

## **FINAL REPORT**

ELEC 392: Principles of Design and Development

Aylesbury 12

Aylesbury Automotive

Connor Wilkin - 20385961

Julien Chagnon - 20390465

Tate Vickery – 20387871

2026-04-06

Prepared for Prosper Allo

## Statement of Originality

We do hereby verify that this machine problem submission is my own work and contains our own original ideas, concepts, and designs. No portion of this report or code has been copied in whole or in part from another source, with the possible exception of properly referenced material

## Executive Summary

This project focused on the design, implementation, and evaluation of an autonomous taxi capable of operating within the Vehicle Positioning and Fare System (VPFS) simulation environment. The primary goal was to develop a reliable vehicle controller that could integrate navigation, control, and system communication to successfully complete taxi fares under realistic constraints such as noisy position data, limited update frequency, and strict task completion requirements. The system was designed around a deterministic, sensor-driven approach prioritizing reliability and repeatability. A PID-based line-following controller using grayscale sensors formed the foundation of vehicle motion, while a graph-based navigation system enabled path planning across the full competition map. Integration with the VPFS allowed the vehicle to receive fare assignments, track its position, and determine proximity to pickup and drop-off locations. Design decisions emphasized stability, fail-safe behavior, and strict adherence to system constraints, including accurate detection of “in-position” conditions.

A key engineering tradeoff involved abandoning camera-based object detection in favor of a purely grayscale sensor approach. Although object detection was initially developed, it proved unreliable under varying lighting conditions and introduced unnecessary complexity. The final system instead focused on deterministic sensing and control, which significantly improved consistency and overall performance. System performance was evaluated based on reliability in line following, turning accuracy, and successful fare completion. The vehicle achieved a line-following success rate exceeding 95% and a turning success rate of approximately 80%, meeting or exceeding design targets. These results translated into strong competition performance, with the team achieving first place in reputation, first overall competition score, and strong results in additional categories.

Despite these successes, several limitations were identified. The system remained sensitive to lighting conditions, lacked robustness in recovering from navigation errors, and did not incorporate full environmental awareness due to the absence of a functional vision system. These limitations highlight key challenges in developing fully autonomous systems and inform future improvements, particularly in perception and adaptive navigation.

Overall, the project successfully demonstrated the integration of autonomous vehicle control with a centralized fleet management system. By prioritizing reliability and system-level coordination, the final design achieved consistent and effective performance within the competitive environment, while also illustrating the challenges that must be addressed for real-world deployment.

# Table of Contents

Statement of Originality.....	ii
Executive Summary (Connor) .....	ii
Table of Contents.....	<b>Error! Bookmark not defined.</b>
1. Introduction .....	1
1.1 Background .....	1
1.2 Problem Statement and Motivation .....	1
1.3 Structure of the Report .....	2
2. Design Criteria and Specifications .....	2
2.1 System Requirements and Constraints .....	2
2.2 Objectives and Success Criteria.....	3
2.3 Scope of the Project.....	3
3. Ethical, Social, and Environmental Considerations .....	4
3.1 Ethical Framework in Design .....	4
3.2 Safety, Fairness, Accessibility, and Sustainability .....	5
3.3 Societal Implications .....	5
4. Design Methodology and System Architecture (Julian) .....	6
4.1 Engineering Design Process .....	6
4.2 Alternatives and Tradeoffs .....	6
4.3 System Architecture .....	6
4.3 Technologies and Tools .....	7
4.4 Algorithms.....	8
4.5 Training .....	9
5. Implementation and Integration (Julian) .....	9
5.1 Description of System Components .....	9
5.2 Integration Strategy and Challenges .....	10
5.3 Implementation Artifacts .....	11
5.4 Project Development and Milestones .....	11
5.5 Development Timeline .....	11
6. Verification, Validation, and Performance Evaluation (Tate) .....	12
7. Limitations, Risks, and Recommendations (Tate) .....	14
8. Conclusions (Tate) .....	16
References .....	17
Appendix A: Team collaboration and professional practice (Tate) .....	18
Appendix B: Individual Reflection on Skills and Performance (All) .....	19

# 1. Introduction

## 1.1 Background

Autonomous driving is a fast-advancing field that integrates robotics, artificial intelligence, computer vision, and control systems to enable vehicles to operate with minimal or no human intervention. In recent years, significant progress has been made in areas such as real-time perception, path planning, and vehicle control which is bringing autonomous systems closer to widespread deployment. These advancements have been driven by the need for safer, more efficient, and more accessible transportation solutions.

One of the most promising applications of this technology is autonomous taxi systems in urban environments. These systems look to provide on-demand transportation services that reduce the reliance on human drivers, lowers operational costs, and improves traffic efficiency. However, reaching this goal requires not only reliable vehicle autonomy but also seamless integration with higher-level systems that manage fleet coordination, task allocation, and real-time monitoring.

The Vehicle Positioning and Fare System (VPFS) used in this project simulates these real-world challenges by providing a system for assigning fares, tracking vehicle positions, and evaluating performance. This creates a realistic environment in which autonomous control algorithms must operate under constraints such as noisy position data, communication delays, and strict task completion requirements.

## 1.2 Problem Statement and Motivation

The objective of this project is to develop and integrate an autonomous vehicle controller capable of interacting with a fleet management system to successfully complete assigned taxi fares. While individual components such as motion control and path planning are well-studied, a key challenge lies in combining these elements with system communication to create a cohesive and reliable solution.

