

Creating an Automated Parking System Enabled by IoT Technology

<https://sddec25-09.sd.ece.iastate.edu>

Client & Advisor: Md Maruf Ahamed

sddec25-09: Thomas Olson, Joseph Schmidt, Harley
Peacher, Sullivan Hart

| sddec25-09

| 05-05-2025

About Us



Team Members

Harley Peacher SE - Frontend

Thomas Olson SE - Frontend

Joseph Schmidt CprE - Server Development

Sullivan Hart CprE - Hardware

Problem Statement

- Finding a parking spot is currently inefficient, time-consuming, and frustrating.
- Drivers often resort to circling lots aimlessly, especially during busy hours.
- Unipark will increase convenience, reduce parking search time, and optimize lot usage.



Different Users

Admin User

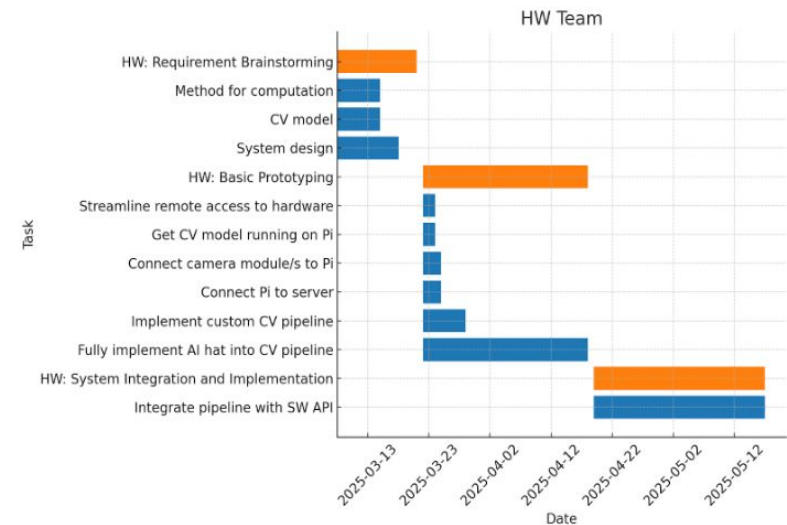
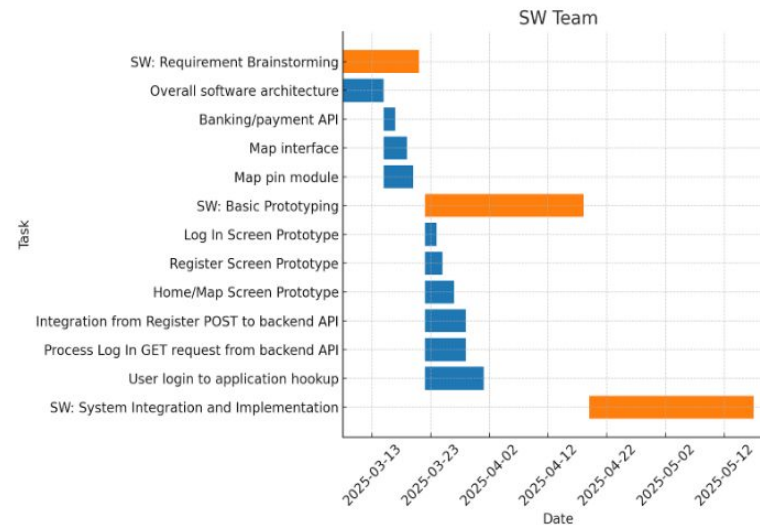
- In charge of configuring and maintaining parking lot
- Sets up hardware
- Configures parking lot in website
- Sets parking lot rates

Parker

- Rushed parkers - focused on finding spot quickly and easily
- Elderly and Disabled parkers - looking for handicap and needs clear and intuitive UI
- Patient Parkers - looking for optimal spot
- Proactive Parkers - wants to secure a parking spot in advance

