Software Requirements Specification

MY UNIVERSITY

**Abdelrahman Mamdouh Ali 4141149**

**Mark Wafik Wiliam 4141127**

**Abdelrahman Mohamed 4141228**

**Mohamed Magdy Ali 4141166**

**Moatsem Salah 4141152**

2017

**Table of Contents**

1. [Introduction 1](#_bookmark0)
   1. [Purpose 1](#_bookmark1)
   2. [Scope 1](#_bookmark2)
   3. [Definitions, acronyms, and abbreviations 1](#_bookmark3)
   4. [References 2](#_bookmark4)
2. [Overall description 4](#_bookmark5)
   1. [Product perspective 4](#_bookmark6)
   2. [Product functions 4](#_bookmark7)
   3. [User characteristics 5](#_bookmark8)
   4. [Constraints 5](#_bookmark9)
   5. [Assumptions and dependencies 5](#_bookmark10)
   6. [Apportioning of requirements 6](#_bookmark11)
3. [Specific requirements 7](#_bookmark12)
   * 1. [User interfaces 7](#_bookmark13)
     2. [Hardware interfaces 8](#_bookmark14)
     3. [Software interfaces 8](#_bookmark15)
     4. [Communications interfaces 9](#_bookmark16)
   1. [Functional requirements 9](#_bookmark17)
      1. [User Class 1 - The User 9](#_bookmark18)
      2. [User Class 2 – Staff Member 14](#_bookmark19)
   2. [Performance requirements 21](#_bookmark20)
   3. [Design constraints 23](#_bookmark21)
   4. [Software system attributes 23](#_bookmark22)

# Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “My University" software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain the systems constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## Scope

The “My University” is a LMS mobile application which helps student stay connected with everything related to their university. Starting from their courses to their social media groups The application should be free to download from either a mobile phone application store or

similar services.

Teacher assistant can provides grades, assignments, materials using the web-portal and check for assignment plagiarism online between other student submissions and over the internet.

Students upload their assignments, view their grades and download their materials.

They can register their courses Online .and receive notification for every important announcement.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the mobile phone application |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| TA | someone that can post grades, assignments and materials for the student |
| Web-Portal | A web application which presents special facilities for university staff |

|  |  |
| --- | --- |
| CGPA | It’s the Cumulative GPA for all the courses that a student attended |
| Application Store | An installed application on mobile phone which helps users find new compatible applications with mobile phone platform and download them from the internet |
| Stakeholder | Any person who interacts with the system and is not a developer. |
| DESC | Description |
| RAT | Rational |
| DEP | Dependency |
| TAG | A unique, persistent identifier contained in a PLanguage statement [2] |
| GIST | A short, simple description of the concept contained in a PLanguage statement [2] |
| SCALE | The scale of measure used by the requirement contained in a PLanguage statement [2] |
| METER | The process or device used to establish location on a SCALE contained in a PLanguage statement [2] |
| MUST | The minimum level required to avoid failure contained in a PLanguage statement [2] |
| PLAN | The level at which good success can be claimed contained in a PLanguage statement [2] |
| WISH | A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement [2] |
| DEFINED | The official definition of a term contained in a PLanguage statement [2] |



## References

1. IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
2. Feldt R,”re\_lecture5b\_100914”, unpublished.
3. Davis M A, “Just Enough Requirements Management: Where Software Development Meets Marketing”, New York, Dorset House Publishing, 2005.
4. Karlsson J, “A Cost-Value Approach for Prioritizing Requirements”, Norges Teknisk- Naturvitenskapelige Uni. 1997
   1. **Overview**

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.

The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

# Overall description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each

type. At last, the constraints and assumptions for the system will be presented.

## Product perspective

This system will consist of two parts: one mobile application and one web portal. The mobile application will be used to make the student able to connect easily to their grades and their assignments while the web portal will be used for managing

The product is supposed to be an open source, under the GNU public license.

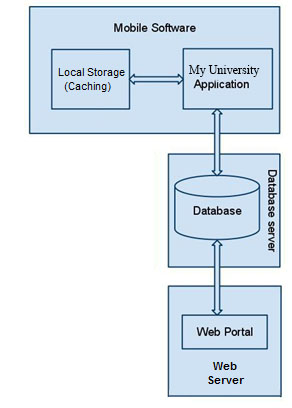
It is a cross platform mobile application with client-server model. The "My University"

will provide the proper methods to make all the possible communication methods for all students with their colleagues and the university staff.

