2DV603 - Software Engineering

Thesis Management System

Requirements Document

Author: Ahmad Abdilrahim

Author: Ahmad Sadia

Author: Caesar Alhawi

Author: Cristian Babes

Author: Marcello Vendruscolo

Supervisor: Mirko D'Angelo

Supervisor: Mauro Caporuscio

Semester: VT 2019

Area: Computer Science

Table of Contents

1 – Introduction

The first section of this document explains the document structure and its purpose. It provides a general overview and information about the context in which the product is being developed as well as technological terms and references which might be essential for readers to thoroughly understand the document. Finally, it presents what the remaining sections of the paper discuss about.

1.1 – Purpose of the Requirements Document

This document, formally known as requirements document, but also defined by other several designations, such as functional specification and software requirements specification, is a formal document regarding the requirements for a system under development. The target audience of a requirements document is all the system stakeholders (people directly or indirectly involved in the project), and the main purpose of the document is to clearly explain to this range of different readers the functionalities of the system without discussing the implementation details. The structure of this document follows the conventional IEEE/ANSI 830-1993 standard which facilitates its comprehension. Furthermore, taking into account that readers have different technical backgrounds, this document was written without the usage of computer science specific terms, and, therefore, no particular prior knowledge is required for one to comprehend the document. However, a general understanding about information systems is suggested.

1.2 – Scope of the Product

The product to be developed is a thesis management system which will operate in a fast-growing, technological university in Sweden. Currently, most of the administrative and management processes that regards graduating students' thesis are done following non-formal and manual protocols, including the communication among the involved people, such as students and professors, that is done through e-mail. However, taking into account the currently

available know-hows, it is crucial to create a technological solution that standardises procedures, optimises the thesis management processes and productivity, assembles all the information in a single system, and, consequently, supports the expansion of the university.

1.3 – Definitions, Acronyms, and Abbreviations

This sub-chapter is not applicable to this document since the definitions, acronyms and abbreviations used throughout the paper are always explicitly explained in text.

1.4 – References

1. Palma, F 2019, Lecture 2: Software Engineering Design (2DV603) Requirements Engineering Understanding and Elicitation of Software Requirements, lecture notes, Linnaeus University, delivered 30 January 2019.

2. Palma, F 2019, Lecture 3: Software Engineering Design (2DV603) Requirements Engineering Requirements Validation and Management, lecture notes, Linnaeus University, delivered 01 February 2019.

3. Palma, F 2019, Lecture 4: Software Engineering Design (2DV603) Requirements Engineering Modelling with UML, lecture notes, Linnaeus University, delivered 06 February 2019.

4. Palma, F 2019, Lecture 5: Software Engineering Design (2DV603) Requirements Engineering Requirements Modelling and Management with Tools, lecture notes, Linnaeus University, delivered 08 February 2019.

5. Perez, D 2019, Lecture 7: Software Engineering Design (2DV603) Performance Evaluation: Operational Laws, lecture notes, Linnaeus University, delivered 21 February 2019.

6. Caporuscio, M & D'Angelo, M 2019, Lecture 15: Software Engineering Design (2DV603) Project Specification, lecture notes, Linnaeus University, delivered 05 April 2019.

1.5 – Document Remainder

The remainder of this document provides an overall report of the thesis management system, including its functional and non-functional requirements, its constraints, and its dependencies. It also provides insights into the requirements engineering process, including the usage of unified modelling language (UML) class diagrams to modelling requirements.

2 – General Description

The second section of this document provides high-level (broad, general) information about the thesis management system for the readers to familiarise themselves with the product to be developed and its main characteristics. Moreover, this section also gives insights into the potential users of the product.

2.1 – Product Perspective

The thesis management web application will provide its users with an efficient and pleasant experience when it comes the time for students to take their degree project course before graduation. The application will address all the thesis management process aspects, from the beginning stage when students have to deliver their thesis project plan and choose a supervisor, to the completion phase when students deliver their final thesis submission and receive the final grade.

2.2 – Product Functions

The objective of this thesis management application is to ease and simplify the users experience regarding the degree project course. The most important functionalities which the system should embody in order to accomplish its main goal are achieved through the following sub-systems. Furthermore, since the system is a customer driven application, it is also important for it to have high interactivity and utility maximization.

- User authentication
- Uploading and downloading submissions
- Visualisation, suggestion and confirmation of supervisor
- Establishment of submission deadlines
- Evaluation of uploaded documents
- Assignment of different responsibilities to different users

2.3 – User Characteristics

The thesis management system targets essentially two categories of users: the first group is composed by the computer science students population, who are soon graduating and are taking the degree project course, at Linnaeus University, and the second group is composed by scientifically qualified individuals in the computer science department who can be involved in the course. The users in the first category have no administrative powers on the system while the users in the second category have some administrative power according to their assigned roles and tasks. The users must also have some knowledge about the system, but they should receive the necessary education and induction through the university.

2.4 – General Constraints

As more and more people are becoming aware of the needs to keep their personal data safe and secure in the digital systems in order to avoid inappropriate use of it, contemporary laws and regulations are reflecting this concern. Therefore, taking into consideration that the application system deals with personal information of students and professors, the thesis management application system's most important constraints are related to data security, data privacy, and system performance. Furthermore, in the current modern world, people appreciate and value interactive and immediate responses, therefore the system should take these aspects into consideration to match the users' expectations. Therefore, the constraints are on designing and implementation strategies to ensure the users expectations, needs, and concerns are met.

2.5 – Assumptions and Dependencies

Several assumptions have been made at this point of the development engineering process:

- The thesis management system has access to the Linnaeus University's user database since the application is being developed for the institution and should be integrated with the systems that are already on place

- Users do not need to be experts in informatics to use the application, they just need common basic knowledge on computers and how to use and the Internet
- The system development environment is to be long-established in the designing and implementation iterations
- Facilities, such as data centres and offices, and infrastructure, such as communication tools and network, will be available during the entire system development
- Stakeholders will be available for consulting and will engage throughout the system development
- The scope of the project will remain as it is, and if necessary, changes will follow approval procedures

3 – Requirements Engineering

In the third section of this assignment, the concepts of requirements elicitation, requirements analysis, requirements validation, and requirements modelling are applied to the thesis management system.