Requirements

- Detection of vehicle in specific parking spots from camera feed
- User-friendly mobile application for parking lot occupancy and reserving available spots
- Secure payment functionalities through mobile application
- End-to-end connection



Ethical Considerations

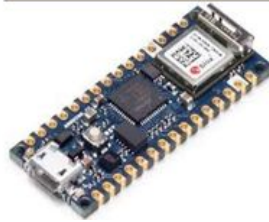
- Handling sensitive user data (passwords, payment, location, etc.)
 - Encrypting data on backend
 - Adding HTTPS for all client-server communications
- Privacy
 - Attackers could try to access parking lot images
 - Access to vehicle and person location
- AI Model Concerns
 - AI Model could potentially have bias, misidentifying a car/parking spot
- App Use
 - Driving while on phone in parking lot



Previous SmartPark Implementation



Arduino Nano 33 IoT	WiFi (NINA)	Ultrasonic Sensor
- WiFi capability - Low cost	Board communication to server	- Accurate - Team familiarity



React Native	Stripe	Firebase
- Uses Javascript - Large community - Our team has experience	- Easy to use API - No setup or monthly fees - Customizable	- Scalable Backend Services - Integrated Analytics



Continuation Project

- sddec24-17

What changed?

- Ultrasonic sensors
- Arduino
- Firebase

Resources kept

- React Native
- Stripe

Motivations for Redesign

More accurate detection

- Increased Accuracy
- Richer information

Scalability

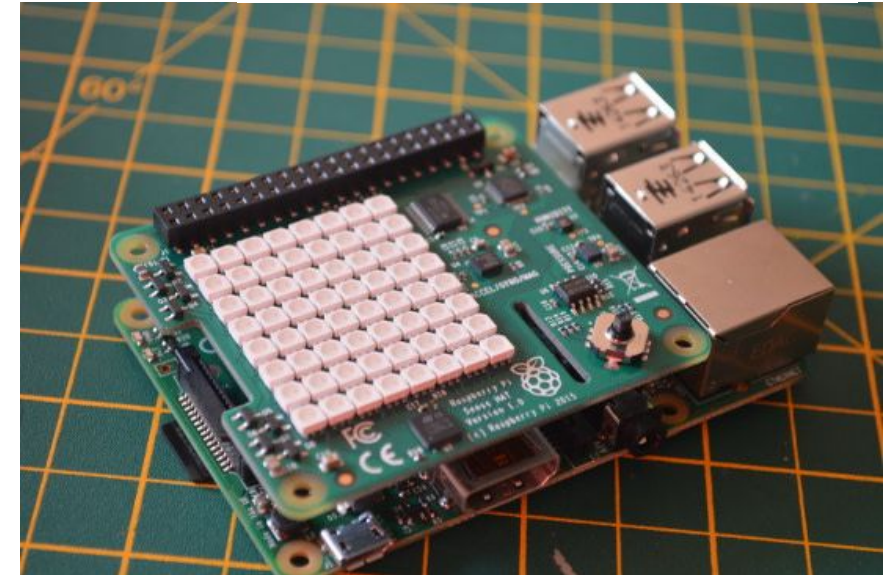
- More flexibility with camera data

Additional Features

- License plate recognition
- Parking lot monitoring

Familiarity

- Used in previous courses, and projects



Projection Introduction

Raspberry Pi w/ Camera

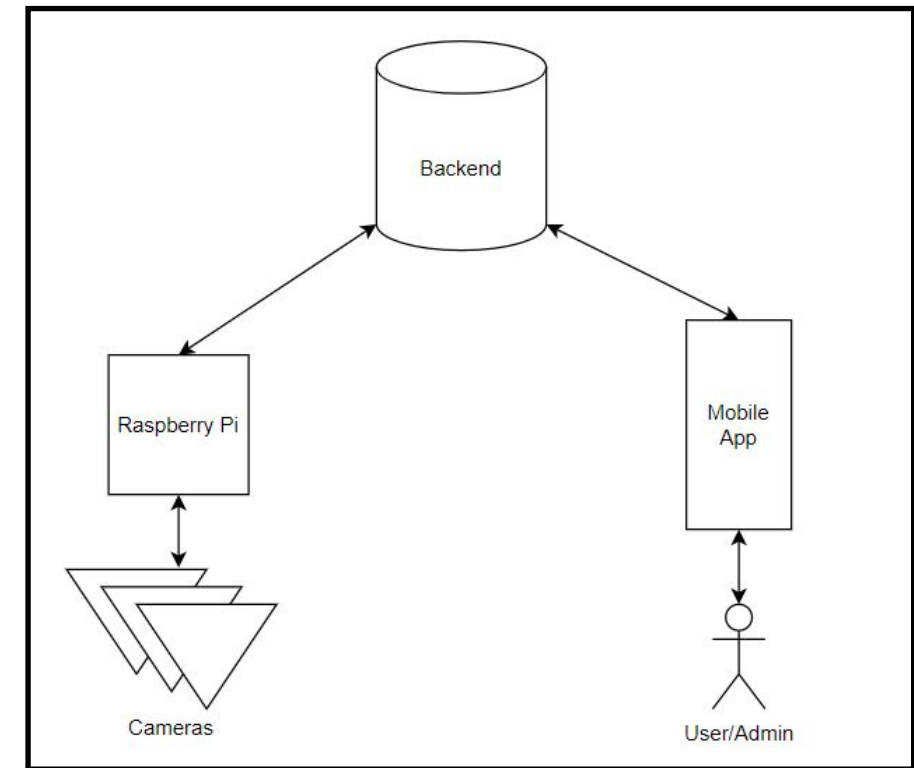
- Data Capturing

Backend Server

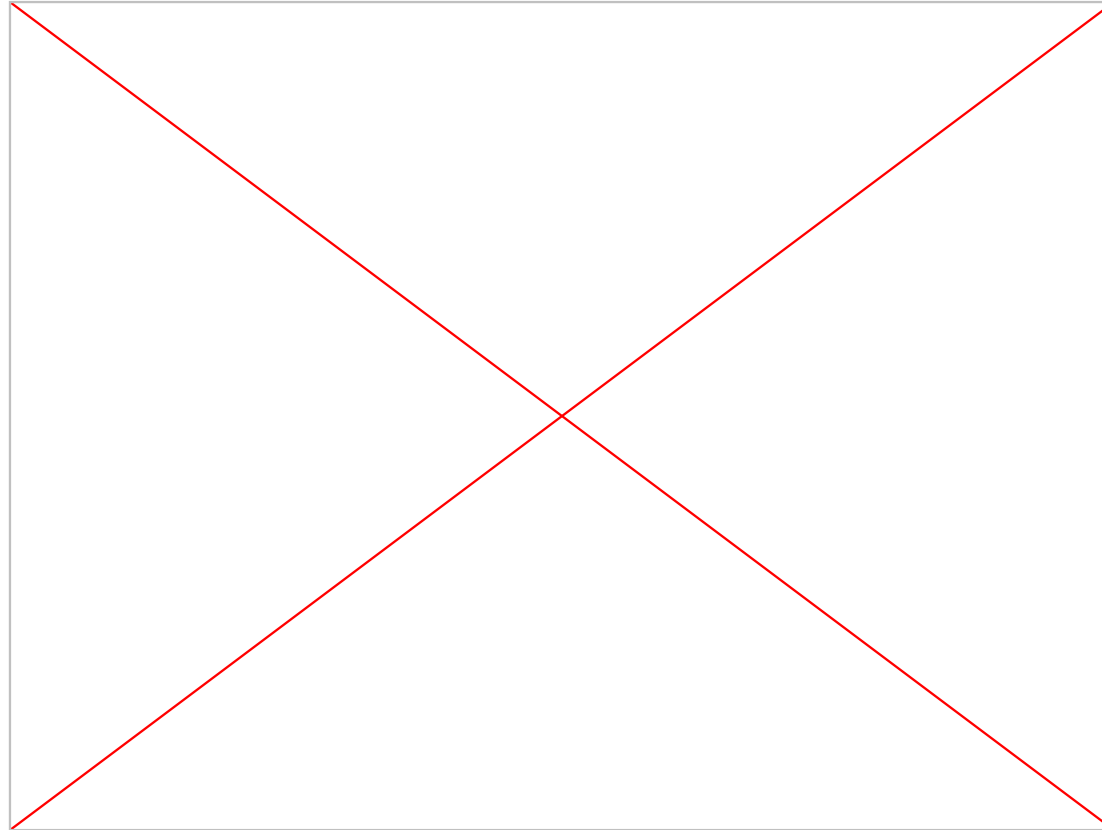
- Hold user and parking data

Mobile App

- Users find and reserve spots
- Admin manages parking lot



Frontend Demo Walkthrough Video



Frontend Implementation

- **Application Built With:**
 - **React Native** - Cross-platform mobile framework with native components
 - **Axios** - HTTP client for API requests
 - **Stripe** - Handles secure in-app payments
- **Application Structure:**
 - **Login/Register page** - Auth via POST requests
 - **Map/Home page** - Map, location, and availability
 - **Reserve page** - Lot information fetched
 - **Payment page** - Stripe flow and reservation trigger

