**GIT Department of Computer Engineering**

**CSE 222/505 - Spring 2022**

**Homework # Report**

**Student Name: Muhammed Alperen Karaçete**

**Student Number: 171044052**

1. **SYSTEM REQUIREMENTS**

User:

User can create new vertex.

User can add given vertex to graph.

User can add edge between two vertices.

User can remove edge between two vertices.

User can remove a vertex with ID.

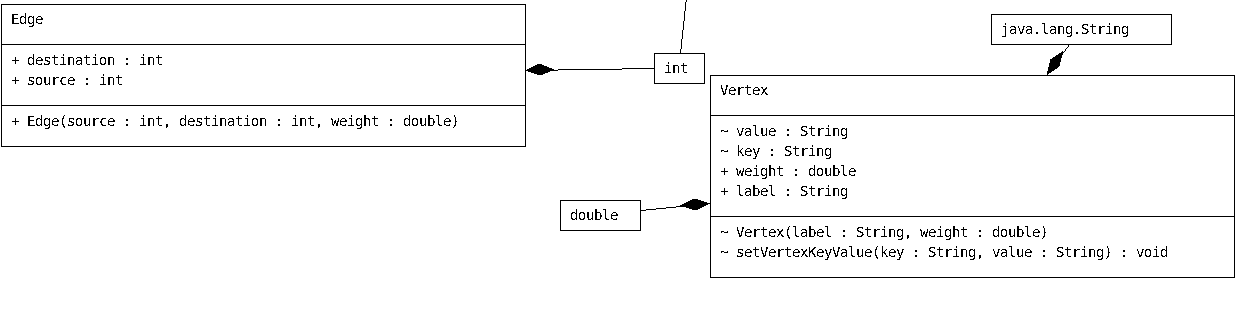
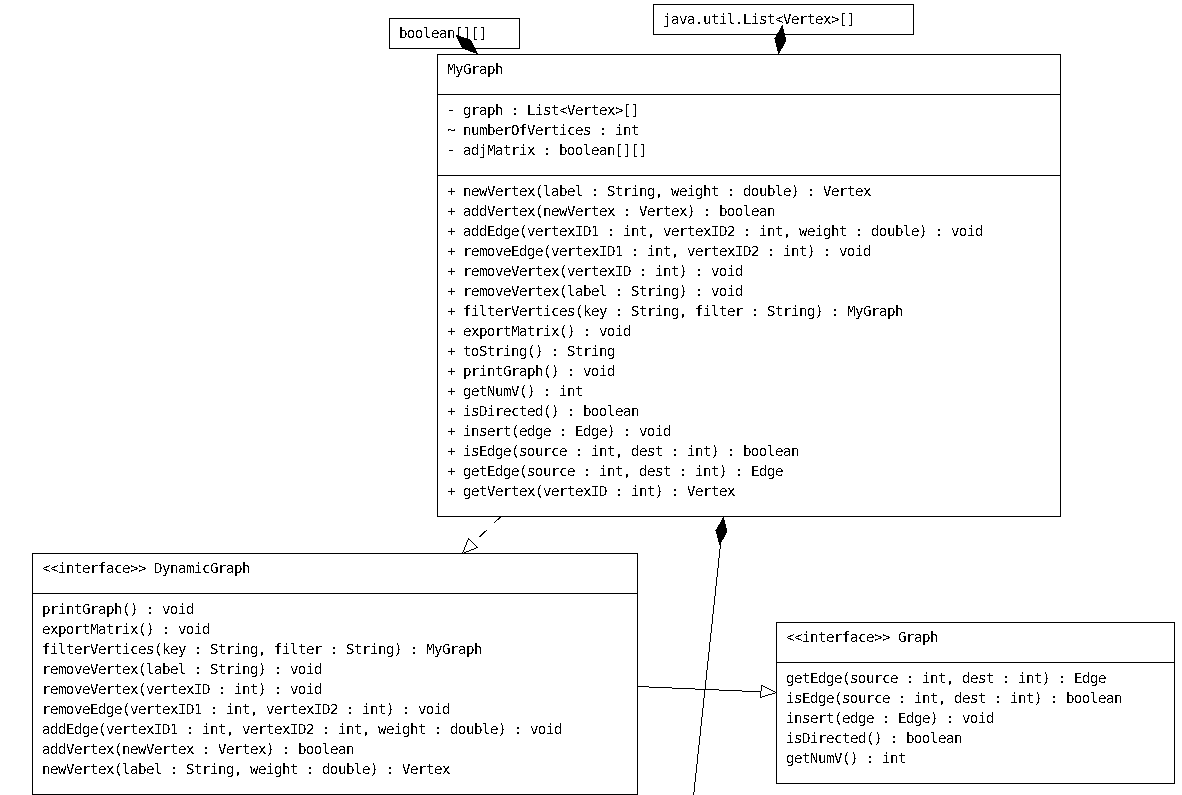
User can remove a vertex with label

User can filter verticex with given key-value values.

