GPIG: Group Report

Joe Lenton, Samuel Pawlikowski, Jack Tolley,
Matthew Turton Parry, Nick Vaughan and Peter While

June 2020

Word count: 9990
(Obtained from Overleaf)

Live Demo: `http://35.242.161.192`

Repository: `https://gitlab.com/JackRTolley/gpig`

Contents

List of Figures

List of Tables

1 Introduction

The concept of smart cities is an emerging paradigm in modern urban design, typically characterised by the large-scale collection and analysis of data, which can be used to vastly improve the efficiency of the city's resources and services. This data is usually gathered by a network of Internet of Things devices, monitoring such metrics as footfall, traffic, and public transport usage, among many others. The wealth of information produced by these sensors can then be collated and reviewed to provide insights into the changing needs of the city, and its population.

By appropriately utilising the potential of smart technologies, it is possible to significantly reduce the waste produced by large urban areas. Commonly cited examples include adaptive street lighting, which adapts the brightness of street lights based on movement [1], and intelligent traffic management systems, which seek to reduce travel times and congestion [2]. In this way, the environmental impact of these cities can be substantially reduced, providing a host of benefits, from improved air quality, to reduced energy costs.

The availability of large amounts of data also poses a potential benefit for businesses; by analysing and understanding this information, companies will be better able to predict the needs and desires of their customers, while at the same time reducing their expenses by improving efficiency.

This paper introduces a novel proposal for an intelligent congestion charge, which adapts the price dynamically to encourage a lower, balanced traffic level across the entire city. By making use of a city-wide network of traffic cameras, footfall detectors, and emissions sensors, powerful machine-learning algorithms can compute the optimal congestion charges for different areas of the city to discourage congestion, without stifling travel into the city.

2 Vision

The proposed system is an intelligent, dynamic congestion charge, which makes use of both real-time and historic data, in conjunction with adaptive machine-learning, to reduce and optimise the flow of traffic around the city of York. Traditionally, congestion charges are implemented as a simple, daily fee for driving within a city during a specified time period, as is the case in London [3]; while effective at reducing congestion, these systems are blunt and inflexible, leaving significant room for improvement.

A more intelligent system of applying congestion charges exists in Stockholm [4] [5], where the city is split into two separate charge zones, each with a fixed price for every hour of the day. The system we are proposing extends this premise in two ways: firstly, by splitting York into several charging zones, and secondly, by utilising artificial intelligence and real-world data to calculate truly dynamic, adaptive congestion charges for each zone. These improvements will allow the system to dynamically 'steer' traffic around York, discouraging heavy traffic in residential areas, and making retail and tourist-heavy areas more pedestrian-friendly, while improving the overall congestion management across the city.

2.1 Motivation

Congestion has been identified as a key challenge for York's economy. In particular, heavy traffic near the northern ring road has been cited as a reason for the unattractiveness of business premises in those areas [6]. This is an issue because York struggles to attract businesses from sectors such as professional services compared to other cities. There is a lack of available quality office space and limited available space to build. Additionally, the bioscience and agri-tech sectors, which have been identified as York's most likely growth areas, have reached lab capacity [6]. A reduction in congestion and subsequent reduction in travel times could make the less occupied areas of York, such as near the northern ring road, more attractive for building new business premises.

Additionally, air quality is a known issue in York [7]. Investigations found air quality at 15 of York's monitoring stations breached WHO guidelines for NO_2 [8]. Additionally, the council acknowledges 3 areas, including the city centre, where NO_2 targets are exceeded, and they are legally required to address this within 12-18 months [7]. More radical action is required; the council itself acknowledges that its own "Low Emission Strategy ... won't be enough to deliver our 'air quality objectives' at all locations in York" [9].

2.2 User Stories

In order to improve the understanding of the system and its context, a number of user stories were brainstormed, each detailing a use case from the perspective of an end user.

- As a regular commuter by car from outside of York, I want to be able to see predicted prices for the future, so I know how much my commute is going to cost me.
- As a regular commuter by car from outside York, I don't want to make individual payments each day, and I don't want to have to pay physically, so that I'm not inconvenienced my the new system.

- As an occasional visitor to York by car, I want to know when I should time my visits, so I can avoid being sat in traffic and save money.
- As a business visitor to York, I'd like the option to pay my congestion charge in advance, so I can budget for a fixed price.
- As someone who drives through York but does not want to check the website / use the internet, I would like charges to be clearly signed, so that I'm not hit with an unexpected bill.
- As a professional services business owner in York, I would like the city's congestion to be lower, so that working for and trading with my business is more attractive.
- As a hotel owner in York, I would like to see York have less traffic in central areas near tourist attractions and during popular events, so it is more accessible to tourists on foot and using public transport.
- As a shop owner in the centre of York, I would like to see York have less traffic in central areas near my shop, so footfall to my shop is increased.
- As a councillor for York, I would like to reduce emissions from traffic in York, so that York can meet its legal street-level emissions targets.
- As a business owner / resident of York, I would like to see emissions from traffic reduced, so that York has healthier air quality and a more pleasant environment.

2.3 Requirements

This section outlines the requirements that the completed system will be expected to fulfil. Functional and non-functional system requirements are mostly derived from the user stories, with some notable additions such as security requirements. These are then used to derive a number of technical requirements, against which the final system can be evaluated.

2.3.1 Functional System Requirements

Ref	System Requirement	Justification
SR1	The system **must** calculate a dynamic congestion price which: • is proportional to the current traffic level • applies proportionate penalties to driving when York is exceeding its legal street level emissions limits • applies proportionate penalties to driving when and where pedestrian footfall is heavy or essential to business/events • does not discourage driving when it is not necessary to do so	The primary aims of the system are to disincentivise contributing to congestion, avoid illegal/unhealthy emissions levels, and improve the accessibility to York for pedestrians. At the same time, to demonstrate an improvement over static congestion charges, the system must not inconvenience road users who are not contributing to these problems.
SR2	The system **must** apply the dynamic congestion charge to drivers automatically.	In order for the system to be effective, it must be easy for drivers to use, and should cause minimum disruption on the road.
SR3	The system **must** make clear to drivers the charges that they will incur driving through a given zone at a given time.	For the system to have an impact on users' driving habits, they must be aware of the congestion charges they will have to pay.
SR4	The dynamic congestion charge **should** always be reasonable enough so as not to prevent essential travel.	If the congestion charge is too high, it could be too effective at discouraging travel into York, adversely affecting commerce and businesses.
SR5	Users **should** be able to book "passes" in advance for travel at a given date/time to "lock in" a price.	This allows users to accurately budget for future trips to York (particularly relevant to businesses with travel expenses).

Table 1: System Requirements

2.3.2 Non-Functional System Requirements

Ref	Non Functional Requirement	Justification
NFR1	The system's handling of personal data **must** be compliant with GDPR.	The system must comply with all relevant legal and ethical standards regarding personal data.
NFR2	The system **must** be cyber-secure.	The system may contain sensitive information, such as users' payment details.
NFR3	The system **must** provide tangible benefit to businesses in York.	Local businesses are a primary stakeholder in this project.

Ref	Non Functional Requirement	Justification
NFR4	The system **must** provide tangible environmental benefit to York.	The purpose of this project is to significantly reduce York's environmental footprint.
NFR5	The system **should** provide tangible benefit to residents and visitors to York.	Residents and visitors are secondary stakeholders in this project.

Table 2: Non-Functional Requirements

2.3.3 Technical Requirements

Ref	System Requirement	Derived from
TR1	The system **must** increase the dynamic congestion charge when and where traffic is likely to be high and vice versa	SR1
TR2	The system **must** increase the dynamic congestion charge when and where footfall is likely to be high and vice versa	SR1
TR3	The system **must** increase the dynamic congestion charge when and where York's emissions measurement stations indicate legal limits may be exceeded	SR1
TR4	The system **must** integrate with the network of ANPR cameras around York	SR2
TR5	The frontend **must** clearly and unambiguously display current congestion prices for each zone	SR3
TR6	The system **must** predict congestion prices 28 days in advance	SR3
TR7	The frontend **should** clearly and unambiguously display future congestion price predictions	SR3
TR8	Electronic signage at entry points to zones in York **must** display prices for entering a given zone to drivers	SR3
TR9	The frontend **should** allow users to book "passes" in advance for travel at a given time/date to "lock in" a price to "lock in" the predicted price referenced in TR6	SR5
TR10	The congestion charge for each zone **should** be capped at a level defined by the council	SR4
TR11	The maximum amount someone can be charged in a day **should** be capped at a level defined by the council	SR4
TR12	All historic traffic data **must** be held in an anonymised form	NFR1
TR13	If/where personal data is held, it **must** be encrypted	NFR1, NFR2

3 System Overview

3.1 Architecture

The system will consist of three services; a variety of frontends, an API, and a backend. The API and backend will share a database with the use of virtual container volumes. This allows the API and backend to be completely separated, meaning that when API endpoints are called, no extra computational load is placed on the backend.

Each service will utilise virtual containerisation for a variety of reasons. Firstly, deployment is made easier as the containers can be run on almost all hardware. Secondly, separating out the system into services allows the software engineers to split into teams, each one assigned a service. In addition, this design allows additional frontends to be developed in the future, such as a mobile app, without having to update the backend.

