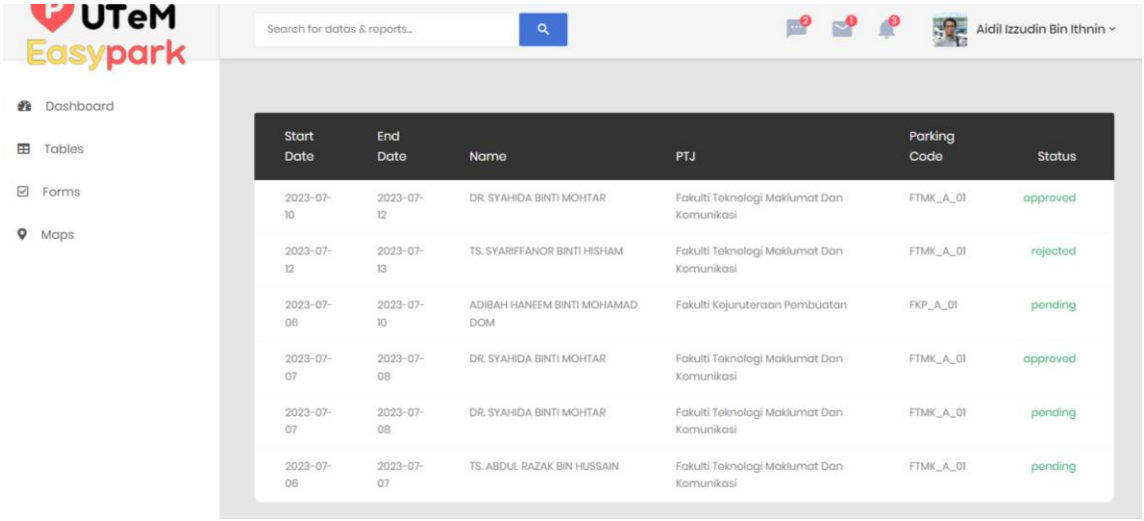
- Update Profile (Approver, read as: Polis Bantuan)





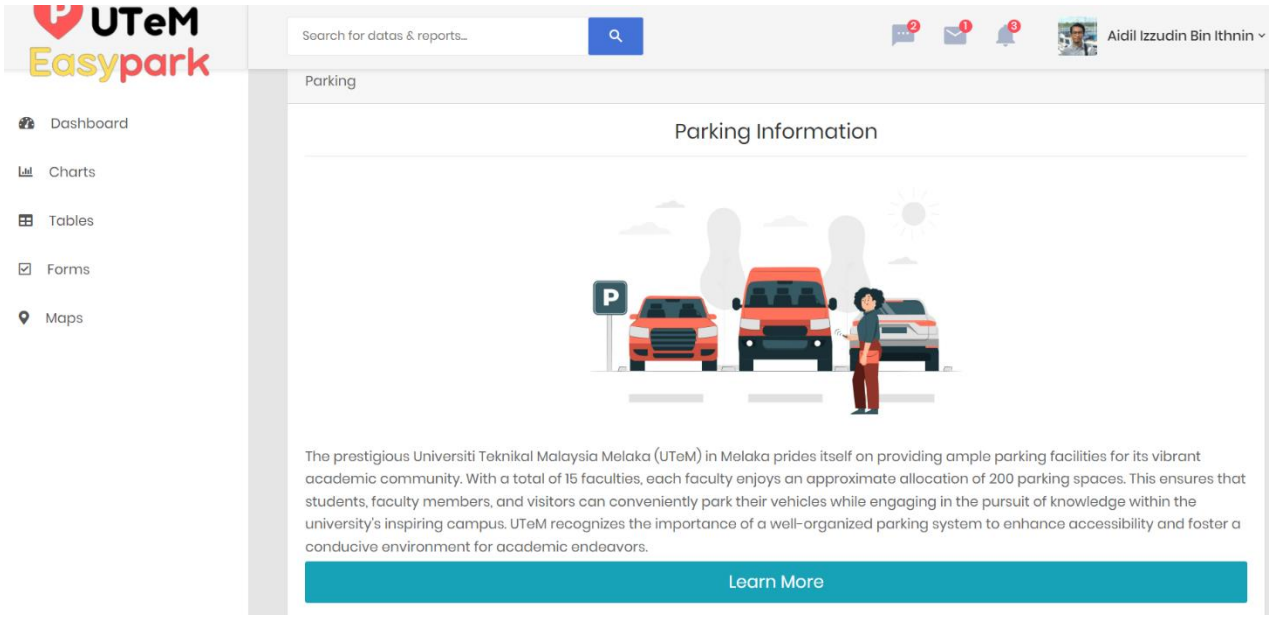


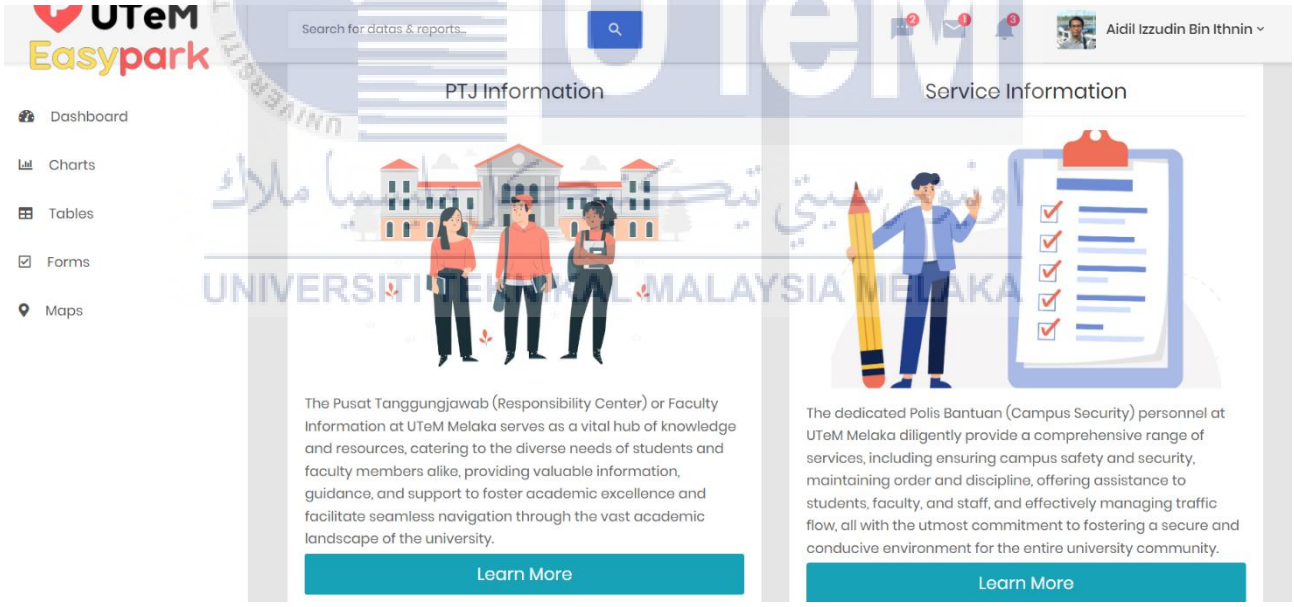
- List of Parking Request



- Verify or update Parking Reservation Status

- Update parking, PTJ, special request information





4.2.2.1 Navigation Design

4.2.2.2 Input Design

In input design, it shows data that will be input into the system by users. The table below shows the data along with details. During input design, designing a user-friendly interface and process is very important. It is to ensure the quality, accuracy, and timeliness of system input.

Table 4.2.2.2.1 Table Applicant Design

No	Field	GUI Control	Validation rule
1	applicant_id	text	not null
2	applicant_email	text	not null
3	applicant_password	text	not null
4	applicant_name	text	not null
5	applicant_phoneno	text	not null
6	applicant_position	text	not null

Table 4.2.2.2.2 Table Applicant Design

No	Field	GUI Control	Validation rule
1	approver_id	text	not null
2	approver_email	text	not null
3	approver_password	text	not null
4	approver_name	text	not null
5	approver_phoneno	text	not null
6	approver_position	text	not null
7	address	text	not null
8	image	file	not null
9	PTJ_id	text	not null

Table 4.2.2.2.3 Table Parking Design

No	Field	GUI Control	Validation rule
1	parking_id	text	not null

2	parking_code	text	not null
3	parking_position	text	not null
4	parking_type	text	not null
5	availability	text	not null
6	PTJ_id	text	not null

Table 4.2.2.2.3 Table PTJ Design

No	Field	GUI Control	Validation rule
1	PTJ _id	text	not null
2	PTJ_fullname	text	not null
3	PTJ_initialname	text	not null
4	PTJ_address	text	not null

Table 4.2.2.2.3 Table Special Request Design

No	Field	GUI Control	Validation rule
1	special_request_id	text	not null
2	request_name	text	not null
3	request_type	text	not null
4	details	text	not null

Table 4.2.2.2.2 Table Reservation Design

No	Field	GUI Control	Validation rule
1	reservation_id	text	not null
2	applicant_id	text	not null
3	purpose	text	not null
4	plateno	text	not null
5	cartype	text	not null
6	PTJ_id	text	not null
7	location	text	not null
8	startdate	file	not null

9	enddate	text	not null
10	parking_id	text	not null
11	quantity	integer	not null
12	status	text	not null
13	special_request_id	text	not null
14	approver_id	text	not null
15	reason_status	text	not null
16	duration	integer	not null

4.2.2.3 Output Design

Output design is the one that meets the requirements of the end user, and presents the information in a way that is clear, easy to read, and visually interactive. The table below shows the output design for the system. Output design shows the output from the system.

Form Name	Output Name	GUI Control	Validation Rule
User authentication	Success login	Text field	Not null
Make parking reservation	Display reservation form	Text field	Not null
Update parking information	Display parking form	Text field	Not null
Update PTJ information	Display PTJ form	Text field	Not null
Update special request information	Display special request form	Text field	Not null
Update Polis Bantuan profile	Display Polis Bantua's profile form	Text field	Not null

Table 4.2.2.3.1 Output Design Table

4.2.3 Conceptual and Logical Database Design

Logical database design describes the attribute of each entity such as data type, length, constraints, and description. Since logical database design requires all objects in the entity relationship diagram to be mapped into specific constructs, a data dictionary will be used for that reason. The main purpose of a data dictionary is to provide detailed information on all tables that will be used in the system. Other than that, a data dictionary is also used to ensure maximum understanding of the tables and attributes within the system's database. The details about the logical database design are described in the tables below.

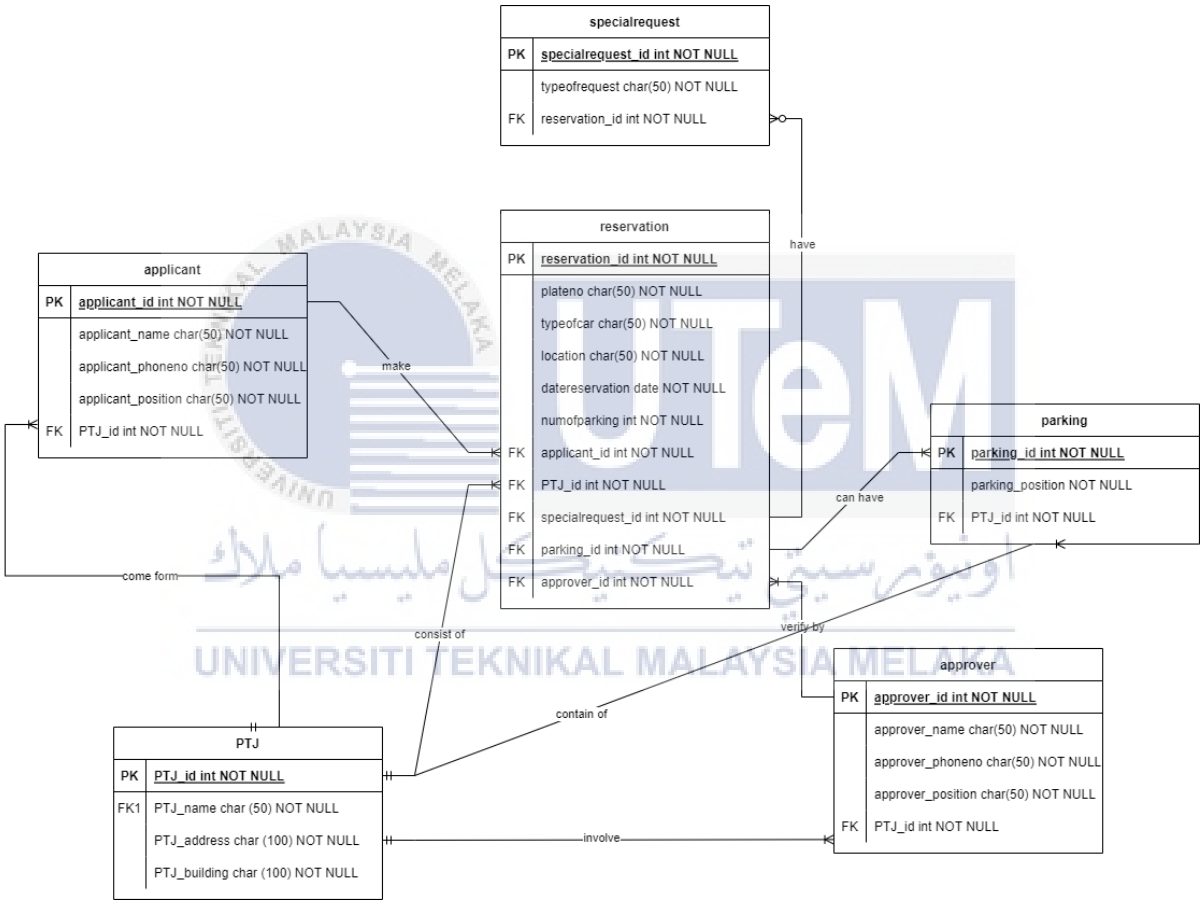


Figure 4.2.3.1 ERD of UTeM Easypark

Business rules:

- One applicant can make many parking reservations.
- Each reservation can be reserved by one applicant.
- Many applicants come from one PTJ.
- Each PTJ consists of many applicants.
- Each reservation only consists of one PTJ.

- Each PTJ consists of many parking spaces.
- Each PTJ involves many approvers.
- Each reservation is verified by one approver.
- Many parking only have one PTJ.
- Many parking consists of one reservation.
- One or optional special request is involved in each reservation.
- Each reservation has no or many special requests.

