EASTJ Engineering, Agriculture, Science and Technology Journal

e-ISSN: 2948-4065
Volume 2 Number 1 July 2023, 18 ~ 32
DOI: https://doi.org/10.37698/eastj.v2i1.198

IUKL Smart Car Parking App Using Automatic License Plate Recognition and Raspberry Pi

Hafiza Ahmad[1], Abudhahir Buhari[1], Suhaila Mohd Nordin[1], Marlon Charles Dominic Bijoux[1]
[1]Infrastructure University Kuala Lumpur, Kajang, Malaysia

ARTICLE INFO

Article history:

Received May 17, 2023
Revised May 23, 2023
Accepted June 01, 2023

Keywords:

Smart Parking System (Sps),
License Plate Recognition,
Raspberry Pi

Conflict of Interest:

None

Funding:

IUKL Internal Grant

ABSTRACT

This Paper outlines the development of the IUKL Smart Parking mobile app developed for the Infrastructure University of Kuala Lumpur, Malaysia. The Android mobile application was developed using android studio and allows users to make reservations for the IUKL campus parking facilities. Furthermore, an automatic license plate recognition algorithm was developed using a Raspberry Pi computer and tested using sample license plate images displayed on a smartphone. The license plate detection algorithm was implemented using Python as the programming language and the Python library Python-tesseract, which is an optical character recognition tool for Python. The automatic license plate recognition technique was created using this Python module. Additionally, the algorithm was tested using sample license plate images and this was accomplished using images displayed from a mobile device and yielded mainly accurate results, though not 100%. The results were displayed by emitting different LED colored lights, "Red" was emitted when a plate number was not identified, and "Green" was emitted when the plate number was identified successfully. Moreover, a database was developed using PhpMyAdmin which consists of data regarding users and parking lots. This was done to solve the issue whereby no information was recorded regarding the availability of parking spaces at the IUKL campus. This paper can also serve as a template and guide for the complete implementation of a smart parking system in a university-based environment.

Corresponding Author: Bijoux Marlon Charles Dominic, Infrastructure University Kuala Lumpur, De Centrum City, Jalan Ikram-Uniten, 43000 Kajang, Selangor, Malaysia. Tel. +2482578384. E-mail: marlonbijoux@outlook.com

1. Introduction

The Infrastructure University of Kuala Lumpur (IUKL) is a renowned private university based in Kajang, Selangor, Malaysia. The university provides a wide variety of courses in numerous fields of study such as Engineering, Architecture, and IT from undergraduate to postgraduate level.

Furthermore, the IUKL campus accommodates students, lecturers, and non-teaching staff, and parking is provided for faculty and students. One of the Main issues with parking provided on the IUKL campus is that only particular lecturers can utilize some parking facilities. Previously access to these parking facilities was controlled by utilizing RFID cards, but this system is no longer in operation. Another contributing factor to the parking issue is that there is no way to verify who is utilizing the car parks around the campus except for the parking lots, which previously could only be accessed by lecturers, but this is no longer the case as the gated barriers which controlled these parking lots are no longer in operation. As a result, there have been instances where Students Park in reserved parking spaces designated for specific faculty members.

At IUKL, there is a need for different user categories to access the different parking lots, the main user categories include lecturers, students, non-teaching staff, and special guests, which include individuals who visit the campus on a less frequent basis. Thus, there is a need for a system that will enable each of the different user categories mentioned above to utilize the parking facilities at IUKL efficiently and enable the management of IUKL to administer the parking facilities by identifying each user based on their specific car through their unique car license plate number.

A smart parking system, or SPS for short, is an application that consists of several technological components which aid in managing the occupancy of parking infrastructure. SPS implements different types of sensors that are used to determine the availability of parking spaces within a parking lot, the information regarding the availability of parking spaces is gathered in real-time and disseminated to the potential users of the parking lots through the application (Karsten, 2018). With the growing number of vehicles on the road and present infrastructure difficulties, the need for mobility and parking efficiencies is becoming increasingly noticeable. Although self-driving cars are becoming more common, the necessity for parking does not appear to be diminishing. At this point, cities must deal with the potential consequences of all future changes. This can be accomplished by accurate and forward-thinking planning and the use of a Smart Parking system to process and analyze data, resulting in a healthier city (Ancion, 2019).

The agile methodology advocates that the correct product is delivered to the customer by outlining principles that ensure that the product is presented frequently. Corrective actions are taken based on customer feedback. For the smart car park being undertaken, the customer will be IUKL, and iterations of the app will have to be presented to the customer for approval and verification. Additionally, the license plate recognition system will have several modules integrated, namely the Raspberry Pi, License plate detection Algorithm, and the App. Thus, utilizing the agile methodology and developing the system in iterations will enhance testing and troubleshooting if any issues occur when each version is released. Conclusively, this system is part of a larger project which is being undertaken by IUKL, which consists of several team members, and the agile methodology is suitable when operating in a team-oriented environment

2. Literature Review

2.1 Parking Guidance and Information System (PGIS)

PGIS delivers real-time information about parking lots in managed parking spaces. It is a critical component of the Intelligent Transportation System (ITS) and is used with the vehicle guiding system (Al-Turjman & Malekloo, 2019). Sensors are installed at the parking lot's entrances and exit to detect and calculate the number of vehicles entering and exiting. The available parking spaces are determined by sensing the number of vehicles entering and exiting available parking. Along with parking lot sensors, various computational methods are used to determine the status of the parking lot. Users can view the processed data via a variety of different interfaces. Users can make informed decisions about parking lots based on the information. Additionally, the PGIS system provides users with car navigation, guiding them to their allotted parking space.