Server Configuration

Docker

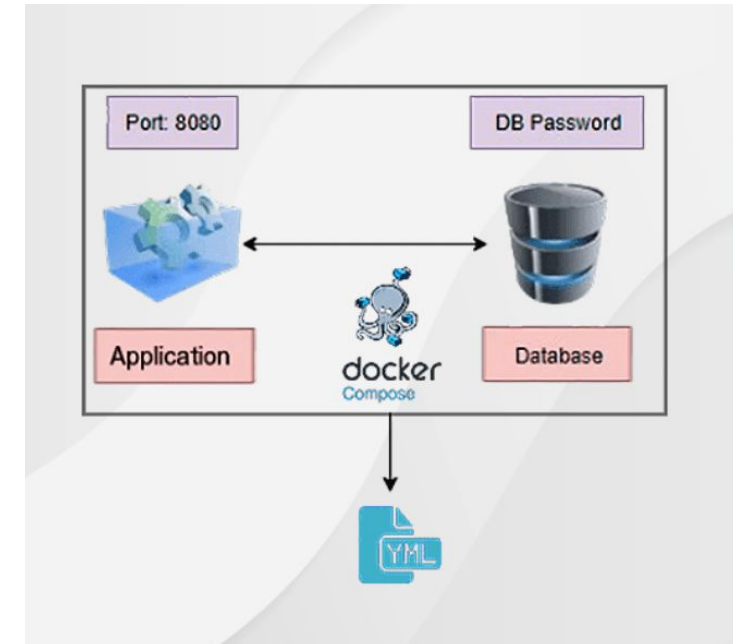
- Portability
- Scalability

Docker Compose

- Spring Boot container
- MySQL container

IASTATE Virtual Machine

- Free and configurable
- Reverse Proxy



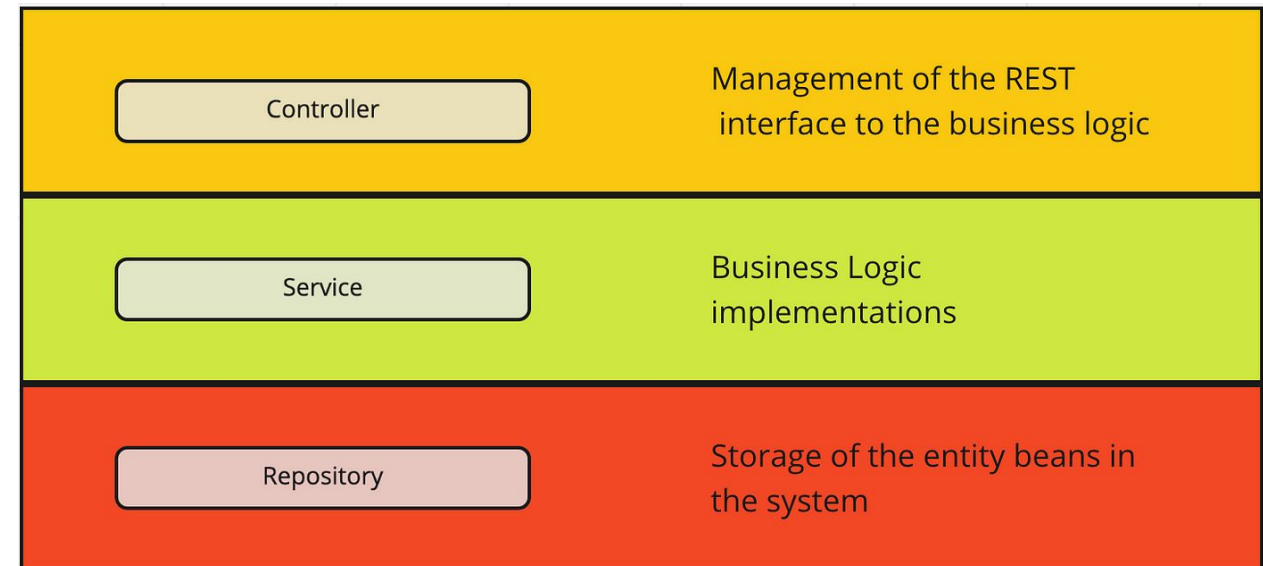
Backend Application

Spring Boot

- Manages complex data relationships