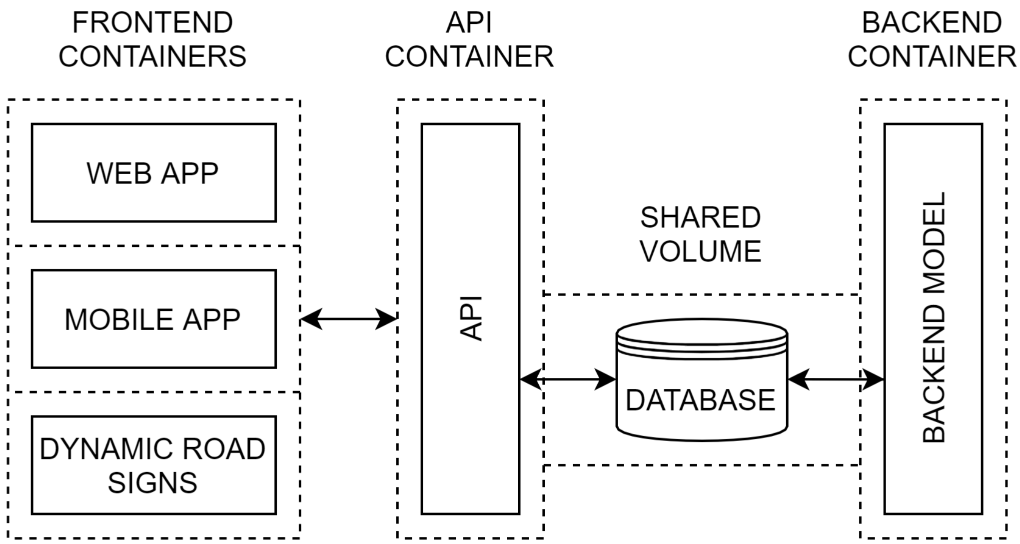


Figure 1: An overview of the services that make up the system

The frontend will display congestion charge information to the user, in addition to other features such route planning, pass purchasing and obtaining event details. To do this, both internal and external API calls will be made; internal to gather and submit data specific to the congestion charge system, and external to make use of graphical widgets such as Google Maps [10]. The internal API will handle data in a JSON format [11], and will consist of a variety of endpoints, each one either retrieving data from or inserting data into the database.

The backend service is where all of the dynamic congestion charges will be calculated, and will consist of a non-terminating application. On the hour, the backend will ”tick”, triggering computations that include the prediction of future data (footfall, emissions levels and traffic counts) and the calculation of congestion charges for each zone at certain dates and times. The results of these calculations are then inserted into the database.

3.2 Data Sets

Availability of data is vital to making York a Smart City. Data must be kept current in order for machine learning models to predict future values accurately.

For each zone, hourly records of footfall, vehicle emissions levels and traffic counts are required. Footfall records are required to monitor pedestrian traffic levels. Vehicle emissions should be recorded to identify when safety standards may be being breached. Knowing the traffic count at any given time allows an appropriate congestion charge to be set.

Footfall per hour is already gathered by City of York Council [12], but this is done at the street level rather than a zonal level, and only the four busiest streets are monitored. Similarly to footfall, for each hour of the day Air Quality England releases emissions data in the form of air quality measurements for the city of York [13], with most measurement stations being located towards the centre of the city. Traffic data in York is currently recorded [14], however, it is not available in an hourly format as required.

In addition to these three mandatory data sets, other data sets may be used to help machine learning models predict future values. One such data set is the weather forecast, as weather almost certainly impacts footfall and may have an effect on traffic levels and emissions. The Twitter API [15] may also be used to try and identify spontaneous events such as traffic accidents as they occur. A demographic data set could also be used to gain an insight into the population density in each zone, another factor that impacts footfall and traffic counts.

3.3 Frontend Applications

As with any system operating in the real world, it is important that users have access to information about the system so that they can make informed decisions. To this end, the information that the system holds needs to be easily accessible through several different frontends.

The most simple, yet effective of these frontends can be using physical enhancements to the infrastructure of the city (such as networked LED signposts and motorway information boards) to display the prices for York city as a whole or the different zones that the user might be about to enter. At its simplest these LED billboards could be bundled with the camera infrastructure previously described.

For more sophisticated presentations of the systems data, it would be appropriate to create and distribute web and mobile applications. The advanced capabilities these platforms provide for creating user interfaces allow users to better understand the current traffic environment of York, plan future journeys and interact with the different business models utilised by the system.

An example user flow diagram for a web application is demonstrated in Figure 2.

3.4 Backend Model Behaviour

3.4.1 Pricing Model

The ”brain” of the system is a complex model that provides a congestion charge price per zone in order to manage the traffic levels throughout York.

Each zone will be assigned a target traffic level for any given date and hour (i.e. the ideal number of vehicles driving within the zone at that time), which is dynamically calculated. Many parameters are used to calculate the target traffic level, including time of day, day of

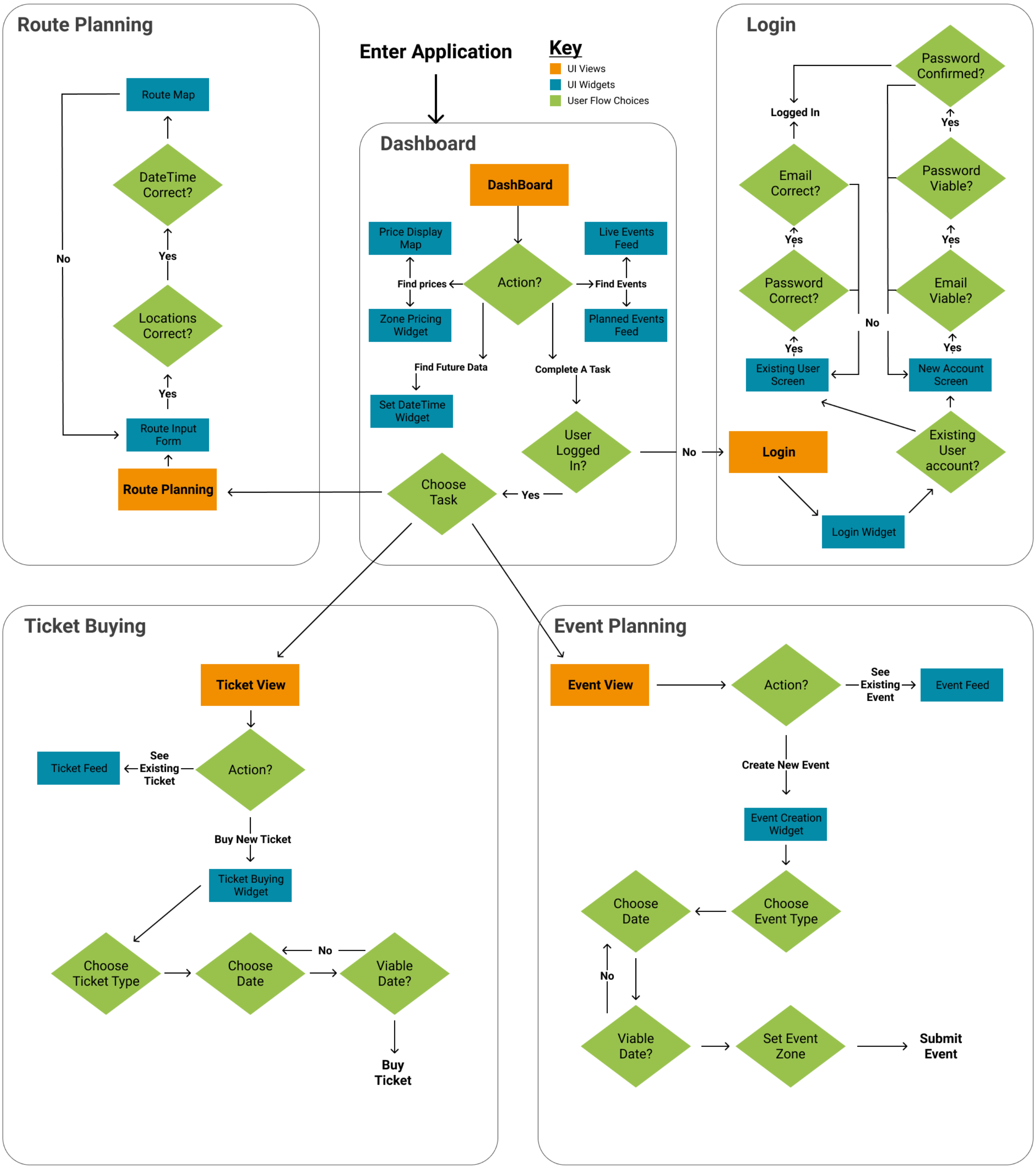


Figure 2: User flow diagram

the week, seasonality effects (such as bank holidays), weather, population density and retail density. A combination of machine learning models and fixed mathematical formulas will take these parameters as input and return a target level.

Target traffic levels can also be adjusted using the predicted footfall and emissions levels. If the predicted emissions levels exceed the council-defined limit then the target traffic level is significantly reduced to compensate for this. If footfall is expected to be relatively high (such as during the tourist season), then once again the target traffic level is reduced as to allow high numbers of pedestrians to roam the zone.

Events defined by the council also have an impact on the price of the zone they are assigned to during the course they are active. Events have a modifier set by the council, and the calculated price of a zone is multiplied by this modifier when the event is ongoing. This means events can be set to either make prices cheaper or more expensive. In addition, this means that when training the machine learning models used to predict future data set values, historical data recorded during the period of an event can be excluded from the training data, so as not to confuse the model.

3.4.2 Future Pricing

Whenever the use of cutting-edge technology is proposed, the system architects must ask whether the costs associated with implementing the technology are justified by the benefits. For the York dynamic congestion charge, machine learning provides enhancements to the predictive capabilities of the system leading to higher quality price forecasts. This in turn allows users to more effectively factor the York dynamic congestion charge into their daily routines. To accomplish this, the predictive model for the system will be built piecemeal upon the historic data sets for the known quantities that the system reacts to (traffic, footfall, emissions etc).

Each data set will be used to train a variety of models. From this group, the best performing models can be selected for the basis of a stacked meta-model, which will provide the definitive prediction for the data set. Once the meta-model predictions have been provided, the usual reactive algorithm can be applied to obtain the predicted price for any given zone when subjected to the predicted data points.

