Exercise Report

Course: 02. Big Data, Open Data y Gestión de Datos (2023-25)

Exercise Title: Open Data App Prototype.

Submitted by: Vitor Oliveira de Souza.

November 28th, 2023.

Open Data App Prototype Report

This report outlines the development process and outcomes of the Open Data App Prototype, a task commissioned for the Master's in Big Data program. Focused on creating an application utilizing real-time open data, the prototype aims to offer a dynamic representation of parking lot occupancy in Málaga City. The report provides a systematic overview of the motivations, domain, target audience, and step-by-step implementation of the application. Its purpose is to serve as a comprehensive documentation of the task completion for the Master's in Big Data program.

Project Overview

Motivations and Objectives

The primary objective of this project is to design and implement an "Open App" that utilizes real-time data from open data repositories. Focused on the city of Málaga, Spain, our goal is to contribute to the Open Data movement by providing a valuable and user-friendly application that visualizes and analyzes parking lot data. This prototype serves not only as a standalone application but also as a foundation for future challenges, with the potential to evolve into a significant Open Data tool with more attractive features for its users.

Target Audience

The intended audience for this application ranges from citizens and tourists to urban planners and policymakers. By presenting live data on parking lot occupancy, the app aims to enhance the user experience for those navigating the city while offering valuable insights for city planning and management.

Implementation Steps

To achieve the project's goals, we followed a systematic approach:

1. *Setting Application Foundations:*
   Defined the motivations, domain, and target audience to guide the application's development.

2. *Identifying Data Sources:*
   Selected the Málaga City Open Data Portal as the primary source for real-time parking lot data.

3. *Tools Selection:*
   Chose appropriate tools for data visualization and preliminary analysis, including Flask for the backend, Leaflet for map rendering, and regular data updates using the Apscheduler library.

4. *App Implementation:*
   Developed the Flask application with functionalities for fetching, processing, and rendering parking lot data on both a map and a table.

5. *Documentation and Reporting:*
   Prepared this report detailing the implementation steps, use case descriptions, links, and visual representations.

This report serves as a comprehensive guide to the thought process and technical aspects of creating the Open Data App Prototype. Let's delve into the details of each step, exploring the challenges faced and the solutions implemented to bring this prototype to life.

Setting Application Foundations: Motivations, Domain, and Target Audience

Motivations:

The driving force behind the Open Data App Prototype lies in the recognition of the transformative potential of open data. In an era where information accessibility is paramount, leveraging open data sources offers an unparalleled opportunity to create applications that enhance public engagement, urban planning, and decision-making. By focusing on real-time data related to parking lot occupancy in Málaga City, the primary motivation is to contribute to the Open Data movement, fostering transparency and efficiency in urban mobility.

Domain:

The domain of the Open Data App Prototype is intricately tied to the urban landscape of Málaga City. Specifically, the application delves into the domain of parking lot management and real-time occupancy data. Understanding the nuances of this domain is crucial for the effective interpretation and utilization of the data. The intricacies of parking infrastructure, patterns of usage, and the dynamic nature of occupancy provide the contextual background necessary for creating a meaningful and relevant application within this domain.

Target Audience:

The Open Data App Prototype caters to a diverse yet interconnected target audience. First and foremost, it addresses the needs of citizens and tourists navigating Málaga City, providing them with real-time information on parking availability to enhance their urban experience. Urban planners and policymakers form another segment of the audience, benefiting from the insights derived from the data to inform city planning, optimize parking resources, and improve overall urban mobility. The broad scope of the application ensures that it serves the dual purpose of meeting the immediate needs of individuals while contributing valuable data-driven perspectives to city-level decision-makers.

In essence, the motivations underscore a commitment to transparency, efficiency, and user-centric urban experiences. The chosen domain of parking lot occupancy aligns with the practicalities of urban life, while the target audience diversity reflects a holistic approach to addressing the varied stakeholders in the urban ecosystem. As we delve into the technical implementation and functionalities of the Open Data App Prototype, these foundational elements will guide the development process, ensuring a purposeful and impactful end product.

Identifying Data Sources

In the process of identifying data sources for the Open Data App Prototype, the Málaga City Open Data Portal was strategically chosen as the primary reservoir for real-time parking lot data. The selection of this portal stems from its role as a comprehensive and authoritative repository of openly accessible information pertaining to various aspects of Málaga's urban landscape.

Málaga City Open Data Portal (URL: https://datosabiertos.malaga.eu/)

This portal serves as a centralized hub for a wide array of datasets that encompass diverse facets of city life, including but not limited to transportation, infrastructure, and public services. By leveraging this resource, the Open Data App Prototype taps into the richness of real-time data specifically related to parking lot occupancy, a critical component for enhancing urban mobility.

