**Safety Check**

**4.0**

**April 10th, 2018**

**Group 09**

**Shuo He, Baikai Wang, Amy Xu, Yunfei Yang**

**Computer Science 2XB3**

**Computer Science Practice and Experience: Binding Theory to Practice**

**Department of Computing and Software**

**McMaster University**

**Revision History**

The first significant revision is that the modification of balanced binary search tree. We add a parent node for the case that the node is not in the graph.

The second significant revision is about the graph. Initially our nodes are set as the integers. However, we eventually decided to change our input as the CrimeADT.

The third significant revision is we add the method getNumber() in CrimeADT to adapt the modification of the graph.

The last significant revision is we add more functions on the interface. It upgraded as the one can accept much more user requirements.

**Team Members and Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student Number | Role | Responsibility |
| Shuo He | 400023520 | Leader | Clarifying the direction of the whole project, designing the requirements of the project, helping members to solve difficulties. |
| Baikai Wang | 400084727 | Normal Member | Organizing the datasets, Deciding the way of implementing the framework of the project. |
| Amy Xu | 400079409 | Normal Member | Modifying the project to make it better, designing the interface, choosing the proper algorithms. |
| Yunfei Yang | 400049426 | Normal Member | Documenting the process of working, conceiving the methods needed in each class, testing the code to find the errors and defects. |

By virtue of submitting this document we electronically sign and date thatthe work being submitted by all the individuals in the group is their exclusive work as a group and we consent to make available the applicationdeveloped through [CS] or [SE]-2XB3 project, the reports, presentations,and assignments (not including my name and student number) for futureteaching purposes.

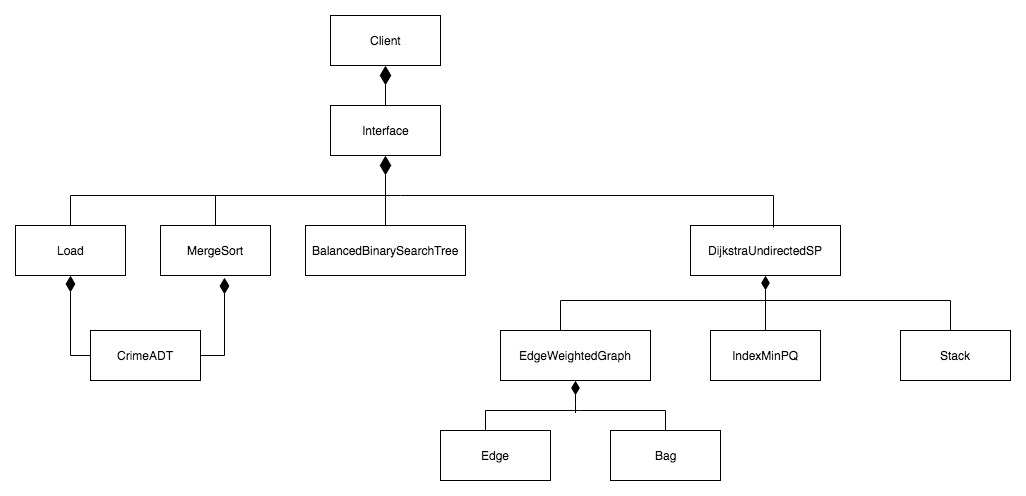
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Contributions | Comments |
| Shuo He | Leader | CrimeADT class | Modifying the interface to the final version.  Help other group members in each step. |
| Load class |
| Interface |
| Baikai Wang | Normal Member | Organizing the data | Organizing the data together with Amy Xu. |
| Edge class |
| Edge weighted graph |
| DijkstraSP class |
| In-code comments |
| Amy Xu | Normal Member | Organizing the data | Organizing the data together with Baikai Wang.  Finishing the primary version of the interface. |
| Balanced binary search tree class |
| The UML diagram |
| Interface |
| Yunfei Yang | Normal Member | Merge class | Reorganizing the data after the first editing. |
| Organizing the data |
| JUnit test case |
| JavaDoc comments |
| Presentation slides |

**Executive Summary**

Our project is designed to protect the user from potential danger in the nearby area of Chicago. We use an interface to accept the address of the user, and then return proper crime cases happened near his/her address. The user can also choose what he/she wants by selecting specific crime types and happened range. We use the CrimeADT to represent the crime from the datasets and sort the datasets in the order of addresses. We use balanced binary search tree and edge weighted graph to map the crime cases. And the Dijkstra algorithm is used for returning the information satisfies the requirements. The special feature of our project is that it can return the relatively frequent-happened crime cases to the user, which tells him/her the specific condition of that area. It is very useful for diverse kinds of people to evaluate the degree of danger they will face to.

