Linnéuniversitetet
Kalmar Växjö

Requirements Document

Thesis Management System

Author: Ahmad Abdilrahim
Ahmad Sadia
Caesar Alhawi
Cristian Babes
Marcello Pietro Vendruscolo
Supervisor: Mirko D'Angelo
Semester: Spring 2019
Course code: 2DV603

Table of contents

1 – Introduction

The first section of this document explains the document structure and its purpose. It provides a general overview and knowledge about the context in which the product is being developed as well as technological terms and references required that might be indispensable in order to better understand the document. Finally, it presents what the remaining sections of the paper discusses 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 the system under development. This document's target audience is all the system stakeholders (people directly or indirectly involved in the project) and its goal is to clearly explain to the range of different readers the functionality of the system without discussing the implementation details. The structure of this requirements document follows the conventional IEEE/ANSI 830-1993 standard in order to facilitate the understanding of itself. However, in order to fully understand this document, no specific technical knowledge is required, but a general understanding about information systems is suggested.

1.2 – Scope of the Product

The environment in which this thesis management system is being developed for is a fast-growing and technological university in Sweden. Currently, all the administrative processes and management of the graduating students' thesis are done following non-formal and manual procedures and the communication between parties is done through e-mail. However, looking to the future, it is essential to create a solution that takes advantage of the current available technology in order to enable the university's expansion and to optimise the thesis management processes. It is fundamental to match the expectations of the people involved, such as students and professors, with a technological solution since it potentially can standardise procedures, assemble all the information in a single system, and, consequently, optimise productivity.

1.3 – Definitions, Acronyms, and Abbreviations

This subchapter 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 – Remainder of the Document

The remainder of this document provides an overall report of the product being developed, as well as its functional and non-functional requirements and its constraints and dependencies. It also provides step-by-step insight on the requirements engineering process as well as UML class diagram modelling for some requirements.

2 – General description

The second section of this document provides a high-level general overview and knowledge about the thesis management system that is currently being designed and developed. However, this paper also explains beyond the system itself and gives insight about the potential users of the product.

2.1 – Product Perspective

The thesis management system application that is being developed will be responsible for providing all users with an efficient and pleasant experience when it comes the time for students to take their degree project course before graduation. The software will be responsible for the entire management process from the beginning showing potential supervisors to the completion when students upload their final submission and receive the final grade.

2.2 – Product Functions

The main purpose of this thesis management system application is to ease and facilitate users experience when it comes to the degree project course. At the moment of this writing, the final product is not fully designed yet, since it is being produced following an iterative development approach. The main general functionalities that the system should embody are the described by the sub-systems below, but it is important to remind that secondary requirements can be modified, added or removed as the development process iterates. However, since the system is a customer driven application, it will be designed to have high interactivity and utility maximization.

- User authentication to the system
- Uploading documents to the system
- Visualisation, suggestion and confirmation of supervisor
- Establishment of deadlines for different submissions
- Evaluation of documents uploaded to the system
- Visualisation of users registered in the system

- Assignment of different responsibilities to different users

2.3 – User Characteristics

In essence, this thesis management service system application targets two categories of users: the first group is composed by the student population of the computer science department of the Linnaeus University who are soon graduating and are taking the degree project course. The users in this first category have no administrative powers on the system. The second group is composed by scientifically qualified people related to the computer science department who are, in different levels, involved in the course. The users in this second category have some administrative power according to their assigned tasks. They must also have some knowledge about the system, but they should receive education regarding it through the company.

2.4 – General Constraints

The software system is being developed for the Linnaeus University and, taking into consideration that this system deals with personal information of students and professors from with different backgrounds, the project's most important present constraints are related to data security, data privacy, and system performance. More and more people are becoming aware of the needs to keep their personal data safe and secure in the systems to avoid inappropriate use and this concern is reflecting in new laws and regulations. Furthermore, people are expecting more and more interactive and immediate responses, therefore the system should take it into consideration to match the users' expectations. Therefore, the constraints are on designing and implementation strategies to ensure the users concerns and needs are met.

2.5 – Assumptions and Dependencies

Few assumptions have already been made at this point of the project development:

- The system has access to the Linnaeus University's user database since the system is being developed for the institution and will potentially be integrated with the already existing systems

- Users do not need to be experts in informatics, they just need common basic knowledge on how to use the Internet to continue with the reservation
- The system development environment is to be long-established in the upcoming iterations and will be proven throughout the functional designing phase.
- 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
- Scope of the project will remain as it is, and if necessary, changes will follow approval procedures

3 – Requirements Engineering

In this third section of the 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 in 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.

- Stakeholders Identification: Stakeholder is any person or organisation that are directly or indirectly involved or affected by that 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: Linnaeus University, specifically the computer science department which requested the development of the system, including the university board members
  - Project managers: this category is composed by Mauro Caporuscio and Mirko D'Angelo who are the people in charge of planning the activities and resources, organizing the project teams, managing the time, monitoring the project progress and ensuring satisfaction of client and customers
  - End-users: students, professors and people who will eventually need to use the system to manage thesis activities
  - Company's employees: this category includes the people who work at Linnaeus University and will have to acquire some knowledge about the new 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 actually 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.

- Functional and Non-Functional Requirements Elicitation and Labelling: 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 list essentially the functional objectives of the system (what the system should do) while the non-functional requirements place qualitative and security measures on how the system should perform its objectives (how the system should perform).

3.1.1 – 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 the content of feedback

F.R.9- As a student, I should be able to view my final grade for the course

F.R.10- As a coordinator, I should be able to view a list of all users registered in the system

F.R.11- As a coordinator, I should be able to assign a supervisor role to a user

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 students' project description submissions

F.R.15- As a coordinator, I should be able to evaluate students' report submissions

F.R.16- As a coordinator, I should be able to evaluate students final report submissions

F.R.17- As a coordinator, I should be able to set deadlines

F.R.18- As a coordinator, I should be able to update deadlines

F.R.19- As a coordinator, I should be able to submit the grade for a thesis project

F.R.20- As a supervisor, I should be able to confirm supervision request

F.R.21- As a supervisor, I should be able to decline supervision request

F.R.22- As a supervisor, I should be able to assess the students' project plan submissions

F.R.23- As a supervisor, I should be able to assess the students report submissions

F.R.24- As a supervisor, I should be able to view students' project plan submissions

F.R.25- As a supervisor, I should be able to view students report submissions

F.R.26- As a reader, I should be able to bid for reports of my preferences

F.R.27- As a reader, I should be able to feedback submissions

F.R.28- As an opponent, I should be able to view assigned report

F.R.29- As an opponent, I should be able to give a feedback to an assigned report

F.R.30-As a coordinator, I should be able to assign a reader role to a user

F.R.31-As a coordinator, I should be able to assign an opponent role to a user

F.R.32-As a coordinator, I should be able to read a user's role

F.R.33-As a coordinator, I should be able to update the supervisor role of a user

F.R.34-As a coordinator, I should be able to update the reader role of a user

F.R.35-As a coordinator, I should be able to update the opponent role of a user

F.R.36-As a coordinator, I should be able to delete the supervisor role of a user

F.R.37-As a coordinator, I should be able to delete the reader role of a user

F.R.38-As a coordinator, I should be able to delete the opponent role of a user

3.1.2 – Non-Functional Requirements

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

N.F.R.2- Safety: Users' private data should be stored and processed in a manner that high security standards and data integrity is a priority.

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 - Performance: 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- Data: The system should accept only .pdf submissions

3.2 – Requirements Analysis

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

The list containing the requirements written before the requirement analysis process can be found in the Appendix section.