Table 4.2.3.1 Table Applicant Data Dictionary

No	Column Name	Data Type and Length	Constraint	Description
1	applicant_id	int(11)	PK	Applicant ID
2	applicant_email	varchar(50)		Applicant email
3	applicant_password	varchar(50)		Applicant password
4	applicant_name	varchar(100)		Applicant name
5	applicant_phoneno	varchar(15)		Applicant phone no
6	applicant_position	varchar(50)		Applicant position
7	PTJ_id	int(30)	FK	Refer to PTJ_id in PTJ table

Table 4.2.3.2 Table Approver Data Dictionary

No	Column Name	Data Type and Length	Constraint	Description
1	approver_id	int(11)	PK	Approver ID
2	approver_email	varchar(50)		Approver email
3	approver_password	varchar(50)		Approver password
4	approver_name	varchar(100)		Approver name
5	approver_phoneno	varchar(15)		Approver phone no
6	approver_position	varchar(50)		Approver position
7	address	varchar(300)		Approver address
8	image	longbob		Approver image
9	PTJ_id	int(30)	FK	Refer to PTJ_id in PTJ table

Table 4.2.3.2 Table Parking Data Dictionary

No	Column Name	Data Type and Length	Constraint	Description
1	parking_id	int(11)	PK	Parking ID
2	parking_code	varchar(300)		Parking code
3	parking_position	varchar(300)		Parking position
4	parking_type	varchar(100)		Parking type
5	availability	varchar(100)		Parking availability
6	PTJ_id	int(30)	FK	Refer to PTJ_id in PTJ table

Table 4.2.3.2 Table PTJ Data Dictionary

No	Column Name	Data Type and Length	Constraint	Description
1	PTJ _id	int(30)	PK	PTJ ID
2	PTJ_fullname	varchar(300)		PTJ full name
3	PTJ_initialname	varchar(300)		PTJ initial name
4	PTJ_address	varchar(300)		PTJ address

Table 4.2.3.2 Table Special Request Data Dictionary

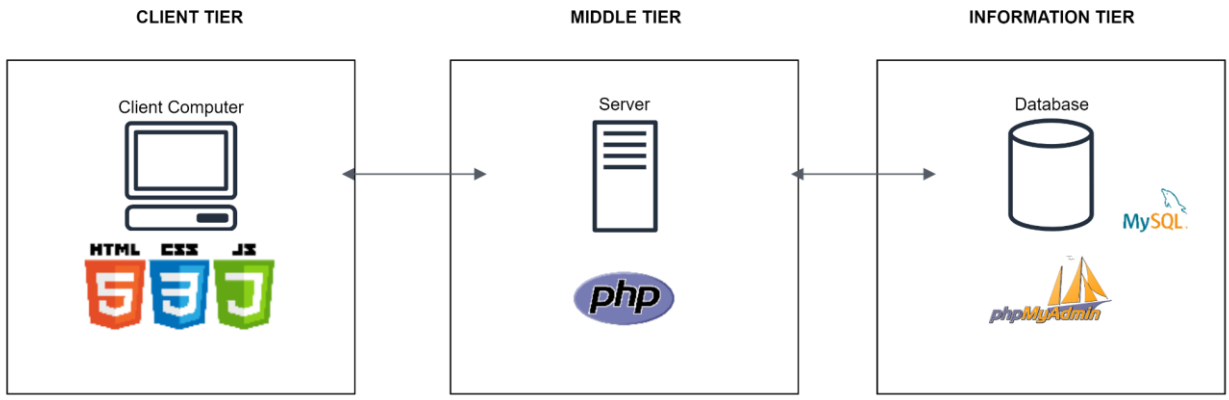
No	Column Name	Data Type and Length	Constraint	Description
1	special_request_id	int(30)	PK	Special request ID
2	request_name	varchar(300)		Special request name
3	request_type	varchar(300)		Special request type
4	details	varchar(300)		Special request detail

Table 4.2.3.2 Table Reservation Data Dictionary

No	Column Name	Data Type and Length	Constraint	Description
1	reservation_id	int(11)	PK	Reservation ID

2	applicant_id	int(30)	FK	Refer to applicant_id in applicant table
3	purpose	varchar(300)		Reservation purpose
4	plateno	varchar(300)		Car plate number
5	cartype	varchar(100)		Car type
6	PTJ_id	int(30)	FK	Refer to PTJ_id in PTJ table
7	location	varchar(300)		Location
8	startdate	date		Start date of Reservation
9	enddate	date		End date of Reservation
10	parking_id	int(11)	FK	Refer to parking_id in parking table
11	quantity	int(30)		Quantity of desired parking
12	status	varchar(300)		Status of Reservation
13	special_request_id	int(30)	FK	Refer to special_request_id in special_request table
14	approver_id	int(30)	FK	Refer to approver_id in approver table
15	reason_status	varchar(300)		Reason of status
16	duration	int(30)		Duration of Reservation

4.3 System Architecture



A system architecture is a conceptual model that defines a system's structure, behavior, and more views. In this subchapter, it will discuss the system architecture. It includes software design and physical database design. For software design, it will discuss in detail every

function according to the data flow diagrams. While in physical database design, it states the coding that will be implemented to create all the tables in the database. The detail explanation is explained as follow.

4.3.1 Software Design

Software design will explain the functions that are provided by the system. It explains the flow of each function from the data flow diagrams.

a. Parking Reservation
   - Lecturer/ Staff request parking space by inserting their details.
   - Lecturer/ Staff choose desired date of parking.
   - Lecturer/ Staff check status of parking request.
   - Process records will be saved into the database.
   - Process end.

b. Manage parking space information.
   - Polis Bantuan provides details of parking space information according to each PTJ available around UTeM.
   - The system records the parking information and save it in the database.
   - Process end.

c. Manage PTJ information.
   - Polis Bantuan provides details of PTJ information.
   - The system records the PTJ information and save it in the database.
   - Process end.

4.3.2 Physical Database Design

In physical database design, the coding to create all tables in the database will be stated. The coding will then be implemented in the MySQL database management system. The SQL codes are stated as follows.

- Applicant table

```
CREATE TABLE `applicant` (
  `applicant_id` int(11) NOT NULL,
  `applicant_email` varchar(50) NOT NULL,
  `applicant_password` varchar(255) NOT NULL,
  `applicant_name` varchar(100) NOT NULL,
  `applicant_phoneno` varchar(15) NOT NULL,
  `applicant_position` varchar(50) NOT NULL,
```

```
  `PTJ_id` int(30) NOT NULL
)
```

- Approver table

```
CREATE TABLE `approver` (
  `approver_id` int(11) NOT NULL,
  `approver_email` varchar(50) NOT NULL,
  `approver_password` varchar(255) NOT NULL,
  `approver_name` varchar(100) NOT NULL,
  `approver_phoneno` int(15) NOT NULL,
  `approver_position` varchar(50) NOT NULL,
  `address` varchar(300) NOT NULL,
  `image` longblob NOT NULL,
  `PTJ_id` int(30) NOT NULL
)
```

- Parking table

```
CREATE TABLE `parking` (
  `parking_id` int(11) NOT NULL,
  `parkingcode` varchar(300) NOT NULL,
  `parking_position` varchar(300) NOT NULL,
  `parking_type` varchar(100) NOT NULL,
  `PTJ_id` int(30) NOT NULL,
  `availability` varchar(100) NOT NULL
)
```


- PTJ table

```
CREATE TABLE `ptj` (
  `PTJ_id` int(30) NOT NULL,
  `PTJ_fullname` varchar(300) NOT NULL,
  `PTJ_initialname` varchar(300) NOT NULL,
  `PTJ_address` varchar(300) NOT NULL,
  `PTJ_image` longblob NOT NULL
)
```

- Reservation table

```
CREATE TABLE `reservation` (
  `reservation_id` int(11) NOT NULL,
  `applicant_id` int(11) NOT NULL,
  `purpose` varchar(300) NOT NULL,
  `plateno` varchar(300) NOT NULL,
  `cartype` varchar(300) NOT NULL,
  `PTJ_id` int(30) NOT NULL,
  `location` varchar(300) NOT NULL,
```

```
  `startdate` date NOT NULL,
  `enddate` date NOT NULL,
  `parking_id` int(11) NOT NULL,
  `quantity` int(11) NOT NULL,
  `status` varchar(300) NOT NULL DEFAULT 'pending',
  `special_request_id` varchar(50) NOT NULL,
  `approver_id` int(11) NOT NULL,
  `reason_status` varchar(300) NOT NULL,
  `duration` int(30) NOT NULL
)
```

- Special request table

```
CREATE TABLE `special_request` (
  `special_request_id` varchar(50) NOT NULL,
  `request_name` varchar(300) NOT NULL,
  `request_type` varchar(100) NOT NULL,
  `details` varchar(300) NOT NULL
)
```

4.4 Conclusion

This chapter has described system design. For the first part, this chapter has described about high-level design. In high-level design, it explained three sub-chapters which are system architecture, user interface design, and conceptual and logical database design. The second part which was detail design described software design and physical database design. The next chapter will discuss the implementation phase. This chapter will be divided into two parts. The first part will discuss about system project and the second part will discuss about database project. In system project, it will discuss about software development environment setup, software configuration management, and implementation status while in database project, it will discuss software development environment setup, database implementation, software configuration management and database implementation status.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

In this chapter, the task of implementing the database design is presented. The database installation and configuration procedure were shown. In the database implementation phase, MySQL is installed on Windows 10 platform. In this phase, Data Definition Language (DDL) and Data Manipulation Language (DML) in the form of SQL statements are implemented in the database. Implementation status is described for each module.

5.2 Database Installation

Installing XAMPP, an application that enables developers to build or test code based on MySQL databases on a local webserver utilizing the host's system, is required in order to install phpMyAdmin MySQL. The three systems on which XAMPP can be installed are Windows, Linux, and Mac.

Figure 5.2.1 Installation for XAMPP

5.3 Procedure using XAMPP

Starting Apache and MySQL are the two buttons required to establish a connection to the MySQL database. The names of the instances whose launch or operation was successful will be highlighted in green. To connect to the database administration or DBA, the button admin must be clicked, at which point the MySQL admin side will show up.

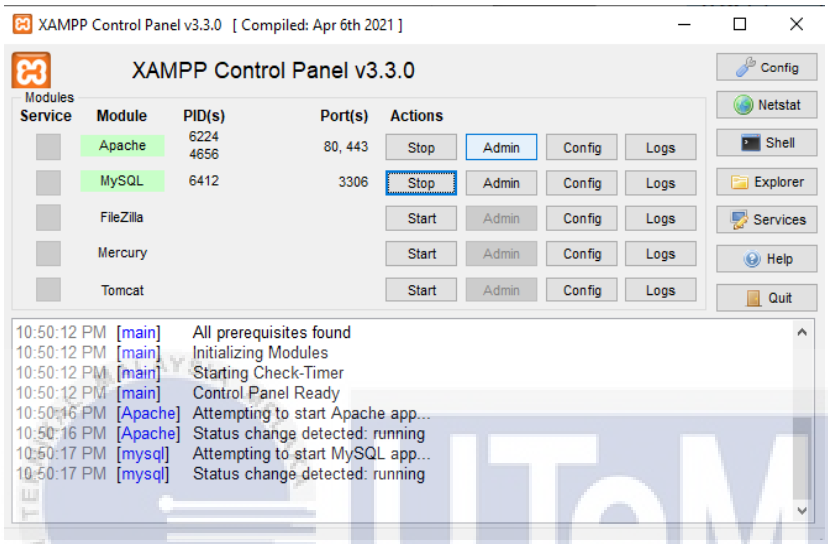


Figure 5.2.1 XAMPP Control Panel

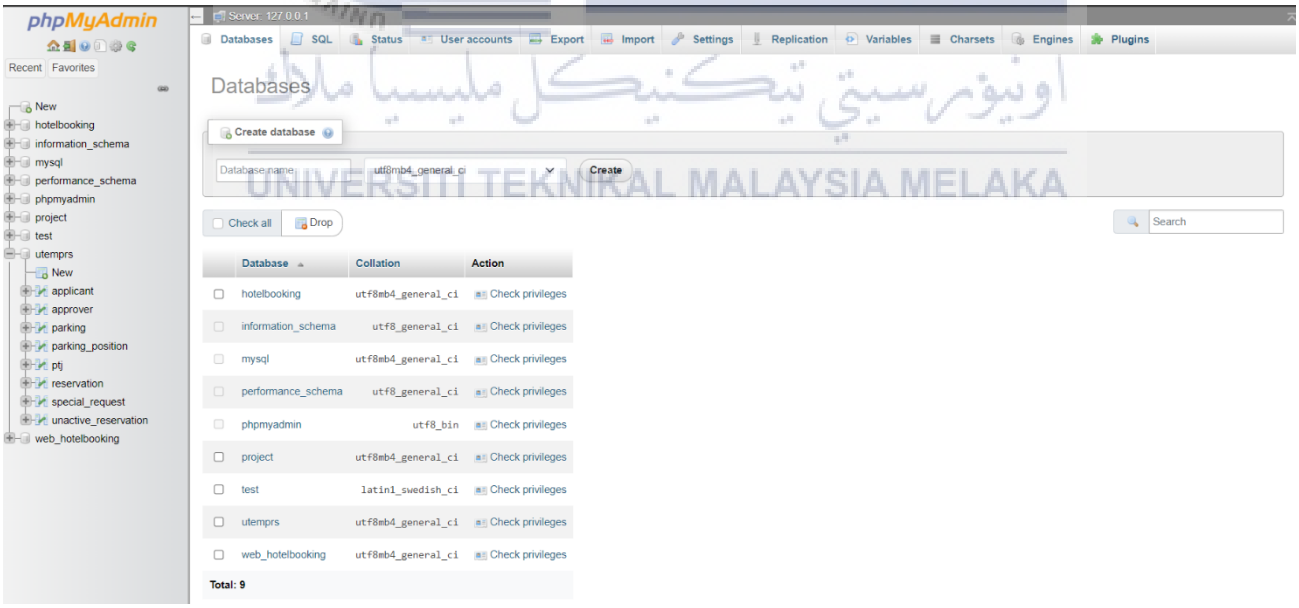


Figure 5.2.1 MYSQL Administrator is successfully connected.