**Table of Content**

1. **UML and Description of the classes**



* CrimeADT.java

This class represents a occurred crime with some properties, such as the location of the crime occurred, crime number and crime type etc.

* Load.java

This class loads the information of all crimes into a CrimeADT array.

* MergeSort.java

This class is used to sort the CrimeADT array based on the location

* BalancedBinarySearchTree.java

This class builds a balanced binary search tree to store the CrimeADT and use search method to search for a specific CrimeADT based on the key, location.

* EdgeWeightedGraph.java

This class will uses the Edge.java and Bag.java. Each node in the graph represent a CrimeADT and the edge will be the distance between two crimes’ location based on the latitude and longitude.

* DijkstraUndirectedSP.java

For this class, EdgeWeightedGraph class will be used in this class, because this class is used to find the path between two node in a graph.

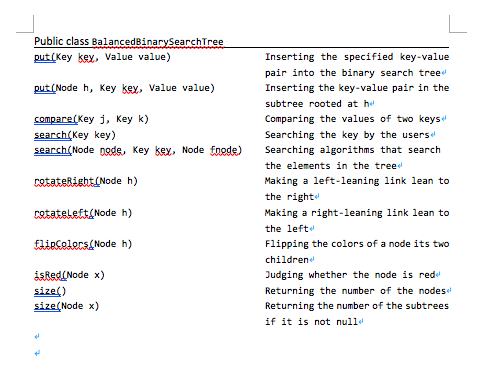
* Interface.java

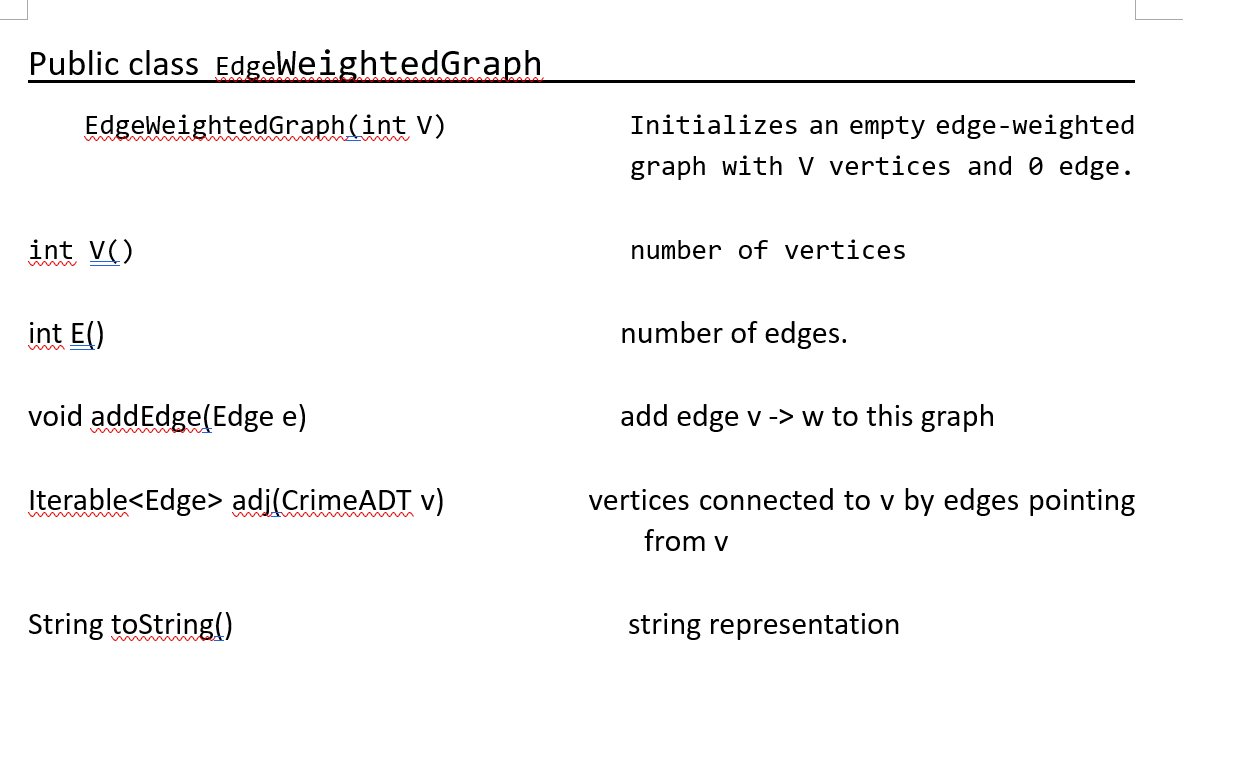
In this class, it will provide a interface for the user to input information. In the interface, there is a textfield for the user to enter the street name and street number. Then, there are checkbox about the crime types, users can check the types they concerned about. Also, some radio buttons for user to choose the searching range. After selecting all the choice, users can click the search button, then program will display number of crimes matched and details of the top six important crimes on the interface.