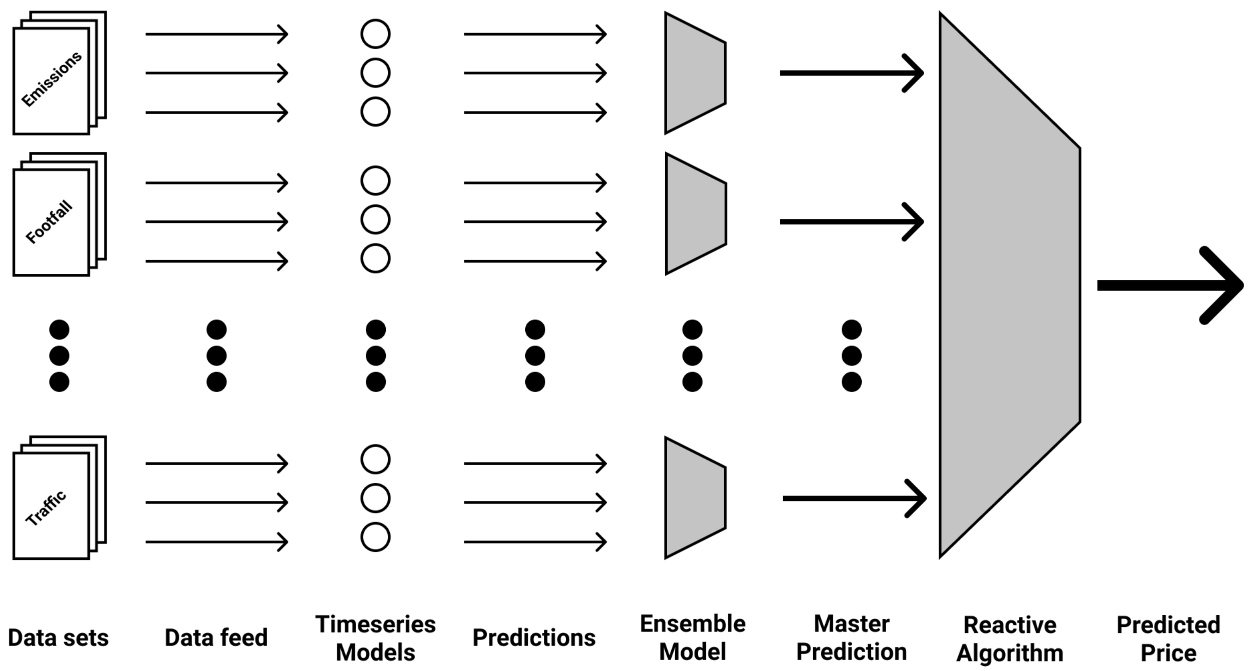


Figure 3: Machine learning model architecture

The data sets that form the basis of the models will contain historic data for the systems inputs, whether that be NO2 emissions, footfall, traffic data or any new metric that is included in the system. It should be noted that once the system is deployed, it will affect the urban environment of York, making the historic data less relevant for predicting current conditions. Therefore the models will need to be retrained on newer data from after the implementation of the system.

The models to be used for prediction will need to be based on ML techniques suitable for time-series data, as standard techniques like linear regression don't account for seasonal changes (e.g. more emissions in winter) or variation across days and weeks (e.g. lighter traffic at midnight on Sunday). Modelling techniques suitable for time series data include auto-regression [16, sec 3.2], moving average [16, sec 3.3], SARIMA [16, sec 4] and variations on neural networks.

By using an intuitive and hierarchical grid search approach, many machine learning techniques can be applied to the data sets. Once the best performing techniques have been identified, a hyper parameter search can be undertaken to establish the best models.

With three or more high performing models being established for each data set, a meta-model can then be created that balances the individual predictions through a process known as ensemble modelling [17]. Assuming that the individual high performing models are better than random guessing, a meta-model can utilise these predictions to reduce the variance and increase the accuracy of the overall master prediction.

Once the models and meta-models have been trained, a specific date and time can be applied to the model to predict future data set values that can be passed to the congestion charge algorithm. This will then generate the predicted price for that date and time which will be an accurate representation assuming that the predictions for York's environment hold true.

In comparison to other techniques, this method will give the highest quality predictions while taking into account the wide variety of seasonal effects time will have on the data. While it may at first seem like reinforcement learning (another ML technique) would respond better to the effects of the York dynamic congestion charge system on its own environment, this comes at the expense of either complexity (in terms of time to become efficient, and model size) or the inability for the model to "remember" the drastic changes between midnight and midday, weekdays and weekends or summer and winter. However, with new advances in deep neural networks for time series analysis there is the potential for moving towards such a system as the research matures and progresses [18].

To aid with predictions, additional data such as weather forecasts will be fed into the machine learning models, but only where appropriate. For instance, only weather forecasts for the next week may be considered accurate and therefore when predicting values over a week away the weather will not be considered. The algorithm will also take into account the number of passes that have been bought for the corresponding zone that would be active at the time.

Prices will be initially predicted 28 days in advance, and revised closer to the time. Prices will only ever be predicted or revised on the tick of the backend model, which occurs on the hour, every hour. After a price is initially predicted, it will be revised weekly until 7 days prior, and thereafter revised daily. For the final 24 hours it will be revised hourly. The amount by which the price may change decreases with each revision, such that fluctuations will become smaller as the date approaches, giving road users greater confidence in the accuracy of the prices.

3.5 Database

Figure 4 shows the tables required in the SQL database of the completed system. York will be split into zones and so storing basic information about these zones, including a list of coordinates marking their geographical location, is required. Accounts are required to distinguish between road users who have purchased passes, and those that have not. A pass will have a start and end date and time, and is active over a set of zones. Events are also stored in the database, and are similar in the sense they have a start and end date and time, however, they are tied to a single zone.

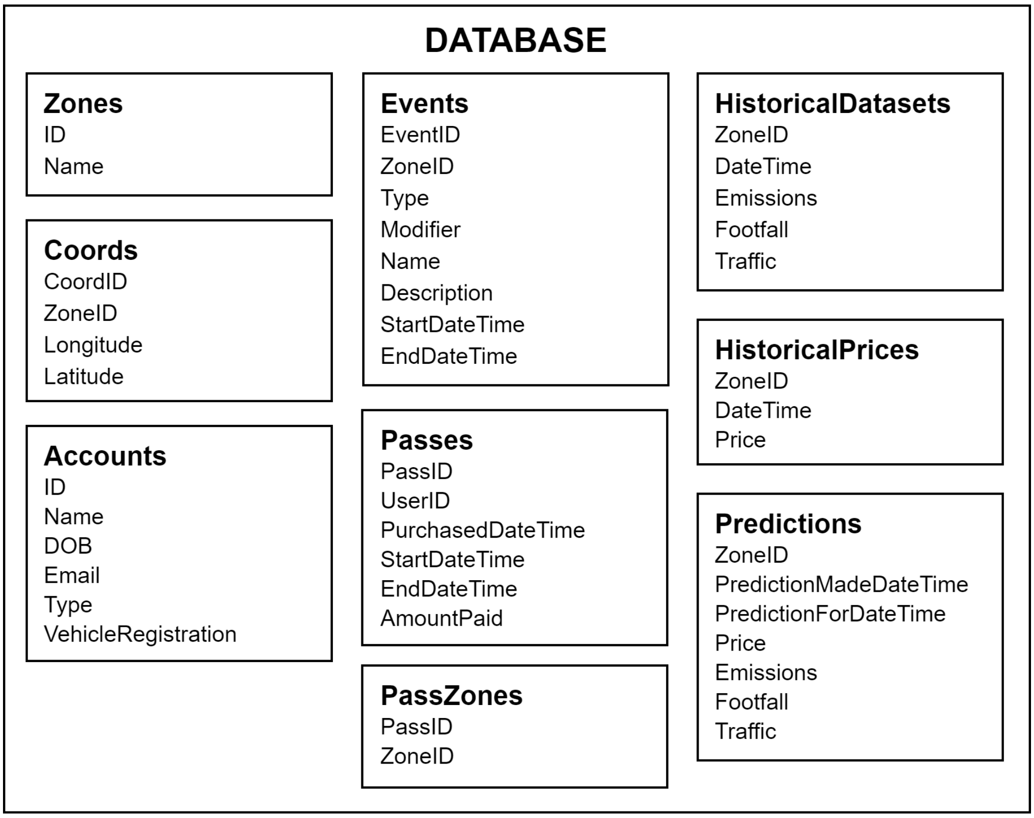


Figure 4: Smallest viable database required for the deployed system

Any data recorded from the network of sensors will be stored in the database and can be obtained by querying a zone, date and time. Putting this data in a single table reduces the number of database queries required by the machine learning models. Predicted data set values are stored in the database as to allow machine learning models to look at the difference between the predicted and actual (historical) values, allowing them to calculate errors and learn to be more accurate. Storing predicted prices allows the API to supply them to the frontend.

3.6 Physical Infrastructure Investment

To implement the system by 2025, substantial infrastructure investments would need to be made by City of York Council. Firstly, additional footfall cameras would need to be set up along the major streets of each zone. There would also need to be at least one air quality measurement station per zone, although multiple would be ideal. All of the sensors described would networked into the system.

The existing ANPR [19] network of cameras in York [20] would need to be extended to match, so that traffic counts can be recorded. The ANPR cameras will have an additional purpose of tracking vehicle registrations in real time. This allows the system to check whether a driver has already paid for a pass, whether they're exempt from a charge (such as driving council vehicles) and if not, bill them (via the DVLA [21]).

Finally, dynamic congestion signs would need to be set up along the borders of zones. These signs would show the current price of all zones, in addition to the next few hours worth of predicted prices.

3.7 Security

The security of any system capable of causing harm [22] to it's users if improperly utilised must be considered a top priority for any system architect.

The frontend applications have several features that must be securely implemented. For example, if the ticket purchasing service is not securely implemented, software bugs may lead to the non-payment for legitimate tickets, duplicate ticket purchases, illegitimate purchases being recorded as well as the dissemination of the private financial data required for users to utilise the service.

A good strategy to combat these issues is to avoid implementing a custom payment processor and instead utilise the services of Stripe [23] or Paypal [24], who provide API's that would allow the frontend team to implement a payment system quickly while outsourcing the considerations for financial regulations and financial data security to these companies. Every transaction that's made on these API's also returns a unique transaction ID, which could be coupled with every Database ticket purchase protecting against illegitimate or duplicate tickets.

If the event booking service is not securely implemented, malicious actors could infiltrate the system and use the service to manipulate the logistics network of York - leading to widespread economic damage in the city. Ideally, the only people that would be able to access the events system would be those in the council - therefore, techniques like encryption, IP address white listing and standard user authentication could be used to ensure that only council staff can add or remove events from the system.