2.2 E-Parking System

E-Parking combines several smart parking facilities onto a single platform by leveraging innovative technology features (Soni, 2018). E-parking systems are software that is either mobile or web-based. Remote access to the facilities is available to anyone who has downloaded the app to their smartphone. To begin using the smartphone application, users must first register by providing the required personal and vehicle information. The user will gain access to the E-Parking facilities after completing the registration process. The E-Parking system provides the user with various services, including information about the occupancy status of various parking lots located within other parking stands, vehicle guidance to the closest parking stand, reservation capability, a variety of payment methods, vehicular security, and lot retrieval capability.

2.3 Computer Vision/Image Processing

SPS based on computer vision/image processing extracts various types of information from image data, such as parking lot occupancy status (Alsafery et al., 2018), automatic license plate recognition (ALPR), and face recognition for billing and security purposes, as well as to generate a report on road traffic congestion (Fahim et al., 2021; Pamula, 2018). Because these systems extract features from real-time parking lot video data, computer vision/image processing systems often have a high data transmission rate from the camera network to the processing units. These SPSs are frequently used in open parking spaces, as a single camera can cover a big area. However, these systems are prone to occlusion, shadow effects, distortion, and changes in light.

2.4 Internet of Things (IoT)

The Internet of Things (IoT) is the hottest technology movement of the present day, in which all things are networked. Each internet-connected device has its own unique identity (UID). These gadgets can take the form of computational, mechanical, or digital equipment. They can convey data without requiring human-to-human or human-to-computer interaction (Karmokar et al., 2020; Kumar et al., 2019; Lao et al., 2020). The Internet of Things is a vital component of developing SPS. The internet connects all sensors and computational units in an IoT-based SPS, exchanging data without human involvement. Sensors, computational devices, and storage units can communicate via the internet via wired or wireless connections.

2.5 Camera

Numerous SPS researchers have implemented a camera or a network of cameras to detect vehicles and monitor parking lots. Numerous researchers have detected automobile license plates using cameras and various computational techniques (such as computer vision, image processing, and so on) for billing, reservation, and authentication applications. SPS provides users with a secure parking solution by utilizing a camera or network of cameras. However, camera based SPS deployment and maintenance are typically prohibitively expensive (Alam et al., 2018; Khalid et al., 2021; Zantalis et al., 2019).

2.6 Smart Phone Application

A substantial chunk of SPS deploys Android or iOS applications to deliver information about smart parking facilities to end-users. As with web-based apps, smartphone applications include a graphical user interface (GUI) that enables users to interact with the system. Additionally, to obtain real-time information on parking lot status, vehicle navigation to the nearest parking area or a lot, parking lot reservations, parking lot information retrieval, and payment of parking fees via internet services or smartphones' Near Field Communication (NFC) technology (Fahim et al., 2021).

2.7 Automatic Number Plate Recognition (ANPR)

Current state-of-the-art techniques to ANPR rely heavily on OCR engines equipped with AI capabilities. Recognition techniques based on Artificial Neural Networks achieve more excellent recognition rates. Integrating ANPR systems with other information and communication technologies (ICT) tools are increasingly gaining popularity. For example, integration of ANPR engines with GPS, online databases, Android/iOS platforms, and RFID, among other tools used in intelligent transportation systems. Future research is needed to underline the technology's importance and incorporate it with other ICT tools that can help the transportation system and its policymaking. The existing CV algorithm's accuracy is limited to areas and standardization for number plates. Additional study is required to develop sophisticated algorithms to work in varied situations with diverse number-plate datasets that are not standardized. With the accuracy of ANPR systems increasing over time and when combined with AI and IoT capabilities, it is anticipated that these disruptive technologies and applications will gain traction in the following years, leading to the introduction of new use cases. It is feasible to convert raw augmented ANPR camera data to helpful information and aid in comprehending traffic flow, including passenger and freight mobility, using the right tools/software. ANPR cameras can be strengthened by the addition of data about the vehicle's category. Conclusively, based on the result of the analysis conducted, multiple algorithms will be used in conjecture to achieve a higher number plate recognition rate.

2.8 System comparison

The following section compares two of Malaysia's most popular smart parking applications. The apps compared in this section are commercial parking apps that can be used by the public depending on the state in which they reside in Malaysia. Additionally, these applications will be analyzed and compared based on their features.

2.8.1 System 1: Smart Selangor parking

SSP is a public convenience in Selangor and Kuala Lumpur. You can pay your parking summons anytime, anywhere, or hassle-free in just two steps. Reload via secure banking or credit card within the application, and you can begin paying parking or summonses at the touch of a button.

2.8.1.1 Features

- Two-step payment method which involves entering vehicle plate number and parking duration.
- Duration timer
- Notification of time expiry
- Paperless utilizes digital receipts
- Credit can be reloaded directly from the application
- Multiple vehicles can be managed by a single account
- Parking fees for multiple vehicles can be paid simultaneously
- Monthly passes available

- Payments can be made at parking lots by scanning barcodes, entering vehicle plate number or car park ID.
- Can book and pay for parking in advance

2.8.1.2 Screenshots

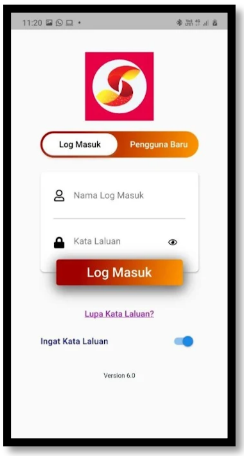


Figure 1. Smart Selangor APP Login Interface

Description: Figure 1 above depicts the login interface of the Smart Selangor APP.

