Gebze Technical University

Computer Engineering

CSE222-2020-SPRING

Homework-8_part2 Report
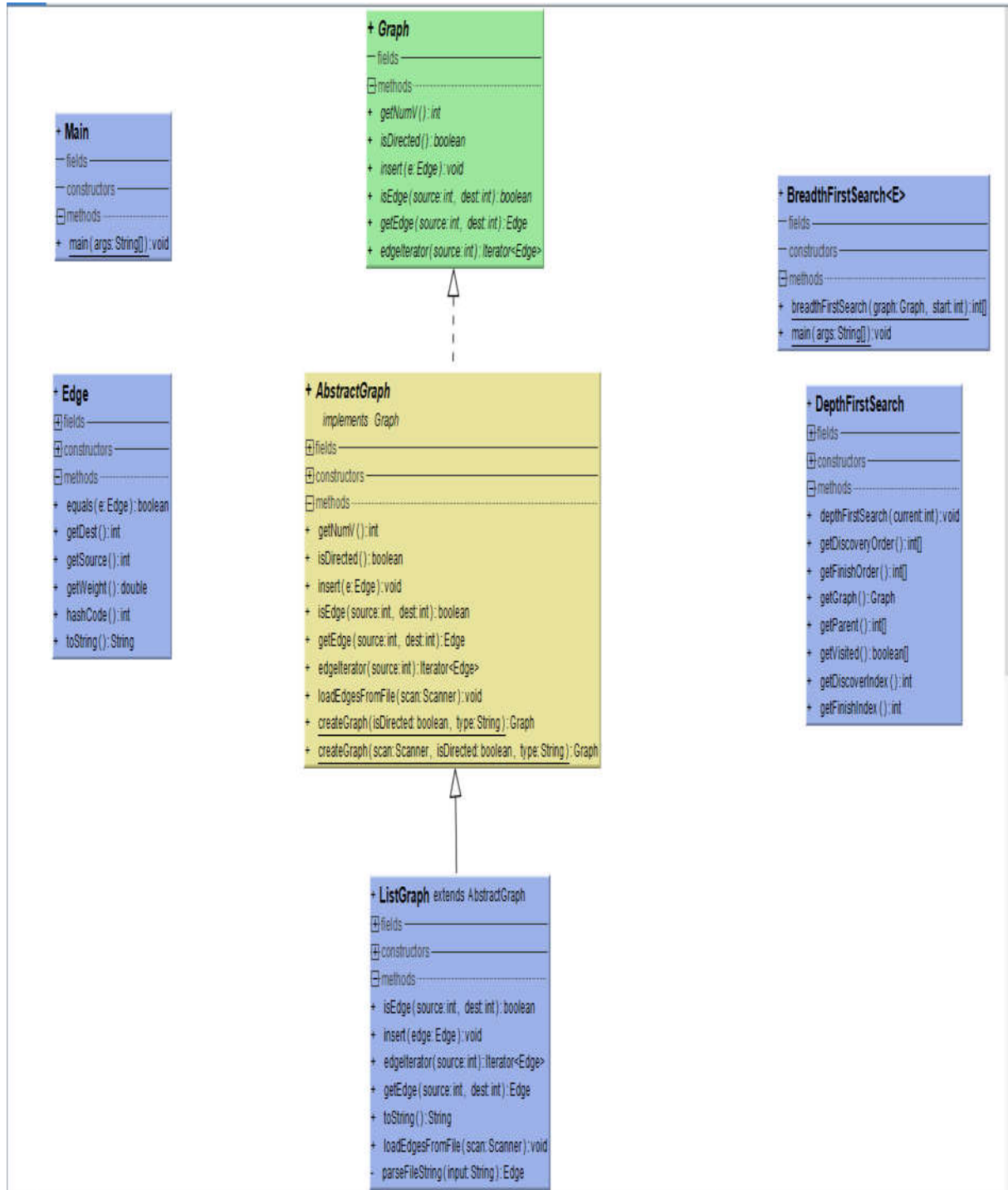
Ferdi Sönmez

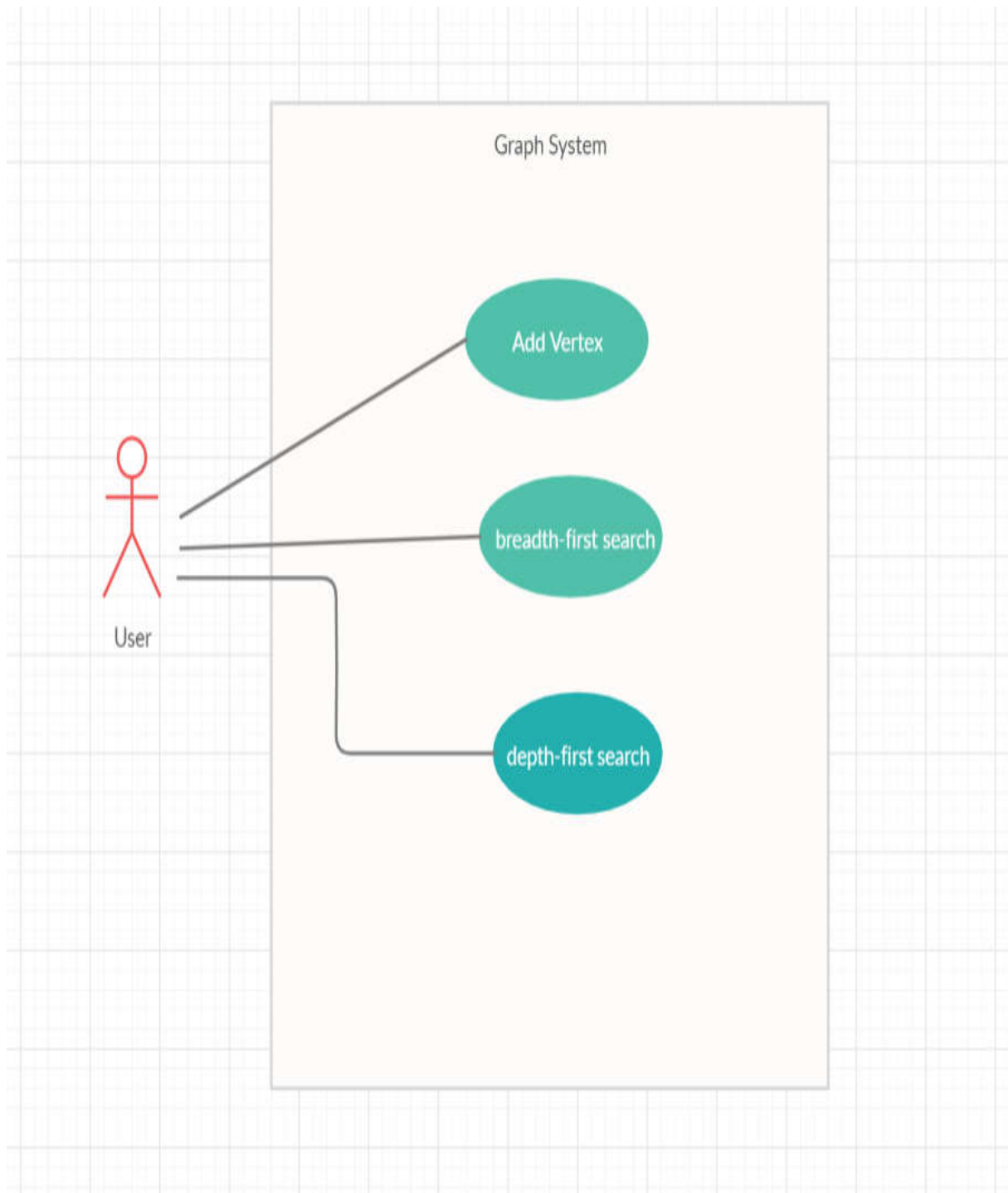# 161044046

1)Problem Solutions Approach

Reads the selected file contents and puts it into the link list structure as 2-D. After the linked list structure is filled, DeepFirstSearch and Breadthfirstsearch operations can be run according to the user request.
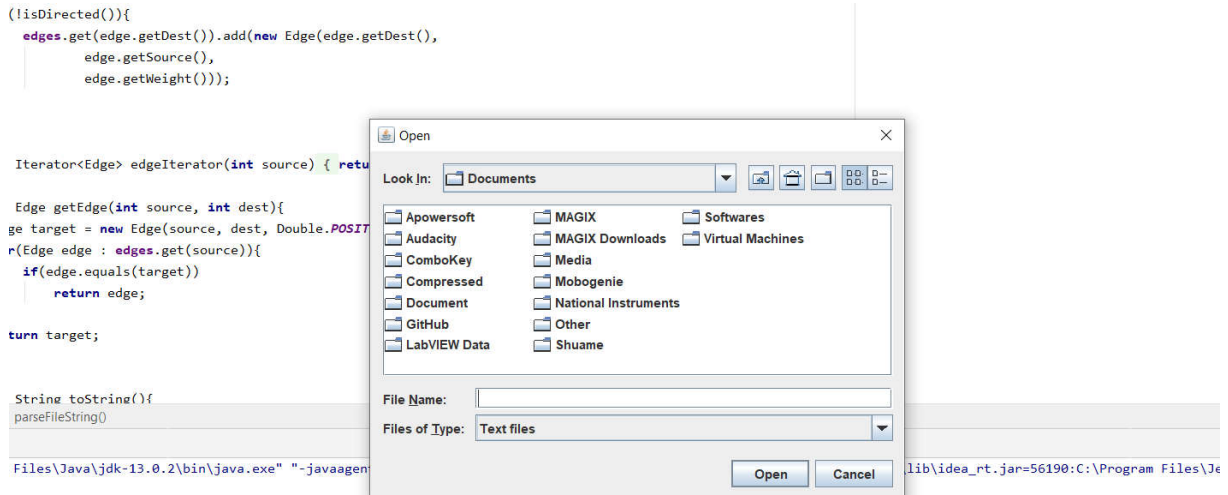
2) Class Diagram

Graph (interface)
- fields
- methods
  - + getNumV() : int
  - + isDirected() : boolean
  - + insert(e:Edge) : void
