# ParkMe: A Smart-Parking Application

CS4991 Capstone Report, 2023

James Zhu
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
jhz4hqc

**ABSTRACT**
Distracted driving is one of the leading causes of car accidents and finding a parking spot is one of the main distractions that every driver must deal with eventually. Our proposed solution, ParkMe, is a smart parking application which informs users where available parking can be found, thus reducing the time drivers spend to find parking and eliminating the stress and distraction involved with the activity. The prototype we created for ParkMe was developed primarily with C++ and used three esp32 Arduino boards. The use of one of the Arduino boards acted as a server to communicate with while the other two Arduino devices acted as the users. Using this miniature model, we were able to ensure that each device was able to send and receive information about parking availability and provided a method for each device to communicate with each other. By eliminating the need to 'find' a parking spot, we hope to remove the many distractions involved in finding parking, lowering the risk of the activity. The future of ParkMe is to expand it past its prototype phase. Moving past Arduino boards, we hope to create a mobile app which users can post and find information regarding parking in their area. By doing so, we aim to make our application more accessible and easier to use/interact with.

## 1. INTRODUCTION

Annually, 50,000 crashes occur in parking lots and parking garages, representing around 20 percent of all automobile accidents which results in 500 deaths and 60,000 injuries according to the National Safety Council (RMG, 2023). As drivers undergo the process of parking, they must continue driving until they find an empty spot. During this period, drivers are likely distracted by numerous external factors such as other parked cars, parking signs, parking meters or parking rates, color of the curb, etc. Additionally, parking lots and garages are likely to have additional pedestrian traffic as other drivers get into or out of their cars.

All of these factors combine to make parking an extremely variable activity and one of the most likely activities leading to a car accident. These factors are exacerbated in cities, with increased road distractions and foot traffic. It is for this reason that our application, ParkMe, is specifically targeted towards cities, where traffic is much worse and available parking is scarce. By eliminating the need for drivers to find a parking spot on their own, they will be able to focus on the road conditions more and drive more safely, without added distractions.

## 2. RELATED WORKS

The main inspiration for our application was in the parking garages of Tyson's Corner in McLean, Virginia. As described on the website for Tyson's Corner Mall, within their parking garages, they have implemented an LED light fixture to set the status of each parking spot—red for occupied and green for available (Macerich, n.d.). Instead of needing to drive down and look through each parking spot for an open space, drivers can now simply look down at the row of LED lights to discern whether there is an available spot. Additionally, before entering the parking

garage, a sign is posted near the entrance indicating the number of spots available on each floor. Despite the large amount of automobile and human traffic at Tyson's Corner, these LED light fixtures made our parking and driving experience much safer and quicker—which is the same experience we want the users of ParkMe to have.

Early in the development of ParkMe, we found many different applications with similar premises. One such inspiration was ParkMobile. ParkMobile is one of the most popular parking applications which displays several metered or paid parking locations on a map. ParkMobile is one of the industry's first and most popular app for parking. Using this app, one can reserve and pay for their parking for any specified number of minutes directly through the app, which provides real-time parking availability to users (Sotomayor, 2023). This revolutionary idea eliminated many physical parking meters in exchange for a much easier and more accessible online payment method. Because of the success and ease of use of ParkMobile, many aspects of our application were directly inspired by this predecessor including the features for reserving a parking spot and using a map to help users locate the parking spot more easily.

The main difference between our app and ParkMobile is that we wanted to map where free parking was available, not just paid parking. From a financial standpoint, ParkMobile does not and likely will not ever display free parking. A large portion of the revenue gained by ParkMobile is from users making payments, like parking meters, to reserve a spot. If an option was given to users to find free parking, a large portion of their users would opt for that instead. We differ however because our main priority is safety. Commonly, to distinguish between paid parking and free parking, drivers would need to read parking signs nearby which will instruct them where parking is allowed, on what days, and specific times that are paid versus free. Having to parse through nearly a paragraph of text from such a sign is a prime example of one of the distractions we would like to eliminate. Ultimately, we want users to utilize ParkMe as a tool to help them find a free parking spot quickly, efficiently, and as easily as using ParkMobile.

## 3. PROJECT DESIGN

### 3.1 MQTT (Message Queuing Telemetry Transport)

The prototype of ParkMe was developed primarily using C++ and utilized three esp32 Arduino boards. These boards communicated through the MQTT protocol, which is one method Internet of Things (IoT) devices use to communicate with each other. Figure 1: MQTT Process depicts this interaction.