The configuration of the local database that used by the UTeM EasyPark shown as below:

```php
<?php
    $server = "localhost";
    $user = "root";
    $password = "";
    $database = "utemprs";

    //create connection
    $conn = mysqli_connect($server, $user, $password, $database);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

5.4 Programming Technique

The interface of UTeM EasyPark system is built on the HyperText Transfer Protocol (HTTP), and the system's functionality was implemented in the programming language known as Hypertext Preprocessor (PHP). The flow and process load are made sure to function properly using JavaScript. The presentation of the system is described using CSS (Cascading Style Sheets), making it more aesthetically pleasing. All of the programming languages utilized in this project were selected for the system's development since they are all free and were created to be used as web scripting languages.

1. IF statement

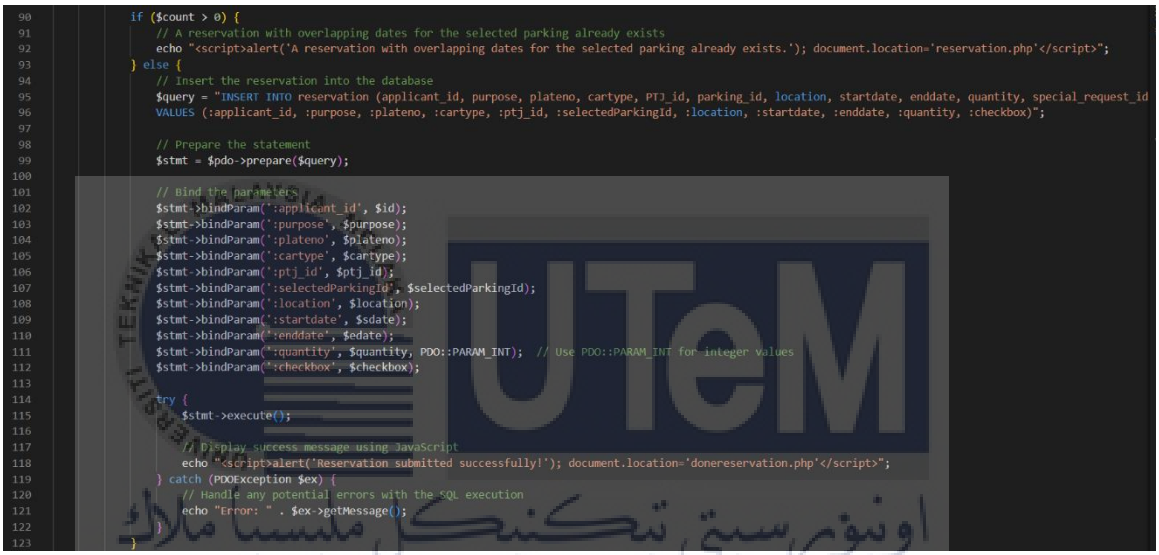
   Figure 5.4.1 shows an IF statement that will check the condition if the requirement is fulfilled then it will proceed with an output. In line 5, the variable used to check if the user clicks on the “submit” button will check the condition of whether the entered applicant email and password has in a database or not.

```php
1  <?php
2  include ("connection/connection.php");
3  session_start();
4
5  if(isset($_POST['login']))
6  {
7    $email = ($_POST['applicant_email']);
8    $pass = ($_POST['applicant_password']);
9        $query = mysqli_query($conn,"SELECT * FROM applicant WHERE applicant_email = '$email' AND applicant_password = '$pass'");
10       //$result = mysqli_fetch_array($query);
11               $result =  mysqli_num_rows($query);
12               if($result>=1)
13               {
```

Figure 5.4.1 IF statement

2. IF...ELSE statement

Figure 5.4.2 shows an IF...ELSE statement that will check the condition if requirement is fulfilled it will proceed with an output, else it will proceed with another output. In line 50, the variable used to check the condition is whether the parking reservation with the same date and parking code already exists in the database or not. If the condition is true (overlapping parking reservation with the same dates or selected parking code exists), it will display a message that alerts the user to choose another available date. If the condition is false, (there are no same dates or parking codes in the reservation table) then the new reservation will be inserted into the reservation table.



```
90          if ($count > 0) {
91              // A reservation with overlapping dates for the selected parking already exists
92              echo "<script>alert('A reservation with overlapping dates for the selected parking already exists.'); document.location='reservation.php'</script>";
93          } else {
94              // Insert the reservation into the database
95              $query = "INSERT INTO reservation (applicant_id, purpose, plateno, cartype, PTJ_id, parking_id, location, startdate, enddate, quantity, special_request_id
96              VALUES (:applicant_id, :purpose, :plateno, :cartype, :ptj_id, :selectedParkingId, :location, :startdate, :enddate, :quantity, :checkbox)";
97
98              // Prepare the statement
99              $stmt = $pdo->prepare($query);
100
101             // Bind the parameters
102             $stmt->bindParam(':applicant_id', $id);
103             $stmt->bindParam(':purpose', $purpose);
104             $stmt->bindParam(':plateno', $plateno);
105             $stmt->bindParam(':cartype', $cartype);
106             $stmt->bindParam(':ptj_id', $ptj_id);
107             $stmt->bindParam(':selectedParkingId', $selectedParkingId);
108             $stmt->bindParam(':location', $location);
109             $stmt->bindParam(':startdate', $sdate);
110             $stmt->bindParam(':enddate', $edate);
111             $stmt->bindParam(':quantity', $quantity, PDO::PARAM_INT);  // Use PDO::PARAM_INT for integer values
112             $stmt->bindParam(':checkbox', $checkbox);
113
114             try {
115                 $stmt->execute();
116
117                 // Display success message using JavaScript
118                 echo "<script>alert('Reservation submitted successfully!'); document.location='donereservation.php'</script>";
119             } catch (PDOException $ex) {
120                 // Handle any potential errors with the SQL execution
121                 echo "Error: " . $ex->getMessage();
122             }
123         }
```

Figure 5.4.2 IF...ELSE statement

3. WHILE statement

Figure 5.4.3 shows WHILE statement creates a loop that executes a specified statement as long as the test condition evaluates to true. The condition is evaluated before executing the statement. In line 286, WHILE statement used to creates loop to fetch PTJ_id and PTJ_fullname from database.

```
279   <div class="u-form-select-wrapper">
280     <select id="select-PTJ" name="ptj_id" class="u-input u-input-rectangle" required onchange="populatePositionDropdown()">
281       <option value="0">Please select</option>
282       <?php
283       // Retrieve PTJ data from the database
284       $ptjQuery = "SELECT PTJ_id, PTJ_fullname FROM ptj";
285       $ptjStmt = $pdo->query($ptjQuery);
286       while ($row = $ptjStmt->fetch(PDO::FETCH_ASSOC)) {
287         $ptjId = $row['PTJ_id'];
288         $ptjName = $row['PTJ_fullname'];
289         echo "<option value=\"$ptjId\">$ptjName</option>";
290       }
291       ?>
292     </select>
293     <svg class="u-caret u-caret-svg" ...></svg>
294   </div>
295   </div>
```

Figure 5.4.3 WHILE statement

4. Insert statement

```php
// Insert the reservation into the database
$query = "INSERT INTO reservation (applicant_id, purpose, plateno, cartype, PTJ_id, parking_id, location, startdate, enddate, quantity, special_request_id)
VALUES (:applicant_id, :purpose, :plateno, :cartype, :ptj_id, :selectedParkingId, :location, :startdate, :enddate, :quantity, :checkbox)";

// Prepare the statement
$stmt = $pdo->prepare($query);

// Bind the parameters
$stmt->bindParam(':applicant_id', $id);
$stmt->bindParam(':purpose', $purpose);
$stmt->bindParam(':plateno', $plateno);
$stmt->bindParam(':cartype', $cartype);
$stmt->bindParam(':ptj_id', $ptj_id);
$stmt->bindParam(':selectedParkingId', $selectedParkingId);
$stmt->bindParam(':location', $location);
$stmt->bindParam(':startdate', $sdate);
$stmt->bindParam(':enddate', $edate);
$stmt->bindParam(':quantity', $quantity, PDO::PARAM_INT);  // Use PDO::PARAM_INT for integer values
$stmt->bindParam(':checkbox', $checkbox);
```

Figure 5.4.4 Insert statement that will insert new data into the reservation table

5. Update statement

```php
// Prepare the update query
$query = "UPDATE approver SET approver_name = :name, approver_email = :email, approver_phoneno = :phone, address = :address, PTJ_id = :ptj WHERE
approver_email = :approver_email";

// Prepare the statement
$stmt = $pdo->prepare($query);

// Bind the parameters
$stmt->bindParam(':name', $name);
$stmt->bindParam(':email', $email);
$stmt->bindParam(':phone', $phone);
$stmt->bindParam(':address', $address);
$stmt->bindParam(':ptj', $ptj);
$stmt->bindParam(':approver_email', $approverEmail);
```

Figure 5.4.5 Update statement that will update existing approver information in the approver table

6. Delete statement

```php
// Prepare and execute the DELETE statement
$stmt = $conn->prepare("DELETE FROM parking WHERE parking_id = :parking_id");
$stmt->bindParam(':parking_id', $parking, PDO::PARAM_INT);
$stmt->execute();

// Check if the deletion was successful
if ($stmt->rowCount() > 0) {
    // Deletion successful
    echo json_encode(['status' => 'success', 'message' => 'Parking entry deleted successfully.']);
} else {
    // No matching record found
    echo json_encode(['status' => 'error', 'message' => 'Failed to delete parking entry.']);
}
```

Figure 5.4.6 Delete statement that will delete existing parking information in the parking table

7. Select statement

```
21
22  // Query to retrieve applicant data
23  $applicantQuery = "SELECT applicant.applicant_name, applicant.applicant_phoneno, applicant.applicant_email, applicant.applicant_position, ptj.PTJ_fullname
24  FROM applicant
25  JOIN ptj ON applicant.PTJ_id = ptj.PTJ_id
26  WHERE applicant.applicant_email = :applicant_email";
27
```

Figure 5.4.7 Select statement that used to retrieve reservation information

8. Join tables

Figure 5.4.8 shows a JOIN statement SQL query used to retrieve reservation information from reservation table where it join from other tables which are applicant, PTJ, parking, and approver. The reservartion information will be display according to email session of the applicant.

```
23  // Query to retrieve reservation data
24
25  $reservationQuery = "SELECT applicant.applicant_name, applicant.applicant_phoneno, applicant.applicant_email, applicant.applicant_position, ptj.PTJ_fullname,
26  reservation.purpose, reservation.plateno, reservation.cartype, reservation.location, reservation.startdate, reservation.enddate, parking.parkingcode,
27  reservation.quantity, reservation.status, reservation.reason_status, reservation.duration, approver.approver_name, approver.approver_position
28  FROM reservation
29  JOIN applicant ON reservation.applicant_id = applicant.applicant_id
30  JOIN ptj ON reservation.PTJ_id = ptj.PTJ_id
31  JOIN parking ON reservation.parking_id = parking.parking_id
32  JOIN approver ON reservation.approver_id = approver.approver_id
33  WHERE applicant.applicant_email = :applicant_email";
34
```

Figure 5.4.8 Join table

9. Submit button function

Verify the "submit" button has been submitted via a POST request. If yes, it will proceed to insert all the reservation information into reservation table in database.

```
48  if (isset($_POST['submit'])) {
49      // Retrieve form inputs
50      $name = $_POST['name'];
51      $phone = $_POST['phone'];
52      $email = $_POST['email'];
53      $position = $_POST['position'];
54      $purpose = $_POST['purpose'];
55      $plateno = $_POST['plateno'];
56      $cartype = $_POST['cartype'];
57      $ptj_id = $_POST['ptj_id'];
58      $location = $_POST['location'];
59      $sdate = date("Y-m-d", strtotime($_POST['startdate']));
60      $edate = date("Y-m-d", strtotime($_POST['enddate']));
61      $quantity = isset($_POST['quantity']) ? $_POST['quantity'] : 0;  // Set a default value if quantity is not set
62      $selectedParkingIds = isset($_POST['selectedParkingId']) ? $_POST['selectedParkingId'] : [];
63      $checkbox = (!empty($_POST['checkbox'])) ? implode(",", $_POST['checkbox']) : '';
64
```

Figure 5.4.9 Submit function

5.5 Error Handling

Error handling refers to the response and recovery procedures from system error conditions. Error management is necessary to guarantee that users can understand the flow of the system through a message or output. It helps the program's regular execution flow continue. In UTeM EasyPark there are a few errors handling has been implemented.

i. Figure 5.5.1 shows the login form for the applicant, all the fields need to be filled up if one of the fields is blank some alert message will appear.

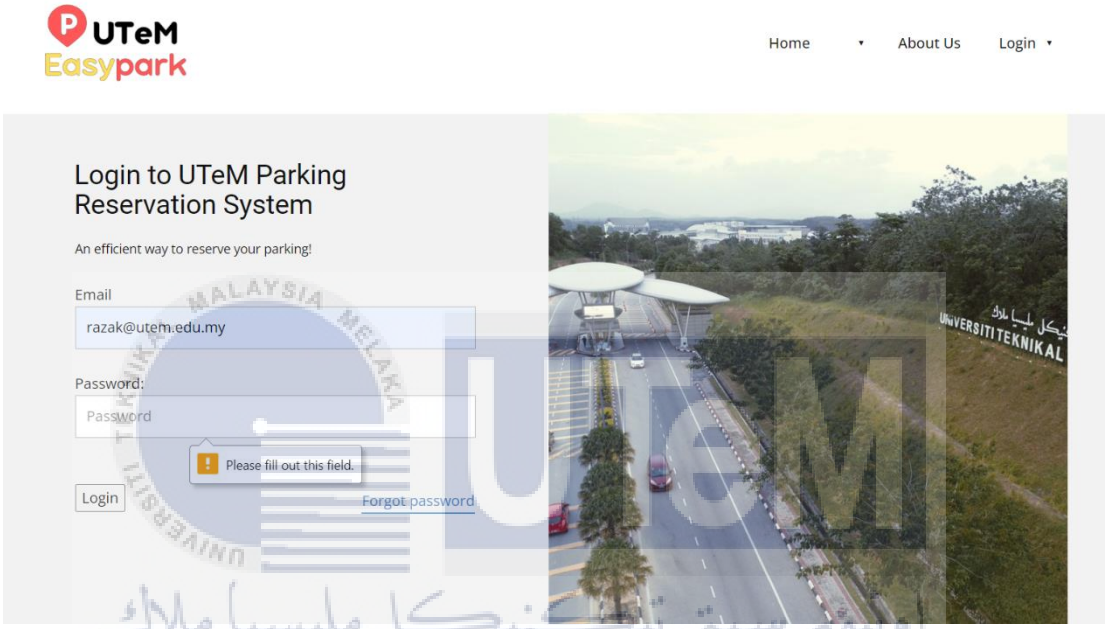


Figure 5.5.1 Applicant Login Page

ii. Figure 5.5.2 shows the alert message to let the user know their email or password might be wrong.