3.1 – Requirements Elicitation

When starting a new software engineering project, one of the first process that should be executed is the requirements elicitation process. This phase involves studying the application domain field of which the project is involved and the problem to be solved. Usually, in order to gain insights into the field, experts should be consulted. In this phase it is also a good practice to acquire organisational knowledge and the organisation business processes. Finally, after understanding the context and the system to be developed, it is possible to identify possible stakeholders and determine the potential requirements of each single involved stakeholder.

3.1.1 – Stakeholders Identification:

A stakeholder is any person or organisation who is, directly or indirectly, involved or affected by the specific system development and deployment. For the thesis management system within the computer science department of the Linnaeus University, the following stakeholders are identified:

- Client: the computer science department of Linnaeus University that requested the development of the system and the university's board members
- Project managers: this category is composed by Mauro Caporuscio and Mirko D'Angelo who are in charge of planning the activities and resources, organizing the project teams, managing the time, monitoring the project progress and ensuring satisfaction of the client
- Users: students, professors, and other people who will eventually use the application to manage the thesis processes
- University's employees: this category includes the individuals who work at Linnaeus University and will have to acquire knowledge about the new application system, such as the IT-technical support team and the departments administrators
- System designers, developers, and testers: this category includes all the engineers involved in developing, implementing, deploying, and maintaining the new system
- Swedish Post and Telecom Authority and European Union Agency for Network and Information Security (ENISA): regulation/certification authorities that regulate and inspect areas of IT to ensure that everyone in Sweden and within the European Union has access to services that are efficient and that comply with information security.

3.1.2 – Functional and Non-Functional Requirements Elicitation: Consulting and taking into account the needs of each stakeholders from the list above, the requirements below are identified and specified for the thesis management system. The functional requirements fundamentally catalogue the functional objectives of the system (what the system shall do) while the non-functional requirements place qualitative and security measures on how the system should perform its objectives (how the system shall perform).

- Functional Requirements

F.R. 1 - Users shall be able to log-in the application
F.R. 2 - Users shall be able to log-out the application
F.R. 3 - Students shall be able to create submissions
F.R. 4 - Students shall be able to read submissions
F.R. 5 - Students shall be able to update submissions
F.R. 6 - Students shall be able to delete submissions
F.R. 7 - Students shall be able to view a list of supervisors
F.R. 8 - Students shall be able to suggest a supervisor
F.R. 9 - Students shall be able to view the content of feedbacks
F.R. 10 - Students shall be able to view their thesis final grade
F.R. 11 - The coordinator shall be able to view a list of all currently registered users
F.R. 12 - The coordinator shall be able to assign the supervisor role to users
F.R. 13 - The coordinator shall be able to view a list of students assigned to supervisors
F.R. 14 - The coordinator shall be able to view all submissions
F.R. 15 - The coordinator shall be able to evaluate project description submissions
F.R. 16 - The coordinator shall be able to evaluate thesis submissions
F.R. 17 - The coordinator shall be able to evaluate final thesis submissions
F.R. 18 - The coordinator shall be able to set deadlines
F.R. 19 - The coordinator shall be able to update deadlines
F.R. 20 - The coordinator shall be able to submit the grade for a thesis project
F.R. 21 - Supervisors shall be able to confirm supervision requests
F.R. 22 - Supervisors shall be able to decline supervision requests
F.R. 23 - Supervisors shall be able to assess the students' project plan submissions

F.R. 24 - Supervisors shall be able to assess the students' report submissions

F.R. 25 - Supervisors shall be able to view students' project plan submissions

F.R. 26 - Supervisors shall be able to view students' report submissions

F.R. 27 - Readers shall be able to bid for reports of their preference

F.R. 28 - Readers shall be able to feedback submissions

F.R. 29 - Opponents shall be able to view assigned reports

F.R. 30 - Opponents shall be able to give feedback to an assigned report

F.R. 31 - The coordinator shall be able to view the role assigned to a user

F.R. 32 - The coordinator shall be able to assign the reader role to users

F.R. 33 - The coordinator shall be able to assign the opponent role to users

F.R. 34 - The coordinator shall be able to update the supervisor role of users

F.R. 35 - The coordinator shall be able to update the reader role of users

F.R. 36 - The coordinator shall be able to update the opponent role of users

- Non-Functional Requirements

N.F.R. 1 - Users shall be able to familiarise themselves with the application within 5 minutes (usability)

N.F.R. 2 - Users' private data shall be stored and processed in a manner that high security standards and data integrity are priorities (safety)

N.F.R. 3 - The log-in functionality shall not be affected by SQL-injection (security)

N.F.R. 4 - The system shall have average response time less than 10 seconds for submitting a document (performance)

N.F.R. 5 - The system shall have 100% uptime throughout the course duration (reliability)

N.F.R. 6 - The system shall comply with the General Data Protection Regulation (GDPR) (privacy)

N.F.R. 7 - The system shall not depend on the customers operation system and shall support the mainstream web browsers (compatibility)

N.F.R. 8 - The system shall be documented with requirements document, design document, and UML diagrams (documentation)

N.F.R. 9 - The system shall answer at least 60 concurrent upload requests in less than 10 seconds (performance)

N.F.R. 10 - The user interface shall be available both in English and Swedish (user interface)

N.F.R. 11 - The system shall accept only .pdf submissions (data)

3.2 – Requirements Analysis

The requirement analysis is a phase of the requirements engineering process that requires time and focus. During this stage, all the identified requirements are individually analysed, and possible requirements issues and/or conflicts that might come to the engineers' attention are noted for further discussion and negotiation with the stakeholders.

3.2.1 – Requirements Refactoring

A good practice is to systematically check each requirement against a checklist in order to ease the process of detecting issues. The checklist questions that are used for analysing the thesis management system requirements are:

1. Does the requirement include premature design or implementation information?
2. Could the description of a requirement be broken down into several different requirements?
3. Is the requirement 'gold plating'? That is, a cosmetic addition to the system which is not really necessary.
4. Does the requirement mean that non-standard hardware or software must be used?
5. Is the requirement consistent with the business goals defined in the introduction to the requirements document?
6. Is the requirement ambiguous, i.e., could it be read in different ways by different people?
7. Is the requirement realistic given the technology which will be used to implement the system?
8. Is the requirement testable, that is, is it stated in such a way that test engineers can derive a test which can show if the system meets that requirement?

Notice that the previously listed requirements are already the outcome of requirements analysis process, while the requirements identified prior to the requirement analysis can be found next in this section and also in the appendices chapter of this assignment.