The decision to use the Málaga City Open Data Portal is strategic for several reasons:

- **Data Credibility**: The portal is maintained by the city authorities, ensuring the authenticity and reliability of the data. This is crucial for the accuracy of the information presented in the application.
- **Comprehensive Coverage**: The portal aggregates data from various municipal sources, providing a comprehensive view of parking lot occupancy across different areas of the city. This breadth of coverage enhances the application's utility for a diverse range of users.
- **Real-Time Updates**: The portal is designed to offer real-time updates, aligning perfectly with the dynamic nature of parking lot occupancy. This ensures that users of the application receive the most current and relevant information.
- **Openness and Accessibility**: As an open data portal, it aligns with the principles of transparency and accessibility. Utilizing open data not only supports the project's objectives but also contributes to the broader ethos of making information freely available to the public.

In summary, the selection of the Málaga City Open Data Portal as the primary data source is a strategic choice, grounded in its credibility, comprehensiveness, real-time capabilities, and alignment with the principles of open data. This decision forms a pivotal aspect of the Open Data App Prototype's foundation, ensuring that the application is built on reliable and up-to-date information sourced directly from the heart of Málaga's municipal data infrastructure.

Tools Selection

In crafting the Open Data App Prototype, a meticulous selection of tools was undertaken to ensure the effective visualization, analysis, and regular updates of parking lot data. The chosen tools contribute to the robustness and efficiency of the application, aligning with the project's objectives.

Choice of a Web Application

The decision to develop the Open Data App Prototype as a web application stems from the inherent advantages it offers in terms of accessibility, usability, and real-time interaction. A web-based approach aligns seamlessly with the project's goals, catering to a broad audience and ensuring a user-friendly experience.

Benefits of a Web Application:

Universal Access: Web applications are accessible from any device with a web browser and an internet connection. This universal access ensures that users can engage with the application on desktops, laptops, tablets, or smartphones, fostering inclusivity and widespread adoption.

- **Cross-Platform Compatibility:** A web application eliminates the need for users to install specific software, making it compatible across various operating systems. Whether on Windows, macOS, or Linux, users can effortlessly access the application without concerns about platform limitations.
- **Real-Time Updates:** Web applications facilitate real-time updates and dynamic content delivery. This is particularly advantageous for the Open Data App Prototype, ensuring that users receive the latest parking lot occupancy information without the need for manual interventions or application updates.
- **User Interaction and Engagement:** Web applications provide an interactive user interface, enabling features such as map navigation, dynamic data filtering, and seamless transitions between different views. This fosters user engagement and enhances the overall user experience.
- **Easy Deployment and Maintenance:** Deploying a web application is relatively straightforward compared to traditional software distribution methods. Updates and maintenance can be executed centrally, ensuring a streamlined process and minimizing disruptions for end-users.

Technologies:

Flask for the Backend:

Flask, a micro web framework for Python, was chosen as the backbone of the application. Its lightweight and modular design perfectly suited the project's requirements. Flask facilitates the creation of web applications with minimal effort, providing a solid foundation for handling HTTP requests, managing routes, and interfacing with the front end. The simplicity and extensibility of Flask make it an ideal choice for developing the server-side logic of the Open Data App.

Leaflet for Map Rendering:

Leaflet, a leading open-source JavaScript library, was employed for dynamic and interactive map rendering. Its user-friendly API and compatibility with various mapping providers make it an

excellent choice for presenting geospatial data. Leaflet's lightweight nature ensures optimal performance, crucial for a smooth user experience when interacting with the map component of the application. The library's versatility allows for seamless integration with other technologies, enhancing the visual representation of parking lot data on the map.

Apscheduler Library for Regular Data Updates:

To ensure the application maintains real-time relevance, the Apscheduler library was integrated for scheduling regular updates of parking lot data. Apscheduler, built on the popular APScheduler package, provides a straightforward solution for executing tasks at predetermined intervals. By employing this library, the Open Data App Prototype fetches the latest parking data from the Málaga City Open Data Portal, keeping the application dynamically synchronized with the evolving state of parking lot occupancy. This automated approach minimizes manual intervention, enhancing the application's efficiency in providing timely and accurate information to users.

The synergy of Flask, Leaflet, and Apscheduler forms a robust technological foundation for the Open Data App Prototype. Flask empowers the backend logic, Leaflet enriches the user interface with interactive map features, and Apscheduler ensures that the application remains current through regular data updates. This toolset reflects a strategic and purposeful approach to technology selection, laying the groundwork for a seamless and impactful user experience.

