Departamentul Automatică şi Informatică Industrială

Facultatea Automatică şi Calculatoare

Universitatea Nationala de Stiinta si Tehnologie

POLITEHNICA din Bucureşti

Splaiul Independenţei 313, 060042, Bucureşti, România

Sala ED 412, Tel. 021/402.92.69

www.aii.pub.ro, email: secretariat.aii@upb.ro

Sesiunea de Comunicări Şiinţifice Studenţeşti

Ediţia 2024

Modelare și Suport Software Integrat Pentru Restricțiile în Vigoare Privind Accesul Auto în Centrul Orașelor din Europa

Autor: Andrei SĂRĂNDAN, Licenţă – Anul 4, Grupa 341 A1, Facultatea de Automatică şi Calculatoare

Adresă e-mail: andreisarandan02@gmail.com

Îndrumător ştiinţific: prof.dr.ing. Anca Daniela IONIȚĂ

Contents

1. Introduction

The World Wide Web represents without a question an indispensable tool in the daily life during present times. On the other hand, the escalating threat of global warming is more prominent than ever before, with recent events occurring across the globe. Ideally, the internet's capabilities must be focused on creating a better world for future generations.

The scope of this project is to present a web-based application that helps users comply to traffic regulations mandated by The European Union, primarily focusing on Low Emission Zones (LEZs). This web application seeks to raise awareness and trust in this scope, enabling users to make informed planning and consider the environmental impact of their travel and vehicle choice. It represents more than just a text repository or a presentation website as it is a comprehensive platform, which includes modern technologies like Application Programming Interfaces (APIs), extensive database model and complex access validation algorithms. All of these aim to deliver a great user experience, from account creation processes, thus raising the possibility of personal vehicle registration, to trip planning features. To improve the user experience, all the complex logic methods, data sets and algorithms are hidden behind the friendly and interactive interface.

The creation of this web application was done in three main steps. Firstly, the application development was based on a tight connection between backend and frontend parts, involving different programming languages and frameworks. Secondly, the extensive relational database model was designed to store relevant information about both the Low Emission Zones and application users. Lastly, in creating the database, a prolongated research was needed to cover the different complexities of the hundreds of LEZs throughout Europe. A key factor was to also fit the everchanging character of the LEZs. The information presented on the web application must be correct, up-to-date, and presented in an understandable manner. By doing so, the web application aims to contribute to broader efforts to create a more sustainable and resilient future for all.

2. Inclusion of the study's domain of interest

The increasing concerns regarding pollution necessitate the implementation of various measures aimed at enhancing air quality. Pollution, resulting from a multitude of sources and presenting in various forms, possesses significant threats to both human health and the environment. Vehicular emissions stand out as one of the most prominent forms of pollution. This is mostly concerning in the urban settlements and metropolises that are suffocated by the raising number of cars, with an average of nearly one car per 1-2 individuals. Passenger cars stand out as a major polluter, accounting for 61% of total CO2 emissions from EU road transport [1]. The European Union is striving to drastically reduce air pollution by implementing new legislation aimed at pushing new vehicles towards achieving zero CO2 emissions. However, the actual progress has proven to be much slower than it was hoped.

The primary methods of reducing CO2 emissions from vehicles are manufacturing more efficient cars and changing the fuel type. This means transitioning to more eco-friendly ones, ideally derived from green or regenerative energy [1]. Even though electric and hybrid cars are taking a considerable part of the newly registered vehicles, the issue of older, higher-emission vehicles must not be overlooked.

Designing a schema to reduce the pollution proves to be a challenging, yet necessary aspect. Such schemas, referred as Urban Vehicle Access Regulations [2], have been implemented in many Western European cities. These rules are found in different forms such as Low Emission Zones, Zero Emission Zones, tolls for driving in congested areas, changing parking rules and limiting traffic in certain areas. The goal is to reach better air quality standards, while also improving traffic [3].

