CAZ App

Introduction

This report details our final project for Code First Girls. We'll walk through the objectives of the product we built, why we built it, our technical approach to solving the problem and the ways of working we employed to get to the final product.

Aim

To empower driver's to make more informed transport decisions in relation to current climate regulation.

Objectives

In order to achieve our aim we will build a platform that:
- Informs the user of the UK clean air zone initiatives
- Allows the user to check their vehicles exemption within Bath's Clean air zone
- Gives options to plan their journey accordingly

Background

In 2019, transport produced 27% of the UK's total emissions, and of this, 91% came from road transport vehicles (Department for Transport, 2019). Cars and taxis were responsible for 61% of emissions from road transport, followed by heavy goods vehicles (18%), and vans (17%).

Considering these statistics, the UK government has started implementing policies to improve air quality in the UK. In order to reduce harmful emissions in cities, many are now adopting a 'Clean Air Zone' (CAZ). A CAZ is an area that penalises highly-polluting vehicles based on their Euro Emission standard. Vehicles that do not meet this standard, and thus have higher emissions, are either unable to enter the zones or are charged to enter (Energy Saving Trust, 2021).

Currently, only a few cities have implemented a CAZ - their expansion across the country was halted during the COVID-19 pandemic. There has been a 'low emission zone' in London since 2008, and this was extended to be an 'ultra low emission zone' in 2020. Bath was the first city outside London to implement a CAZ in March 2021, shortly followed by Birmingham in June 2021, and Portsmouth in November 2021. Other cities will soon have a CAZ, including Bradford, Bristol, Greater Manchester, Sheffield, and Tyneside.

Bath Clean Air Zone

Bath has a 'Class C' CAZ, which means all higher emissions vehicles (except most private cars and motorcycles) will be affected and charged to drive though the city centre. Older models of buses, vans, lorries and taxis are under obligation to pay a daily pollution charge, which ranges from £9 to £100 (depending on the type of vehicle). All roads leading into the zone have automatic number plate recognition cameras, which is checked against the DVLA (Driver and Vehicle Licensing Agency) database. Drivers with non-exempt vehicles who do not pay the charge, and drive through the CAZ boundary, are subject to paying a fine.

CAZ App

There is still relatively little knowledge of the CAZ in Bath and limited signage near the boundaries, leading to a substantial amount of fines sent to citizens. It is estimated that there were 28,000 fines given in the first hundred days of the CAZ being implemented. As such, there has been some public anger and distrust over the CAZ. Drivers (especially non-local drivers) are often caught unaware of the rules pertaining to what constitutes a compliant vehicle. Some members of the public argue it is affecting their businesses; for example, if they require large vans (deliveries, or manual labour work) or passenger travel companies that use minibuses or taxis.

At present, there are no applications either on the government website or the BANES website to guide users around the CAZ. Popular directions applications, such as Google Maps, do not outline the CAZ or offer any solutions on avoiding it. Considering the popularity of such apps, it is feasible to suggest an app that provides this information would be popular and beneficial.

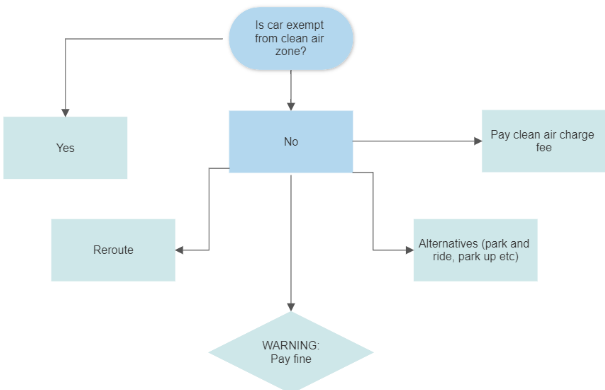
With our interactive website app, we hope to solve the current controversies between drivers and the BANES CAZ. We hope to encourage users to interact with the zone more efficiently. Users will be fully informed about the CAZ and to make more climate-friendly transport decisions. If successful, it is our hope we can extend the app to cover the other clean air zones around the UK.

