

Automated Vehicle Parking Management System

Computer/Electrical Engineering Capstone Design Project

Toronto Metropolitan University, F2023

by

*Taskin Abdur-Rahman*

*Zohraan Badar*

*Kisoban Rajendran*

*Yadu Krishnan Madhu*

# Acknowledgement

We would like to express my sincere gratitude to all those who contributed to the successful completion of this engineering design report. The completion of this project would not have been possible without the support and collaboration of various individuals and organizations.

We extend our appreciation to:

* Dr. Alagan Anpalagan: The Faculty Lab Coordinator (FLC) for this project, who is recognized for his guidance and assistance for students throughout their design project.
* ECB Department: The Department of Electrical, Computer, and Biomedical Engineering for facilitating the capstone design project experience for students and for their continued support for the learning opportunities available for students.

Furthermore, we want to acknowledge the contributions of my peers and colleagues who provided valuable feedback and assistance at various stages of the project. Lastly, we appreciate the resources and facilities provided by the Faculty of Engineering and Architectural Science.Their collective efforts have significantly enriched the quality of this engineering design report. Thank you all for your support.

# Certification of Authorship

“I hereby certify that I/we are the author(s) of this document and that any assistance I/we received in its preparation is fully acknowledged and disclosed in the document I/we have also cited all sources from which I/we obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I/we also certify that this paper was prepared by me/us for this purpose.”

|  | Student 1 | Student 2 | Student 3 | Student 4 |
| --- | --- | --- | --- | --- |
| Name | Taskin Abdur-Rahman | Zohraan Badar | Kisoban Rajendran | Yadu Krishnan Madhu |
| Email | navid.rahman@torontomu.ca | z1badar@torontomu.ca | kisoban1.rajendran@torontomu.ca | ymadhu@torontomu.ca |
| Phone | 647-920-6501 |  | 647-939-2069 | 437-213-9830 |
| Signature |  |  |  | Y.M |

# 

# 

# Table of Contents

[**Acknowledgement 2**](#_n7o6k5ldtkzl)

[**Certification of Authorship 3**](#_tt5b28mye8is)

[**Table of Contents 4**](#_yu0yuouow458)

[**Abstract 5**](#_3znysh7)

[**Introduction & Background 6**](#_2et92p0)

[**Objectives 7**](#_tyjcwt)

[**Approach & Methods 9**](#_1t3h5sf)

[Touch-free system 9](#_cn4x5yt6j0aw)

[Automation 9](#_f64ntln5j5yj)

[Minimizing traffic 10](#_gxf5mq3z9ih5)

[**Design Analysis and Synthesis 11**](#_185ofowbgqdt)

[Parking Lot Automation Subsystem 11](#_enpkf7mb8x4w)

[Entrance & Exit 11](#_dz4uxrpmrs43)

[Parking Lot 12](#_ogpfadhicr2a)

[Web Application 14](#_aen69bt4ealz)

[Frontend 14](#_u16saqf0m6it)

[Backend 15](#_ryjgkinftw3j)

[Data Analysis and Prediction Algorithms 15](#_vdgv5gfhvadh)

[Cloud Database Solution 17](#_22f1hcl2a9a)

[**Implementation Plan (GANTT Chart) 18**](#_2s8eyo1)

[**Conclusions 19**](#_35nkun2)

[**References 20**](#_1ksv4uv)

[**Appendices 20**](#_2jxsxqh)

# Abstract

As urban areas continue to face challenges related to limited parking spaces and increasing vehicular traffic, the demand for innovative parking solutions has become more pressing. This paper introduces an Automated Parking Management System (APMS) designed to address these challenges by leveraging advanced technologies to optimize parking operations and enhance user experience. The APMS employs a combination of sensor networks, machine learning algorithms, and smart infrastructure to automate the entire parking process. The system begins with real-time monitoring of available parking spaces using sensors strategically placed throughout the parking facility. These sensors continuously collect data on space occupancy and transmit the information to a centralized control unit. Utilizing machine learning algorithms, the APMS analyzes historical parking patterns and predicts peak usage times, allowing for dynamic management of parking resources. This predictive capability enables the system to optimize space allocation and minimize congestion during high-demand periods. Furthermore, the system employs smart payment and reservation features, allowing users to pre-book parking spaces and complete transactions seamlessly through a mobile application or web interface. The APMS prioritizes sustainability by minimizing the environmental impact of parking operations. Through efficient space utilization and reduced vehicle idle times, the system contributes to lower carbon emissions and promotes a greener urban environment. The implementation of the APMS not only enhances parking facility efficiency but also significantly improves the overall user experience. Users benefit from reduced search times, convenient payment options, and a seamless integration of parking services with their daily routines. Additionally, the system provides parking facility operators with valuable insights through data analytics, enabling informed decision-making for further optimization. This research contributes to the ongoing discourse on smart city initiatives and urban mobility solutions, providing a scalable and adaptable framework for the deployment of Automated Parking Management Systems in diverse urban landscapes. The results showcase the potential for significant improvements in parking efficiency, user satisfaction, and environmental sustainability through the integration of cutting-edge technologies in urban parking management.