This project addresses the need for:

- Reliable interpretation of fare assignments and system state from the VPFS
- Accurate navigation to pickup and drop-off locations using planned paths
- Clean handling of position uncertainty and update latency
- Compliance with system constraints such as stationary “in-position” requirements

The motivation behind this work is to bridge the gap between low-level vehicle control and system-level integration. Demonstrating this capability is essential for real-world

autonomous taxi systems, where vehicles must operate as part of a coordinated fleet rather than as isolated units.

## 1.3 Structure of the Report

The remainder of this report is organized as follows:

Section 2: Design Criteria and Specifications: outlines the functional and performance requirements that guided the system design.

Section 3: Ethical, Social, and Environmental Considerations: discusses the broader impacts of autonomous taxi systems, including safety, accessibility, and sustainability.

Section 4: Design Methodology and System Architecture: Describes the overall system design, including the interaction between the vehicle controller and the VPFS.

Section 5: Implementation and Integration: details how the system components were developed and integrated, including API communication and control logic.

Section 6: Verification, Validation, and Performance Evaluation: presents the testing approach and evaluates system performance under various conditions.

Section 7: Limitations, Risks, and Recommendations: identifies key challenges, system limitations, and potential improvements.

Section 8: Conclusions: summarizes the outcomes of the project and highlights future directions.

## 2. Design Criteria and Specifications

### 2.1 System Requirements and Constraints

The design of the autonomous taxi system is formed by a set of technical, financial, ethical, and social constraints that together shape its implementation and performance. From a technical perspective, the system must integrate seamlessly with the Vehicle Positioning and Fare System (VPFS) and operate within its predefined API framework. This requires reliable handling of fare assignments, position updates, and task completion logic. A key challenge arises from the nature of the provided positioning data, which is externally generated at a relatively low frequency and is subject to noise and latency. As a result, the system must handle uncertainty while still maintaining accurate navigation and control. On top of that, the vehicle must satisfy strict operational constraints, including reaching designated pickup and drop-off locations within a specified tolerance and staying stationary for a required duration to meet “in-position” conditions.

Computational efficiency is another important constraint, as the control and planning algorithms must operate in real time without excessive overhead processing. This causes the need for use of efficient path planning and control strategies that can respond quickly to updates while maintaining stable vehicle movement.

From a financial standpoint, the project is constrained to a simulated environment using existing software infrastructure. While this eliminates the cost and complexity associated with physical hardware, it also limits the ability to validate the system under real-world sensing and actuation conditions. Consequently, the design emphasizes software-based solutions that can generalize physical implementation.

Ethical and social considerations influence the system's design. The vehicle must have safe, predictable, and reliable behavior, even within a simulated environment. This includes avoiding erratic motion, adhering to system constraints, and ensuring consistent task completion. Reliability is particularly important, as failures such as missed pickups or incorrect navigation would lower user trust in autonomous transportation systems.

## 2.2 Objectives and Success Criteria

The primary objective of this project is to develop an autonomous vehicle controller capable of effectively interacting with a centralized fleet management system to complete taxi fares within a simulated urban environment. This requires the successful integration of navigation, control, and communication components into a cohesive system.

To meet this objective, the system must accurately interpret fare information provided by the VPFS and generate appropriate navigation strategies to reach both pickup and drop-off locations. It must maintain stable and accurate motion despite uncertainties in position data and delays in system updates. Furthermore, the system must correctly identify when it has reached a target location and satisfy the required “in-position” conditions to ensure successful fare completion.

The success of the project is evaluated based on the system’s ability to reliably complete fares under the constraints of the competition environment. Performance is assessed through factors such as the consistency of fare completion, the accuracy of navigation to target locations, the smoothness and stability of vehicle motion, and adherence to VPFS rules and timing requirements. A successful system is one that demonstrates robust, repeatable performance while maintaining efficient and predictable operation.

## 2.3 Scope of the Project

The scope of this project is focused on the design, implementation, and evaluation of an autonomous taxi controller within the VPFS simulation framework. The work centers on

integrating vehicle control, path planning, and system communication to enable end to end fare completion.

The project includes the development of navigation and control algorithms for waypoint-based movement, as well as the implementation of logic required to interact with the VPFS for fare assignment, tracking, and completion. It also encompasses system-level integration and testing within the provided simulation environment to evaluate performance under realistic operating conditions.

However, the scope is intentionally limited in several areas. The project does not involve the design and limited testing of physical vehicle hardware. It also doesn't include advanced perception capabilities such as object detection or obstacle avoidance. Additionally, it does not address large-scale fleet optimization or multi-vehicle coordination beyond the requirements of the provided system. Broader real-world considerations, such as regulatory compliance and infrastructure deployment, are also outside the scope of this work.

By clearly defining these boundaries, the project remains aligned with the course objectives and competition requirements while focusing on the core challenge of integrating autonomous control with a centralized system.

## 3. Ethical, Social, and Environmental Considerations

### 3.1 Ethical Framework in Design

Ethical considerations were incorporated directly into the system design rather than treated as a separate evaluation step. The team followed a practical engineering ethics approach focused on safety, reliability, and predictable system behavior. Design decisions were made to minimize the risk of incorrect actions, particularly given the uncertainty in position data and communication delays from the VPFS.

A key function applied was fail-safe operation. The controller was designed to avoid aggressive or unstable motion by using conservative thresholds for navigation and task completion. For example, stricter conditions were used to determine when the vehicle is "in-position," ensuring that pickup and drop-off events are only triggered when the vehicle is reliably at the target location. This reduces the likelihood of incorrect task completion.