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. Yes. The project engineers already have a fairly idea of how this requirement can be designed and implemented based on previous projects. 2. No. This requirement statement tackles only one functionality and therefore cannot be broken down. 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. In order to access to the provided services, a user has to log in to the system 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. Yes. The project engineers already have a fairly idea of how this requirement can be designed and implemented based on previous projects. 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. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve user satisfaction, these tools 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. Yes. The project engineers already have a fairly idea of how the first part this requirement can be designed and implemented based on previous projects. 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.. 4. No. This functionality can be implemented and should operate in standard hardware or software. 5. Yes. In order to achieve user satisfaction. 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. Yes. The project engineers already have a fairly idea of how the first part this requirement can be designed and implemented based on previous projects.
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 edit their submissions.
4. No. This functionality can be implemented and should operate in standard hardware or software in cooperation with an insurance service provider.
5. Yes. In order to achieve user satisfaction, this functionality 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.5- As a student, I should be able to delete submissions

Checklist:

1. Yes. The project engineers already have a fairly idea of how the first part this requirement can be designed and implemented based on previous projects.
2. No. This requirement statement tackles only one functionality and therefore cannot be broken down.
3. No. This functionality is interest by the student.
4. No. This functionality can be implemented and should operate in standard hardware or software in cooperation with an insurance service provider.
5. Yes. In order to achieve customer satisfaction, this functionality 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.6- As a student, I should be able to view a list of supervisors

Checklist:

1. Yes. The project engineers already have a fairly idea of how this requirement can be designed and implemented..
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 business to save time and effort and avoid misunderstandings and equivocate actions by the office staff.
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. Ensuring that the customer picks up the correct car in the correct day is to promote customer satisfaction.
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. Partially. The project engineers already have a fairly idea of how this requirement can be designed and implemented based on previous projects.
2. No. This requirement statement tackles only one functionality and therefore cannot be broken down.
3. No. This functionality is necessary to because it is 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.
6. Possibly. This requirement could better state to avoid the misunderstanding that the office staff will prepare the car itself while what actually happen is that the office staff contact third people to clean and check and prepare the car for pick up.
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 registered in the system
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.
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 and the supervision system.
4. No. This functionality can be implemented and should operate in standard hardware or software.
5. Yes. Since the system aims to a distributed workload and organised information, 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 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.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 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. 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.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 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.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 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:

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:

9. No. This requirement does not include a premature design or implementation information.
10. No. This requirement tackles only one objective and therefore cannot be broken down.
11. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system.
12. No. This functionality can be implemented and should operate in standard hardware or software.
13. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role.
14. No. This requirement is fairly straight-forward.
15. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported.
16. 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:

17. No. This requirement does not include a premature design or implementation information.
18. No. This requirement tackles only one objective and therefore cannot be broken down.
19. No. This requirement is a main feature in the system thus this is not a cosmetic addition to the system.
20. No. This functionality can be implemented and should operate in standard hardware or software.
21. Yes. Since it facilitates the management and it satisfies the user experience for the supervisor role.
22. No. This requirement is fairly straight-forward.
23. Yes. Currently this requirement can be implemented in almost all up to date technologies and is broadly supported.
24. 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. This requirement doesn't include any premature design or implementatio information.
2. No, this cannot be divided down into different requirements.
3. The system should be easy to use (aka user friendly) so this requirement is not suga coating.
4. No, no non-standard information will be used.
5. The requirement is consistent with and key to the business goals.
6. The requirements is not ambiguous.
7. Yes, the requirement is realistic.
8. Yes, the requirement is easily testable.

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, the requirement only mentions security expectations.
2. No, the requirement cannot be broken down.
3. No, safety and security are essential parts of today's software.
4. Only standard security and safety technologies will be used.
5. Yes, security is an important part of all software today.
6. The requirement can be simplified, modification that can help add clarity.
7. Yes, the requirement is realistic given the technologies to be used.
8. Yes, the requirement is easily testable.

N.F.R.3

Security: The log-in function should not be affected by SQL-injection

Checklist:

1. No, SQL injection is a common attack, but it does not mean the system should us a SQL Database.
2. No, the requirement only targets on part of the system.
3. Security is a very important part of all online software.
4. No, SQL injection, while being common there are many of off-the-shelf solution to it.
5. Yes, security is a very important part of software.
6. No, the requirement is written in a clear way.
7. Yes, the requirement is easy to implement in the system.
8. Yes, the requirement is easily testable.

N.F.R.4

Performance: The system should have the average response time for submitting a document less than 10 seconds

Checklist:

1. The requirements doesn’t 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.5

Reliability: The system should have 100% uptime during the duration of the course.

Checklist:

1. The requirements doesn’t 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 doesn’t 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 complications o 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 doesn't 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 doesn't 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 doesn’t 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 doesn’t 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 doesn't 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.

After analysing the requirements through a systematic checklist, it is noticed that refactoring some requirements is necessary in order to avoid ambiguity and combined requirements in some statement cases. In addition, a second step to further refine the set of requirements is to classify them using the faceted approach and then assess their risks in a systematic manner. By classifying the requirements, it is possible to relate requirements that are defined into the same category, find harmonies and unforeseen connections between requirements. 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- As an end user, I should be able to log-in the system and log-out of the system

Classification	Risk Assessment
Communication Security	Safety and security risks External risks

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

Classification	Risk Assessment
Database User Interface	Database risks Process risks

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

Classification	Risk Assessment
Database User Interface	Database risks Process risks

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

Classification	Risk Assessment
Database User Interface	Database risks Process risks

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

Classification	Risk Assessment
Database User Interface	Database risks Process risks

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

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

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

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

- F.R.8- As a student, I should be able to view the content of feedback

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

- F.R.9- As a student, I should be able to view my final grade for the course

Classification	Risk Assessment
Database User Interface	Performance risks

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

Classification	Risk Assessment
Database User Interface	Database risks

- F.R.11- As a coordinator, I should be able to assign a supervisor role to a user

Classification	Risk Assessment
Database User Interface	Database risks performance risks

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

Classification	Risk Assessment
Database User Interface	Database risks

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

Classification	Risk Assessment
Database User Interface	Database risks

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

Classification	Risk Assessment
Database User Interface	Database risks Process risks

- F.R.15- As a coordinator, I should be able to evaluate students report submissions

Classification	Risk Assessment
Database User Interface	Process risks

- F.R.16- As a coordinator, I should be able to evaluate students final report submissions

Classification	Risk Assessment
Database User Interface	Process risks Database risks

- F.R.17- As a coordinator, I should be able to set deadlines

Classification	Risk Assessment
Database User Interface	Process risks Database risks

- F.R.18- As a coordinator, I should be able to update deadlines

Classification	Risk Assessment
Database User Interface	Process risks Database risks

- F.R.19- As a coordinator, I should be able to submit the grade for a thesis project

Classification	Risk Assessment
Database User Interface	Process risks Database risks

- F.R.20- As a supervisor, I should be able to confirm supervision request

Classification	Risk Assessment
Database User Interface	Process risks Database risks

- F.R.21- As a supervisor, I should be able to decline supervision request

Classification	Risk Assessment
Database User Interface	Process risks Database risks

- F.R.22- As a supervisor, I should be able to assess the students' project plan submissions

Classification	Risk Assessment
Database User Interface	Performance risks Database risks

- F.R.23- As a supervisor, I should be able to assess the students report submissions

Classification	Risk Assessment
Database User Interface	Database risks

- F.R.24- As a supervisor, I should be able to view students' project plan submissions

Classification	Risk Assessment
Database User Interface	Performance risks Database risks

- F.R.25- As a supervisor, I should be able to view students report submissions

Classification	Risk Assessment
Database User Interface	Database risks

- F.R.26- As a reader, I should be able to bid for reports of my preferences

Classification	Risk Assessment
Database User Interface	Database risks Process risks

- F.R.27- As a supervisor, I should be able to feedback submissions

Classification	Risk Assessment
Database User Interface	Database risks Process risks

- F.R.28- As an opponent, I should be able to view assigned report

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

- F.R.29- As an opponent, I should be able to give a feedback to an assigned report

Classification	Risk Assessment
Database User Interface	Database risks Performance risks

- F.R.30-As a coordinator, I should be able to assign a reader role to a user

Classification	Risk Assessment
Database User Interface	Database risks Process risks

- F.R.31-As a coordinator, I should be able to assign a opponent role to a user

Classification	Risk Assessment
Database User Interface	Database risks Process risks

- F.R.32-As a coordinator, I should be able to read a user's role

Classification	Risk Assessment
Database User Interface	Database risks Process risks

- F.R.33-As a coordinator, I should be able to update the supervisor role of a user

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

- F.R.34-As a coordinator, I should be able to update the reader role of a user

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

- F.R.35-As a coordinator, I should be able to update the opponent role of a user

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

- F.R.36-As a coordinator, I should be able to delete the supervisor role of a user

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

- F.R.37-As a coordinator, I should be able to delete the reader role of a user

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

- F.R.38-As a coordinator, I should be able to delete the opponent role of a user

Classification	Risk Assessment
Database	Database risks
User Interface	Process risks

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

Classification	Risk Assessment
System User Interface	External risks

- 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

Classification	Risk Assessment
Database Security	Database risks Stability risks

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

Classification	Risk Assessment
Security	Performance risks Implementation technology risks

- N.F.R.4- Performance: the online service application should also have fast data transmission and response rates with average response time for submitting a document less than 10 seconds

Classification	Risk Assessment
System	Performance risks Implementation technology risks

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

Classification	Risk Assessment
System	Performance risks External risks

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

Classification	Risk Assessment
Database	Database risks Stability risks

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

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

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

Classification	Risk Assessment
System	Schedule risks

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

Classification	Risk Assessment
System	Performance risk External factors

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

Classification	Risk Assessment
User Interface	External factors Schedule risks

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

Classification	Risk Assessment
System Database	Performance risks Database risks

3.3 – Traceability Analysis

An important step in the validation process is the application of requirements management techniques, such as traceability information maintenance. By applying this technique, different kind of traceability information are maintained. In this thesis management system, the traceability information maintenance can be observed through the traceability table below:

	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	F R 37	F R 38	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																																																	
F.R.2	X																																																
F.R.3	X	X																																															
F.R.4	X	X																																															
F.R.5	X	X																																															
F.R.6	X																																																
F.R.7	X					X																																											
F.R.8	X																																																
F.R.9	X																																																
F.R.10	X																																																
F.R.11	X									X																																							
F.R.12	X										X																																						
F.R.13	X																																																
F.R.14	X												X																																				
F.R.15	X												X																																				
F.R.16	X												X																																				
F.R.17	X																																																
F.R.18	X																X																																
F.R.19	X																																																
F.R.20	X						X																																										
F.R.21	X						X																																										

	Depends on																																																
	FR 1	FR 2	FR 3	FR 4	FR 5	FR 6	FR 7	FR 8	FR 9	FR 10	FR 11	FR 12	FR 13	FR 14	FR 15	FR 16	FR 17	FR 18	FR 19	FR 20	FR 21	FR 22	FR 23	FR 24	FR 25	FR 26	FR 27	FR 28	FR 29	FR 30	FR 31	FR 32	FR 33	FR 34	FR 35	FR 36	FR 37	FR 38	NFR 1	NFR 2	NFR 3	NFR 4	NFR 5	NFR 6	NFR 7	NFR 8	NFR 9	NFR 10	NFR 11
F.R.22	X																							X																									
F.R.23	X																								X																								
F.R.24	X																																																
F.R.25	X																																																
F.R.26	X																																																
F.R.27	X																																																
F.R.28	X																																																
F.R.29	X																											X																					
F.R.30	X																																																
F.R.31	X																																																
F.R.32	X																																																
F.R.33	X																																																
F.R.34	X																													X																			
F.R.35	X																														X																		
F.R.36	X																																																
F.R.37	X																													X																			
F.R.38	X																														X																		
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																																																	

Finally, the final step in the validation process for this rental car service 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 at this stage, since this document is being produced before the design and implementation stages (not iteratively), it is only being vaguely defined which aspect of the requirements that can be tested. Later in the development process, these definitions will be translated into actual test cases.

ID: TC1
(FR1) End user, I should be able to log-in to the system
To be tested: Registering functionality - E-mail account provided validation - Password validation (to be later defined number and type of characters) - System database

ID: TC2
(FR2) Student should be able to create submissions
To be tested: Submission functionality - User interface - System database to store submitted documents

ID: TC3
(FR3) Student should be able to read submissions
To be tested: Submission functionality - User interface - System database to store submitted documents

ID: TC4
(FR4) Student should be able to update submissions
To be tested: Submission functionality - User interface - System database to store submitted documents

ID: TC5
(FR5) Student should be able to delete submissions
To be tested: Submission functionality - User interface - System database to store submitted documents

ID: TC6
(FR6) Student should be able to view a list of supervisors
To be tested: - User interface - Database functionality

ID: TC7
(FR7) Student should be able to suggest a supervisor
To be tested: Supervisor suggestion - Security and privacy validation - Database functionality

ID: TC8
(FR8) Student should be able to view feedbacks
To be tested: To have access to the feedback - User interface - Security and privacy validation - Database functionality

ID: TC9
(FR9) Student should be able to view grade
To be tested: To have access to the grade - User interface - Security and privacy validation - Database functionality

ID: TC10
(FR10) Coordinator should be able to view all users registered in the system
To be tested: Viewing users list - User interface - Database functionality

ID: TC11
(FR11) Coordinator should be able to create supervisors role
To be tested: Creating supervisors role - User interface - Database functionality

ID: TC12
(FR12) Coordinator should be able to view list of students assigned to each supervisor
To be tested: Viewing list of students assigned to each supervisor - User interface - Database functionality

ID: TC13
(FR13) Coordinator should be able to view all submissions
To be tested: Viewing all submissions - User interface - Database functionality

ID: TC14
(FR14) Coordinator should be able to evaluate the student’s project description submissions
To be tested: evaluate the student’s project description submissions - User interface - Database functionality

ID: TC15
(FR15) Coordinator should be able to evaluate the student’s report submissions
To be tested: evaluate the student’s the student’s report submissions - User interface - Database functionality

ID: TC16
(FR16) Coordinator should be able to evaluate the student’s report submissions
To be tested: evaluate the student’s the student’s report submissions - User interface - Database functionality

ID: TC17
(FR17) Coordinator should be able to to set deadlines
To be tested: set submission deadlines - User interface - Database functionality

ID: TC18
(FR18) Coordinator should be able to to update deadlines
To be tested: update submission deadlines - User interface - Database functionality

ID: TC19
(FR19) Coordinator should be able to submit the grade for a thesis project
To be tested: submit the grade for a thesis project - User interface - Database functionality

ID: TC20
(FR20) supervisor should be able to confirm supervision request
To be tested: supervisor should be able to confirm supervision request - User interface - Database functionality

ID: TC21
(FR21) supervisor should be able to decline supervision request
To be tested: supervisor should be able to decline supervision request - User interface - Database functionality

ID: TC22
(FR22) Supervisor should be able to assess the student’s project plan submissions
To be tested: Supervisor should be able to assess the student’s project plan submissions - User interface - Database functionality

ID: TC23
(FR23) Supervisor should be able to assess the student’s report submissions
To be tested: Supervisor should be able to assess the student’s report submissions - User interface - Database functionality

ID: TC24
(FR24) Supervisor should be able to view students project plan submissions
To be tested: Supervisor should be able to view students project plan submissions - User interface - Database functionality

ID: TC25
(FR25) Supervisor should be able to view students report submissions
To be tested: Supervisor should be able to view students project plan submissions - User interface - Database functionality

ID: TC26
(FR 26) Supervisor should be able bid for reports of their preferences
To be tested: bidding for reports of their preferences - User interface - Database functionality

ID: TC27
(FR 27) Supervisor should be able to feedback submissions
To be tested: give a feedback for submissions - User interface - Database functionality

ID: TC28
(FR 28) Opponent should be able to view assigned report
To be tested: viewing assigned report submissions - User interface - Database functionality

ID: TC29
(FR 29) Opponent should be able to give a feedback to an assigned report
To be tested: Giving a feedback to an assigned report - User interface - Database functionality

ID: TC30
(FR 30) As a coordinator, I should be able to assign a reader role to a user
To be tested: As a coordinator, I should be able to assign a reader role to a user - User interface - Database functionality

ID: TC31
(FR 31) As a coordinator, I should be able to assign a opponent role to a user
To be tested: As a coordinator, I should be able to assign a opponent role to a user - User interface - Database functionality

ID: TC32
(FR 32) As a coordinator, I should be able to read a users role
To be tested: As a coordinator, I should be able to read a users role - User interface - Database functionality

ID: TC33
(FR 33) As a coordinator, I should be able to update the supervisor role of a user
To be tested: As a coordinator, I should be able to update the supervisor role of a user - User interface - Database functionality

ID: TC34
(FR 34) As a coordinator, I should be able to update the reader role of a user
To be tested: As a coordinator, I should be able to update the reader role of a user - User interface - Database functionality

ID: TC35
(FR 35) As a coordinator, I should be able to update the opponent role of a user
To be tested: As a coordinator, I should be able to update the opponent role of a user - User interface - Database functionality

ID: TC36
(FR 36) As a coordinator, I should be able to delete the supervisor role of a user
To be tested: As a coordinator, I should be able to delete the supervisor role of a user - User interface - Database functionality

ID: TC37
(FR 37) As a coordinator, I should be able to delete the reader role of a user
To be tested: As a coordinator, I should be able to delete the reader role of a user - User interface - Database functionality

ID: TC38
(FR 38) As a coordinator, I should be able to delete the opponent role of a user
To be tested: As a coordinator, I should be able to delete the opponent role of a user - User interface - Database functionality

ID: TC39
(NFR 1) customers should be able to be familiarised themselves with the system within 5 minutes
To be tested: Usability - User interface

ID: TC40
(NFR 2) 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
To be tested: Safety - User interface - Security - Privacy - Database

ID: TC41
(NFR 3) the log-in function should not be affected by SQL-injection
To be tested: Security - Database - Security

ID: TC42
(NFR 4) the online service application should also have fast data transmission and response rates with average response time for submitting a document less than 10 seconds
To be tested: Performance - User interface - Security

ID: TC43
(NFR 5) the system should have 100% uptime during the duration of the course.
To be tested: Safety - User interface - Security

ID: TC44
(NFR 6) the system should comply with GDPR
- To be tested: Security - Integrity data system