2.8.2 System 2: JomParking

JomParking is a self-service parking application that enables you to save time and money by paying for parking on the go. JomParking is a simple-to-use application that enables you to pay parking fees wherever you are; there is no need to scoop or travel in the rain to do so.

2.8.2.1 Features

- Can be used for both on-street and off-street parking
- Parking duration can be extended
- Notification of remaining parking duration.
- Payments can be made for friends and family by one user.
- More than one vehicle can be registered per user on the App.
- Credit can be reloaded directly from the application

2.8.2.2 Screenshot

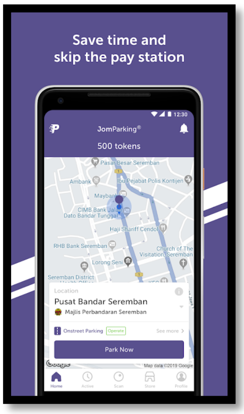


Figure 2. JomParking APP Home Page

Description: Figure 2 above depicts the home page interface of the JomParking APP.

3. Method

According to the agile software development process, the product must be built incrementally across a series of cycles. Additionally, significant improvements can be made following each cycle depending on user feedback and incorporated into subsequent development cycles. In this case, the agile methodology is used to develop a smart parking mobile application for the University of IUKL because changes may be required after the product is implemented based on user and customer feedback, and traditional development methodologies such as the waterfall method will preclude such changes.

Additionally, by employing the agile methodology, the application will be evaluated and reviewed following each development cycle to verify that it meets the client's requirements, which in this case will be the university. Additionally, because this project contains multiple components, incremental development will ensure that each component functions satisfactorily before moving on to the next, and concerns linked to a single component's malfunction can also be detected and remedied promptly.

3.1 System Design

The architectural design pattern used in this project is a structural pattern composed of three primary components: the mobile application, the database, and the raspberry pi. First, the mobile application will be composed of numerous components, including the user interface and the computer code necessary for the app to fulfil its functions. The second component of the system architecture will be the SQL database, which will contain all pertinent data, including customer and vehicle information. Additionally, both the mobile application and raspberry pi will access the database. The final component of the system will be the raspberry pi, which will include modules such as a camera that will capture the vehicle's license plate. Once the plate is captured, the raspberry pi will perform image processing to determine whether the plate captured by the camera exists in the database. Additionally, all operations will be triggered through the mobile application, which will act as a conduit between the user and the system. Finally, it is critical to note the system's inputs and outputs, including a sample image displayed via a smartphone and subsequently taken by the raspberry pi camera. The final output will be the raspberry pi's LED light.

3.1.1 Structural design

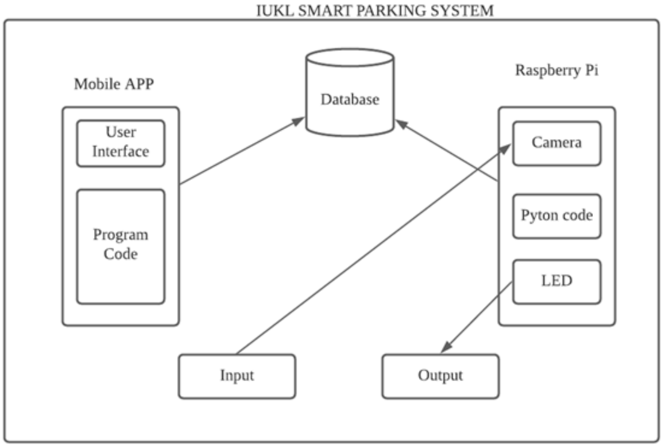


Figure 3. Structural Diagram of IUKL SPS

Description: Figure 3 above depicts the structural components of the IUKL smart parking system.

3.1.2 Hardware implementation

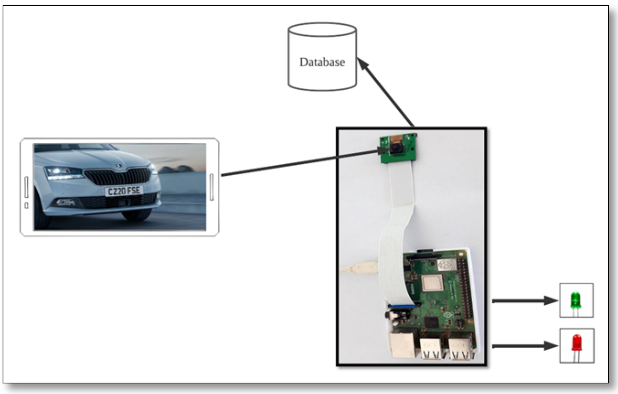


Figure 4. Hardware Implementation of License Plate Detection

Description: The implementation of ALPR on a test-based basis is depicted in Figure 4 above. By capturing photos from a mobile phone with the raspberry pi camera, the raspberry pi will recognise the licence plate number and search the database for matching bookings. Following that, depending on the outcome of the database search, either a green or red led will illuminate.

3.2 UML Diagrams

The following section will provide illustrations of different types of UML diagrams for the IUKL smart parking system. Moreover, some of the diagrams depicted are Use case, Activity, Sequence, Class, and State-Chart diagrams. These diagrams are based on a fully conceptualized implementation of a smart parking system for the University of IUKL; in terms of the project, it will be a more test-based implementation and will serve as a template for the future implementation of a smart parking system at the University of IUKL.