The security of the database is of paramount concern for this system, as the database contains sensitive information in order to facilitate the sending of congestion charges to a user (this could be their address, online account or financial data) and information that allows the tracking of a vehicle around York city (opening up privacy concerns). Therefore, the database needs to be secured with modern security techniques including the encryption of sensitive information so that it is not stored plain-text in the database, the explicit white listing of networks that need to access the database, background checks of all system administration personnel who have access to the database as well as the hierarchical organisation of database access privileges so information is restricted to a need to know basis [25].

Software considerations for integration's into the system are important, as the open access to data required in order to facilitate some of these integration's also comes with security risks that the integration's can maliciously access private data. However, since the system is not a platform - the best defence against malicious intentions is to vet each integration carefully and set up unique and tailored permissions for each integration. For instance, the route planning integration would not need direct access to the database and can operate effectively utilising the API and frontend services - which would then send only the necessary information (origin/destination/time) for a user to receive their calculated route.

The use of physical infrastructure presents an additional set of concerns. For instance, all sensors used to capture footfall, emissions levels and traffic counts must send their data to the backend service in real time. Some of this data won't be anonymised, such as number plate tracking required for billing road users, and therefore strict cybersecurity measures must be put in place. Simple controls such as data encryption will be implemented, both at the application layer (RSA [26]) and transport layer (SSL [27]).

Finally, by decoupling all of these services using a micro services structure we can avoid introducing a single point of failure into the system. For instance, by separating out a front end application we can ensure that if the application was ever hit by a denial of service attack it shouldn't affect the backend service and it's ability to process traffic movements and the subsequent congestion charges.

3.8 Route Planner Integration

An integration with the route planning software of another team is planned. This integration would allow individuals to calculate the *best* route through York from one location to another, taking into account congestion charge. What constitutes the *best* route will be defined by the user depending on what is important to them; cheapest or fastest route for a given date and time, route with the least zone border crossings and cheapest time of day are all valid examples of what users may be looking for. Based on the user's choices a multi-objective fitness function will be defined and solved using optimisation algorithms. This route planner would be totally unique in adding predicted congestion and price as additional dimensions in the route planning optimisation.

3.9 Business Model

Frontend users will be able to create an account, in which they register their vehicle and specify the type of road user they are (personal or business use). By creating an account the user can specify how they wished to be charged, by potentially setting up a Direct Debit or topping up their balance. If they are in debt to the system, they will also have a means of paying off the amount owed to the council.

Accounts have additional benefits, such as the ability to purchase passes. Passes are active through a collection of zones, specified by the pass buyer, along with a start and end date and time. The purchase of passes should be encouraged as it helps aid future traffic predictions - the greater the proportion of users using passes, the more that is known about future traffic. In addition, registered commercial vehicles will receive a fixed discount on any passes purchased.

Road users who do not hold an account in the system, such as occasional visitors, would pay through a retrospective system much like that used by the Dartford Crossing [28]. In this system, users must pay online or by phone before midnight on the day of travel. Users who do not pay would have charges recovered by Penalty Charge Notices posted to the registered vehicle address using the DVLA system [21].

All charges, regardless of payment method, will be subject to a daily cap set by the council, so the maximum a road user may pay per day is known.

4 Prototype

4.1 Architecture

The software architecture of the prototype is designed with continuous deployment in mind. Each service (frontend, API and backend) is deployed in its own Docker [29] container, and can be configured to run on the same or separate hardware. This implementation matches the proposed architecture from Figure 1, with a single frontend.

The frontend is powered by Vue.js [30], a web app JavaScript framework that allows elements to be dynamically updated via API calls without the need to refresh or load additional web pages. This was chosen as it fits in with our architecture perfectly, with the frontend making calls to the API service.

The architecture of the backend is shown in Figure 8. The backend was designed so that the main model assigns prices to zones using outputs from the modular component models that predict traffic, footfall, and emissions data. All data used by the system is stored in an SQLite database [31] that implements the same structure as defined in Figure 4. Both the API container and backend container have access to the database using Docker shared volumes.

4.2 Data Sets

As detailed in Section 3.2, our system would utilise new and existing datasets to make smart decisions about congestion pricing. Since the data required is not available in all cases, the prototype uses simulated data based on real historic data and some assumptions.

Hourly footfall data from four measurement stations is published by York City Council [12]. These measurement stations are located on York's busiest retail streets - Coney Street, Parliament Street, and Stonegate (all located in the Guildhall ward), and Micklegate (in the Micklegate ward). Therefore, for the Guildhall and Micklegate wards, the prototype uses real footfall data from 2019. This is deemed acceptable as footfall outside of Guildhall and Micklegate is much lower and thus has little effect on calculated prices.

Hourly air quality data is published in much the same way as footfall data. Again, the measurement stations are more heavily concentrated in the centre of York, with some of our zones having none at all. Therefore, the prototype uses the data from the York Holgate [13] measurement station for all zones, again using data from the corresponding time and date in 2019.

Since the system has not been in operation, there is no bank of historic traffic data from which to make predictions. Therefore, the prototype makes traffic level predictions based on data from TomTom [32]. Data from York was not available, so data from the Leeds/Bradford conurbation is used instead. This data is used to establish the proportional traffic levels for different times and weekdays.

Finally, demographic data for each of the zones is used. Geographical data from MapIt is used for the zone borders [33]. Existing electoral ward boundaries are used to avoid confusion. For the prototype, only population and area are used. Population data was sourced from government data [34], and area data was sourced form MapIt [33]. This data is used in the calculation of maximum traffic level limits for each zone, which affect the price. Traffic level limits are lowered during night time for areas with higher population density.

4.3 Frontend Design

The Frontend of the system was designed to be lightweight and easy to use. Figma [35], a collaborative web-based design tool, was used to visualise the website before development began. A high-level hierarchical diagram of the frontend components is shown in Figure 5.

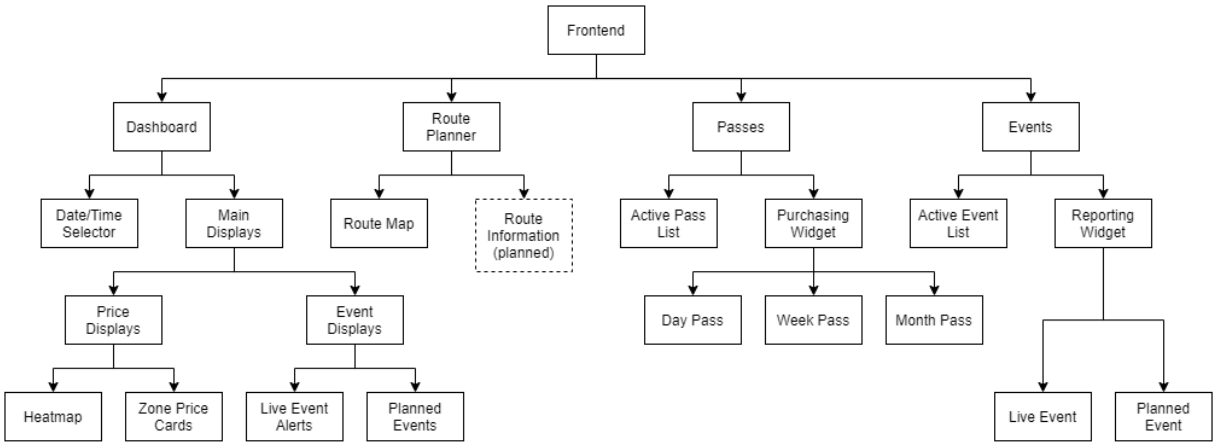


Figure 5: Hierarchical diagram of frontend design

The default view of the website is a main *Dashboard* view, which displays at-a-glance the relevant information a user might need to access quickly before a journey. The main feature of this view (Figure 6a) is a map overview showing the status of all the zones using a coloured overlay, with each zone colour-coded using the traffic light system, with deep reds representing the highest prices and bright greens the cheapest.

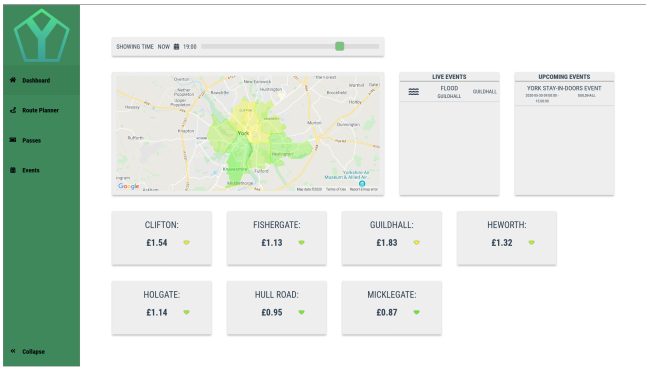
Below this map are the 'price cards', one for each zone, which contain an indicator of whether the price will increase or decrease in the next hour, allowing users to assess price trends at-a-glance.

In the top-right, feeds which display events have also been incorporated into the *Dashboard*, which show all planned and live events in the city, to alert users to events which may effect their journey.

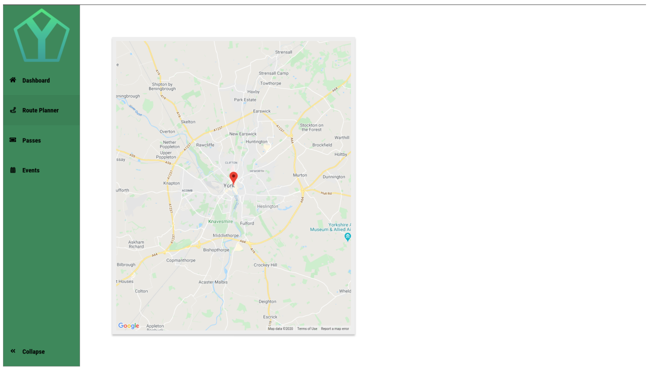
The other views are accessed through the use of a navigation bar along the left side of the app. This position was chosen as it is the standard position for a navigation bar, and therefore will be easily picked up by users.

The *Route Planner* view (Figure 6b) is a placeholder that was prepared for planned integration with route planning system developed by one of the other groups.