```php
if(isset($_POST['login']))
{
  $email = ($_POST['applicant_email']);
  $pass = ($_POST['applicant_password']);
      $query = mysqli_query($conn,"SELECT * FROM applicant WHERE applicant_email = '$email' AND applicant_password = '$pass'");
      //$result = mysqli_fetch_array($query);
            $result =  mysqli_num_rows($query);
            if($result>=1)
            {

                $_SESSION['applicant_email'] = $email;
                $_SESSION['applicant_password'] = $pass;
                $_SESSION['valid'] = true;
                $_SESSION['timeout']= time();
                echo "<script>alert('You have entered valid username and password. Welcome to UTeM EasyPark');document.location='home.php'</script>";
            }
            else
            {
                echo "<script>alert('Your email or password might be wrong, please try again!');document.location='lecturerlogin.php'</script>";
            }
```

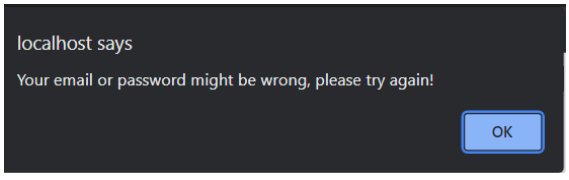


Figure 5.5.2 Alert message at Applicant Login Page

iii. Figure 5.5.3 shows some error messages that remind the applicant to select an end date greater than the start date and the system will automatically reset the date, and the applicant should select the right date which is end date must be greater than the start date.