**Figure 1: MQTT Process**

Within MQTT, there are 3 main processes: the broker, publisher, and subscriber. The broker acts as a server through which IoT devices communicate and generally sets up some type of wireless connection system; in our case Wi-Fi, where other devices can connect to, as well. Once connected through Wi-Fi, information is processed through the broker and is responsible for sending and receiving new information from all connected devices.

Publishers, as the name suggests, publish data into the broker and primarily function as a sender. Similarly, a subscriber's function is to subscribe to the broker, receiving any new data that is processed by the broker. It is important to note that a device can be both a publisher and a subscriber since these functions are not mutually exclusive. For ParkMe, this dual functionality is utilized. Each successive esp32 board aside from the broker is both a publisher and a subscriber and is supposed to mimic the functionality of a

smartphone and represent an automobile. For this reason, the prototype can be expanded with more esp32 boards, which would all act as another automobile.

### 3.2 Process Flow of ParkMe

Before using ParkMe, two specific setup steps must be taken. First, each individual parking spot must be pre-calculated. Every parking spot is roughly a rectangle, which is depicted through its four corners. To check whether someone is in a parking spot, we take their GPS location and verify whether it falls within the boundaries of a rectangular parking spot. Computationally, this can be a difficult process as it takes four coordinates to obtain just one parking spot. However, once set up, the parking spot does not ever need to be recalculated. The information for each parking lot is stored within the broker. When a GPS location is sent, the broker is responsible for completing the verification that the user's location matches with a parking spot.

Second, each esp32 board must have a GPS module attached to obtain the geographical data for each board. A button press on the esp32 board would act as a vehicle switching its parking status, and the change in status would be published to the broker. If a car was not parked already, the button press would indicate that the car has now parked at its current location. Conversely, pressing the button on an already-parked spot represents leaving the spot and freeing the availability of that spot. In essence, button presses send a signal to the broker that data is being published. In the case of ParkMe, the data is the switching of parking availability.

To operate ParkMe in its prototype version, we are assuming that each vehicle has an Arduino board available. With this assumption, we each individual esp32 board represents an individual vehicle. When the application begins, users are immediately sent a list of parking spot that are available near them. In the prototype version of ParkMe, the distance between the user and each parking spot is a direct distance—ignoring any road restrictions, obstacles, traffic, and other factors. This may lead to many inaccuracies regarding which parking spots are close to the user but can still give a general sense of where parking spot may be available. The list of available parking spots is constantly updated by the broker and will be re-sent to each user every few seconds, providing the user with constantly updated information.

When the user has found a parking spot and parked the vehicle, they must press the button on the esp32 board, sending the current coordinate of the user to the broker. The broker, upon receiving new data, will go through its list of parking spots and confirm that the user has parked in a valid parking spot by verifying that the GPS location of the esp32 board falls within the rectangular boundary of a parking spot. If the spot is invalid, a message will be sent to the user indicating that either the user's current location is not a parking spot or that specific parking spot was not registered in the setup phase. If the spot is valid, the availability of that parking spot will switch to unavailable and will no longer be shown to other users. Whenever the user is ready to leave a parking spot, they simply need to press the button again on the esp32 board. The broker performs the same operations as mentioned previously only this time, that parking spot is switched to available again and will now be shown to other users.

## 4. ANTICIPATED RESULTS

The prototype of ParkMe was never published and therefore, the results are merely speculative. The desired result of ParkMe is to reduce the number of parking related car accidents by eliminating parking distractions. By directly navigating drivers to their parking spots, drivers no longer need to find a spot and parse through numerous road and parking restrictions to get there. In this way, we are freeing drivers to focus on driving and road conditions.

This goal is achievable with ParkMe but only when proper foundations have been set. First, ParkMe relies on a large user base. Without a large majority of drivers using ParkMe, there would be many inaccuracies regarding available parking spots. For instance, consider the following situation: 5% of a parking lot is occupied by ParkMe users. Other incoming users may see many available parking spots in this lot since only 5% of the lot is being reported as taken. Upon arrival, the ideal situation is that all users within that lot are ParkMe users and only 5% of the parking lot is truly taken—as reported in the app. However, what is more likely to occur, especially in large cities, is that the entire parking lot is full. This occurs because the other 95% of drivers in that parking lot are not ParkMe users and are not reporting that these parking spots are being taken. These inaccuracies and high error rates are only reduced as the number of users increases, but in the early stages of development, acquiring users may be difficult.