# Introduction & Background

Parking lots can be very crowded and have very limited parking spaces (ex. at Airports, malls, downtown) due to unorganized traffic and pedestrians that might get in the way. Because of this it can cause accidents that may harm pedestrians or other drivers without organizing the traffic of a parking lot. The aim of this project is to design an automated parking management which is touch-free, fully automated, time efficient, user friendly. With the rising shortage of real estate and increasing population the need for efficient parking management systems has also increased. This parking system should reduce traffic on public roads and congestion in parking lots through the use of availability prediction algorithms from collected data. The parking lot is made touch-free with the help of QR code authentication, vehicle detection sensors, and automatic license plate recognition with cameras. A backend server automates all activities in the parking lot, such as entrance authentication, parking space occupancy monitoring, exit authentication, payment, and data collection. The remote backend server controls devices in the parking lot such as cameras, sensors and lights, through IOT devices across a secure network connection. A user-friendly interface web-application allows users to easily register for a parking spot, manage parking spot bookings, and access the parking lot using a generated QR code. This system removes the need for users to pay through a kiosk and place the ticket in their car, making for a time efficient and easy to use system.

Another goal of this design is to improve on the previous solution to this problem. The improvements shall be done with respect to cost efficiency, usability, and security and reliability. Our design aims to leverage an advanced software design architecture to improve efficiency of cameras and image processing, reducing the cost of the system. Additionally, including automated parking verification in our design strengthens the security system, removing the need for manual parking inspection. Enforcing a pre-booking system removes the need for payment verification, although it restricts the usability of the parking system for users. Our design allows users to enter without a booking, this allows anyone to drop-in and park while maintaining a user-friendly payment system.

# Objectives

Some of the features, requirements and performance metrics this project aims to achieve are documented in the section below.

**Main high-level features**

* The APMS shall have a parking lot automation subsystem that automates various processes in a parking lot.
* The parking lot automation subsystem shall be driven by IOT technology based architecture.
* The parking lot automation subsystem shall make use of sensors to monitor parking occupancy.
* The parking lot automation subsystem shall be driven by a central server that communicates with a device with Wi-Fi capabilities.
* The central server shall collect long term parking spot occupancy data to be later analyzed (data analysis).
* The system shall provide a GUI subsystem that allows users to use the parking lot easily and efficiently.
* The GUI subsystem shall provide a parking spot booking interface for users.
* The GUI subsystem shall provide recommendations to the user for parking spots using availability prediction algorithms.

**Parking lot automation subsystem**

* Users must book a parking spot to enter the parking lot.
* Users shall be able to enter only within the time of a valid booking.
* Users must exit within their booked time.
* System shall authenticate users with a booking one at a time.
* Regular parking users shall be able to park at any “regular” spot.
* Premium parking users shall be able to park at a specific premium spot.
* Disabled parking users shall be able to park at any disabled spot.
* System shall have a physical indicator in the parking lot to indicate that a spot is occupied.
* System shall track the occupancy of parking spots.
* Vehicle length and width shall not exceed parking spot length and width.
* System shall use Internet of Things (IOT) technology to implement tracking software for parking spots.
* System shall perform real-time monitoring of parking spots.
* System shall use sensors to detect the presence of a vehicle in a parking spot.
* System shall authenticate users for “special” (premium , disabled, etc.) parking spots.

**Data analysis**

* System shall track the real-time parking occupancy in a remote server.
* System shall collect and store parking occupancy data in a remote database.
* System shall use prediction algorithms using collected data to predict parking occupancy at certain times at specific locations.
* System shall track peak hours at specific parking spots.
* System shall track the most popular parking spots.

**Web-Application GUI**

* Users must have a registered account to login to the application.
* Users shall be able to register an account using the application.
* Users shall be to login to use the application features and services.
* Users shall be able to store personal information associated with their account profile.
* Users shall be able to logout of the application.
* Users shall be able to book parking spots for available times.
* Booking information shall be used to authenticate users at the parking lot.
* Users shall be able to cancel their bookings at any time.
* Users shall be refunded if booking is canceled prior to the booked time slot.
* Users are required to pay at the time of booking.
* Users shall be able to see real-time parking occupancy of spots.
* Users shall be able to see peak hours at specific spots.
* Users shall be able to see occupancy rate of each spot.
* Users shall receive recommended booking times based on predicted availability of parking spots.