The system was also designed to behave consistently under similar conditions. Deterministic control logic was used to ensure that outcomes are predictable and reproducible, which supports debugging and accountability. This aligns with standard engineering expectations for transparent and reliable system behavior.

## 3.2 Safety, Fairness, Accessibility, and Sustainability

Safety was the primary factor influencing system design. The controller prioritizes stable and smooth motion to handle noisy and low-frequency position updates. Sudden or erratic movements were avoided to reduce the risk of instability. In addition, the “in-position” requirement was implemented conservatively to ensure that the vehicle stays stationary for the required duration before completing a task.

Fairness was addressed by strictly following the rules defined by the VPFS. The system does not attempt to exploit edge cases in fare assignments and operates within the constraint of one active fare at a time. This ensures equal participation under the competition rules.

Accessibility is reflected in the broader goal of autonomous taxi systems, which aim to provide transportation for users who may not be able to drive. While this project is simulation-based, the design emphasizes reliability and consistency, which are necessary for real-world accessibility.

From an environmental perspective, autonomous systems have the potential to improve efficiency by reducing unnecessary travel and idle time. Although environmental impact is not directly measured in this project, the system is designed to complete tasks efficiently, which supports this goal.

## 3.3 Societal Implications

Autonomous taxi systems have broader impacts on society, particularly in terms of automation and mobility. Increased automation can improve efficiency and safety but may reduce the need for human drivers. This project reflects a small-scale implementation of this shift by demonstrating how a vehicle can operate within a centralized system without direct human control.

In terms of mobility, autonomous taxis can improve access to transportation, especially in urban environments. By removing the requirement for a human driver, these systems can provide more flexible and widely available services.

Finally, user trust is an important consideration. For autonomous systems to be accepted, they must behave in a predictable and reliable manner. The design of this system emphasizes consistent performance and clear task execution, which are important factors for user confidence in this scenario.

## 4. Design Methodology and System Architecture (Julian)

### 4.1 Engineering Design Process

The team employed an iterative, test-driven design process across three phases. As a three-person team (reduced from four due to enrolment changes), the team made an early decision in Week 3 to prioritize reliability and depth of core features over breadth, a scoping choice that guided every subsequent trade-off. In the exploration phase (Weeks 2-6), team members worked on parallel workstreams including sensor calibration and hardware interfacing, dataset collection and annotation for object detection, and 3D CAD modelling for the vehicle shell. The integration phase (Weeks 7-10) brought these components together on the physical PiCar platform, culminating in the first on-track testing in the BAIN Lab, which exposed significant discrepancies between desk-based development and real track performance in areas such as steering behaviour and intersection detection. The refinement phase (Weeks 11-12) was conducted on the full 20ft x 16ft competition map in Beamish-Munro Hall, where over 50 combined team hours were spent tuning per-edge parameters and validating the navigation system against the complete road network.

### 4.2 Alternatives and Tradeoffs

The most significant design decision was choosing not to use object detection in the final competition implementation. The team initially invested substantial effort training a YOLOv5 model (models/yolov5\_TRAINED.ipynb) for six categories (Duck, Left-Oneway, No-Entry, Right-Oneway, Stop, Yield), converting it to int8\_edgetpu.tflite to run on the Google Coral Edge TPU. However, the model exhibited inconsistent accuracy across lighting conditions and viewing angles, and the PiCar's small camera field of view could not reliably capture all traffic signs in its vicinity. The team therefore opted for a purely greyscale-sensor-based approach (deterministic) over camera-based sign detection (non-deterministic), as greyscale sensors provided consistent, low-latency readings directly mappable to steering through a PID controller. This also meant the USB camera purchased with the \$50 budget for dedicated side-view sign detection was ultimately not used, and sign detection code was retained in the codebase with `DETECT_ENABLED=0` but disabled for competition.

### 4.3 System Architecture

Starting from the Picarx base, the perception layer used the three grayscale sensors with a single unified threshold of 630 (midpoint between tape reading  $\sim 1100$  and background  $\sim 160$ ) for tape detection, plus the ultrasonic sensor for forward obstacle detection. The navigation layer consisted of the GraphNavigator class parsing a yEd GraphML map file of the full course and planning BFS routes between nodes, operating as a step-through FSM

that advanced one node each time the grayscale sensors detected an intersection even. For fare pickup/dropoff locations, the VPFS position system was polled to determine proximity. The control layer used a PID line-following controller computing steering angles from grayscale readings, with a tank-turn pivot system for intersections and configuration managed through commands.txt with per-edge flag overrides from the graph file. Finally, an HTTP/SocketIO communication layer interfaced with the Vehicle Positioning and Fare System for fare management, position reporting, and match status.



*Figure 1: Photograph of the car on competition day*

## 4.4 Technologies and Tools

The hardware platform was a Sunfounder PiCar-X with a Raspberry Pi 4B running Raspberry Pi OS Bookworm, equipped with a Google Coral USB Accelerator, the built-in grayscale line-following and ultrasonic sensors, and a Pi Camera Module (640x480 via Picamera2). Additional hardware purchased from DigiKey included a push-button emergency stop and a USB camera. The team added LEDs on a breadboard for brake lights and turn signals, rubber bands on all four wheels for increased traction, and an ArUco marker on top for VPFS overhead camera tracking.

The software stack was built on Python 3.11, TFLite for Edge TPU inference during development, YOLOv5 for model training, Roboflow for dataset annotation, yEd Graph Editor for creating the road network map, xml.etree.ElementTree for GraphML parsing, and requests/python-socketio for VPFS communication. Project management used the GitHub

classroom repository for version control, GitHub Issues with weekly reflection templates, a structured logbook, and SSH over provided course wifi for Pi terminal access.