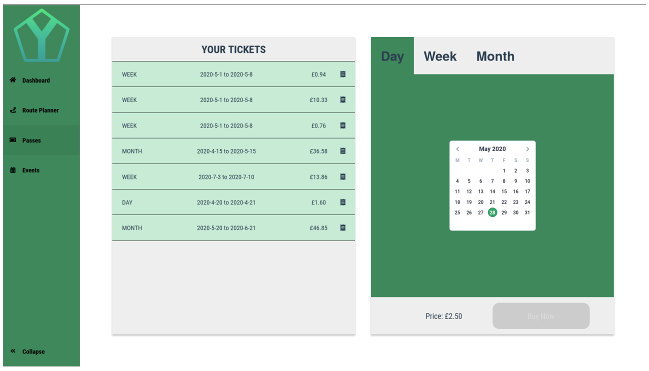
The final two views are the *Passes* (Figure 6c) and *Events* (Figure 6d) views. The processes of purchasing passes and registering events involve more user interaction, so they separated into their own views to avoid complicating the *Dashboard*. The *Passes* view features a feed which contains a users current passes that they have purchased, and a pass-buyer widget which can be used to purchase either a day, week or month pass using the calendar-style selector to choose the start date for the pass. The *Events* view is similar, and allows user to view and register future events which would affect traffic flow.



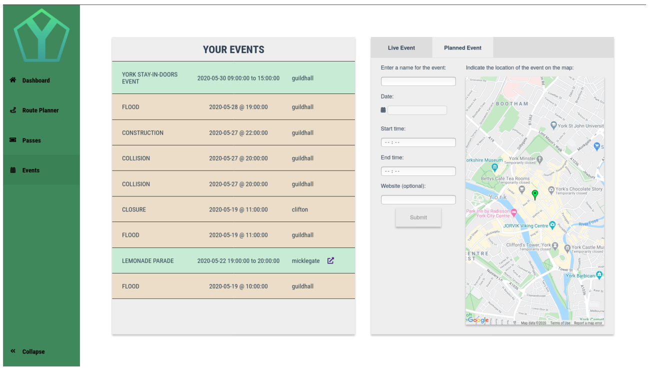
(a) Frontend *Dashboard* View



(b) Frontend *Route Planning* View



(c) Frontend *Passes* View



(d) Frontend *Events* View

Figure 6: Frontend screenshots

4.4 API Design

The endpoints of the implemented API is shown in Figure 7. The API was designed to be RESTful [36], an ideal property for such a system to inherit. RESTful API's are stateless and therefore any endpoint can be called at any time by any other system, allowing the multiple frontend vision of the system to materialise. All information related to a request is passed within that request, either as query string parameters or in the request body itself. All responses are in a JSON format [11].

4.5 Backend Design

Figure 8 describes the implemented architecture of the backend model in a UML format. Execution begins in the *Model* script which initially clears the database of all its content. Next, the demographic data from a specified directory is used to create *Zone* instances. These instances are contained within a single *City* instance, which differentiates between them by an ID it assigns to each one. The *Zone* instances are then inserted into the database.

Next, a simulation is run in which prices for each hour and zone are predicted up to 4 weeks in advance. As previously described, in order to make predictions on future prices, future footfall, emissions levels and traffic levels must be estimated. This is where the *Footfall-Handler*, *AirQualityHandler* and *TrafficHandler* objects are utilised. Given a specified date,

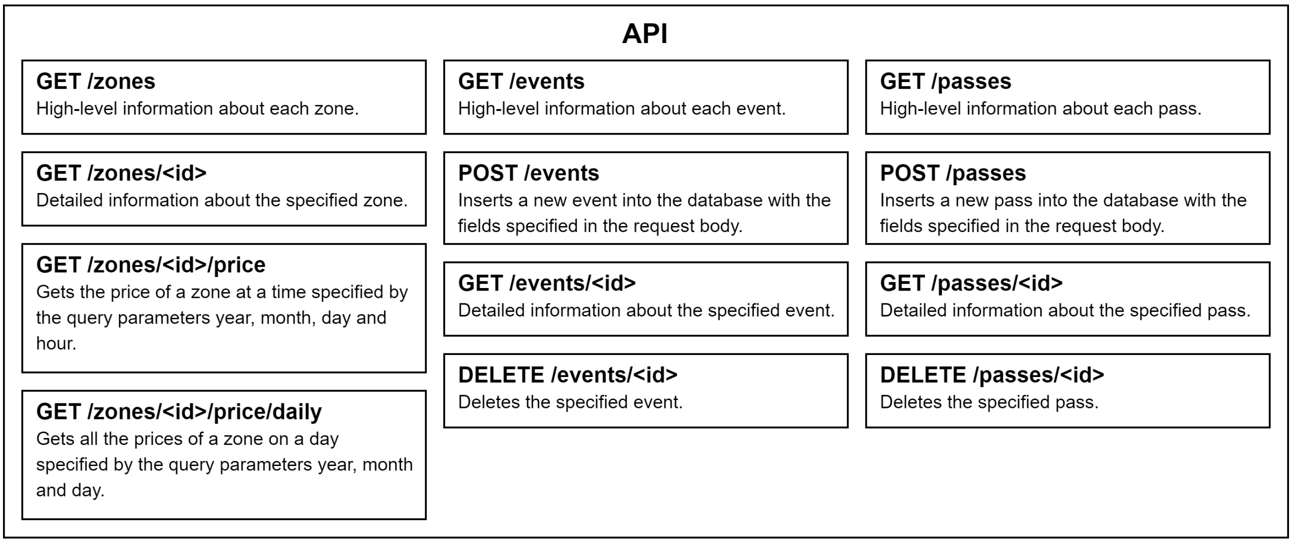


Figure 7: The implemented API endpoints

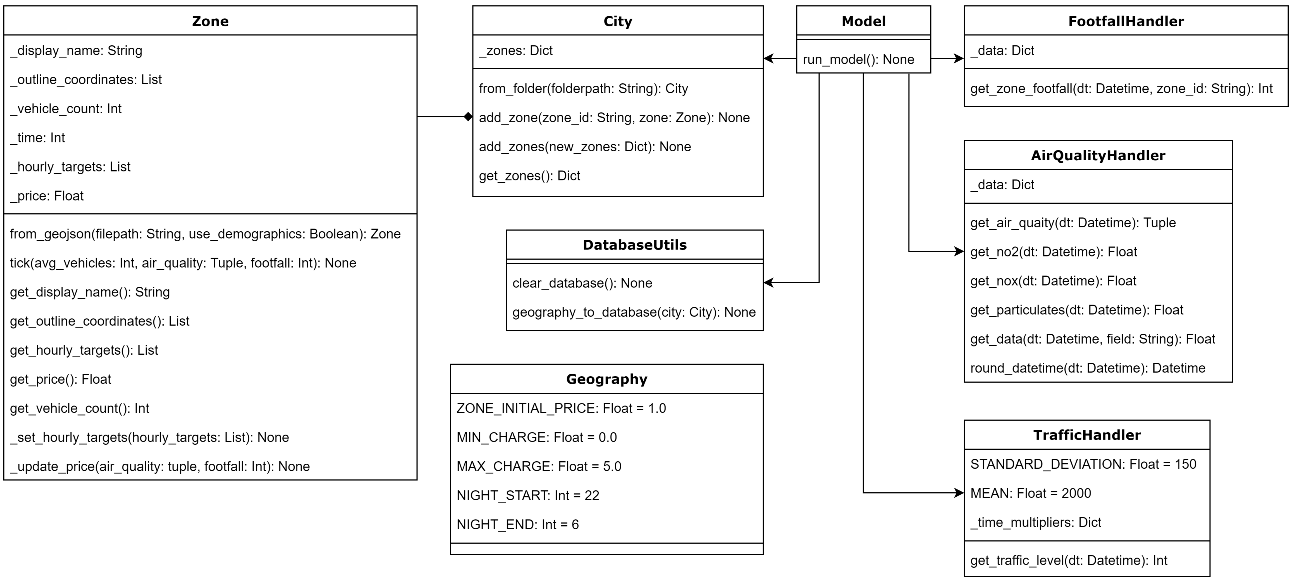


Figure 8: The architecture of the backend model as it is currently implemented

time and zone, they will return a predicted value for their appropriate data set. As previously mentioned, rather than implementing machine learning models to predict future values, data sets from 2019 were used where available instead. Therefore, all the handlers upon their instantiation load data from CSV files.

When requesting footfall from a specific zone, if the *FootfallHandler* does not have data for that zone then it will return a hard-coded value of footfall depending on the time of day. Only one air quality data set is used for the entirety of York in this prototype, therefore when retrieving this data specifying a zone isn't necessary. In order to make the data slightly different per zone, Gaussian noise is applied by *AirQualityHandler* upon each call of its get methods. This is deemed acceptable for a prototype as air quality only varies slightly within the city and this is a close enough approximation. Very similar logic is employed by *TrafficHandler*, where traffic is global and therefore not zone specific. The day of week and time of day correspond to a modifier from the proportional traffic level data set, which is multiplied by the *MEAN* constant. The reason for this is to scale the mean to establish a realistic traffic estimate for York. The scaled mean, along with the *STANDARD_DEVIATION* constant, forms another Gaussian distribution from which a value is selected and returned.

When a zone is asked to predict a price for a future date and time, it is passed the values returned by the handlers. The next step is to generate an appropriate target traffic level. A basic estimate is established based on the time of day and resident demographic data. During the day a fixed level is used, and in at night a level is chosen based on the demographic data; the higher the population density, the lower the target traffic level. This is discourage large volumes of traffic passing through residential zones at night.

The initial target traffic level is then tweaked based on the predicted values of the data sets. If any air quality figures are above council defined maximums then the target is reduced to compensate for this. Similarly, as footfall increases the target traffic is lowered (but capped at a predefined level to prevent it becoming unrealistically small).

To generate a congestion charge, the predicted traffic level is divided by the target traffic level and raised to a power (which is a float to allow fractional powers). If the council want a more aggressive charge then the power can be increased. This price is then capped in a range as to prevent road users getting unfairly large charges.

4.6 Evaluation

Distinct evaluation methods were used to evaluate the frontend and the backend against the relevant requirements. Test steps are traced to technical requirements, which are in turn traced to system requirements, which are derived from the user stories. Therefore, if all test steps pass, it can be said with confidence that each user story is resolved and therefore that the system is successful in its aims.