The second major foundation that needs to be set for ParkMe is within the setup phase. As mentioned previously, each parking spot must be pre-calculated as a rectangular area. For ParkMe to work, a large number of parking lots must agree to use ParkMe and register their spots into the system. Since this step is only performed once and without much cost, this step is doable but will require parking lots to invest in the system. In the same way that ParkMobile was able to eliminate the need for physical parking meters and get parking lots to use their electronic system, ParkMe should be able to establish a relationship with parking lot managers and get registered into the ParkMe's system. ParkMe has a strong potential for positive development; however, is unlikely to yield extraordinary results immediately.

## 5. CONCLUSION

ParkMe was developed as the final project for the class Internet of Things at the University of Virginia. The overall goal of the application is to reduce the number of car accidents when parking a vehicle, in turn reducing the number of parking-related injuries and fatalities. The goal of the final project was to utilize IoT devices through some type of wireless connections, which we had been studying all semester, to solve some proposed problem. Because of this, many of the design decisions were fitted to satisfy the requirements of the course. For instance, the use of esp32 Arduino boards is primarily due to the requirements set forth by the class. We understand that Arduino boards are not readily accessible and are not the most intuitive way to create our application. However, it would serve as an adequate prototype for what we wanted to create.

ParkMe is more than just an application, it is an idea. The idea is to use technology and the tools around us to assist users, making everyday life easier. Countless technologies before our idea built on this philosophy. Navigation, real time traffic reports, dash cameras, and far more have revolutionized the way we drive, creating a safer and quicker way of transportation. ParkMe, an application inspired by such beliefs, strives to be another tool humans can use in the future to aid in their driving experience.

## 6. FUTURE WORK

One of the major weaknesses of ParkMe is the use of the geographic coordinate system (GCS). The geographic coordinate system is used to map out the Earth's surface, allowing us to pinpoint a certain location if given longitude, which measures east to west distance, and latitude, which measures north to south distance. Because GCS is used to map out such a large area, a single degree of change in any direction represents around 70 miles. ParkMe uses this coordinate system as well; however, we are measuring objects on a much smaller scale. The average parking spot has a 16-foot length and 8-foot width. To accurately assess which parking spot a vehicle is currently in, we would likely have to maintain a 1-meter accuracy, which is about one ten thousandths of a degree in accuracy.

Additionally, GCS coordinates are unable to differentiate height, making the prototype unusable for multi-layered parking garages. With our current hardware, the esp32 boards, maintaining a 1-meter accuracy is difficult.

Another weakness of ParkMe is its reliance on having the proper foundations. Needing a large user base and having the infrastructure to input parking spots individually is a daunting task for a new startup. Given this, we have considered an alternative to starting the application ground up and instead, using ParkMe as an extension of other, more popular applications such as Google Maps or another popular parking application. With this method, we would be able to observe the results of ParkMe more quickly as these applications have an established user base and likely have infrastructure to setup ParkMe on a large scale.

Our immediate goal for ParkMe is to release it from its current prototype model with new and more accessible hardware, and to fix the most glaring issues the app faces. Incrementally, as we shift away from the prototype, we may find that some issues can be fixed with upgraded hardware. Other issues, however, may require a different design decision. Despite these issues, the core idea of ParkMe—helping users find parking and lowering the risk of fatalities and injuries to drivers—will remain.

## 7. ACKNOWLEDGMENTS

A special thank you to my group members who worked on this project with me, Josh Lee and Abel Ressom, both undergraduate students at the University of Virginia. Additionally, I would like to credit our professors, Brad Campbell and Nabeel Nasir for providing the knowledge necessary to complete this project. Our professors also provided us with all the hardware equipment needed to complete the prototype of ParkMe, as well as much-needed guidance throughout the project. Their love for teaching and IoT devices inspired us to think outside the box, ultimately resulting in the idea of ParkMe.

## REFERENCES

RMG. (2023, March 8). *How Often Do Car Crashes Occur in Parking Lots And Garages*. Nelson Macneil Rayfield. https://nelsonmacneil.com/blog/how-common-are-parking-lot-car-crashes/#:~:text=Parking%20Lot%20Accidents%20Statistics

Sotomayor , D. (2023, October 17). *What Is Smart Parking In IoT* [Review of *What Is Smart Parking In IoT*]. Robots.net. https://robots.net/tech/what-is-smart-parking-in-iot/

Macerich. (n.d.). *Tysons Corner Center | Visit Us*. Www.tysonscornercenter.com. Retrieved December 1, 2023, from https://www.tysonscornercenter.com/Visit