## 4.5 Algorithms

The core driving algorithm was a PID controller operating on the three grayscale sensor readings, with the error computed as  $(\text{right\_raw} - \text{left\_raw}) / (\text{left\_raw} + \text{mid\_raw} + \text{right\_raw})$ , producing a normalized  $[-1, +1]$  value. Final tuned parameters were  $K_p=17.0$ ,  $K_i=0.0$ ,  $K_d=1.0$ , and  $\text{max\_angle}=35$  degrees. Special cases handled all three sensors on tape (intersection detected, steer straight, PID reset), and no sensors on tape (line lost, hold last direction and widen search angle, eventually stopping automatically as an off-tape safety mechanism that contributed significantly to the team's safety score by preventing the car from veering into the duck population on the map).

The navigation system treated every location where the grayscale sensors detected road width wider than regular tape as a "node event," occurring when all three sensors simultaneously read above threshold, indicating another tape intersected the current road. Due to limitations in greyscale colour detection, the car could not differentiate between stops, crosswalks, and intersections, so all were treated uniformly as nodes and "stop events" before continuing the graph walk. Intersection detection used a hold timer ( $\text{INTERSECTION\_HOLD}=0.15\text{s}$ ) to filter false positives from corners, a cooldown timer ( $\text{INTERSECTION\_COOLDOWN}=0.3\text{s}$ ) to prevent re-triggering.

Turning was implemented as a "tank turn" where one rear wheel spun backward and the other forward. The pivot had two phases: a minimum spin time ( $\text{PIVOT\_MIN\_SPIN}=0.3\text{s}$ ) to clear the current tape, then continued spinning until a grayscale reading within the detection range indicated a new tape segment was found. Despite these tuning efforts and rubber bands on all wheels, pivot turns achieved only approximately 80% reliability due to inconsistent traction on the rubber mat surface, making this by far the car's biggest limitation, identical flags applied to the same turn could succeed on one attempt and fail on the next.

Obstacle avoidance used the ultrasonic sensor with a 13 cm stop threshold; if blocked for 3 seconds, the horn was played. For fare routing in VPFS auto-route mode, the car scored fares as  $\text{payout} / (\text{total\_turns} + 1)$  to favour high-paying, low-turn fares. The only time the car polled the VPFS /whereami endpoint was for fare pickup and dropoff proximity detection, where a dedicated background thread polled at 0.1s intervals without blocking the main loop. When the destination was the next node in the path, speed was reduced to increase proximity accuracy, though the car still drove past endpoints approximately 20%

of the time, especially at difficult locations like P5 and P11 that were placed directly after turns or far from overhead cameras.

## 4.6 Training

The object detection model was trained using YOLOv5 on a Google Colab script. The dataset was collected by photographing the 3D-printed kit signs from various angles and distances then annotated in Roboflow across six categories. The model was quantized to INT8, and compiled with the Edge TPU compiler to produce best-int8\_edgetpu.tflite for the Coral USB Accelerator. The initial conversion workflow using Pascal VOC format proved incompatible with the Coral runtime, requiring a switch to a direct YOLOv5-to-TFLite path. The model was validated through live camera testing on the PiCar with a desktop detection frame display set up in Week 9. While it achieved reasonable accuracy in controlled conditions, performance degraded significantly under variable competition lighting and at oblique viewing angles encountered during actual driving. Combined with the slow inference speed for six classes, these factors led to the decision to disable object detection for competition.

# 5. Implementation and Integration (Julian)

## 5.1 Description of System Components

The main control loop ran at 50 Hz, reading sensor data, computing PID steering, managing the navigation FSM, and driving actuators. The emergency stop button on digital pin D0 proved extremely useful during competition as it allowed team members on the mats to halt the car and manually reset it to the correct graph node, enabling very quick and efficient restarts. Brake and turn signal lights were implemented as LEDs on a breadboard; due to limited input pins on the Pi, both braking LEDs were connected in parallel to the same input, while turn signals used PWM channels P8 and P9 driven by background blinking threads, with hazard mode flashing both simultaneously during stops, fare pickups, and obstacle encounters.

The graph navigation system's per-edge flags, embedded in the yEd graph file's edge descriptions, were essential for handling the diverse geometric situations across the map. A screenshot of the graph is seen below:

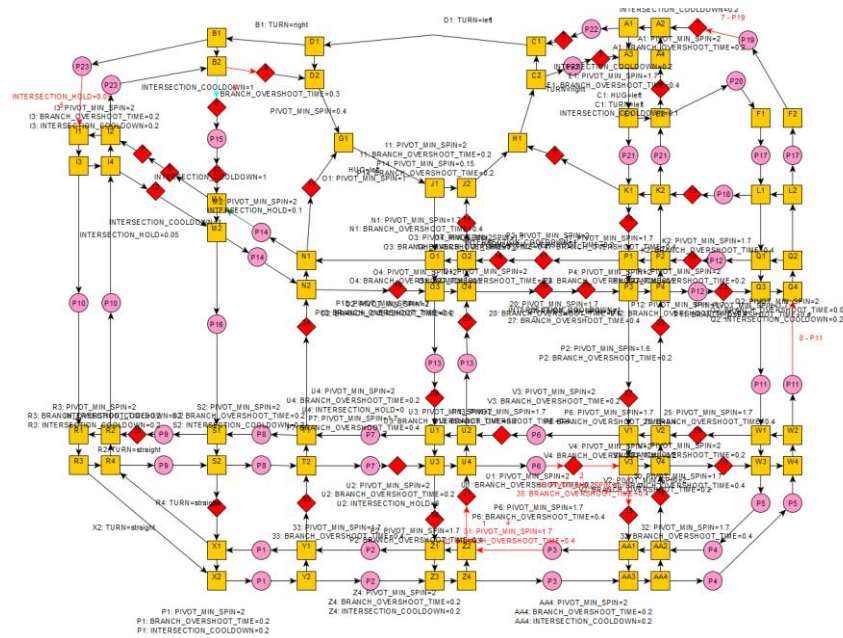


Figure 2: Screenshot of graph used for car navigation, intersections are represented as yellow squares, stop lines and crosswalks as red diamonds and pickup/dropoff locations as pink circles.