3.2.1 Use case diagram

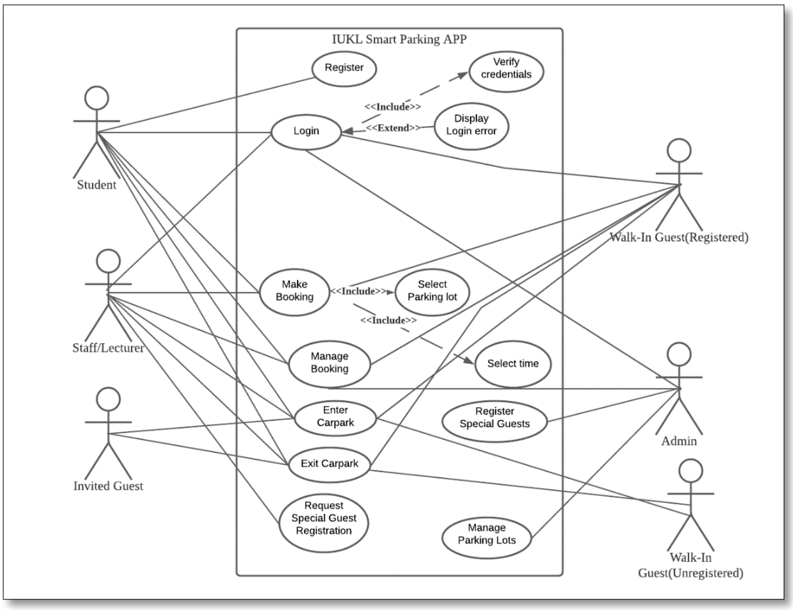


Figure 5. Use Case Diagram for IUKL SPS

Description: The above figure depicts the use case diagram of the IUKL smart parking application, which will be developed. Firstly, there are ten use cases and five actors: student, staff, invited-guest, Walk-In guest, and Admin. Subsequently, each use case will have a specific function, which one or more actors will perform.

3.2.2 Activity diagram for making bookings

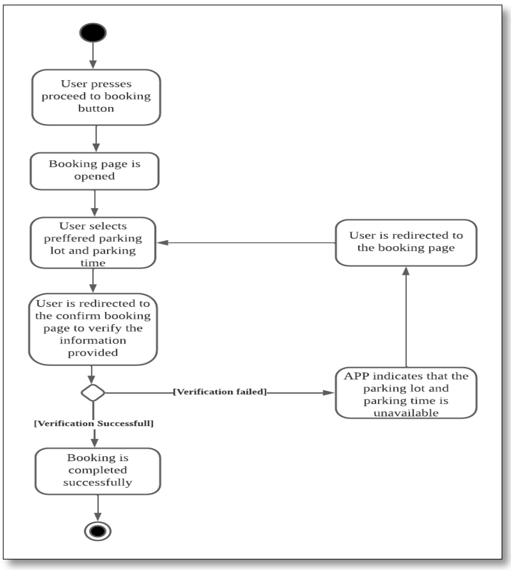

User presses proceed to booking button
Booking page is opened
User selects preffered parking lot and parking time
User is redirected to the booking page
User is redirected to the confirm booking page to verify the information provided
[Verification failed]
APP indicates that the parking lot and parking time is unavailable
[Verification Successfull]
Booking is completed successfully

Figure 6. Make Booking Activity Diagram

Description: Figure 6 above illustrates the activity diagram for the make booking use case. Firstly, this process begins by the user pressing the booking button in the APP. Next, the booking page is opened whereby the user enters their booking information such as preferred parking lot and time. The user is then prompted to review the booking details and confirm the booking. If the selected booking slot is not available, the user is prompted to select a different slot. Otherwise, the booking is completed successfully.

