Parking Garage Management System: A Smart Solution for Modern Parking Challenges

Chris Najem, Mark Rehn, and Tony Mai

Department of Electrical & Computer Engineering, University of Central Florida, Orlando, Florida, *32816-2450*

Abstract **--- This paper presents the Parking Garage Management System, an intelligent solution designed to address common frustrations associated with urban parking. By integrating real-time occupancy monitoring, guided navigation, enhanced security through License Plate Recognition (LPR), and an automated payment system, our project aims to streamline the parking experience for drivers and administrators alike. The system utilizes ultrasonic sensors for real-time spot availability, overhead LED indicators for visual guidance, and LPR cameras for permit validation and security alerts. A robust power management system, including a battery backup, ensures continuous operation. This paper details the system's design, component selection, hardware design and software architecture, demonstrating a scalable and reliable approach to modern parking challenges.**

Index Terms **- Parking management, License Plate Recognition, Ultrasonic sensors, Real-time monitoring, Battery management system, Smart parking.**

I. INTRODUCTION

The rapid pace of urbanization and the increasing vehicle population have turned the simple act of parking into a frustrating daily challenge. In dense areas like city centers and university campuses, drivers often spend excessive time and fuel circling garages in search of a spot—worsening congestion and emissions. Despite broad technological progress, many parking systems remain outdated and minimally automated.

This project, the Parking Garage Management System, tackles these issues through an intelligent, scalable solution that incorporates real-time monitoring, guided navigation, license plate recognition (LPR), and streamlined payment processes. The goal is to reduce time spent searching for parking, lower fuel waste, and bring order to a chaotic experience.

From the moment a vehicle approaches the parking facility, our system initiates its intelligent operation. A strategically placed digital signage display at the garage entrance provides real-time updates on parking spot availability for each level, empowering drivers to make immediate, informed decisions regarding their optimal parking destination. As drivers navigate within the parking levels, overhead LED indicators intuitively guide them: a green light signifies an available parking space, while a red light indicates an occupied one. For users who prefer proactive planning, a dedicated website offer real-time occupancy counters, enabling them to determine parking availability even before their arrival.

Beyond improving navigation and convenience, the system significantly bolsters security through the implementation of License Plate Recognition (LPR) software. LPR cameras, positioned at both entry and exit points of the garage, automatically scan vehicle license plates. This functionality enables the system to instantly verify valid parking permits and generate immediate alerts for security personnel if an unauthorized vehicle is detected or if a vehicle overstays its allotted parking duration. For visitors without pre-registered permits, an integrated online payment portal facilitates seamless permit purchase with minimal friction.

Built with reliability and user experience in mind, the system supports multi-level garages, features a centralized admin dashboard, and includes a battery backup to ensure continuous operation of critical functions during power outages. This project represents a shift toward intelligent, automated, and resilient parking infrastructure for the modern city.

II. SYSTEM COMPONENTS

The Parking Garage Management System consists of multiple subsystems working together to manage real-time parking availability, enforce permit compliance, and enhance user interaction. These components include sensors, microcontrollers, cameras, power control modules, and user interface systems. This section introduces the hardware and software components integrated into the final system.

A. Microcontroller Unit (MCU)

The system uses one microcontroller unit (MCU) to process signals from input devices and control output

components. The MCU is responsible for controlling the ultrasonic sensors, parking LED indicators, and sending parking availability to the database. It processes input from sensors to determine car presence and communicates with display units and the database to update occupancy data. The MCU that the team chose to use was the ESP32-S3 for the multiple GPIO pins, allowing for more devices to be connected, whether it's for more ultrasonic sensors or more displays. With built-in Wi-Fi and Bluetooth LE, the ESP32-S3 also enabled wireless expansion without needing additional modules. Its dual-core processor and robust interrupt handling allowed the system to run real-time occupancy detection and web updates in parallel.

