Bucharest

Overview

This project aims to develop a sophisticated digital twin tool tailored for the cities of Bucharest and Paris. This tool will leverage open-source GIS platforms, extending their functionality to create a highly interactive and administrable solution for city stakeholders. The tool will enable the visualization of 2D and 3D city maps, real-time data integration from city sensors, and an advanced system for managing Points of Interest (PoIs) and associated alternative proposals. This proposal outlines the detailed functional requirements and implementation plan to meet the needs of both cities.

Functional Requirements

1. **Use Cases:**
    - **Communicate Information to Stakeholders:**
        - Provide detailed information to various stakeholders regarding ongoing and future city projects.
        - Enable stakeholders to visualize the impact of different proposals and alternatives. (across KPI's)
    - **Gather Feedback for Internal Decision Making:**
        - Implement an Opinion Gathering System (OGS) to collect feedback from citizens and stakeholders.
        - Use feedback to inform and guide city planning and decision-making processes.
2. **Scope:**
    - **Current and Future Projects:**
        - Focus on both ongoing projects and future initiatives.
        - Ensure the tool can be easily updated and maintained for future use cases.
    - **Administrative Capabilities:**
        - Provide robust administrative tools for city officials to manage data, PoIs, and proposals/alternatives.
3. **Visualization:**
    - **2D/3D Visualization:**
        - Display 2D cartography, orthophotos, and 3D terrain models.
        - Include self-modelled buildings and other 3D objects.
    - **User Interaction:**
        - Users can move freely within the map using a 6 Degrees of Freedom (6DoF) system
4. **Real-Time Data:**
    - **Sensor Data Integration:**
        - Incorporate real-time data from city sensors into the visualization.
    - **Data-driven Decision Making:**
        - Display KPIs related to the project impact, such as green area increase and pollution reduction.
5. **Points of Interest (PoI):**
    - **Creation and Management:**

        - Administrators can create, modify, and delete PoIs.
        - PoIs will include linked alternatives and their associated KPIs.
    - **Role-Based Filtering:**
        - Display PoIs based on the user's role (e.g., local business, citizen).
6. **Opinion Gathering System (OGS):**
    - **Integrated Feedback Mechanism:**
        - Implement a system for collecting opinions from users via surveys, rating systems, and textboxes.
    - **Data Analysis:**
        - Analyse collected data to provide insights, such as commonly used words and overall sentiment.
7. **Administrative Capabilities:**
    - **Object and Layer Management:**
        - Create and manage objects (2D/3D) with various attributes.
        - Assign objects to specific layers and proposals.
    - **Proposal Management:**
        - Define proposals containing a set of layers, each representing different scenarios or alternatives.
        - Link KPIs and real-time data to proposals.

Technical Specifications

1. **Platform:**
    - **Base Platform:**
        - Utilize an open-source GIS platform such as QGIS or MapServer as the foundation.
    - **3D Engine:**
        - Integrate a 3D rendering engine (e.g., CesiumJS, Three.js) to handle 3D visualization and navigation. In case not possible, fake a isometric view system allowing only 3DoF (translation) with fixed rotation angle
2. **System Design:**
    - **Architecture Design:**
        - Design the system architecture, including frontend, backend, and database components.
    - **Data Model:**
        - Define the data model for storing cartography, 3D objects, sensor data, and feedback.
3. **Development Phase:**
    - **Frontend Development:**
        - Develop the interactive map interface and integrate 3D visualization capabilities.
        - Implement hotspot pop-ups and the OGS interface.
    - **Backend Development:**
        - Set up the spatial database and develop APIs for data management and real-time updates.
        - Integrate sensor data ingestion pipelines and real-time data processing.
4. **Testing and Validation:**
    - **Unit and Integration Testing:**
        - Conduct thorough testing of individual components and their integrations.

- **User Acceptance Testing (UAT):**
    - Perform UAT with a focus group to gather feedback and identify issues.
    - Iterate on the design and functionality based on user feedback.

5. **Deployment and Maintenance:**
    - **Deployment:**
        - Deploy the application on cloud infrastructure (e.g., AWS, Azure).
    - **Monitoring and Maintenance:**
        - Implement monitoring tools to track system performance and user activity.
        - Establish a maintenance plan for regular updates and bug fixes.