



**APP**arking Lot

# Our Team



**Tamir Reznik**

308346170

Team Leader, Prod Owner



**Yonatan Blum**

308131333

DBA



**Anna Meleshko**

319346045

Tech Writer



**Sapir Madmon**

209010230

QA engineer , UI



**Avichai Eliyahu**

204309025

Dev Ops, Scrum master



# Introduction

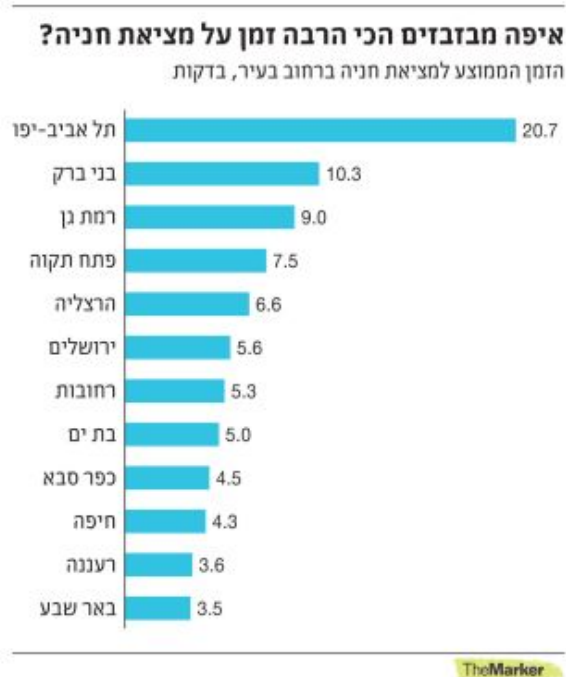
**APParking Lot** is a social application that allows users to share data about parking spaces. The vision of the system is:

- Making parking spots easier to locate
- Shorten the searching time for parking
- Shared experience

# Motivation

## Why did we choose to make this project?

Many of us know the situation when we are looking for parking... So we got an idea to have a social application which decreases the searching time via sharing data about free or occupied parking lots.



