Concept: A Smart tourism portal

1. Motivation and Vision

The primary goal of creating the Tourism Statistics Portal is to address the shortcomings of traditional methods used for gathering and analyzing tourism data. Currently, challenges such as fragmented data sources, labor-intensive manual analysis, and delays in obtaining insights impede the ability of city officials to make timely and informed decisions. This often leads to poor resource allocation, an inability to respond swiftly to changing tourism trends, and missed opportunities to attract visitors and generate revenue for the city.

To overcome these issues, a streamlined, user-friendly platform is needed to centralize and automate the collection and analysis of tourism data. Such a platform would provide a comprehensive view of visitor behavior, spending patterns, and preferences, thereby empowering authorities to make data-driven decisions that enhance the city's attractiveness to tourists and improve their overall experience.

Designed specifically for public authority stakeholders—including city officials, district leaders, public transport providers, and attraction managers—the portal offers a wide range of analytical tools, from basic to advanced. This ensures that it meets the diverse needs and preferences of these user groups. Additionally, the possible future integration of external services, such as social media insights, could enrich the data further, offering even deeper context and enhancing the overall quality of the insights provided.

2. Stakeholders

2.1 Primary Stakeholders

Mayor and City Council

The Mayor and City Council will utilize the portal to monitor city-wide tourism trends and maintain an overview of upcoming events. By analyzing these trends and anticipating future developments, they can allocate resources more effectively and coordinate with city infrastructure services to prepare for significant events. Additionally, high-level insights into the city's tourism patterns will highlight areas needing attention, enabling the Mayor to reach out to district leaders, or directly task the tourism department or external marketing teams with developing targeted campaigns to attract more tourists to under-visited districts. Ultimately, the portal will support informed decision-making, aligning with the city's economic and cultural development goals.

District Leaders

District Leaders will use the portal to analyze tourism trends and address the specific needs within their districts. By comparing current data with past performance and other districts, the portal will help identify the strengths and weaknesses in attracting and retaining tourists. It will also highlight potential issues, such as insufficient or ineffective infrastructure, that impacts the tourist experience. Equipped with this information, local leaders can implement targeted improvements and initiatives tailored to the current situation in their districts.

Tourism Department

The Tourism Department will use the portal to gain deep insights into tourist behavior. Utilizing the portal's advanced analytical tools, they will develop highly targeted and effective marketing campaigns and promotional activities based on real-time data. This approach will maximize the city's visibility and attractiveness, while also distributing tourists more evenly across various attractions, thereby alleviating pressure on the main tourist hotspots. Ultimately, this will boost tourist numbers and enhance the city's image as a desirable destination.

2.2 Secondary Stakeholders

City Infrastructure Services

City infrastructure services, such as public transport services, can adjust their operations based on tourism patterns identified through the portal. Firstly, they will be able to flexibly adapt to expected events in the city. Secondly, insights into tourist hotspots and peak times can be utilized to optimize provided services, such as transport schedules and routes, improving service efficiency and enhancing the overall satisfaction of both tourists and residents.

Attractions Managers

Attractions Managers will have access to detailed tourist data and feedback specific to their attractions, as well as average values for comparison. This information, combined with additional context from e.g. social media feeds, will highlight opportunities to improve visitor experiences and operational efficiency. The portal will help managers understand visitor demographics and preferences, allowing them to accordingly enhance the quality of their services. By making data-driven adjustments, attractions can increase visitor satisfaction, boost both new and repeat-visits, and ultimately increase the profitability of both their attractions and the city as a whole.

Local Businesses

Local businesses will benefit from the portal by accessing valuable insights into tourist demographics and spending patterns. This information will enable them to tailor their products, services, and marketing efforts to better meet the needs and preferences of tourists. By understanding where tourists are spending their time and money, businesses can adjust their offerings and promotions to attract more customers. As a result, they can enhance their business performance, which in turn enriches the local economy and improves the overall tourist experience.

Event Organizers

Event organizers will use the portal to gain insights into tourist demographics and preferences, as well as maintain an overview of other events. This will help them plan and promote events that appeal to both tourists and residents, increasing attendance and engagement. The portal will also help identify optimal times and locations for events, avoiding scheduling conflicts and ensuring maximum impact. The increased success of individual events will contribute to the city's economy and help establish it as a destination with a dynamic and appealing cultural scene.