**Non-functional/Performance requirements**

* The APMS parking lot automation system shall be time efficient compared to modern traditional methods.
* The APMS parking lot payment system shall be more user friendly and secure than traditional payment methods.
* The APMS shall automate authorization and security to eliminate the need for a security team.
* The APMS shall reduce traffic congestion and improve process efficiency using availability prediction algorithms.
* The APMS shall provide users a touch-free experience at the parking lot.
* The APMS shall provide a user-friendly and effortless experience.

# Approach & Methods

To address the problems in current parking management systems, our design approach prioritizes the following:

* Touch-free system
* Automation
* Minimizing traffic

## Touch-free system

A touch-free system improves the efficiency of the authentication process. The benefits of a touch-free system is more clear when compared to its alternative. Modern parking systems use payment machines and parking slips which require a parking inspection crew. This method increases the risk of failure and maintenance costs of the payment machines. It increases the security risk as there is a chance of users forging a parking slip which goes unnoticed by ticket inspectors. It also allows for human error to affect the performance of the system as there is a need for a security crew.

The solution to this problem is a QR-code based authentication system in which a user scans the uniquely generated QR code to be granted access to the parking lot as shown in **Figure X**. This QR code is also used to verify payment at the time of exit as users are required to scan it at the exit. Users are required to pay for their parking after they park, similarly to how they would pay at a machine in a regular parking lot. If users do not pay for parking within the allotted time they will be charged extra at the time of payment. If a user leaves without paying they will be reported to authorities by their license plate. A camera at the entrance will retrieve a user's license plate before they are allowed to enter the parking lot using pattern recognition technology. Therefore this design ensures a complete touch-free system.



**Figure X:** Example of QR-code based authentication using computer vision.

## Automation

Our design eliminates the need for a management/security team to manually authenticate users inside the parking lot. When a user parks in a “special” location [1], a camera performs character recognition on the license plate, to check if the license plate is from an authorized user . If an unauthorized user is parked in the “special” parking, an alarm will be triggered to warn the user to move their vehicle. If the vehicle is not removed within the allotted time, a parking violation report will be issued to the authorities and they will be charged a fine at payment. If they remove their unauthorized vehicle from the “special” parking the alarm will be turned off.



**Figure X:** Example of parking spot detection (left) and license plate recognition (right).

## Minimizing traffic

This design allows users to make parking reservations along with a view of the availability of spots using a web-application. The web-application is accessible on the internet, instead of on a local network. Allowing users to see the availability of the parking lot means there would be less traffic on public roads as it ensures users are able to park at their desired time and location. Since users can see if a parking lot is full, they will not drive to the location. The web-application will also show patterns in availability/occupancy allowing the user to make an informed trip. This also reduces congestion in the parking lot itself as users that are planning a trip can see the availability of spots ahead of time. Prediction algorithms are used to predict availability of parking spots throughout the day. This is done by collecting daily occupancy data and using a machine learning algorithm to detect patterns. Prediction algorithms can also assist in generating a profit model, by leveraging availability patterns, time-zone based parking rates can be used. The most popular spots can be made premium spots to improve profit and to reduce the congestion associated with popular spots.

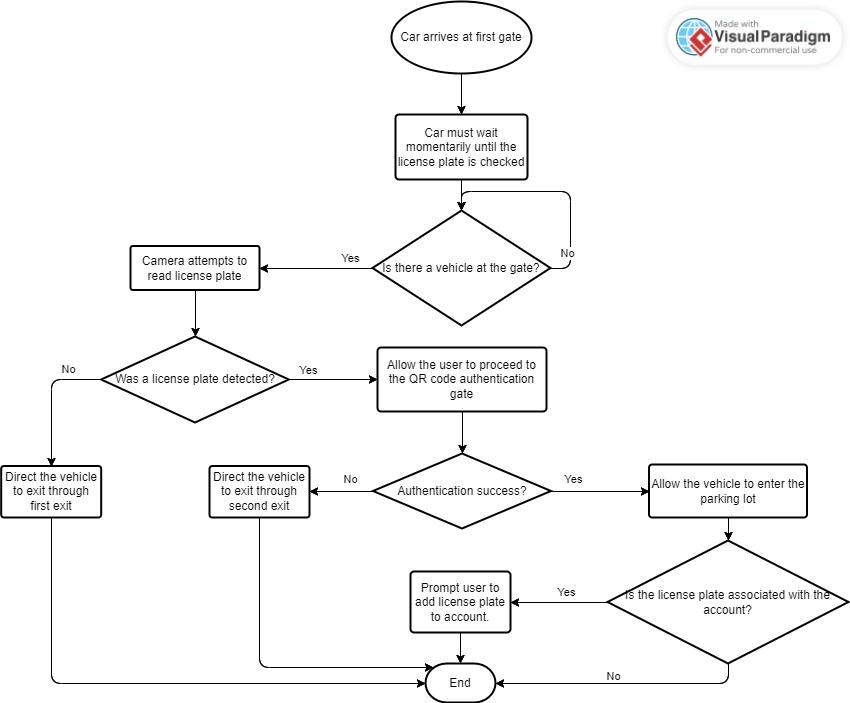
# Design Analysis and Synthesis

The automated parking management system is broken down into smaller subsystems for an analytical and systematic approach to simplifying the problem. Through decomposition, and defining system specific functional requirements, 4 main subsystems handle solving unique problems leading to an overall cohesive and robust system. The primary subsystems include the “Parking lot automation subsystem”, the “Web-application”, the “Data analysis and prediction algorithms”.