## Product functions

The product depending on a Learning Management System with web portal for the university staff and mobile application for the students.

The staff will be able to set assignments, quizzes and grades they can send announcements to all students. Aside every assignment submitted from a student there will be two percentages for plagiarism; the first is a comparison between all the assignments that are submitted from all the students and the other compares the submitted assignments with the resources on the Internet. The percentage is a percentage of similarity and they can post public and anonymous surveys for the students. Students get an announcement for quizzes, assignments and grades. They can download materials and submit assignments only before deadlines. They can chat with each other and with the staff with private message but with permission from the receiver first. and as public message as a post. All the students are divided automatically by courses, groups, sections and individuals so every action on the system can be done according to these groups.



**Figure 1 - Block diagram**

## User characteristics

Our user characteristics is allegedly dynamic, built on permissions so every set of permission make a new user role but basically they come down to two roles.

Our users will be two types of users that interact with the system, users of the mobile application: students, and university staff. Each of these two types of users use the system differently so each of them has their own requirements.

The mobile application users (students) can use the application to check quizzes, assignments and grades. They can download materials and submit assignments only before deadlines. they can chat with each other and with the staff with private message but with permission from the receiver first, and announce public message as a post.

The staff will be able to set assignments, quizzes, grades, and send announcements to all students. Alongside every assignment submitted from a student there will be 2 percentages for plagiarism the first is a compare between all the assignments that is submitted from all the students and the other comparing the submitted assignments will the resources on the Internet the percentage is a percentage of similarity and they can post public and anonymous surveys for the students.

## Constraints

The system is constrained by the system of every university because different universities have different routines and students have different experiences so SYS will have be customized for universities.

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

Both the web portal and the mobile application will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefor increase the time it takes to fetch data.

## Assumptions and dependencies

One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the

application does not work as intended or even at all.

## Apportioning of requirements

In the case that the project is delayed, there are some requirements that could be transferred to the next version of the application. Those requirements are to be developed in the third release.

# Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

* 1. **External interface Requirements**

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

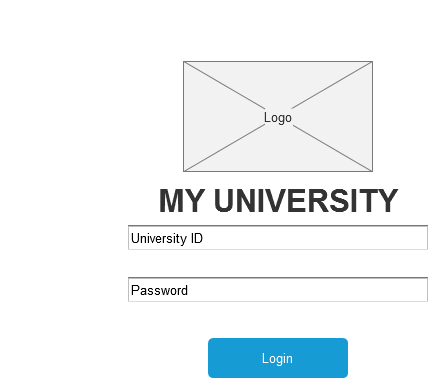
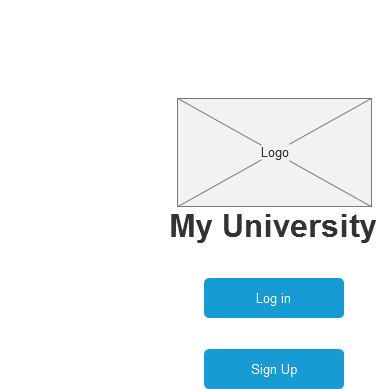
### User interfaces