```
338
339         <div class="u-form-date u-form-group u-form-partition-factor-3 u-form-group-9">
340             <label for="date-end" class="u-label u-text-grey-40">End Date of Reservation</label>
341             <input type="date" placeholder="YYYY-MM-DD" id="date-end" name="enddate" class="u-input u-input-rectangle">
342             <p id="error-message" style="color: red; display: none;">End date cannot be before start date.</p>
343         </div>
344
```

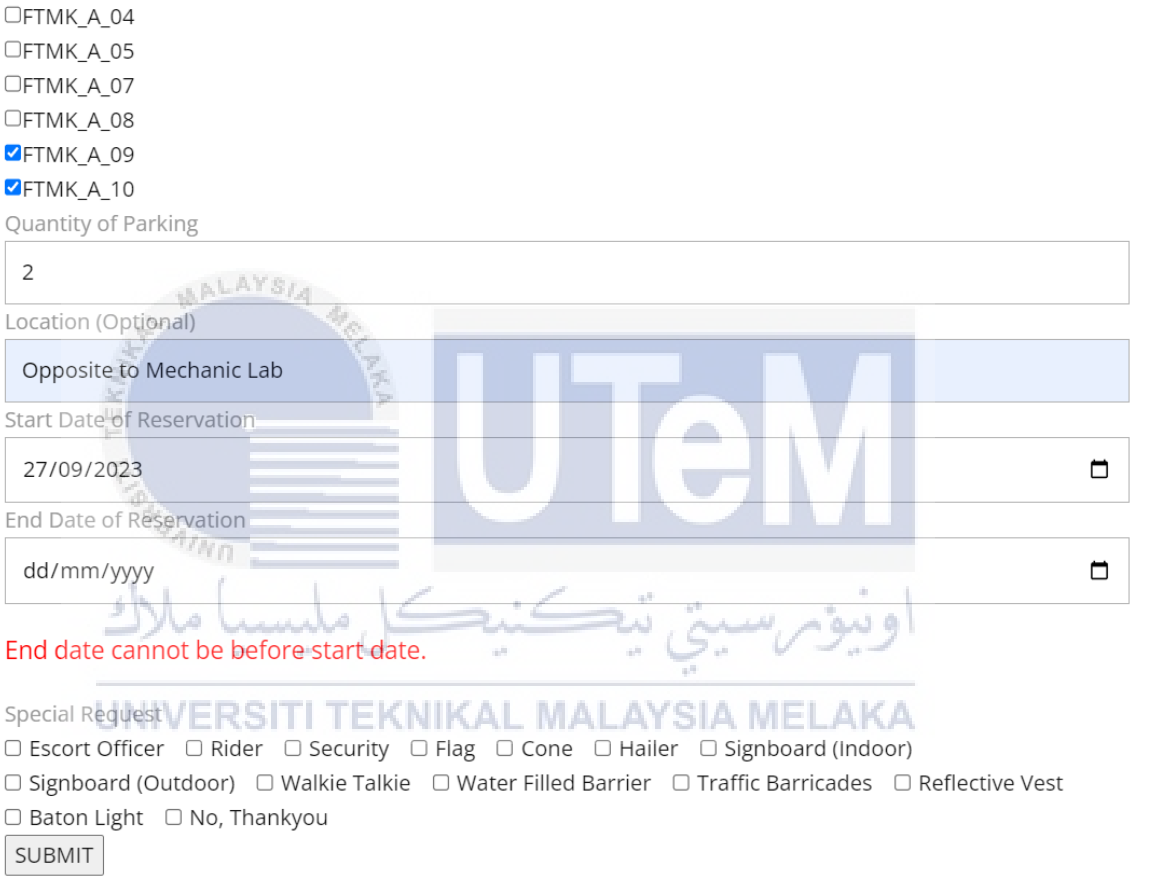
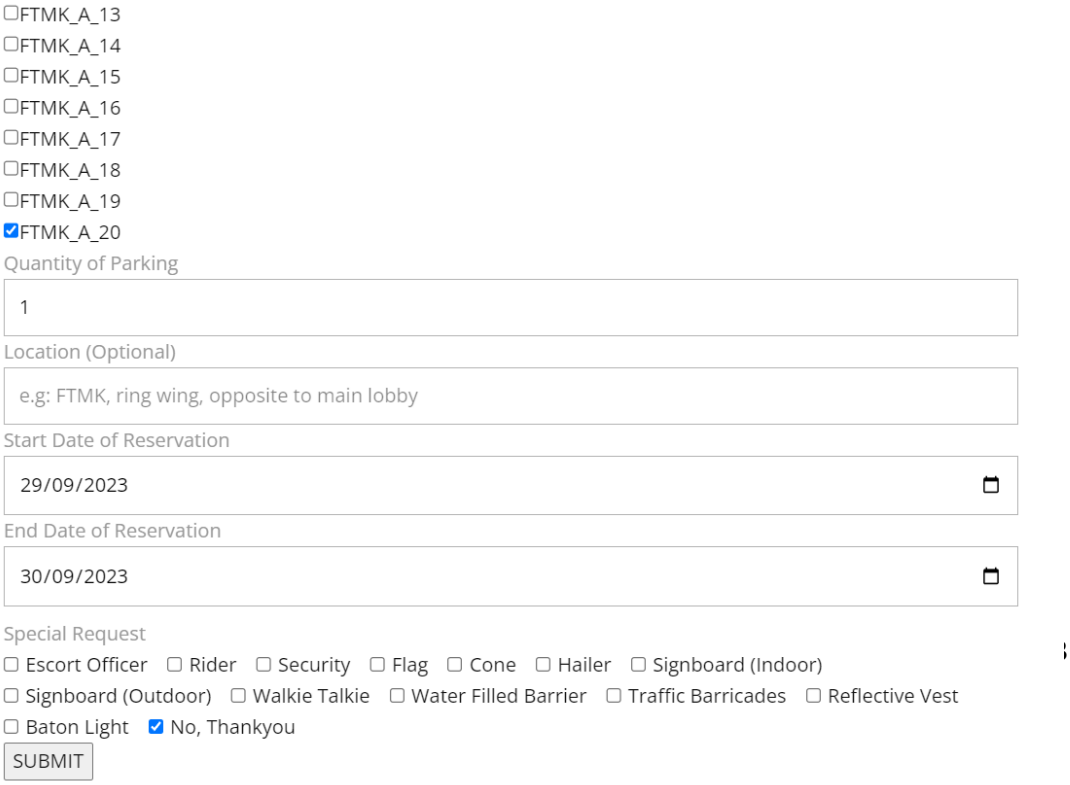


Figure 5.5.3 Alert message in red text to make let user alert before selecting the end date.

iv. Figure 5.5.4 shows Add PTJ Information page, where the PB staff can insert PTJ information.

Figure 5.5.4 Alert message at Add PTJ Page

v. Figure 5.5.5 shows the Reservation Page. The applicant needs to fill up all the fields. However, an error message will appear after the applicant clicks on the "submit"



button. The purpose is to let the applicant know where their selected date and parking code are not available since there are overlapping parking reservations with other applicants. Then, the applicant needs to choose another available parking code, and date.

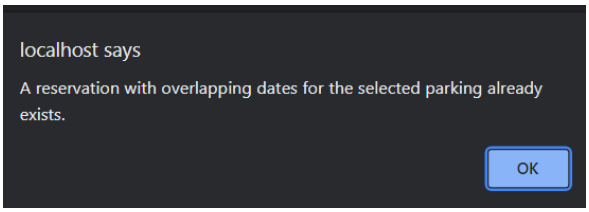


Figure 5.5.5 Alert message at Reservation Page

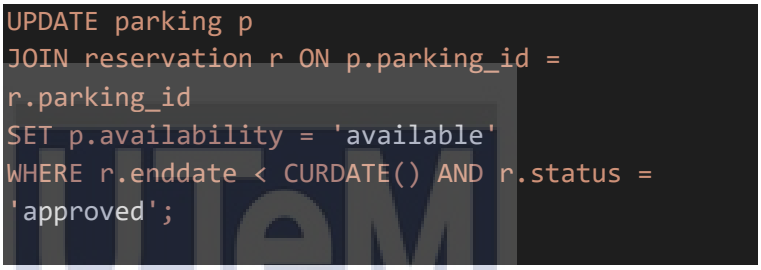
5.6 Triggers

Explanation	Query

It is used to calculate the duration of parking reservations based on the difference between the start date and end date selected by the applicant.	`DELIMITER $$`
	`CREATE TRIGGER `calculate_duration` BEFORE`
	`INSERT ON `reservation` FOR EACH ROW BEGIN`
	`    SET NEW.duration = DATEDIFF(NEW.enddate,`
	`NEW.startdate) + 1;`
	`END`
	`$$`
	`DELIMITER ;`

It is used to set the default quantity for parking quantity in the reservation table.	`DELIMITER $$`
	`CREATE TRIGGER `set_default_quantity` BEFORE`
	`INSERT ON `reservation` FOR EACH ROW BEGIN`
	`    SET NEW.quantity = 1;`
	`END`
	`$$`
	`DELIMITER ;`

It is used to set the status of a parking reservation as “pending” before the approver can verify or update the status of the parking reservation	`DELIMITER $$`
	`CREATE TRIGGER `set_reservation_status_pending``
	`BEFORE INSERT ON `reservation` FOR EACH ROW`
	`BEGIN`
	`    SET NEW.status = 'pending';`
	`    SET NEW.approver_id = 3;`
	`END`
	`$$`
	`DELIMITER ;`

It is used to automatically update parking availability for approved status reservations, and update parking availability from “available” to “not available”	
It is used to update parking availability when a parking reservation made by the applicant has ended. The system will update from “not available” to “available” if the end date is less than the current date and the status of the reservation is “approved”	

5.7 Procedure

Explanation	Query

It is used to calculate the available parking spaces based on the total parking spaces and occupied parking spaces.	DELIMITER $$
	
	CREATE PROCEDURE CalculateAvailableParkingSpaces(IN PTJ_ID INT)
	BEGIN
	    SELECT
	        PTJ_id,
	        total_parking_spaces,
	        occupied_parking_spaces,
	        total_parking_spaces - occupied_parking_spaces AS available_parking_spaces
	    FROM
	        parking
	    WHERE
	        PTJ_id = PTJ_ID;
	END$$
	
	DELIMITER ;

It is used to update parking reservation where end dates have passed from "approved" or "cancel" to "unactive". The "unactive" reservation will be moved to the "unactive_reservation" table.	DELIMITER $$
	
	CREATE PROCEDURE UpdateAndMoveToUnactive2()
	BEGIN
	    -- Update reservations with 'approved' or 'cancel' status to 'unactive' where end date has passed
	    UPDATE reservation
	    SET status = 'unactive'
	    WHERE (status = 'approved' OR status = 'cancel') AND enddate < CURDATE();
	
	    -- Move updated reservations to the "UNACTIVE_RESERVATION" table
	    INSERT INTO UNACTIVE_RESERVATION
	    SELECT * FROM reservation WHERE status = 'unactive';
	
	    -- Delete moved reservations from the "reservation" table
	    DELETE FROM reservation WHERE status = 'unactive';
	END;
	
	DELIMITER ;

CHAPTER 6

TESTING

6.1 Introduction

Testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software meets the specified requirements or not and to identify the defects to ensure that the system is defect-free. In this chapter, verification and validation will be done on UTeM EasyPark (UTeM Parking Reservation System) .

The two main goals of testing UTeM EasyPark (UTeM Parking Reservation System) are:

i. To demonstrate UTeM EasyPark (UTeM Parking Reservation System) to the end users has met its user requirement

ii. To discover any bugs or faults in the UTeM EasyPark (UTeM Parking Reservation System) with different test strategy

Furthermore, system testing is an important phase in Database Life Cycle (DBLC). UTeM EasyPark (UTeM Parking Reservation System) testing phase consists test plan which includes test organization, test schedule and test environment.

6.2 Test Plan

In a technical document known as a test plan, the test strategy, scope, testing resources (including personnel, software, tools, and hardware), test timetable, and test deliverables are all described. The test plan comprises detailed explanations of the system's operations and processes, as well as instructions on how each will be tested to see if the system works as intended, to find bugs, and to figure out its true limitations.

6.2.1 Test Organization

The UTeM EasyPark (UTeM Parking Reservation System) test organization has two users: PB (Polis Bantuan) staff and Lecturers or Staff of UTeM. Each user will be tested for functional and non-functional needs.

Table below shows the individuals who are currently involved in the testing phase of the UTeM EasyPark (UTeM Parking Reservation System).

TesterID	Users	Responsibilities
T1	PB (Polis Bantuan) staff (Cik Syira)	• Testing the system by following the test script given • Testing the PB staff module • Defect and bug detection
T2	Lecturer (TS. Abd Razak bin Hussain)	• Testing the system by following the test script given • Testing the Lecturer module • Defect and bug detection

Table 6.2.1 User Responsibilities List

6.2.1 Test Organization

Test Environment is a setup of software, hardware, operating system, tool required, and network configuration for the testing teams to execute test cases. In table 6.2 and 6.3 show the hardware and software list required for UTeM EasyPark (UTeM Parking Reservation System) test environment.

Environment Specification	Description
Laptop	Acer Nitro 5 AN515-45-R7ND Gaming Notebook
Processor	AMD Ryzen 5 5600H with Radeon Graphics
Keyboard and Mouse	HP GK1100 Gaming Keyboard and Mouse
Random Access Memory (RAM)	8 GB

Table 6.2 Test Environment Hardware List

Environment	Description
Database (MySQL)	To manage data in the database table that runs on a server.
Web Server (XAMPP Server)	To create a local web server and use to deployment and testing purposes.
Operating System / Platform (Windows 11 Home Single Language)	To manage computer hardware and software resources and then provide the service or tool for computer program operation.
Web Browser (Google Chrome)	To run the PHP source code and test the system interface functionality.
Microsoft Visual Studio Code	To write PHP code
Microsoft Word	To write a final report and make aslide for presentation.

Table 6.2 Test Environment Software List

6.2.3 Test Schedule

A test schedule was defined as a timetable summary that included important test milestones based on the function's complexity. Scheduling is a test activity that depends on the date for the testing system to be completed. When creating the test schedule, dates are estimated and then revised as necessary. The tester needs to become familiar with the model's documentation. The tester then contrasts the actual system activities with those predicted by the model.

Testing Task	Description	Start Date	End Date
1. Unit Testing	Test that ensures that the modules involved meet the requirement.	1/9/2023	3/9/2023
2. Integration testing/ system testing	Demonstrate the system's modules to ensure successful performance.	4/9/2023	14/9/2023
3. Acceptance	To confirm that the system meets its requirement.	15/9/2023	18/9/2023

6.3 Test Strategy

Test strategy is important because it enables the developer to use different testing t techniques that are suitable to the testing phase. It is also necessary for guidelines that explain test design and determine how testing needs to be done, the technique to be used and which modules to test.

The level of confidence that the program satisfies the model is based on the completeness and correctness of the derived tests with respect to the model and the completeness and consistency between the model and the implementation. Test strategies are divided into two basic testing. There are:

1. Black-Box Testing

The software testing technique known as "black box testing" is used to test the software without knowing the internal structure of the code. Black box testing does not involve implementation or programming skills because this form of testing is typically carried out by the tester. Black Box testing primarily focuses on the system's functioning, including both functional and non-functional criteria. Additionally, this testing approach is frequently used in high-level testing, including system and user acceptance testing.

2. White-Box Testing

White Box testing is a type of software testing where the tester is aware of the internal workings of the product. Software developers implement white box testing, therefore implementation and programming expertise are required. White box testing concentrates on testing the programming code of the system, such as the branch and code structure. Additionally, this testing technique is frequently applied to high-level testing, including unit and integration testing.

6.3.1 Classes of Test

This UTeM EasyPark (UTeM Parking Reservation System) phases present classes that have been involved in the testing procedure and tasks. Below are components of testing:

1. Unit Testing

   Unit testing is a level of software testing where individual units/ components of a software are tested and the purpose is to validate that each unit of the software performs as designed. In UTeM EasyPark (UTeM Parking Reservation System) the unit modules like PB (Polis Bantuan) staff login, and Lecturer login were tested to ensure the modules function well as designed. Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. It is also known as white box testing, this testing is applied to UTeM EasyPark (UTeM Parking Reservation System) where the tester can test for a particular module.

2. System Testing

   System testing is a level of software testing in which whole and integrated software is tested. This test's objective is to ascertain whether the system complies with the specified requirements and whether each system's handling of the data is correct.

3. Performance Testing

   The system doesn't require an endless supply of time or resources to function. There are times when system design errors that result in decreased system performance are referred to as "performance bugs". Performance testing can be used to find performance bottlenecks, compare and assess performance, and more.

4. Error Handling Test