3. Functional Requirements

Note: The listed features would be required for a real-world implementation of a smart portal system, some sections are however out of scope of our simplified project

Data Collection and Integration

1. **Real-Time Data Collection**: The portal must be capable of collecting real-time data from various sources, including ticketing systems, social media platforms, and event calendars.
2. **API Integration**: The portal should support API integration with existing city infrastructure and possibly third-party services to integrate various data sources, providing complex insights.
3. **Data Storage**: The system must include a robust and scalable database to store large volumes of tourism data securely.

User Access and Authentication

4. **User Authentication**: The portal should provide a secure authentication mechanism, including support for single sign-on (SSO) and multi-factor authentication (MFA) to ensure that only an authorized person can access the system.
5. **Role-Based Access Control**: Role-based access control ensures that users only have access to the data and functionalities relevant to their responsibilities. Different roles such as administrators, district leaders, and local businesses should have distinct levels of permissions.

Data Analysis and Visualization

6. **Data Analytics Tools**: Integrate analytical tools that allow users to perform data analysis, including trends, predictions and comparative data across different time periods and districts.
7. **Customizable Dashboards**: Users should be able to customize their portal dashboards to display their preferred metrics and reports. The system should support various data visualization formats such as charts, graphs, and heatmaps.

Event Management

8. **Event Tracking**: The portal should include functionality to track details of city events, including dates, locations, and anticipated attendance. This data should be integrated into the overall analysis to assess its impact on tourism patterns.
9. **Conflict Detection**: Detect potential scheduling conflicts for events and provide recommendations for optimal timing and location based on historical data and current trends.

Performance and Scalability

11. **Scalability**: The portal must be designed to scale horizontally to accommodate increasing volumes of data and user activity without compromising performance.
12. **Performance Monitoring**: Include tools to monitor the performance of the portal, ensuring that data processing and user queries are handled efficiently and in a reasonable response time.

User Support

13. **Helpdesk Support**: Implement a way for users to report issues with the portal platform and receive support.

Security and Compliance

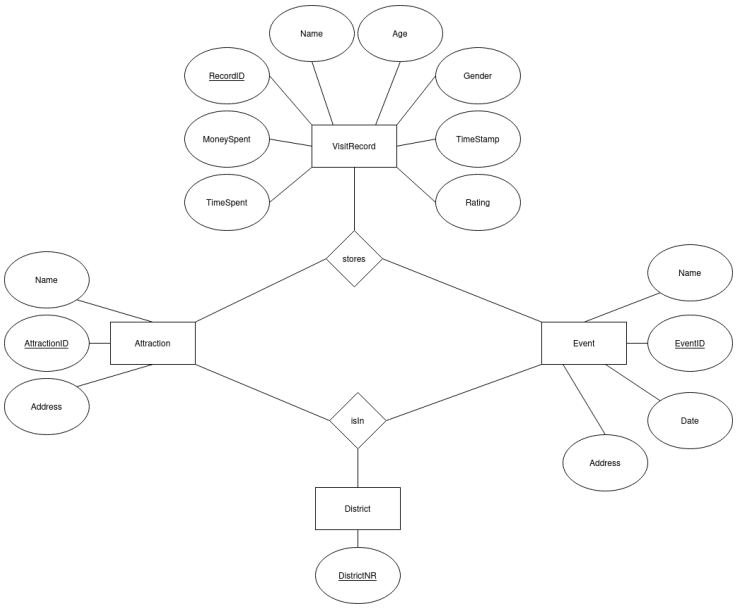
15. **Data Anonymization:** The collected data will be stripped of personal identifiers such as name or surname, replaced by anonymised values. Other sensitive data such as gender, age or money expenditure will be carefully handled, adhering to data protection principles
16. **Data Encryption**: Ensure that all data transmitted and stored by the portal is encrypted to protect against unauthorized access and data breaches.
17. **Compliance**: The portal must comply with local and international data protection regulations, especially GDPR. Regular audits should be conducted to ensure ongoing compliance.

System Maintenance and Updates

17. **Automatic Updates**: Implement a system for automatic updates and patches to ensure that the portal remains secure and up-to-date, also allowing for adding of latest features and improvements. This is achieved through the usage of CI/CD pipeline.
18. **Backup and Recovery**: Ensure regular data backups and provide a robust recovery plan to minimize data loss in case of system failures or disasters.