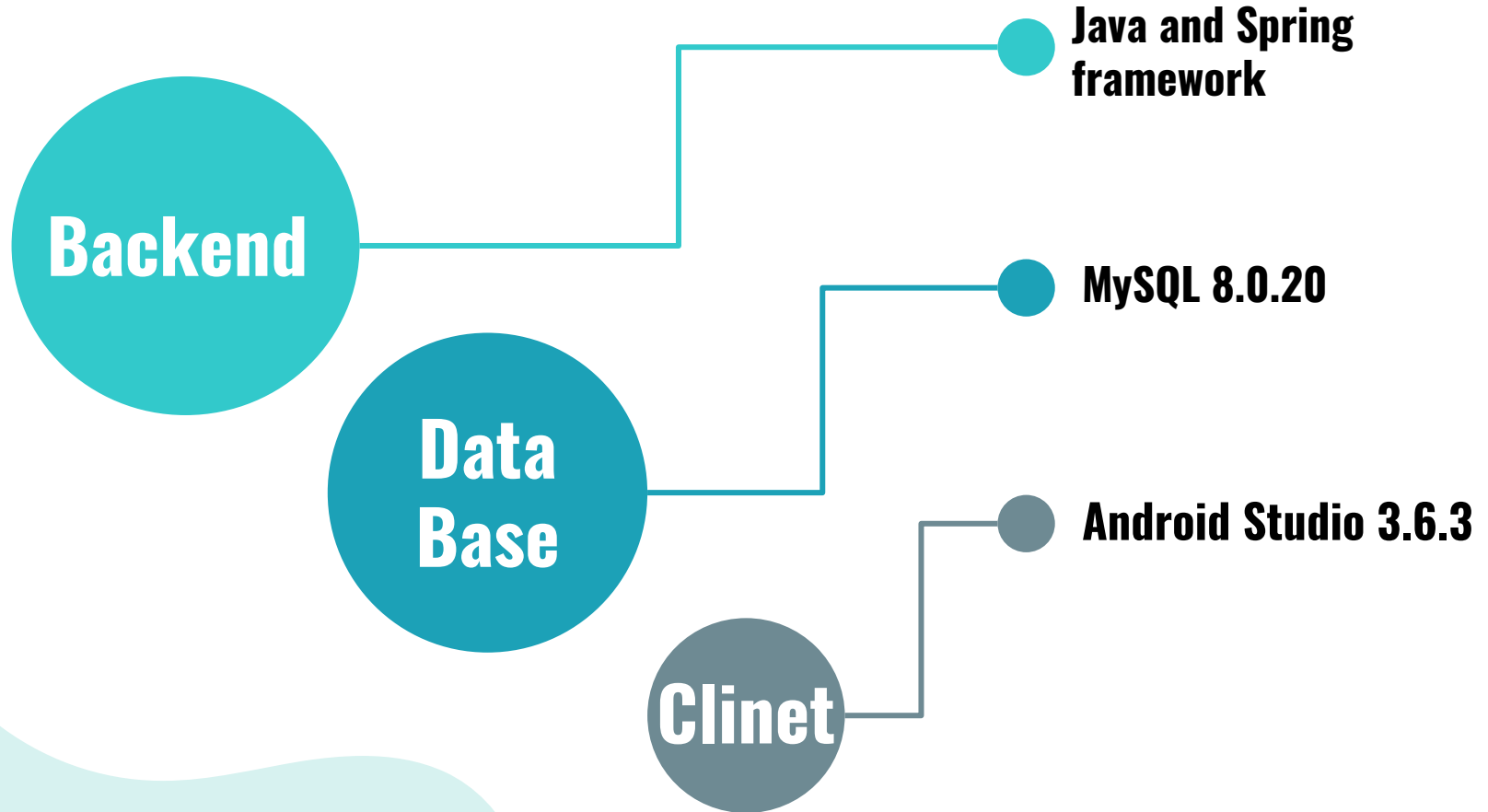


## **Purpose Of The System**

The purpose of the system is to help users find parking spaces in parking lots or on the street faster and easier.

The system aims for that goal using shared experience: users can report taken or free parking spaces.

# Main Tools



# Players In The System

## Driver - Player

Representation of a user who releases /catch/searches for a parking lot/parking space.

