Software Requirements Specifications

Development of a repository for Software Models and Metadata

Version 2

Clément Candellier

Table of Contents

I) Introduction....1
I.1) Purpose....1
I.2) Scope....1
I.3) Definitions, acronyms and abbreviations....1
II) Overall description....2
II.1) Project perspective....2
II.2) Product functions....2
II.3) Constraints for the final user....3
III) Specific requirements....3
III.1) Functional requirements....3
III.2) Additional requirements....5
IV) Prioritization and Release Plan....7
IV.1) Prioritization of the requirements....7
IV.2) Release plan....8

Version history:

- Version 1: 15/03/2018
- Version 2: 22/03/2018

  Correction of the functional and additional requirements. Better presentation of the project (II). Addition of the prioritization of the functional requirements.

I) Introduction

This section provides a description of the following document as well as the general purpose of the project. It also gives a list of definitions of terms that will be used throughout this document.

I.1) Purpose

The purpose of this document is to give a first approach of the detailed requirements for the project regarding the development of a repository for software models and metadata. It will provide general information on the software, as well as giving the different functions that the project has to cover.

I.2) Scope

This project of development of a repository for software models and metadata will enable the final users, researchers, students, to search into a wide variety of programs' diagrams and to add their own files to the shared documents.

The search can be done among various criteria such as the number of classes or the field of application of the programme.

The software needs an Internet connection to work, as well as a web browser, to display the graphical interface. All the information are stored into a database; the user will query data from this database when making a search. The database will be stored into a web-server.

This project is a modernisation of an already existing project that can be found at:
www.models-db.com

The project must use more modern technologies and have a better design than the previous one: with the improvements in recent technologies, this project is rapidly becoming outdated.

I.3) Definitions, acronyms and abbreviations

Term	**Definition**
User	Someone who interacts with the web portal
Web portal	A web application that is used to query the database, upload content, and display the results
UML	Unified Modelling Language, it is a general purpose modelling language that is used to provide a standard way to visualize the design of a system
XMI	XML Metadata Interchange, it is a language intended to provide a standard way for programmers and other users to exchange information about metadata.It has been specifically designed to help programmers using the Unified Modeling Language (UML) with different languages and development tools to exchange their data models with each other

Table 1: Definitions

II) Overall description

This section will give an overview of the project. It will describe more precisely the goal of the project, and present the different functions that it should cover.

II.1) Project perspective

The centre of the project is a web portal linked with a database containing UML files, stored as images and XMI files containing programs' classes and dependencies. The user will be able to query the database through the web portal. The web portal will be hosted into a web server, and it represents the interface between Human language and machine language.

The portal will be used for managing the files and the system as a whole. It will provide informations on the database, and will present the concepts of the project. A page will be dedicated for researching files from tags, and an other page will enable the user to upload content. He/She will also be able to add tags to already stored images. All of that adds the perspective of crowdsourcing into the project.

All the communications between the portal and the database are done over the Internet, so the user will need to be connected in order to use the project the functions of the project.

II.2) Product functions

The goal is to create and manage a repository for models and metadata. All the models have from zero to an important number of tag(s) that will be used to find a particular UML diagram. The initial database will be extended with the help of the users: each one can upload one or multiple models, regrouped into projects. The database will be extensible, according to the number of files stored in it. There is already an existing database, containing, at the moment, around one thousand files, that will be used as a base for the development. Other databases will be merged with the current one to get a bigger amount of data.

The web portal will include a graphical interface where the user will find all the functions, such as searching through the database or uploading models. Results of the search will be presented into a table. Statistics about tags and search can be added into graphs. The user will see the name of the file, a graphical representation of it and the tags associated with it.

Searching will be done with several criteria, the results will have to match all of them in order to be relevant.

The project will be divided into different parts, that are still linked together: first, what is needed is to create the website and implement the basic functions, such as querying data. At this point, the link between the web portal and the database must be functional.