The edge specific flags were:

Table 1: Edge-specific flags used in map graph

Edge flag	Purpose
BRANCH_OVERSHOOT_TIME=<s>	Drive time past intersection before pivoting
PIVOT_MIN_SPIN=<s>	Debounce spin before tape detection can stop the pivot
SPEED=<int>	Override forward speed on this segment
HUG=left right	Steering bias used for difficult corners (eg. roundabout)
INTERSECTION_HOLD=<s>	Hold duration before intersection fires
INTERSECTION_COOLDOWN=<s>	Suppresses re-triggering after leaving node

## 5.2 Integration Strategy and Challenges

Several significant challenges emerged during integration. The FSM nature of navigation meant missing a single node put the car completely out of sync, making every subsequent turn incorrect. This made robust node detection absolutely critical, driving significant effort into tuning INTERSECTION\_HOLD timers. An implementation that contributed strongly to the safety score was automatic stopping when the car detected it was off the line tape, preventing it from veering into the duck population on the map. For fare stops, pickup/dropoff locations (ellipse nodes) had no tape markings, so the car polled the VPFS

/whereami endpoint with a proximity threshold of 22 cm; however, the car drove past endpoints roughly 20% of the time, especially at P5 and P11 which were far from overhead cameras or placed directly after turns.

### 5.3 Implementation Artifacts

All source code is maintained in the GitHub classroom repository (elec-392-project-aylesbury-12). Key files include src/main.py (main control loop, PID follower, VPFS client, sign detector, LED control, e-stop, auto-route state machine), src/graph\_nav.py (GraphML parser, BFS routing, GraphNavigator FSM, edge flag parsing), src/print\_route.py (route diagnostics), BMHmap.graphml (complete road network graph in yEd format), commands.txt (all configuration parameters), models/yolov5\_TRAINED.ipynb (training notebook), and logbook/ (weekly development logs).

### 5.4 Project Development and Milestones

A significant portion of the final implementation was done in the last week (Week 11), as that was when the team first had access to the full mat setup and could test the car on the complete Quackston map. The graph based navigation system could only be fully validated on the actual competition layout. The compressed period reflected both a limitation (late map access) and a strength, as the team had built robust, highly configurable infrastructure in earlier weeks that allowed rapid adaptation once the full map was available, with the per-edge flag system proving especially valuable for tuning individual intersections without code changes.

The early decision to prioritize deterministic greyscale navigation over non-deterministic object detection proved correct, producing a far more reliable and repeatable system. The per-edge flag architecture, emergency stop button, and off-tape auto-stop all worked reliably throughout testing and competition. Conversely, the pivot turn inconsistency (~80% reliability) remained the single largest source of failures and could not be fully resolved within the project timeline. Late access to the competition map compressed the critical tuning period, the speaker issue was never resolved, and the object detection work (~25+ person-hours) was ultimately unused.

### 5.5 Development Timeline

Table 2: Project timeline on week-by-week basis

Week 2	Kit verification; Raspberry Pi access confirmed.
Week 3	Project scope defined for 3-person team; proposal distributed.
Week 4	Sensor, camera, SSH tested; PiCar measurements taken.
Week 5	Sign dataset collected/annotated in Roboflow; first TPU conversion.

Week 6	E-stop and USB camera ordered; YOLOv5 retrained to edgetpu.tflite.
Week 7	Oral assessment / design review with TA.
Week 8	Edge TPU model deployed on PiCar; 3D shell finalized.
Week 9	DPFS research; detection + line following integrated; SolidWorks shell created; first BAIN Lab track test (all members).
Week 10	PID tuned ( $K_p=17$ , $K_d=1$ ); pivot turns, signals, e-stop wired.
Week 11	30+ team hours on full map; per-edge flags validated; diagonal detection introduced; nav system tuned.
Week 12	Final report distributed; PiCar kit returned.

## 6. Verification, Validation, and Performance Evaluation

The team tracked two main metrics for the performance of the vehicle. Line following success rate and turn success rate. It was decided that these two metrics were the most important factors in a successful vehicle, and other metrics, such as task completion time, would be dependent on them. In competition, both of these metrics were found to have a very high success rate. Line following had a very high (>95%) success rate, failing only three times in the entire competition day. This exceeded the designed for goal of 90%. Similarly, the steering system performed successfully as well. Due to the imperfect conditions of the rubber mat, as well as the friction on the tires, the successful turn rate was slightly lower, at a success rate of 80%. This met the designed for goal. The high success rate of the line following coupled with the high success rate of the steering translated into a very successful competition vehicle.