ID: TC45
(NFR 7) the system should not depend on the customers operating system and should support the mainstream web browsers
To be tested: Compatibility

ID: TC46
(NFR 8) the system should be documented through UML diagrams, requirements document and design document
To be tested: Documentation

ID: TC47
(NFR 9) The system should answer at least 60 concurrent upload requests in less than 10 seconds.
To be tested: Performance

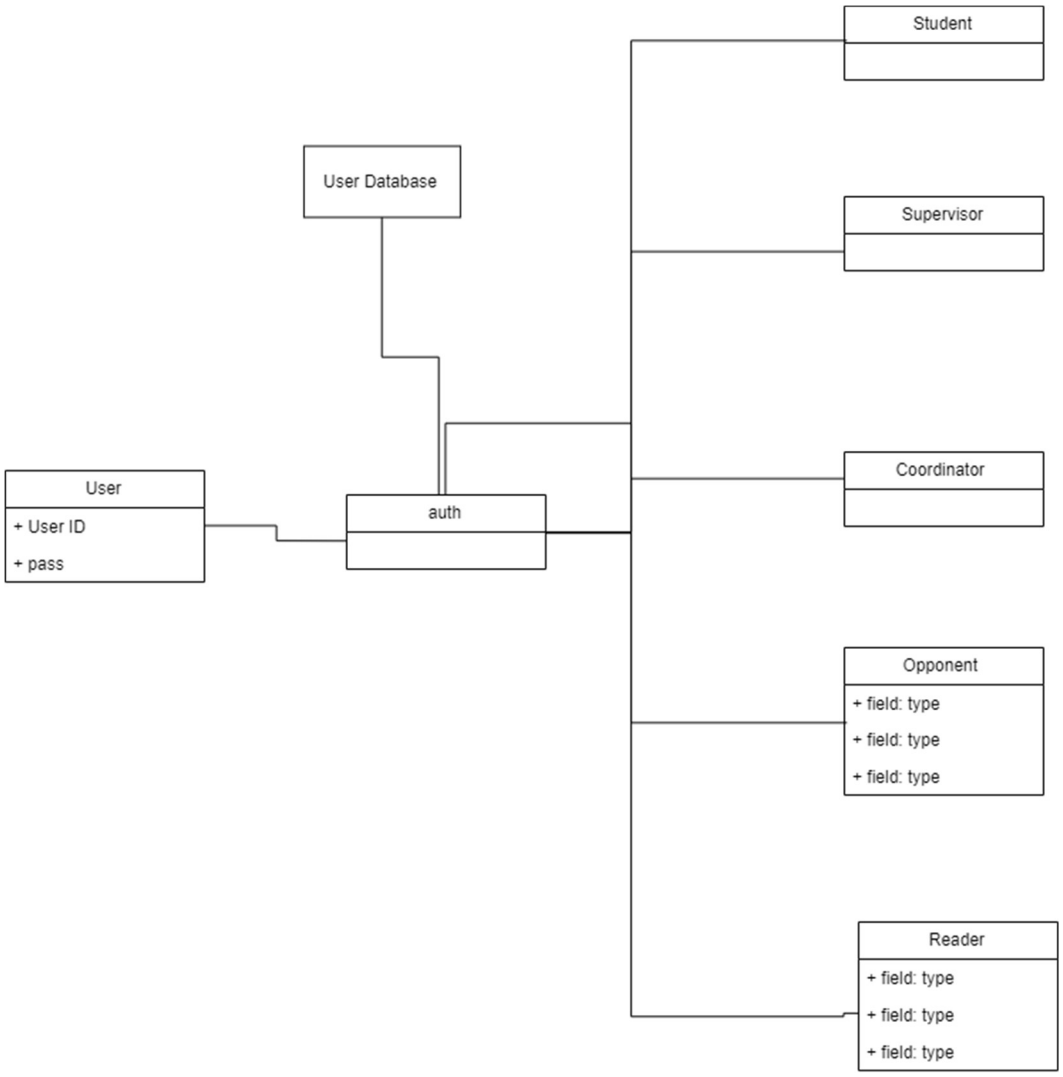
ID: TC48
(NFR 10) The user interface should be available both in Swedish and English
To be tested: User interface - Language spelling and grammar

ID: TC49
(NFR 11) The system should accept only.pdf submissions
To be tested: User interface - Resources

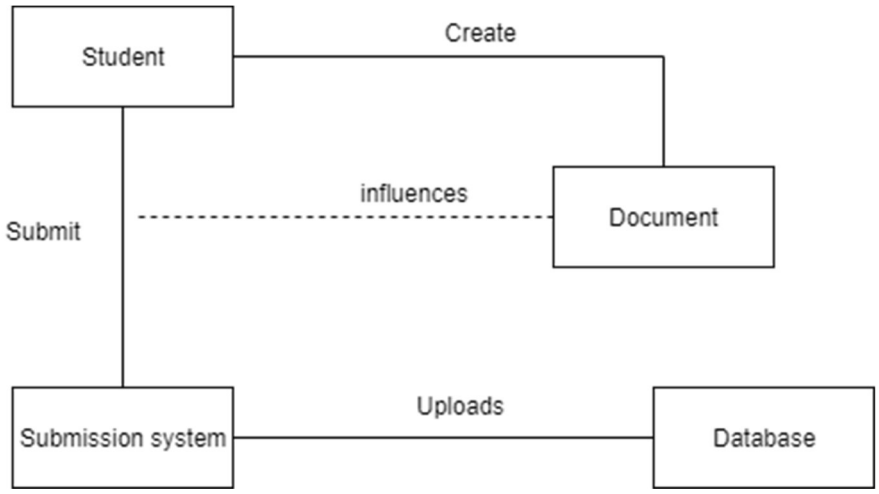
3.4 – Requirements Modelling

For this phase of the requirement engineering process, the following requirement was chosen to be modelled according to the UML Class Diagram model. Not all the requirements were modelled because some were added later.

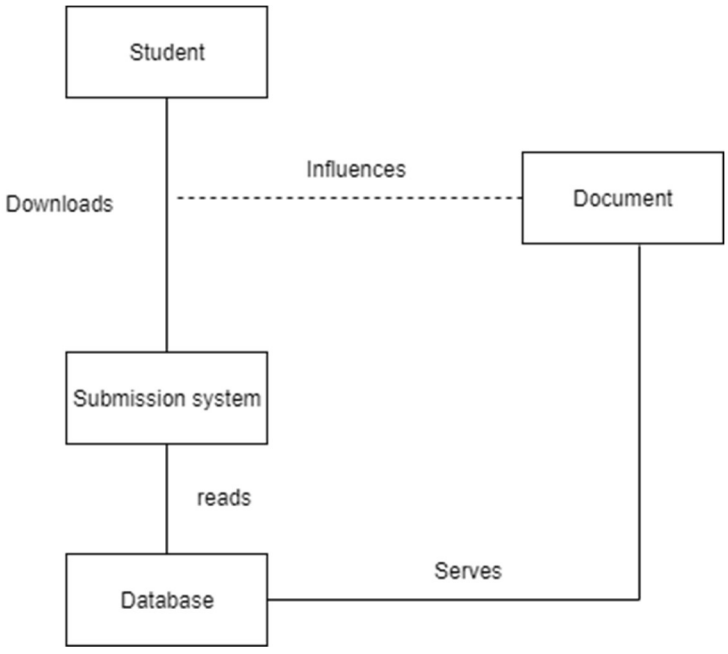
F.R.1- As an end user, I should be able to log-in the system and log-out of the system



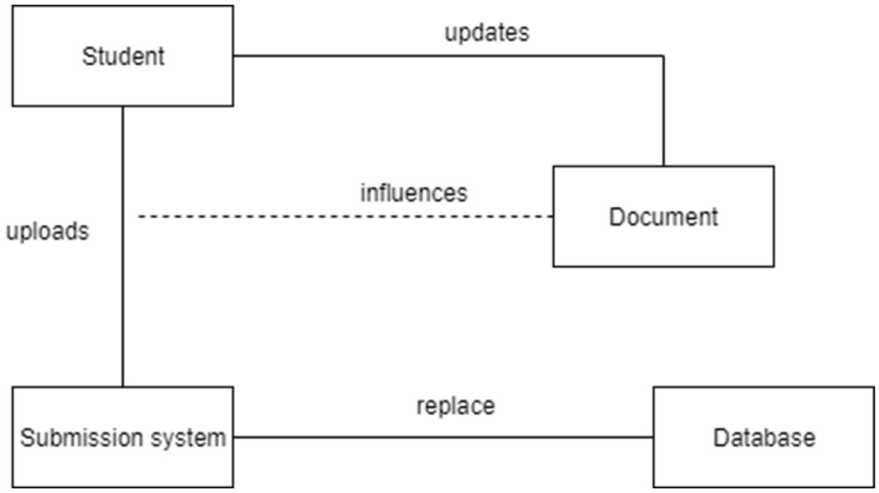
F.R.2- As a student, I should be able to create submissions