B. Ultrasonic Sensors

Ultrasonic sensors are mounted above each parking spot to detect whether a vehicle is present. These sensors send data to the MCU to help determine the status of the parking spot. If a car is detected, the system updates the LED indicator to red and decreases the number of available parking spots in the database and on physical signs. The ultrasonic sensor the team chose was the HC-SR04 ultrasonic sensor for the price and ease of use. The HC-SR04 operates by emitting ultrasonic pulses and calculating distance based on the return time. For improved accuracy and noise rejection in the parking garage, sensors were placed at a consistent height and angle above each spot. The system also filters transient readings using moving averages to prevent false occupancy updates from temporary obstacles or environmental interference.

C. LED Indicators and Display

Each parking spot has an LED indicator that changes color based on occupancy: green when available and red when occupied. Additionally, a digital display near the garage entrance shows the current number of free spaces as well as signs on each floor showing the amount of free parking spaces on that floor. This information is synchronized with an online website in real-time. The team chose LEDs that were bright enough to see in a parking garage and the displays used were the MAX7219 Dot Matrix LED displays. To ensure maximum visibility in all lighting conditions, the team used high-intensity red and green LEDs with diffused casings for wider viewing angles. The MAX7219-based matrix displays were chained together using SPI to allow for longer words to be displayed. The digital signs were mounted at vehicle eye-level and synchronized to the backend using real-time database polling.

D. Camera System

Two license plate recognition (LPR) cameras are installed at the garage entrance and exit. These cameras capture and process the license plate of each car to determine whether the vehicle is registered or has a permit. This data is used to decide whether to notify security, allow the car to remain, or log its departure.

Each camera is connected to a dedicated Raspberry Pi for processing. The cameras we chose were the Raspberry Pi Camera Module V2.

E. Raspberry Pi Units

Each camera is paired with a Raspberry Pi that handles local image processing and communicates license plate data to the central system. The Raspberry Pis are also responsible for forwarding data to the main database and interfacing with the UI logic on the website and app. Power to the Raspberry Pis is regulated to ensure stability during operations. The Raspberry Pi 4 was the team's first choice due to price difference and prior experience. Each unit was enclosed in a ventilated housing to avoid thermal throttling in outdoor garage conditions.

F. Power Management System

To ensure consistent performance, the system includes a power management setup consisting of a primary power supply, battery backup, regulators, and a power path controller. The battery management system ensures safe charging and discharging of backup batteries and allows seamless power switching in case of outages. Voltage regulators stepped the battery output down to match the ESP32 and peripheral requirements. For the batteries, the team went with 18650 batteries and a HX-3S-FL25A-A BMS.

G. Database and Web Interface

The system maintains a centralized database to record license plates, track parking activity, and update spot availability. The user interface includes a mobile-friendly app and website showing live occupancy, permit status, and entry/exit logs. The database also enables analytics and long-term logging for administrative review. MongoDB was the team's choice for the database.

III. SYSTEM CONCEPT

This The Parking Garage Management System is designed to automate and optimize the processes of vehicle detection, permit verification, occupancy tracking, and user communication within a multi-level parking facility. The system leverages sensor input, visual recognition, and real-time data updates to ensure an efficient and secure parking experience for both users and administrators.

At its core, the system performs two main tasks:

- It verifies if a vehicle has permission to park by analyzing its license plate.
- It updates and displays real-time parking availability based on sensor feedback.

Upon a vehicle's arrival at the garage entrance, the system's front-facing camera captures the license plate. The Raspberry Pi runs a lightweight Python-based recognition model, which achieves high accuracy under varying lighting and plate types. The Raspberry Pi then processes that captured image and cross-references the extracted data with the internal database to determine permit status. If the vehicle is unauthorized, the system logs the event and optionally triggers a security alert. If authorized, the car proceeds into the garage.

Once inside, ultrasonic sensors mounted above each parking spot detect occupancy. This data is sent to an ESP32-S3 microcontroller, which controls the corresponding RGB LED indicators (green for available, red for occupied). Simultaneously, this information updates a MAX7219-based LED matrix display near the entrance and is pushed to the online web app and dashboard for real-time access. The ESP32 packages sensor status using JSON over HTTP and relays it to the MongoDB backend hosted in the cloud, which feeds both the signage and admin dashboard. This diagram shows how the software part of our project works, with each side of the flowchart showing each main task and how it works.

