Hanoi University of Science and Technology

School of Information and Communication Technology

Software Requirement Specification – SRS

Visualization of Algorithms
Minimum Spanning Tree and Shortest Path in Graph Theory

Subject: *Software Engineering*

Nguyễn Duy Tiến – 20194857

Trần Quang Hải – 20194755

Trần Lâm - 20194787

Table of contents

1 Introduction

1.1 Purpose

This document provides a detailed description of the User Management Module, the user group, and their functions available at run time. Document describing the purpose and features of the system, interfaces, and constraints of the system to be implemented in response to external stimuli.

1.2 Scope

The purpose of the software is to create the user management module, the role of the user, and the functions that the user / user role can use at runtime.

This software is offline and open to everyone. Each time the user selects a function on the menu, the interface corresponding to that function will be displayed.

1.3 Glossary

1.4 References

- Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne
- SRS-UGMS-Sample-VN.doc, Nguyen Thi Thu Trang

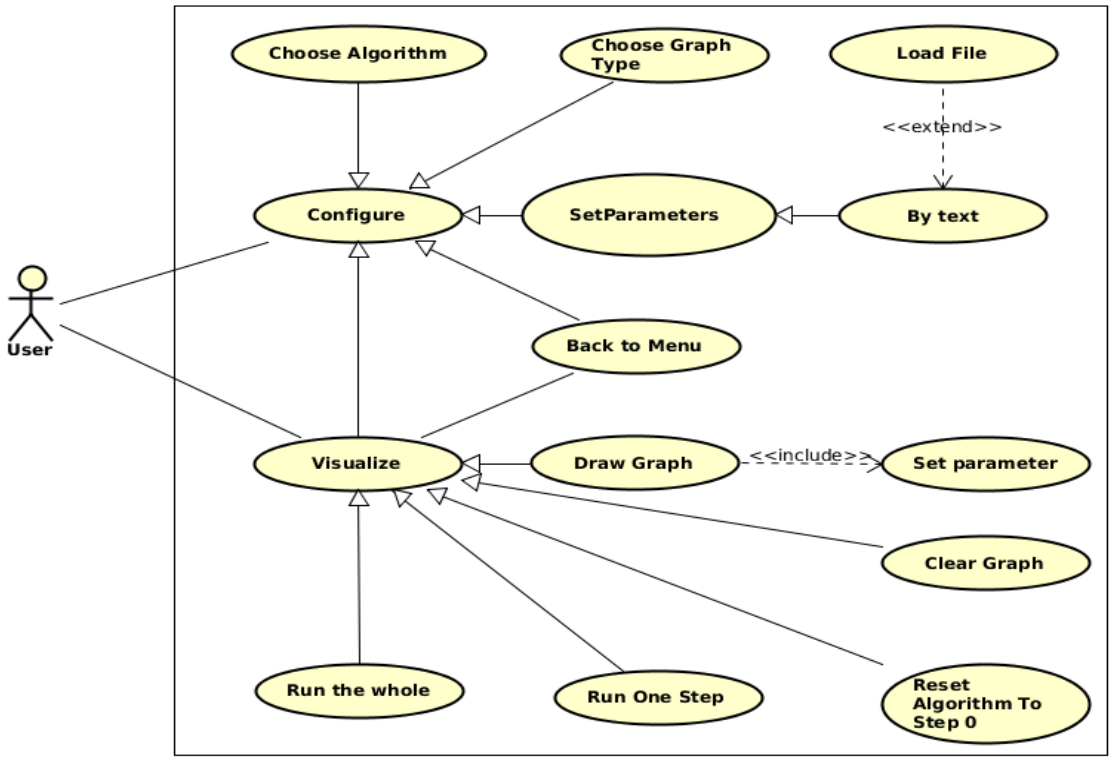
2 General Description

2.1 Agents

This software has only one agent, which is the user. They have full functionality to interact with software via the Graphical User Interface.

2.2 Use case overview diagram

The user can choose the algorithm that they want to be performed, setting the parameters needed and ask the software to visualize through each process of the algorithm. To set the parameters, user can tick the option they want, and input the data for the graph. Graph input can be loaded from a file, typed by hand. When the user is satisfied, they can continue to visualize their graph. After that, tinkering the graph is still available in the form of clicking the drawing board.