Generally, vehicle categorization and access are based on the vehicle type, fuel type and emission class (Euro emission standard). Additionally, such regulations usually impose some form of toll or fee that must be paid in order to gain access into a specific zone. Out of the many existing forms of Urban Vehicle Access Regulations, most of them are Low Emission Zones (LEZs), accounting for 73% of the current regulations [2].

Low Emission Zones (LEZs) are geographical areas, usually located in large urban settlements, where specific restrictions are imposed on more polluting vehicles. Typically, this kind of vehicles are prohibited from entering the zone, although in certain instances, a fee can be paid

for access. Low Emission zones have proved to be a great method of reducing air pollution, especially targeting fine particles such as NO2, which are highly correlated to several respiratory diseases. The European Union proposes a strategy to gradually implement such areas in the following years [4]. This approach facilitates the acceptance and adoption of Low Emission Zones and allows each nation to fine-tune the schemas to better fit the needs and possibilities of the country and its citizens. Local authorities can effectively manage the requirements and priorities, enhancing the efforts to mitigate pollution as much as possible. According to the report made by statista.com [6], forecasts expect a significant expansion of Low Emission Zones throughout Europe, with an estimated total of 510 zones across 17 countries. This increase highlights the European Union's commitment to minimizing air pollution.

3. Project Statement

The scope of this project revolves around the development of an application that improves user experience by making Low Emission Zones easier to understand. Since this concept is very new and still under development, the main scope is to raise awareness and promote their acceptance. LEZs may pose complexities for many individuals, hence efforts are needed to simplify comprehension and facilitate access. This can be done with a focus on algorithms, thus validating vehicle access within LEZ cities. The aim is to create an intuitive web application that provides information and help to access these Low Emission Zones throughout Europe.

Considering this is an ever-changing topic, the database system is incorporated to cover the regulations that are effective in 2024. It is designed to be adapted for ongoing updates, in order to provide valid information to the end-user.

While online resources and information are indeed available, usually the information is dispersed across many different platforms, with segmentation made by countries or even by cities. Fragmented information can be overwhelming for regular users, who would often find themselves spending significant time navigating through multiple websites when gathering relevant information.

The development of a new web-based application comes from recognizing a specific set of individuals who frequently travel by car. The goal of the application is to eliminate all barriers of

entry, such as subscription fees or even prior expertise in the domain. This empowers the audience to make informed decisions before buying a new vehicle, while also expanding the impact and reach of LEZs. The platform allows users to save vehicles to their account and make use of the route planner features in order to facilitate navigation across European LEZs. The intuitive design allows users to quickly obtain the information they require without navigating through extensive pages of content. Over more, the application features informative text-based pages, for comprehensive insights into LEZs.

At the core, the application makes use of a robust database and extensive algorithms for assigning the correct registration to a vehicle and LEZ access validation. These technical features ensure the accuracy, reliability, timeliness and relevance of the provided information. Even though the application is designed for individual consumers, the database model and it's free-to-use nature make it suitable for enterprises, accommodating a larger number of scenarios, vehicles and routes within the same account.

4. Application Development

4.1 Backend Development

4.1.1 Foundations

Backend development represents the backbone of any web application's architecture and it can be considered the engine that powers the entire system. It is responsible for a range of critical activities, like logic implementation, data processing, and handling communication with the user interface. A robust backend is essential for applications intended to fulfil a large purpose, empowering them to manage data effectively and to interact with users. Without the complex backend, web applications are essentially limited to static website pages, lacking the dynamic functionalities available in most modern digital experiences.

In the context of this application, the backend architecture carries a crucial importance, as it is responsible for handling data management, LEZ access validation through algorithms, user account creation, HTML requests and responses. One of the key aspects of the backend architecture in a web application is the establishment of routes which act as pathways for directing

incoming web requests to the appropriate resources within the application. Managing a meticulous route configuration, ensures smooth user interaction and navigation within the web application.

