**1. Introduction:**

The given concept, known as the City Data Manager, was created to give a user an effective tool for working with information about cities. It enables the users to complete the tasks like displaying the information of a given city, mapping of a city with all its parameters, sorting the cities by density of population, and creating a new city. This report aims at explaining the structure of the code; class diagrams, class descriptions, output of running the methods in the classes, and references made.

**2. Overview of the Code:**

The code consists of three main components:

* **City**: Represents a city with attributes such as name, population, latitude, longitude, etc.
* **Map**: Manages a collection of cities and provides operations to interact with them, such as adding cities, displaying information, sorting, etc.
* **CityManagerApp**: Implements a PyQt GUI application to interact with the **Map** and perform various operations.

**3. UML Diagrams:**

***Class Diagram:***

+------------------+ +----------------------+ +--------------------------+

| main.py | | city\_manager\_app.py| | data\_loader.py |

+------------------+ +----------------------+ +--------------------------+

| - CityManagerApp| | - CityManagerApp | | |

| - City | | | | + load\_data\_from\_csv() |

| - Map | | | +--------------------------+

| - run\_application| +----------------------+

+------------------+ |

| |

| |

+---------------+-------------+

|

|

+-------------v------------+

| city.py |

+--------------------------+

| - City |

+--------------------------+

**4. Explanation of Classes:**

**1. main.py:**

* **Purpose:**
  + Acts as the entry point of the application.
  + Reads data from a CSV file using the **data\_loader** module.
  + Initializes the PyQt GUI application (**CityManagerApp**) and runs it.
* **Relationship with other files:**
  + Imports **CityManagerApp** from **city\_manager\_app.py**.
  + Utilizes the **data\_loader** module to load city data.

**2. city\_manager\_app.py:**

* **Purpose:**
  + Implements the PyQt GUI application (**CityManagerApp**).
  + Handles user interactions and operations.
* **Relationship with other files:**
  + Imports **Map** from **map.py** and **City** from **city.py**.
  + Utilizes **Map** to perform operations and display results in the user interface.

**3. data\_loader.py:**

* **Purpose:**
  + Contains functions to load data from a CSV file.
* **Relationship with other files:**
  + Typically used by **main.py** to load city data from a CSV file.

**4. city.py:**

* **Purpose:**
  + Defines the **City** class, representing a city with attributes such as name, population, latitude, longitude, etc.
* **Relationship with other files:**
  + Used by **Map** and **CityManagerApp** to create city objects and display city information.

**5. map.py:**

* **Purpose:**
  + Defines the **Map** class, managing a collection of cities and providing methods to interact with them, such as adding cities, displaying information, sorting, etc.
* **Relationship with other files:**
  + Used by **CityManagerApp** to perform operations and display results in the user interface.

**5. Sample Outputs:**

**1. Display City Information:**

City Name: Berlin Population: 8398748 Latitude: 40.7128 Longitude: -74.0060

**2. Display the Most Crowded N Cities:**

City: Berlin, Population: 3644826.0

City: Hamburg, Population: 1841179.0

City: Munich, Population: 1471508.0

City: Cologne, Population: 1085664.0

**3. Display the Least Populated N Cities:**

City: Cochem, Population: 5312.0

City: Kusel, Population: 5405.0

City: Seelow, Population: 5426.0

City: Neustadt, Population: 5727.0

**4. Generate Entered City Map:**

Map generated successfully.

**5. Display all Cities:**

City: Berlin, Population: 3644826.0

City: Hamburg, Population: 1841179.0

City: Munich, Population: 1471508.0

City: Cologne, Population: 1085664.0 ...

**6. Sort Cities according to their Population:**

City: Cochem, Population: 5312.0

City: Kusel, Population: 5405.0

City: Seelow, Population: 5426.0

City: Neustadt, Population: 5727.0 ...

**6. References:**

* PyQt documentation: https://www.riverbankcomputing.com/static/Docs/PyQt5/
* Python documentation: <https://docs.python.org/3/>
* Stack Overflow: Various questions and answers related to PyQt and Python programming.