



# SIMPLIPARK

17.01.2023

SCSE TechFest 2023 Hackathon

SimpliPark Link: <https://weikangg.github.io/SimpliPark/>

## Team 51

	Name	Email	Telegram Handle
1.	Sua Qi Rong	QSUA001@e.ntu.edu.sg	@QiRonggg
2.	Tan Yue Yang Kane	Ktan201@e.ntu.edu.sg	@kanecocaine
3.	Chong Wei Kang	wchong036@e.ntu.edu.sg	@weikangg
4.	Fong Ye Xuan	yfong007@e.ntu.edu.sg	@kenfyxx

## Overview

Our team aims to develop a web application (Simplipark) targeted towards the general public in improving their experience at finding available carpark lots. Simplipark utilizes computer vision in identifying empty car park lots and outputs the real time information on the website for the general public. This allows each and every individual to be able to leverage on advanced technology to not only check on car park vacancies but also to have a stronger sense of ownership for the well-being of their estate.

## Identified Problem

HDB manages over 2000 car parks, with more than 660k car parks and 7000 equipment that supports the operations of the car park. Our car parks are visited by 500k unique motorists and the system processes 2.5m transactions daily. It is not possible for HDB to deploy staff/maintenance teams to every car park, due to the sheer number of car parks and equipment. Also, the situation is exacerbated by the manpower crunch brought about by our aging population.

For drivers, they face many problems finding empty parking spaces with drivers needing to drive around aimlessly in hopes of finding a carpark with empty lots.

## Inspiration

We found that there is an underlying problem where individuals always have difficulty of finding an empty parking lot and we would like to save the amount of time that individuals spend looking for one.

## Solution

Therefore, our group seeks to allow the residents to rely on SimpliPark to more easily find car parks with empty lots. Additionally, residents are also allowed to take ownership of their estate's well-being by allowing them to report any problems within their neighborhood swiftly and easily with the click of a button.

## Features

### 1. Display empty and filled carpark lots

SimpliPark allows users to choose their desired car park, or to detect the car parks nearest to them using their location. Then, a live feed of the car park is displayed, with the number of empty lots and the capacity of the car park. This allows users to make better decisions as to which car park they should go, saving time and fuel.

### 2. User Reporting System

Through observing the live feed, users are able to report problems for the specific selected car parks. This can facilitate timely maintenance of car parks without requiring heavy deployment of staff/maintenance teams, encouraging residents to take ownership of their estate's well-being.

### 3. Leaderboard

Leaderboard is created to exalt local heroes in our community who actively report damages rather than let it go unnoticed. This will serve to encourage other users to report damages such as broken gantry or damaged camera and whatnot.

## Specifications

### 1. Computer Vision (CV)

Usage of Python open source CV libraries to detect empty parking lots in car parks. This image will also be shown on the SimpliPark as a live feed for users to monitor the current situation in their car park.

### 2. Front-end Web Development

HTML, BootStrap, CSS, JavaScript to build the SimpliPark and display all car parks around the area.

## Limitations

### 1. Night Model

The program written for the computer vision algorithm is calibrated for day time usage by installing image sensors in car parks. If the brightness level at night is too dark, infrared sensors may be required to augment the computer vision system to continue ensuring high accuracy for the information provided on the website.

### 2. Multi-storeyed structure of HDB car parks

Our prototype is created using an overhead view of open-air car parks, where one image is able to capture hundreds of parking lots. In HDB carparks where they are usually multi-storeyed, it will be difficult to capture many parking lots in one image. Therefore, many cameras would be required, increasing cost. Furthermore, allocating parking slots to the computer vision algorithm will be harder as the slots will not be as clearly defined as those seen through a bird's eye view.

## Future Plans

### 1. Mobile Version (SimpliPark Mobile)

We intend to build SimpliPark Mobile that leverages on our web application for ease of use by drivers on the road.

### 2. Reward System

In order to further encourage residents to utilize SimpliPark, we plan to collaborate with HDB and other companies to leverage a reward system that will incentivise users to use the app and actively take part in the community. Rewards can include free parking vouchers after clocking a certain number of hours with the app or reporting a certain number of damages.

## What we learnt

We picked up some useful technology that are not usually taught in school yet at our Year 2 Level such as image processing and Computer Vision techniques through youtube tutorials



and also CSS frameworks such as Bootstrap to design websites and make our frontend more beautiful.

## How we built our project

We first did the Python code for our computer vision to identify empty parking lots and screen recorded those videos to show the feed of the carpark on our website. We then went on to brainstorm about the design of our website before working on the implementation of it.

## Challenges

Since most of the techniques were mostly not taught in school by the professors, we had to quickly adapt and learn as much as we could from online sources such as Stack Overflow, the official documentations, and Youtube itself.