

P A R K I F Y

# PARKIFY



A SMART PARKING SOLUTION



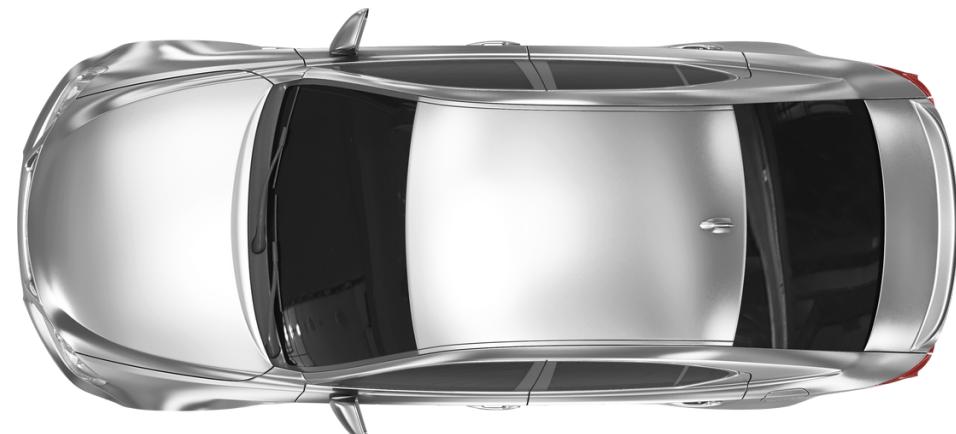
Ollscoil  
Teicneolaíochta  
an Atlantaigh