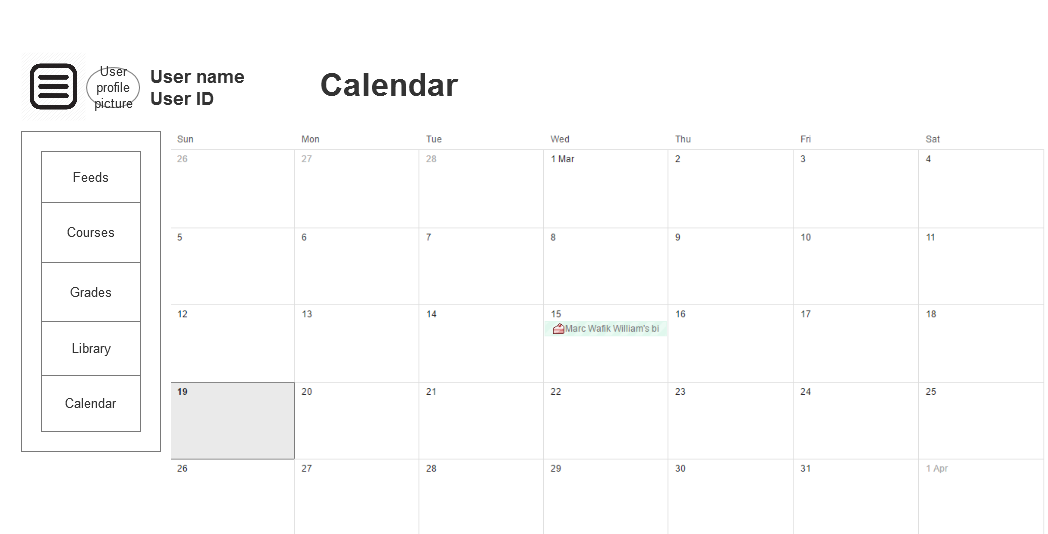
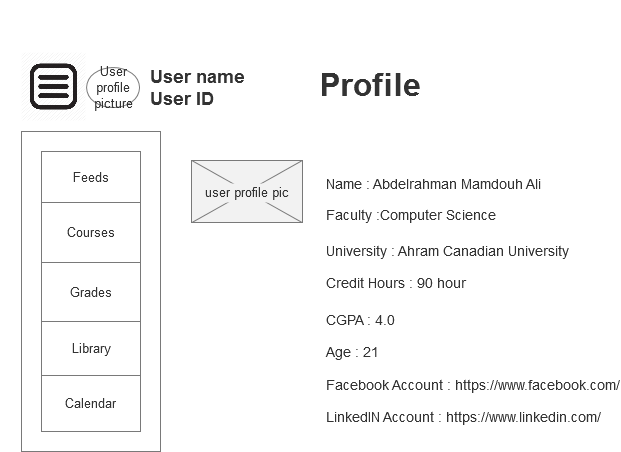
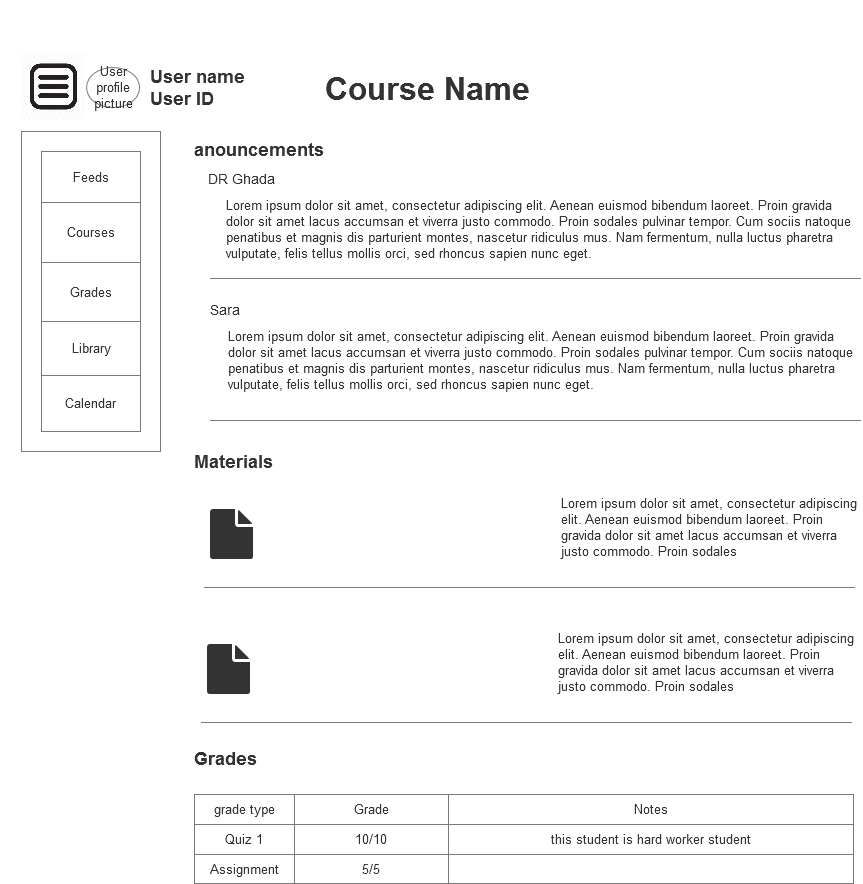
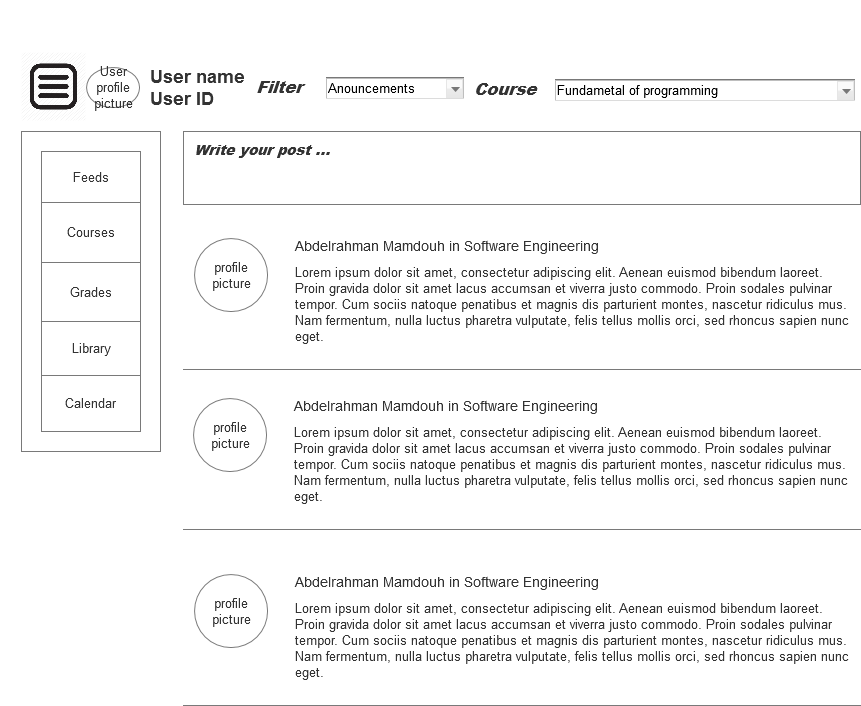
A first-time user of the mobile application should see the log-in page when he/she opens the application, If the user has not registered, he/she should be able to do that on the log-in page.

