**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**School of Information and Communication Technology**

*Hanoi, June 8, 2020*

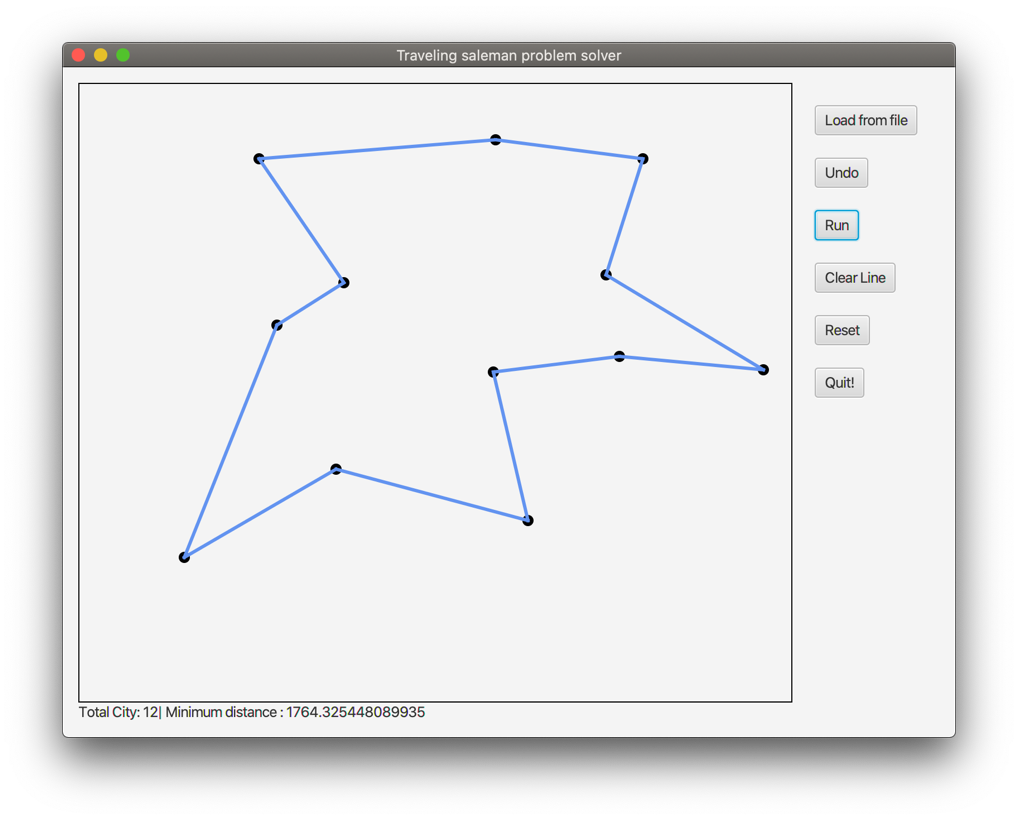
**OBJECT – ORIENTED LANGUAGE AND THEORY (JAVA)**

**MINI – PROJECT REPORT**

1. **Introduction:**

Problem:

The **travelling salesman problem (TSP)** asks the following question: "Given a list of cities, and their locations, what is the shortest possible route that visits each city and returns to the origin city?"

.

Team members:

* Nguyễn Văn Lực – 20176812
* Vũ Ngọc Anh – 20176685
* Vũ Tùng Lâm – 20184284

1. **Mini-project description:**

**Requirements**:

* Fully functional GUI with simple design
* Function to input cities from user (mouse clicks)
* Giving acceptable results

**Use case Diagram:**

**A picture containing text, map

Description automatically generated**

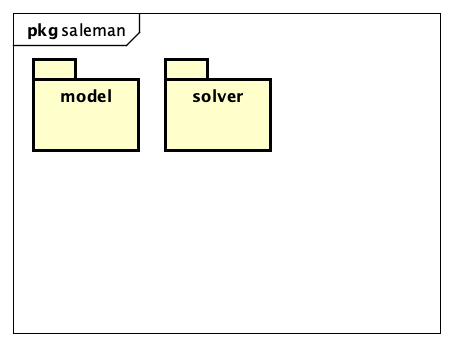
Explanation:

* ­*Load City:* The user will add cities by mouse clicking or loading from \*.txt file in a specific format (each line, representing a city, contains two number, seperated by a space, shows the location of the city in 2-dimensional coordinate).
* *Run algorithm:* Start running the program to find the shortest route.
* *Clear current line:* Clear current route so that the user can view and add more city more clearly.
* *Quit:* Quit the program
* *Print Lines:* The program displays the shortest path found when the algorithm finished.
* *Notify the result:* The program display the Total city added by user and the distance of the shortest path found.