Requirement: F.R.1- As an end user, I should be able to log-in the system and log-out of the system
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. Yes. This requirement statement tackles two functionalities and, therefore, can be broken down into at least two separate requirements. 3. No. This is a functionality required to acquire the users' data and establish a communication channel with the customer. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. The services provided by the application should be used only by registered users. 6. No. This requirement statement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.2- As a student, I should be able to create submissions
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and, therefore, cannot be broken down. 3. No. This functionality is of high-level interest by the students and required for the application to provide a thorough range of thesis management services. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve one of the main goals of the application, the functionality of uploading a document must be implemented and delivered. 6. No. This requirement statement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.3- As a student, I should be able to read submissions
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and, therefore, cannot be broken down. 3. No. This functionality is of high-level interest by the students so they can read their own submissions to ensure the correct document version has been uploaded, for example. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve user satisfaction, it is necessary to meet the users expectations. 6. No. This requirement statement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.4- As a student, I should be able to update submissions
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and, therefore, cannot be broken down. 3. No. This functionality is of high-level interest by the students when they need to update a new version of their submissions. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve user satisfaction, this functionality must be implemented and delivered since students expect to be able to update a new version of their submission if they notice something is wrong on the old version, as long as it is before the deadline. 6. No. This requirement statement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.5- As a student, I should be able to delete submissions
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and therefore cannot be broken down. 3. No. This functionality can be of interest by the students. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve customer satisfaction, this functionality must be implemented and delivered since students must have the possibility to delete a previously uploaded document if they desire to re-submit again. 6. No. This requirement statement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.6- As a student, I should be able to view a list of supervisors
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and therefore cannot be broken down. 3. No. This functionality is of high-level interest by the students and supervisors. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to deliver one of the main functionalities of the application, students should be able to visualize all the supervisor that are currently registered in the system and willing to supervise students. 6. No. This requirement statement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.7- As a student, I should be able to suggest a supervisor
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and therefore cannot be broken down. 3. No. This requirement is necessary because it address one of the main services in the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve customer satisfaction, this functionality must be implemented and delivered since students have to suggest a supervisor to go further into the thesis process. 6. No, this requirement is well written and unambiguous. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.8- As a student, I should be able to view feedbacks
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and, therefore, cannot be broken down. 3. No. This functionality is necessary to ensure that the students registered in the system can improve their submissions based on the provided feedback. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to ease and centralize the management process, it is necessary that students can access the provided feedback through the system under development. 6. Yes. This requirement could be misunderstood and could be read that students can view the list of feedback but not the content of the feedback. Therefore, it should be modified to explicit that the students should be able to view the content of feedback 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.9- As a student, I should be able to view grade
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and, therefore, cannot be broken down. 3. No. This functionality is necessary to ensure that the grading system used is transparent and that students have access to their final grade in the project-degree course 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve the thesis management process unification, this functionality must be implemented and delivered since students should have access to their final grades once the thesis final submissions are evaluated. 6. No. This requirement is fairly straightforward. However, it could explicitly state that the grade being viewed is the final grade. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.10- As a coordinator, I should be able to view all user
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. No. This requirement statement tackles only one functionality and, therefore, cannot be broken down. 3. No. This functionality is necessary to ensure that the coordinator, at any time, have access to the people currently using the system, and can access individual information required for the decision-making process. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve a unified management process, it is necessary that person responsible by the management process has access to information of all the system users. 6. Possibly. This requirement statement could be more specific that the coordinator can view a list of all users registered in the system. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.11- As a coordinator, I should be able to create, read, update and delete supervisor, reader and opponent roles
Checklist: 1. No. This requirement does not specify implementation or design choices. 2. Yes, since this requirement tackles several functionalities and several users of the system, it can be broken down into several requirements, one for each functionality and for each user of the system. 3. No. This requirement contains essential functionalities that should be delivered in order to the users of the system to perform the tasks they are supposed to which are several of the main goals of the application. 4. No. These functionalities can be implemented and should operate in standard hardware or software. 5. Yes. The thesis management process involves several stages and each stage depends on the tasks that each role in the process has, therefore it is consistent with the business goals. 6. No. This requirement is fairly straightforward. However, it should be split into several requirements. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.12- As a coordinator, I should be able to view list of students assigned to each supervisor
Checklist: 1. No. This requirement does not include implementation details or design choices. 2. No. This requirement tackles only one objective and, therefore, cannot be broken down. 3. No. This requirement is a tool required so the coordinator of the degree-project course can analyse and manage the supervisor sand the supervision system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since the application aims to a distributed workload between supervisors and organised information system, the list of students categorised by supervisor is consistent with the business goals. 6. No. This requirement is fairly straightforward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.13- As a coordinator, I should be able to view all submissions
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and, therefore, cannot be broken down. 3. No. This requirement addresses one of the main features in the system since the coordinator should have access to all documents. Thus, this requirement is not a cosmetic addition. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. This requirement facilitates the management procedure, and it enhances the coordinator application usage experience. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.14- As a coordinator, I should be able to evaluate submissions
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and, therefore, cannot be broken down. 3. No. This requirement tackles one of main features in the application and is necessary for the thesis management procedure. Thus, this requirement is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. This requirement facilitates the management procedure and is required for the grading protocols. 6. Yes. This requirement could be rewritten to explain which submissions the coordinator should be able to evaluate. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.15- As a coordinator, I should be able to set deadlines
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and, therefore, cannot be broken down. 3. No. This requirement tackles one of the main features in the system since coordinators must establish the deadline for each submission. Thus, this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since the degree-project has a certain duration, it is necessary to define the deadlines in the system for each step of the thesis process. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.16- As a coordinator, I should be able to update deadlines
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and, therefore, cannot be broken down. 3. No. This requirement tackles one of the main features in the system since coordinators must be able to edit deadlines as necessary. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since this requirement increases the flexibility of the system since it allows the coordinator to adjust deadlines when needed. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: FR.17- As a coordinator, I should be able to submit the grade for a thesis project
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and, therefore, cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the coordinator role. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.18- As a supervisor, I should be able to confirm or decline supervision request
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. Yes. This requirement tackles two objective and therefore must be broken down to two separate requirements. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.19- As a supervisor, I should be able to assess submissions
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and therefore cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. Yes. This requirement could be rewritten to explain more details. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.20- As a supervisor, I should be able to view submissions
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and therefore cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. Yes. This requirement could be rewritten to explain more details. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.21- As a supervisor, I should be able to bid for reports of their preferences
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and therefore cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.22- As a supervisor, I should be able to feedback submissions
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and therefore cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. Yes. This requirement could be rewritten to explain more details. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.23- As an opponent, I should be able to view assigned report
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and therefore cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