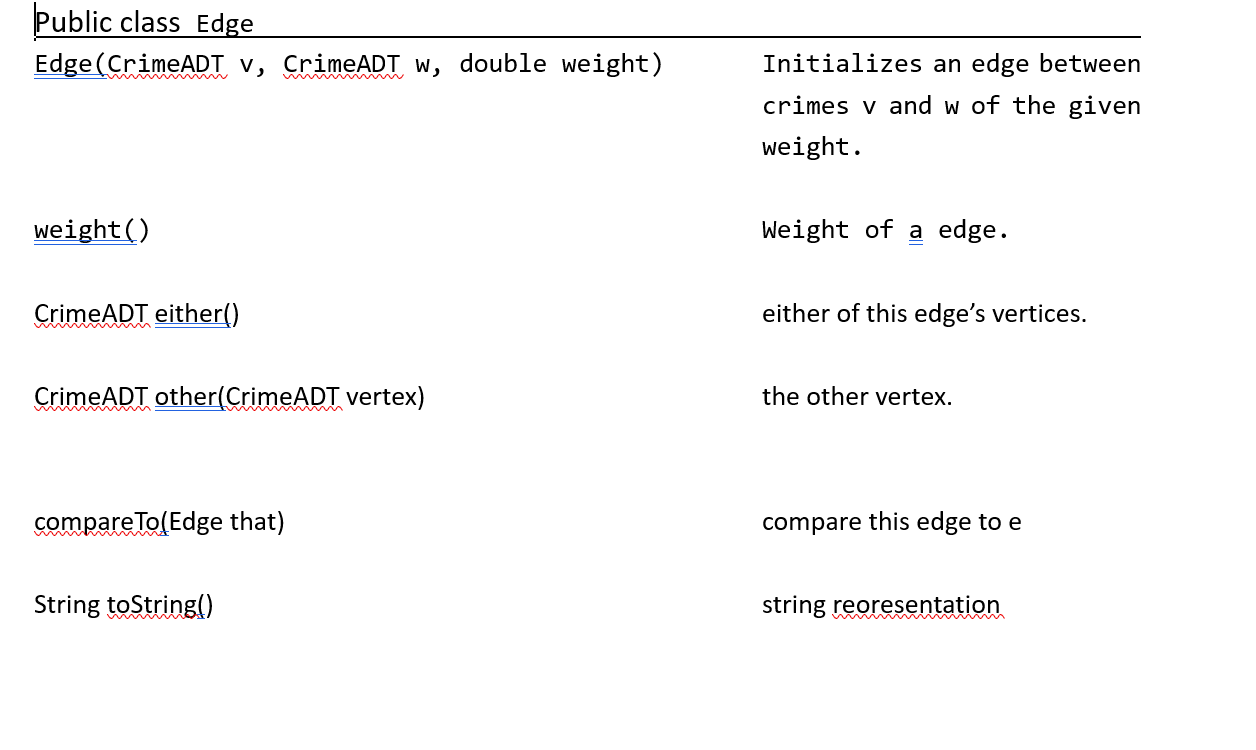
* Client.java

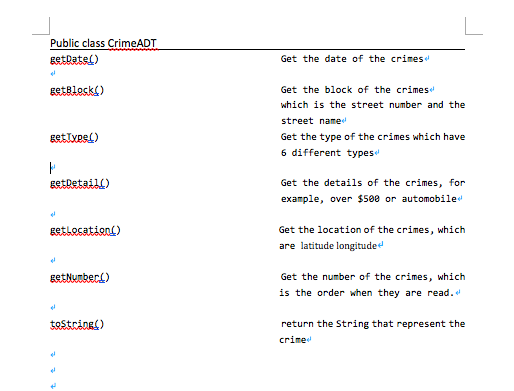
Basically, this class is used to execute the interface class.

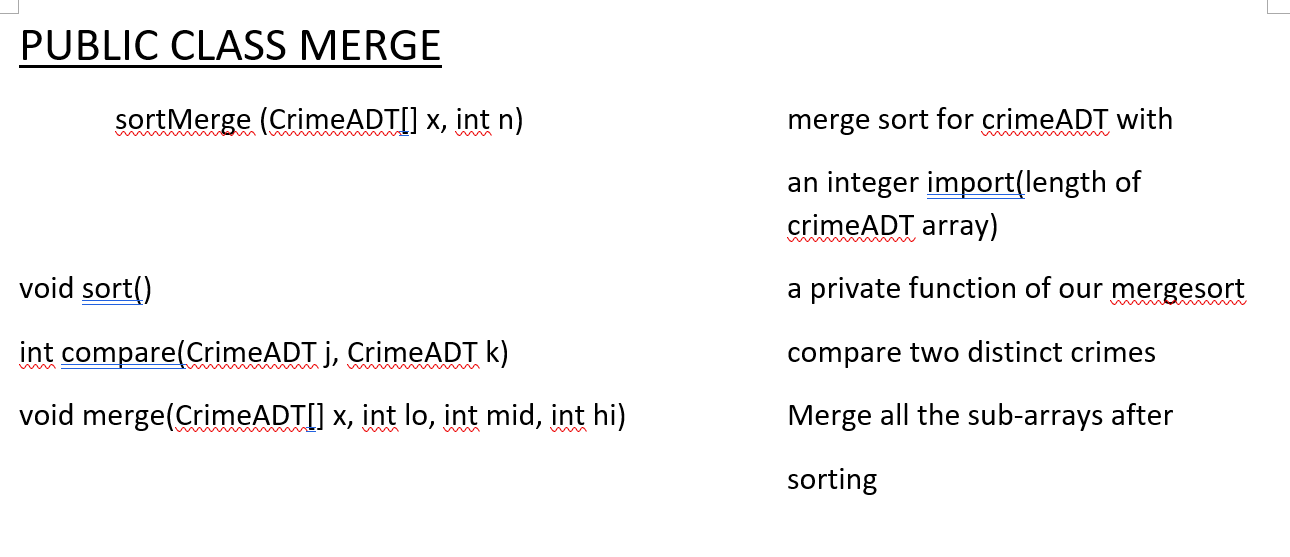
1. **API of classes (public entities)`**











1. **Description of the implementation of private entities**

Balanced Binary Search Tree:

RED represent the node is red

BLACK represent the node is black

The RED and BLACK are the keys of the red-black tree, which trace the node and tell the program whether they are valid or not. And the private method, rotate left, rotate right, flip colors are based on these two important variables.

In Balanced Binary Search Tree, we changed the search method to make the program work better. Because the crimes cannot happen at any place in Chicago, some address will not be recorded in our data set. So, when we search the keys in BBST, we record the parent node of the current, if the search miss, we will return the father node, which guarantees there will be an address. Also, this kind of trick could return the nearest address with users’ requirement.

1. **Trace back to the requirements**

Basically, we followed our requirements of the program, which we could meet the require of our users, and provide them with useful information. The users can chose which kind of crimes that they want to check, and also the distance from themselves. However, we did not make the idea of providing a better place for the user to go. It is the pity of our program.

1. **Evaluation of the design**

The design of this program is basically useful in our daily life, as there are so many unstable and dangerous events happened around us, to check what happens will be very important. Although there are still some improvements that we should do, for example, the interface of the program is very basic, if possible, the mobile phone app will be the interface in the future.