Atlantic  
Technological  
University

**TAN MIN HAN**

G00377211

BEng (Hons) Software and Electronic  
Engineering



# Agenda



Introduction to Parkify

02

Architecture Diagram

03

Project Timeline

04

Technologies behind Parkify

05

Challenges

06

Results

07

Conclusions

07

01

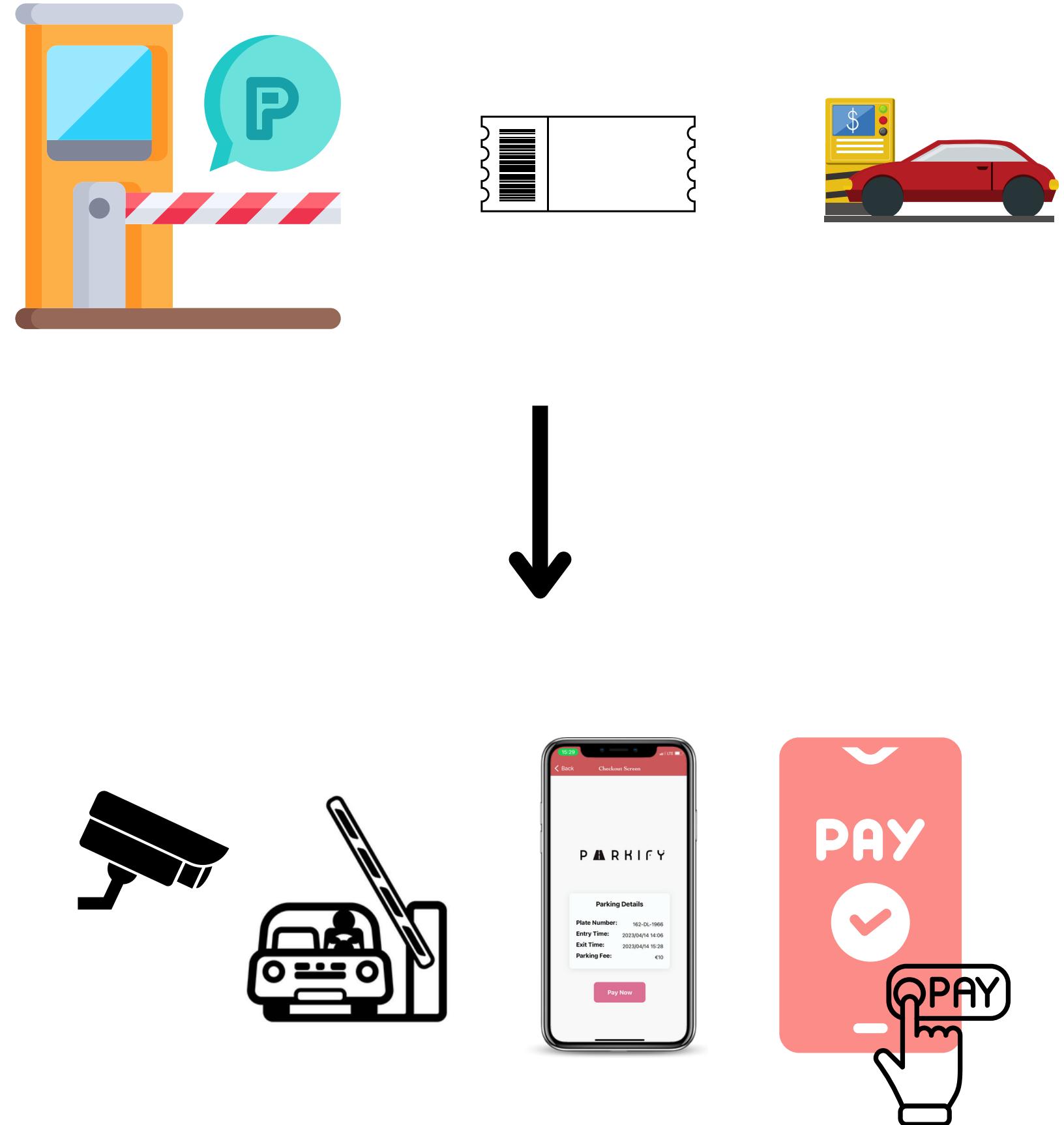
# Introduction

## Goal

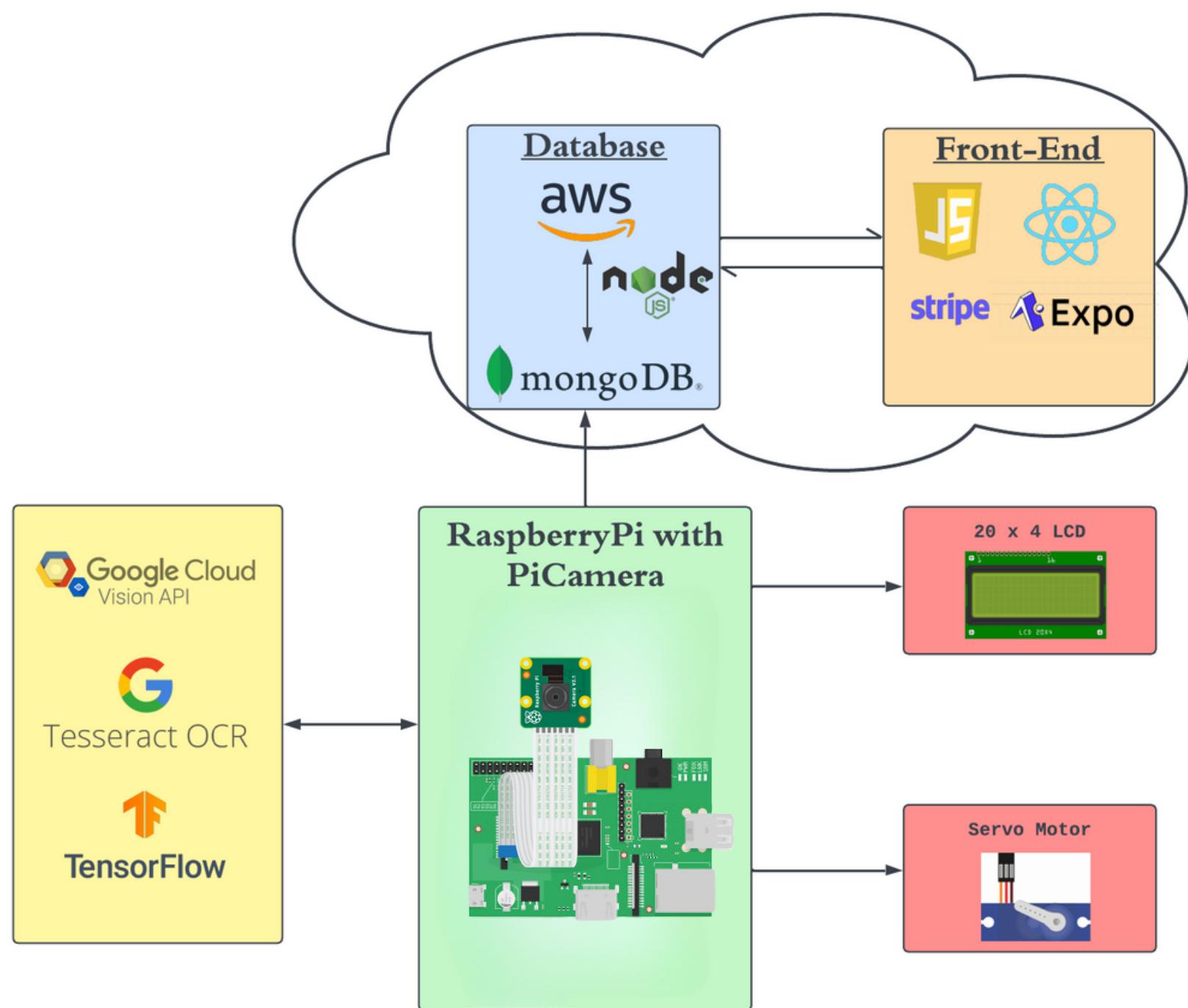
The goal of Parkify is to provide real-time information about parking availability and automate the process of parking.