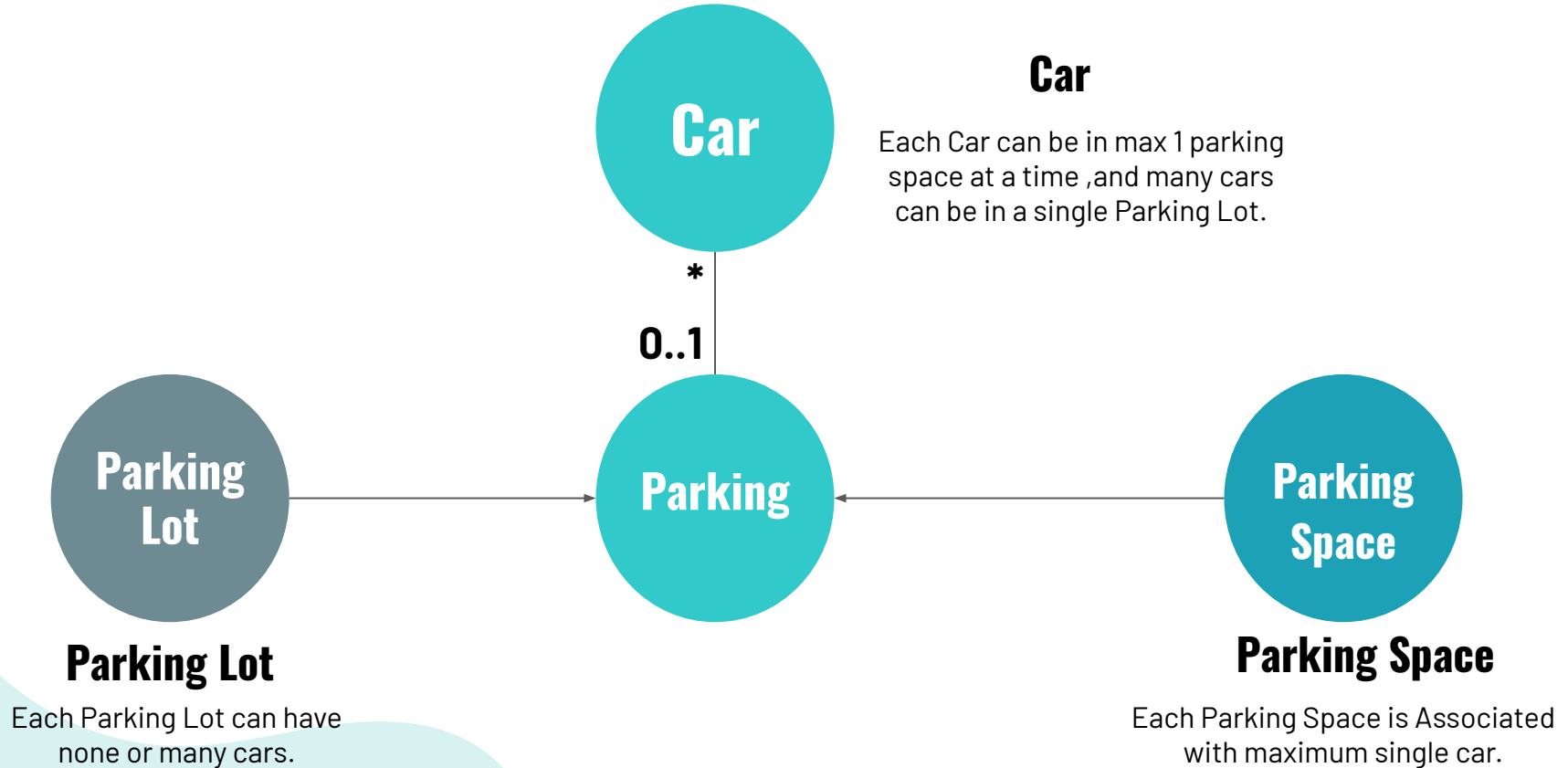
## Manager

Representation of user that can create parking lot/update parking lot or parking space/delete parking.

## Admin

- Can delete all users/ actions/elements
- Can export all actions/users in the system.