Product specifications

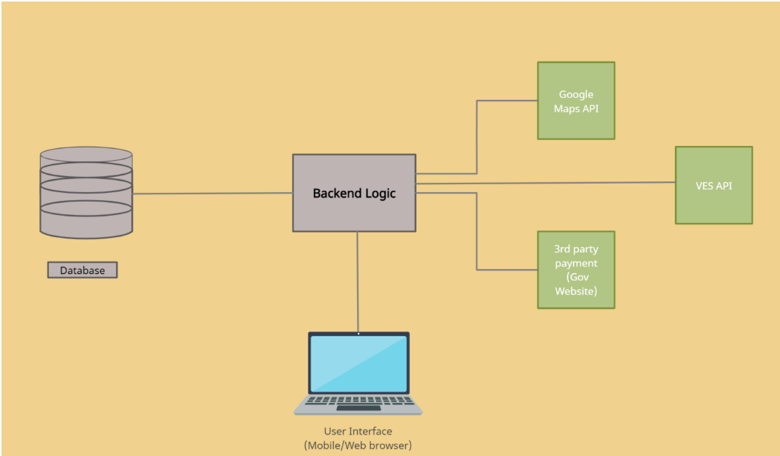
Once we decided we wanted to build a product to help motorists manage their Clean Air Zone liability we narrowed down the features the product should have.

Feature	**Description**
User Interface	Simple website design including: ● Welcome page: summary of Clean Air Zones with a thumbnail photo and date of each city that has or will be implementing a CAZ ● Clear logo ● Clear buttons for navigation
Account registration/Log in	● The SQL database will store user information (email address, first name, car reg and county). ● Entering reg number will call the vehicle information from the DVLA API and pre-populate the details linked to that registration (fuel type, colour, tax status, vehicle type, make, MOT status, tax due date)
Interactive Map	● The UI will display a map with data layers including the CAZ boundary and road points that intersect the boundary.
Registration Input	● API info and Python code is used to return vehicle exemption status.
Pay Charge button	● In the case a vehicle is not exempt, the user will be given the option to pay via a link to the government website.
Support page	● Users can contact us if they have any issues. They can enter their first name, email, and county, and send a support request.