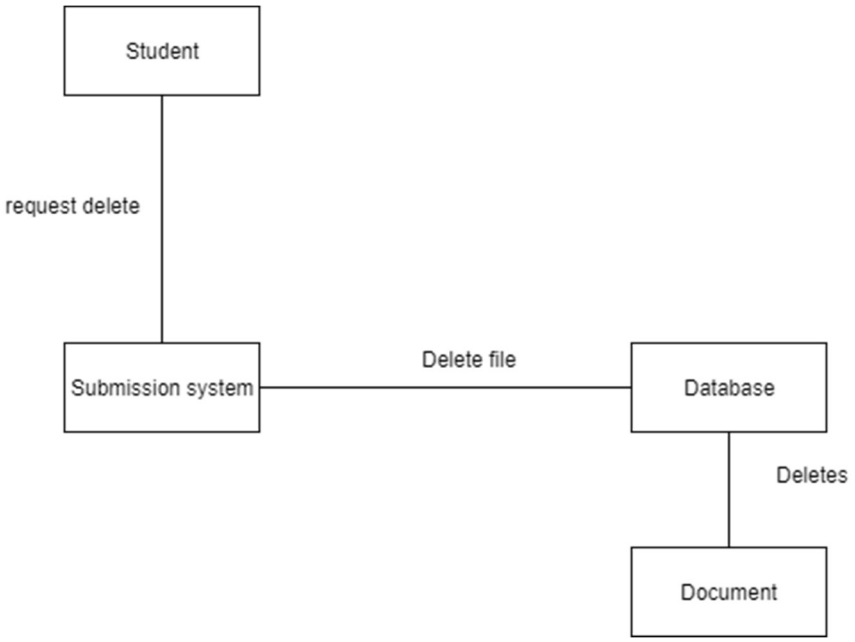
F.R.3- As a student, I should be able to read submissions



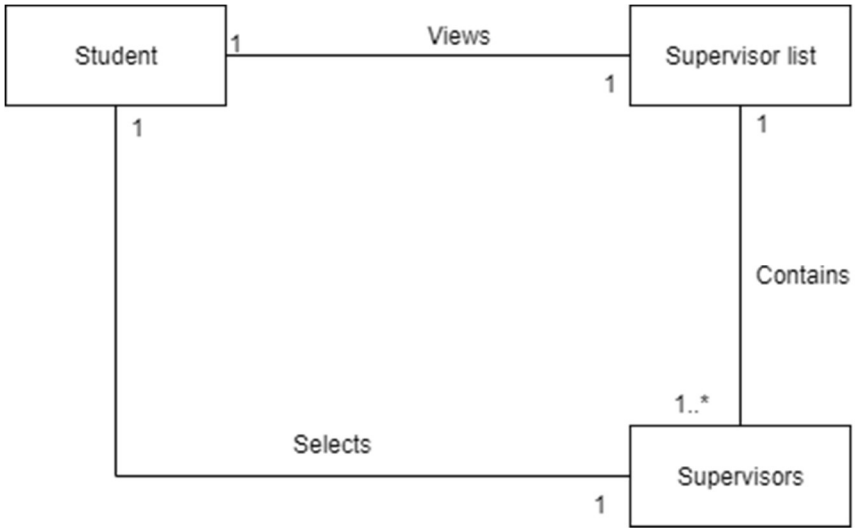
F.R.4- As a student, I should be able to update submissions



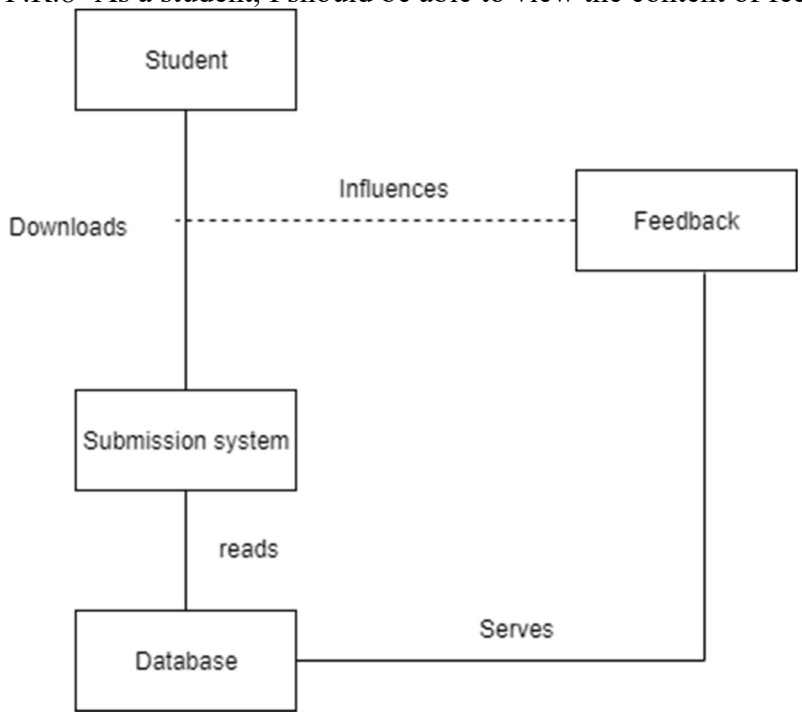
F.R.5- As a student, I should be able to delete submissions



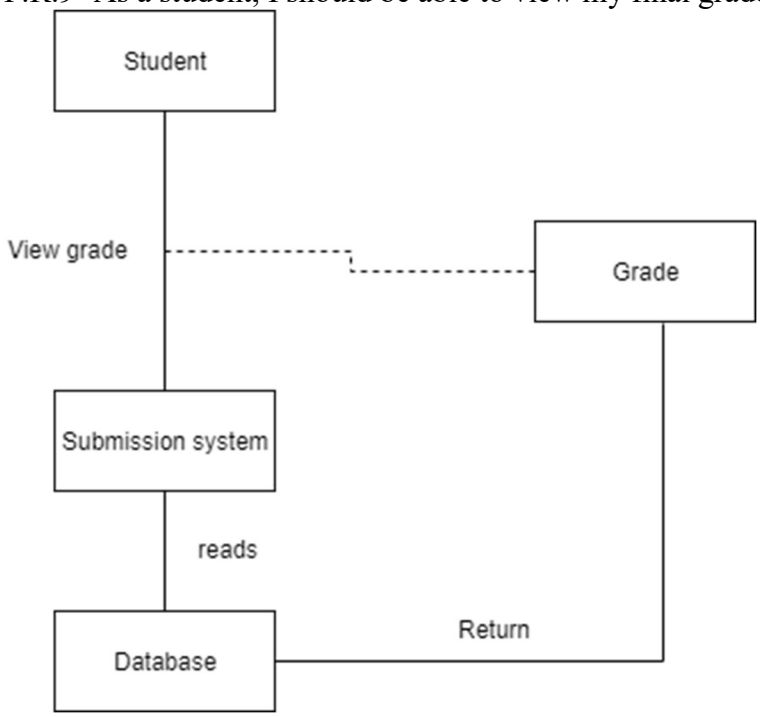
F.R.6- As a student, I should be able to view a list of supervisors



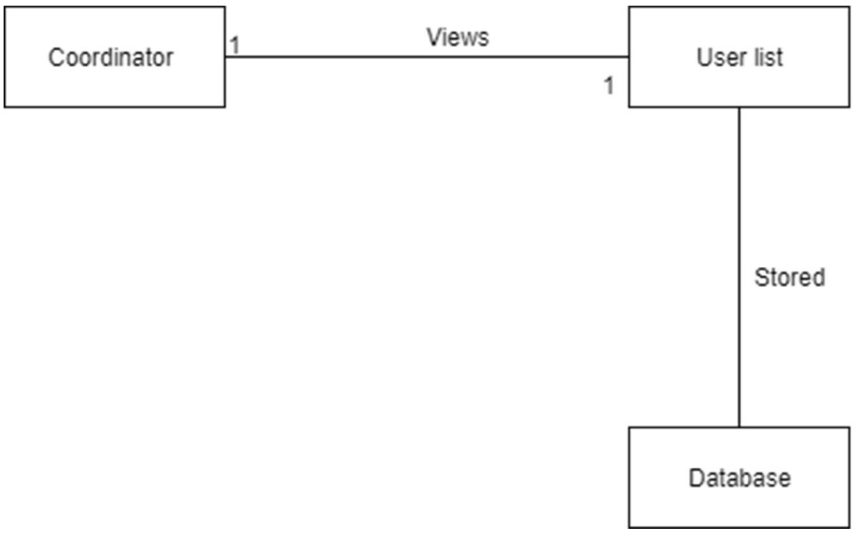
F.R.8- As a student, I should be able to view the content of feedback



