

User REquirements Specification



First Version

Bilger yahov

oLEKSANDR SUPRUNENKO

ILIA NIKUSHEV

GEORGI CHISHIRKOV

LYUBOMIR DIMOV

MENGCHUAN LIU

Table of Contents

[Introduction 2](#_Toc444550556)

[Functional requirements (use-cases) 2](#_Toc444550557)

[1. Undo last change 2](#_Toc444550558)

[User interface 3](#_Toc444550559)

[Non-functional requirements 3](#_Toc444550560)

[Appendix A: Definitions 4](#_Toc444550561)

# Introduction

User Requirements Specification document introduces you to the requirements which our application will meet and the features it will possess. Every software application which is to be developed needs to meet two kinds of requirements – functional and non-functional.

We described the functional requirements using use-cases.

Use-cases can be very helpful describing the flow of actions between the user and the system. Inside the document there are [X] use-cases. They are described in a structured and neat way. Each use-case has its own extensions below their Main Success Scenario.

Non-functional requirements of an application have to deal with its quality aspects. We briefly emphasized on the most important non-functional requirements. We included brief examples about accessibility, efficiency, maintainability, usability and reusability.

# Functional requirements (use-cases)

THIS ONE IS EXAMPLE !!!

## 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” option in the menu

2. System depicts the network by one action backwards.

Extensions:

1a. if there are no changes

* The undo button is not active.

Note: The user can click on the button a maximum of 5 times, or until no changes are found and the button is disabled.

## Create a new file

Goal level: Sea level

Actor: User

Pre-condition: The application is running

Main Success Scenario:

1. User clicks “New” button.
2. System creates an empty project.

Extension:

1. Other project is already open.

System will ask the user whether to save or not save the current project.

## Start a simulation

Goal level: Sea level

Actor: User

Pre-condition: The simulation is not currently running

Main Success Scenario:

1. User clicks “Start” button
2. System check the status
3. System starts simulation

Extension:

1. There is uncompleted required information.

System will display errors and let the user to fix them.

1. There are required information not meet the rules.

System will display errors and let the user to fix them.

## Undo an action

Goal level: Sea level

Actor: User

Pre-condition: The simulation is not currently running

Main Success Scenario:

1. User clicks “Undo” button
2. System undo the last action

Extension:

1. The simulation is still running.

System will display an error message and ask the use to stop the simulation.

## Exit application

Goal level: Sea level

Actor: User

Pre-condition: The application is running

Main Success Scenario:

1. User clicks “x”(on the top right) button.
2. System closes the application.

Extension:

1. The currently project is not saved.

System will ask the user whether to save or not save the current project.

## Show simulation result

Goal level: Sea level

Actor: User

Pre-condition: The application is running

Main Success Scenario:

1. User clicks “Show Simulation” button.
2. System pops up a load simulation result dialog.
3. User chooses the simulation result file.
4. System displays the simulation result

Extension:

1. There is current simulation result shown.
2. System will check whether the simulation result is saved or not
3. System will ask the user whether save or not the result if it is not saved.
4. System will ask the user whether close the current and open a new one or not.

# 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 Usability, Reliability, Performance and Maintainability?

1. Usability – Our application will meet all the requirements to achieve quantified objectives with effectiveness, efficiency and user satisfaction.
2. Performance – The amount of useful work accomplished by the application compared to the time and resources used is relatively huge.
3. Reliability – Our application will be able to function under stated conditions for a specified period of time. All the results returned from the software will be absolutely correct and precise.
4. Maintainability – Our software will be free of defects, it will meet all the requirements and if and error occurs it will be easily dealt with.

# Appendix A: Definitions