Requirement: F.R.24- As an opponent, I should be able to give a feedback to an assigned report
Checklist: 1. No. This requirement does not include a premature design or implementation information. 2. No. This requirement tackles only one objective and therefore cannot be broken down. 3. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role. 6. No. This requirement is fairly straight-forward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

N.F.R.1 Usability: Customers should be able to be familiarised themselves with the system within 5 minutes
Checklist: 1. No. This requirement does not include any premature design or implementation information. 2. No, this requirement cannot be divided down into different requirements since it deals with only one category. 3. No. The system should be easy to use and user friendly. Therefore, this requirement is not cosmetic addition to the system. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. This requirement is consistent with and key to the business goals. 6. No. This requirement is unambiguous and straight-forward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

N.F.R.2 Safety: Customers' private data should be stored and processed in a manner that high security standards and data integrity is a priority.
Checklist: 1. No. This requirement does not include any premature design or implementation information. 2. No. This requirement cannot be broken down since it only deals and mentions security expectations. 3. No. Safety and security are major concerns of users and, therefore, essential for the application system. 4. No. This functionality can be implemented and should operate in standard hardware or software since only standard security and safety technologies will be used. 5. Yes. Security is an important part of all software today and expect by software users. Therefore, the thesis management application must be secure as well. 6. Yes. This requirement can be simplified in order to add clarity. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

N.F.R.3 Security: The log-in function should not be affected by SQL-injection
Checklist: 1. No. This requirement does not include any premature design or implementation information. SQL injection is a common attack, but it does not mean the system should use a SQL database. 2. No, this requirement cannot be divided down into different requirements since it deals with only one category. 3. No. This requirement does not gold-plate because security is an extremely important feature of web applications. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes, since the application deals with users' personal data and the application functionalities should be available only to registered users, the log-in correct validation is crucial. 6. No. This requirement is unambiguous and straight-forward. 7. Yes. Currently, this requirement can be implemented in almost all up to date technologies and is broadly supported. 8. Yes. It is possible to design test cases for such requirement and test if the system meets it.

N.F.R.4 Performance: The system should have the average response time for submitting a document less than 10 seconds
Checklist: 1. No. This requirement does not include any premature design or implementation information. 2. No. This requirement cannot be broken down since it only deals and mentions the system performance regarding the document submitting process. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

N.F.R.5 Reliability: The system should have 100% uptime during the duration of the course.
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

N.F.R.6 Privacy: The system should comply with GDPR
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is not easily testable. However, this is because the complication of the legislation.

N.F.R.7 Compatibility: The system should not depend on the customers operating system and should support the mainstream web browsers
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

N.F.R.8 Documentation: The system should be documented through UML diagrams, requirements document and design document
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

N.F.R.9 Performance: The system should answer at least 60 concurrent upload requests in less than 10 seconds.
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

N.F.R.10 User Interface: The user interface should be available both in Swedish and English
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

N.F.R.11 Data: The system should accept only .pdf submissions
Checklist: 1. The requirements do not include any design or implementation information. 2. No, the requirement cannot be broken down. 3. No, the requirement is not gold plating. 4. No special technologies are needed for satisfying the requirements. 5. Yes, the requirement is consistent with the business goals. 6. No, the requirement is clear. 7. Yes, the requirement is realistic. 8. The requirement is easily testable.

3.2.2 – Requirements Classification

After analysing the requirements through the checklist, it is noticed that refactoring some requirements is necessary in order to avoid ambiguity and combined requirements in some statements. Moreover, a second step to further refine the identified set of requirements is to categorise them using the faceted approach classification, and then assess their risks in a systematic manner. By classifying the requirements, it is possible to relate find harmonies and unforeseen connections between requirements that are defined into the same category. For the faceted approach, the selected categories (keywords) are:

System, User Interface, Database, Communications, and Security

Following are the new refactored requirement statements, their classification according to the above chosen categories, and their associated risks with high or medium level assessment. Please note that risks that not apply for a specific requirement or have low level assessment are not included in the risk assessment table.

F.R. 1 – Users shall be able to log-in the application

Classification	Risk Assessment
Communication User Interface Database Security	Safety and security risks External risks Performance risks

F.R. 2 – Users shall be able to log-out the application

Classification	Risk Assessment
Communication Security	Safety and security risks External risks

User Interface	

F.R. 3 – Students shall be able to create submissions

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks
System	Performance risks

F.R. 4 – Students shall be able to read submissions

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

F.R. 5 – Students shall be able to update submissions

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks
System	Performance risks

F.R. 6 – Students shall be able to delete submissions

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

F.R. 7 – Students shall be able to view a list of supervisors

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

F.R. 8 – Students shall be able to suggest a supervisor

Classification	Risk Assessment
Database User Interface Security	Database risks Performance risks

F.R. 9 – Students shall be able to view the content of feedbacks

Classification	Risk Assessment
Database User Interface System	Database risks Performance risks

F.R. 10 – Students shall be able to view their thesis final grade

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

Security	

F.R. 11 – The coordinator shall be able to view a list of all currently registered users

Classification	Risk Assessment
Database User Interface Security	Database risks

F.R. 12 – The coordinator shall be able to assign the supervisor role to users

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

F.R. 13 – The coordinator shall be able to view a list of students assigned to supervisors

Classification	Risk Assessment
Database User Interface System	Database risks

F.R. 14 – The coordinator shall be able to view all submissions

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

F.R. 15 – The coordinator shall be able to evaluate project description submissions

Classification	Risk Assessment
Database User Interface System	Database risks Process risks Performance risks

F.R. 16 – The coordinator shall be able to evaluate thesis submissions

Classification	Risk Assessment
Database User Interface System	Process risks Performance risks

F.R. 17 – The coordinator shall be able to evaluate final thesis submissions

Classification	Risk Assessment
Database User Interface System	Process risks Database risks Performance risks

F.R. 18 – The coordinator shall be able to set deadlines

Classification	Risk Assessment
Database User Interface	Process risks Database risks