## What is Parkify?

- A system for the parking lots integrated with automated license plate recognition technology, and mobile application.
- It uses a combination of object detection and OCR to detect and read license plates accurately
- Provides real-time parking data
- The system can be integrated with existing parking management systems to improve efficiency.



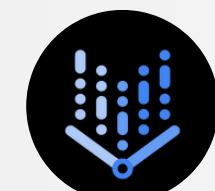
# Architecture Diagram



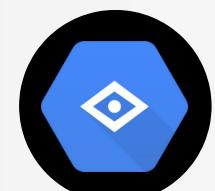
## Technologies Integration

### MACHINE LEARNING

GOOGLE CLOUD  
VERTEX AI



GOOGLE CLOUD  
VISION



TESSERACT OCR

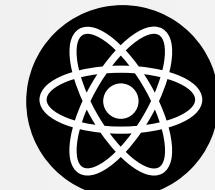


TENSORFLOW

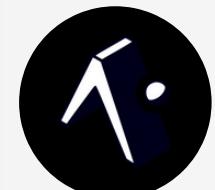


### FRONT END

REACT NATIVE



EXPO



JAVASCRIPT



### BACK END

MONGODB ATLAS



NODEJS



STRIPE



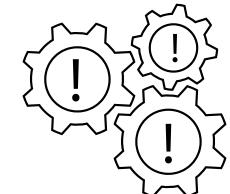


# Challenges with Parkify

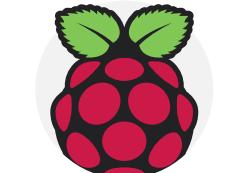
Training and Optimising Model



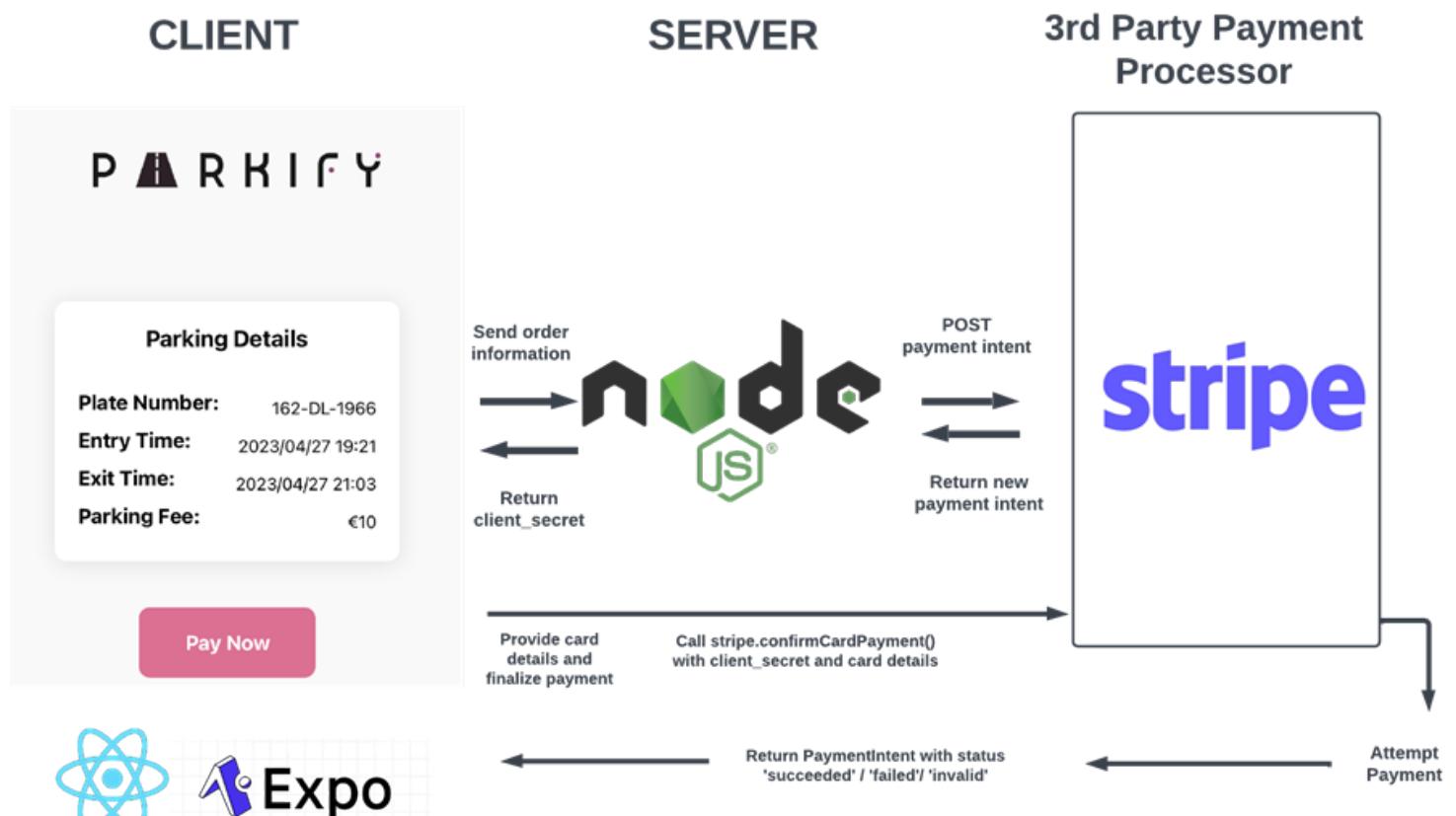
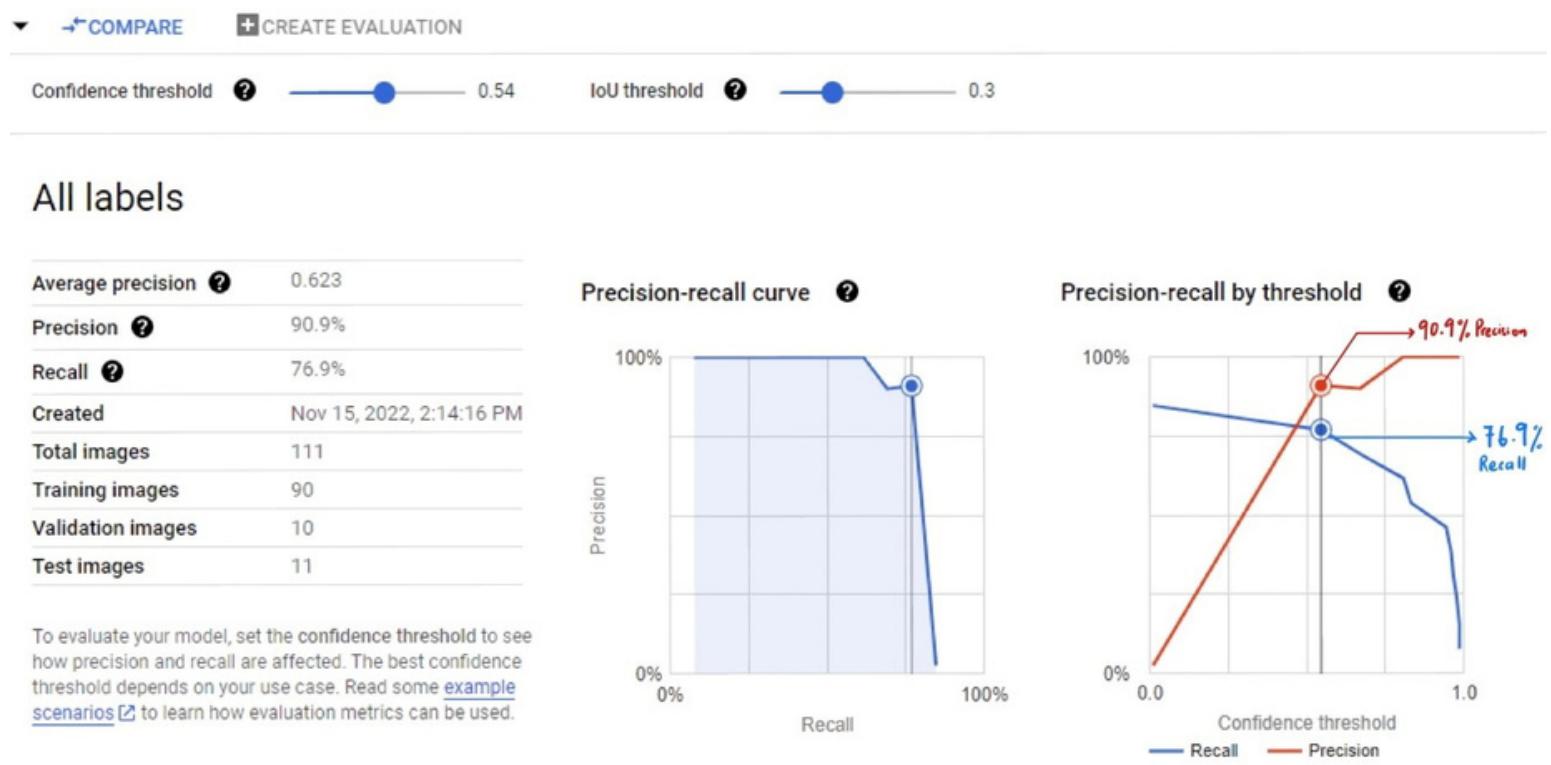
Testing and debugging