3.2.3 Class diagram

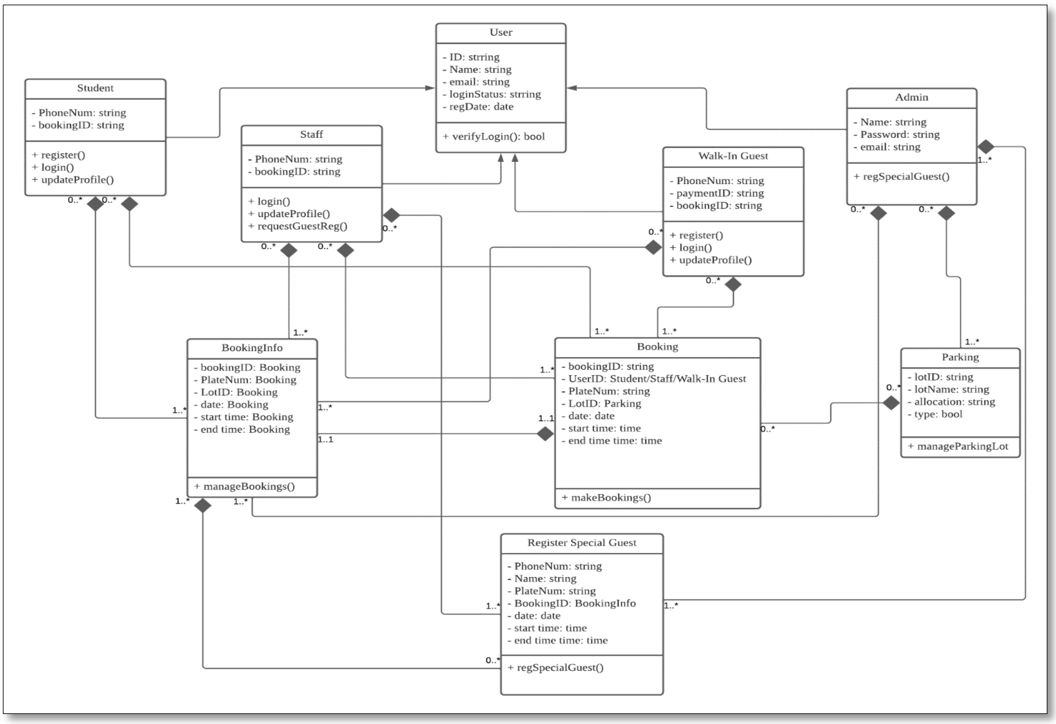


Figure 7. Class Diagram

Description: Figure 7 above illustrates the database architecture for the IUKL smart parking system; there are ten tables within the database, and each table is related to another table. Firstly, there is inheritance within the architecture as the user table serves as the parent class and the staff, student and walk-in guest and admin class are represented as its children. The user class contains the primary detail that every user must provide to register; the user class also contains a verified login function that will authenticate users' login. Furthermore, the staff, student and walk-in guest classes are similar in their attributes and functions. However, there is an exception in the staff class whereby they do not need to register, and they can request the registration of special invited guests. Moreover, there are four more classes, namely Booking Info, which consists of all booking information, booking, which is a process class that enables bookings to be made by users. Additionally, the parking class holds the information on parking lots. Only admin can edit it, and the Register special guest is another process class that involves staff and admin. Composite relationships are also ever-present in the diagram above, which entails that one class depends on another. An example of this shown above is that the booking class is dependent on users such as staff, walk-in guests, and students to make bookings. Conclusively, another aspect of class diagrams is the multiplicity constraint. An example of this illustrated above is that students can make zero to many (0...*) bookings, and bookings can be made by zero to many (0...*) students.

3.2.4 State Chart diagram for entering carpark

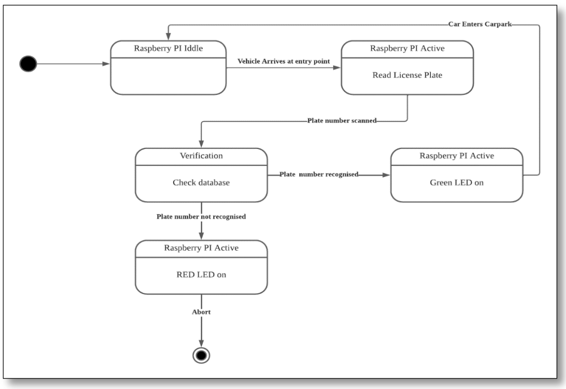


Figure 8. Entering Carpark State Chart Diagram

Description: Figure 8 above illustrates a series of processes undertaken to enter the carpark. Additionally, the state of the hardware component, which is the raspberry pi, is indicated at each step. Firstly, the Raspberry Pi is idle; once a vehicle arrives at the entry point, the raspberry pi captures the license plate image. Furthermore, the image is verified as the raspberry pi checks the database; if the plate number is found in the database, the raspberry pi emits a green led light, the user enters the car park, and the raspberry pi is returned to an idle state. On the other hand, if the plate number is not found in the database, the raspberry pi emits a red led light, aborting the process.

4. Results and Discussion

This chapter will go over some of the interfaces of the created mobile application. Furthermore, the functionality of each interface and its relationship to the overall system flow will be thoroughly described. Moreover, the actors who will utilize each interface individually, as well as the actors who will be impacted by actions on each interface, will be stated. Additionally, this chapter will provide an overview of the testing done to make sure that each part of the IUKL smart parking system is meeting its set objectives. The testing approach used was manual testing, which means that no automated software testing tools were used during the test. The tests' primary goal will also be to find any faults, errors, or defects in the mobile application created as well as the Python code for the license plate recognition that was used with the Raspberry Pi. Even though manual testing is the most archaic testing method, it is nevertheless useful in identifying serious software application defects.

4.1 System Interfaces

This section will highlight some of the interfaces of the IUKL smart parking system, detailing the function of each interface of the mobile application.

4.1.1 Login interface

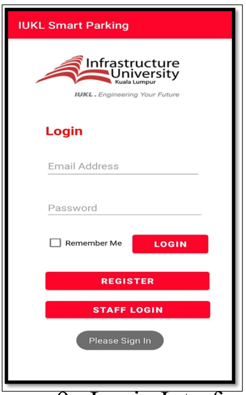


Figure 9. Login Interface

Description: The main login interface of the IUKL Smart Parking App is seen in Figure 9. On this page, students or guests with registered accounts can log in to the app by entering their credentials, specifically their registered email address and account password, and pressing the "LOGIN" button. Furthermore, new users will be able to register a new account by pressing the "REGISTER" button. Staff can also access the staff login interface by pressing the "STAFF LOGIN" button. Furthermore, the "Remember Me" checkbox allows users to remain logged in, and if this is checked, the user will be automatically redirected to the main menu screen when they relaunch the application on their mobile device.