# The Elements In The System



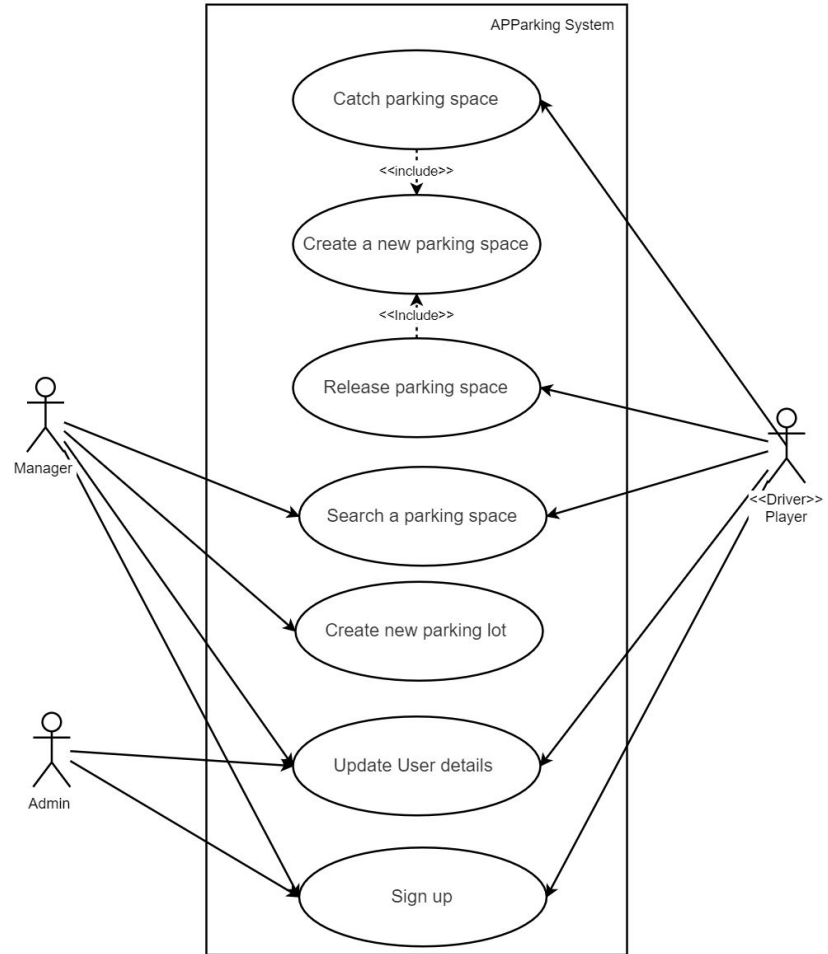


# Functional Requirements

## Use Cases



# Use Case Diagram



# Login User



## Open The App

The user opens the app and opens a main window.

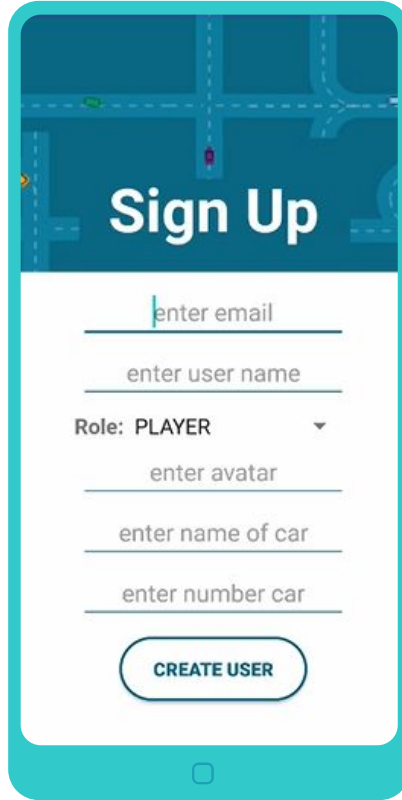
## Fill In User Information

Fill in domain and email details.

## Login Button

Clicking the login button and The system verifies the details .

# Sign Up



**Sign Up**

enter email

enter user name

Role: PLAYER ▼

enter avatar

enter name of car

enter number car

**CREATE USER**

## **Sign Up Window**

The system displays a registration window

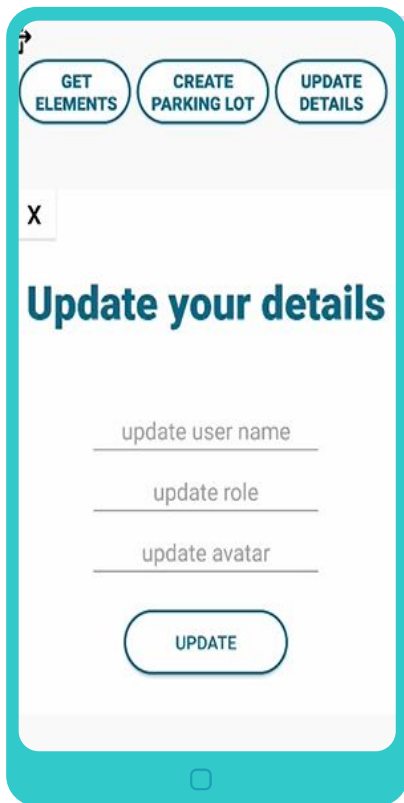
## **Fill In The Details**

The user fills in his car details, and his private details and choose a Role (Player ,Manager or Admin).

## **‘Create User’ Button**

The user is registered on the system.

# Update User Details



A mobile application interface for updating user details. The screen has a teal border. At the top, there are three buttons: 'GET ELEMENTS', 'CREATE PARKING LOT', and 'UPDATE DETAILS'. Below these is a close button 'X' in a small box. The main heading is 'Update your details' in teal. There are three input fields with placeholder text: 'update user name', 'update role', and 'update avatar'. At the bottom is a large 'UPDATE' button. The bottom of the screen shows a teal home indicator bar.

## Open The App

The user logs in with his details to the application.