Design/User Flow



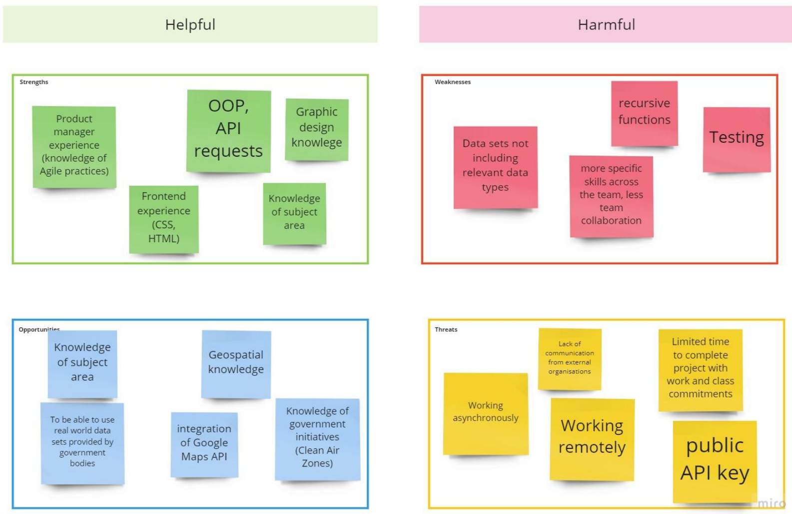
Architecture



Implementation And Execution

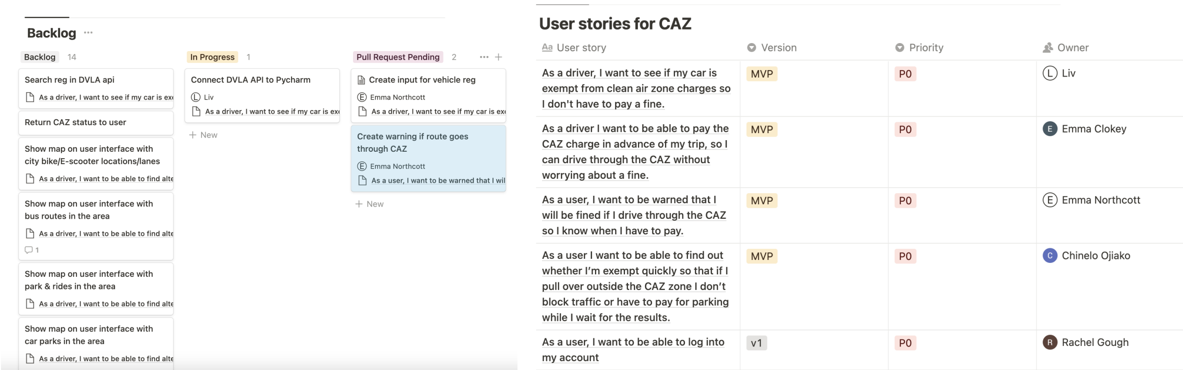
First, we determined the feasibility of the project and whether we could implement our ideas in the given time frame. This included outreach to various authorities to obtain necessary datasets/APIs that are not readily available. We contacted government agencies (mainly BANES council and the DVLA) to obtain access to their API datasets. We connected with them on the basis of common ground - for instance, we asked BANES for their CAZ information so we could assist users navigate the zone and direct them to their website to pay the charges, ideally to aid them in their goal of reducing emissions in the city. This encouraged an active engagement towards our project. In addition, government data access requests were submitted (e.g. specialist vehicle data from the DVLA) with justifications on usage and storage.

The result of this initial outreach determined our project to be specific to Bath, and allowed for planning to begin. After an initial SWOT analysis, we determined our collective strengths, weaknesses, opportunities and threats regarding the project.



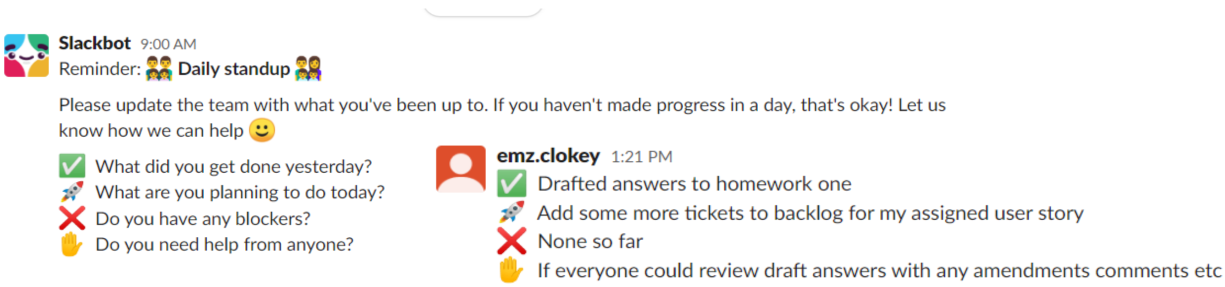
Our Swot Analysis.

This analysis identified two main constraints to our project: working remotely and asynchronously. To remedy this, we ensured that all work would be uploaded to respective, collaborative documents: Notion for backlogs, meeting notes, and essential files; a shared GitHub repository for Python code; a Slack channel for reminders, issues and other elements.



Example of our collaborative backlog (Left) connected to user stories on Notion (Right).

Our development approach was based on agile development methodology. We implemented aspects of the Scrum frameworks that we adapted to our online learning and different time schedules. We assigned the following roles based on the individual's knowledge and experience: Emma Northcott: Product Manager, Liv Moran: Scrum Master, Rachel Gough: Backend, Emma Clokey: Backend, Chinelo Ojiako: Frontend. Daily Standups were conducted via our Slack workspace to ensure each individual remained accountable, were able to track their progress, and ask for help within a 24-hour timeframe. We met regularly before Monday evening classes for 15 minutes, and before other classes when required. This meeting was like our daily scrums, with a chance to show the team what each person had worked on, any blockers they had, and to ask for the team's help if necessary. A group meeting was held every Friday, doubling as a sprint retrospective and sprint planning to establish clear priorities.



Example of daily standup reminder and responses.