4.1.2 Registration interface

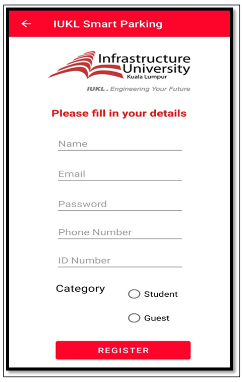

← IUKL Smart Parking
Infrastructure University Kuala Lumpur
IUKL . Engineering Your Future
Please fill in your details
Name
Email
Password
Phone Number
ID Number
Category
○ Student
○ Guest
REGISTER

Figure 10. Registration Interface

Description: Figure 10 depicts the IUKL Smart Parking App’s student/guest registration screen. On this interface, new students or guests can create an account by entering information such as their name, email, password, phone number, and ID number, which can be a school student ID, passport number, or IKAD. Furthermore, once the user has entered their information, they will be redirected to the registration success interface if the registration is successful. Upon completion of the form and category selection, the user can then click on the “REGISTER” button to complete their account registration and will be redirected to the registration success interface if the registration is successful.

4.1.3 Bookings interface

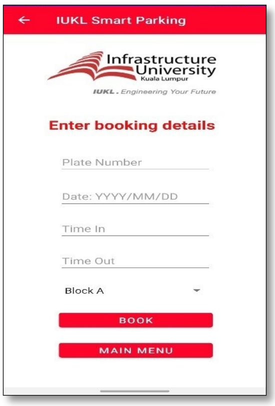
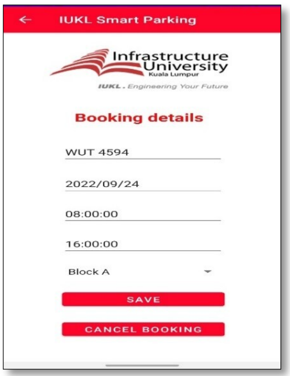


Figure 11. Bookings Interface

Description: The booking interface of the IUKL Smart Parking App is depicted in Figure 11. The user will be able to enter their booking details on this interface, and each detail entered will be validated. Once the user has completed the booking form, they can press the “BOOK” button, and their booking will be processed. The user will be sent to the booking success interface if the booking is successful. In addition, the user can return to the application’s main menu by pressing the “MAIN MENU” button.

4.1.4 Edit booking interface


IUKL Smart Parking
Infrastructure University Kuala Lumpur
IUKL. Engineering Your Future
Booking details

WUT 4594

2022/09/24

08:00:00

16:00:00

Block A

Figure 12. Edit Booking Interface

Description: Figure 12 depicts the IUKL Smart Parking application's edit booking interface. This interface will allow users to alter booking details, and any changes they make will be validated. To save the changes, users can press the "SAVE" button. In addition, users will be able to cancel bookings through this interface by pressing the "CANCEL BOOKING" button.

4.2 Testing

4.2.1 User registration test case

Pre-condition: User presses the "REGISTER" button on main login screen

Actors: students & guests

Dependencies: NONE

Test Case ID: TC1_UserReg

Test Title: Valid User Registration

Test Summary: to verify the registration functionalities with valid inputs of name, email, password, phone number, ID number and category.

Table 1. User Registration Test Case

Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Comments (If any)
Step 1	Enter Name	The text field for name should be editable and acceptable.	The text field for name is editable and acceptable.	Pass	
Step 2	Enter Email	The text field for email should be editable and acceptable.	The text field for email is editable and acceptable.	Pass	
Step 3	Enter Password	The password text field should be editable, acceptable, and should not accept inputs of more than 8 characters.	The password text field is editable and does not allow inputs of more than 8 characters.	Pass	
Step 4	Enter Phone Number	The phone number text field should be editable and acceptable, with input masked to match the format of local phone numbers.	The text field for the phone number is editable and masked to match the format of local phone numbers.	Pass	
Step 5	Enter ID Number	The text field for ID number should be editable and acceptable.	The text field for ID Number is editable and acceptable.	Pass	
Step 6	Select category	Category radio buttons should be selectable; however, the user should only be able to choose one.	The category radio buttons are selectable, and the user can only select one.	Pass	
Step 7	Press register button	The user account is created, and the user is redirected to the registration success screen.	The user account is created, and the user is redirected to the registration success screen.	Pass	

Description: Table showing the results of the tests done to make sure the registration interface is working properly.

4.2.2 User login test case

Pre-condition: User opens the application on their mobile device.

Actors: students & guests

Dependencies: User must have a registered account

Test Case ID: TC2_Login

Test Title: Valid functionality of “LOGIN” button

Test Summary: to verify the login functionalities with valid inputs of email and password.

Table 2. User Login Test Case

Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Comments (If any)
Step 1	Enter Email	The text field for email should be editable and acceptable.	The text field for email is editable and acceptable.	Pass	
Step 2	Enter Password	The password text field should be editable, acceptable, and should not accept inputs of more than 8 characters.	The password text field is editable and does not allow inputs of more than 8 characters.	Pass	
Step 2	Press "LOGIN" button	The user is successfully logged in and is redirected to the main menu.	The user is successfully logged in and is redirected to the main menu.	Pass	

Description: Table showing the results of the tests done to make sure the login interface is working properly.

4.2.3 Booking test case

Pre-condition: User presses the "BOOK" button on the main menu screen.

Actors: students & guests

Dependencies: Users must be logged in to the application.

Test Case ID: TC3_Booking

Test Title: Valid functionality of "Book" button

Test Summary: to verify the booking functionalities with valid inputs of plate number, date, time in, time out and parking lot selection.

