ngOC KIEU THANH huYNH – 2688093

Bilal Butt – 2688700

Raima khan –2692686

Armin ROSHAN-

USER requirements specification

GROUP MEMBERS:

Contents

[INTRODUCTION 2](#_Toc449719392)

[Functional Requirements (use-cases) 2](#_Toc449719393)

[USE CASE: ADD COMPONENTS: 2](#_Toc449719394)

[USE CASE: REMOVE PIPELINE. 3](#_Toc449719395)

[USE CASE: REMOVE COMPONENT 3](#_Toc449719396)

[USE CASE: CREATE A NEW NETWORK DRAWING 3](#_Toc449719397)

[USE CASE: CREATE A NEW NETWORK DRAWING FILE 4](#_Toc449719398)

[Use Case: OPEN A NETWORK DRAWING FILE 4](#_Toc449719399)

[Use Case: SAVE A NETWORK DRAWING FILE 4](#_Toc449719400)

[Use Case: SAVE As A NETWORK DRAWING FILE 5](#_Toc449719401)

[Use Case: CLOSE A NETWORK DRAWING FILE 5](#_Toc449719402)

[Use Case: EXIT APPLICATION 6](#_Toc449719403)

[User Interface 7](#_Toc449719404)

[Non-functional Requirements (other requirements) 8](#_Toc449719405)

# INTRODUCTION

This document is the user requirements specification for building a flow network application. This application allows user make a drawing network for the fuel transported systems. This application provides the functionalities to build a network containing components like pump, sink, splitter, adjustable splitter and merger. Every kind of components have their own attributes and functionalities, the application helps user connect them and manages the amount of fuel of every pipeline. Besides, indicating the exceeded pipeline also need to be done.

In first section of this document, the functional requirements is represented by use-cases. The second section mentions about user interface design of application. The last section introduces some un-functional requirements of this application.

# Functional Requirements (use-cases)

All use cases have the system user as the only actor involved, furthermore all the use cases are of the sea-level.

## USE CASE: ADD COMPONENTS:

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**MSS:**

1. User picks the component they want to place from the list of components in the toolbox on the right side of the form.
2. System makes the selected component as the actively chosen component (eg. Component is highlighted in toolbox).
3. User clicks on the position he wishes to place the component on in the drawing screen.
4. System draws a copy of the selected component on the drawing screen at the position chosen by the user.

**Exceptions:**

1.a. The component chosen is a pump.

1. System prompts user for max flow and current flow to assign to the pump. (can be implemented in various ways including on bottom of toolbox)
2. User assigns values for max and current flow.
3. Continue from step 2 in MSS

1.b. The component chosen is a adjustable splitter.

1. System prompts user for upper and lower output flows to be assigned to the adjustable splitter. (can be implemented in various ways including on bottom of toolbox)
2. User assigns values for upper and lower output flow.
3. Continue from step 2 in MSS

## USE CASE: ADD PIPELINE BETWEEN TWO COMPONENTS

**Pre-condition**: There must exist at least two unconnected components on the drawing screen.

1. User clicks on starting component for the pipeline.
2. System chooses the selected component as the starting point of pipeline.
3. User may click on one or more points on the drawing screen that do not have a component on them.
4. System draws lines connecting each point on the screen clicked with the previous point clicked.
5. User ends pipeline by clicking on the final component on the drawing screen.
6. System selects the final component as ending point of the pipeline and labels the line with the current flow going through it.

## USE CASE: REMOVE PIPELINE.

**Pre-Condition:** The drawing screen must have at least two components with at least one pipeline connecting the components.

**MSS:**

1. User selects remove pipeline button on the toolbox on the right side of the form.
2. System goes into delete pipeline mode.
3. User selects the pipeline he/she wants to delete from the drawing screen.
4. System removes the selected pipeline.

## USE CASE: REMOVE COMPONENT

**Pre-condition:** Component must exist on the drawing screen.

**MSS:**

1. User selects remove component button from the toolbox on the right side of the form.
2. System goes into delete component mode.
3. User selects the component to delete from the drawing screen.
4. System prompts user asking if he/she wants to delete the component and gives warning that deleting the component will also remove all connected pipelines.
5. User confirms that they want to delete the selected components.
6. System removes the component and its connected pipelines from the drawing screen.