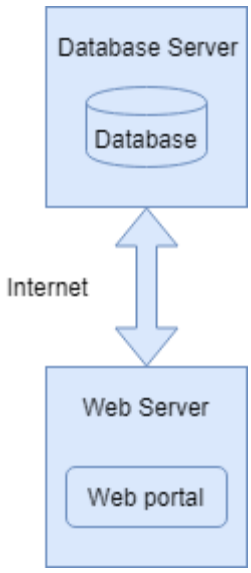

Figure 1: Block Diagram

The next step is to enable the user to add tags to already stored data.

Then, the design will be implemented. This was one of the major points to be improved from the older version. Finally, the feature for uploading data will be added.

The following diagram gives a first look at what a user will be able to do while using the project:

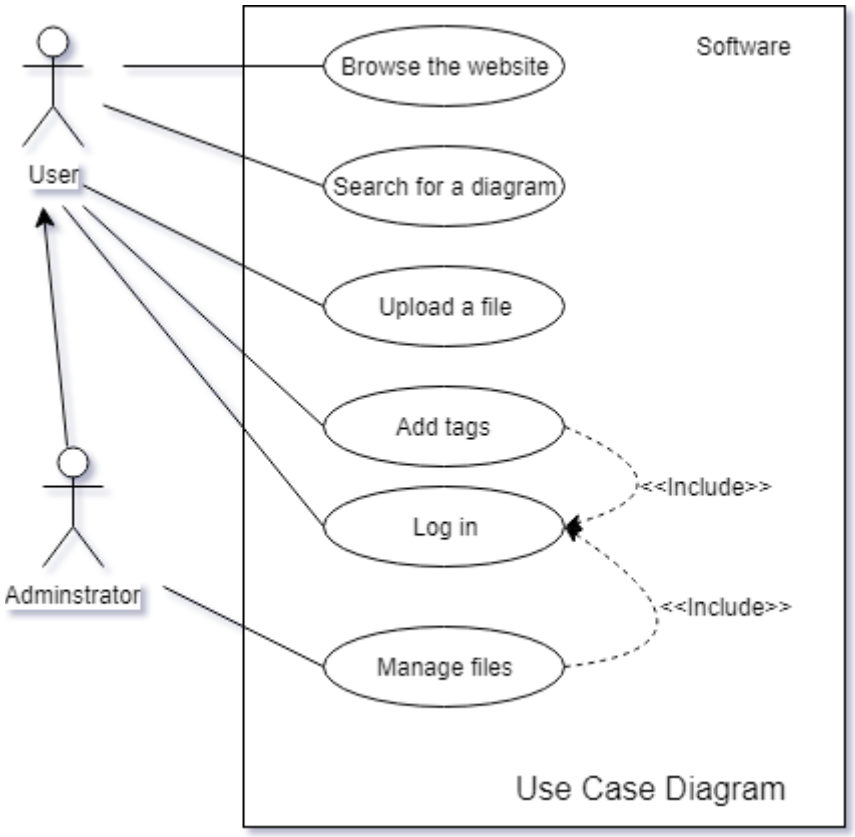

Figure 2: Use Case Diagram

II.3) Constraints for the final user

There are not many constraints for the final user: he/she just need a web browser with an Internet connection. Since all the data is stored into a database server, the Internet connection is crucial: there is no possible access to the project without it.

III) Specific requirements

This part contains all the requirements of the project, classified under their priorities. It gives a detailed description of the system and all it's features.

III.1) Functional requirements

The functional requirements are the main features of the project. No progress can be done if these factors are not satisfied.

FR1: Web portal

The web portal is the centre of the project: every other action will be done through it. Querying the database, uploading files, managing the tags, all the functions will be centralized into one website. This is a front-end application.

From there, the user will be facing a graphical interface displaying all the actions he/she can do:

- log in/register
- go to the query page
- go to the upload page
- have a brief description of what the site is about and why it was done

FR2: Study and link to the database

The main purpose of the project is to query an already existing, expendable, database. Creating the link between the web portal and the database is one of the basic features that have to be implemented immediately.
It is also important to note that the current solution for the database may have to be improved or modified, in terms of structure or technology. In all cases, multiple databases containing various projects will have to be merged to get the final version of the project. This is a back-end function.