   This testing is used to validate that only the correct and accurate data input from the user can be filled into the input field. For example, in UTeM EasyPark (UTeM Parking Reservation System), for the login module, both users which are Polis Bantuan Staff or the Lecturer need to input the right email type that includes “@” and password. This testing was implemented to ensure the user enters the correct data before the data is stored in the database and an error message will be popped on the screen to inform the user regarding null input.

5. Security Testing

   Security tests will be conducted to validate and verify the user email and password during the login process. In UTeM EasyPark (UTeM Parking Reservation System), this system was tested for both users at login sessions where only the right and match email and password only are allowed to enter the system.

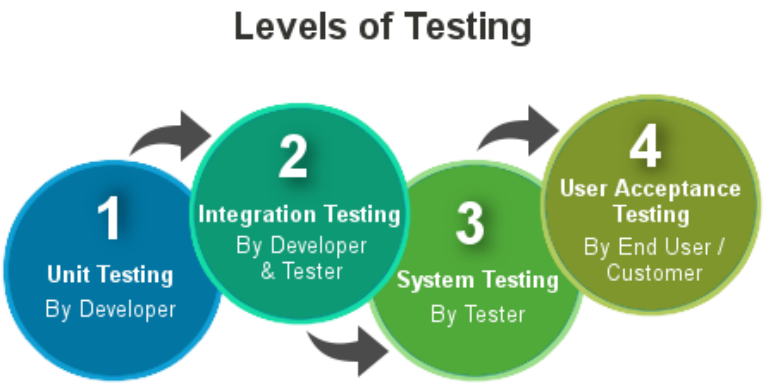
6. Integration Testing

   The integration test is to ensure that this system captures data into the database correctly based on the key in the data. It is done by moving through each and every menu item in the interface. Therefore, this testing was implemented in UTeM EasyPark (UTeM Parking Reservation System) to ensure the data are stored correctly and focus on the data transfer among the modules. For example, Polis Bantuan staff as approvers have th

7. User Acceptance Testing

   This test is carried performed prior to the final user acceptance of this system. This is done to make sure that the user and the entire system work properly. The user acceptance test process is identical to that of the system and integration tests.

	White Box	Black Box
Type of test	<u>Unit Testing</u> - Individual units / components of a software are tested <u>Integration Testing</u> - Focused on data transfer among modules	Unit Testing Integration Testing <u>System Testing</u> - Check the complete end to end scenarios <u>User accetance testing</u> - To check whether the system meet the user requirements
Test Design Techniques	• Fault Insertion • Error Handling • String testing • Statement Coverage • Decision Coverage • Condition coverage • Path coverage	• Equivalence Analysis • Boundary Analysis • Path Analysis • Use case • Decision tables



The type of testing and test design method in between white box and black box. In UTeM EasyPark (UTeM Parking Reservation System), white box testing and black box testing were implemented. In white box testing, unit testing and integration testing will be used to concentrate on the correct data input type and the data stored and transferred among modules. In black box testing, equivalence partitioning analysis will be used to test the divides of the input data of a software.

6.4 Test Design

The test description and test data are explained by the test design. The test description contains a list of the tasks necessary for the test design as well as information on the test data. The user acceptance test is explained in the testing description together with the test cases, expected outcomes, and test data.

6.4.1 Test Description

Every module test case has a test case description that explains the test case identification, kind of testing, pre-condition, test requirements, step procedure in each test case identification, and expected output result.

6.4.2 Test Data

Real-world data and artificial data can both be chosen and used as test data during testing. All of the data sets used in regular business operations are considered real-life data. It is simpler for the user to defend the accuracy of the system flow and output when real-world data is used. In the testing phase of the development process, developers create synthetic data, which is unreal data. Based on this system design, a sample of input data has been described.

1. Login Module

Test Id	TC-001- Polis Bantuan Login (admin)			
Testing Type	Unit testing and integration testing			
Test Strategy	White box testing			
Test Class	Security and error handling testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC01_1	Validate the login function if the email and password provided are valid.	The user has a valid email and password	1. Navigate to the loginpage. 2. Provide valid email. 3. Provide a validpassword. 4. Click on the login button.	Login successful and the system redirects the user to the admin homepage
TC01_2	Validate the login function is not been filled up.		1. Navigate to the login page. 2. Click on the loginbutton	Login failed.The system displays error message **"**Your email or password might be wrong, please try again!**"**
TC01_3	Validate the login function is incorrect username and password		1. Navigate to login page 2. Provide valid username 3. Provide invalid password 4. Click on Login button	Login failed. System display error message **"**Your email or password might be wrong, please try again!**"**

Table 19: Test description for PB login

1. Parking Reservation module

Column Name	Test Data 1	Test Data 2	Test Data 3
Test Case ID	TD01_1	TD01_02	TD01_3
Name	TS. Abdul Razak bin Hussain	DR. Syahida bt Mohtar	TS. Abdul Razak bin Hussain
Phone Number	+6062702452	+6062702482	+6062702452
Email	razak@utem.edu.my	syahida@utem.edu.my	razak@utem.edu.my
Position	Pensyarah Kanan	Pensyarah Kanan	Pensyarah Kanan
Staff's Faculty	Fakulti Teknologi Maklumat Dan Komunikasi	Fakulti Teknologi Maklumat Dan Komunikasi	Fakulti Teknologi Maklumat Dan Komunikasi
Purpose of Reservation (Optional)	Career Fair at FTMK	Innovation Day	Career Fair at FTMK
Plate Number (Optional)	MDS 3183	MDG 8721	MDS 3183
Type of Car (Optional)	Toyota Yaris	Honda City	Toyota Yaris
PTJ / Centres / Faculties	Fakulti Teknologi Maklumat Dan Komunikasi	Fakulti Teknologi Maklumat Dan Komunikasi	Fakulti Teknologi Maklumat Dan Komunikasi
Parking Position	Left-wing of faculty	Left-wing of faculty	Left-wing of faculty
Parking Code	FTMK_A_01 FTMK_A_02	FTMK_A_02	FTMK_A_01 FTMK_A_02
Quantity of Parking	2	1	2
Location	Opposite to main lobby	Opposite to main lobby	Opposite to main lobby
Start Date of Reservation	2023-09-29	2023-09-29	2023-09-31
End Date of Reservation	2023-09-30	2023-09-30	2023-09-25
Special Request	Cone	Cone	Cone
Result Test Data	Parking reservation successfully saved in database.	Parking reservation was unsuccessfull due to overlapping parking reservation.	Parking reservation failed because start date is greater than end date.

1. Login Module

Column Name	Test Data 1	Test Data 2	Test Data 3
Test Case ID	TD02_1	TD02_2	TD02_3
Email	izrin@utem.edu.my	izrin@utem.edu.my	izrinutem.edu.my
Password	Izr!n123	Qweert132	Izr!n123
Result Test Data	Applicant login successfully	Applicant login failed because the password does not match the stored password in database.	Applicant login failed because username and password does not match the stored in database.

User: Applicant

Module: Login

Test Scenario: Verify the user can log in with the correct credentials.

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC1: Check a user with the correct applicant_email and password should be allowed to log in.	1. Insert applicant_email 2. Insert password Click "login" button	Applicant_email: msyahida@utem.edu.my password: sy@hida123	Display the message "You have entered valid username and password. Welcome to UTeM EasyPark"	Display the message "You have entered valid username and password. Welcome to UTeM EasyPark"
TC2: Check a user with correct applicant_email and incorrect password should not be allowed to log in	1. Insert applicant_email 2. Insert incorrect password 3. Click "login" button	Applicant_email: msyahida@utem.edu.my password: qwerrrt	Display error message "Your email or password might be wrong, please try again!"	Display error message "Your email or password might be wrong, please try again!"
TC3: Input an empty data	1. Make sure applicant_email and password field are empty. Click "login" button	Applicant_email: - Password: -	Display the message "Please fill out this field"	Display the message "Please fill out this field"

User: Applicant

Module: Reservation

Test Scenario: Verify the end date of parking reservation is greater than start date of parking reservation

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC 1: Verify the end date is greater than the start date	1. Select start date 2. Select end date	Start date: 24/09/2023 End date: 27/09/2023	There is no message appear, reservation page remains.	There is no message appear, reservation page remains.
TC 2: Verify the end date is less than the start date	1. Select start date 2. Select end date	Start date: 25/09/2023 End date: 23/09/2023	Display the error message "End date cannot be before the start date."	Display the error message "End date cannot be before the start date."

User: Applicant

Module: Reservation

Test Scenario: To insert reservation information into the reservation table

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC1: TC1: Check all the text boxes are filled up	1. Display applicant information using session 2. Fill up all reservation information 3. Select PTJ, parking position from dropdown listbox 4. Parking codes will be displayed based on selected PTJ, and parking position 5. Select parking codes from checkbox 6. Display quantity based on selected parking codes 7. Select the start and end date from the calendar 8. Select special request from	Name: DR. SYAHIDA BINTI MOHTAR Phone: +6062702482 Email: msyahida@utem.edu.my Position: PENSYARAH Staff's Faculty: Fakulti Teknologi Maklumat Dan Komunikasi Purpose: Career Day at FTMK Plate No: MDG 5142 Type of Car: Toyota Vios PTJ: Fakulti Teknologi Maklumat Dan Komunikasi Parking Position: Centre of faculty Parking Code: FTMK_A_11 Quantity of Parking:1 Location: Opposite to main lobby	Display the message "Reservation submitted successfully!"	Display the message "Reservation submitted successfully!"

	checkbox 9. Click on “submit” button	Start date: 24/09/2023 End date: 27/09/2023 Special Request: No, thankyou		
TC2: TC 2: Leave one box empty	1. Display applicant information using session 2. Fill up all reservation information 3. Select PTJ, parking position from dropdown listbox 4. Parking codes will be displayed based on selected PTJ, and parking position 5. Select parking codes from checkbox 6. Display quantity based on selected parking codes 7. Select the start and end date from the calendar 8. Select special request from checkbox 9. Click on “submit” button	Name: DR. SYAHIDA BINTI MOHTAR Phone: +6062702482 Email: msyahida@utem.edu.my Position: PENSYARAH Staff’s Faculty: Fakulti Teknologi Maklumat Dan Komunikasi Purpose: Career Day at FTMK Plate No: MDG 5142 Type of Car: Toyota Vios PTJ: Fakulti Teknologi Maklumat Dan Komunikasi Parking Position: Centre of faculty Parking Code: FTMK_A_11 Quantity of Parking:1 Location: Opposite to main lobby Start date: 24/09/2023 End date: 27/09/2023 Special Request: _____	Display the error message “Please fill up all the text box in reservation form.”	Reservation page become blank. No message “Please fill up all the text box in the reservation form”

User: Applicant

Module: Reservation

Test Scenario: To verify if there any overlapping parking reservations with same parking code, start date and end date.