4.6.1 Frontend Evaluation

Evaluation of the frontend involved getting users who are not familiar with the system to perform several predefined tasks with the system to test its utility and usability. The prescribed tasks directly correspond to the user stories defined in Section 2.2. The time that each user took to accomplish each task was recorded to identify areas of the system that may be difficult for new users to navigate. The user tests were supervised by members of the development team to note how new users interact with the system, but users were not given any assistance or guidance on performing their tasks so that a true new-user experience could be observed.

Due to the COVID-19 pandemic, it was not possible to conduct a formal focus group to evaluate the frontend. Instead, the evaluation was performed with a limited number of user tests obtained from relatives of members of the development team. The tasks presented to users were as follows:

1. Find the current congestion charge in Clifton.
2. Find the congestion charge in Clifton at 8 AM tomorrow.
3. Find the time between 10 AM and 4 PM tomorrow when the congestion charge in Guildhall is the cheapest.

4. Purchase a pass for next week.
5. Report a collision on Bridge Street.

For Task 1, all users correctly assumed that the prices displayed when the webpage loads are the current prices and easily identified the price of a specific zone.

Task 2 was a slightly more involved task, and illuminated some usability concerns. While users were able to select the correct time using the slider after a brief period of experimentation, doing so proved to be more difficult when using a touch screen. Furthermore, it was not apparent to users that congestion charges changed only every hour. Both of these issues could be addressed by using more discrete time intervals on the time slider (such as by the hour, to reflect the update interval of the system) or by supporting an alternative, more explicit method for specifying the time. One user also had trouble changing the date, remarking that the calendar icon was too small and that its purpose was not clear. This could be solved by altering the design of the icon to be larger and more clearly interactive.

Users were able to perform Task 3 with the current system by gradually changing the time on the *Dashboard* while manually watching the price of Guildhall. However, some users first attempted to click on Guildhall's price card. This indicates that the current system misses the opportunity to provide additional utility by interacting with the price cards. For example, hovering the mouse pointer over a card could highlight the corresponding zone on the heat-map (solving an additional issue of distinguishing zones and their boundaries) and clicking on the zone could show a graph of its congestion charges over time. This task proved to be difficult using only the *Dashboard*'s utilities, but in the full system it should be more straightforward using the Route Planner (which, promisingly, some users attempted to use during this task.

For Tasks 4 and 5, users were able to intuitively identify and navigate to the view which would help them accomplish each task. Users were able to purchase a week pass for the correct dates, but were occasionally confused by the list of owned tickets (which, due to the incomplete nature of the system, displayed a list of all tickets in the system). Better separation between purchasing and viewing passes would likely simplify this task significantly.

Task 5 proved to be the most difficult task; all users were able to select the correct type of event, but they failed to select the location of the event for numerous reasons. Firstly, users expected to be able to select a location by simply clicking on the map, not by clicking and dragging on the pin icon. Furthermore, users had difficulty identifying the correct street. This is partly due to the users' lack of knowledge of the city and its layout, but that was not the only factor which obstructed the usability of the app. The map was too small, and Google Maps' landmark icons obstructed users' view of the streets. Finally, some users had trouble locating the instructions which accompanied the map, and those that did found them to be unhelpful. These issues could be resolved by allowing users to select a location by clicking on it, enlarging the map, reducing the number of landmarks shown, and providing more prominent and explicit instructions.

4.6.2 Backend Evaluation

It should be noted here that the backend software had already been extensively tested before completion of the prototype. For final deployment, acceptance testing was automated using a Python script which exercised the price-setting functionality of the backend to ensure it met technical requirements TR1-3 and TR10. This involved isolating each of the three factors which should affect pricing (traffic, pedestrian footfall, air quality) and verifying that they have the expected effect.

Table 4 describes the acceptance test script. Each verifiable outcome must be met in order for the corresponding technical requirement to be deemed to have passed. Traffic levels referred to in the test script refer to different times of day on the 26th May 2020 (the date is held constant for consistency). In the test steps related to TR2, the three times of day are arbitrarily chosen for particular traffic levels - traffic is particularly heavy at 8am, particularly low at 4am, and moderate at 12 noon.

Footfall levels refer to an integer number of people counted in a zone. Air quality levels are expressed as (A, B, C) where A is the NO2 level, B is the NOX level, and C is the level of carbon particulates.

Step	Description	Verifiable Outcome	Relevant Requirement(s)
1.1	Tick each zone object with 8am traffic, footfall=0, airquality=(0, 0, 0), record each zone price	None	N/A
1.2	As 1.1, except with 1pm traffic	None	N/A
1.3	As 1.1, except with 4am traffic	None	N/A
1.4	Compare zone prices from steps 1.1 to 1.3	For all zones, zone price from 1.1 >= zone price from 1.2 >= zone price from 1.3	**TR1**
2.1	Tick each zone object with 12 noon traffic, footfall=1000, airquality=(0, 0, 0), record each zone price	None	N/A
2.2	As 2.1, except with footfall=500	None	N/A
2.3	As 2.1, except with footfall=0	None	N/A
2.4	Compare zone prices from steps 2.1 to 2.3	For all zones, zone price from 2.1 >= zone price from 2.2 >= zone price from 2.3	**TR2**
3.1	Tick each zone object with 12 noon traffic, footfall=0, airquality=(50.0, 60.0, 60.0), record each zone price	None	N/A
3.2	As 2.1, except with airquality=(40.0, 50.0, 50.0)	None	N/A
3.3	As 2.1, except with airquality=(30.0, 40.0, 40.0)	None	N/A
3.4	Compare zone prices from steps 3.1 to 3.3	For all zones, zone price from 3.1 >= zone price from 3.2 >= zone price from 3.3	**TR3**

Step	Description	Verifiable Outcome	Relevant Requirement(s)
4.1	Tick each zone object with 8am traffic, footfall=2000, airquality=(100.0, 100.0, 100.0)	All zone prices set to maximum	**TR10**
4.2	Tick each zone object with 4am traffic, footfall=0, airquality=(0.0, 0.0, 0.0)	All zone prices set to minimum	**TR10**

Table 4: Acceptance test script

All of the verifiable outcomes were as expected so this test was deemed to have passed, and therefore it can be concluded with some confidence that TR1, TR2, TR3, and TR10 are met.

It is acknowledged that this acceptance test is by no means exhaustive, but it was deemed that in conjunction with the tests performed during development, there is enough evidence to say with confidence that the backend performs well enough for a prototype. Clearly though, more rigorous testing would be required for deployment of the backend of a full system.

No attempt was made to verify the following four technical requirements for the reasons outlined below.

- TR4 ("The system must integrate with the network of ANPR cameras around York") was not verified as the integration with York's ANPR cameras was not implemented in the prototype, so this requirement is out of scope for the prototype.
- TR11 ("The maximum amount someone can be charged in a day should be capped at a sensible amount") was not verified as the prototype did not model movement of individual cars, only overall traffic trends. TR11 is therefore out of scope for the prototype.
- TR12 ("All historic traffic data must be held in an anonymised form") was not verified by test as it is trivial to argue that it is satisfied, since the traffic data used in the prototype is constructed from TomTom data which is already anonymised [32].
- TR13 ("If/where personal data is held, it must be encrypted") was not verified as no personal data is held by the prototype, so again this requirement is out of scope.

In addition to the testing against requirements, a basic stress test was conducted to determine whether the prototype was stable to run for long periods of time. It should be noted that this stress test did not verify the ability of the prototype to withstand heavy load, just that it remained stable over time. The plan was to deploy the system for a week and periodically exercise all features in order to check the performance is still as expected. This test was deemed to pass, and in fact the prototype has remained deployed ever since, and at the time of writing all functionality still works as expected. It is therefore reasonable to conclude that the prototype is stable.

4.7 Road To Deployment

The prototype has several limitations that must be addressed before it could be deployed in practice. Currently, the models which predict future footfall, traffic, and congestion levels are heavily and naively reliant on historic data for the requested time and location. Ideally, these models would be enhanced with machine learning so that predictions could be refined over time.

The system would need a method of user authentication, to allow user sign on and purchase of passes, which would also need to be modified so that a pass is tied to a single account. Additionally, a feature allowing a user to register a vehicle to an account would need to be implemented so that a vehicle is not automatically charged when the account tied to that specific vehicle owns a pass for that area and day. Further, the prototype is not integrated with an e-commerce service to process payments for passes and charges reliably and securely. The prototype's lack of authentication also means that any user is allowed to register events with the system. To prevent malicious use of this feature, access to it must be restricted to the city council.

Furthermore, the physical infrastructure would need to be fully integrated with the system and missing physical infrastructure would need to be deployed where needed. This includes sensors which capture data such as footfall, emissions and missing traffic sensors, an extensive array of ANPR cameras which are used to detect cars as they enter and leave zones, and integration with existing road signs as well as new road signs to be able to display live congestion charges and events to road users. The cost of this extensive deployment of infrastructure is potentially very expensive.

5 Market Assessment

This chapter outlines how a dynamic congestion charge will benefit businesses in York, and why we believe it is viable to introduce one in York by 2025.

5.1 Benefits to Local Businesses

An intelligent, dynamic congestion charge would be a major benefit to local businesses in York, particularly given its largely service- and tourism-oriented economy; Tourism and retail are the largest sectors of York's economy, contributing 13% and 16% respectively [6].

Since the introduction of Stockholm's semi-dynamic congestion charge, the city-centre retail sector has seen a 6% increase in business [37] [38], largely due to the reduced traffic in shopping areas and reduced journey times. Increased accessibility for pedestrians would also make York more attractive to tourists and outside shopping visitors, which would be valuable for key sectors of York's economy.

York is visited by 7 million tourists per year [6]. Notably, very few visitors to York use cars - 89% of visitors walk around the city, and 88% do not use a car at all [6]. The compact size and proximity of York's main attractions is cited as a positive factor by visitors, as is the city's "ambience" [6]. Clearly then, visitors prefer to be able to walk around the city and "take it in", and York should capitalise on this. A reduction in traffic from a congestion charge would increase accessibility to pedestrians, and make the city a more pleasant environment to walk in, potentially increasing York's advantage in tourism.