4. Service Description

The City Tourist Data Portal is designed to assist city authorities, including the mayor, by providing access to and exploration of tourist data. The portal supports decision-making processes, particularly in marketing, by offering detailed statistics and visualizations based on data gathered from an integrated ticket system of the city attractions. All operations will be performed either directly through the API or via a user interface that makes the necessary API calls. The data behind the microservice has the following structure:



5. API Operations

GET /attractions

- **Description** : Retrieve all tourist attractions.
- **Endpoint** : /api/v1/attractions
- **Inputs** : None
- **Outputs** :


```
[
        {
        "attractionID": "string",
        "name": "string",
        "district": "string",
        "address": "string"
        }
]
```


GET /attractions/{districts}

- **Description**: Retrieve tourist attractions within specified districts.
- **Endpoint**: /api/v1/attractions/{districts}
- **Inputs**:
  - **districts**: (list of strings, required) A list of district names.
- **Outputs**:


```
[
 {
        "attractionID": "string",
        "name": "string",
        "district": "string",
        "address": "string"
 }
]
```


GET /attractions/{id}

- **Description**: Retrieve a specific tourist attraction by ID.
- **Endpoint**: /api/v1/attractions/{id}
- **Inputs**:
  - **id**: (string, required) The ID of the attraction.
- **Outputs**:


```
{
 "attractionID": "string",
 "name": "string",
 "district": "string",
 "address": "string"
}
```

GET /events

- **Description**: Retrieve all events.
- **Endpoint**: /api/v1/events
- **Inputs**: None
- **Outputs**:

```
[
 {
        "eventID": "string",
        "name": "string",
        "date": "string",
        "district": "string",
        "address": "string"
 }
]
```

GET /events/{districts}

- **Description**: Retrieve events within specified districts.
- **Endpoint**: /api/v1/events/{districts}
- **Inputs**:
    - **districts**: (list of strings, required) A list of district names.
- **Outputs**:

```
[
 {
        "eventID": "string",
        "name": "string",
        "date": "string",
        "district": "string",
        "address": "string"
 }
]
```

GET /events/{id}

- **Description**: Retrieve a specific event by ID.
- **Endpoint**: /api/v1/events/{id}
- **Inputs**:
    - **id**: (string, required) The ID of the event.
- **Outputs**:

```
{
 "eventID": "string",
 "name": "string",
 "date": "string",
 "district": "string",
 "address": "string"
}
```

GET /stats/{district}/ratings

- **Description**: Retrieve ranking data for a specific district, filtered by age range, gender, and time range.
- **Endpoint**: /api/v1/stats/{district}/ratings
- **Inputs**:
    - **district**: (string, required) The name of the district.
    - **ageRange**: (string, optional) The age range of tourists.
    - **gender**: (string, optional) The gender of tourists.
    - **timeRange**: (string, optional) The time range for the data.
- **Outputs**:

```
{
"ratingMonthly": [
        "year":  "number",
        "month": "number",
        "avgMonthlyRating": "number"
]
"avgRating": "string",
 "top5Attractions": [
        {
        "attractionID": "string",
        "name": "string",
        "rating": "number"
        }
 ],
 "bottom5Attractions": [
        {
        "attractionID": "string",
        "name": "string",
        "rating": "number"
        }
 ]
}
```

GET /stats/{district}/time

- **Description**: Retrieve data on amounts of time spent at attractions in a specific district, filtered by age range, gender, and time range.
- **Endpoint**: /api/v1/stats/{district}/time
- **Inputs**:
    - **district**: (string, required) The name of the district.
    - **ageRange**: (string, optional) The age range of tourists.
    - **gender**: (string, optional) The gender of tourists.
    - **timeRange**: (string, optional) The time range for the data.
- **Outputs**:

```
{
"timeSpentMonthly":[
        "year":  "number",
        "month": "number",
        "avgMonthlyTimeSpent": "number"
]
"avgTimeSpent": "number",
```

```
"totalTimeSpent": "number",
 "top5Attractions": [
            {
            "attractionID": "string",
            "name": "string",
            "timeSpent": "number"
            }
 ],
 "bottom5Attractions": [
            {
            "attractionID": "string",
            "name": "string",
            "timeSpent": "number"
            }
 ]
}
```


GET /stats/{district}/spending

- **Description**: Retrieve data on the amount of money spent at attractions in a specific district, filtered by age range, gender, and time range.
- **Endpoint**: /api/v1/stats/{district}/spending
- **Inputs**:
    - **district**: (string, required) The name of the district.
    - **ageRange**: (string, optional) The age range of tourists.
    - **gender**: (string, optional) The gender of tourists.
    - **timeRange**: (string, optional) The time range for the data.
- **Outputs**:


```
{
“avgSpendingMonthly”:[
            “year”:  “number”,
            “month”: “number”,
            "avgMonthlySpending": "number"
]
 "avgSpending": "number",
totalSpending": "number",
 "top5Attractions": [
            {
            "attractionID": "string",
            "name": "string",
            "spending": "number"
            }
 ],
 "bottom5Attractions": [
            {
            "attractionID": "string",
            "name": "string",
```

```
        "spending": "number"
        }
 ]
}
```