FR3: General design

This project focuses on improving some of the features of an older project. One of the issues with it is that it's design was ineffective. Having a graphical touch to it is one of the goal of the current project.

FR4: Search features / Filters

The website is mainly created to query a database. The search feature is, naturally, a mandatory step in the development of the project. Searching through the database can be done with several criteria:

- the user wants to search for a specific project: a research field will be implemented to reply to such requests

- the user wants to search for a specific class name: a field will be added with such possibility

- the user wants to search for a specific attribute name: the same will apply

- the user wants to search for a specific operation name: the same will apply

Any combinations of the criteria above would work as well: the user can search for a specific attribute within a particular class, that is contained inside a specific project.

Then, filters in the terms of search can be added: maybe the user wants a specific number of results to be displayed, or he/she is looking for a program with a certain number of attributes or operations. Those features can be implemented as a part of the research process.

FR5: Clear identification of the functions / Easy access to them, always on the page (menu)

Having an interface will enhance the accessibility to the web portal and should be treated as one of the priorities.

FR6: Add tags / Properties

The user will have the possibility, once he/she received the collection of diagrams, to select one or several diagram(s) and add tags to it/them, if he/she thinks that they are relevant. This is done towards crowdsourcing: the project aims at being used and upgraded by everyone. The additional tags will be stored in the same database as the images.

This can be done either after having queried the database and selected a diagram, or when the user wants to upload his/her own projects or files.

FR7: Uploading files: images and UML files

One of the aspects of the project is it's crowdsourcing factor. Users will be able to add their own contribution to the project, by uploading their own files into the database, and tagging them. The files have to be of the correct format in order to be treated and added into the database. The user will have an interface allowing him/her to use these functionalities.

The system will have a function that will prevent from storing the same file multiple times. If a user wants to upload a diagram, or a project, the system will tell him/her if the file is already stored. Maybe the diagram is the same, but the name is slightly different, the system has to recognize such cases and not let the user upload the data.

FR8: Use of a particular system to store and query the results (SQL, NoSQL)

The traditional system to query a database is using SQL requests. This is used for databases that are working with the SQL system as a base. Nowadays, there are databases that do not use SQL requests, called NoSQL. They are used when the data is not not necessarily linked with logical connections, or just has a few of those links. In the situation of the project, having an overview of the said NoSQL technologies can be interesting, even if, considering that the database has to be very connected with regards to logical links, the SQL solution is still the best.

III.2) Additional requirements

The following requirements are considered as non essential: the project can work without them but adding those would be an improvement over the standard functionalities or design.

A1: Login / Account

If the user has an account on the website, he will be able to see statistics of his/her activity. Also, the administration will be done through a secure logged account, so that not all users can have access to all the database, and maybe modify it's structure.
In addition, the user will be able to see a summary of his/her actions on the website: the number of queries he/she made, the number of uploaded files and probably other statistics.

A2: Use of a particular language

To realise the web portal, several languages can be used. However, mainly using only one of them, if it is possible, will lead to less development time and less confusing maintenance.

A3: Download the result of the search

Once the user has searched and found the correct data, he may want to have it on his/her device. If he/she is able to download the resulting diagram, he/she will be able to use it more effectively.
The result could be downloaded in two formats: as an image first, that will give a graphical representation of the diagram, or as an XMI file, in order to have the file in a more technically usable format.

A4: Display of the results

Once a query has been made, the user has to understand the results of it: that means that the display of the results mus be graphical: it is possible to think about a table, or a sliding display of the diagrams, represented by their image format.

If the user wants a more in-depth display of the results of his/her query, he/she will have an option to see charts of the result. The user will be able to tell how many diagrams have a certain amount of classes for example, which will enable him/her to see if the code is quiet basic or very sophisticated for example.

