Requirements Document
Cent

Calum McLellan, Tino Klingebiel, Niklas Spitzer

13. Mai 2013

Inhaltsverzeichnis

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the proposed graph drawing system that will be used as a contractual agreemend between the client and developer. This document copmlies with the IEEE.830-1998 Software Requirements Specification.

1.2 Scope of Project

The software to be developed will provide visual representation of graph classes. The user can select a graph class and view a heirarchical representation of all sub- and superclasses. The problems that can be solved by the classes and the time taken to solve them will also be available. The user can also select a problem and the system will display the graph classes that can be used to solve the problem.

1.3 References

IEEE 830-1998 Recommended Practice for Software Requirements Specifications.

1.4 Overview of Document

The next chapter will contain a general description of the functionality of the software; this section is aimed towards non-developers and hence uses clear, non-technical terminology. This is followed by more specific requirements in chapter 3, primarily for developers.

2 Overall Description

2.1 Profuct Perspective

The system provides an interface for users to draw graphs. The system will use the y-files library.

2.2 Product Functions

The system will list all graph classes and all problems which can be solved by those classes. It can draw the graph classes in an heirarchial structure. The user can zoom in and out and can select objects (graph classes, problems) in order to display furthur details about that object. The user can also start by selecting a problem and showing the graph classes that can be used to solve that problem.

2.3 User Characteristics

The system will be publicly available meaning the user will be a large range of different skill levels. Hence the system should be usable without any training. The system owner is expected to be computer and internet literate.

2.4 Constraints

- yFiles libary should be used for drawing graphs
- the programm should run as a java applet

2.5 Assumptions and Dependencies

- None

2.6 Apportoining of Requirements

- All requirements must be implemented in the first release of the product.