Implementing ALPR and Raspberry Pi



Security features



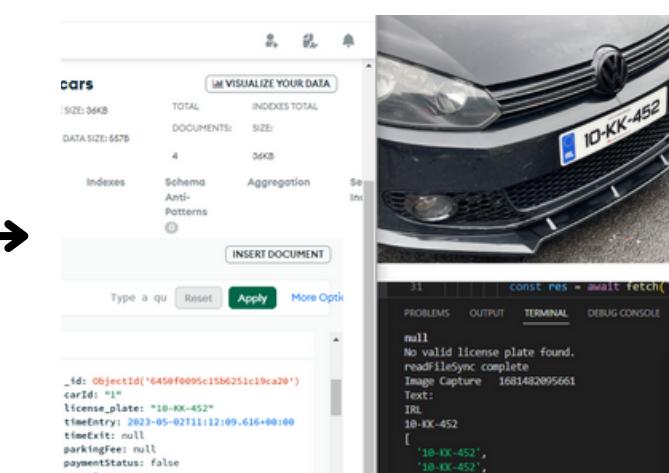
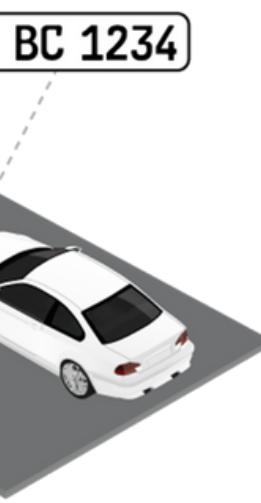
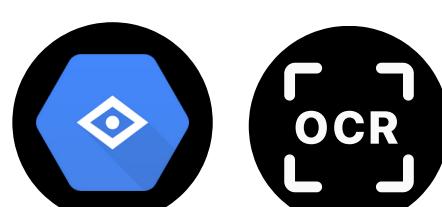
# How Parkify works?