However, this vehicle was not without failures. Two main issues arose in real world implementation. The first, and most prominent, was changing light levels in the testing area. Relying on the greyscale sensor for implementation meant that the car was very sensitive to dimmed light levels. As daylight faded the car became less reliable on nodes it had previously completed due to missing crossings and then losing position. This issue was not as prominent during the competition due to daylight being present. However, the team solved this issue by lowering the threshold for a successful detection, and in some cases holding a flashlight on nodes that were particularly dark on specific routes. Additionally, the team had trouble with a specific seam between mats. One mat was slightly higher than the other, and when the vehicle would approach the seam would be detected as an obstacle. As there was no feature to “check” if there really was an obstacle in front of the vehicle, a team member would have to push the seam down before the vehicle could proceed.

When comparing this vehicle to the team’s initial goals and expectations, the vehicle performed very well, meeting and exceeding expectations in most criteria. The main attribute the team wanted to focus on was reliability. Overall, the success in both turning and line following meant that this vehicle performed very reliably, with very few safety violations, and was able to claim and complete multiple fares in a session.

## Violations



Figure 3: Screenshot of week 11 lecture content showing highest overall safety score.

This reliability translated into an excellent competition performance for the team. The team placed first overall in reputation, second overall in cash made, and first overall in overall competition score. Additionally, the team placed first in the “All-Star Game” held at the end of the competition day. While results this high were not expected from the beginning, the vehicle was designed and tuned to function very reliably and that was seen in the competition. This is due to our very successful turning and line following functions, as well as the robust navigation system that was developed. The vehicle performed very well and the competition was an overall success for it and the team.

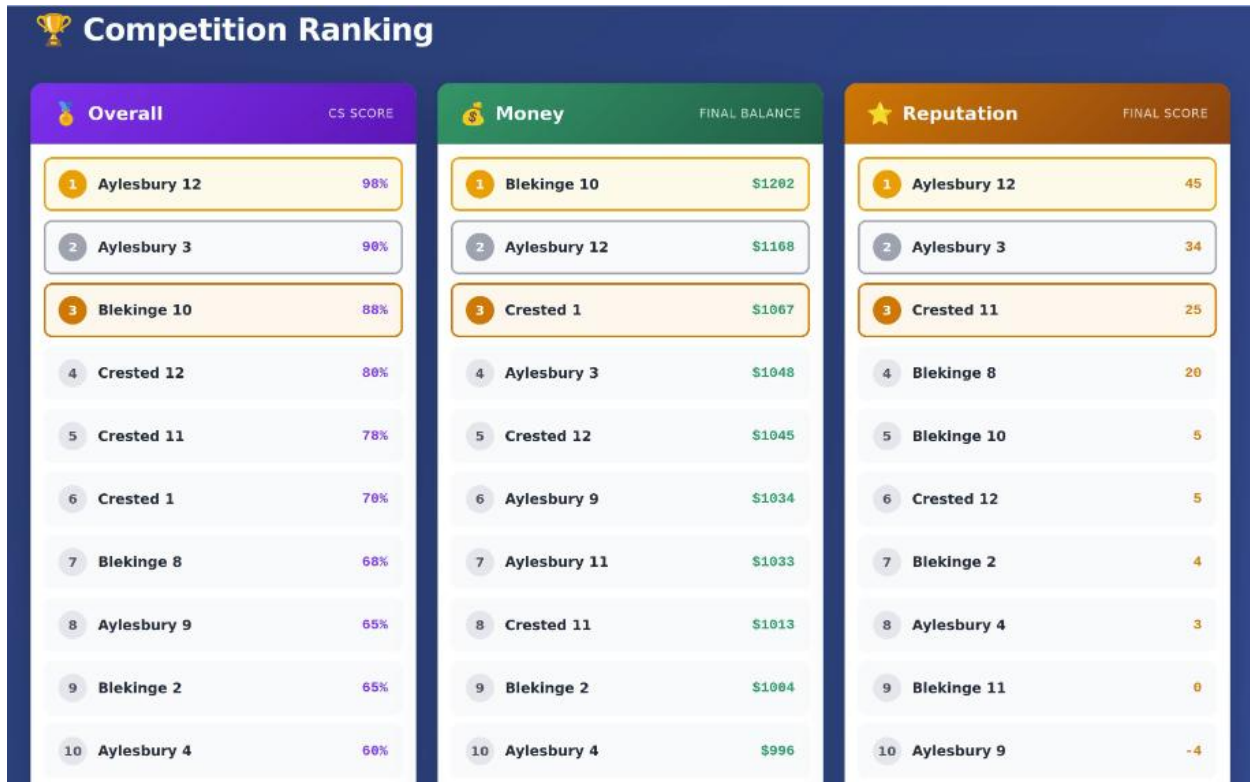


Figure 4: Screenshot of competition day results.

## 7. Limitations, Risks, and Recommendations (Tate)

In the implementation and tuning stages of the project. The vehicle faced some challenges. The first was the sensitivity to light. Relying on the greyscale sensor for navigation meant that changing light levels had a massive effect on the reliability of the system. This was a challenge that had to be solved through tuning the sensor to the specific light level the vehicle was operating in.

Another challenge was the navigation system of the vehicle. The group chose to have a preprogrammed graph of Quackston in the code of the vehicle for navigation. This meant that the VPFS was only used to determine the location of pick-ups and drop-offs. So, when the vehicle's navigation failed in some way, such as missing a turn, the vehicle could not determine where it was and find a new route. It would think it had made the turn and make further mistakes as navigation continued. In cases like this a team member had to manually reset the vehicle. The last challenge faced was integration of the camera in a useful way. Due to the poor FOV provided, it was very difficult to use the camera to provide an accurate picture of what was happening around the vehicle. Ultimately, the camera was not used in the final product because of this. A large limitation of the vehicle arose from the camera not being used. The vehicle had no way of knowing what was around it. It could

only use the ultrasonic sensor to detect objects in front of it. This meant that it would not “look” before turning. An unsafe behaviour which could have led to a collision between vehicles in competition.