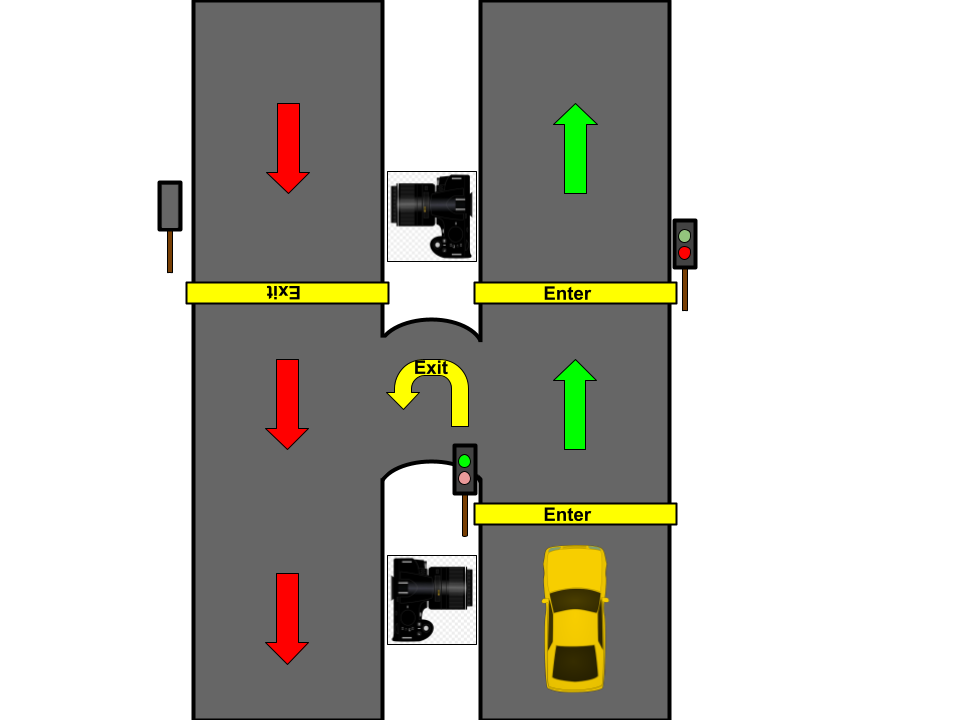
## Parking Lot Automation Subsystem

### Entrance & Exit

The following flowchart **Figure 4.1** describes the user interaction flowchart at the entrance of the parking lot. In summary, the system uses 2-step verification, first it verifies the QR code, then it attempts to scan the license plate. If both are successful, the user is allowed to enter the parking lot. **Figure 4.2** shows the entrance design. The user must scan their entry QR-code to exit the parking lot. During the exit process the system checks if the user paid for their parking within the time limit, and charges the user accordingly. If the user has not paid at the time of exit the license plate will be used to file a report to the authorities.



