



Team/ Instructor VLOG #3

Team Apollo - Dec 4th

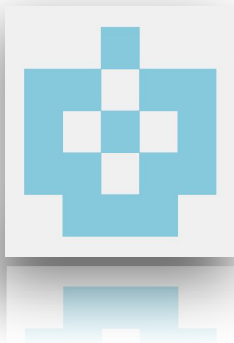
Kecheng Yu

Yilin Ren

Zhuo Zhou

Ziwen Tan

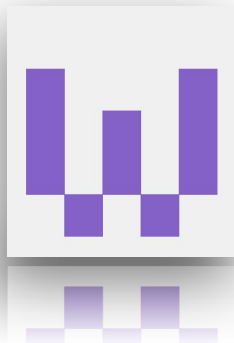
Team member introductions



Kecheng Yu
AI vehicle counter
algorithm



Zhuo Chen
AI vehicle counter
algorithm



Ziwen Tan
backend
development



Yilin Ren
Front-end developers
Team Scrum management



Project Activities

Meet with mentor at Oct 13th: Kecheng, Ziwen and Zhuo (discussed about assumption)

Group meeting at Oct 14th: Yilin (Frontend) Ziwen(Backend) Kecheng (research for AI alternative)
Zhuo(research for hardware alternative)

Group meeting at Oct 21st: All member attended (made decision)

➡ Met with mentor at Nov 24th: Kecheng, Ziwen, Zhuo and Yilin (discussed about vehicle detection)



Status description

 Green



85%

Milestone 1.5 in progress:

Done

- System design
- Parking lot management frontend
- Parking lot management backend
- Project roadmap
- RACI chart

In Process

- Parking lot vehicle detection feasibility test



Project issues

Met with mentor and discovered our AI vehicle detection.

YOLO may quired a large demand of computing power on a single board computer.

Working on minimum processor requirements for YOLO



AI - YOLO investigation

YOLOv5: No specific minimum hardware requirements

Plans:

- Limit the features (only accounting vehicle numbers is needed)
- Downgrade from v5 to a lightweight version: v3 or v2
- Back up plan: VGG16



Project changes

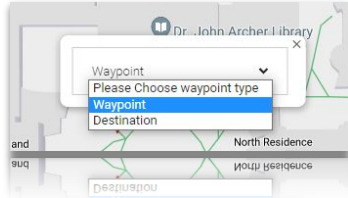
Current solution: YOLOv5 with limit features

Potential changes: Downgrading, VGG 16

Project/technology update

Frontend

- Design and implemented new data structure to store waypoints.
- Implemented a dialog to require the user to set the waypoint properties each time the waypoint is created.
- Preview



Backend

- Create pointWay
- Delete pointWay
- Update pointway
- Authorization

Backend demo

The screenshot displays a REST client interface with the following components:

- Request Bar:** Method: GET, URL: http://localhost:8083/, Send button.
- Headers Tab:** Contains 7 headers:

Key	Value	Description
User-Agent	PostmanRuntime/7.29.2	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
token	eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJmZDZj...	
Key	Value	Description
- Body Tab:** Shows the response in Pretty format: 1 "this is a test api".
- Status Bar:** 200 OK, 138 ms, 441 B, Save Response button.

Backend demo

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8083/
- Buttons:** Send, Cookies
- Tabs:** Params, Authorization, Headers (7), Body, Pre-request Script, Tests, Settings
- Headers:**
 - ☒ User-Agent: PostmanRuntime/7.29.2
 - ☒ Accept: */*
 - ☒ Accept-Encoding: gzip, deflate, br
 - ☒ Connection: keep-alive
 - ☐ token: eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJmZD...
- Body:** Key, Value, Description
- Response:** 403 Forbidden, 8 ms, 522 B, Save Response
- Response Body (JSON):**

```
1 {
2   "timestamp": "2022-12-04 13:11:27",
3   "status": 403,
4   "error": "Forbidden",
5   "path": "/"
6 }
```



Next up

We will:

- Accomplish milestone 1.5 before the end of December.
- Start working on milestone 2.0
- Front-end interface optimization and back-end structure optimization
- YOLOv5 deploy



Team reflection/retrospective

In good status

- Everything is working fine, and the frontend is pretty much done at this stage.
- The backend is looking for ways to implement an A* algorithm.
- The AI part now needs to consider whether the performance of the application can be supported with the equipment we currently have, and how to reduce the performance requirements.