4.1.2 Route Handling in Flask

Route handling represents a fundamental aspect of web backend development, as it is responsible for dictating how URLs are mapped to specific functions within the application. Considering that Flask is the main framework used in the development of this application, this mapping enables the execution of specific code sequences when a user accesses a particular endpoint.

In Flask, routes are generally defined using Python functions which are decorated with a specific syntax, that includes the web application instance [6]. The python function decorated by this operator is responsible for handling the request when users navigate to the specified page. Additionally, the decorator accepts the methods parameter, which refers to the HTTP requests accepted by the configured route, such as GET and POST.

4.1.3 Database

At the core of this web application is the relational database model displayed in figure 1. This application uses SQLAlchemy, an Object-Relational Mapping (ORM) library which allows developers to define database tables using Python classes. The database is created and managed in the *models.py* file [7].

Structurally, the *Zone* table is responsible for storing the main information for the Low Emission Zones. This represents key information regarding vehicle access, such as required registrations or minimum European emission standards. This data is accessed by internal algorithms and usually queried by class functions that are also defined in the *models.py* file.

ZoneTemporaryData table is related to *Zone* table through a foreign key. Specifically, This each instance from *ZoneTemporaryData* belongs to a single instance in *Zone* table. As its name suggests, *ZoneTemporaryData* table is designed to hold data regarding the temporary characteristics of some Low Emission Zones. This is especially relevant for countries like Italy or Bulgaria which have Winter Low Emission Zone schemas.

Another essential part of the database model is the *User* table and the ones related to it. The *User* table is created to store the essential information for account creation. The amount of information is minimal, as the nature of the application does not require more, but this is also favourable for safety and resource storage reasons. The main fields, email and password are need for user login. In order to keep the same standards as most modern applications, the database stores a hashed version of the password. The encryption is made using the SHA256 algorithm, which is one of the most secure and used hashing algorithms today [8].

As both names suggest, *Car* and *SavedRoute* tables store the relevant data for the user's account. They are related to the *User* table through a foreign key, which creates a one-to-many relationship. This means that a user can have multiple car instances or saved routes associated with his *id.* While in reality a vehicle has plenty of details, the application is designed to store only the data relevant to validating the access in the already existing LEZs. The completion of this data is the user's responsibility and he should respect the information registered in the vehicle book.

Over more, for each country that has a Low Emission Zone, a class extends *GeneralRegistrations* class. The latter is an abstract class and serves as a blueprint for the other country-based registrations, as it does not create a table in the database.