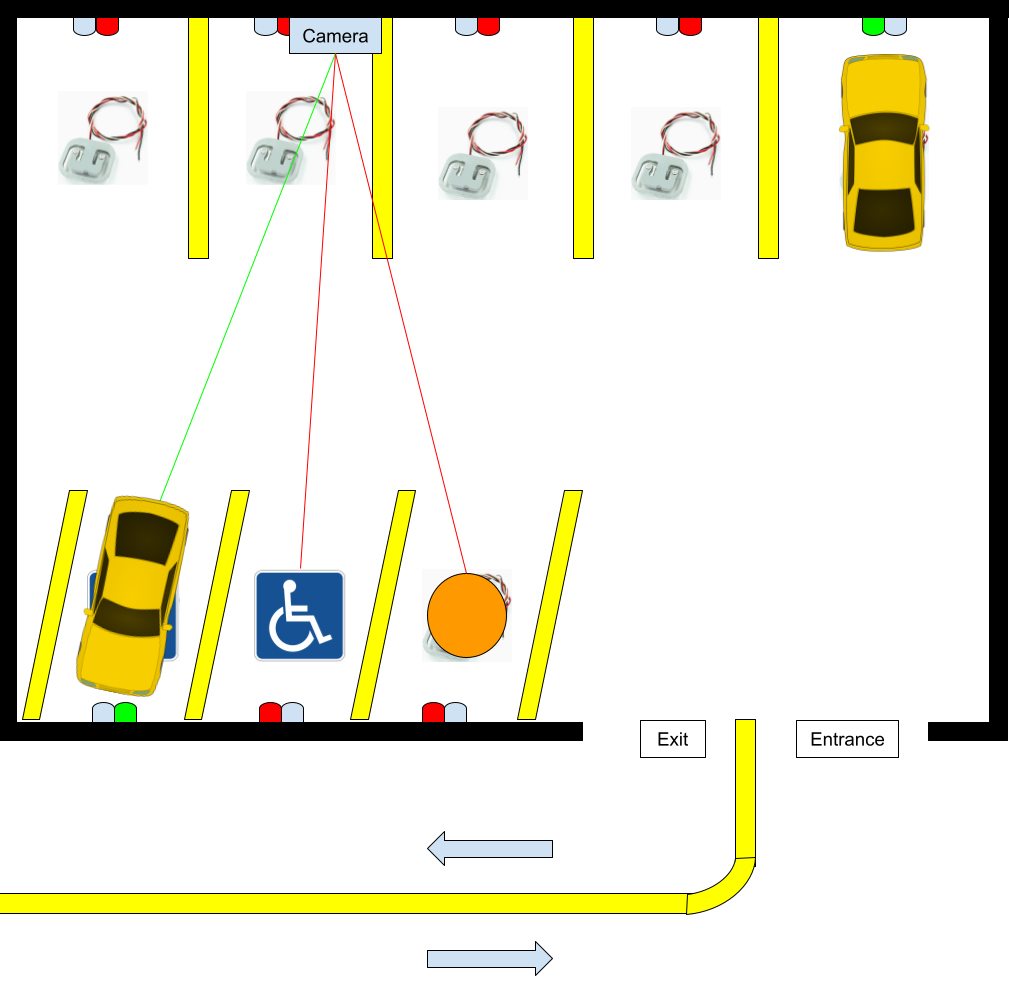
**Figure 4.1:** Flowchart for the entrance authentication process.



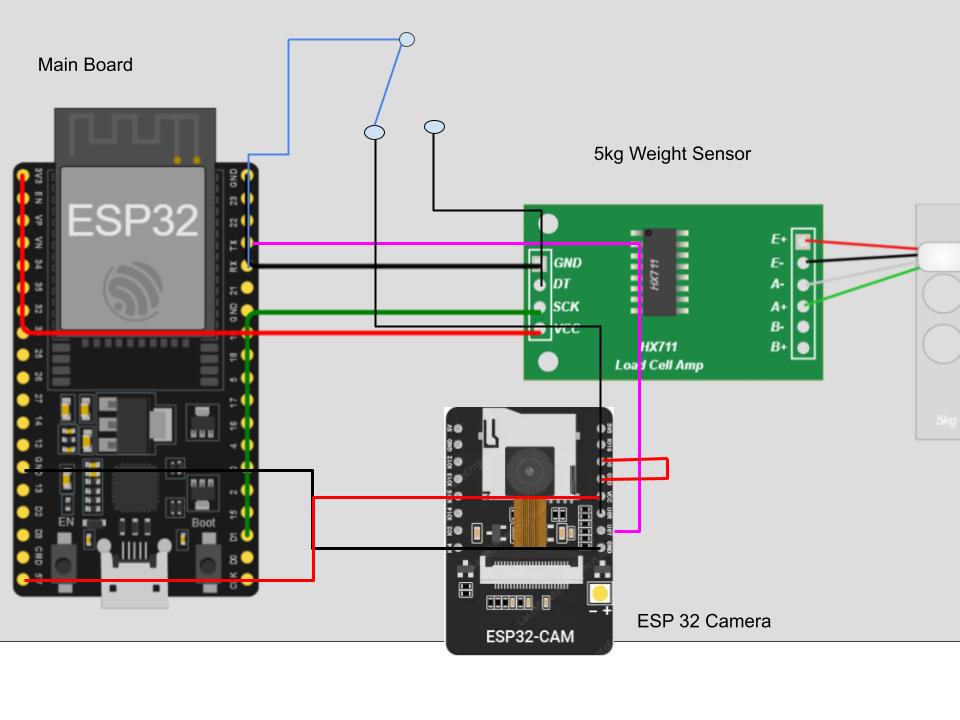
**Figure 4.2:** Parking lot entrance design.

### Parking Lot

A user can park in either a regular parking spot or in a ***special*** parking spot (Any parking spot that requires security inspection), depending on their booking. Special parking spots can be divided into several categories, all of which require parking security inspection, which is done automatically with a camera. Special spots are less common within a parking lot allowing for the strategic placement of cameras for the monitoring of multiple spots at once. Since verification is only done for “special” spots, regular parking users can park at any location provided it is indicated as regular parking, it should also be noted that special parking spots require a pre-booking. In **Figure 4.2** below, the camera monitors the 3 special parking spots, 2 disabled spots and 1 premium spot. When a car parks the camera reads the license plate and queries into the database to authorize the user. If successful the light at the parking spot turns green. The special spots need to go through a verification process, therefore it will take about a minute before the light is green, where the regular parking spot light shall turn green as soon as a car is parking on the load sensor. A load sensor is used at regular spots to detect the presence of a parked vehicle, while in a special parking spot it triggers the start of the verification process.