  - + isEdge(source:int, dest:int) : boolean
  - + getEdge(source:int, dest:int) : Edge
  - + edgeIterator(source:int) : Iterator<Edge>

Main
- fields
- constructors
- methods
  - + main(args:String[]) : void

BreadthFirstSearch<E>
- fields
- constructors
- methods
  - + breadthFirstSearch(graph:Graph, start:int) : int[]
  - + main(args:String[]) : void

Edge
- fields
- constructors
- methods
  - + equals(e:Edge) : boolean
  - + getDest() : int
  - + getSource() : int
  - + getWeight() : double
  - + hashCode() : int
  - + toString() : String

AbstractGraph implements Graph
- fields
- constructors
- methods
  - + getNumV() : int
  - + isDirected() : boolean
  - + insert(e:Edge) : void
  - + isEdge(source:int, dest:int) : boolean
  - + getEdge(source:int, dest:int) : Edge
  - + edgeIterator(source:int) : Iterator<Edge>
  - + loadEdgesFromFile(scan:Scanner) : void
  - + createGraph(isDirected:boolean, type:String) : Graph
  - + createGraph(scan:Scanner, isDirected:boolean, type:String) : Graph

DepthFirstSearch
- fields
- constructors
- methods
  - + depthFirstSearch(current:int) : void
  - + getDiscoveryOrder() : int[]
  - + getFinishOrder() : int[]
  - + getGraph() : Graph
  - + getParent() : int[]
  - + getVisited() : boolean[]
  - + getDiscoverIndex() : int
  - + getFinishIndex() : int