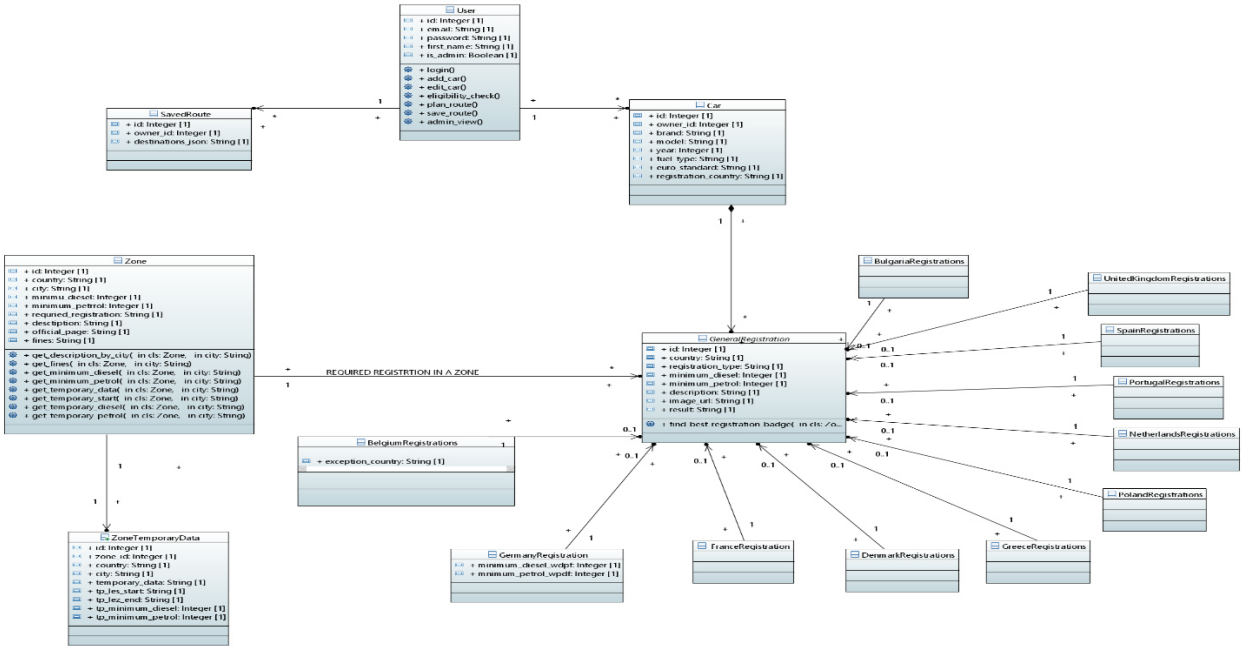

Figure 1 Database model.

4.1.4 Access validation algorithms

The system encompasses a different algorithm for each country that has a LEZ schema. This strategy is necessary for covering all possible scenarios while assuring the validation of the relevant criteria. The structure is designed to ensure easy upgrading (covering more countries or LEZs, including different vehicle types, etc). This is ideal because of the evolving nature of the Low Emission Zones. In order for the application to stay relevant in the future, slight adjustments might be needed in the access validation algorithms. Separating the algorithms for each country means the backend can be viewed as a set of interchangeable modules. Each module represents a country that has a Low Emission Zone schema and is structured in three parts: the validation algorithm, registrations/minimum requirements for access and the country objects.

Since each nation evolves at its own pace, a modular approach is needed for an easier maintenance. This is beneficial for both the developer and the stakeholders, therefore deploying updates will become much simpler. While the responsible researchers can monitor the evolution of each country separately, the developer can implement small updates only in the specific module. This software architecture ensures minimum downtime of the application and quick bug fixes.

The validation algorithms are used in the Route Planner page in order to display a short summary of the relevant information. The displayed information includes a concise statement regarding the access in the selected LEZs and other key details, facilitating user compliance to the regulations. This way, the users do not need to personally research all the regulatory requirements when planning a trip. They only need to fill in all the interest points and select one of their previously saved vehicles. By pressing the button *GO*, the POST request is sent to the application backend and the relevant algorithms are triggered. For each country listed in the request, the corresponding algorithm is triggered. Since most countries have different restrictions from one region to another, the algorithms are designed to check for the zone received in the request. Running a separate algorithm for each country proves to be effective in handling all the possible scenarios (LEZ access granted or forbidden, LEZ active/inactive during selected travel period, city or country does not have LEZ, etc.). This is highly challenging when checking the access for several points of interest at the same time, as it involves verifying different attributes of the selected vehicle.

While the frontend interface of the Route Planner page seems simple, the database queries, the POST request and the validation algorithms are all processed behind the scenes, in the application's backend. As soon as the POST request is received, the *navigation* function handles the data inside the request. Since one request can include several points of interest, they are stored in JSON format and loaded in a list variable. By parsing this list, we can call the correct access validation algorithm for each item.