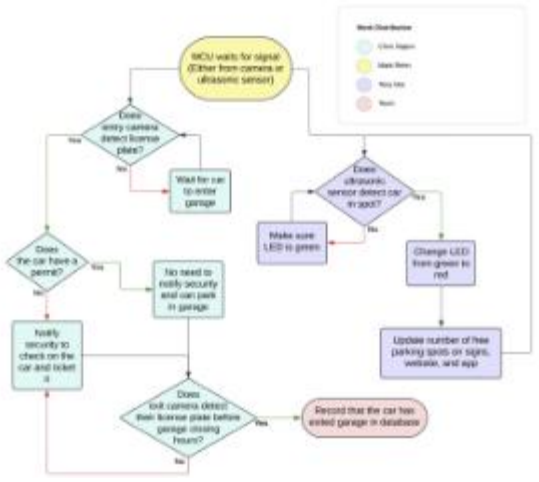

Figure 1: Parking Garage System Flowchart

In addition to its data-processing architecture, the system is equipped with a robust power design to maintain continuous operation. Each Raspberry Pi and ESP32 unit is powered through a regulated circuit that includes 18650 lithium-ion batteries, a BQ24130 charger, and automatic power switching. This ensures uninterrupted performance during outages or power instability. The next diagram shows how each part of our project is connected all together, showing where each part is connected in the main system.

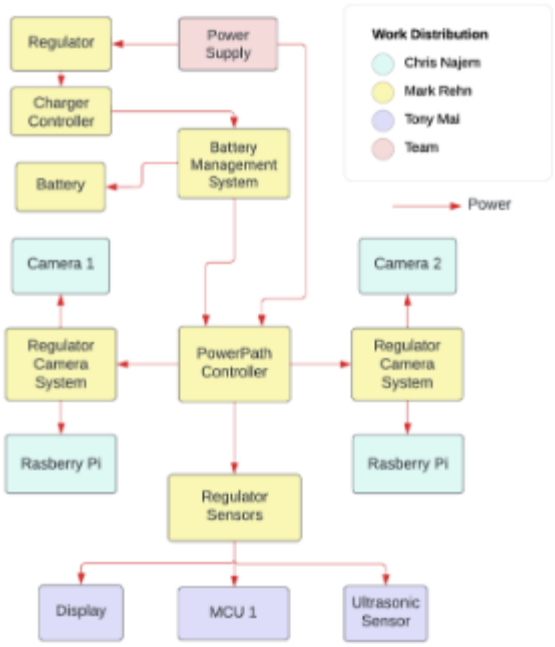

Figure 2: Power and Component Distribution

By combining license plate recognition, sensor-based detection, cloud-synchronized displays, and layered power control, the system offers a scalable and intelligent approach to managing parking infrastructure. It is modular, expandable to additional levels or zones, and provides an intuitive interface for administrators and users alike.

IV. HARDWARE DESIGN

The hardware design integrates all power, control, and sensing subsystems into a compact and modular platform. The system architecture was developed to support real-time monitoring, reliable power management, and seamless data communication across multiple microcontrollers. A high-level overview of the hardware subsystems is shown in Figure 3, which illustrates the key components and their interconnections on the main PCB.

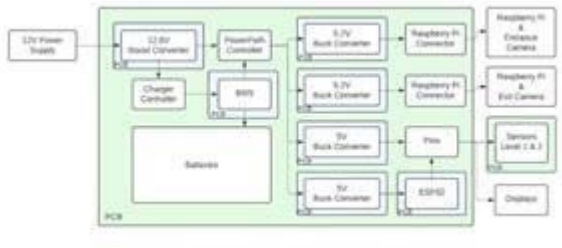

Figure 3: Subsystem Block Diagram

The system is powered by a 12V wall adapter, which supplies input to a boost converter that steps the voltage up to 12.6V. This voltage feeds two critical power management circuits: the PowerPath Controller and the Charger Controller. The PowerPath Controller ensures seamless switching between the primary power source and the onboard battery pack. Meanwhile, the Charger Controller works in conjunction with a three-cell Battery Management System (BMS) to regulate charging current and voltage, ensuring safe and efficient battery operation.