- Controller-service-repository architectural model
- RESTful APIs

Postman

- Testing endpoints



Hardware Implementation

Camera Module

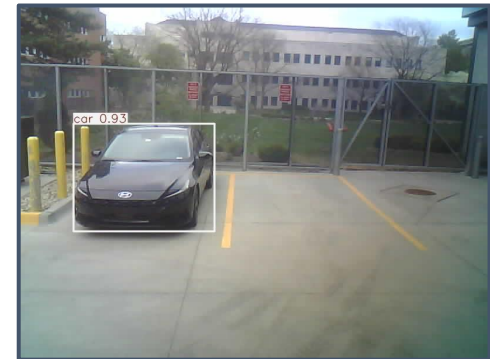
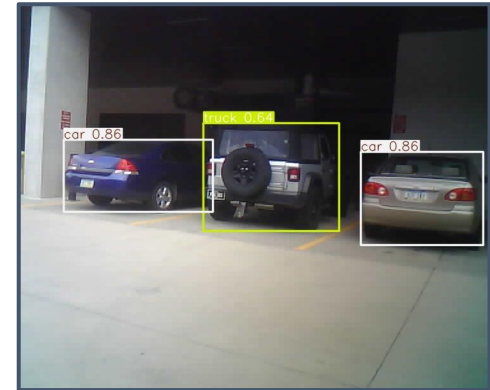
- ESP32 Camera Module
- 3.7V Lithium Battery
- Boost converter and Charging Board