**Exceptions:**

5.a User do not want to remove the component and its connected pipelines.

.1: At this point, end of this use case.

## USE CASE: CREATE A NEW NETWORK DRAWING

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button open new network drawing.

**MSS:**

1. System shows the new drawing screen.

**Exceptions:**

1.a There is existing network drawing already open

.1: Reference to user case close network drawing.

## USE CASE: CREATE A NEW NETWORK DRAWING FILE

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button add new network drawing.

**MSS:**

1. System shows the new drawing screen.

**Exceptions:**

1.a There is existing network drawing already open

.1: Reference to user case close network drawing file.

## Use Case: OPEN A NETWORK DRAWING FILE

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button open network drawing.

**MSS:**

1. System shows the file dialog
2. User chooses the location of the file or types the name of required file.
3. System loads the file and shows it in drawing screen.

**Exceptions:**

3.a User cannot find the file

.1: User closes the dialog

3.b There is existing file already open

.1: Reference to user case close file.

## Use Case: SAVE A NETWORK DRAWING FILE

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button save network drawing.

**MSS:**

1. System save the file.

**Exceptions:**

1.a The file is new

.1: Reference to the use case save as

1.b The file is already existing .

.1: Overwriting the existing file.

.2: User closes the dialog

## Use Case: SAVE As A NETWORK DRAWING FILE

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button save as network drawing.

**MSS:**

1. System opens the file dialog
2. User chooses the location to save the file
3. User gives the file’s name
4. System save the file in the chosen directory

**Exceptions:**

3.a The file’s name is already existing.

.1: System notifies user that name already exist and prompts user to give new name.

.2: User gives a new name and return to MSS at step 4.

## Use Case: CLOSE A NETWORK DRAWING FILE

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button close network drawing.

**MSS:**

1. System close the file

**Exceptions:**

1.a User has not save the file yet

.1: Reference to the use case save

## Use Case: EXIT APPLICATION

**Pre-condition:** The user has the main form for the GUI of the pipeline app open on his/her PC.

**Trigger**: User clicks on the button exit button or click on the close button of the main form.

**MSS:**

1. System asks user for saving the file
2. Use do not want to save the opening file
3. System closes the application

**Exceptions:**

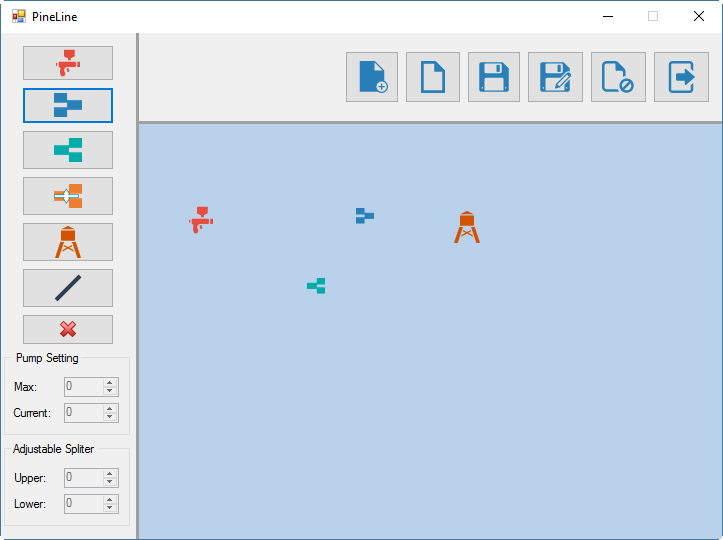
1.a User saved the file previously

.1: Go directly to MSS at step 3

2.a User would like to save the file

.1: Reference to the use case save

# User Interface



# Non-functional Requirements (other requirements)

* The pipeline connections between two components is allowed some points between.
* The components cannot be overlap on each other.
* If the fuel is exceeded the limit fuel, the application need to notify users by changing the color of exceeded pipelines. Different colors can be used there to indicate exceeded pipelines. This helps the application becoming friendlier to use for users.
* This application should be more reliable. For example if user will see a button with name “Add pipeline”, then this will actually add pipeline as expected rather than adding some other component.