Table 3. Booking Test Case

Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Comments (If any)
Step 1	Enter plate number	The plate number text field should be editable and acceptable, and the input should be masked to accept only local plate number formats.	The plate number text field is changeable, acceptable, and masked to fit the format of local numbers.	Pass	
Step 2	Enter Date	The date text box should be editable, acceptable, and masked to follow the given date format.	The date text field is editable, acceptable, and masked to follow to the specified date format.	Pass	
Step 3	Enter Time In	The time in the text field should be editable, acceptable, and masked to allow just time input.	The time in the text field is editable, acceptable, and masked to only allow time input.	Pass	
Step 4	Enter Time Out	The time out the text field should be	The time out the text field is	Pass	

		editable, acceptable, and masked to allow just time input.	editable, acceptable, and masked to only allow time input.		
Step 5	Select Parking Lot	Parking lot drop down list elements should be selectable, and the user should only be able to select one item from the list.	Parking lot drop down list elements are selectable, and the user is only able to select one item from the list.	Pass	
Step 6	Press book button	The user's booking is successfully processed, and the user is redirected to the booking success screen.	The user's booking is successfully processed, and the user is redirected to the booking success screen.	Pass	

Description: Table showing the results of the tests done to make sure the booking interface is working properly.

4.2.4 Edit booking test case

Pre-condition: User presses the “BOOKINGS” button on the main menu screen.

Actors: students and guests

Dependencies: Users must be logged in to the application.

Test Case ID: TC4_EditBooking

Test Title: Valid functionality of “Save” button.

Test Summary: to verify the booking editing functionalities with valid inputs of plate number, date, time in, time out and parking lot selection.

Table 4. Edit booking Test Case

Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Comments (If any)
Step 1	The user chooses a booking from the list to update.	The user should be taken to the booking screen for the chosen booking.	The user is taken to the booking screen for the chosen booking.	Pass	
Step 2	Edit plate number	The plate number text field should be editable and acceptable, and the input should be masked to accept only local plate number formats.	The plate number text field is changeable, acceptable, and masked to fit the format of local numbers.	Pass	
Step 3	Edit Date	The date text box should be editable, acceptable, and masked to follow the given date format.	The date text field is editable, acceptable, and masked to follow to the specified date format.	Pass	
Step 4	Edit Time In	The time in the text field should be editable, acceptable, and	The time in the text field is editable, acceptable, and	Pass	

		masked to allow just time input.	masked to only allow time input.		
Step 5	Edit Time Out	The time out the text field should be editable, acceptable, and masked to allow just time input.	The time out the text field is editable, acceptable, and masked to only allow time input.	Pass	
Step 6	Select Parking Lot	Parking lot drop down list elements should be selectable, and the user should only be able to select one item from the list.	Parking lot drop down list elements are selectable, and the user is only able to select one item from the list.	Pass	
Step 7	Press save button	The user's booking is successfully edited, and the user is redirected to the manage bookings screen.	The user's booking is successfully edited, and the user is redirected to the manage bookings screen.	Pass	Proceed to Module Manage Bookings

Description: Table showing the results of the tests done to make sure the edit booking interface is working properly.

4.3 Discussion

This chapter outlined some of the mobile application interfaces that were created for the IUKL smart parking system. Furthermore, by explaining the links between the various interfaces, each interface description provides an overview of the entire flow of the system. Moreover, the users who will have access to each interface are highlighted, offering a more in-depth understanding of how different users would interact with the application. Additionally, this chapter also detailed the extensive testing performed for the IUKL smart parking system using the manual testing approach. To begin, each main component of the developed mobile application was tested, including user login, booking, and registration. The testing techniques ensured that each function of the mobile application was fulfilling its desired outcomes. Finally, this chapter has presented the outcome of a major element of the IUKL smart parking system, namely the mobile application, which is the principal mode of interaction between users and the system.

5. Conclusion

This chapter will culminate the IUKL Smart Parking project and discuss the project's details and outcomes. To begin, the project's accomplishments will be detailed, as will how the system developed has resolved challenges encountered by users of the current system. Furthermore, as previously stated, the system produced can only be applied in a test-based manner; consequently, there are key limitations to the system built, which will be fully discussed in this chapter. Furthermore, one of the primary goals of this project was to create a system that could be used as a template to create a parking system within a campus-based environment. As a result, this system can be used to guide the implementation of a completely automated parking system. Finally, this will bring this paper to a close and provide a detailed outlook on the project's overall outcomes.

5.1 Project Achievements

This section will go over the project's accomplishments in accordance with that the project's objectives must first be recalled. The first goal was to create a smart parking mobile application that allowed users to make advance bookings to use a specific car park, among other things. Furthermore, it can be concluded that this project has met the requirements of the first objective; thus, the first achievement of the project is the IUKL Smart Parking mobile app, which is an android mobile application developed using android studio that allows users to make reservations for the IUKL campus parking facilities.

Furthermore, the project's second objective was to develop an automatic license plate identification algorithm on a Raspberry Pi computer and test it using sample license plate images displayed on a smartphone. The first half of this goal was to construct the license plate detection algorithm using Python as the programming language and the Python library Python-tesseract, which is an optical character recognition tool for Python. The automatic license plate recognition technique was created using this Python module.

The second project objective was also to evaluate the implemented algorithm using example license plate images. This was accomplished using images displayed from a mobile device and yielded mainly accurate results, though not 100%. As a result, it is worth noting that the project's second objective was also met. Furthermore, a few other project accomplishments can be extrapolated from problem statements. The first problem statement was that there were no details recorded regarding the availability of parking spaces at the IUKL campus, which was accordingly handled in the system's development by developing a database that included tables for parking lot information such as name and capacity.