**Figure 4.3:** Parking Lot Design.



**Figure 4.4:** Sample wiring schematic for load sensor and camera.

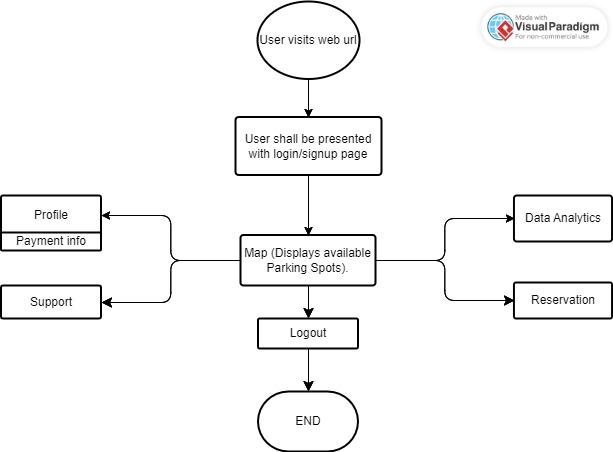
## Web Application

APMS Web application subsystem introduces the power of web technologies to connect users to the system even while outside of the system. There are many tools and frameworks available for developing web applications. Any framework that speeds up and simplifies development is a strong candidate for the current use case. The “React”[5] framework is a popular one with extensive documentation suitable for developing web applications at a rapid pace. Alongside extensive documentation, there exists several plugins and packages for easy plug and play style development resulting in extremely fast developmental times.

Having picked the correct framework, a solid process plan must be laid down that explains the app functionalities at its core. This was achieved based on the previously established functional requirements for the overall system. Figure 5.1 shows the established navigational process of the app. Based on these requirements, the core components were built with accessible web design concepts in mind.

### Frontend

Frontend involves designing the app with user experience in mind, hence UI design becomes a principal component and one that is of the utmost importance to keep the users engaged and proactive. External tools like FIGMA [1] for web design can also be used to accelerate development. FIGMA is an online collaborative tool which serves many user cases. It is primarily known for its simplicity in helping with UI design, UX design, diagramming, brainstorming etc… Web designers often use FIGMA to design entire web applications which can be referenced while building out their own applications. Figures 5.2 through 5.6 are PMS web application prototype design pages made with FIGMA. It consists of a welcome page, sign up page, search page, booking page and a QR code page which cover the core functionalities of the web application by specifications.



**Figure 5.1:** User-GUI FlowChart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Figure 5.2:** Welcome Page | **Figure 5.3:** Sign Up Page | **Figure 5.4:** Search Page | **Figure 5.5:** Booking Page | **Figure 5.6:** QR Code Page |

With the FIGMA pages designed, developing the application in react is trivial.

### Backend

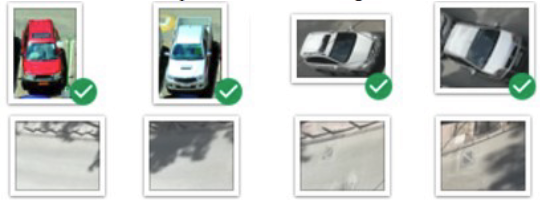
The web application is also connected to the backend server which is a cloud based database solution in Firebase [3], a popular cloud database solution built by Google. It is capable of hosting databases, implementing services, handling authentication and integration of apps and many more. The databases portion of the sub-section will discuss this in more detail. Through optimal integration with the backend, the application

## Data Analysis and Prediction Algorithms

Analysing parking lot data can provide valuable insight behind many variables which can be used to develop prediction algorithms to improve the cohesiveness of the system. Algorithms developed with the parking lot pattern flow in mind can help improve operational efficiency, optimize parking space utilization and enhance the overall user experience. The main data points to look out for include general parking lot information, the occupancy rates, peak hour, seasonal variations. This information becomes crucial for predicting future scenarios like high demand periods, most occupied parking spots or mitigating potential congestion points. It is also necessary to distinguish the data types and data sources and to recognize the meaning behind each data point that is extracted from the system. Camera’s for example provide visual data points which must be extracted to identify the objects. The parking lot system cameras are capable of identifying multiple parking spaces and also track moving objects such as vehicles. Through the identification process, and the classification process, a relationship between the parking space and vehicle can be established. The relationship can be defined with respect to time for example which can help define the occupancy rate on a weekly or monthly basis. Another example is one that is based on parking space costs where a profit model relationship is defined. By defining such relations, further algorithms can be built specifically for future predictability to mitigate and optimize the system, resulting in a more intelligent and responsive parking system infrastructure that not only meets the current demands, but also proactively adapts to evolving patterns.