App Implementation

In this section, we'll discuss the development process of the Open Data App Prototype, highlighting the Flask application's key functionalities for fetching, processing, and rendering parking lot data on both a map and a table. In order to not overload this report, all the code to generate the application can be check in the GitHub repository:

https://github.com/vitorodesouza/malaga_parkinglots_app

Backend implementation

The Flask application serves as the backbone of the Open Data App Prototype, orchestrating the retrieval, processing, and presentation of real-time parking lot data. Let's dissect the key components and functionalities of the implemented Flask script:

1. **Flask App Initialization**: The script initializes a Flask application instance, providing the foundation for building a web-based interface to interact with the parking data.
2. **Global Variable for Parking Data**: A global variable, parking_data, is defined to store the retrieved parking lot data globally. This allows seamless sharing of data across different functions within the Flask application.
3. **Data Fetching Function:** The fetch_parking_data function is responsible for making periodic requests to the Málaga City Open Data Portal API. It updates the global parking_data variable with the retrieved JSON data, ensuring the application remains synchronized with the latest parking occupancy information.
4. **Data Fetch Scheduler:** To maintain real-time relevance, a scheduler is implemented using the BackgroundScheduler from the apscheduler library. This scheduler triggers the fetch_parking_data function at regular intervals (every minute), ensuring the application stays up-to-date with the latest parking data.
5. **Flask Routes:** Two Flask routes are defined:
    - The '/' route renders the main page using the index.html template, passing the parking_data variable for dynamic content.
    - The '/map' route returns the raw parking data. This can be useful for potential future functionalities or testing purposes.
6. **Initial Data Fetch and App Start:** On script execution, the initial data fetch is triggered, populating the parking_data variable. Subsequently, the Flask application is started in debug mode, allowing for real-time debugging during development.

This Flask script orchestrates a seamless flow of data from the Málaga City Open Data Portal to the web interface, ensuring users are presented with the most up-to-date parking occupancy information.

Frontend implementation

The HTML file provides the structure of the frontend, including the header, map container, and table for parking data. It also links external stylesheets for Leaflet and includes inline styles for the page.

The JavaScript code handles the dynamic updates and rendering of parking data on the map and table. It uses Leaflet for map creation and manipulation, fetches data from the server (/map endpoint), and updates the map and table based on the received data. The code ensures that markers are placed on the map for locations with available parking spots, and corresponding rows are added to the table.

This combination of HTML and JavaScript creates a dynamic and responsive frontend that interacts with the Flask backend to present real-time parking occupancy information.

Roadmap for Application Enhancement

User Interface (UI) Refinement

Objective: Enhance the visual appeal and usability of the application.

Tasks:

- Implement a responsive design to optimize the app for various screen sizes.
- Introduce a user-friendly color scheme for better readability.
- Explore additional styling improvements for a more modern look.

User Interaction Enhancements:

Objective: Improve the overall user experience and engagement.

Tasks:

- Implement interactive features, such as tooltips or pop-ups, to provide more information about parking lots.
- Allow users to filter parking lots based on criteria like availability, location, or status.
- Include a legend for the map markers to help users interpret the displayed information.

Real-Time Notifications:

Objective: Provide users the ability to create timely updates and alerts on parking information.

Tasks:

- Integrate a notification system to alert users of significant changes in parking lot availability or status.
- Explore options for push notifications to keep users informed even when the app is not actively open.

User Feedback Mechanism:

Objective: Encourage user feedback and improve the application based on user suggestions.

Tasks:

- Implement a feedback form or feature within the app.
- Regularly review and analyze user feedback to identify areas for improvement.

Multilingual Support:

Objective: Expand the app's reach by supporting multiple languages to improve tourists' usability.

Tasks:

- Implement language localization to accommodate users who prefer languages other than English and Spanish.
- Provide an option for users to select their preferred language.

Performance Optimization:

Objective: Ensure the app operates efficiently, especially during periods of high user traffic.

Tasks:

- Optimize code for faster data processing and rendering.
- Implement caching strategies to reduce server load and enhance overall app responsiveness.

Documentation and Help Section:

Objective: Assist users in understanding and effectively utilizing the app.

Tasks:

- Create a comprehensive help section or documentation within the app.
- Include FAQs, tutorials, or tooltips to guide users through various features.

Collaboration with Local Authorities:

Objective: Strengthen collaboration with local authorities for more accurate and comprehensive data.

Tasks:

- Establish communication channels with relevant city departments to enhance data accuracy.
- Explore opportunities for data enrichment, such as additional information on parking regulations or special events affecting parking.

Application screenshot