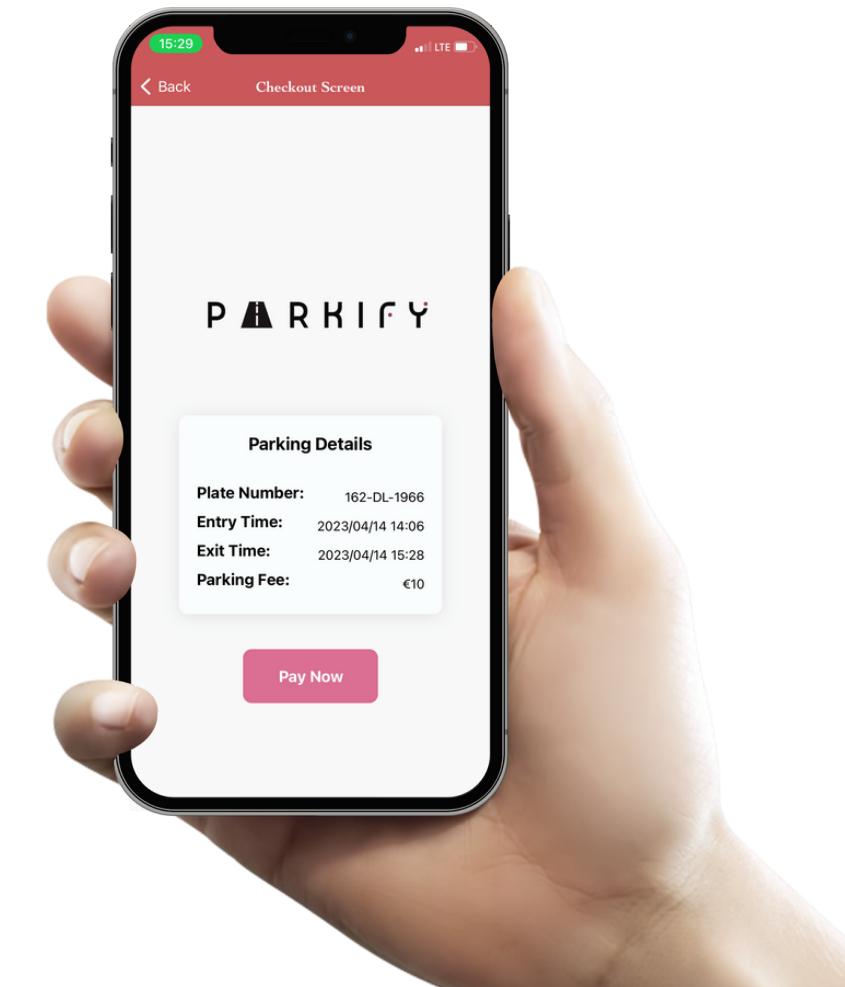
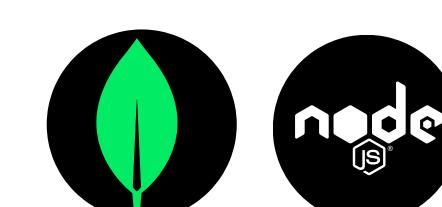
Raspberry Pi integrated with the TensorFlow model to detect the license plate using a trained model.



Google enhanced Tesseract OCR will extract the characters and numbers from the license plate



Save the license plate characters as text or image into database

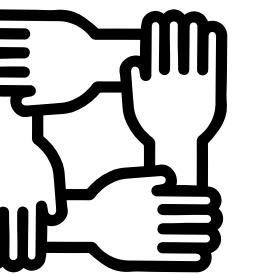


Pay the parking fee with the mobile application



# TEAMWORK

- Stand-up meetings (Weekly Updates)
- Teams Call
- Report/Poster/Video Review
- Jamming Session with Corey



## Stand-up Meeting

Myself:

- Continue working on the project for next part
- Did some research during holiday about the database for SQL, MongoDB
- Created a project log ([kanban board](#)) to track my progression on GitHub
- I also create a Gantt Chart to track my progression
- I did show my teammates my KanBan board and Gantt Chart

Cian:

- After speaking with Corey, he looked into using [HiveMQ](#) as an [mqtt broker](#) as a way to publish his card readings
- He looking into using google cloud for his project.
- He setup database instance in google cloud and connected it to his backend.