User can generate the adjacency matrix representation of the graph.

User can print the graph in adjacency list format.

1. **CLASS DIAGRAM**

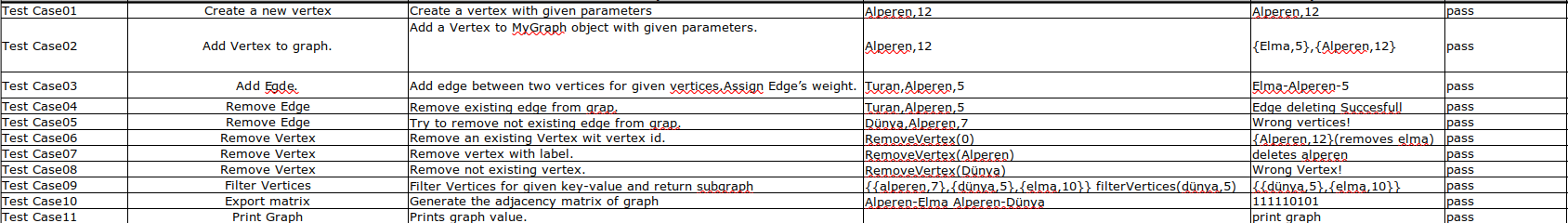


1. **PROBLEM SOLUTION APPROACH**

Question1:

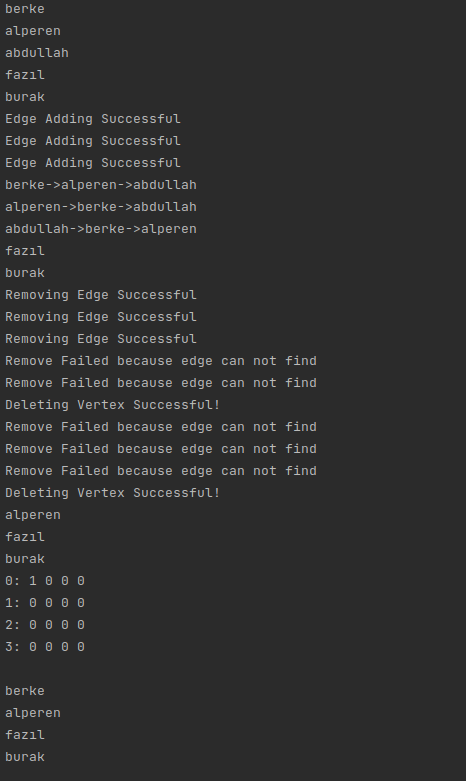
*Firstly I implemented Graph and DynamicGraph interfaces.Then I implemented MyGrap class with this interfaces.I created a list of vertex array named by graph to take my graph informations in this list array.*It’s size is one.I am incrementing it’s size for every vertex adding operation.I took vertices on array size of graph data and I am hiding edge values in my array list.So graph[0].get(0) represents my first vertex.grap[0].get(1) represents edge which is connected to my vertice.graph[1] represents second vertex.The hardest part for me was filtering the vertices.In tis function I coppied my graph values to sub graph value.With that operation I coppied all my vertecises and edges to another sub graph.Then I removed any other non required vertices from it so edges too are removed from subgraph.Then I returned it.I created edge and vertex class too.I use vertex class alot but I didnot use edge class.Just create it for a proper view.

1. **TEST CASES**



1. **RUNNING AND RESULTS**





Time Complexity:

newVertex (string label, double weight):= Teta(1) Returns Vertex.

addVertex (Vertex new\_vertex): O(n) for loop

addEdge (int vertexID1, int vertexID2, double weight): Teta(1) only assign.

removeEdge (int vertexID1, int vertexID2): Teta(1) only remove.

removeVertex (int vertexID): O(n) for loop

removeVertex (string label): O(n^2) 2 for loop inside

filterVertices (string key, string filter): O(n^2) 2 for loop inside

exportMatrix(): O(n^2) 2 for loop inside

printGraph(): O(n^2) 2 for loop inside