Tools and library	Why we used them
Canva.com	• For the creation of our logo • Created a an animated logo for our project name • Used a gif editor to set the loop to once so it plays only when the page is loaded
Bootstrap	• Used to style and design our frontend page

PyCharm	● Used to write Python and to store most of our codes, including the licence plate checker and the fine warning ● Flask framework and json to build our web applications and to handle json request using the REST method ● For debugging and unittests ● From pyproj - imported Prog and transform which was used to convert coordinates within datasets from eatern-northern to longitudes and longitudes
mySQL	● To write SQL code to store user data such as email, name and reg number
Folium	● Import our geojson file to create a map layered with the CAZ boundary
Google Maps API	● External API used to include maps on our website ● Add eco-friendly options e.g. park-and-rides to our map
DVLA API	● Returns all vehicle data based on registration number ● Return specific data for the licence plate checker function (reg. number, date of first registration, vehicle type) ● Return specific data for the account information (as above but with extra information such as vehicle colour, make, MOT/Tax status and Tax due date.
Pandas	● Used to analyse external data files (.csv file sent to us by the council) ● Manipulates files to correct format for our usage
Notion	● Used to organise our meeting notes notes ● Add project requirement documents ● Organise tasks in terms of priority and assign to a team member ● Create user stories and set up tickets
Json (geojson):	● Transforms our data (e.g. DVLA API data) into a readable format ● Converts data to a file
Github	● Enabled remote working by providing a remote repo to upload code ● Enabled team members to work different branches and then merge everything together
Unsplash	● Background pictures for frontend

Implementation Challenges

The initial scope of our project was quite large and ambitious. We managed to create an MVP of the product, but it doesn't have all the functionality we'd originally intended to build. Some particular areas we struggled with:

- Maps API: We originally intended to use Google Maps, but we discovered that you needed to use JavaScript. We weren't able to upskill quickly enough in JavaScript to build what we needed. We made the late decision to switch to Folium instead. This map has more limitations on functionality so we weren't able to map directions and provide rerouting like we originally intended.

- Map Overlay: Mapping the Clean Air Zone over the Bath map was proving very difficult. It turned out the shape file data for the overlay was using a different coordinate system to the map. This meant that we couldn't align the overlay over the map. In order to overcome this we converted all the data into a consistent format, latitude and longitude. A similar method was used for the road entrance points layer.

```python
import json
from pyproj import Proj, transform

with open('data.geojson') as input_data_file:
    data = json.load(input_data_file)

coordinates = data['features'][0]['geometry']['coordinates'][0]

northings_eastings = Proj('epsg:27700')
latitude_longitude = Proj('epsg:4326')

for index, coordinate_pair in enumerate(coordinates):
    easting, northing = coordinate_pair
    latitude, longitude = transform(northings_eastings, latitude_longitude, easting, northing)
    coordinates[index] = [longitude, latitude]

with open('newdata.geojson', 'w') as output_data_file:
    json.dump(data, output_data_file, indent=2)
```

- Collaboration: Everyone in our team had different schedules. Some people worked full-time, some part time. We had other personal commitments that made it challenging to meet up regularly. Bearing this in mind, we designed asynchronous ways of working to make sure that we could still make progress even when we're unable to meet together.

Testing And Evaluation

Our testing strategy was to both manually test our code as well as write unittests to make sure the functions were working as intended.

Conclusion

We learned a lot through this project. Initially we overestimated what we could have achieved in the time period. There was a steep learning curve along the way. We've all improved our coding skills and our understanding of how to collaborate with other engineers. In the end we've built a product that taught us a lot and supports the government's efforts to have a cleaner future.

REFERENCES

Chua et al. 2020 (https://bjo.bmj.com/content/106/5/705.abstract)
Doiron et al. 2018 Air pollution, lung function and COPD: results from the population-based UK Biobank study
DEFRA, 2014 - Guide to UK air pollution information resources
(https://uk-air.defra.gov.uk/assets/documents/reports/cat14/1406191156_060618_Guide_to_UK_Air_Pollution_Information_Resources-issue_2-FINAL.pdf)

Department for transport, 2021 Official Statistics
Transport and environment statistics: Autumn 2021
Published 19 October 2021

Energy saving trust, 2021 https://energysavingtrust.org.uk/a-quick-guide-to-englands-clean-air-zones/

Health effects institute, 2019,
Health Effects Institute. State of Global Air 2019. Special Report (Health Effects Institute, Boston, 2019). ISSN 2578-6873.

Liang and Gog, 2020 Urban and air pollution: a multi-city study of long-term effects of urban landscape patterns on air quality trends

World Health Organisation, 2021
https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health