ListGraph extends AbstractGraph
- fields
- constructors
- methods
  - + isEdge(source:int, dest:int) : boolean
  - + insert(edge:Edge) : void
  - + edgeIterator(source:int) : Iterator<Edge>
  - + getEdge(source:int, dest:int) : Edge
  - + toString() : String
  - + loadEdgesFromFile(scan:Scanner) : void
  - + parseFileString(input:String) : Edge
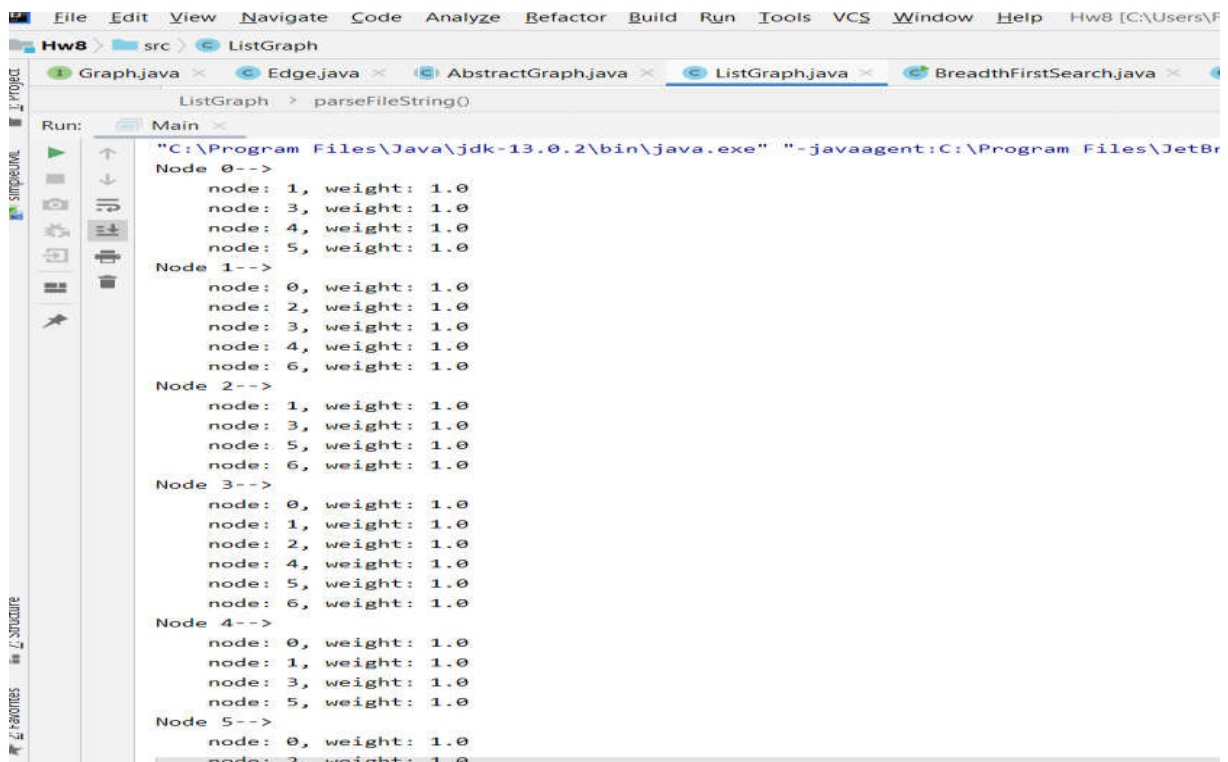
3)Use Case Diagram

4)Test Case

a) The user is asked which file he wants to choose.

b) Graph structure created after reading the file.

c) User can do Deep-First-Search if he/she wants.

```
***DepthFirstSearch***
Discovery and finish order:
0 6
1 5
2 4
3 3
4 2
5 1
6 0
```

d) User can do Breadth-First-Search if he/she wants.

```
***BreadthFirstSearch****
Node and Parent in tree:
0 -1
1 0
2 1
3 0
4 0
5 0
6 1
```