F.R. 19 – The coordinator shall be able to update deadlines

Classification	Risk Assessment
Database User Interface	Process risks Database risks

F.R. 20 – The coordinator shall be able to submit the grade for a thesis project

Classification	Risk Assessment
Database User Interface	Process risks Database risks

F.R. 21 – Supervisors shall be able to confirm supervision requests

Classification	Risk Assessment
Database User Interface Security	Process risks Database risks

F.R. 22 – Supervisors shall be able to decline supervision requests

Classification	Risk Assessment
Database User Interface Security	Process risks Database risks

F.R. 23 – Supervisors shall be able to assess the students' project plan submissions

Classification	Risk Assessment
Database User Interface System	Performance risks Database risks

F.R. 24 – Supervisors shall be able to assess the students' report submissions

Classification	Risk Assessment
Database User Interface System	Database risks Performance risks

F.R. 25 – Supervisors shall be able to view students' project plan submissions

Classification	Risk Assessment
Database User Interface	Performance risks Database risks

F.R. 26 – Supervisors shall be able to view students' report submissions

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

F.R. 27 – Readers shall be able to bid for reports of their preference

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

F.R. 28 – Readers shall be able to feedback submissions

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks
System	Performance risks

F.R. 29 – Opponents shall be able to view assigned reports

Classification	Risk Assessment
Database	Database risks
User Interface	Performance risks

F.R. 30 – Opponents shall be able to give feedback to an assigned report

Classification	Risk Assessment
Database	Database risks
User Interface	Performance risks

F.R. 31 – The coordinator shall be able to view the role assigned to a user

Classification	Risk Assessment
Database User Interface	Database risks Process risks

F.R. 32 – The coordinator shall be able to assign the reader role to users

Classification	Risk Assessment
Database User Interface	Database risks Process risks

F.R. 33 – The coordinator shall be able to assign the opponent role to users

Classification	Risk Assessment
Database User Interface	Database risks Process risks

F.R. 34 – The coordinator shall be able to update the supervisor role of users

Classification	Risk Assessment
Database User Interface	Database risks Process risks

F.R. 35 – The coordinator shall be able to update the reader role of users

Classification	Risk Assessment
Database User Interface	Database risks Process risks

F.R. 36 – The coordinator shall be able to update the opponent role of users

Classification	Risk Assessment
Database User Interface	Database risks Process risks

N.F.R. 1 – Users shall be able to familiarise themselves with the application within 5 minutes

Classification	Risk Assessment
System User Interface	External risks

N.F.R. 2 – Users' private data shall be stored and processed in a manner that high security standards and data integrity are priorities

Classification	Risk Assessment
Database Security	Database risks Stability risks

N.F.R. 3 – The log-in functionality shall not be affected by SQL-injection

Classification	Risk Assessment
Security Database	Performance risks Database risks Implementation technology risks

N.F.R. 4– The system shall have average response time less than 10 seconds for submitting a document

Classification	Risk Assessment
System	Performance risks Implementation technology risks Database risks

N.F.R. 5 – The system shall have 100% uptime throughout the course duration

Classification	Risk Assessment
System	Performance risks External risks

N.F.R. 6 – The system shall comply with the General Data Protection Regulation (GDPR)

Classification	Risk Assessment
Database Security	Database risks Stability risks

N.F.R. 7 – The system shall not depend on the customers operation system and shall support the mainstream web browsers

Classification	Risk Assessment
System Database	Performance risks Implementation technology risks Stability risks

N.F.R. 8 – The system shall be documented with requirements document, design document, and UML diagrams

Classification	Risk Assessment
System	Schedule risks

N.F.R. 9 – The system shall answer at least 60 concurrent upload requests in less than 10 seconds

Classification	Risk Assessment
System	Performance risk External factors

N.F.R. 10 – The user interface shall be available both in English and Swedish

Classification	Risk Assessment
User Interface System	External factors Schedule risks

N.F.R. 11 – The system shall accept only .pdf submissions

Classification	Risk Assessment
System Database	Performance risks Database risks

3.3 – Requirements Validation

The requirements validation step has the main objective of checking the previously identified requirements for potential hidden problems, such as ambiguous, conflicting or lack of conformance with the requirements. These are problems that must be solved in order to move to next stage in the system engineering process. The requirements validation stage applied to the thesis management system is executed through a thorough traceability analysis and the establishment of requirements test cases.

3.3.1 – Traceability Analysis

An important step in the validation process is the application of requirements management techniques, such as traceability information maintenance (traceability analysis). In this thesis management system, the traceability information maintenance can be observed through the following traceability table