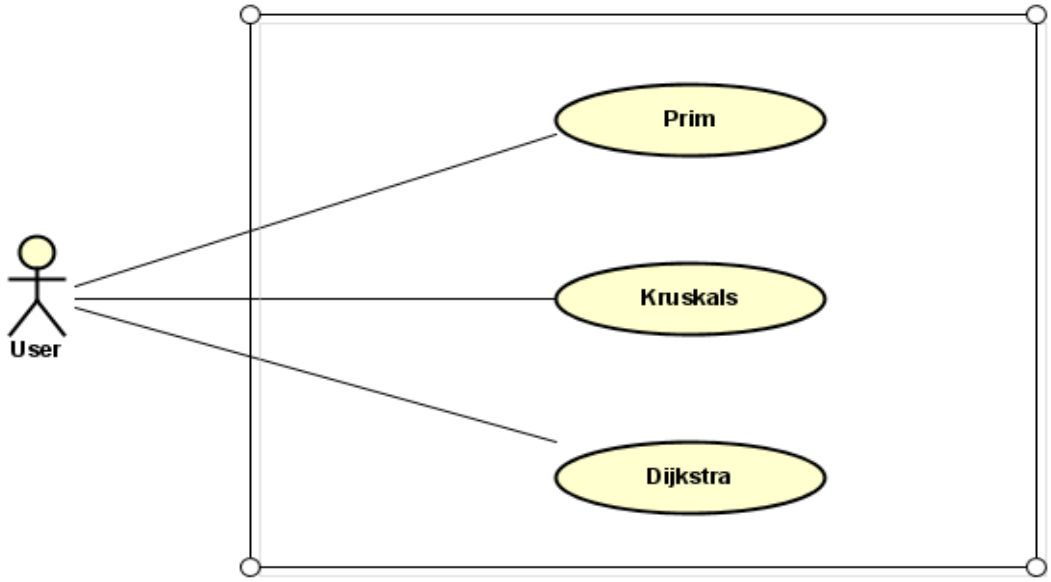


Finally, the fun part. The user will create their graph and run the algorithm with it. User can cancel the running process anytime and redraw their graph, or change the algorithm, graph type.
The use cases in this general use case diagram are the composite

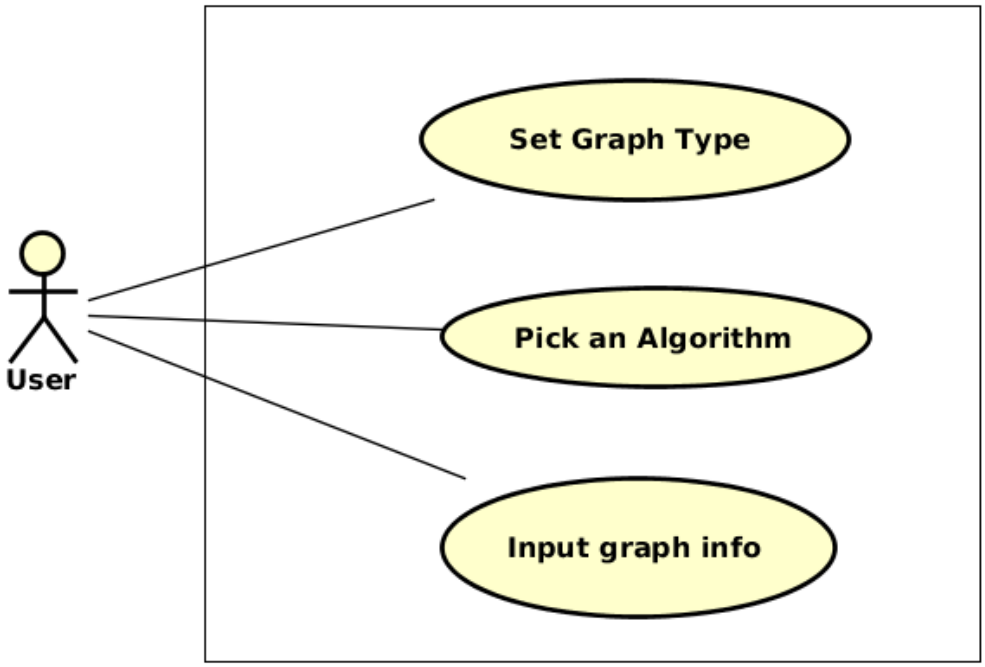
use cases of a group of use cases. Details of these use cases are given in the breakdown diagrams in the following section.