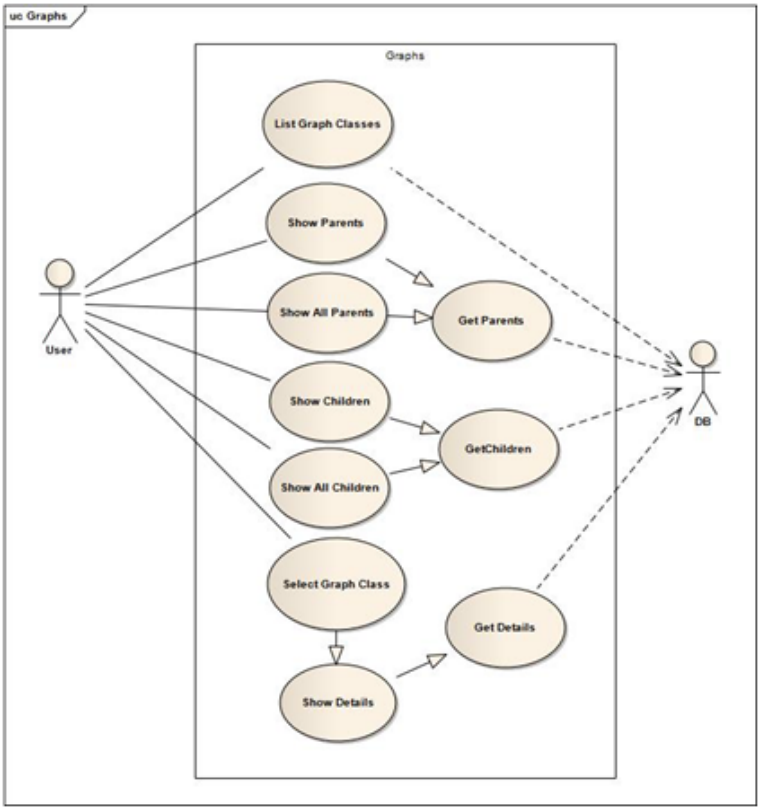
3 Requirements Specification

3.1 External Interface Requirements

- The system will need an user interface for drawing graphs

3.2 Functional Reqirements

3.2.1 Graph Use Cases

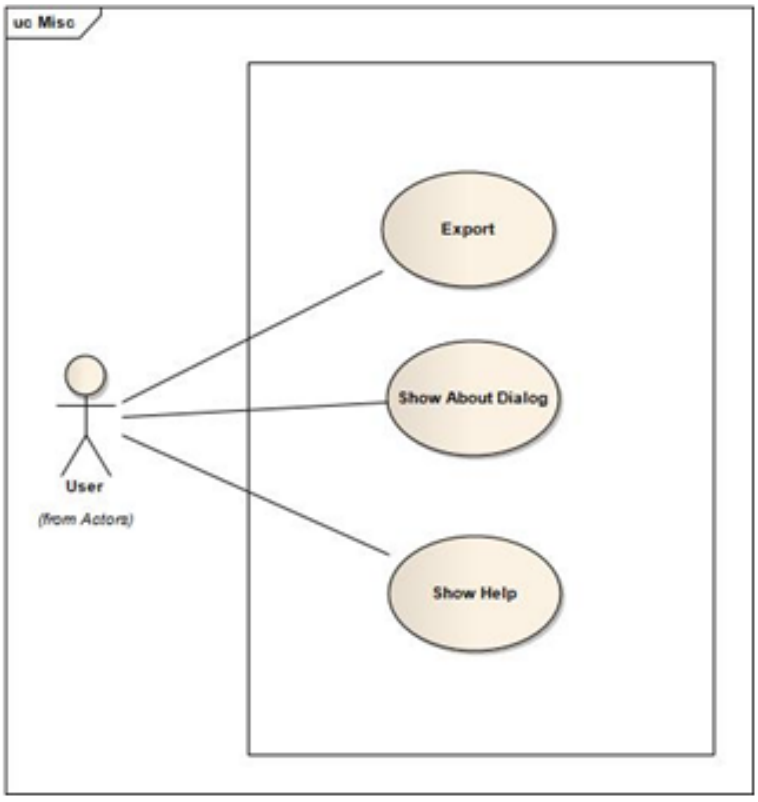


List Graph Classes	List all available graph classes in the database.
Select Graph Class	The user selects one of the available classes and the details of the class are shown (Show Details).
Show Details	Shows the detail of any selected graph class.
Show Children	Shows the immediate children of the selected graph class.
Show All Children	Shows the entire child hierarchy of the selected graph type.
Show Parents	Shows the immediate parents of the selected graph class.
Show All Parents	Show the entire parent hierarchy of the selected graphtype.

3.2.2 Problems Use Cases

List Problems	List all the problems available in the database.
Select Problem	The user selects one of the available problems. The system then gets all the graph classes available for the selected problem (and thair runtime) and displays the information to the user (Get Graph Classes and Runtime for Problems).
Show Details	Shows the detail of any selected graph class (GetGraph Classes and Runtimes for Problems).
Select Runtime for Problem	The user selects a runtime for the selected problem, all graph classes that can solve the problem in this time are displayed (Show Graph Classes for Runtime).
Select Graph Class for Problem	Selects a graph class for the selected problem and displays the runtime for that class. Optionally, the derived graph class can be retrievedand displayed (Show Runtime for Derived Class).

3.2.3 Misc Use Cases



Exports	Exports the current display to an image file.
Show About Dialog	Shows about dialog containing information about the software.
Show Help	Shows the software help.

3.3 Performance Requirements

All requests should be completed on the server side in an average time of at most 5 seconds.

3.4 Logical Darabase Requirements

The database of the current system will be used.

3.5 Design Constraints

The resulting java applet should run on java runtime 1.7

3.6 Software System Attributes

3.6.1 Availability

The system must be available at all times with the only exception being downtime on the hosting system. During downtime a maintenance page should be shown to any visitors to the site.

3.6.2 Security

The database should not be accessable from the outside of the webserver.

3.6.3 Reliability

Relaibility is determined by the host system which will be made availalble by the customer. Hence there are no project reliability contraints.

3.6.4 Maintainability

It should require minimal effort to make small changes such as adding a language.

3.6.5 Usability

The system should be easy to use with a simple but functional user interface.

3.6.6 Autonomy

The system should be able to run unattended for at least 1 week, allowing for a weekly maintenance plan.