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC1: Check parking reservation with same parking code	1. Display applicant information using session 2. Fill up all reservation information 3. Select PTJ, parking position from dropdown listbox 4. Parking codes will be displayed based on selected PTJ, and parking position 5. Select parking codes from checkbox 6. Display quantity based on selected parking codes 7. Select the start and end	Name: TS. SYARIFFANOR BINTI HISHAM Phone: +6062702482 Email: syariffanor@utem.edu.my Position: PENSYARAH Staff's Faculty: Fakulti Teknologi Maklumat Dan Komunikasi Purpose: Innovation Day at FTMK Plate No: WUW 7865 Type of Car: Minicooper PTJ: Fakulti Teknologi Maklumat Dan Komunikasi	Display the message "A reservation with overlapping dates for the selected parking already exists", and back to reservation page.	Display the message "A reservation with overlapping dates for the selected parking already exists", and back to reservation page.

	date from the calendar 8. Select special request from checkbox 9. Click on “submit” button	Parking Position: Centre of faculty Parking Code: FTMK_A_11 Quantity of Parking:1 Location: Opposite to main lobby Start date: 24/09/2023 End date: 27/09/2023 Special Request: No, thankyou		

User: PB's Officer

Module: Login

Test Scenario: Verify the user can login with the correct credentials.

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC1: Check a user with the correct approver_email and password should be allowed to log in.	1. Insert approver_email 2. Insert password 3. Click "login" button	applicant_email: aidil@pb.utem.edu.my password: 5757	Display the message "You have entered valid username and password. Welcome to UTeM EasyPark"	Display the message "You have entered valid username and password. Welcome to UTeM EasyPark"
TC2: Check a user with the correct approver_email and incorrect password should not be allowed to log in	1. Insert applicant_email 2. Insert incorrect password 3. Click "login" button	applicant_email: aidil@pb.utem.edu.my password: qwexd	Display error message "Your email or password might be wrong, please try again!"	Display error message "Your email or password might be wrong, please try again!"
TC3: Input an empty data	2. Make sure applicant_email and password field are empty. 3. Click "login" button	applicant_email: - password: -	Display the message "Please fill out this field"	Display the message "Please fill out this field"

User: PB's Officer

Module: Edit Profile

Test Scenario: Verify the user information can be updated in the database

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC1: The user should be able to update their profile	1. Click on "Account" at the right sidebar to view the profile 2. Click on "edit" button 3. Update phone no 4. Update address 5. Click on "save changes" button	Full Name: Aidil Izzudin bin Ithnin Email: aidil@pb.utem.edu.my Phone: 0126180421 Address: No 20 Jalan Cempaka 6, Taman Seri Cempaka 75460 Melaka In charge in PTJ: 5 Full Name: Aidil Izzudin bin Ithnin Email: aidil@pb.utem.edu.my **Phone: 0178824630** **Address: C923-1 KM 11 Jalan Pernu 4 Kampung Pernu 75460 Melaka** In charge in PTJ: 5	Display the message "Your information updated successfully"	Display the message "Your information updated successfully"

User: PB's Officer

Module: PTJ Information

Test Scenario: To insert PTJ information into the PTJ table

Test Case	Flow Step	Test Data	Expected Result	Actual Result
TC1: Verify the PTJ information saved into the database	1. Enter PTJ Full Name 2. Enter PTJ Initial Name 3. Enter Address 4. Upload image 5. Click on "submit" button	PTJ Full Name: Fakulti Teknologi Maklumat dan Komunikasi PTJ Initial Name: FTMK Address: Durian Tunggal Image: [BLOB - 5.6 MiB]	Display the message "PTJ information added successfully"	Display the message "PTJ information added successfully"

6.6 Conclusion

The testing strategy for this system is described in this chapter. In order to manage project cycles and length, the testing environment and test schedule were explained. This part also covered how the chosen approach was used in conjunction with test case identification, test cases, and test results as well as analysis, which indicated if the system was successful or not. The project's conclusion, together with observations on its strengths, faults, and suggestions for project improvement, will be covered in the following chapter.

CHAPTER 7

PROJECT CONCLUSION

7.1 Introduction

This chapter will be discussing about the overall conclusion for the UTeM EasyPark (UTeM Parking Reservation System) which will consist of a strengths and weaknesses analysis of the system. Besides that, propositions on improvements according to the strength and weakness analysis, and contributions to this project will be defined and elaborated in this chapter as well.

7.2 Observation of strengths and weakness

This chapter will discuss the human resources system's problems and strengths for UTeM EasyPark (UTeM Parking Reservation System). As the adage goes, there must be both strengths and weaknesses for each application or system created for a certain reason. This should not, however, be used as a justification for doing subpar work. The output's strengths are something to be proud of and indicate a job well done. If there are any shortcomings, these must be fixed in the future for the result to be much more significant.

7.3 System Strength

The strengths of UTeM EasyPark (UTeM Parking Reservation System) are:

a. **Human Error Reduction**

   UTeM EasyPark (UTeM Parking Reservation System) implemented the trigger and stored procedure functional to calculate the duration of parking reservations and manage their status significantly reduces the risk of human errors. By setting reservations to "pending" until they are verified by the appropriate authority, the system ensures a methodical and accurate approval process, minimizing potential discrepancies. Furthermore, the automatic update of parking availability, shifting reservations from "not available" to "available" upon meeting specific conditions, prevents human oversight, guaranteeing that accurate parking information is consistently presented. Additionally, the automated calculation of available parking spaces based on occupancy data diminishes the possibility of inaccuracies that could

arise from manual computations. Lastly, the systematic update of reservation statuses from "approved" or "cancel" to "unactive," followed by their appropriate relocation to a designated table, is conducted seamlessly by the system, eliminating potential human oversights and data handling errors. Overall, the automation of these processes significantly mitigates human-related errors and ensures a reliable and efficient parking reservation system.

b. Interface

An intuitive and user-friendly interface makes it easy for users to navigate and interact with the system, enhancing overall usability. It also has a well-designed interface that allows users to quickly and efficiently reserve parking spaces, streamlining the reservation process. Visual cues, such as color-coded indicators or graphics, provide a clear representation of parking spaces, buildings, road, and others which aids users in making informed decisions.

c. Time-Saving

Users can save valuable time by knowing in advance if parking spots are available, avoiding unnecessary delays and reducing stress associated with parking.

d. Optimized Resource Utilization

The system optimizes the use of parking resources by accurately calculating available parking for each PTJ, preventing overbooking or underutilization of parking spaces.

e. Reduced Congestion

By enabling reservations and efficiently managing parking, the system helps reduce traffic congestion around UTeM areas, creating a smoother traffic flow.

f. Improved User Experience

Users benefit from a seamless parking experience, knowing they have a guaranteed parking spot, which positively impacts their overall perception and satisfaction with UTeM's parking facilities.

g. **Real-Time Insights**

The system provides real-time data on parking availability, allowing users to make informed decisions based on current parking conditions.

h. **Prevention of Overlapping Reservation**

By accurately calculating available parking spaces for each PTJ, the system minimizes the possibility of overlapping reservations, ensuring a fair and organized allocation of parking spots.

7.4 System Weakness

The weakness of UTeM EasyPark (UTeM Parking Reservation System) are:

a. Recover and backup features

This system can be improved by an automatic backup function. For example, automatic backup will be done at 1200 am every day to ensure the backup data is up to date. Polis Bantuan staff can manage the frequency of automatic backup to daily, weekly, and monthly based on the system requirements, and the system will recover automatically when the system is corrupted.

b. Lack of notification reminder

Notification reminder for the applicant by email or SMS when the parking reservation has been approved or rejected. For example, the system will send a notification to the applicant by email to let the user know whether their parking application has been approved or not. It would be good if this system also implemented an email notification before one day of the reservation session.

c. Forgot password function

This system can be improved by the forgot password function. For the user who forgets the password, the system will send a new password to their email. Once the user gets the new password, the system will prompt up the message that requires the user to reset their own password.

d. Unable to check status

The applicant is unable to check and view the status of the parking reservation when there are two reservations made by the same applicant. For example, applicant A has made a parking reservation at FTMK, selected FTMK_A_01, and FTMK_A_02 as his/her preferred parking spaces, and the date of reservation from 22/09/2023 to 25/09/2023. However, if he/she also wants to make another parking reservation at FTMK, select FTMK_A_03, FTMK_A_04, and FTMK_A_05 as his/her preferred parking spaces and the date of reservation are the same as before which are from 22/09/2023 to 25/09/2023. When the situation above happens, the system is only able to display one parking reservation information when both reservations has been approved.

7.4 Contribution

UTeM Parking Reservation Web-Based System offers a significant contribution by enhancing the efficiency and convenience of parking facilities. Through this system, individuals can proactively reserve a parking spot, promoting the efficient utilization of parking spaces. This leads to a reduction in vacant parking spots, ultimately minimizing the time individuals spend searching for parking spaces. The implementation of a reservation system is instrumental in alleviating congestion within the parking facility and its surrounding areas, allowing applicants to plan their arrival, park swiftly, and contribute to a smoother traffic flow.

Moreover, the system significantly elevates user satisfaction by granting applicants the assurance of a guaranteed parking spot, thus reducing stress and enhancing their overall parking experience. Furthermore, the system's capability to reserve parking slots for special events proves invaluable, saving considerable time for lecturers, staff, and authorities who would otherwise spend excessive time on manual reservation processes.