Once power is conditioned, several buck converters are used to distribute stable voltages to the rest of the system:

- 5V buck converters power the ESP32-S3 microcontroller, ultrasonic sensors, LED indicators, and displays.
- 5.2V buck converters power the two Raspberry Pi modules, each of which is paired with a camera.

At the core of the sensing subsystem is the ESP32-S3, which manages occupancy detection for parking spaces. Each spot includes an HC-SR04 ultrasonic sensor to detect vehicle presence and red/green LEDs for visual indication. The ESP32 also controls the displays that show real-time availability to incoming drivers. All these components interface with the ESP32 through GPIO pins.

The ESP32 connects to the database over Wi-Fi to upload live parking availability data, which is used by the website and admin dashboard. In parallel, two Raspberry Pi units are dedicated to License Plate Recognition (LPR). Each Pi is paired with a camera (one at the garage entrance and one at the exit) to capture and process images. These Raspberry Pis also upload data directly to the database to log license plate entries and exits, verify permits, and generate alerts for unauthorized or overstaying vehicles. This distributed architecture cleanly separates sensing and visualization from image processing and security, resulting in a modular, scalable smart parking solution.

Figure 4 provides a closer view of the power and communication flow among the ESP32, sensors, displays, and Raspberry Pi subsystems.

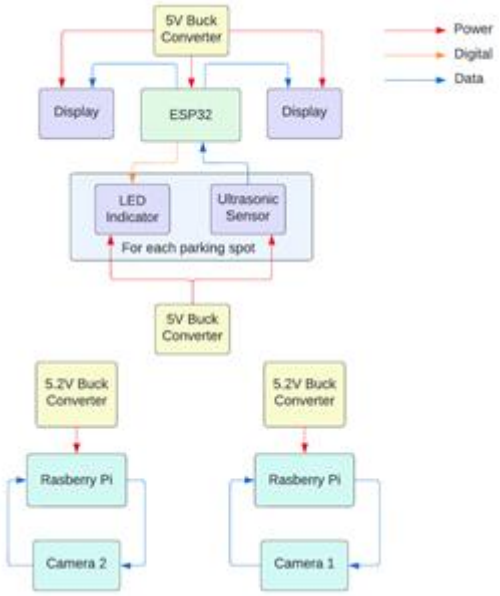

Figure 4: Communication and Power Flow Diagram for the ESP32 and Raspberry Pi Subsystems

All these components interface through a custom-designed main PCB, which consolidates power routing, signal control, and physical connections. Removable modules, such as the ESP32 board and voltage regulator circuits, are integrated using headers, allowing for ease of replacement, testing, and future upgrades.

This integrated and modular hardware design provides a reliable foundation for the smart parking system, enabling accurate detection, responsive feedback, and efficient data handling.

V. PRINTED CIRCUIT BOARD DESIGN

The PCB design integrates the system's power regulation, sensing, and control subsystems into a reliable and compact hardware platform. All schematics and layouts were created using Autodesk Fusion 360 and fabricated through JLCPCB. Throughout the design process, care was taken to adhere to industry best practices and layout standards such as IPC-2221, which govern trace spacing, grounding, and signal integrity for reliable electrical performance.

A total of five different custom PCBs were developed for the system: the Main PCB, Boost Converter PCB, Buck Converter PCB, ESP32 PCB, and Sensor PCB. The Main

PCB serves as the central hub and is the only non-removable board. It integrates the PowerPath Controller (LTC4412), Charger Controller (BQ24130), and interfaces for all modular subcomponents. A Boost Converter, based on the LM3478 IC, steps up the 12V input to 12.6V for battery charging. Four removable Buck Converter PCBs, each using the LM2596 IC, are used to step down the voltage as needed: two 5.2V converters supply power to the Raspberry Pi modules, while two 5V converters power the ESP32, ultrasonic sensors, and displays.

Figure 5 shows the schematic layout of the Main PCB, which consolidates power switching, charging, and signal routing for all subsystems.

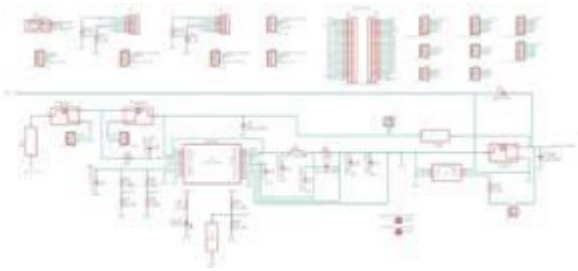

Figure 5: Overall System Schematic

The ESP32 PCB is built around the ESP32-S3-WROOM-1-N8R8 module and interfaces with several GPIO-controlled peripherals, as described in the hardware design section. Designed as a modular unit, it connects to the Main PCB via pin headers, enabling rapid prototyping, simplified testing, and easy upgrades.

The Buck Converter PCB uses a shared layout for both 5V and 5.2V versions. While the PCB design remains the same across all units, the component values are adjusted depending on the target voltage output. This approach provides flexibility while maintaining consistency in design and assembly.

The Sensor PCB is responsible for managing the ultrasonic sensors and LED indicators that detect and signal parking space availability. It connects to the Main PCB through wired connections to male pin headers. These headers are tied to the ESP32's GPIO pins, as well as to 5V and ground lines, which were routed and exposed on the Main PCB specifically to support this interface. Using wires allows each sensor module to be physically positioned above its corresponding parking spot while maintaining a clean and modular connection back to the control system.

To ensure signal integrity and minimize electromagnetic interference (EMI), all PCBs were designed with continuous ground planes and polygon stitching. Bypass and bulk capacitors were placed near key ICs and regulators to minimize current loop area and improve power delivery stability. The use of header-based modularity across the system enhances ease of testing, streamlines debugging, and supports future scalability of the hardware platform.

VI. SOFTWARE DESIGN

The Parking Garage Management System's software architecture is designed to orchestrate real-time data acquisition, processing, and user interaction. It comprises distinct yet interconnected modules for the ESP32 microcontroller, the Raspberry Pi-based License Plate Recognition (LPR) system, and the user-facing website/application. This section details the design of these software components and their collaborative operation.

A. Overall System Architecture

The overarching software architecture is designed to facilitate seamless communication and data flow between the physical hardware components and the digital user interfaces. It manages vehicle entry and exit, parking spot occupancy detection, and security enforcement. The system continuously monitors signals from cameras and ultrasonic sensors. Upon detecting a vehicle or a change in spot occupancy, it processes the information, updates relevant databases, and triggers necessary actions or notifications. The system relies heavily on Wi-Fi communication protocols to ensure robust data exchange between the embedded devices and the central database.

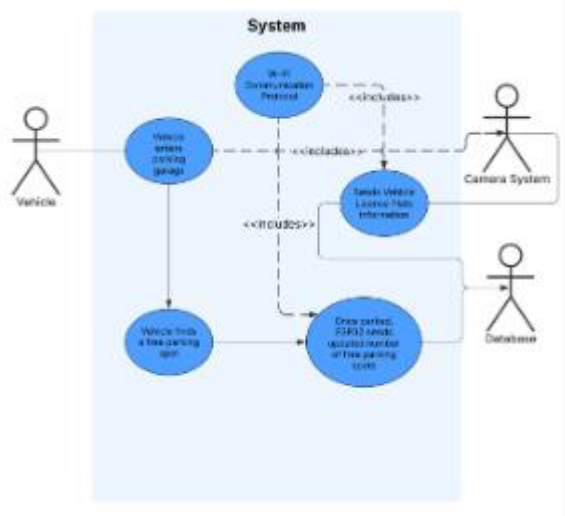

Figure 6: Overall System Use Case Diagram

B. ESP32 System Logic

The ESP32 microcontroller is central to real-time parking spot management. Its primary function is to interpret data from ultrasonic sensors and control the corresponding LED indicators. The system continuously measures the distance to a parking spot. If the distance falls below a predefined threshold (indicating occupancy), the green LED turns off, and the red LED turns on. Conversely, if the spot becomes vacant, the LEDs revert to their default state (green on, red off). This logic ensures immediate visual feedback on parking availability and updates the central parking spot counter displayed on the website and other interfaces.