Based on the values returned by the access validation algorithms, the type and the information of the notifications will be decided. The type of the notifications is success or error and creates a suggestive design. The text inside of the notifications is queried from the Zone database through a series of class methods. These methods are created inside *models.py* file and take the city name parameter. Each of them is responsible for retrieving key information for the user, like short description of the LEZ, penalties for not complying, official authority's webpage, etc. All this data is stored again in the JSON format and sent as a response to the frontend of the application, where unpacking of the information takes place. The instant communication between the frontend and the backend creates a user-friendly page. Moreover, the abstraction principle was used, meaning the complex logic of the application is hidden from the user, who is only interacting with the system through the graphic interface.

4.2 Frontend Development

Essentially, the web application consists of a multitude of HTML, CSS and JavaScript files, structurally interconnected. The HTML files are stored inside the *templates* folder, while the CSS and JavaScript files are stored inside the *static*. The application also uses Jinja templating across all web pages, as they represent an extension of *base.html* file. For example, the navigation bar, which represents a key element as it is present on all webpages, was created in the *base.html* file.

While the market provides a wide range of options, Maps JavaScript API fits perfectly to the requirements of this web application. The API allows creation and web integration of custom, dynamic and interactive maps. All the available Low Emission Zones can be dynamically displayed right from the database. Further, the Route Planner page uses Navigation and Geocoding APIs to display the best route. The map element is updated in real time, without the need to open a new session or refreshing the page. This is achieved by determining the directions through a

JavaScript function, using the navigation feature of the Google API. This JavaScript function is invoked by the asynchronous function which is responsible for handling the form submission. When the user clicks the *GO* button, the POST request containing the HTML form data is submitted.

Asynchronous programming is a key feature of JavaScript and is usually done by creating asynchronous functions. These functions use the *await* key word structures to suspend execution until specified conditions are fulfilled. This behaviour is called promise-based, because async functions return Promise objects. Based on their value, conditional programming is executed with structures like try-catch. Through the async function, the system gathers the user-filled information, and creates the *FormData* element. It is included in the body of the POST request that will be sent to the application's backend. Afterwards, the frontend interface waits for a response from the backend algorithms. Basically, the execution is suspended, by using the *await* key word, until a response from the backend is received.

While granting users the freedom to navigate the application is favourable, guidance in certain selections ensures prevention of errors. This precaution is crucial for validation algorithms to receive accurate input. The application uses dynamic forms based on select fields that are either populated or displayed conditionally. Implementing such constraints helps prevent illogical scenarios, such as saving false data for vehicles, which not only lack practicality but also risk raising errors during the validation process.

Another way the application guides user interaction is through the autocomplete functionality facilitated by the Places API by Google. The feature is optimized for location predictions, significantly enhancing efficiency by minimizing typing time. It includes various points of interest, such as cities, streets, hotels and more. The use of autocomplete function ensures accuracy for selected places, guaranteeing precise input for the Google Geocoding API.

The restrictions are not implemented to hinder user access. They are rather designed to improve user experience, guarantee system functionality and data accuracy.

5. Application usage

Even though the website can be accessed without creating an account, to benefit from all the application features, users should go through the registration process. Such features include associating multiple vehicles to their accounts and saving the most frequent or favourite route. Users do not need to enter any personal information, considering that the purpose of the accounts is to unlock database access.

Registering a vehicle is the first step after account creation. Here, users can fill in the fields relevant for LEZ access validation, like fuel type or European emission standards or even registrations that are already assigned to their vehicle. Later, they have the possibility to edit any of these features, to correct potential mistakes or to assign newly obtained registrations. During this process, users have very few restrictions which are meant to help them fill in the correct data, but the responsibility lies on their shoulders to check the vehicle registration book beforehand.

After registering the vehicle in the application database, the user can access the Eligibility Check page to test the car for potential registrations. This page only gives information about registrations that fit the selected vehicle. The feature is especially helpful for countries like France, that have a multitude of registrations available, or in addition to the Route Planner page.