GET /stats/{district}/visits

- **Description**: Retrieve data on the amount of visits of each attraction for a specified district (with relevant filters applied)
- **Endpoint**: /api/v1/stats/{district}/visits
- **Inputs**:
    - **district**: (string, required) The name of the district.
    - **ageRange**: (string, optional) The age range of tourists.
    - **gender**: (string, optional) The gender of tourists.
    - **timeRange**: (string, optional) The time range for the data.
- **Outputs**:


```
{
“avgVisitsMonthly”:[
        “year”:  “number”,
        “month”: “number”,
        "avgMonthlyVisits": "number"
]
"avgVisits": "number",
"totalVisits": "number",
"top5Attractions":
[
{
"attractionID": "string",
"name": "string",
"visits": "number"
}
],
"bottom5Attractions": [
{
"attractionID": "string",
"name": "string",
"visits": "number"
}
]
}
```


6. UI Concept

The UI will have a list of districts. Upon clicking on a district, the user will see a drop-down list of the Attractions and Events taking place in that district. In addition to that there will be a “Get More Information” Button, that when clicked will show a graph of the monthly changes of the visitor ratings, time spent, money spent and visits (line graph). There will also be a list of the top and bottom 5 attractions for each of these 4 categories, together with their respective values. All of this information will be procured by making an API call to **GET /stats/{district}/, /events/ & /attractions/** in the background.

Each attraction and event will also have a drop-down list that, when called by the user, will display all the information procured by the **/events/ & /attractions/** endpoints.

If possible (depending on the time and resources) there will be a map of Vienna (provided by Open Street Maps) where all the events and attractions will be represented by pins. These pins will be clickable and will show the information relevant to the respective item when requested by the user. The districts themselves will also be clickable and will provide the **/stats/{district}/** information.

7. Architecture

Components:

7.1. Backend

- Spring Boot: the backend will be implemented in java and the java spring boot framework will be used as it simplifies the setup of the application with minimal configurations and has some key features as:
    - Spring Security: For authentication and authorization mechanisms.
    - RESTful Services: Exposing endpoints for frontend interaction.
    - Spring Data JPA: For data access and ORM (Object-Relational Mapping) with the relational database.

  Which we are planning to use.

7.2.Relational Database

We will use a relational database (e.g., MySQL, PostgreSQL, etc.) to store and manage our data. A rough schema has already been outlined in this document to illustrate the database structure.

7.3. Frontend

The frontend will be built with the following structuring and design components:

- HTML
- CSS
- Bootstrap

As for rendering dynamic content and server-side processing we will use:

- Java Server Pages

Java Server Pages: JSP enables the development of server-side web applications with the ability to generate dynamic content based on user requests. It simplifies the creation of dynamic web pages through embedding java code within HTML using tags.
JSP was chosen as it is easy to implement and to adapt fast by every member of the team, so that the workflow won't be slowed down learning a complex technology.

8. Functional Capabilities

- Utilization of different graphical components/interfaces
- Utilization of APIs to get the desired results
- Scalability to handle big chunks of data in the database
- ...will be extended as development continues

9. Deployment with Dependencies

The deployment is automated as a CI/CD pipeline in the Git environment. Every push/commit to the master branch results in:

- Build of software
- Creation of a docker image
- Deployment of the docker image on the eis.dke.univie.ac.at server

Dependency management is simplified by using Docker - application and its dependencies will be packaged into a single container.

So far we require following dependencies:

- Java Development Kit (JDK): Required for running the Spring Boot application.
- Spring Boot Framework: Used for backend development and integration.
- Relational Database Management System (RDBMS)
- Web Server
- Frontend Libraries: Bootstrap, HTML and CSS files
- ... (will be extended as development continues)

10. Legal Considerations

So far we are not aware of any legal considerations except for the standard ones relating to plagiarism and use of AI generated content. Further Legal considerations might become relevant should we decide to include access to external API services.

ChatGPT was employed for text formulation and editorial suggestions, with careful consideration to ensure the content's originality and to prevent inadvertent plagiarism. No external sources were intentionally reproduced in the drafting of this manuscript.