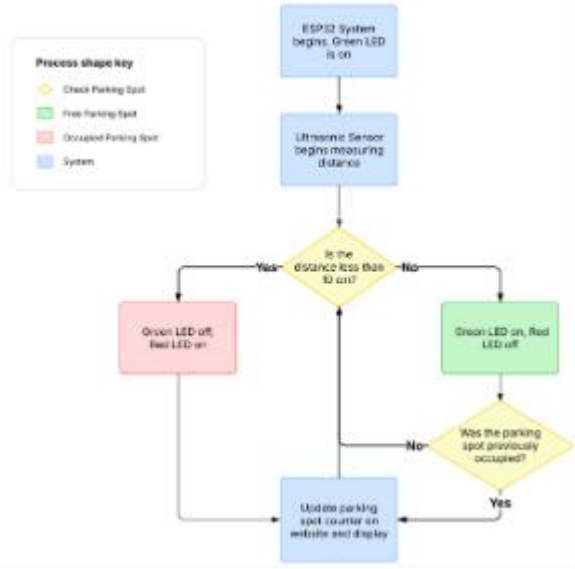

Figure 7: ESP32 System Flowchart

C. LPR System Flow

The License Plate Recognition (LPR) system is a critical security and automation component. It processes camera feeds to detect and recognize vehicle license plates. Upon detection, the system queries a database to verify permit validity. If a permit is valid, the vehicle is allowed to proceed without further action. If no valid permit is found or if the vehicle is unauthorized, a security notification is triggered. The system also tracks vehicle exits, flagging any vehicles that do not exit before garage closing hours, thus enhancing security and compliance. This process ensures efficient entry/exit management and robust monitoring.

Figure 8: LPR System Flowchart

The License Plate Recognition (LPR) system is built upon a distributed and interconnected architecture to ensure robust and scalable operation. At the hardware layer, the system utilizes two dedicated Raspberry Pi units: one positioned as the "Entry Camera" and the other as the "Exit Camera." These Raspberry Pis are responsible for capturing real-time video streams and performing initial processing for license plate detection. The extracted and processed license plate data is then transmitted directly to the MongoDB Atlas database, which serves as the persistent storage for all license plate logs, valid permit information, and vehicle status updates. A central backend server, developed using Flask, interacts with this MongoDB Atlas database to retrieve license plate logs, verify permit validity, and manage security alerts. This Flask backend then serves the data and necessary logic to the frontend web application. The figure below visually represents the LPR system's architecture components.

Figure 9: LPR System architecture

D. Website System Overview

The web-based application serves as the primary interface for both customers and administrators, offering real-time parking availability, permit management, and security monitoring. For customers, it provides a home screen displaying parking availability, a permit page for

purchasing various permit types (daily, weekly, monthly, 6-month, annual), and login/registration functionalities. Administrators have access to a dedicated login, enabling them to view vehicle authorization, manage permits, and review security alerts for unauthorized or overstaying vehicles.

Figure 10: Website System Diagram

VII. TESTING

Our system testing has been going well as most of the components work as intended. The 3 main parts of the system are the power, parking sensor, and license plate recognition.

A. Power Testing

For our backup power system, all components work together correctly for seamless power switching and charging. Our system was able to switch from the wall adapter to the batteries quickly, which allows our system to continue working even when power is cut off unexpectedly. The batteries have been tested extensively, being discharged fully to test the charging capabilities of the system. Even with multiple Raspberry Pis, parking sensors, and displays connected to our system, both the wall adapter and battery system can handle it.

B. Parking Sensor Testing

We performed 2 tests for the parking sensors: one indoors with a smaller 3D printed stand to simulate the parking sensors connected to the ceiling of a parking garage, and one in a garage to have a more accurate real-life test with a car. The indoor testing was to show how fast the system updates the displays to show the most accurate parking availability. We had multiple objects move in and out of the “parking spaces” to see if the sensors could handle multiple cars entering and exiting a parking space, which they could. The LED illumination was bright enough to tell if a space was free, but the red LED was not as bright as we hoped.