	Depends on																																														
	F.R. 1	F.R. 2	F.R. 3	F.R. 4	F.R. 5	F.R. 6	F.R. 7	F.R. 8	F.R. 9	F.R. 10	F.R. 11	F.R. 12	F.R. 13	F.R. 14	F.R. 15	F.R. 16	F.R. 17	F.R. 18	F.R. 19	F.R. 20	F.R. 21	F.R. 22	F.R. 23	F.R. 24	F.R. 25	F.R. 26	F.R. 27	F.R. 28	F.R. 29	F.R. 30	F.R. 31	F.R. 32	F.R. 33	F.R. 34	F.R. 35	F.R. 36	N.F.R. 1	N.F.R. 2	N.F.R. 3	N.F.R. 4	N.F.R. 5	N.F.R. 6	N.F.R. 7	N.F.R. 8	N.F.R. 9	N.F.R. 10	N.F.R. 11
F.R. 1	X																																					X	X		X	X	X	X			
F.R. 2	X	X																																							X	X	X	X			
F.R. 3	X		X																																					X	X	X	X	X	X		X
F.R. 4	X		X	X																																					X	X	X	X			
F.R. 5	X		X		X	X																																		X	X	X	X	X			X
F.R. 6	X		X			X																																			X	X	X	X			
F.R. 7	X						X																															X			X	X	X	X			
F.R. 8	X						X	X																														X			X	X	X	X			
F.R. 9	X		X				X	X	X					X	X	X					X		X	X				X		X											X	X	X	X			
F.R. 10	X		X				X	X	X	X		X		X	X	X	X	X		X	X		X	X	X		X	X	X	X		X	X					X			X	X	X	X			
F.R. 11	X										X																											X			X	X	X	X			
F.R. 12	X										X	X																										X			X	X	X	X			
F.R. 13	X						X	X					X								X																	X			X	X	X	X			
F.R. 14	X		X											X																								X			X	X	X	X			
F.R. 15	X		X											X	X																									X	X	X	X	X			X
F.R. 16	X		X				X	X	X			X		X	X	X		X			X		X	X			X	X	X	X		X	X					X		X	X	X	X	X			X
F.R. 17	X		X				X	X	X			X		X	X	X	X	X			X		X	X			X	X	X	X		X	X					X		X	X	X	X	X			X
F.R. 18	X																	X																							X	X	X	X			
F.R. 19	X																	X	X																						X	X	X	X			
F.R. 20	X		X				X	X	X			X		X	X	X	X	X		X	X		X	X	X		X	X	X	X		X	X					X			X	X	X	X			
F.R. 21	X						X	X													X																	X			X	X	X	X			
F.R. 22	X						X	X														X																X			X	X	X	X			
F.R. 23	X		X				X	X				X									X		X		X															X	X	X	X	X			X
F.R. 24	X		X				X	X				X		X	X	X					X		X	X	X	X	X					X	X					X		X	X	X	X	X			X
F.R. 25	X		X				X	X				X													X																X	X	X	X			
F.R. 26	X		X				X	X				X		X	X						X		X		X	X															X	X	X	X			
F.R. 27	X		X				X	X				X		X	X	X					X		X		X		X					X									X	X	X	X			
F.R. 28	X		X				X	X				X		X	X	X					X		X		X		X	X				X	X					X			X	X	X	X			X
F.R. 29	X		X				X	X				X		X	X	X					X		X		X		X		X			X	X					X			X	X	X	X			
F.R. 30	X		X				X	X				X		X	X	X					X		X		X		X		X	X		X	X					X		X	X	X	X	X			X
F.R. 31	X																														X							X			X	X	X	X			
F.R. 32	X																														X	X						X			X	X	X	X			
F.R. 33	X																														X		X					X			X	X	X	X			
F.R. 34	X											X																			X	X	X	X				X			X	X	X	X			
F.R. 35	X											X																			X	X	X		X			X			X	X	X	X			
F.R. 36	X											X																			X	X	X			X		X			X	X	X	X			

	F.R.1	F.R.2	F.R.3	F.R.4	F.R.5	F.R.6	F.R.7	F.R.8	F.R.9	F.R.10	F.R.11	F.R.12	F.R.13	F.R.14	F.R.15	F.R.16	F.R.17	F.R.18	F.R.19	F.R.20	F.R.21	F.R.22	F.R.23	F.R.24	F.R.25	F.R.26	F.R.27	F.R.28	F.R.29	F.R.30	F.R.31	F.R.32	F.R.33	F.R.34	F.R.35	F.R.36	N.F.R.1	N.F.R.2	N.F.R.3	N.F.R.4	N.F.R.5	N.F.R.6	N.F.R.7	N.F.R.8	N.F.R.9	N.F.R.10	N.F.R.11
N.F.R. 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X						X	X		X	
N.F.R. 2																																					X											
N.F.R. 3																																						X										
N.F.R. 4																																							X									
N.F.R. 5																																								X								
N.F.R. 6																																									X							
N.F.R. 7																																										X						
N.F.R. 8																																											X					
N.F.R. 9																																												X				
N.F.R. 10																																													X			
N.F.R. 11																																														X		

3.3.2 – Requirements Test Cases

The second step in the requirements validation process for the thesis management system is the proposal of a test case for each of the requirements in order to check if the system achieves each and all the identified requirements. However, it is important to take into consideration that, in this requirement document, the test cases and which aspects of the requirements that can be tested are vaguely defined. During the implementation process, these test cases definitions are translated into actual test cases.

ID: TC1
F.R. 1 – Users shall be able to log-in the application
To be tested: Logging module - Logging-in functionality - E-mail account provided validation - Password validation - System database - User interface

ID: TC2
F.R. 2 – Users shall be able to log-out the application
To be tested: Logging module - Logging-out functionality - System database - User interface

ID: TC3
F.R. 3 – Students shall be able to create submissions
To be tested: Submission module - Uploading file functionality - User interface - System database

ID: TC4
F.R. 4 – Students shall be able to read submissions
To be tested: Submission module - Viewing file functionality - User interface - System database

ID: TC5
F.R. 5 – Students shall be able to update submissions
To be tested: Submission module - Updating file functionality - User interface - System database

ID: TC6
F.R. 6 – Students shall be able to delete submissions
To be tested: Submission module - Deleting file functionality - User interface - System database

ID: TC7
F.R. 7 – Students shall be able to view a list of supervisors
To be tested: Students module - Viewing list functionality - User interface - Database functionality

ID: TC8
F.R. 8 – Students shall be able to suggest a supervisor
To be tested: Student module - Supervisor suggestion functionality - Security and privacy validation - User interface - Database functionality

ID: TC9
F.R. 9 – Students shall be able to view the content of feedbacks
To be tested: Students module - Viewing file functionality - User interface - Security and privacy validation - Database functionality

ID: TC10
F.R. 10 – Students shall be able to view their thesis final grade
To be tested: Students module - Viewing/displaying grade functionality - User interface - Security and privacy validation - Database functionality

ID: TC11
F.R. 11 – The coordinator shall be able to view a list of all currently registered users
To be tested: Coordinator module - Viewing all the registered users functionality - User interface - Security and privacy validation - Database functionality

ID: TC12
F.R. 12 – The coordinator shall be able to assign the supervisor role to users
To be tested: Coordinator module - Creating supervisors' role functionality - Assigning role to user functionality - User interface - Database functionality

ID: TC13
F.R. 13 – The coordinator shall be able to view a list of students assigned to supervisors
To be tested: Coordinator module - Viewing list of students assigned to a supervisor functionality - User interface - Database functionality

ID: TC14
F.R. 14 – The coordinator shall be able to view all submissions
To be tested: Coordinator module - Viewing file functionality - User interface - Security and privacy validation - Database functionality

ID: TC15
F.R. 15 – The coordinator shall be able to evaluate project description submissions
To be tested: Coordinator module - Uploading document functionality - User interface - Database functionality

ID: TC16
F.R. 16 – The coordinator shall be able to evaluate thesis submissions
To be tested: Coordinator module - Uploading document functionality - User interface - Database functionality

ID: TC17
F.R. 17 – The coordinator shall be able to evaluate final thesis submissions
To be tested: Coordinator module - Uploading document functionality - User interface - Database functionality