Furthermore, the administrative user account can control these parking spaces using the developed mobile application. The second problem statement implies that there is no method to determine which type of user is utilizing the parking, whether they are students, staff, or others. The system handles this issue as well, as there are clear designations for each sort of user, and the administrator can see which type of user is using each car park based on their plate number.

The final problem statement concerns the usage of RFID cards and scanners, which were previously used to verify admission into car parks; however, the system developed avoids this issue by scanning each car license plate before entering the car park to validate entry. Overall, the project's primary accomplishments are the creation of a smart parking mobile app, the implementation of a license plate recognition algorithm, the creation of a database to store parking lot and user information, and the deployment of license plate recognition technology.

5.2 Limitations

Although the implemented smart parking system addresses many of the concerns that plagued IUKL users of the prior parking system, the technology has limits that must be addressed. This section will go into detail about the potential constraints. To begin, the system was developed in a test-driven environment and so cannot be fully deployed without various changes, such as the use of more powerful hardware components.

The raspberry pi computer, for example, is a low-spec module with limited computing power. Also, the raspberry pi camera module used is only 8 megapixels and does not enable auto focus, thus adequate lighting conditions must be made available to obtain decent images. In terms of system software, the application was designed using Android Studio and can only be used on Android devices; this can be considered as a serious issue for iPhone users who will be unable to use the application and, as a result, the IUKL carpark.

Furthermore, under the current system, the database is only hosted on a local computer and cannot be viewed by users on a different network; while this is sufficient for a test-based environment, the database will need to be publicly accessible to fully implement the parking system. Finally, as compared to commercially based parking apps, the app's functionality is notably lacking. This is not a major issue because the application developed does not serve the same purpose as the apps but could include some additional features such as map-based guidance to selected parking lots.

It should be noted that to implement more advanced features, additional sensors such as motion sensors and proximity sensors will be required for the project. In conclusion, while the developed system has some limitations, it achieves its goals and serves as the foundation for completely implementing a smart parking system in a campus-based setting.

5.3 Future Work

Although it has been emphasized several times in this chapter that the developed system is merely implemented in a test-based manner, this section will discuss how the developed system can be utilized to fully deploy a smart parking system at a university. To begin, a version of the mobile application for iPhone users should be developed, allowing most drivers who visit the university to download and install the app on their preferred mobile devices. Furthermore, on the software side, certain elements of the mobile application can be improved, such as adding notifications and allowing users to prolong booking durations, which would improve the app's usability.

On the administrative side, functionality such as reports can be added to provide a summary of parking lot usage, such as peak parking hours, which can be used and studied by the university's parking facility management team. On the other hand, future hardware enhancements can be developed to completely integrate the parking system, such as adding a parking bar that will allow the user to enter the car park provided their plate number is verified. In addition, a proximity sensor will be required to detect when the driver's vehicle approaches the carpark's entry or exit and initiate the license plate verification process. Furthermore, to capture license plate images, a higher-end camera module will be necessary to improve the accuracy of the number plate recognition algorithm.

The computing module will also need to be more powerful to process multiple requests in a timely manner, and rather than using just one computing module, several will need to be distributed across the campus car park at each entry and exit, and all modules will need to be integrated. To summarize, a large financial expenditure will be necessary to completely implement the smart parking system blueprint outlined in this paper, but once completed, the project will undoubtedly have a significant positive impact on the management of parking infrastructure at an institution.

Acknowledgements

We would like to acknowledge the Infrastructurre University of Kuala Lumpur which has provided the grant for the completion of this research. Additionally, each of the author's have provided a significant contribution in the writing and proofreading of the paper.

References

Alam, M., Moroni, D., Pieri, G., Tampucci, M., Gomes, M., Fonseca, J., Ferreira, J., & Leone, G. R. (2018). Real-time smart parking systems integration in distributed ITS for smart cities. *Journal of Advanced Transportation*, *2018*.

Alsafery, W., Alturki, B., Reiff-Marganiec, S., & Jambi, K. (2018). Smart car parking system solution for the internet of things in smart cities. *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 1–5.

Al-Turjman, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. *Sustainable Cities and Society*, *49*, 101608.

Ancion, K. (2019, March 19). *What are the benefits of smart parking for drivers?* https://www.parkeagle.com/2019/03/19/what-are-the-benefits-of-smart-parking-for-drivers/

Fahim, A., Hasan, M., & Chowdhury, M. A. (2021). Smart parking systems: comprehensive review based on various aspects. *Heliyon*, *7*(5), e07050.

Karmokar, P., Bairagi, S., Mondal, A., Nur, F. N., Moon, N. N., Karim, A., & Yeo, K. C. (2020). A Novel IoT based Accident Detection and Rescue System. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 322–327. https://doi.org/10.1109/ICSSIT48917.2020.9214296

Khalid, M., Wang, K., Aslam, N., Cao, Y., Ahmad, N., & Khan, M. K. (2021). From smart parking towards autonomous valet parking: A survey, challenges and future Works. *Journal of Network and Computer Applications*, *175*, 102935.

Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, *6*(1), 1–21.

Lao, L., Li, Z., Hou, S., Xiao, B., Guo, S., & Yang, Y. (2020). A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR)*, *53*(1), 1–32.

Pamula, T. (2018). Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks. *IEEE Intelligent Transportation Systems Magazine*, *10*(3), 11–21.

Soni, V. D. (2018). IOT Based Parking Lot. *International Engineering Journal for Research & Development*, *3*(1), 9.

Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A review of machine learning and IoT in smart transportation. *Future Internet*, *11*(4), 94.