

User REquirements Specification



Nikola Nikushev

Bilger Yahov

Lyubomir Dimov

Tao

Table of Contents

[Introduction 2](#_Toc436006935)

[Functional requirements (use-cases) 2](#_Toc436006936)

[1. Create an element 2](#_Toc436006937)

[2. Remove an element 2](#_Toc436006938)

[3. Add a pipeline 3](#_Toc436006939)

[4. Remove a pipeline 3](#_Toc436006940)

[5. To edit an adjustable splitter 3](#_Toc436006941)

[6. Splitting the flow 4](#_Toc436006942)

[7. Clearing the grid 4](#_Toc436006943)

[8. Saving a grid 4](#_Toc436006944)

[9. New File 5](#_Toc436006945)

[10. File Open 5](#_Toc436006946)

[11. Undo last change 5](#_Toc436006947)

[User interface 5](#_Toc436006948)

[Non-functional requirements 6](#_Toc436006949)

# Introduction

“Flowly” is an interesting, clever and very useful application created to manage liquid or gas flows in a system. Using it you can create your own working space or so-called grid. Adding components and connecting them between each other has never been easier than now! You just pick an element from the toolbox and place it to desired position on the grid. For your easiness apart from having the ability to use pumps and sinks, we also added some useful elements as (adjustable) splitters and mergers. Your task is just to create systems that really make sense! You can save your flow diagram and open it up later. Of course our software “Flowly” is really intelligent and can immediately detect if you try to overlap elements. Managing your flow system has never been easier!

# Functional requirements (use-cases)

## Create an element

Goal level: Sea level

Pre: No

Actor: User

Main Success Scenario:

1. User selects and element from the toolbox and places it on the grid.

2. System depicts the element on the desired position on the grid.

Extensions:

1a. The newly created element overlaps an old one.

1a.1 User chooses another place on the grid.

## Remove an element

Goal level: Sea level

Pre: There is at least one element on the work space (grid)

Actor: User

Main Success Scenario:

1. User selects an already drawn element from the grid, right clicks and clicks delete.

2. System shows a message that all the connected lines will be deleted automatically.

3. User clicks “OK”.

4. System deletes the selected element together with the adjacent lines.

Extensions:

3a. User clicks “Deny”.

3a.1 System aborts the operation.

## Add a pipeline

Goal level: Sea level

Pre: There is at least two elements on the work space (grid)

Actor: User

Main Success Scenario:

1. User clicks to pipeline icon from the toolbox.

2. System selects pipeline element.

3. User selects two already drawn elements from the workspace (grid).

4. System depicts the pipeline between the two elements.

Extensions:

3a. System shows a message that it is not possible to add (more) pipelines between these two elements.

3a.1 System aborts the operation.

## Remove a pipeline

Goal level: Sea level

Pre: There is at least two elements on the work space which are connected with a pipeline

Actor: User

Main Success Scenario:

1. User selects an already drawn pipeline from the grid, right clicks and clicks delete.

2. System shows a message asking if the user is sure about the action.

3. User clicks “Yes”.

4. System deletes the selected pipeline.

Extensions:

3a.1 User clicks “Deny”.

3a.1 System aborts the operation.

## To edit an adjustable splitter

Goal level: Sea level

Pre: There is an adjustable splitter placed

Actor: User

Main Success Scenario:

1. User double clicks adjustable splitter

2. System shows window to edit the splitter

3. User edits the splitter

4. System shows the changes made

Extensions:

3a. If the splitter exit is connected to an element.

3a.1 Shows a message warning the user that the change might affect the elements

## Splitting the flow

Goal level: Sea level

Pre: There are at least one pump and one sink in the network

Actor: User

Main Success Scenario:

1. User chooses splitter or adjustable splitter and adds it to the network

2. System draws the splitter

3. User makes connection between pump, splitter and sink.

4. System splits the flow between the upper and lower output of the splitter

Extensions:

4a. Splitter is adjustable splitter

4a.1 User can adjust the percentage fuel that leaves the upper output of the splitter

## Clearing the grid

Goal level: Sea level

Pre: There are some elements on the grid

Actor: User

Main Success Scenario:

1. User opens menu

2. User clicks on clear

3. User is shown a confirmation dialog to confirm that they wish to clear the grid.

4. User presses “Yes”.

3. System clears the grid

Extensions:

4a.1 User presses “No”.

4a.1 System aborts the operation.

## Saving a grid

Goal level: Sea level

Pre: There are some elements on the grid

Actor: User

Main Success Scenario:

1. User opens menu

2. User clicks on save file

3. System shows save file dialog

4. User specifies file name and location

5. System saves the grid

Extensions:

4.а User does not specify name or location.

4.а.1 The system shows an error message informing the user

## New File

Goal level: Sea level

Actor: User

Main Success Scenario:

1. User presses the button “new file”

2. System creates new blank file

3. System displays blank file to user

## File Open

Goal level: Sea level

Actor: User

Main Success Scenario

1. User presses the button to open file

2. System displays file explorer

3. User selects the needed file

4. User presses “Ok” button

5. System opens the selected file

Extensions:

5a: It is unable to open selected file (wrong format or file was damaged)

5a.1: System displays message that file cannot be opened

## Undo last change

Goal level: Sea level

Actor: User

Pre: At least one modification had been made

Main Success Scenario:

1. User clicks on undo button

2. System depict the network by one action backwards

# User interface

# Non-functional requirements

Of course when using an application the things that can bother us or make us happy are not always related to the product’s functionality. What about Accessibility, Efficiency, Maintainability, Reusability, Usability?

1. Everything that you will ever need creating flow diagrams or systems, you can easily navigate to through the toolbox. The toolbox is placed on the left side of the working space. One point for accessibility!
2. You do not want to lose your time drawing lines? “Flowly” does it for you! Just pick an element from the toolbox and WISELY place it on the grid! One point for efficiency!
3. You hardly ever can get an error while working with “Flowly”. Everything on the software is properly set-up! If you get an error the System will not give the possibility for breakdown! One point for maintainability!
4. Do you know that you can safe and load/open flow systems? Yes you can with “Flowly”. You can work on your project whenever you want! One point for Reusability!
5. Summing it all our application is easy to use, it has friendly User Interface, it is error-prone and really efficient to use! One point for usability!