If the user is not a first-time user, he/she should be able to see the feeds page directly when the application is opened. Here the user get his all news feeds. And he writes the posts that he needs.

Every user should have a profile page where they can edit their e-mail address, phone number and password. Also, the user can set the mobile application.

Every course page the student will check the announcements, materials and grade and in the calendar he will see all his event and lectures





### Hardware interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

### Software interfaces

The mobile application communicates with the database in order to get the information about the students, see Figure 1. The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operations.

### Communications interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the web portal.

## Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

### User Class 1 - Student

* + - 1. ***Functional requirement 1.1***

**ID: FR1**

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.

RAT: In order for a user to download the mobile application. **DEP: None**

* + - 1. ***Functional requirement 1.2***

**ID: FR2**

TITLE: Download and notify users of new releases

DESC: When a new/updated version or release of the software is released, the user should check for these manually. The download of the new release should be done through the mobile phone in the same way as downloading the mobile application.

RAT: In order for a user to download a new/updated release**. DEP: FR1**

* + - 1. ***Functional requirement 1.3***

**ID: FR3**

TITLE: User registration - Mobile application

DESC: Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application. The user must provide university ID, password, Registration code and e- mail address . The user can choose to provide a regularly used phone number.

RAT: In order for a user to register on the mobile application. **DEP: FR1**

* + - 1. ***Functional requirement 1.4***

**ID: FR4**

TITLE: Mobile application (Save Information)

DESC: Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

RAT: In order for a user to register on the mobile application. **DEP: FR1, FR3**

* + - 1. ***Functional requirement 1.5***

**ID: FR5**

TITLE: Retrieve password

DESC: Given that a user has registered, then the user should be able to retrieve his/her password by e- mail.

RAT: In order for a user to retrieve his/her password. **DEP: FR1,FR3**

* + - 1. ***Functional requirement 1.6***

**ID: FR6**

TITLE: feeds

DESC: Given that a user I logged in to the mobile application, then the first page that is shown should be the feeds page. The student will get feeds for the post, announcements, grades , materials and surveys from different courses that he registered in and every post have a sender and a group and a date and files.

RAT: For a student to get the latest news of the university. **DEP: FR3, FR2, FR1**

* + - 1. ***Functional requirement 1.7***

**ID: FR7**

TITLE: comment on the posts in the feeds

DESC: every student can post reply’s on the posts in the university news feeds and can reply to the reply of the other users .

RAT: replying to the posts. **DEP: FR6, FR3**

* + - 1. Functional requirement 1.8

ID: FR8

TITLE: survey

DESC: student can vote on public surveys and private surveys : so we have to kind of surveys

1. public : the name of the voters will appear on the submissions
2. private : the submissions on the surveys will be private so there will not be a name on the submissions .

he survey will be a MCQ questions for all the courses at a time

RAT: student will submit survey .

**DEP: FR6,FR3**

* + - 1. ***Functional requirement 1.9***

**ID: FR9**

TITLE: Mobile application - polls

DESC: A user will vote on polls that will be uploaded by any staff member.

RAT: user will vote on polls that staff will make.

**DEP: FR3, FR8 FR1**

* + - 1. ***Functional requirement 1.10***

**ID: FR10**

TITLE: courses

DESC: every student have a list of courses every semester that he will enrolled in. every course will be divided into groups and every group have sections and every section have list of students and every course will have at least a Teacher assistant and a doctor and have a list of announcements and materials and grades.

RAT: student will enroll in courses. **DEP: FR7, FR8**

* + - 1. ***Functional requirement 1.11***

**ID: FR11**

TITLE: announcements

DESC: student will receive an announcement and every announcement should be related to specific course or from collage management .

RAT: student will get announcements from different courses. **DEP: FR7, FR8**

* + - 1. ***Functional requirement 1.12***

**ID: FR12**

TITLE: grades

DESC: A student will have a list of grades with every course divided to quizzes , assignment and different exams and every course will have this list with a total grade and every student have a CGPA for all the course that the student already attended

RAT: In order for a user to get his grades. **DEP: FR8,FR3,FR7**

* + - 1. ***Functional requirement 1.13***

**ID: FR13**

TITLE: view transcript

DESC: A student can view his own transcript with all his previous courses with CGPA

RAT: student can view is transcript

**DEP: FR3**

* + - 1. ***Functional requirement 1.14***

**ID: FR14**

TITLE: Registration

DESC: student can register in courses every semester with the consideration of the prerequisites of every course and the maximum numbers of students in every section and the major and the minor department of every student

RAT: In order for a student to register his courses. **DEP: FR3,FR2**

* + - 1. ***Functional requirement 1.15***

**ID: FR15**

TITLE: download bulk materials

DESC: A student can download material as a bulk for all the course. RAT: For a student to download all the materials.

**DEP: FR3**

* + - 1. ***Functional requirement 1.16***

**ID: FR16**

TITLE: Crud user profile data.

DESC: A student can add, delete or update his own profile (picture, name , LinkedIn link and .....) .

RAT: In order for a user to search by specific dish. **DEP: FR3**

* + - 1. ***Functional requirement 1.16***

**ID: FR16**

TITLE: search and filter the feed.

DESC: A student can search and use filtration in his courses feeds.

RAT: For a user to search by specific course.

**DEP: FR3, FR10**

* + 1. ***User Class 2 - University Staff***
       1. ***Functional requirement 2.1***

ID: FR1

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download. RAT: For a user to download the mobile application.

**DEP: None**

* + - 1. ***Functional requirement 2.2***

ID: FR2

TITLE: Download and notify users of new releases

DESC: When a new/updated version or release of the software is released, the user should check for these manually. The download of the new release should be done through the mobile phone in the same way as downloading the mobile application.

RAT: For a user to download a new/updated release. . **DEP: FR1**

* + - 1. ***Functional requirement 2.3***

ID: FR3

TITLE: User log-in

DESC: Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

RAT: For a user to register on the mobile application. **DEP: FR1**

* + - 1. ***Functional requirement 2.4***

ID: FR5

TITLE: Retrieve password

DESC: Given that a user has registered, then the user should be able to retrieve his/her password by e- mail.