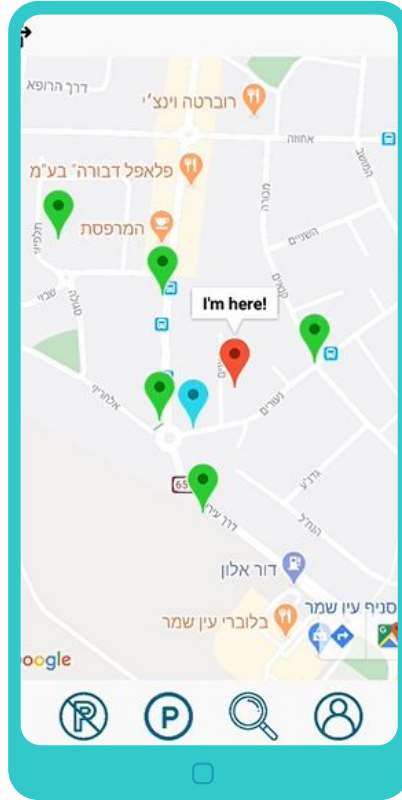
## Update User Details

The user fills in the details he wants to update.

## Update Button

Clicking on update button and the system updates the details in the system and pops a success message.

# Find A Parking Space



## Arrive to destination

The driver looking for a parking space.

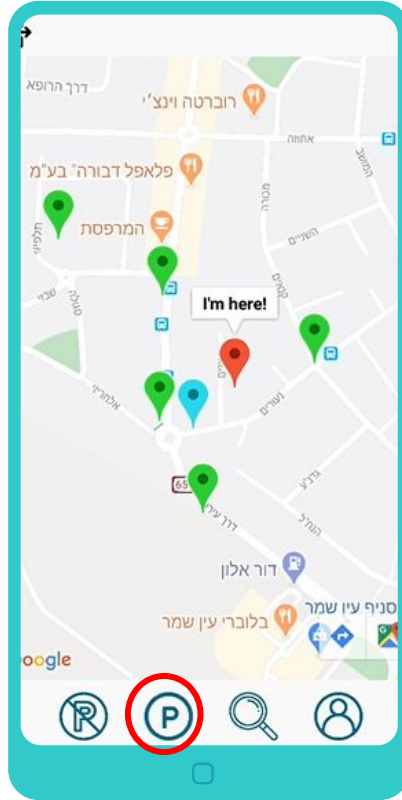
## Open the App

checking if there were some reports near his location.

## Report about empty locations

The system locates empty parking locations near the user and reports about this.

# Catch Parking Space



## Park in parking space

The driver parks in parking space.

## Report - Catch

The driver indicates in the app that he catch the parking space

## Update

The system updates the new parking status.

# Release Parking Space



## **'Depart' Button**

The driver click on the 'Depart' button.

## **Identify Parking Space**

The system identifies the parking space in the DB.

## **Release parking space**

The system updates the new parking status to active mode.



# Create Parking Lot



The image shows a mobile application interface for creating a parking lot. At the top, there are three buttons: 'GET ELEMENTS', 'CREATE PARKING LOT', and 'UPDATE DETAILS'. Below these buttons is a search bar with an 'X' icon. The main title 'Create Parking-Lot' is displayed in a large, bold font. Below the title, there are four input fields: 'name of parking-lot', 'Location:', 'Latitude', and 'Longitude'. The 'Location:' field is followed by a 'Parking lot capacity' field. At the bottom of the form is a 'CREATE' button. The entire interface is enclosed in a light blue border.

## Open The App

The manager logs in with his details to the application.

## Create Parking Lot

The manager fills in the details of the parking lot he want to add.

## Create Button

Clicking on create button and the system updates the DB, and pops a success message.

# Non - Functional Requirements

1. Usability
2. Supportability



# 1. Usability

## 1.1

Works on an android  
6.0.1 and above.

# 2. Supportability

## 2.1

Document the actions on the  
system using a log file.

## 2.2

Automated tests (JUnit)

## 2.3

The implementation of the  
system is done according to  
the layer model architecture

# Technologies



**THANKS  
FOR  
LISTENING!**

