1. **Problem Statement**

Find out whether two cities are connected. Two cities are considered connected if there is a series of roads that can be traveled from one city to another

1. **Introduction**

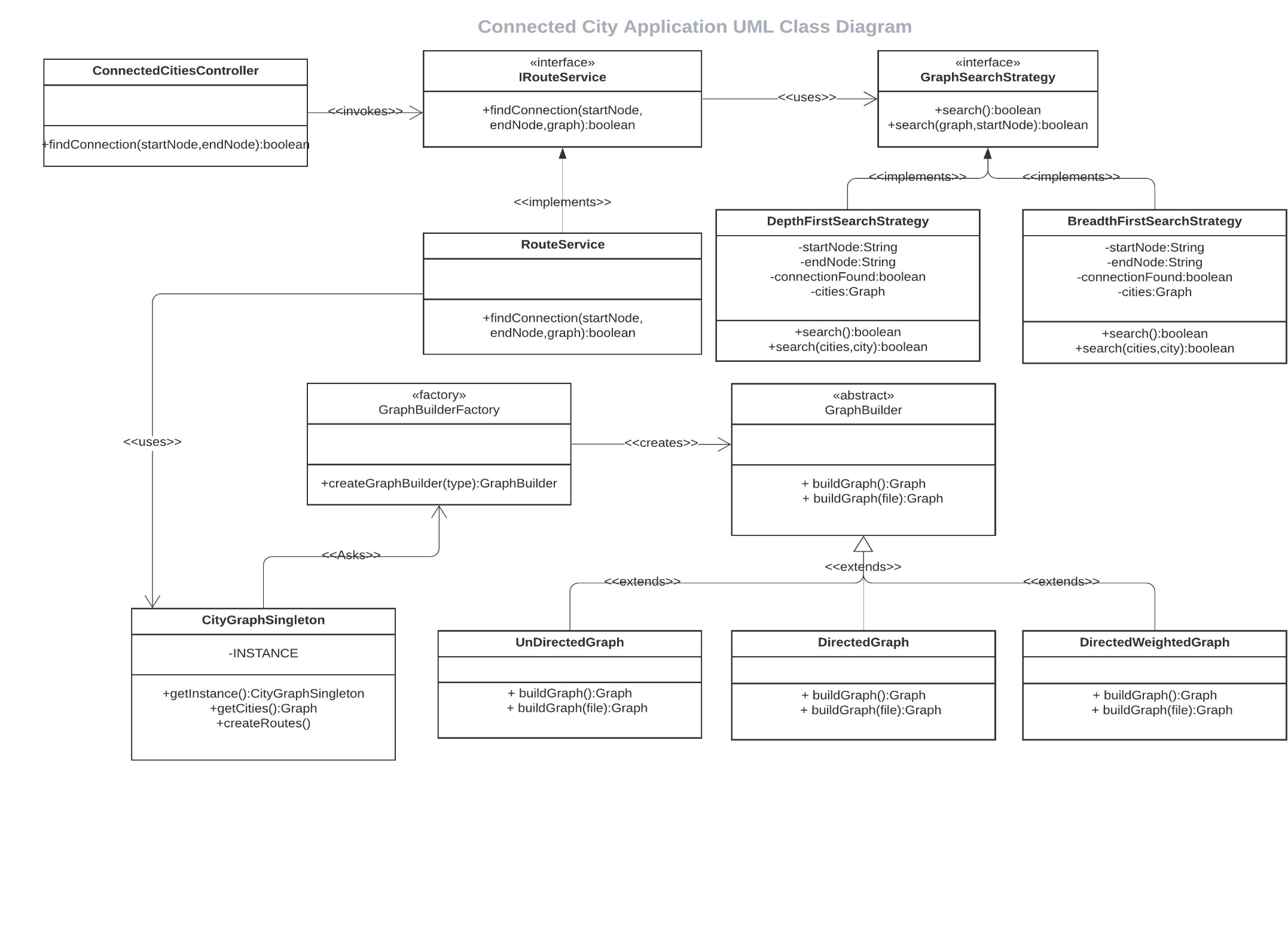
Develop a spring boot based web application to resolve whether given two cities are connected. The application will expose well defined RESTful Endpoints which accepts origin and destination cities as input and will find out whether path exist between the given cities.

1. **System Design**

The system design revolves around 3 key J2EE design patterns and it resolved most of the system wide design issues and it is flexible and extensible to adopt future demand and requirements.

* Singleton
* Factory Pattern
* Strategy Pattern

The following diagram depicts the core classes in the application and its interaction with other classes in the application.



During the startup of the application it will look for cities.txt file in resource folder of the application (resource folders are includes in the class path by default in spring boot ) and it bootstrap the process of creating the necessary memory structure and later on to use the Graph structure to find out whether path exist between 2 cities or not.

The load process will create a Undirected Graph structure in the memory and the singleton pattern plays the key role to hold only one copy of the Graph Structure in the memory at any time.

During the Graph Structure creation in the memory the systems uses Factory Pattern to create any of the following graph structure

* Undirected
* Directed
* Directed Weighted

Once the system up and running it will start accept any number of concurrent request to find out the connection between inputted cities.

The Strategy Design pattern help to choose which graph traversal algorithm to user to search the graph.

The system uses ThreadLocal variable to hold the state of the so far traversed paths in the graph and it has its own copy of the state machine for each and every individual requests.