ID: TC18
F.R. 18 – The coordinator shall be able to set deadlines
To be tested: Coordinator module - Setting deadline for submission functionality - User interface - Database functionality to store deadlines

ID: TC19
F.R. 19 – The coordinator shall be able to update deadlines
To be tested: Coordinator module - Updating deadline for submission functionality - User interface - Database functionality

ID: TC20
F.R. 20 – The coordinator shall be able to submit the grade for a thesis project
To be tested: Coordinator module - Setting students' final grade functionality - User interface - Database functionality

ID: TC21
F.R. 21 – Supervisors shall be able to confirm supervision requests
To be tested: Supervisor module - Confirming supervision request functionality - User interface - Database functionality

ID: TC22
F.R. 22 – Supervisors shall be able to decline supervision requests
To be tested: Supervisor module - Declining supervision request functionality - User interface - Database functionality

ID: TC23
F.R. 23 – Supervisors shall be able to assess the students' project plan submissions
To be tested: Supervisor module - Uploading file functionality - User interface - Database functionality

ID: TC24
F.R. 24 – Supervisors shall be able to assess the students' report submissions
To be tested: Supervisor module - Uploading file functionality - User interface - Database functionality

ID: TC25
F.R. 25 – Supervisors shall be able to view students' project plan submissions
To be tested: Supervisor module - Viewing file functionality - User interface - Database functionality

ID: TC26
F.R. 26 – Supervisors shall be able to view students' report submissions
To be tested: Supervisor module - Viewing file functionality - User interface - Database functionality

ID: TC27
F.R. 27 – Readers shall be able to bid for reports of their preference
To be tested: Readers module - Bidding functionality - User interface - Database functionality

ID: TC28
F.R. 28 – Readers shall be able to feedback submissions
To be tested: Readers module - Uploading file functionality - User interface - Database functionality

ID: TC29
F.R. 29 – Opponents shall be able to view assigned reports
To be tested: Opponent module - Viewing file functionality - User interface - Database functionality

ID: TC30
F.R. 30 – Opponents shall be able to give feedback to an assigned report
To be tested: Opponent module - Uploading file functionality - User interface - Database functionality

ID: TC31
F.R. 31 – The coordinator shall be able to view the role assigned to a user
To be tested: Coordinator module - Viewing the users' roles functionality - User interface - Database functionality

ID: TC32
F.R. 32 – The coordinator shall be able to assign the reader role to users
To be tested: Coordinator module - Creating reader role functionality - Assigning role to user functionality - User interface - Database functionality

ID: TC33
F.R. 33 – The coordinator shall be able to assign the opponent role to users
To be tested: Coordinator module - Creating opponent role functionality - Assigning role to user functionality - User interface - Database functionality

ID: TC34
F.R. 34 – The coordinator shall be able to update the supervisor role of users
To be tested: Coordinator module - Editing role of user functionality - User interface - Database functionality

ID: TC35
F.R. 35 – The coordinator shall be able to update the reader role of users
To be tested: Coordinator module - Editing role of user functionality - User interface - Database functionality

ID: TC36
F.R. 36 – The coordinator shall be able to update the opponent role of users
To be tested: Coordinator module - Editing role of user functionality - User interface - Database functionality

ID: TC37
N.F.R. 1 – Users shall be able to familiarise themselves with the application within 5 minutes
To be tested: Usability - User interface

ID: TC38
N.F.R. 2 – Users' private data shall be stored and processed in a manner that high security standards and data integrity are priorities
To be tested: Safety - User interface - Security - Privacy - Database

ID: TC39
N.F.R. 3 – The log-in functionality shall not be affected by SQL-injection
To be tested: Security - Database - Security

ID: TC40
N.F.R. 4– The system shall have average response time less than 10 seconds for submitting a document
To be tested: Performance - User interface - Security

ID: TC41
N.F.R. 5 – The system shall have 100% uptime throughout the course duration
To be tested: Safety - User interface - Security

ID: TC42
N.F.R. 6 – The system shall comply with the General Data Protection Regulation (GDPR)
To be tested: Security - Integrity data system

ID: TC43
N.F.R. 7 – The system shall not depend on the customers operation system and shall support the mainstream web browsers
To be tested: Compatibility - User interface - Database functionalities

ID: TC44
N.F.R. 8 – The system shall be documented with requirements document, design document, and UML diagrams
To be tested: Documentation