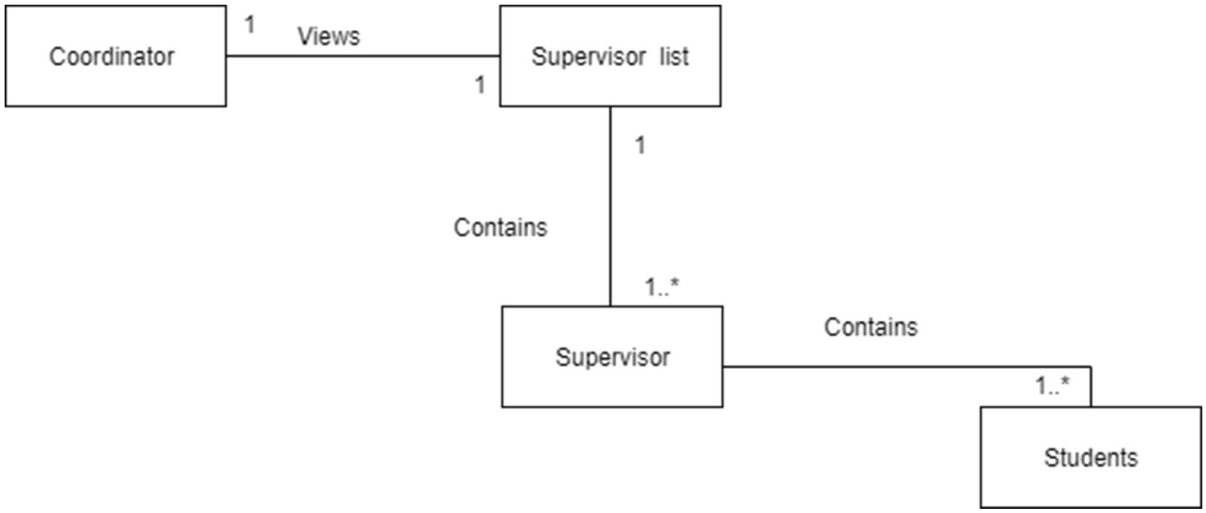
F.R.9- As a student, I should be able to view my final grade for the course



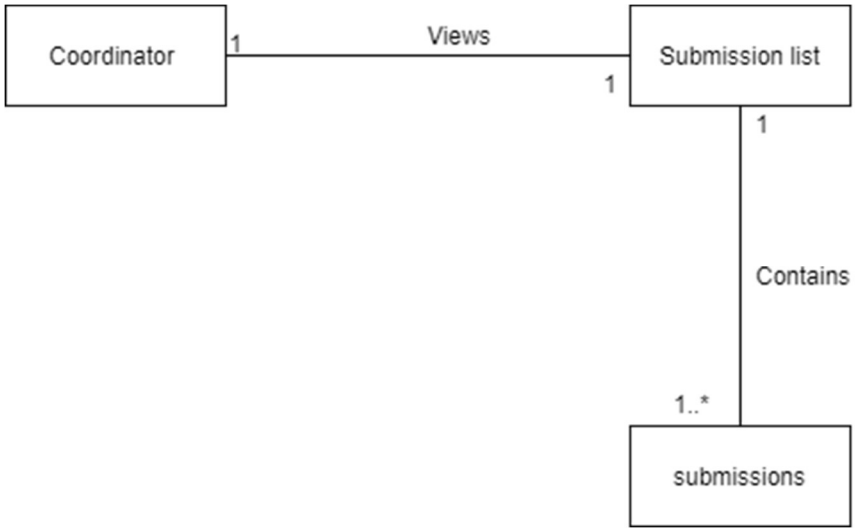
F.R.10- As a coordinator, I should be able to view a list of all users registered in the system



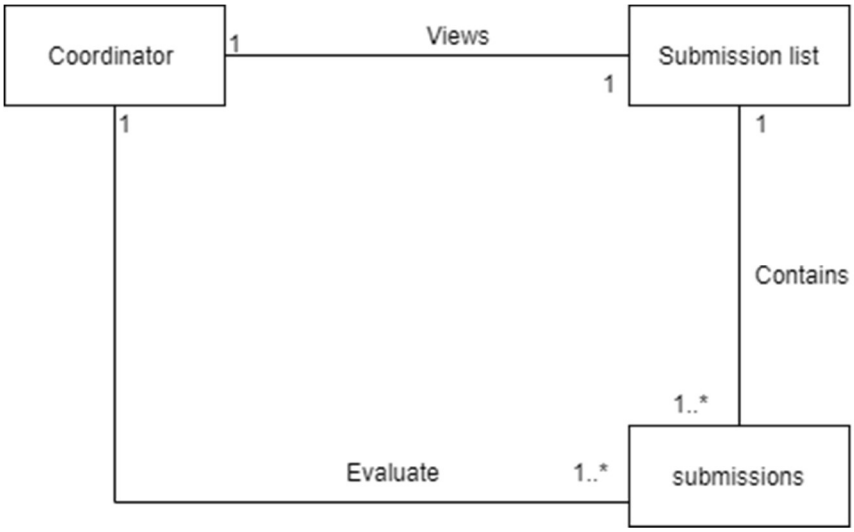
F.R.12- As a coordinator, I should be able to view list of students assigned to each supervisor



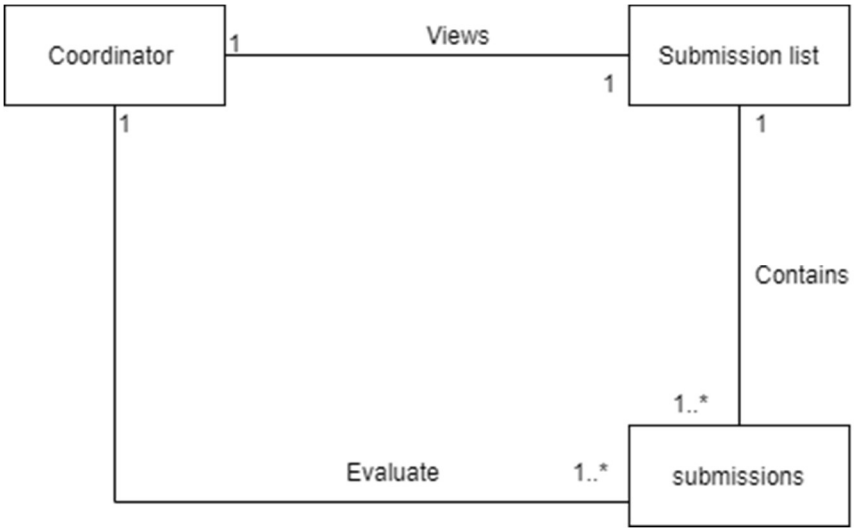
F.R.13- As a coordinator, I should be able to view all submissions



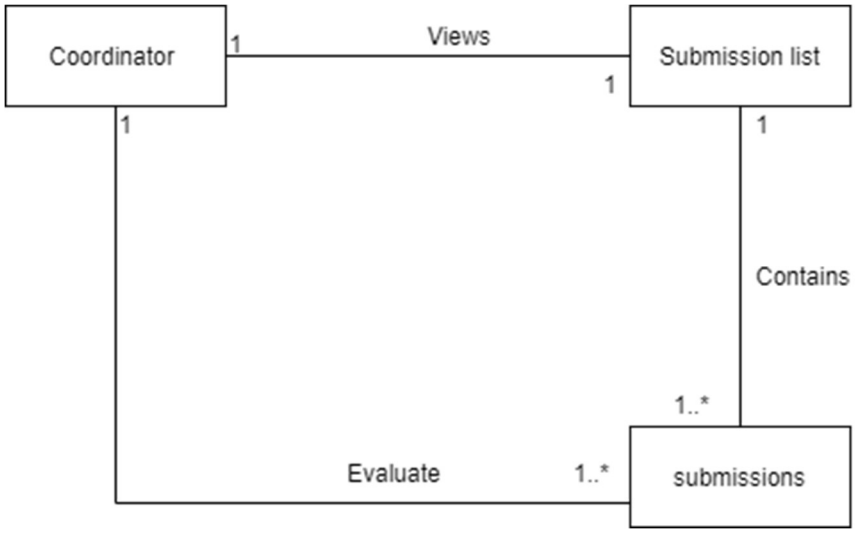
F.R.14- As a coordinator, I should be able to evaluate students' project description submissions



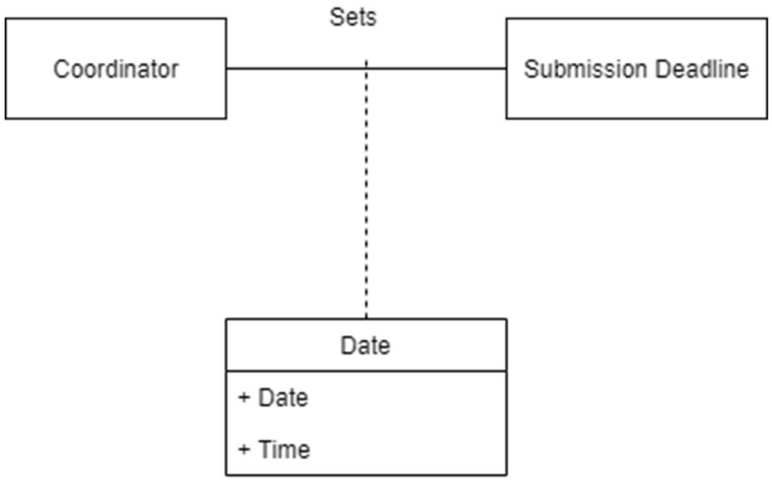
F.R.15- As a coordinator, I should be able to evaluate students report submissions



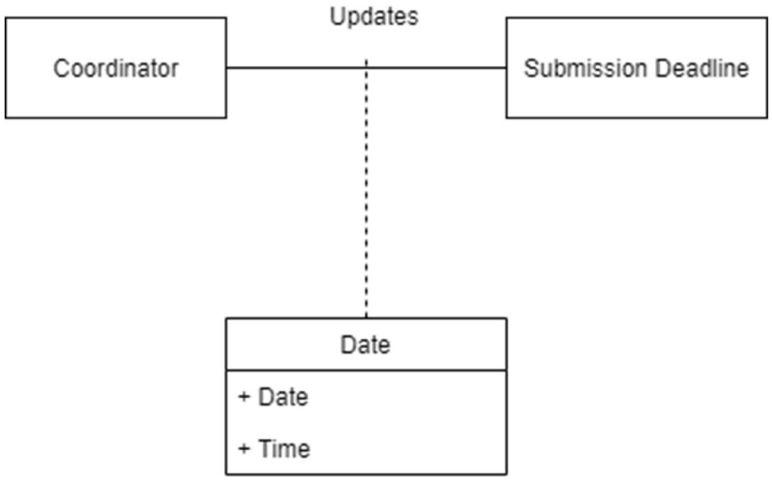
F.R.16- As a coordinator, I should be able to evaluate students final report submissions



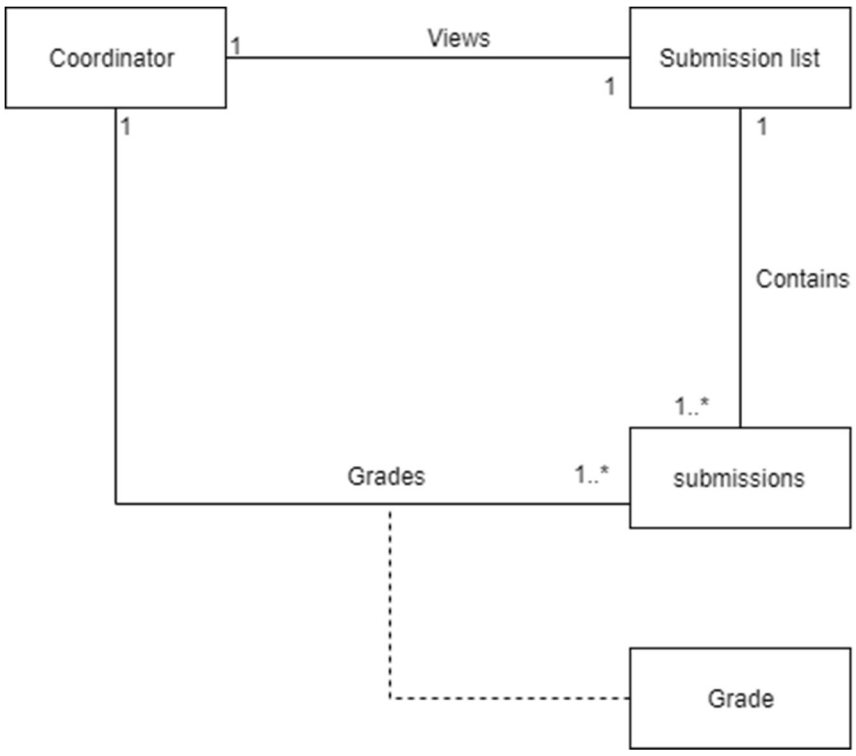
F.R.17- As a coordinator, I should be able to set deadlines



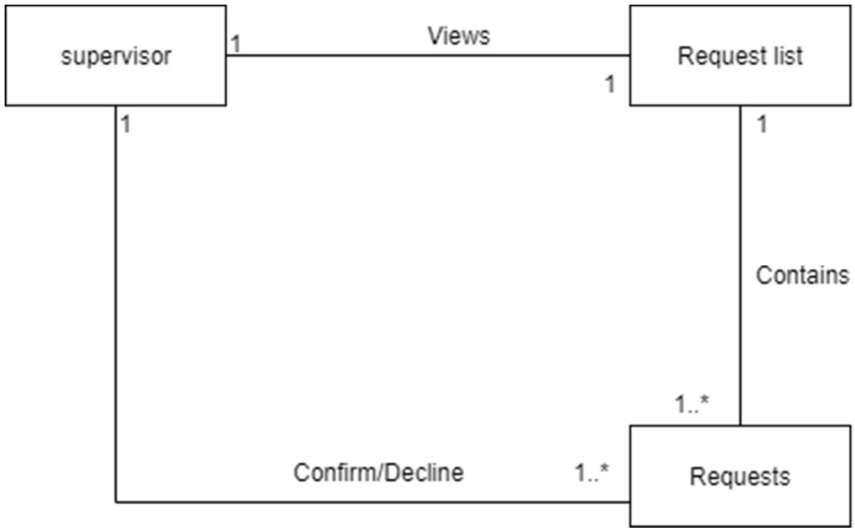
F.R.18- As a coordinator, I should be able to update deadlines



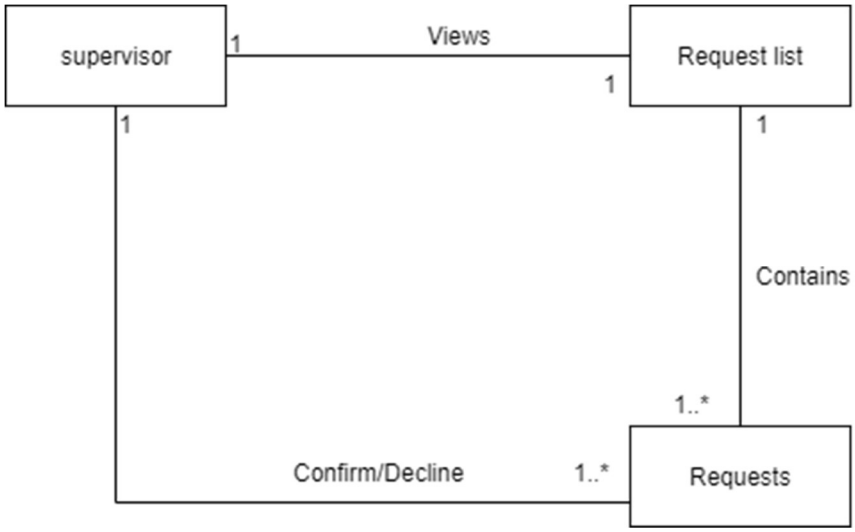
F.R.19- As a coordinator, I should be able to submit the grade for a thesis project



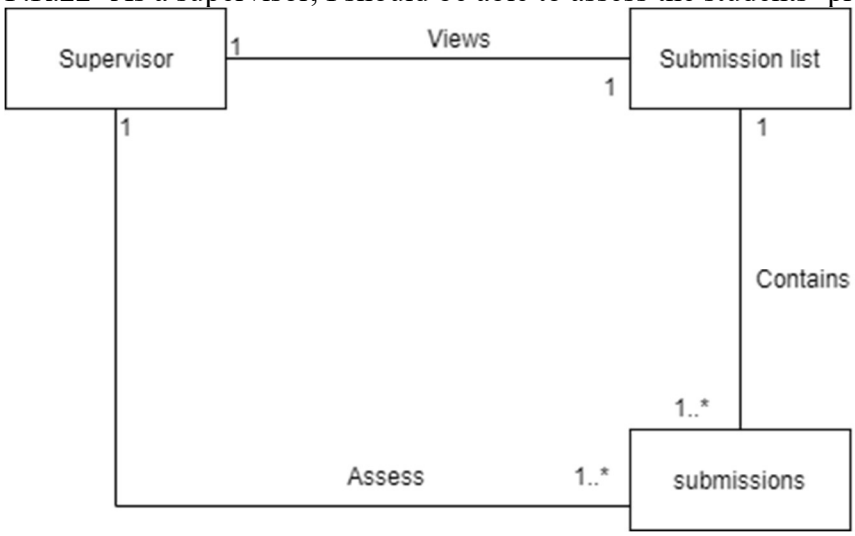
F.R.20- As a supervisor, I should be able to confirm supervision request



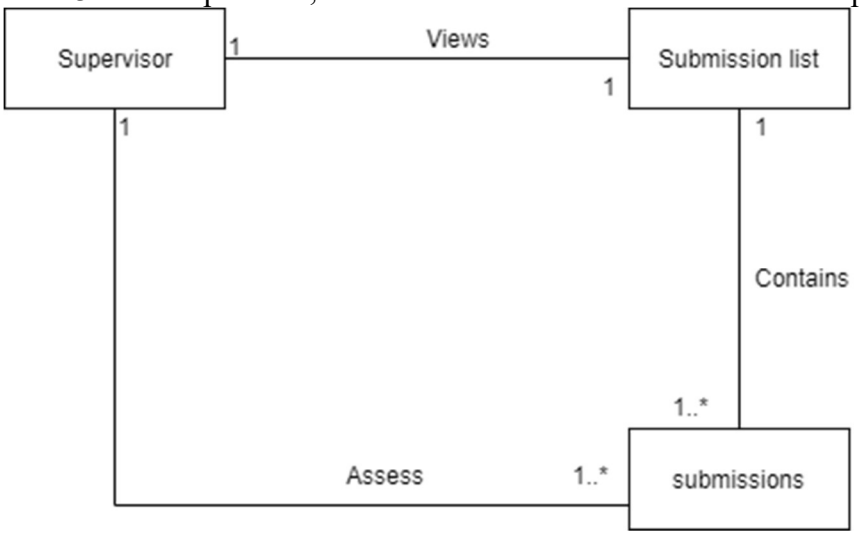
F.R.21- As a supervisor, I should be able to decline supervision request



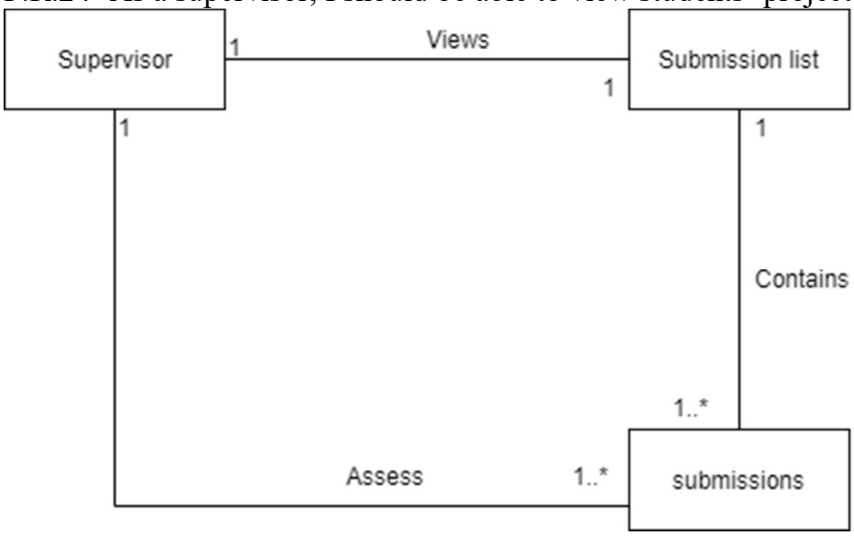
F.R.22- As a supervisor, I should be able to assess the students' project plan submissions



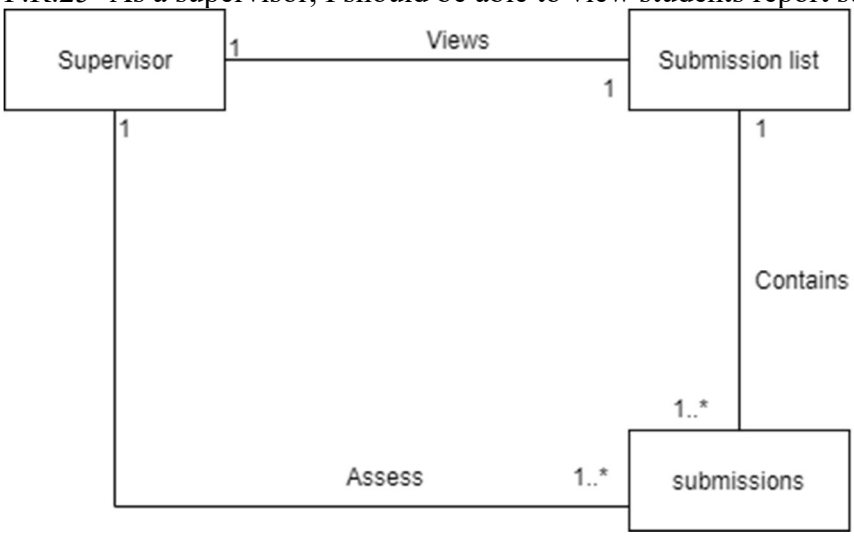
F.R.23- As a supervisor, I should be able to assess the students' report submissions



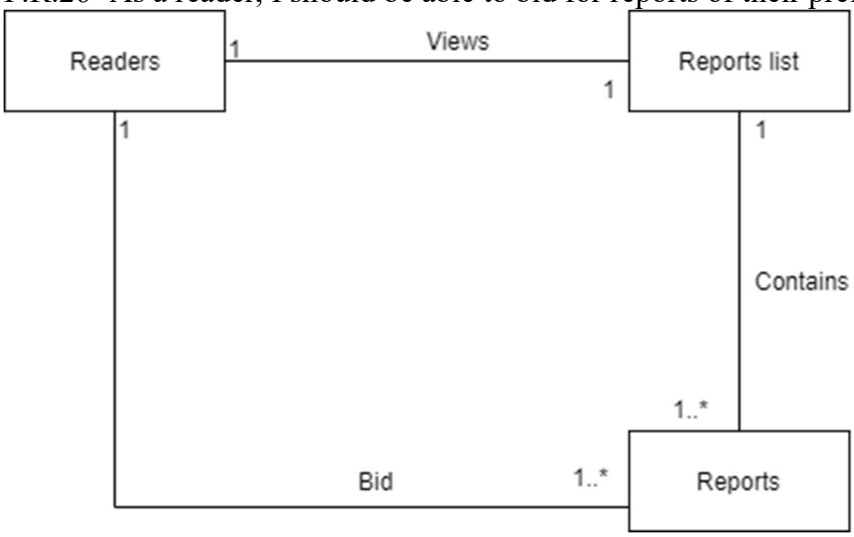
F.R.24- As a supervisor, I should be able to view students' project plan submissions



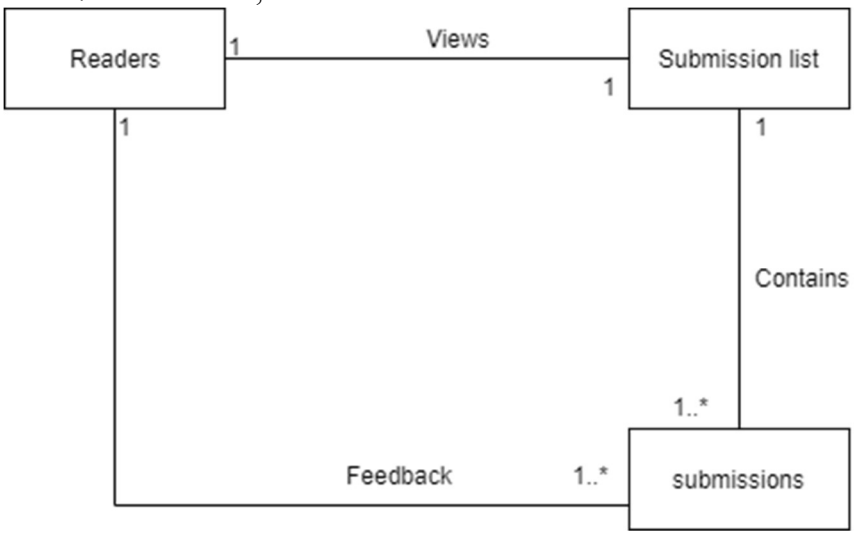
F.R.25- As a supervisor, I should be able to view students report submissions



F.R.26- As a reader, I should be able to bid for reports of their preferences



F.R.27- As a reader, I should be able to feedback submissions



4 – Appendices

4.1 – Old 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 registered in the system

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 – Old 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