1. Introduction:

This document is about the group project for "Object Oriented Programming with

Java - Advanced" course in winter semester 2019-2020. By the use of advanced object-oriented concepts in Java programming language, the aim of this project is to create a program has fully functional, meet the given requirements and provides a good documentation for users can install, run and see the expected results.

The intended audience of this document is the course instructor, who will use it as the basis for a determination of a portion of our grade. The developed project is based on basic graph definitions, people who are interested in graph can use this as a tool to refer some specific properties of graph.

The secondary goal is to collaborate among team members, assign tasks to members, manage and plan the project to implement the project on time and effectively.

2. Work distribution:

3. Project documentation:

3.1. Technical description:

In this project, a basic analysis of a graph-based communication network model had been implemented, from the input of a model over the appropriate processing of the model data to the output of the information. In particular:

- Reading communication network information from a network model ﬁle (graphml).

- Processing the network model data and acquisition of the necessary information in accordance with the above communication network properties.

- Output the results of the processing into the command line interface (CLI) or the file specified by user.

Given network model has basic properties of graph, this project is possible to:

- Outputs number of nodes, edges and their identities (IDs)

- Determines whether the model is connected or not and calculate its diameter

- Prints shortest path between two vertices according to the Dijkstra algorithm

- Calculates the betweenness centrality measure for a selected node.

The program has to meet following requirements:

- The program was developed using an adequate object orientation.

- Information will be stored in suitable Java Collections or appropriate alternative data structures.

- The program contains adequate error handling.

- The program works with streams and ﬁles. At least one reading and writing ﬁle access has to be made.

- The program contains at least two threads.

- The program is developed using the clean code standard(s) as presented in the lecture.

- Appropriate logging is provided using java.util.Logging or comparable.

3.2. Things have been implemented:

Discussion with course instructor to make the project scope clear. Discussion among team members about project requirements, sample input files, graph implementation and computations were proceeded. Tasks are created and assigned to members. In details:

- Input processing:

+ Reading input from users and parse arguments

+ Load network model from user-specified file using regular expression

+ Store the given graph model by using HashMap in Java Collections

- Algorithm to determine connectivity, calculate diameter

- Output number of nodes, edges and their identities (IDs)

3.x: Những hạn chế và khó khăn đang gặp phải:

The program has to meet following requirements:

-

Data flow diagram: