REPORT

Discipline: **Network Programming**

Task: **Develop a service/program that implements a ServerSocket which listens in some port to get requests and provide relevant information.**

Student: Istrati Daniel

Athens 2025

Resources used:

- https://edurank.org/geo/md/
- https://www.gov.uk/government/publications/moldova-list-of-medical-facilities-and-practitioners/list-of-medical-facilities-in-moldova
- https://en.wikipedia.org/wiki/List_of_airports_in_Moldova
- https://en.wikipedia.org/wiki/List_of_museums_in_Moldova
- https://www.mywanderlust.pl/things-to-do-in-chisinau-moldova/
- https://www.tripadvisor.co.uk/Attractions-g294456-Activities-Chisinau_Chisinau_District.html
- https://www.yellowpages.net/places/MD

For the noun declaration we have automatised the process using ChatGPT. It has been fed the correct way of declaring cases and then given each txt file. Each declaration has been verified to correspond according to Romanian language rules. Top 10 biggest cities sorted by population where used, as well as 100 enitites in total according to the task.

Objectives

- Provide quick access to city and entity information.

- Implement a user-friendly GUI for input and response visualization.

- Demonstrate the use of Java sockets for client-server communication.

Implementation

This program allows users to search for city details and entities related to those cities based on predefined data files. It uses both console-based interaction and a graphical user interface (GUI) for user input. The system also supports client-server communication via sockets.

System Design

Architecture:

The program follows a modular design with a client-server architecture. The server processes queries, while the client provides an interface for user interaction.

Modules:

- `CityClass`: Represents city details.

- `EntityClass`: Represents entities linked to cities.

- `Main Class`: Handles server functionality, file loading, and query processing.

- `GUIClient`: Provides a graphical interface for user interaction.

Functionality

Core Features:

- Query cities by name using a command like `City Chisinau`.

- Retrieve entities associated with a city, e.g., `Sights Bălţi `.

- Display results in the console or via a GUI.

Additional Features:

- Socket-based communication between the client and server.

Implementation Details

Programming Language: Java

Key Components:

- `CityClass` and `EntityClass` store and manage data.

- `searchCity` and `searchEntities` methods process user queries.

- Server listens for client queries via `ServerSocket`, while the GUI client sends requests and displays responses.

Execution Instructions

1. Compile the server: `javac Main.java`.

2. Run the server: `java Main`.

3. Compile the client: `javac GUIClient.java`.

4. Run the client: `java GUIClient`.

5. Enter queries in the GUI, such as `City Chisinau`.

Test Cases

- Query: `City Chisinau`

  Expected Output: Displays detailed information about Chisinau.

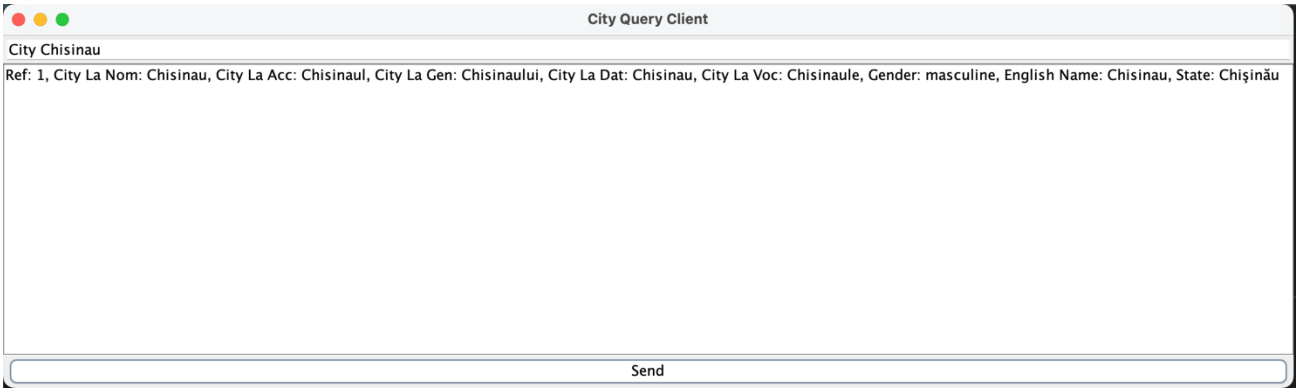

Figure 1. Display Information about City in query

- Query: `Sights Bălţi`

  Expected Output: Lists all sights in Bălţi.

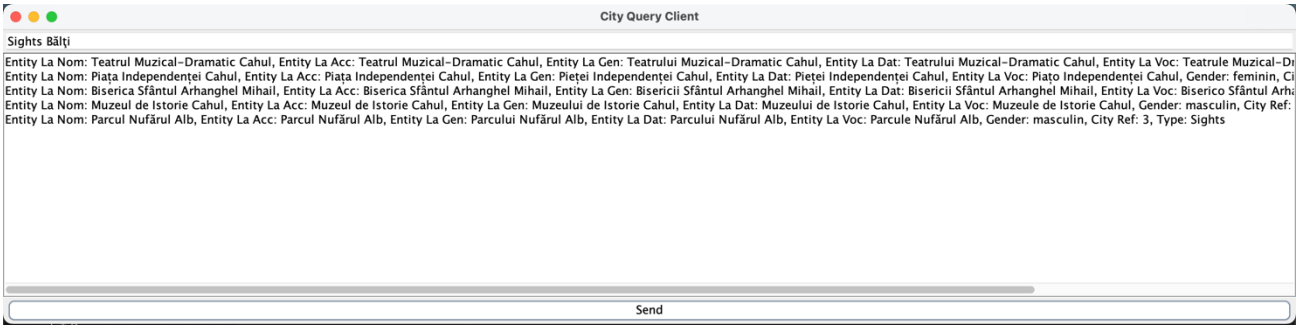

Figure 2. Display Information about Sights in Bălți

Conclusion

The City Query and Entity Search System successfully allows users to retrieve information about cities and related entities. Its GUI and socket-based architecture enhance usability and scalability.