IOT Project Smart Parking Navigation System - SRS

# Introduction

This project is a POC for a smart parking navigation system targeted at closed, primarily underground, parking lots where visibility is limited, orientation is challenging and parking spaces are scarce.  
By infusing current solutions with new tech, we want to enhance the ability to navigate through a closed roof parking lot by using a camera system that maps the driver’s position using AI and ML to read license plates and send new locations through azure services directly to a web application in the driver’s smartphone.

# System Overview

The smart parking system will consist of these key components:

* Module app
  + Camera sensors will be positioned strategically within the parking lot to capture license plate information and monitor vehicle location.
  + The 1st camera will be placed at the entrance of the parking lot and will initialize a unique URL for the license plate of the car at the gate.
  + QR code directed to the unique URL will be rendered on a screen accessible to the driver.
  + Other camera sensors will provide strategic capture points throughout the lot and will send updates to captured license plates matching URLs.
* Web App
  + Use a pre rendered image of the parking lot map on the navigation screen.
  + Get directions to the chosen parking section on the application via updates from the camera sensors information or use occupancy status to make the decision.
  + Cross the data between each camera in the lot, an occupancy status can be attached to the different sections of the lot and displayed on the web application.

# Functional Requirements

* Mobile\Web Application:
  + Provide a user-friendly interface displaying a map of the parking lot and its occupancy, floor levels and parking spots labeled and easy to change by controlling the UI.
  + Display real-time updates of occupancy according to the level of traffic captured by the cameras.
  + Provide navigation guidance to desired parking spaces.
  + Create an option to store your parking spot on the web app by using offline page saving/ cookies.
  + Create a search for finding your way to a specific spot.
* Module application
  + Azure connection.
  + Control QR generator and camera sensors.
* Camera Sensors:
  + Capture and recognize license plate information accurately.
  + Send license plate data to the server for processing and mapping.
* QR Code Scanner:
  + Generate QR codes at the entrance gate.
  + Initiate the mapping process by the entrance’s camera sensor.

# Non-Functional Requirements

* Performance:
  + Get close to real-time updates with minimal delay.
  + License plate recognition accuracy and efficiency.
  + Mobile\web application responsive and easy to use experience.
* Reliability:
  + Assume normal operating conditions.
  + Hardware and sensor functionality on continuous use.
  + Ability to recover from unexpected errors.
* Scalability:
  + POC will consist of 2 cameras. 1 for entrance and the other for updates and occupancy tracking. Software solutions should be scalable to real life scenarios and functionality.
  + The design should be able to make it easy to apply in different parking lots.
* Security:
  + Keep license plate information of registered vehicles hidden from other users.

# Hardware Specifications

The hardware components required for our the smart parking system POC:

* 1x Raspberry Pi 4
* 1x LED screen
* 3x Camera sensors

# Use Cases

Use Case #1: navigate the parking lot for a free space.  
Description: use the web application to get directions for the low occupancy section of the lot.  
Actors: user.  
Preconditions: user uses a vehicle and is in possession of a modern smartphone with internet connectivity.  
Main Flow:

1. Approach closely to the entrance gate for the front camera to read car’s license plate
2. Scan QR code on the screen on the driver’s side.
3. Check plate number was registered correctly
   * Select a section on the map or let the app choose a low occupancy section on its own
   * Ask for navigation
4. Follow application’s directions until arrival
   * Save parking location on the app

Postconditions: user safely arrives at location and parked his car in an available space.

Use Case #2: check for parking lot occupancy.  
Description: remotely check if there are free parking spaces in the parking lot.  
Actors: user.  
Preconditions: user has a modern smartphone with internet connectivity.  
Main Flow:

1. Find a wanted parking lot web application on the internet or through google maps.
2. Check for occupancy information on the app.

Alternative Flows:

1. Alternative flow 2: check for specific or wanted sections by navigating the map of the parking lot for more accurate and detailed occupancy information.

Postconditions: users can make a decision about arriving at the parking lot with information from the web application.

Use Case #3: locate a parking space  
Description: user saved his parking space number and wants to be able to navigate himself to it  
Actors: user  
Preconditions: user has a modern smartphone with internet connectivity.  
Main Flow:

1. Open the URL provided when entered the parking lot
2. Click on “החנייה שלי”
3. The parking space will be shown on the map for the user to guide himself to

Postconditions: user can locate a specific parking spot inside the lot.

Use Case #4: manage parking lot  
Description: ability to get information about current and past occupancy, daily drive throughs and parking space distribution of the parking lot.  
Actors: administrator  
Preconditions: secure access to IOT device functionality and software.  
Main Flow:

1. Admin enters to the azure page of the lot
2. Admin can choose to view current information or observe the data through the past timeline which azure is enabling for him to see

Postconditions: administrator can gather the information he needs on the parking lot.

# Proof of Concept

* Allocate an environment to simulate a parking lot drive space and position our modules in a manner which allows for a strategic mapping for the underground navigation system.

# Integration Options

* Maps (e.g Google Maps, Waze links)
* Payment options (e.g Pango)