2.3 Use case decomposition diagram

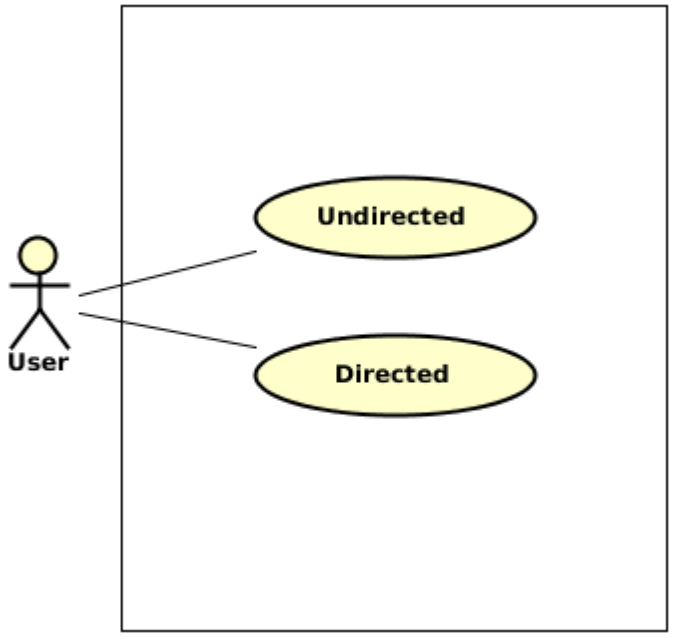
2.3.1 Use case “Choose Algorithm”



2.3.2 Use case “Set Parameters”



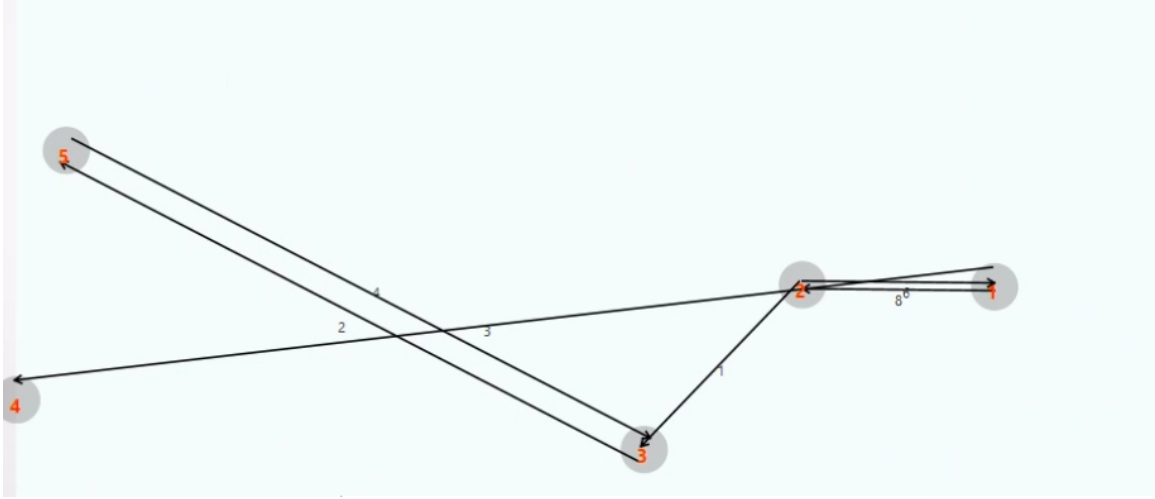
2.3.3 Use case “Choose Graph Type”



2.4 Usecase: Visualize

2.4.1. Draw Graph

To draw the graph, you can click on empty space on the board to create a vertex. Add an edge by choosing two vertices. When two vertices are chosen, a text box will pop up to ask you for the edge weight. If you accidentally choose an unwanted vertex, unselect it by right clicking on the board



2.4.2 Clear Graph

To quickly discard the current graph and create a new one, you can click on the “Clear Graph” button. This will keep the option you chose concerning graph type and algorithm.

2.4.3 Back to menu

Clicking the button with a triangle on the upper-left corner of the screen will put you back to the input menu. In here, you can try experimenting with new algorithm and graph type.

2.4.4 Run one step

Clicking the “Run one” button will run one step of the algorithm. The graph is locked – You cannot make any change to the graph at this point. For simplicity, some algorithms will randomly choose the root and destination vertex for you. Edges and Vertices are highlighted, indicating the step taken in the algorithm.

2.4.5 Run the whole

Clicking the “Run all” button will run the algorithm until very end without stopping. The graph is locked, edges and vertices are highlighted, indicating the steps taken in the algorithm.

2.4.6 Reset Algorithm to step 0

The “Reset” button will refresh the graph back to the initial state when no algorithm step was taken. It also unlocks the graph – if it was previously locked. You can modify the graph as you see fit, and then run again when you choose to.

3 Specification of functions

3.1 Specification of use case UC001 "Configure"

<table>
<tr><td>Use case code</td><td>UC001</td><td>Use case name</td><td>Configure</td></tr>
<tr><td>Agent</td><td colspan="3">User</td></tr>
<tr><td>Prerequisite</td><td colspan="3">None</td></tr>
<tr><td>Default</td><td colspan="3">Graph Type: Undirected</td></tr>
<tr><td>Main flow of event (Success)</td><td colspan="3">
<table>
<tr><th>No.</th><th>Performed by</th><th>Action</th></tr>
<tr><td>1.</td><td>User</td><td>Tick the target graph type</td></tr>
<tr><td>2.</td><td>System</td><td>Change the applicable algorithms list.</td></tr>
<tr><td>3.</td><td>User</td><td>Choose an algorithm.</td></tr>
<tr><td>4.</td><td>System</td><td>Enable the user to continue to the next step</td></tr>
<tr><td>5.</td><td>User</td><td>Go the the next step: Graph drawing</td></tr>
</table>
</td></tr>
<tr><td>Alternate flow of event</td><td colspan="3">
<table>
<tr><th>No.</th><th>Performed by</th><th>Action</th></tr>
<tr><td>4a</td><td>User</td><td>Type input into the text box</td></tr>
<tr><td>4b</td><td>User</td><td>Choose a file to get data from</td></tr>
<tr><td>4c</td><td>User</td><td>Notify the user about incorrect graph</td></tr>
</table>
</td></tr>
<tr><td>Postrequisite</td><td colspan="3">Both graph type and Algorithm chosen</td></tr>
</table>

3.2 Specification of use case UC002 "Visualize"

<table>
<tr><td>Use case code</td><td>UC002</td><td>Use case name</td><td>Visualize</td></tr>
<tr><td>Agent</td><td colspan="3">User</td></tr>
<tr><td>Prerequisite</td><td colspan="3">Input stage successfully completed</td></tr>
<tr><td>Main flow of event
(Success)</td><td colspan="3">
<table>
<tr><th>No.</th><th>Performed by</th><th>Action</th></tr>
<tr><td>1.</td><td>User</td><td>Draw the graph</td></tr>
<tr><td>2.</td><td>User</td><td>Run the algorithm</td></tr>
<tr><td>3.</td><td>System</td><td>Test for graph validity</td></tr>
<tr><td>4.</td><td>User</td><td>Keep running</td></tr>
<tr><td>5.</td><td>User</td><td>Reset the algorithm</td></tr>
<tr><td>6.</td><td>User</td><td>Run Again</td></tr>
<tr><td>7.</td><td>User</td><td>Go back to input menu because they are bored with the current algorithm</td></tr>
<tr><td>8.</td><td>User</td><td>Run the new algorithm</td></tr>
<tr><td>9.</td><td>User</td><td>Shutdown the application</td></tr>
</table>
</td></tr>
<tr><td>Alternate flow of event</td><td colspan="3">
<table>
<tr><th>No.</th><th>Performed by</th><th>Action</th></tr>
<tr><td>3a</td><td>System</td><td>Notify error: The algorithm might not be applicable for this graph (it may has many disconnected components, for example)</td></tr>
</table>
</td></tr>
<tr><td>Postrequisite</td><td colspan="3">No</td></tr>
</table>