Conor:

- He is looking into database stuff and doing research about database
- I told him that I did some research as well so we can look into together

Corey:

## Teamwork with Niall

Talked with Niall McLoughlin about our project since we got very similar parts for our project and we discussed about the PiCamera shutter and gain how we need to adjust it for a clearer image when PiCamera captured in class. We also both discussed together to help each other since we got similar parts and talked about together since we start this project our own.

- We both will checking our progression to monitor our project and help each other with the parts we used
- Niall might using object detection some sort of technology to detect QR Code and I did the object detection research and also similar enough with his project so I might able to help me with some guide for that.
- I also find out some interesting article that related to his project and hopefully will help him out with the project he working on
  - <https://circuitdigest.com/microcontroller-projects/qr-code-scanner-using-raspberry-pi-and-opencv>

## Teamwork with Conor

Conor and myself talked to each other and decide to look into the databases we should use each other and which is more better and suits our own project scenarios.

Some progression we did:

- Researched the difference between MongoDB vs SQL Databases.

```
File Selection View Go Run Terminal Help
Payments - Fetchlocalhost - Visual Studio Code
EXPLORER 1 unsaved
JS App.js JS Payments.js ● app.json package.json JS List.js
components > JS Payments.js > JS Payment > JS payApi
1 import { View, Text, TouchableOpacity, Alert } from 'react-native'
2 import React from 'react'
3 import {useStripe} from "@stripe/stripe-react-native"
4
5 const Payment = () => {
6   const stripe = useStripe();
7
8   const payApi = async() =>{
9     try {
10       const res = await fetch('https://1f83-77-75-244-145.eu.ngrok.io/pay', {
11         method: "POST",
12         headers: {
13           "Content-Type": "application/json",
14           "ngrok-skip-browser-warning": "69428",
15         },
16         body: JSON.stringify({
17           parkingFee: 1000, //10euro in unit of cents
18           paymentStatus : boolean? status//ADDING STATUS TO CHECK IT PAY OR NOT
19         }),
20       });
21
22       const data = await res.json();
23       console.log(data);
24       console.log("Button Pressed");
25
26       if(!res.ok) return Alert.alert(data.message);
27       const clientSecret = data.clientSecret;
28       const initSheet = await stripe.initPaymentSheet({
29         paymentIntentClientSecret: clientSecret
30       })
31       if(initSheet.error) return Alert.alert(initSheet.error.message)
32       const presentSheet = await stripe.presentPaymentSheet({
33         clientSecret
34       })
35       if(presentSheet.error) return Alert.alert(presentSheet.error.message)
36       Alert.alert('Payment complete, thank you!');
37     } catch (err) {
38       console.log(err);
39     }
40   }
41
42   const data = await res.json();
43   console.log(data);
44   console.log("Button Pressed");
45
46   if(!res.ok) return Alert.alert(data.message);
47   const clientSecret = data.clientSecret;
48   const initSheet = await stripe.initPaymentSheet({
49     paymentIntentClientSecret: clientSecret
50   })
51   if(initSheet.error) return Alert.alert(initSheet.error.message)
52   const presentSheet = await stripe.presentPaymentSheet({
53     clientSecret
54   })
55   if(presentSheet.error) return Alert.alert(presentSheet.error.message)
56   Alert.alert('Payment complete, thank you!');
57 } catch (err) {
58   console.log(err);
59 }
```

PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

L00 {"clientSecret": "p1 3MpKdIWfjneG7h8lav2Au secret JKY8bG9uFxEuTG9yC5vdBftj2", "message": "Payment initiated"}

COREY EGAN - STUDENT 17/04 12:06 pm Edited  
Going to run the Jam Session in the next class here's the doc of steps and apk to install  
(Will have to allow install from unknown sources in phone settings) Student - FYP Niall Team

Parkere.apk Student-FYPNiallTeam >General

Jam Session Park'ere.docx Student-FYPNiallTeam >General

CE

CF error on phone rotation, once orientation of the phone is changed app restarts

CD Park'ere jam session results: Demo was successful, only thing to note was that there is a delay to close p

MT JAM SESSION:  
There is no clear instructions with how to select the parking location on the map to open the parking space interface. Demo is good, functionality is fine and smooth

CE Thanks guys, I'm going to try fix these issues

Reply



*Thank  
You*

Feel free to ask any questions!