ngOC KIEU THANH huYNH – 2688093

Bilal Butt – 2688700

Raima khan –2692686

Armin ROSHAN- 2487128

Design document

GROUP MEMBERS:

Contents

[INTRODUCTION 2](#_Toc451436073)

[USER INTERFACE 3](#_Toc451436074)

[FUNCTIONAL REQUIREMENTS (USE-CASES) 6](#_Toc451436075)

[USE CASE: ADD COMPONENTS 6](#_Toc451436076)

[USE CASE: ADD A PUMP 6](#_Toc451436077)

[USE CASE: ADD AN ADJUSTABLE SPLITTER 7](#_Toc451436078)

[USE CASE: ADD PIPELINE BETWEEN TWO COMPONENTS 7](#_Toc451436079)

[USE CASE: REMOVE PIPELINE. 8](#_Toc451436080)

[USE CASE: REMOVE COMPONENT 8](#_Toc451436081)

[USE CASE: CREATE A NEW NETWORK DRAWING FILE 9](#_Toc451436082)

[USE CASE: OPEN A NETWORK DRAWING FILE 9](#_Toc451436083)

[USE CASE: SAVE AS A NETWORK DRAWING FILE 10](#_Toc451436084)

[USE CASE: SAVE A FILE WHICH THE NAME ALREADY EXISTS 10](#_Toc451436085)

[USE CASE: SAVE A NETWORK DRAWING FILE 10](#_Toc451436086)

[USE CASE: CLOSE A NETWORK DRAWING FILE 11](#_Toc451436087)

[USE CASE: EXIT APPLICATION 11](#_Toc451436088)

[NON-FUNCTIONAL REQUIREMENTS (OTHER REQUIREMENTS) 12](#_Toc451436089)

# INTRODUCTION

This document is the design document for building a flow network application.

In the first section of this document, the class diagram of application are represented by diagrams and their brief description. The second section presents some sequence diagrams.

# CLASS DIAGRAM

## Class diagram

## Description of the classes and their members

#### COMPONENT CLASS

|  |  |  |
| --- | --- | --- |
| Properties | | |
| ID | Integer | This is unique number to indicate the components of each network drawing. It is increased automatically when the component is added |
| Location | Point | Represents an ordered pair of integer x- and y-coordinates that defines a location of components on the screen |
| Operations | | |
| Component(id: int, location: Point): constructor of the class | | |

#### PUMP CLASS

|  |  |  |
| --- | --- | --- |
| Properties | | |
|  |  |  |
|  |  |  |
|  | | |
|  | | |

# SEQUENCE DIAGRAMs

## Draw a component



Description:

This sequence diagram describe how a component to the drawing screen and add

that component to the list of components belonging to the object currentNetwork. The drawComponent method has three argument parameters which include objects of type Graphic, ImageList and Component. The Graphic object can come from the Form's PaintEventArgs or otherwise by creating the Graphic object of the form or a control. The ImageList consists of the images for each of the images for the components that the user would like to place on the drawing screen.

The component object comes from the temComponent property of the System class. This property changes its reference to a new Component object whenever the user clicks on a button for different components on the toolbox. Clicking on a specific component button would create a Component of that type and change the tempComponents reference to that newly created Component object. The component argument in the method is then used to call the method addComponent on the currentNetwork property of the static System class.

The method addComponent takes an argument of type Component which would be the same as tempComponent in this case. The method addComponent then adds the Component to the List of Components property of Network

class. However when the addComponent method is called there is another method call within this method to the method for checkOverlap method which would return a boolean. If the return value is true then the component would be successfully added to the list of components.

Once the component is successfully added to the list of components the Graphic object of the method will use the location property of the component to draw the image from the ImageList corresponding to the component that needs to be drawn on the drawing screen.

## Draw a PIPELINE



This sequence diagram for drawing a pipeline. The drawPipeLine method takes two argument parameters, a Graphics object type and a PipeLine object. The PipeLine object is created whenever the click on pipe-line button in form event is raised. In this event the tempPipeLine property of System class is assigned a new PipeLine object without that PipeLine object having any start and end Component properties assigned.

The next time whenever the user clicks on the screen the event for screeen click goes through various checks that first include to check if the tempPipeLine is empty and if not the nested statements check for the following: whether the startComponent of the tempPipeLine is null and assigns a startComponent to it via the method findComponent, if startComponent is not null and endComponent is null and calling the method findComponent returns a null then the point from the EventArgs is added to that tempPipeLines list of clickLocations and finally if startComponent is not null and endComponent is null and calling the method findComponent returns a non-null value then the startComponent is assigned whichever component is returned by findComponent function unless that component is the same as the startComponent.

Once the PipeLine object given in the drawPipeLines argument is complete in that it at least has a start and end Components that are non-null, then the Graphic object is used to draw the PipeLine according to the startComponent and endComponent locations as well as the locations given in the list of clickLocations of the PipeLine object from the parameter.