This limitation was a product of one of the assumptions made by the team. This assumption was that there would be sufficient spacing between the vehicles on the track to the point where object detection with the ultrasonic sensor would be efficient. This was an assumption that held up during competition. There were no cases where multiple vehicles were at an intersection at the same time due to vehicle spread. So, the system limitation of not looking for other vehicles was not a problem that the team faced. The other assumption the team made was that the navigation system would not fail. This was because there was no failsafe when the vehicle became lost. While this assumption was not 100% accurate, it was still accurate enough for the team to be successful. There were failures of the navigation system, but overall, the system was very reliable and did not require much team intervention.

From this project, first lesson the team learned was how difficult it is to build a truly autonomous car. While our implementation was successful on competition day, it was not a perfect vehicle. It had a preprogrammed graph of the entire map and had no functionality to determine if it was lost and how to recalculate a route. It would not be possible for our system to operate outside of the predefined bounds of Quackston. However, another lesson the team learned was that reliability was the most important factor for success at competition. While the vehicle had no way of righting itself if it was lost, if the vehicle could line follow and turn accurately, the chance of getting lost was very low. This was an assumption the team made early in the design process and it proved to be correct. So, while features to build on the autonomy of the car are important. The most important factor was ensuring the vehicle could operate reliably in a predefined environment.

Due to the limitations of our system. This vehicle is not ready to go to mass market. There are too many gaps in the safety systems for it to be a vehicle that can operate in the real world. The assumption that there would be enough space to not have to worry about other vehicles proved to be valid in Quackston but is definitely not valid in the real world. If this system was brought to market, there would be numerous collisions due to this fact.

To improve our system to make it market viable, the main recommendation the team has is to improve the camera system in future iterations. A better camera system would be more useable as part of the navigation systems of the vehicle and would allow the vehicle to “see” what is around it. This would allow the vehicle to operate safely in the competition environment and the real world, as well as bring it closer to a true autonomous vehicle.

## 8. Conclusions

The Aylesbury 12 team successfully designed and implemented an autonomous vehicle for use as a taxi in the simulated urban environment of Quackston. This successful integration serves as a steppingstone to base further autonomous vehicles off of.

The primary outcome of this project was a highly reliable vehicle that performed very highly in a competitive environment. Key takeaways include the success that came from prioritizing reliability and modularity of the vehicle and its navigation system. Allowing for quick granular modification of specific nodes to ensure the vehicle was successful in all environments.

The early focus on reliability and modularity of the vehicle led to use of a per-edge flag system in a GraphML map. This allowed for very quick coarse tuning before the full map was available, and then very quick fine tuning without extensive code changes required. Specific flags could be changed for different edges depending on real world performance without impacting the rest of the graph. This allowed for imperfections in the map to be easily accounted for in the software of the system. This methodology led to a very successful competition vehicle. The team achieved first in reputation, second in cash made, and first overall in the entire competition.

To improve this vehicle for the future, the team has two main recommendations. The first is improvement of the object detection system by implementing a more robust camera with a wider field of view. This would allow for more reliable object detection at intersections for collision avoidance. The second is to implement a recovery feature utilizing the VPFS. This would allow the vehicle to see when it had missed a turn, and recalculate the best way to get to the destination.

However, the overall project was still very successful. The team implemented a vehicle that functioned reliably and performed extremely well at competition. This performance showcases the importance of reliability and predictable behaviour in autonomous systems, and how with the addition of a few key safety and positioning features, this vehicle can provide a solid base for future work in accessible and sustainable autonomous transportation solutions.

## References

- [1] American Society of Civil Engineers, “Autonomous vehicles are here — but where are they going?” Civil Engineering Source, Dec. 8, 2025. [Online]. Available: <https://www.asce.org/publications-and-news/civil-engineering-source/article/2025/12/08/autonomous-vehicles-are-here-but-where-are-they-going>. Accessed: Mar. 30, 2026.
- [2] “ELEC 392 GitBook” GitBook, 2025. [Online]. Available: <https://elec392.gitbook.io/elec-392/8Ty9Nx3I84dHok0cVgCY>
- [3] Encyclopædia Britannica, “Autonomous vehicle,” Britannica, [Online]. Available: <https://www.britannica.com/technology/autonomous-vehicle>. Accessed: Apr. 2, 2026.

## Appendix A: Team collaboration and professional practice

The overall philosophy of the team for this project was allowing everyone to be independent in their work to promote quick, efficient, work, that was not dependent on the work of someone else. This helped remove bottlenecks in the design process and contributed to the overall success of the group.

Responsibilities were distributed as to who the team thought was best to handle them. In the early design phases the team used the weekly meetings to discuss what needed to be done and would then assign and complete tasks from there. In the testing phase the group member who's system was being tested would do tests as required and the other group members would watch the vehicle from the mat and provide feedback as to where the vehicle failed and then provide recommendation on flags to change. This system worked well, where the system designer had the information and feedback, they needed to make their system work reliably and successfully. Responsibilities/workload equity was self-managed and regulated. If a team member required assistance, they would ask for help from the team, and the other members would assist as they were able. This system worked well because the entire team shared the same goal of competition success, and were willing to help each other.

The team used the weekly meetings to handle larger design decisions, discussing ideas, landing on a solution and assigning tasks from there. However, the team made the decision to allow team members to make many independent decisions within the system they were working on. This allowed them to implement the vision they had for their system and not have to wait for group approval while working. While a conflict resolution plan was in place, it was not required.

Giving such a high level of independence to team members was a system that worked well for the team. However, the overall workload distribution could have been managed better by the team. The navigation system was mostly given to one person and was a very high level of work. While this system worked well for us, it very easily could have meant that one team member became burnt out and could not complete the entire system on time, delaying timelines. To improve this, the team should have better analyzed the requirements of each task in the early stages of the design cycle to see the level of work required and distribute more equally from there.

## Appendix B: Individual Reflection on Skills and Performance (All)

### Individual Reflection – Connor

My primary contributions to this project were the training of the object detection model and the implementation of the VPFS integration. We used Roboflow and Google Collab to train a model, then converted it for deployment onto the Edge TPU. I also helped with implementing communication with the VPFS, including retrieving fairs, tracking vehicle position and handling pickup and drop-off logic.

Through this work, I developed practical skills in machine learning workflows, API integration and real-time system interaction. I gained experience working with tools such as Roboflow and Colab, as well as dealing with issues like unreliable model performance and system latency. Some gaps I identified for myself were the need for deeper experience in low-level control and debugging on hardware and having better/more consistent communication with my team.

A key lesson from this project was the importance of reliability over complexity, especially when your timeline is relatively short. Although object detection was initially a major focus, it proved less effective than simpler, deterministic methods in the competition environment. In future projects, I would prioritize earlier testing and a more critical evaluation of whether complex features provide real performance benefits.

### Individual Reflection – Julien

My role was software lead and systems integrator. I wrote the core competition codebase, including the PID line-following controller, graph-based navigation system, main control loop, VPFS fare optimization function, and per-edge flag override architecture, and handled some of the hardware wiring on the car. I also led the Edge TPU model conversion pipeline, wrote the majority of logbook entries and meeting minutes, and authored the Design Methodology and Implementation sections of this report. My most intensive contribution was the Week 11 full-map testing in Beamish-Munro Hall, where I spent a full weekend systematically running every route segment, identifying failing intersections, tuning per-edge overrides, and fixing the turn direction inversion bug.

On the technical side, I developed a much better sense of how to design configurable software for real-time embedded systems. The commands.txt system and per-edge flags

started as a practical workaround because redeploying over SSH was too slow during track testing, but it turned out to be the project's best investment. I also built stronger intuition for PID tuning through the BAIN Lab sessions, and more broadly gained experience with the full cycle of training an ML model, quantizing it, and deploying it to edge hardware, even though the detection system was ultimately not used in competition. Beyond technical skills, this project gave me real experience coordinating work across a small team, maintaining consistent documentation under time pressure, and making scope decisions that traded ambition for reliability.

Looking back, there are a few things I would do differently. I took on too much of the implementation work myself, which created a bottleneck where the car's progress depended heavily on my availability during the critical final weeks. I should have done more to distribute the software work earlier and pair-program with teammates on the navigation and control systems so that more than one person could debug and tune the car on the mats. I also underestimated the importance of mechanical reliability. The pivot turn inconsistency was ultimately a traction and weight distribution problem that no software fix could fully address, and the speaker never worked at all. In a future project I would invest time in physical prototyping and stress-testing hardware earlier rather than assuming it will cooperate once the software is ready. More broadly, our team left too much of the full-map integration work to the final week because that was when we first had access to the Beamish-Munro setup. While the configurable architecture allowed us to adapt quickly, better planning around facility access and earlier requests for map time would have reduced the pressure significantly. The most important lesson I am taking forward is to build diagnostic and testing tooling before writing core logic, and to plan the project schedule around when physical testing resources are actually available, not when they are ideally needed.

## Individual Reflection – Tate

My primary contributions to this project were the design of the 3D shell of the car, the development and installation of the CARE system, and assistance with the testing of the car. I took measurements of the vehicle in the 2<sup>nd</sup> meeting and built a SolidWorks assembly to better view how the shell would look on the car. From there I designed the shell and conceptualized a rubber band-based CARE system to secure the ducks in transit while also allowing for easy installation and removal of a duck. Additionally, I worked with Julian to help test the vehicles navigation system when Quackston was available for testing. Julian had developed the system, so he planned out routes while I would monitor how and when the vehicle failed so we could tune the system to function reliably.

A very large skill gap I encountered in this project was my lack of experience in developing computer systems and coding knowledge. I felt as though this project had little to no work for me as a third-year electrical engineer to apply the knowledge I had learned in class. Due to restrictions placed on touching the electrical system the only real electrical work was installing the brake and turn signal lights, which did not offer any skill development for me. This left me as someone who struggled to code in a project where coding was the only work to be done. In a team setting, this meant that I struggled to contribute because the computer engineers in my group could complete a task much faster than I could and if a coding task was assigned to me, it would end up slowing the group down. I was able to develop skills in physical system design by designing the vehicles shell and CARE system, but I did not feel like this project offered me the skill development I initially expected.

However, I was still able to draw some lessons about electronic systems from this project. The success of our vehicle showed me that reliability and predictability are incredibly important in an autonomous system. Our vehicle was not the fastest on the map, but because we had spent such a long time making it behave predictably, it worked extremely well when other vehicles were failing quickly instead of succeeding slowly. Overall, while I am grateful to have learned lessons like this I would recommend that future projects provide electrical engineers more work tailored to their skill development so that both electrical and computer engineers can find this project equally valuable.