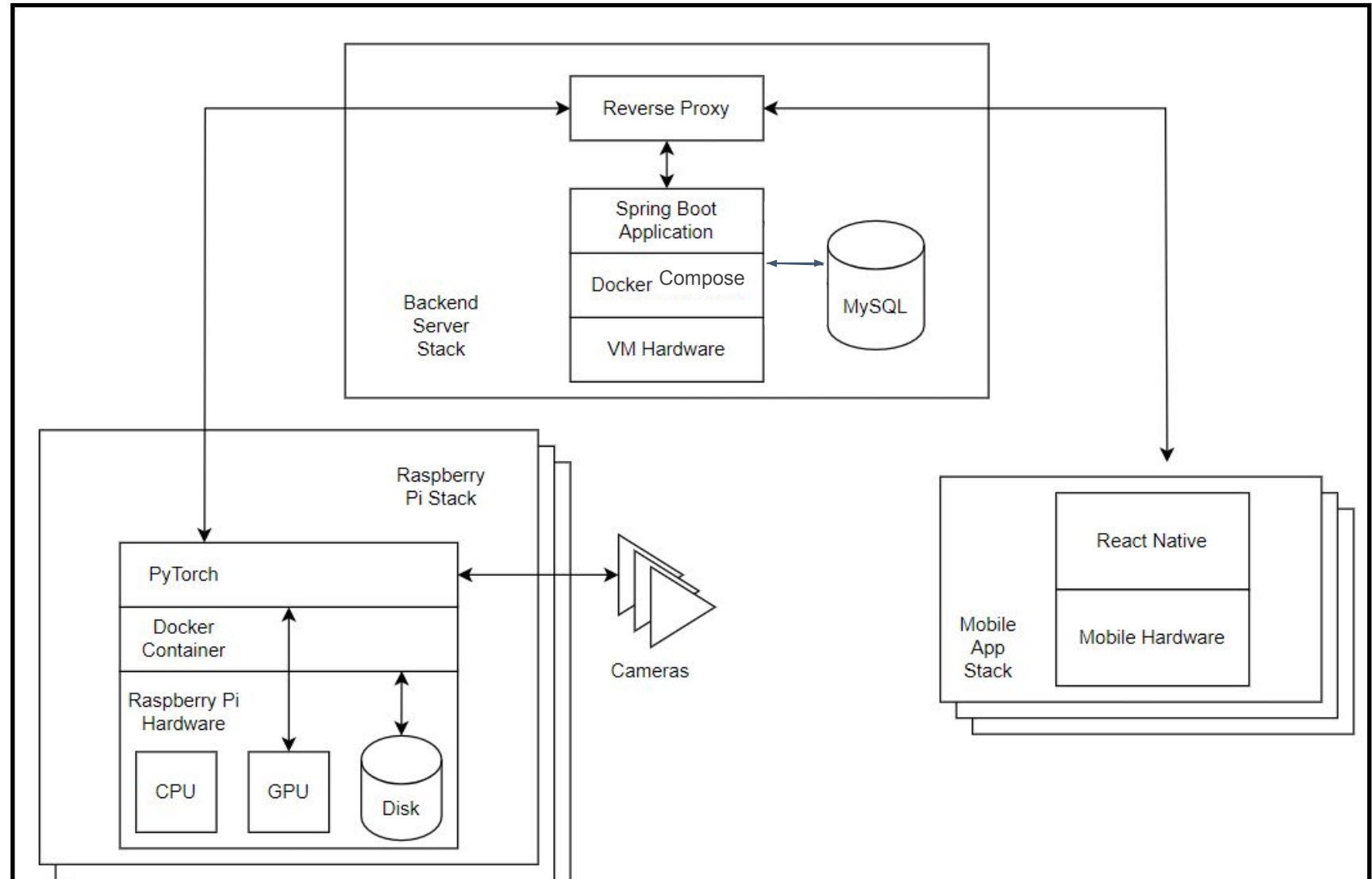
Raspberry Pi

- Pi 5 8GB
- Docker Container with Watchtower
- AI Hat+ 26 TOPS

Machine Learning

- YOLOV8





Cost

Camera Modules: \$20.07

Raspberry Pi w/ AI Hat: \$269.95

Server: Free through ISU

Price per lot: $\$269.95 + \$20.07 * \# \text{ of spots}$



Challenges



Learning New Frameworks

- Docker Compose
- React Native

Technical Issues

- ESP32 to internet
- VM access outside of internal network
- Cross-platform compatibility
- Testing back-end code
- Page Navigation Issues
- Expo Go App version compatibility

Future Plans

Admin Website

- Creating dynamic parking lots
- Assigning cameras to specific spots

Support For Special Parking

- Handicap, compact, motorcycle, etc.