C. License Plate Recognition Testing

For the license plate recognition testing, we tested both with printed out license plate images and a real car as well. The images of the license plates were used for testing with the entrance and exit camera, to see if the database could log the license plates correctly and see when they entered and exited accurately with the timeslots. The outdoor testing with a real car only used the entrance camera, to see if the camera could scan a real license plate for an accurate real-life test. Both tests were successful, and the license plates were detected in the database. One thing that could be better would be to get a better camera, as that could lead to faster and more accurate image processing for the license plates.

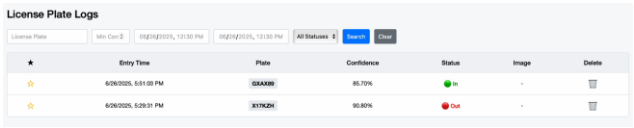

Figure 11: License Plate Logs Website

VIII. CONCLUSION

The Parking Garage Management System project successfully addresses the prevalent challenges associated with urban parking, transforming a frustrating daily task into an efficient and secure experience. Initiated to mitigate common issues such as wasted time, fuel consumption, and traffic congestion, the system integrates advanced technologies inspired by existing smart parking solutions.

Designed for scalability and ease of installation, our solution requires no extensive construction, making it suitable for integration into existing parking infrastructures. This all-in-one system offers enhanced user convenience, improved security, and reliable backup power, marking a significant step towards modernizing parking facilities. This project not only provides a practical solution to a widespread urban problem but also serves as a valuable demonstration of integrated hardware and software design in a real-world application.

This report has detailed the project's goals, comprehensive research into technologies and components, and the software/hardware system diagrams. This project has provided invaluable experience in group collaboration and technical problem-solving, which will be crucial for our

future careers. We hope this smart parking solution can benefit parking facilities and drivers alike.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project advisor, Dr. Wei, for his invaluable guidance and continuous support throughout this project. We also extend our appreciation to Dr. Weeks for his crucial contributions to the PCB hardware implementation, which was instrumental in the successful completion of our project.

THE ENGINEERS

Chris Najem is a 23-year old Computer Engineering student graduating in Summer 2025. Chris has a strong passion for hardware design, particularly in the areas of silicon architecture, system design, and embedded real-time systems. He is excited by the challenges of developing efficient, high-performance hardware that pushes the boundaries of modern technology.

Mark Rehn is a 24-year old Electrical Engineering student, graduating in Summer 2025 with a Bachelor's degree. Originally from Brazil, Mark is an international student who hopes to pursue a career in hardware development or embedded systems.

Tony Mai is a 22-year old Computer Engineering student, graduating in Summer 2025 with a Bachelor's. Tony hopes to get a career in computer hardware with companies like AMD, Intel, etc.

REFERENCES

[1] Espressif, "ESP32-S3 SoC," [Online]. Available: https://www.espressif.com/en/products/socs/esp32-s3

[2] SparkFun, "HC-SR04 Ultrasonic Distance Sensor," [Online]. Available: https://www.sparkfun.com/products/15569

[3] Adafruit, "MAX7219 8x8 LED Matrix Display," [Online]. Available: https://www.adafruit.com/product/453

[4] Raspberry Pi Foundation, "Raspberry Pi Camera Module V2," [Online]. Available: https://www.raspberrypi.com/products/camera-module-v2/

[5] Raspberry Pi Foundation, "Raspberry Pi 4 Model B," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[6] Texas Instruments, "BQ24130 Standalone 1-3 cell Buck battery charger for Li-ion and Super capacitor," [Online]. Available: https://www.ti.com/product/BQ24130

[7] BatteryJunction, "Panasonic NCR 18650," [Online]. Available: https://www.batteryjunction.com/panasonic-ncr18650b-3200

[8] PhippsElectronics "HX-3S-FL25A-A 3S 25A 18650 BMS," [Online]. Available: https://www.phippselectronics.com/product/hx-3s-fl25a-li-ion-18650-battery-protection-bms-board-3s-25a/

[9] MongoDB, "MongoDB Atlas," [Online]. Available: https://www.mongodb.com/cloud/atlas