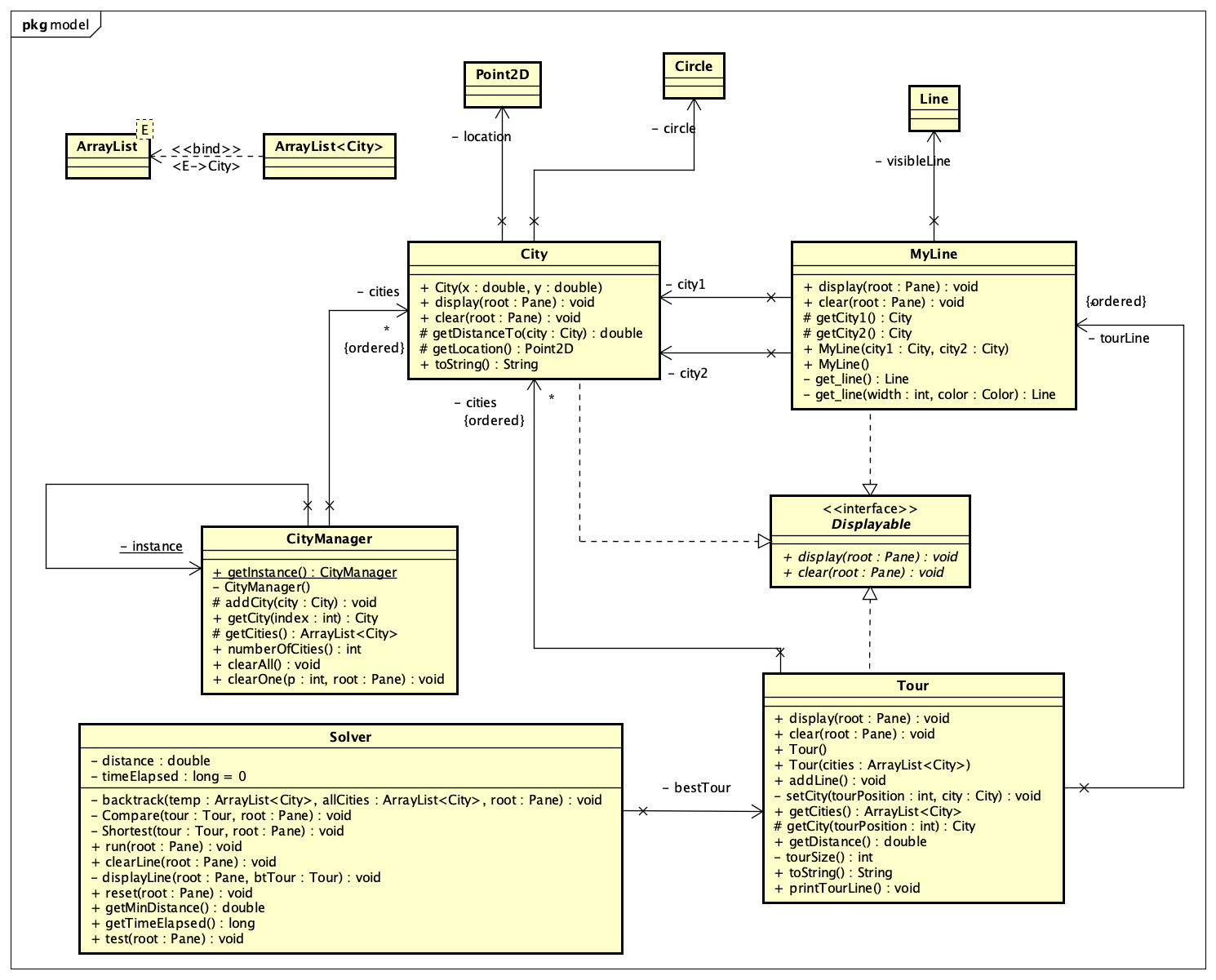
1. **Class diagram and design ideas**

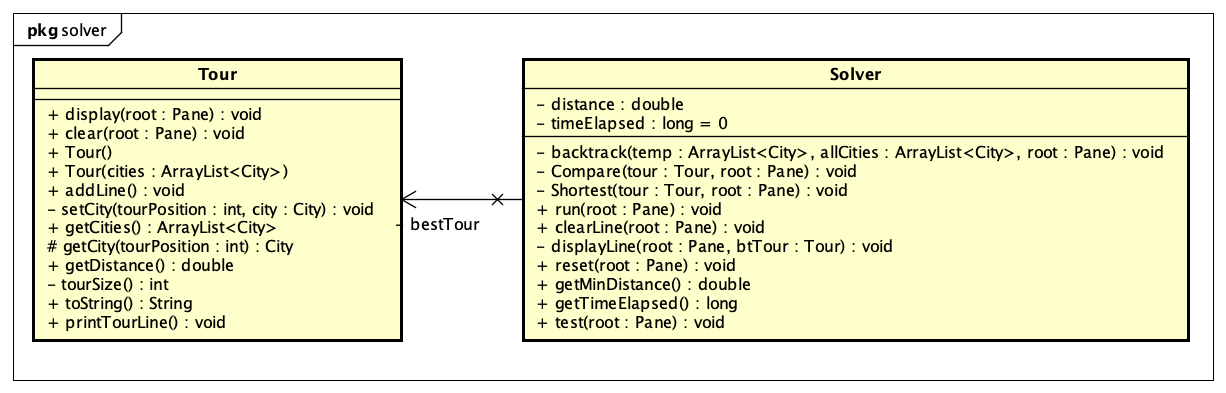
**Class diagram:**

General class diagram:

****

Detailed





Explanation:

* *City:* Represent city added by the user, contains a location in javafx.geometry.Point2D format, a circle in javafx.scene.shape.Circle format, its methods are:
  + Double getDistance(city *c*): return distance between itself and a city *c* passed.
  + Point2D getLocation() : return its location in Point2D format;
* *CityManager:* A class to manage cities added by the user.
* *MyLine:* Create a *visibleLine* in javafx.scene.shape.Line format to connect between two cities.
* *Tour:* Represent a tour, created by an ArrayList of cities, also include an ArrayList of *MyLine*  in order to be displayed. Its methods are:
  + *void addLine():* add *MyLine* to *Tour* when the Tour needed to be displayed.
  + *Int tourSize():* Return the size of the ArrayList *cities*.
* *Solver:* Represent the solving algorithm to find the shortest route. Its methods are:
  + *void backtrack(ArrayList<City> temp, Tour tour) :* generate permutation of *tour* using backtracking algorithm.
  + *void Compare(Tour tour):* Compare the *tour* with the current *bestTour.*
  + *void Shortest(Tour tour):* Find the shortest path and add *Lines* represent the tour into the *bestTour*.
  + *void reset(Pane root):* clear the Pane and start from the begin.
  + *void run(Pane root) :* run the algorithm and display the result to the Pane *root.*
* *Displayable*: interface represent the properties to be displayed and cleared.

1. **Assignment of members:**

* Nguyễn Văn Lực – 20176812: Implements City, CityManager class, Displayable interface and its methods in sub-classes, Solver.run(Pane), add ArrayList<MyLine> tourLine and its methods to Tour class; GUI on App.java.
* Vũ Ngọc Anh – 20176685: Implements MyLine class, draws diagrams.
* Vũ Tùng Lâm – 20184284: Implements Tour and Solver class with backtracking algorithm.

We have used the idea of this source code and modify it to create City and Tour class:

<https://github.com/mhrimaz/TSPVisualizationFX/>