Encryption

- Sensitive data

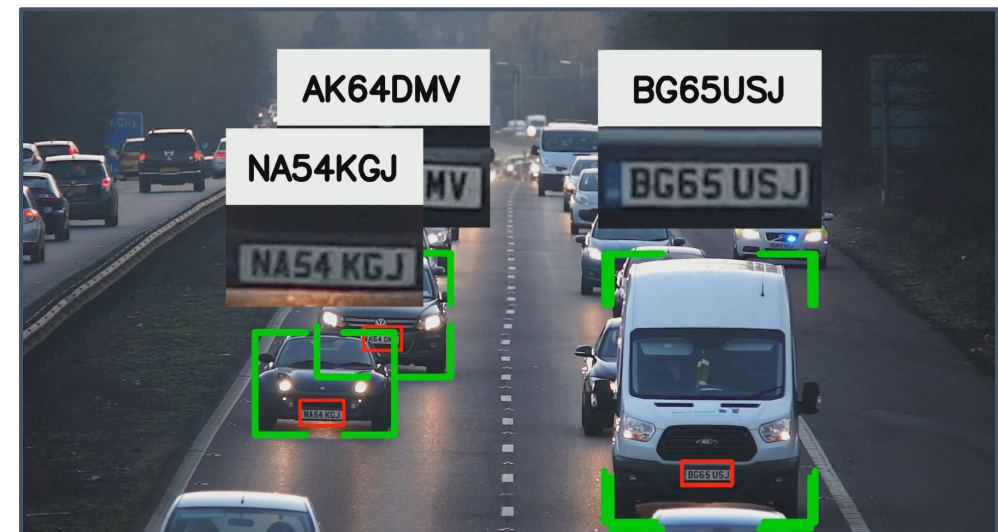
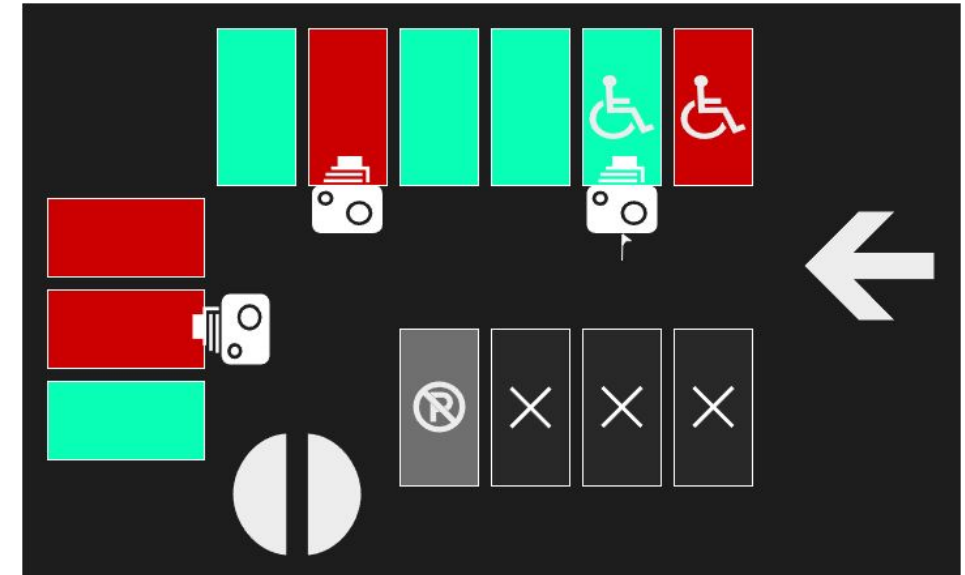
Find your Car

- Uses license plate to find car in lot

Code Coverage Report

Email Verification

Camera Container



Questions?

Testing

Unit Testing

- Postman

Integration Testing

- ESP32 camera to Pi
- Pi to server
- server to application

System Testing

- End to end run
- Car to mobile app updates

User Testing



Edge Cases

Questions:

- User parks in invalid parking spot
- User parks poorly (takes up multiple spots)
- User parks without paying
- Someone takes a user's reserved spot

Solutions:

- Contact parking division
- Notify parking division after set duration
- Contact parking division
- Contact parking division

THANK YOU

| PRESENTER NAME

| DATE DATE