5.2 Project 2025

Congestion charges carry the stereotype of being unpopular with drivers. Elected councils are sure to be held accountable to transport decisions. To be viable in the long term, the dynamic congestion charge would need to win popular support to survive changes in council administration. Evidence from cities where congestion charges have been implemented, such as London, suggests that they become more popular in the long term, once the effects can be seen retrospectively [39]. However, this project needs to be politically viable by 2025.

One such method for gaining public support could be an initial trial period followed by a referendum, as used successfully by Stockholm [4]. It was easier to gain political support for a trial than a permanent implementation. The trial, running from September 2005 to December 2006 was hugely successful at demonstrating to people the benefits of the system, with public attitudes shifting from 55% believing it to be a "very or rather bad idea" to ultimately winning the referendum on permanent implementation [4] [5]. Attitudes to Stockholm's congestion charge have continuously become more positive since. The initial referendum was won with a 52% yes vote, but 1 year on in 2007 showed 66% support, and 5 years on in 2011, public support was at 74% with 18% even wanting to increase the charges [5].

Technically, the dynamic congestion charge would not be challenging to integrate with York's infrastructure. In London, ANPR cameras are used to detect ingress and egress to the congestion charge zone [3]. York already has a network of ANPR cameras across the city [20] which could be used to detect movement within and between different zones, and this network could be added to with additional cameras at zone boundaries where they are not already in place. Systems for advance and retrospective payment of charges as detected by ANPR are already implemented elsewhere in the UK for the London Congestion Charge [40] and the Dartford Tunnel [28], with mechanisms for recovering payment from fare-dodgers using the same council systems as parking fines.

As discussed Section 3.6, the only other required physical infrastructure is emissions measurement and footfall measurement. As with ANPR, this infrastructure already exists in some locations in York; York's network simply needs to be extended to cover all of our zones.

5.3 Expansion into markets beyond York

Cities across the UK are seeking more radical solutions to problems of congestion and air quality. Currently several are planning "Clean Air Zones" (CAZs), and indeed Birmingham, Leeds, Nottingham, Derby, and Southampton have been mandated by the government to introduce CAZs [41]. Manchester and Edinburgh each recently considered static congestion charges as used in London [42] [43], and Manchester has been told it must implement a CAZ unless an alternative can be identified [44]. A successful roll-out of the Dynamic Congestion Charge in York could lead to Manchester and Edinburgh considering them.

Perhaps the market where the Dynamic Congestion Charge could be adopted the quickest would be Durham, which has probably the most similar system in the UK. Durham has a single "Charge Zone" in the centre, which applies charges Monday to Saturday between 10 and 4pm using ANPR cameras [45], in exactly the same fashion as our system. Since Durham and York have similar economic and human geographical characteristics, it is probable that success in York would generalise to Durham.

6 Development Summary

6.1 Agile Software Development Lifecycle

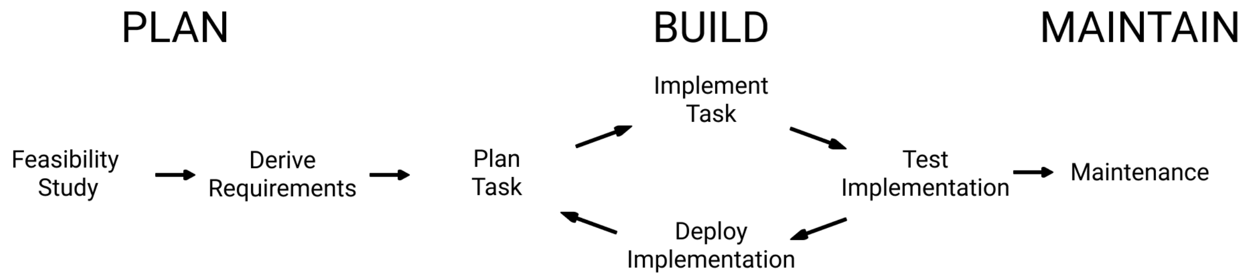


Figure 9: Agile development diagram

For this project, the Agile software development methodology was chosen. The methodology itself consists roughly of the same stages as the traditional software development life cycle, but executes the design/planning, development, testing and deployment phases in a iterative cycle rather than the linear progression of the waterfall methodology. The benefits of the Agile methodology include the reduction of planning costs (as planning can be done in smaller units), the emphasis on acceptability testing, and the ability to demonstrate a unfinished version of the product to the clientele earlier in the product development life cycle.

The feasibility study and requirement planning phases of the project are undertaken linearly, with the group analysing existing systems in the domain as well as good software engineering practises to derive requirements (see section 2.3).

Once requirements are derived, work can begin utilising the iterative process of the design, development, testing and deployment loops. One requirement is taken and planning begins of breaking the requirement down into it's component tasks, these tasks are then assigned to different team members who design for the task, implement the design, test that design and then submit the task for deployment.

Every task involves these four phases, and once complete, every task is deployed to the product. This product can then be shown to clientele or project supervisors who can recommended changes to features that weren't implemented to the expectations of the client. This ensures constant evaluation of the product and avoids the need for sweeping changes that might occur in projects without a consistent dialog, a process that would waste a substantial amount of effort.

This cycle can continue until all features are developed and the project moves into the maintenance phase.

6.2 Group and Project Organisation

A risk register for the project is available in Table 5 in Appendix A.

Overall, the project progressed well with a strong body of work being produced for the frontend, API and backend services (including the integration between these different components) - as evidenced by the following final code metrics.

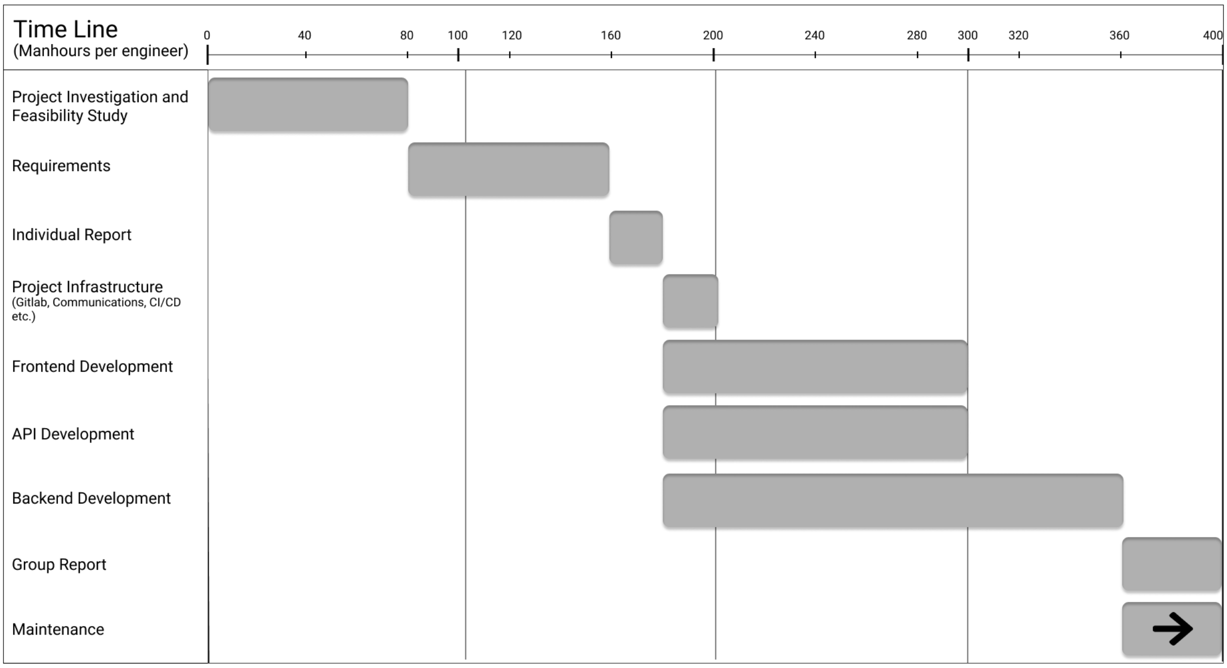


Figure 10: Project Gantt chart

Project	Lines Of Code	Merge Requests
Frontend	2751	28
API	780	7
Backend	574	5

Work was evenly distributed throughout the team, with no individual team member contributing less than 10% or more than 25% of the total merge requests (very reasonable considering the differing levels of effort required between requests).

Finally, all sections of work were progressed to a reasonable stage within their allotted time (as represented in figure 10), leading to the completion of the project as a whole with no 'crunch time' and no unnecessary stress.

6.2.1 Tooling

Communication and independence were priorities for the group when implementing the infrastructure for the project, in order to minimise risk.

Communication was handled through the online platforms Discord [46], Google Meet [47] and Gitlab [48], remote-first project structure would give team members the flexibility to work at their discretion, increasing productivity. This proved especially useful when the entire world switched to remote working. Discord provides a instant messaging server as well as VOIP capabilities, and was primarily used for day-to-day communication between group members, allowing for easy queries and informal design chats about different tasks. Google Meet was used for the twice-weekly team meeting, allowing the group to report and discuss the undertakings of the week as well as serving as a forum for requests between the frontend and backend teams. Gitlab was used to host the code repository, task manager and wiki of the project. This allowed for members to easily obtain the up-to-date code base, view tasks that still needed to be completed and upload their design documentation to a central authority.

Independence was achieved through tools provided by Gitlab. By utilising the task manager, groups members could claim tasks which fit their individual skill profile. It also ensured no team members were left idle waiting to be assigned tasks. Gitlab also provides ”merge request” functionality, which we utilised heavily. This ensured code could not be pushed to deployment without being reviewed by another team member, ensuring code quality as well as distributing knowledge of the code in case the original author was temporarily incapacitated (e.g. catching a contagious and novel virus). Additionally, Gitlab provides features for continuous integration and continuous deployment. Continuous integration allows the project to automatically verify the code that's pushed to production, while continuous deployment allowed the project to automatically push code to our production servers.

Finally, Overleaf [49] (an online multi-user Latex editor) was used to create and compile the report document.