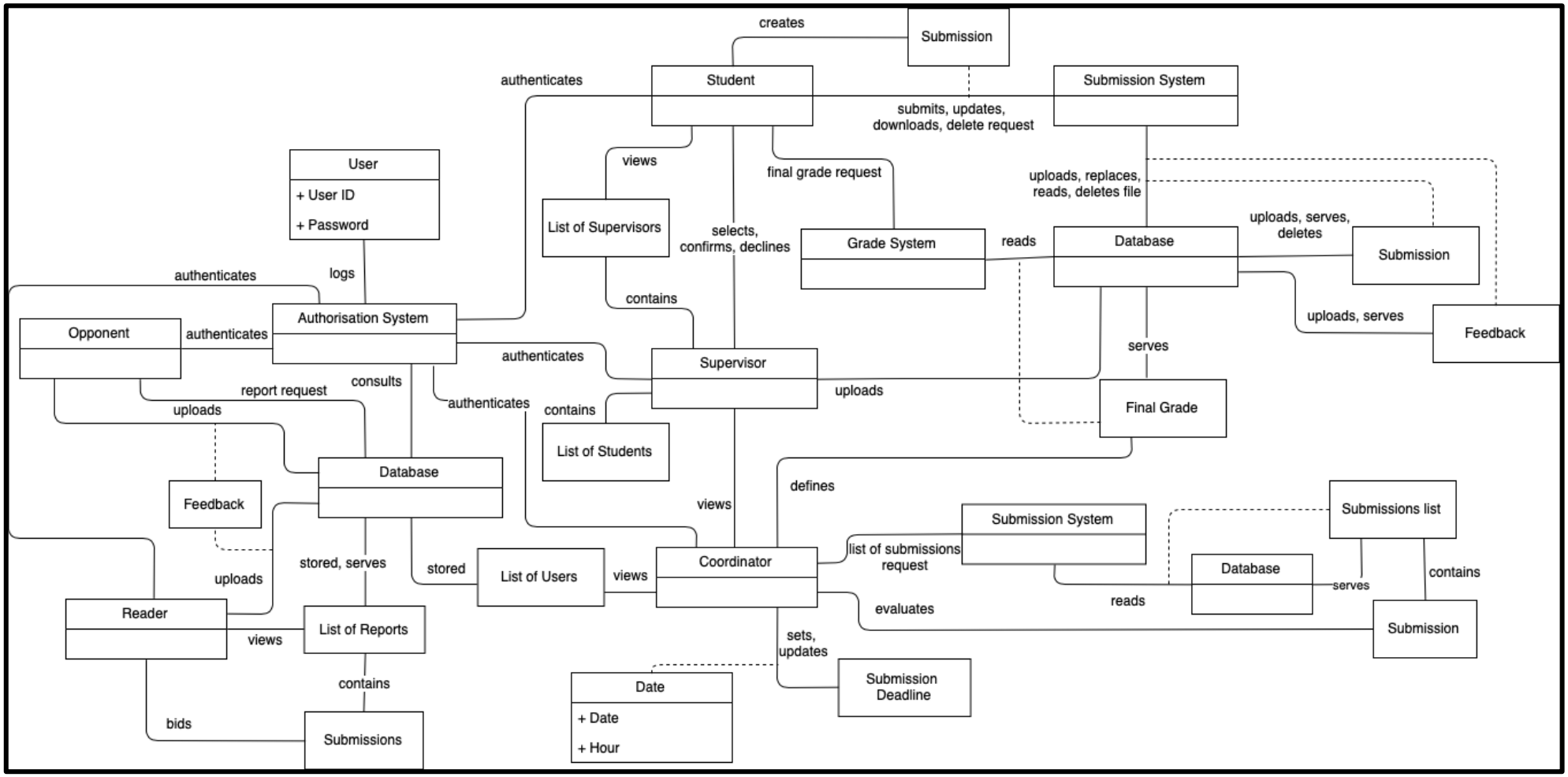
ID: TC45
N.F.R. 9 – The system shall answer at least 60 concurrent upload requests in less than 10 seconds
To be tested: Performance

ID: TC46
N.F.R. 10 – The user interface shall be available both in English and Swedish
To be tested: User interface - Language spelling and grammar - User interface

ID: TC47
N.F.R. 11 – The system shall accept only .pdf submissions (data)
To be tested: User interface - Resources

3.4 – Requirements Modelling

In this last chapter of the requirements document, the most important and fundamental requirements are modelled in an explanatory domain approach diagram following the UML Class Diagram guidelines. The diagram shows an overview of the requirements and several of the most important relationships. It is important to notice that some information was omitted though, such as the multiplicity of the relationships between entities, in order to simplify the diagram. Moreover, it is also important to notice that some entities, such as database, are duplicate to avoid association lines crossing each other, but they represent the same entity in the system. The dashed line represents the associations that some objects directly affect. Finally, notice that some associations have been combined together in the same association line in order to simplify the diagram again, but they represent different requirements.


creates
Submission
authenticates
Student
Submission System
submits, updates,
downloads, delete request
User
+ User ID
+ Password
views
final grade request
uploads, replaces,
reads, deletes file
List of Supervisors
selects,
confirms, declines
Grade System
reads
Database
uploads, serves,
deletes
Submission
authenticates
logs
contains
Authorisation System
Opponent
authenticates
uploads, serves
Feedback
authenticates
Supervisor
serves
report request
consults
uploads
authenticates
contains
uploads
Final Grade
List of Students
Database
Feedback
views
defines
Submission System
Submissions list
list of submissions
request
stored, serves
stored
List of Users
views
Coordinator
Database
contains
uploads
reads
serves
Reader
evaluates
Submission
views
List of Reports
sets,
updates
contains
Date
+ Date
+ Hour
Submission
Deadline
bids
Submissions

4 – Appendices

4.1 – Initial Functional Requirements

F.R.1- As an end user, I should be able to log-in the system and log-out of the system

F.R.2- As a student, I should be able to create submissions

F.R.3- As a student, I should be able to read submissions

F.R.4- As a student, I should be able to update submissions

F.R.5- As a student, I should be able to delete submissions

F.R.6- As a student, I should be able to view a list of supervisors

F.R.7- As a student, I should be able to suggest a supervisor

F.R.8- As a student, I should be able to view feedbacks

F.R.9- As a student, I should be able to view grade

F.R.10- As a coordinator, I should be able to view all user.

F.R.11- As a coordinator, I should be able to create, read, update and delete supervisor, reader and opponent roles

F.R.12- As a coordinator, I should be able to view list of students assigned to each supervisor

F.R.13- As a coordinator, I should be able to view all submissions

F.R.14- As a coordinator, I should be able to evaluate submissions

F.R.15- As a coordinator, I should be able to set deadlines

F.R.16- As a coordinator, I should be able to update deadlines

F.R.17- As a coordinator, I should be able to submit the grade for a thesis project

F.R.18- As a supervisor, I should be able to confirm or decline supervision request

F.R.19- As a supervisor, I should be able to assess submissions

F.R.20- As a supervisor, I should be able to view submissions

F.R.21- As a supervisor, I should be able to bid for reports of their preferences

F.R.22- As a supervisor, I should be able to feedback submissions

F.R.23- As an opponent, I should be able to view assigned report

F.R.24- As an opponent, I should be able to give a feedback to an assigned report

4.2 – Initial Non-Functional Requirements

N.F.R.1- Usability: customers should be able to be familiarised themselves with the system within 5 minutes

N.F.R.2- Safety: customers' private data should be stored in a system that offers high standards of security and integrity in a manner that only authorized people can access this information and should be confidential

N.F.R.3 - Security: the log-in function should not be affected by SQL-injection

N.F.R.4 - Performance: the system should have the average response time for submitting a document less than 10 seconds

N.F.R.5 - Reliability: the system should have 100% uptime during the duration of the course.

N.F.R.6 - Privacy: the system should comply with GDPR

N.F.R.7 - Compatibility: the system should not depend on the customers operating system and should support the mainstream web browsers

N.F.R.8 - Documentation: the system should be documented through UML diagrams, requirements document and design document

N.F.R.9 - The system should answer at least 60 concurrent upload requests in less than 10 seconds.

N.F.R.10- User Interface: The user interface should be available both in Swedish and English

N.F.R.11- the system should accept only.pdf submissions