As the name suggests, the Route Planner page was designed to help users plan their trips thought Europe. Firstly, the user must select one of the vehicles saved to his account, and then insert all the destinations he plans to visit. This is important, since LEZs primarily affect city centres, which are generally avoided by the navigation systems. The validation algorithms only apply to the cities filled in by the users. With the help of Google Places Autocomplete, users can select any point of interest, the city and country being automatically determined. Therefore, they do not need to directly specify those. The access validation algorithms test the selected vehicle against the regulations imposed by the selected cities. In an instant, users will get all the information relevant to their route. For each city, a short notification pops up, informing the user about whether the LEZ is imposed, the minimum European emission standards or registration requirements and the possible penalties. In addition, the links relevant for more information on the respective LEZs are prompted.

Having all the information in one place represents a unique element of the application and should help users comply to the newly imposed regulations. This complex access validation system was designed to integrate a seamless transition between its component elements. By using the pages described above, specifically the Eligibility Check and Route Planner ones, the users can find the relevant information for their particular case. Further, the user is also guided to official pages, such as government or local authorities' websites.

6. Conclusions and Future Plans

The application successfully represents an online platform that facilitates users' access to information regarding Low Emission Zones (LEZs) across Europe. Its user-friendly interface and intuitive features contribute to an engaging experience and reduce the barrier of entry in this field. On the other hand, functionalities, such as account creation and car saving, foster client loyalty and trust, thereby enhancing user retention and satisfaction. This lays a solid foundation and ensures a positive user experience, as individuals can quickly search for accurate information based on their specific circumstances.

It is exclusively designed for passenger car drivers. However, a future expansion can encompass all vehicle types, from motorcycles to heavy vehicles. This strategic plan not only broadens the application's user base, but also creates possible collaborations with enterprises for managing fleets of cars. This can be easily achieved by continuing the research the same way it has been done so far and extending the database model and validation algorithms.

Furthermore, the integration of a shop feature would enable users to directly purchase registrations through the application. This additional function not only enhances user convenience but also diversifies revenue streams, thus increasing the application's sustainability.

Crucially, ongoing updates are imperative to ensure the application remains aligned with the ever-evolving landscape of LEZs across Europe and adapts to regulatory changes. Maintaining up-to-date information is imperative to the application's effectiveness. There is also potential of covering other urban regulations, such as Zero-Emission Zones or road tolls, further enhancing the application's utility and relevance in addressing contemporary mobility challenges.

Bibliography

[1] European Parliament. "CO2 Emissions from Cars: Facts and Figures." [Online]. Available: https://www.europarl.europa.eu/topics/en/article/20190313STO31218/co2-emissions-from-cars-facts-and-figures-infographics. Accessed on 10 march 2024.

[2] European Commission. "Urban Vehicle Access Regulations." [Online]. Available: https://transport.ec.europa.eu/transport-themes/urban-transport/urban-vehicle-access-regulations_en. Accessed on 13 February 2024.

[3] DieselNet. "Low Emission Zones in the EU." [Online]. Available: https://dieselnet.com/standards/eu/lez.php. Accessed on 5 May 2024.

[4] MDPI. "Impact of Low Emission Zones on Urban Air Quality: A Review." [Online]. Available: https://www.mdpi.com/2071-1050/15/23/16260. Accessed on 23 March 2024.

[5] GermanEmissionsSticker.com. "The German Diesel Ban." [Online]. Available: https://www.germanemissionssticker.com/the-german-diesel-ban/. Accessed on 10 February 2024.

[6] Statista. "Low Emission Zones in Europe by Country." [Online]. Available: https://www.statista.com/statistics/1321264/low-emission-zones-europe-by-country/. Accessed on 3 February 2024.

[7] M. Grinberg, Flask Web Development, 2nd Edition. O'Reilly Media, Inc., 2018.

[8] Permana, I., Hardjianto, M., & Baihaqi, K.A. "Securing the website login system with the SHA256 generating method and time-based one-time password (TOTP)." Systematics, vol. 2, nr. 2, pp. 65–71, 2020.