6.2.2 Organisational Evaluation

Overall, the organisation techniques and tooling chosen proved vital to the progression and success of the project. Between the Agile management techniques adopted and the use of the powerful features provided by Gitlab, the team managed to consistently produce high quality code, leading to our finished product. No tools or organisational techniques were judged to be superfluous as clear value was demonstrated by each component of the project set up. The weekly meetings, Discord conversations and Gitlab task system provided a clear dialogue around how the project was progressing and which priority tasks needed to be tackled to propel it forward.

On review, no changes would be made to the existing development methods. However, while merge requests functioned well to protect the product and increase code quality in lieu of pair programming techniques, many team members found the occasional informal pair programming session to improve their understanding and ability to complete tasks present in the project. Upon starting a new project of a similar description, we would introduce capabilities to formally request pair programming sessions for difficult tasks or engineers working with an unfamiliar technology.

A Risk Register

Issue	Impact	Likelihood	Severity	Mitigation
Team member taken out of action	Loss of available manpower Loss of knowledge about existing code.	Probable	Medium	Add slack to project planning. Use code review techniques to ensure redundancy in code knowledge.
Deletion of code base	Complete loss of work up to the deletion event.	Low	Very High	Use a version control system to ensure redundant repositories of the code base. Utilise container software to keep a historical record of deployable versions of the code.
Unknown complexity of engineering task at hand	Project could result in an unfinished product	High	High	Utilise agile techniques to establish a complete minimum viable product that is then iterated upon, ensuring we also have something to demonstrate. Heavy research to understand complexities of engineering task at hand.
Team member confused about how to contribute work	Lost productivity that could add up over time	High	Low	Weekly meetings to establish problems and tasks for every team member. Project management system that clearly demonstrates the tasks that need completing as well as an intuitive system for completing them.
Poor software quality	Accumulation of technical debt, leading to lost productivity solving technical and communications issues	High	Medium	Use code review techniques to establish quality in code committed to the code base. Use continuous integration to deploy automatic code format checks and regression testing to ensure new code doesn't break the existing code base.

Table 5: Risk register

References

[1] New Atlas. (2006). Oslo to cut streetlight energy costs by 30% while increasing safety, [Online]. Available: https://newatlas.com/go/5475/ (visited on 05/18/2020).

[2] New York City DOT. (2012). Nyc dot announces expansion of midtown congestion management system, [Online]. Available: https://www1.nyc.gov/html/dot/html/pr2012/pr12_25.shtml (visited on 05/18/2020).

[3] Transport for London. (2020). What do i need to know about the central london congestion charge camera system? [Online]. Available: http://content.tfl.gov.uk/cc-cameras.pdf (visited on 05/19/2020).

[4] J. Eliasson, L. Hultkrantz, L. Nerhagen, and L. S. Rosqvist, "The stockholm congestion–charging trial 2006: Overview of effects," *Transportation Research Part A: Policy and Practice*, vol. 43, no. 3, pp. 240–250, 2009.

[5] M. Börjesson, J. Eliasson, M. B. Hugosson, and K. Brundell-Freij, "The stockholm congestion charges—5 years on. effects, acceptability and lessons learnt," *Transport Policy*, vol. 20, pp. 1–12, 2012.

[6] City of York Council. (2013). An overview of york's economy, [Online]. Available: https://democracy.york.gov.uk/documents/s106043/Annex%5C%203%5C%20An%5C%20Overview%5C%20of%5C%20Yorks%5C%20Economy.pdf (visited on 05/19/2020).

[7] ——, (2019). Air quality in york, [Online]. Available: http://jorair.co.uk/air-quality-in-york/ (visited on 05/19/2020).

[8] V. Thompson, "Revealed: 15 york locations breaching air pollution guidelines," *York Press*, Aug. 2018. [Online]. Available: https://www.yorkpress.co.uk/news/16397793.revealed-15-york-locations-breaching-air-pollution-guidelines/ (visited on 05/19/2020).

[9] City of York Council. (2020). Low emission strategy, [Online]. Available: https://www.york.gov.uk/air-pollution/low-emission-strategy (visited on 05/19/2020).

[10] Google. (2020). Google maps, [Online]. Available: https://www.google.co.uk/maps/.

[11] J. Friesen, "Introducing json," in *Java XML and JSON*, Springer, 2019, pp. 187–203.

[12] City of York Council. (2020). Footfall - footfall hourly - york open data, [Online]. Available: https://data.yorkopendata.org/dataset/footfall/resource/3db574a2-05ed-4ea4-83a4-5c0e59345c30.

[13] Air Quality England. (2020). York holgate latest data - air quality monitoring service, [Online]. Available: https://www.airqualityengland.co.uk/site/latest?site_id=YK8.

[14] Department for Transport. (2020). Road traffic statistics - local authority: York, [Online]. Available: https://roadtraffic.dft.gov.uk/local-authorities/202 (visited on 05/26/2020).

[15] Twitter, Inc. (2020). Developer, [Online]. Available: https://developer.twitter.com/en (visited on 05/30/2020).

[16] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[17] J. D. Wichard and M. Ogorzalek, "Time series prediction with ensemble models," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, IEEE, vol. 2, 2004, pp. 1625–1630.

[18] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[19] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (alpr): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2013.

[20] City of York Council. (2020). Bus lane enforcement, [Online]. Available: `https://www.york.gov.uk/BusLaneEnforcement` (visited on 05/19/2020).

[21] Driver and Vehicle Licensing Agency. (2020). Driver and vehicle licensing agency, [Online]. Available: `https://www.gov.uk/government/organisations/driver-and-vehicle-licensing-agency`.

[22] D. J. Solove and D. K. Citron, "Risk and anxiety: A theory of data-breach harms," *Tex. L. Rev.*, vol. 96, p. 737, 2017.

[23] Stripe. (2020). Stripe payments: Features, [Online]. Available: `https://stripe.com/gb/payments/features`.

[24] Paypal. (2020). Paypal developer documentation, [Online]. Available: `https://developer.paypal.com/home/`.

[25] E. Bertino and R. Sandhu, "Database security-concepts, approaches, and challenges," *IEEE Transactions on Dependable and secure computing*, vol. 2, no. 1, pp. 2–19, 2005.

[26] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[27] A. Freier, P. Karlton, and P. Kocher, "The secure sockets layer (ssl) protocol version 3.0," *IETF*, vol. 3, pp. 1–67, 2011.

[28] Department for Transport (DfT). (2020). Pay the dartford crossing charge (dart charge), [Online]. Available: `https://www.gov.uk/pay-dartford-crossing-charge` (visited on 05/19/2020).

[29] Docker Inc. (2020). Empowering app development for developers, [Online]. Available: `https://docker.com`.

[30] E. You. (2020). Vue.js - The Progressive JavaScript Framework, [Online]. Available: `https://vuejs.org/`.

[31] D. R. Hipp. (2020). Sqlite, [Online]. Available: `https://sqlite.org/index.html`.

[32] TomTom. (2020). Leeds-bradford traffic report, [Online]. Available: `https://www.tomtom.com/en_gb/traffic-index/leeds-bradford-traffic`.

[33] MapIt UK. (2020). Children of city of york council, [Online]. Available: `https://mapit.mysociety.org/area/2625/children.html`.

[34] City of York Council. (2019). York ward profiles 2018-19, [Online]. Available: `https://data.gov.uk/dataset/d7137c37-e43e-41de-bdc1-aa3a04aae9d3/york-ward-profiles-2018-19`.

[35] Figma. (2020). Figma: The collaborative interface design tool, [Online]. Available: `https://www.figma.com/`.

[36] R. Battle and E. Benson, "Bridging the semantic web and web 2.0 with representational state transfer (rest)," *Journal of Web Semantics*, vol. 6, no. 1, pp. 61–69, 2008.

[37] This Big City, *The success of stockholm's congestion pricing solution*, 2011. [Online]. Available: https://thisbigcity.net/the-success-of-stockholms-congestion-pricing-solution/.

[38] IBM Corporation. (2006). The stockholm congestion charging trial, [Online]. Available: http://www.stockholmsforsoket.se/upload/Infomaterial%20MAK/IMPACTS_IBM_v2.pdf.

[39] Centre for Public Impact. (Apr. 2016). London's congestion charge, [Online]. Available: https://www.centreforpublicimpact.org/case-study/demand-management-for-roads-in-london/ (visited on 05/19/2020).

[40] Transport for London. (2020). Pay to drive in london, [Online]. Available: https://tfl.gov.uk/modes/driving/pay-to-drive-in-london (visited on 05/19/2020).

[41] Fleet News. (2020). What are the proposed uk clean air zones (caz)? [Online]. Available: https://www.fleetnews.co.uk/fleet-faq/what-are-the-proposed-uk-clean-air-zones-caz (visited on 05/21/2020).

[42] D. Ottewell, "C-charge: A resounding 'no'," *Manchester Evening News*, Apr. 2010. [Online]. Available: https://www.manchestereveningnews.co.uk/news/greater-manchester-news/c-charge-a-resounding-no-976016 (visited on 05/21/2020).

[43] BBC News, "Edinburgh rejects congestion plan," *BBC News*, Feb. 2005. [Online]. Available: http://news.bbc.co.uk/1/hi/scotland/4287145.stm (visited on 05/21/2020).

[44] ——, "Greater manchester congestion charge: Mayor andy burnham rejects plan," *BBC News*, May 2017. [Online]. Available: https://www.bbc.co.uk/news/uk-england-manchester-39882454 (visited on 05/21/2020).

[45] Durham County Council. (2020). Durham road user charge zone (congestion charge), [Online]. Available: https://www.durham.gov.uk/article/3437/Durham-Road-User-Charge-Zone-congestion-charge (visited on 05/21/2020).

[46] Discord Inc. (2020). Discord homepage, [Online]. Available: https://discord.com/ (visited on 05/30/2020).

[47] Google. (2020). Google meet page, [Online]. Available: https://meet.google.com/ (visited on 05/30/2020).

[48] Gitlab. (2020). Gitlab website, [Online]. Available: https://gitlab.com/ (visited on 05/30/2020).

[49] Overleaf. (2020). Overleaf website, [Online]. Available: https://overleaf.com/ (visited on 05/30/2020).