Primary tools that are used for data collection include the IOT devices which will be integrated within the “Parking Lot Automation Subsystem”. IOT devices mentioned include ESP32, ESPCAM, Weighing sensors. The ESP32 with its wifi capabilities will transmit valuable sensor data from the camera’s and sensors which will form as the base dataset. To conserve and optimize algorithm usage, data from the cameras are transmitted only when entry sensors are triggered which includes the ESP CAM and the weight sensors, indicating that vehicles are entering the system. With the captured images, they are further chopped to optimize for database storage. The dataset will be stored in a cloud database which will be further discussed in the next sub-section.

One of the ways to solve parking spot occupancy rates is to detect and keep track of the empty parking spots with respect to time. Deep learning algorithms can be utilized for the task of recognizing empty and preoccupied parking spaces. First an appropriate dataset is required which must also be labeled, meaning, the parking spot images must confirm to one of the predefined states which are either occupied, empty or blocking where blocking can mean that the parking spot is covered by anything other than a vehicle. With a labeled dataset, Deep learning algorithms can be utilized to classify them based on the labels [2]. Figure 6.1 shows a typical labelling process where only the parking spot is used as a label image to classify the status of the parking space.



**Figure 6.1:** [2] - Labelling process.

From the image dataset, CNN can be applied to extract image features to match with the labelling images to output the status of the image. CNN, a kernel is a feature detector that traverses images and generates new images. The kernel through convolution generates new images depending on the number of layers that are utilized. Layers can be defined to identify similar features found in parking lots such as headlights, windshield, mirrors etc…The process in the end classifies the image. Based on the image classification, the occupancy rates can be calculated with respect to time. Overtime, as the dataset grows, predictive model becomes more accurate leading to a more predictive and robust system

## Cloud Database Solution

After collecting the license plate from the customer, it is stored into the Cloud Database with records of all customers that have used the automated parking system and their membership status. This is implemented through firebase. Firebase[4] is an application from Google which can provide different types of services. For the purposes of this project, it will be used as for the database that is to be implemented.

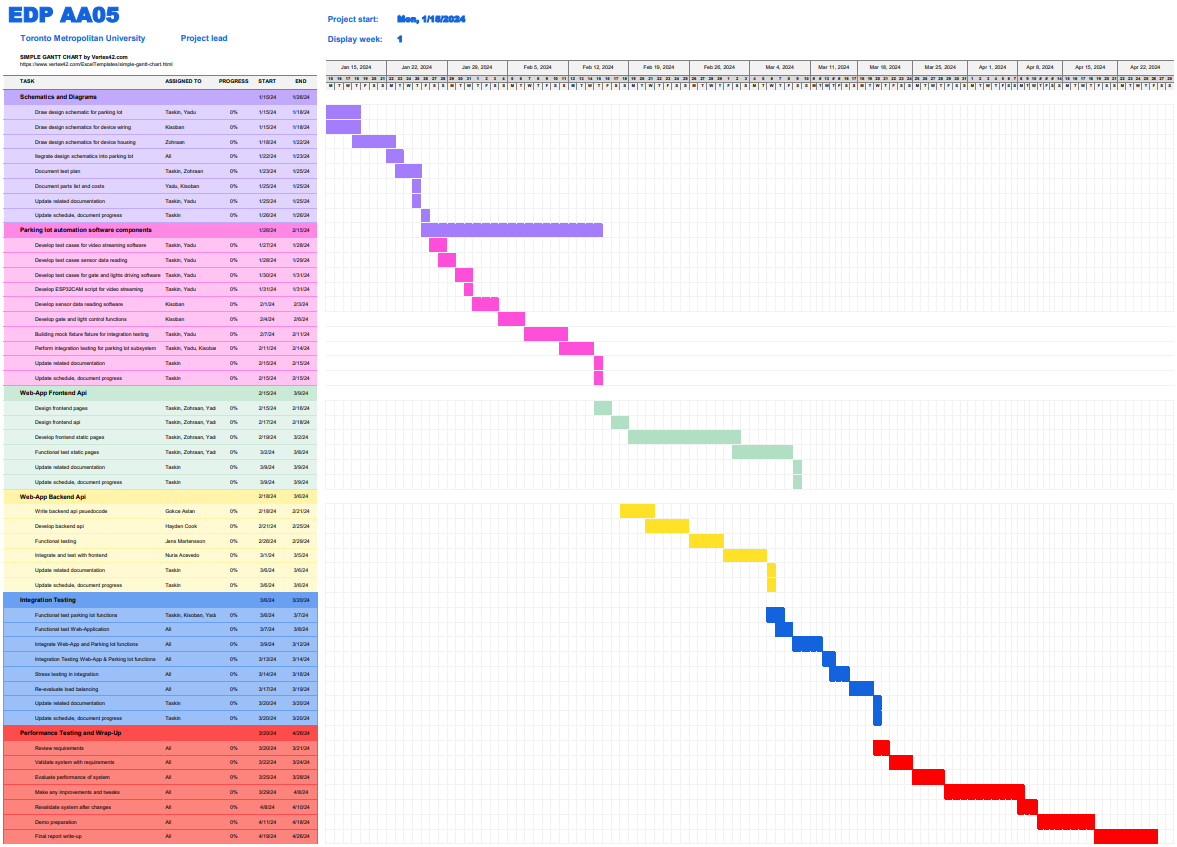
The process of database development involves 6 primary steps which include:

1. Defining purpose and goal: The main purpose of the database is to store information that serves the PMS web application, serve image data to build ML models and conduct data analysis.
2. Gathering requirements: Requirements for the database involves analyzing the functional requirements of the overall system. The key entities that play and interact with the system include the:
   1. User: User information such as name, email, license plate, qr code etc.. will be stored for identification purposes.
   2. Parking spot: Unique information regarding each parking spot such as its Id, its priority, its occupancy status are recorded in the database.
   3. Sensor data: Sensor data in the form of event trigger timings, frequency of events, the class of events or camera feed/ camera images are recorded in the database for data analysis purposes.
   4. Parking Lot information: Key information that identifies the parking lot itself, such as its name, address etc are stored in the database.
3. Creating a schema: Based on the above collected requirements, a database table schema is developed where 4 primary tables are defined which include:
   1. USER Table
   2. BOOKING Table
   3. SENSOR\_DATA Table
   4. PARKING\_LOT Table
4. Creating an entity relationship diagram: An entity relationship diagram is essential to identify any redundancy and cause for table reduction where relations can be unnecessary.
5. Implementing and Testing: Table is then implemented in Firebase and thorough testing is conducted for debugging and performance optimization purposes.
6. Optimizing, deploying and maintenance: Finally the database once established and deployed can continue to be maintained, optimized for minor performance gains.

Overall, the database that will be implemented will contribute to a user-friendly environment such that customers will feel comfortable in parking using the automated parking system.

# Implementation Plan (GANTT Chart)

The gantt chart for the Winter 2024 semester is shown below, to clearly see the details download the file by clicking on the link in the image below.

[](https://docs.google.com/spreadsheets/d/1UhgNxM_baBUC7WxHzpK3SNZCTXv4qBSa/edit?usp=sharing&ouid=100496205286305151144&rtpof=true&sd=true)

**Figure 7.1:** Gantt chart showing the task breakdown for Winter 2024.

# 

# Conclusions

In summary, this report includes the design for all the high-level system requirements for the APMS. The design fulfills the functional requirements for each of the subsystems, as well as the performance requirements. This solution is more reliable and user-friendly than alternative designs as it leverages machine learning to recognize patterns within a select area, providing automated security inspection on multiple spots at the same time. Using license plate verification allows for a more secure payment system, as users can be reported to city authorities if they fail to pay. Other designs limit users to pre-booked access to the parking lot, our design allows users to drop-in and pay in regular parking, reserving special spots for booking only.

Some issues that have not been addressed is the business model for the parking system, describing what payment options are available and at which times. Another issue being the need for sensors in special parking spaces, as it still needs to be discussed if camera verification is sufficient for monitoring special parking spaces. The main difficulty of this project is the lack of progress in terms of low-level system design involving detailed schematics, parts, measurements, software flowcharts and architectures. This is because more time was spent perfecting high-level design decisions as it would later impact lower level design.

The next steps would be to prepare an implementation plan and prepare all necessary technical design documents. This would involve the completion of all low-level drawings, schematics, parts and specifications list and an execution plan for the parking lot subsystem. For the software, the diagrams and flowcharts describing the Web-application api, parking lot functions and data-collection functions need to be created. These steps refer to the first phase of the Implementation Plan, which if done correctly should ensure the completion of the project.

# 

# 

# References

1. *The Collaborative Interface Design Tool*. Figma. (n.d.). https://www.figma.com/
2. S. Gören, D. F. Óncevarlk, K. D. Yldz and T. Z. Hakyemez, "On-Street Parking Spot Detection for Smart Cities," 2019 IEEE International Smart Cities Conference (ISC2), Casablanca, Morocco, 2019, pp. 292-295, doi: 10.1109/ISC246665.2019.9071760.
3. Keras, “Keras: The Python Deep Learning library,” [online] Available: <https://keras.io/>
4. Google. (n.d.). Google. <https://firebase.google.com/>
5. React. (n.d.). https://react.dev/

# Appendices

N/A