A5: Add comments

Some diagrams can be confusing or maybe a user has a solution for improving a particular code. If the user is able to add text next to a diagram, seen as comments, this will improve the crowdsource aspect of the project.
This aspect is different from tagging because a tag is a single word or a very short phrase. A comment can be a paragraph of text that the user wants to add to a project or a diagram. Queries will not be based on the comments, what is inside this field is just to get, maybe, more information on the context of the project, or in what circumstances it was made.

A6: Exporting results in different formats

As the database will grow, the number of results could become more and more important. Displaying a lot of matching results is one part, but if the user wants to work with these results, he/she has to have them in an exploitable format (PDF, Word format for example). Having the opportunity to export the results could be a potentially very useful feature.

A7: Requirements in terms of resources

The project in itself, seen from the client side, should not require too much resources from the CPU or the GPU. The client only has to use the web portal, which is normally a light application to run, but keeping in mind that the used resources should always be minimal is a good way to develop an application.

A8: Performance

The database is normally configured to be consulted by an important number of users at the same time but it is important to take care of the modifications that can occur while a user is querying the database. The problem could be that a result is shown on the user interface, but has been deleted by an administrator a few seconds after the query. Then, the user will not be able to access to one of the result of his/her query, when it could have been the most relevant.

A9: Display of general informations on the first page

To explain what the project is about, it is important that a new user immediately gets a vision of what the portal can do and the goals of the project.

A10: Watermark the images / Protection

When displayed on the result screen, the images could be protected, so that, for example, the user has to be registered to download it. This protection can simply be done by adding a watermark to the image. When the user is logged in, the watermark will disappear.

A11: Display the number of results found

It might be interesting for a user to know how many results his/her query has generated. This will give information about the query, that is easily understandable by every user.

A12: Links to external resources

Such resources can be access to the Chalmers website, or definitions of the most difficult terms. The users will not necessarily be a confirmed programmer, who works with UML files every day; explaining the terms could be an interesting feature.

A13: Add an “About” page

This page will explain the process of the project: how the query works, why the project was made in the first place, what technologies are used and why.

IV) Prioritization and Release Plan

In order to release a working solution before the end of the given time, some requirements have to be prioritized. This section discusses the prioritization of the requirements and gives a first idea of how the work will be separated, that can evolve as the project progresses.

IV.1) Prioritization of the requirements

The requirements as presented in the list before, are classified into two categories that can be described as important requirements and less important ones. Among the important requirements, some have to be done before the others, either because they are going to be used as a base for the others, or because they represent the core part of the project.

For the functional requirements: a classification has been done by the author of this document for which requirement has a higher priority compared to the others. Because this job has been done by only one person, it is subject to changes when read by others. The classification is realised with numbers from 1 to 8, 1 being the highest priority in the project.

Requirement :	**Priority :**
FR1 : Web portal	1
FR2 : Link to the database	2
FR3 : General design	5
FR4 : Search features / Filters	3
FR5 : Menu	7
FR6 : Add tags / Properties	4
FR7 : Uploading files	6
FR8 :Use of a particular system to store and query the results (SQL, NoSQL)	8

Table 2: Priorities

One of the main features this project should have is the addition of tags or properties. In order for this function to work, the basic features have to be implemented first. It is impossible to add tags if the web portal is not functional for example. The requirements have been classified regarding this as well.

For the additional requirements, it is possible to make a ranking based on the usefulness and the period at which it can be implemented: if there is a bit of time left during one of the project's parts, maybe considering adding additional requirements would be a good idea. Obviously, the main focus should stay on the functional requirements, but adding other functions is possible and will bring great added value to the project.

IV.2) Release plan

In the first release, the foundations of the project will be included, together with the first requirements. This will serve as a base for all the other requirements. The goal of this first release is to have the framework of the project: something that will be easily comprehensible and extendable afterwards.

Depending on the progress of the project, there might be multiple other releases. Each one will include more functionalities than the previous.

Considering the time dedicated to this project, around three months, there is a possibility that not all the requirements will be achieved. The goal, however, is to have 100% of the project working in the end.