The convenience factor is further accentuated as the UTeM Parking Reservation System provides crucial insights into available parking spaces for each building or faculty, simplifying administrative tasks. Additionally, it streamlines the monitoring of monthly parking reservation requests by Pejabat Keselamatan, contributing to a more convenient life for the UTeM community. Lastly, the system significantly improves data management by collecting valuable data on parking usage and applicant

behavior. This data serves as a foundation for optimizing parking operations and enhancing service quality, showcasing the system's multifaceted contributions to the UTeM community and its parking infrastructure."

7.5 Conclusion

The UTeM Parking Reservation Web-Based System stands as a solution for enhancing parking efficiency and user satisfaction within Universiti Teknikal Malaysia Melaka (UTeM). By introducing a seamless reservation process for parking spaces, this system addresses the critical issue of inefficient parking management prevalent in many academic institutions. It serves as a testament to technological advancement positively impacting the everyday experiences of students, staff, and visitors.

The system's primary goal is to optimize the utilization of parking spaces, thus minimizing congestion, and reducing the time spent in search of available parking spots. This not only enhances traffic flow within the campus but also significantly contributes to a greener environment by reducing unnecessary fuel consumption. The implementation of a reservation system, particularly for special events, showcases its adaptability to cater to varying parking needs, effectively saving time for event organizers and participants.

Furthermore, the UTeM Parking Reservation System increases user satisfaction by providing a hassle-free and streamlined parking experience. Users can secure their parking slots in advance, eliminating the stress associated with uncertain parking availability. In addition to benefiting users, the system's data-driven approach enables efficient parking data management and comprehensive insights into parking usage trends and user behaviours. The data collected offers a foundation for data-driven decision-making, leading to further improvements and optimization of parking services.

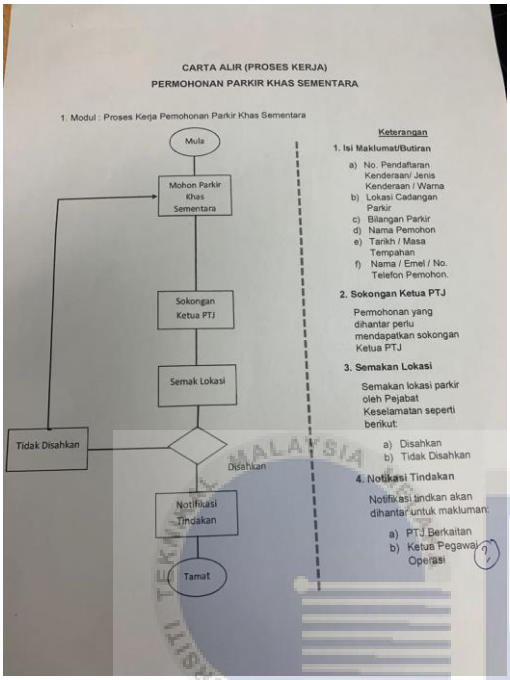
The successful implementation of the UTeM Parking Reservation Web-Based System enhances the overall quality of life for the UTeM community. The journey toward an improved parking experience does not end here. Continuous evaluation, user feedback, and technological upgrades will further refine the system, ensuring that it remains a vital asset in UTeM's pursuit of enhanced efficiency, sustainability, and user satisfaction in parking management.

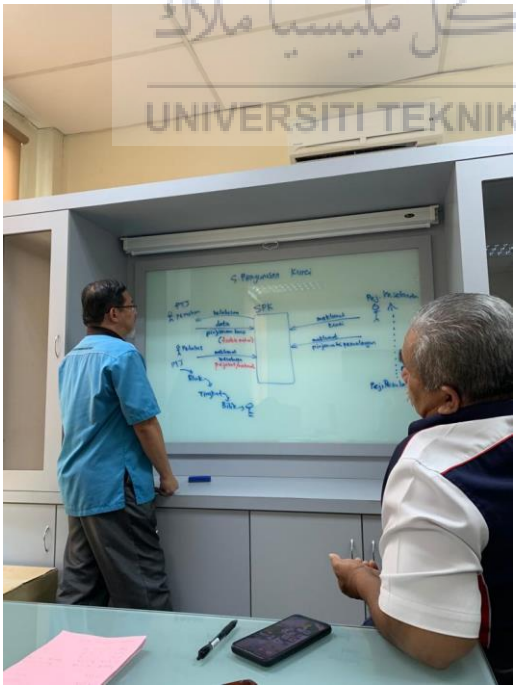
REFERENCES

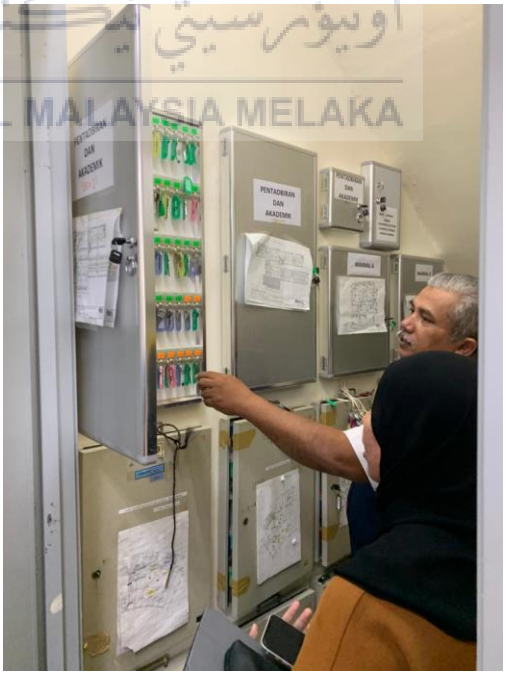
1. Buitleir, D.de (2023) *Car Park Booking System for offices - the ultimate guide (updated 2022), Wayleadr Blog*. Available at: https://wayleadr.com/blog/car-park-booking-system-office/ (Accessed: April 5, 2023).
2. Macea, L.F., Serrano, I. and Carcache-Guas, C. (2023) "A reservation-based parking behavioral model for parking demand management in urban areas," *Socio-Economic Planning Sciences*, 86, p. 101477. Available at: https://doi.org/10.1016/j.seps.2022.101477.
3. *Online parking booking system* (2018) *Nevon Projects*. Available at: https://nevonprojects.com/online-parking-booking-system/ (Accessed: April 5, 2023).
4. *(PDF) online vehicle parking reservation system - researchgate* (no date). Available at: https://www.researchgate.net/publication/324517934_ONLINE_VEHICLE_PARKING_RESERVATION_SYSTEM (Accessed: April 5, 2023).
5. Hongwei Wang and Wenbo He (2011) "A reservation-based Smart Parking System," *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* [Preprint]. Available at: https://doi.org/10.1109/infcomw.2011.5928901.
6. Draw.io. Data Flow Diagram free. (2023). Flowchart Maker & Online Diagram Software. Diagrams.net. https://app.diagrams.net/
7. What is Data Flow Diagram (DFD)? How to Draw DFD? (2023). Visual-Paradigm.com. https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp
8. Software Testing - Quick Guide. (2023). Tutorialspoint.com. https://www.tutorialspoint.com/software_testing/software_testing_quick_guide
9. What is User Acceptance Testing (UAT)? Examples. (2023, August 4). Guru99. https://www.guru99.com/user-acceptance-testing.html
10. Build software better, together. (2021, April 11). GitHub. https://github.com/topics/booking- system?l=php&o=asc&s=forks

APPENDICES

Gather some information about proposed system, at Pejabat Keselamatan with our beloved Supervisor, TS Abd Razak bin Hussain and have some good initial discussions with SJN Rahim.







JENIS DAN JUMLAH PARKIR KESELURUHAN KAMPUS INDUK

BIL	LOT PARKIR	KHAS	STAF	OKU	PELAJAR	PELAWAT	TERBUKA	MOTOR STAF	MOTOR TERBUKA	BAS
1	POS A	-	-	-	-	-	6	-	-	-
2	FTMK	25	88	2	149	-	82	10	282	-
3	PKU	4	-	2	-	-	18	-	10	-
4	SATRIA	-	20	8	-	-	222	-	1172	2
5	LESTARI	-	6	-	-	-	147	-	335	-
6	FKEKK	14	60	2	64	-	-	21	179	-
7	CAFÉ 1	-	-	-	-	-	43	-	65	-
8	FKE	19	65	1	-	4	65	46	91	-
9	PUSAT SUKAN	4	4	3	-	2	184	-	66	10
10	POS B	-	-	-	-	-	2	-	-	-
11	RUMAH TUNGGU SEDIA	-	-	-	-	-	40	-	60	-
12	DEPOH BAS	-	-	-	-	-	-	-	-	25
13	PPKU	60	-	-	-	-	10	-	-	-
14	PEMBANGUNAN	18	-	2	-	-	20	-	22	-
15	CANSELORI	42	-	2	-	2	173	-	35	-
16	CAFÉ STAF	-	-	1	-	-	124	-	40	-
17	MASJID	10	-	1	-	-	132	-	40	-
18	PPPK	17	-	2	-	2	7	-	25	-
19	CAFÉ 2	-	-	2	-	-	30	-	12	-
20	PPP	8	-	2	-	-	34	-	25	-
21	PBPI	11	41	1	-	-	-	-	30	-
22	LAMAN HIKMAH	20	-	2	-	-	152	24	319	-
23	DEWAN CANSELOR	-	-	2	-	-	539	-	219	-
24	DEWAN CANSELOR A	-	-	-	-	-	110	-	-	-
25	DEWAN CANSELOR B	-	-	-	-	-	147	-	73	-
26	DEWAN CANSELOR C	-	-	-	-	-	133	-	-	-
27	FKP BARU	12	-	4	-	-	163	-	267	-
28	FKP LAMA	2	37	-	8	-	15	-	92	4
29	PPS	17	-	2	-	-	48	-	26	-
30	CAFÉ 3	-	-	-	-	-	-	-	-	-
31	PKU	-	-	-	-	-	43	-	-	-
JUMLAH		283	321	41	221	10	2689	101	3485	41

PERMOHONAN PERKHIDMATAN / PERALATAN
APPLICATION FOR SERVICES / EQUIPMENT

* Permohonan hendaklah dibuat sekurang-kurangnya dalam tempoh 3 hari bekerja sebelum tarikh penggunaan
** Application must be made within 3 working days before the date of use*
* Pemohon adalah bertanggungjawab terhadap perkhidmatan / peralatan sehingga selesai program
** The applicant is responsible for the service/ equipment until the completion of the program*
* Permohonan perkhidmatan / peralatan tidak akan diproses / diluluskan sekiranya tiada lampiran berkaitan
** Service/ equipment application will not be processed/ approved if there is no relevant attachment*
* Pejabat Keselamatan tidak bertanggungjawab ke atas sebarang kehilangan peralatan pihak penganjur
** The Security Office shall not be liable for the lost of the equipment of the organizer*
* Keperluan peralatan bergantung kepada jumlah semasa peralatan tersebut di Pejabat Keselamatan
** total equipment depends on the current amount of equipment at Office Security*
* Sila sertakan salinan dokumen berkaitan aktiviti/ program
** Please attach a copy of the document related to the activity/ program*
* Sila tandakan (√) / jumlah di dalam petak berkenaan
** Please tick (√) / amount in the relevant box*

A. KEGUNAAN PEJABAT KESELAMATAN (*Security Office Use*)

Tarikh Terima Permohonan : ...../...../.....
Date of application receive

STATUS PERMOHONAN : ☐ Lulus *Approved* ☐ Tidak Lulus *Not Approved*
Application Status

B. INFORMASI (*Information*)

Tujuan Penggunaan : .....
Purpose of Use

.....

Tarikh : ...../...../..... hingga : ...../...../.....
Date *To*

Masa : ..... hingga : .....
Time *To*

C. *KEPERLUAN PERKHIDMATAN (*Service Requirement*)

☐ Pegawai Pengiring *Escort Officer*
☐ Pengiring Motosikal *Rider*
☐ Kawalan Lalu lintas *Traffic Control*
☐ Kawalan Keselamatan *Security*
☐ Lain-lain (nyatakan) : ..... *Others (Specify)*

D. *KEPERLUAN PERALATAN (*Equipment Requirements*)

☐ Bendera *Flag*
☐ Kon *Cone*
☐ Pembesar Suara *Hailer*
☐ Papan Tanda (Dalaman) *Signboard (Indoor)*
☐ Halangan *Water Filled Barrier*
☐ Baju Pantulan cahaya *Reflective Vest*
☐ Papan Tanda (Luaran) *Signboard (Out door)*
☐ Halangan (Besi) *Traffic Barricades*
☐ Lampu Tongkat *Baton Light*
☐ Walkie Talkie *Walkie Talkie*
☐ Lain-lain (nyatakan) : ..... *Others (Specify)*

E. MAKLUMAT PEMOHON (*Applicant Information*)

Status Pemohon :
Status of Applicant

☐ Staf *Staff* ☐ Pelajar *Student* ☐ Lain-lain (Nyatakan) : ……..……. *Others (Specify)*

Nama :
Name :

No Staf / No Matriks / No Kad Pengenalan :
Staff Number / Matric Number / Identification Card Number :

No. Telefon Bimbit :
Handphone Number :

–

Pusat Tanggungjawab / Fakulti :
Departments / Centres/ Faculties :

Jawatan :
Designation :

F. AKUAN PEMOHON *(Applicant's Declaration)*

Saya mengaku bahawa butir-butir yang terkandung di atas adalah benar. Saya akan bertanggungjawab ke atas keselamatan peralatan dan penggantian peralatan yang rosak atau hilang.

I hereby declare that above particulars are true and I shall be responsible for the safety of equipment and replacement to the damage or lost equipment

…… ……

Tandatangan Pemohon *Applicant's signature* Tarikh *Date*

Diluluskan oleh :
Approved by :
(Ketua PTj Pemohon / Dekan / Ketua Jabatan Fakulti / Ketua Unit PTj / Ketua Bahagian)
(Head of Aplicant PTj / Dean / Head of Faculty Department / Head of PTj Unit / Head of Division)

……

(Tandatangan & Cop Rasmi)
(Signature & Official Stamp)

* Permohonan bagi pelajar hendaklah diluluskan oleh Pejabat Hal Ehwal Pelajar terlebih dahulu
** Applications from students must be approved by the Student Affairs Office*

G. KELULUSAN PEJABAT KESELAMATAN (APPROVAL)

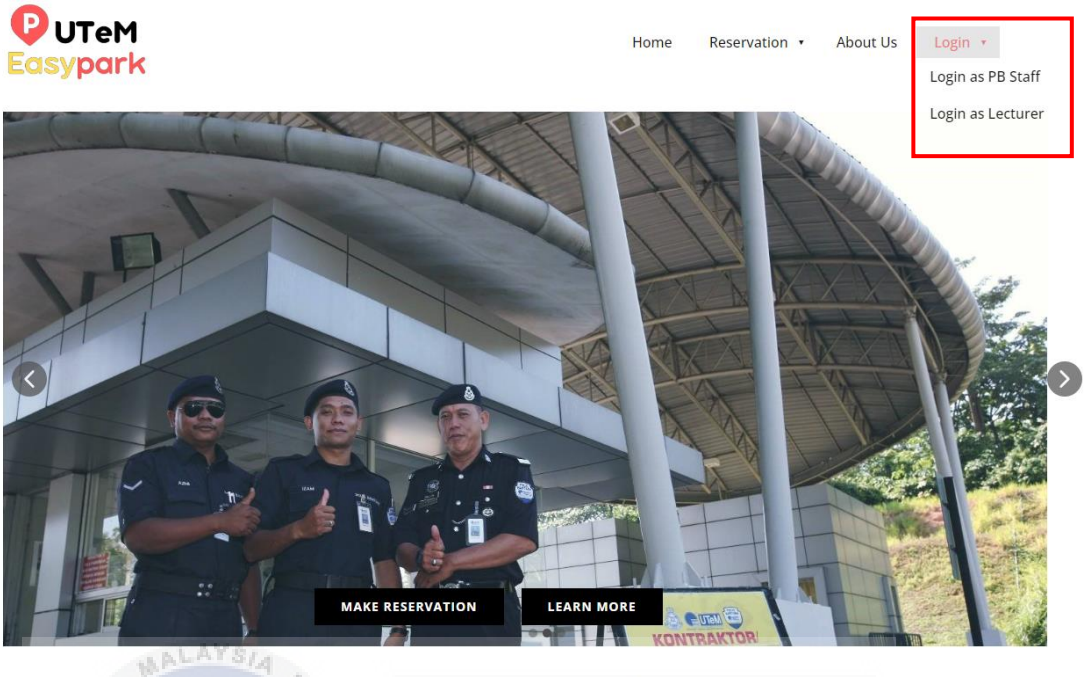
Ulasan : ……
Comment :

…… ……

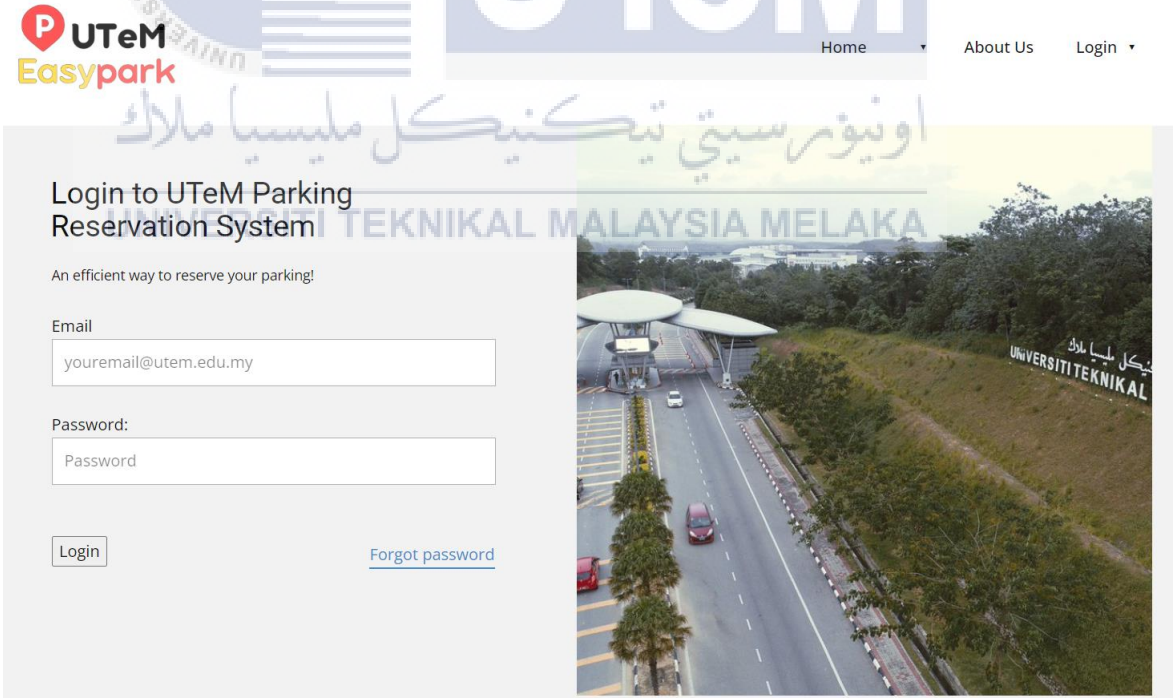
Tandatangan Pegawai *Officer Signature* Tarikh *Date*

- Homepage of UTeM Easypark

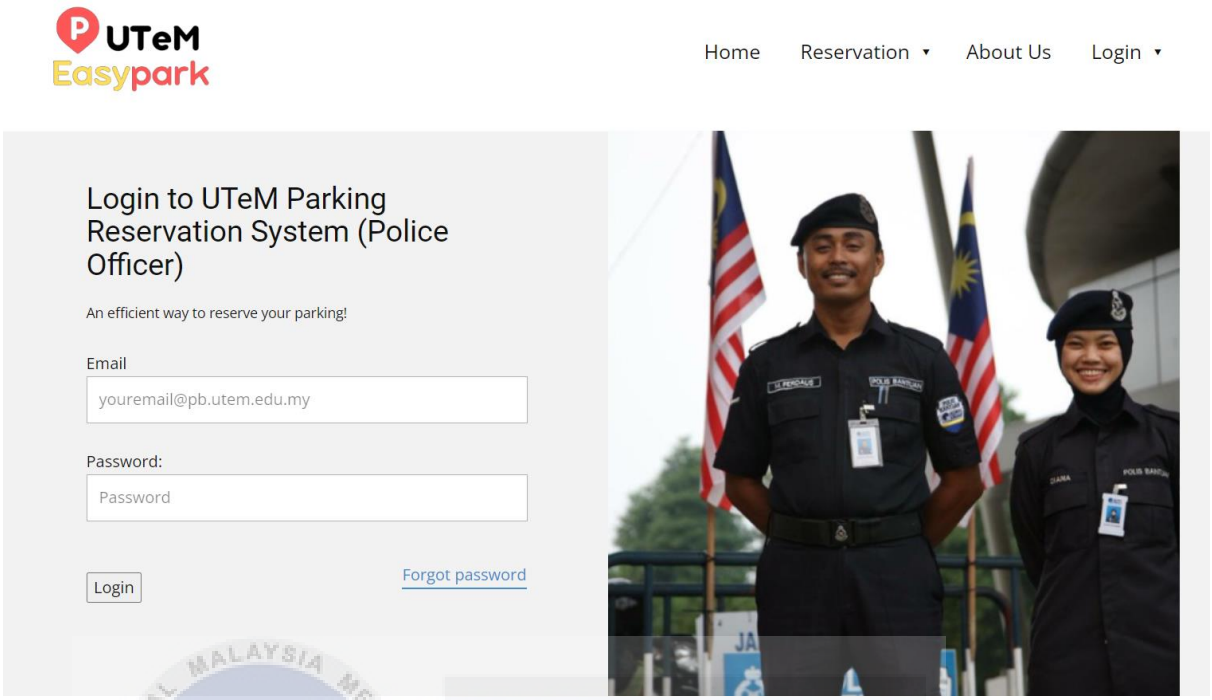
- Login section



- Lecturer/ Staff login page

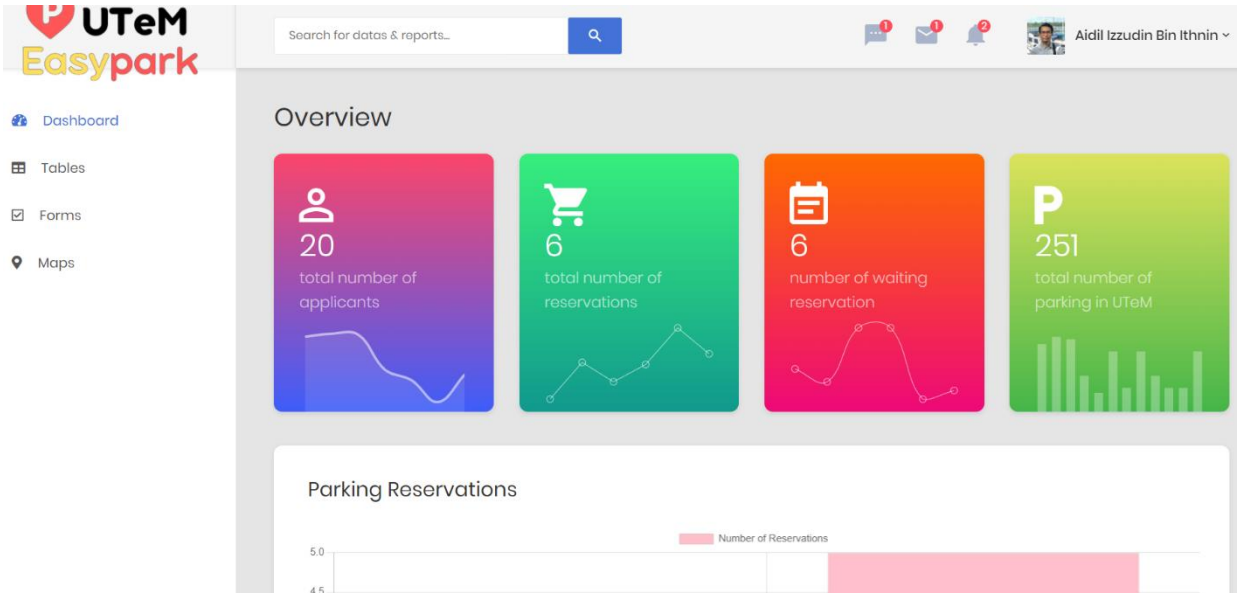


- Polis Bantuan login page

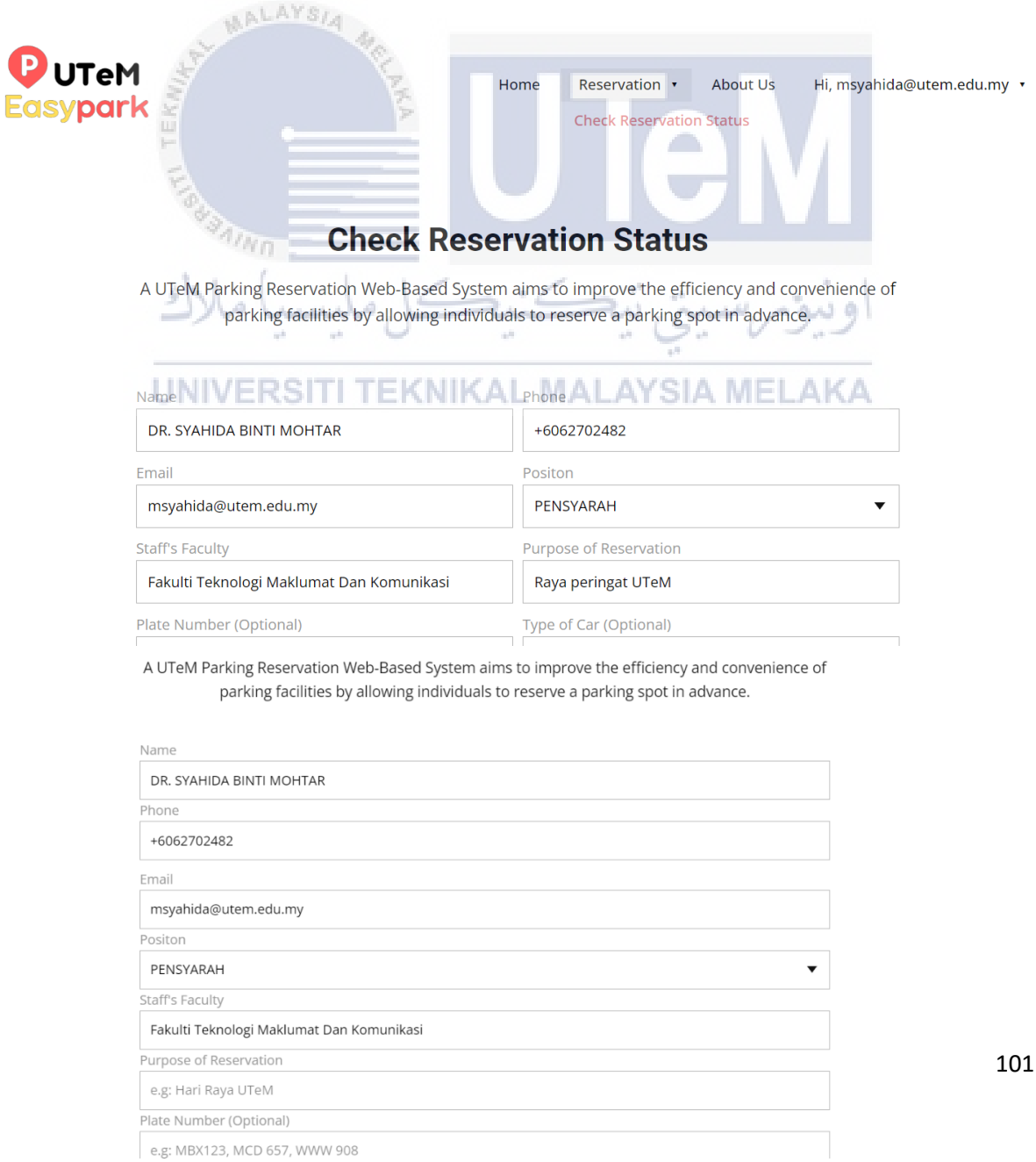


- Homepage for Applicant

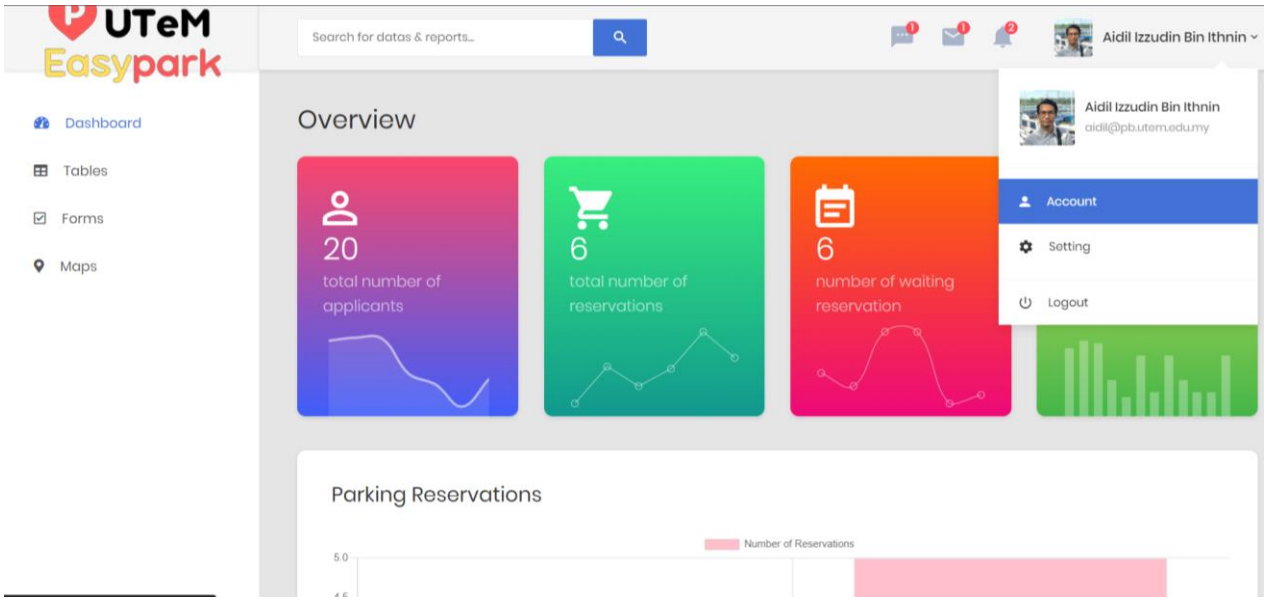
- Homepage for admin



- Check Reservation Status



- Update Profile (Approver, read as: Polis Bantuan)



Home / Account / Edit Profile

Aidil Izzudin bin Ithnin

Ketua Bahagian Keselamatan

Website	https://www.utem.edu.my
Website	https://keselamatan.utem.edu
Twitter	@pejabatkeselamatanutem
Instagram	@pejabatkeselamatanutem
Facebook	Pejabat Keselamatan UTeM

Full Name	Aidil Izzudin bin Ithnin
Email	aidil@pb.utem.edu.my
Phone	126180421
Address	No 20 Jalan Cempaka 6, Taman Seri Cempaka, Peringgit, 75400, Melaka
In charge in PTJ	5

Edit

Total Parking of PTJ in charge:

Canselori

FTMK

FKEKK

Aidil Izzudin bin Ithnin

Ketua Bahagian Keselamatan

No 20 Jalan Cempaka 6, Taman Seri Cempaka, Peringgit, 75400, Melaka

Website	https://www.utem.edu.my
Website	https://keselamatan.utem.edu
Twitter	@pejabatkeselamatanutem
Instagram	

Full Name	Aidil Izzudin bin I
Email	aidil@pb.utem.edu.my
Phone	126180421
Address	No 20 Jalan Cempaka 6, Taman Seri Cempaka, Peringgit, 75400, Mela
In charge in PTJ	5

Save Changes

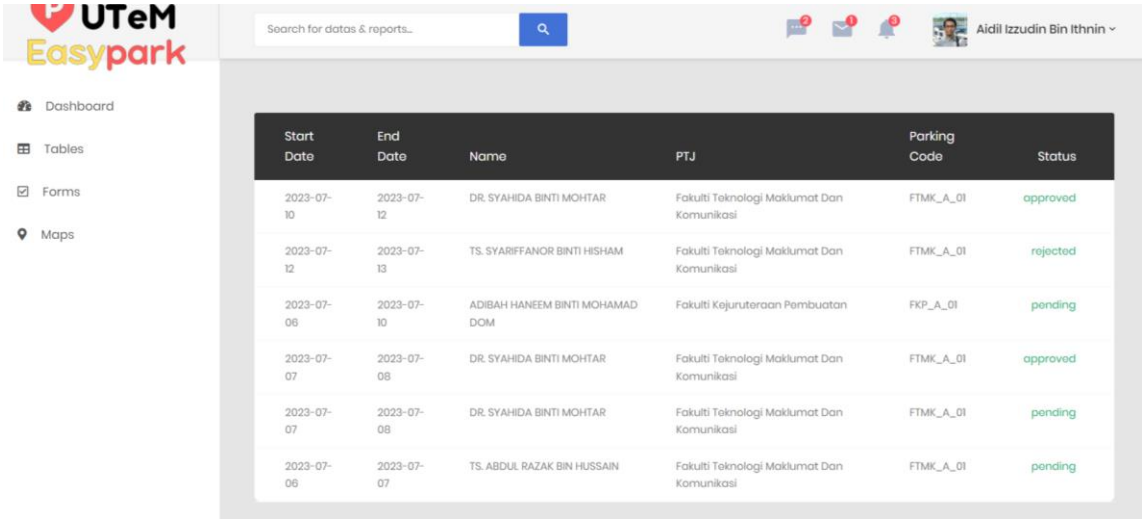
Parking Reservation

Opposite to Mechanic Lab

FTMK, right wing infront of the main lobby

FTMK, right wing infront of the main lobby

- List of Parking Request



- Verify or update Parking Reservation Status

- Update parking, PTJ, special request information