RAT: For a user to retrieve his/her password. **DEP: FR3**

* + - 1. ***Functional requirement 2.5***

ID: FR6

TITLE: feeds

DESC: Given that a user is logged in to the mobile application, then the first page that is shown should be the feeds page. The student will get feeds for the post, announcements, grades, materials and surveys from different courses that he registered in and every post have a sender and a group and a date and files exactly like the students and the feeds will come from the courses that they are teaching.

RAT: For a staff to get the latest news of the university for the courses that he is teaching. **DEP: FR3**

* + - 1. ***Functional requirement 2.6***

ID: FR7

TITLE: comment on the posts in the feeds

DESC: every staff member can post reply on the posts that in the university news feeds and can reply to the reply of the other users.

RAT: replying to the posts. **DEP: FR3**

* + - 1. ***Functional requirement 2.7***

ID: FR8

TITLE: survey

DESC: staff can set questions and MCQ answers for every semester evaluation and decide when to start the survey and when to end it and all the surveys is private survey:

RAT: staff will set survey details. **DEP: FR3**

* + - 1. ***Functional requirement 2.8***

ID: FR9

TITLE: polls

DESC: A staff member will set polls at any time to check the opinion of the students for any matter. RAT: staff member will make polls for the students at any time.

DEP:, **FR3**

* + - 1. ***Functional requirement 2.9***

ID: FR10

TITLE: courses

DESC: every staff member has a list of courses every semester that they teach. every course will be divided into groups and every group have sections and every section have list of students and every course will have at least a one staff member and have a list of announcements and materials and grades that he can crud all of them at any time.

RAT: student will be enrolling in courses. **DEP: FR3**

* + - 1. ***Functional requirement 2.10***

ID: FR11

TITLE: announcements

DESC: staff member will send an announcement and every announcement should be related to specific course.

RAT: staff member will send announcements in different courses. **DEP: FR3,FR9**

* + - 1. ***Functional requirement 2.11***

ID: FR12

TITLE: grades

DESC: A staff member will set a list of grades with every course divided to quizzes, assignment and different exams and every course will have this list with a total grade that can be set or updated at any time.

RAT: staff member can set student grades or update it. **DEP: FR3,FR9**

* + - 1. ***Functional requirement 2.12***

ID: FR14

TITLE: Registration

DESC: staff member can register any student member in courses with no consideration to the usual rules

RAT: For a staff member can register student in courses **DEP: FR3**

* + - 1. ***Functional requirement 2.13***

ID: FR15

TITLE: download bulk materials

DESC: A student can download material as a bulk for all the course. RAT: For a student to download all the materials.

**DEP: FR9,FR3**

* + - 1. ***Functional requirement 2.14***

ID: FR16

TITLE: Crud user profile data.

DESC: A staff member can add, delete or update his own profile (picture, name, LinkedIn link etc) .

RAT: For a user to search by specific dish.

**DEP: FR3**

* + - 1. ***Functional requirement 2.14***

ID: FR16

TITLE: set courses prerequisite.

DESC: staff member can set all the prerequisite for every course.

RAT: For a course to have a prerequisite.

**DEP: FR9,FR3**

* + - 1. ***Functional requirement 2.15***

**ID: FR16**

TITLE: search and filter the feed.

DESC: A staff member can search and use filtration in his feeds.

RAT: For a user to search by specific course.

**DEP: FR3**

* 1. ***Performance requirements***

The requirements in this section provide a detailed specification of the user interaction with the software

and measurements placed on the system performance.

* + 1. ***Prominent search feature***

**ID: QR1**

TITLE: Prominent search feature

DESC: The search feature should be prominent and easy to find for the user. RAT: for a user to find the search feature easily.

DEP: none

* + 1. ***Usage of the search feature***

**ID: QR2**

TITLE: Usage of the search feature

DESC: The different search options should be evident, simple and easy to understand. RAT: for a user to perform a search easily.

DEP: none

* + 1. ***Usage of the result in the list view***

**ID: QR3**

TITLE: Usage of the result in the list view

DESC: The results displayed in the list view should be user friendly and easy to understand. Selecting an element in the result list should only take one click.

RAT: for a user to use the list view easily. DEP: none

* + 1. ***Usage of the result in the map view***

**ID: QR4**

TITLE: Usage of the result in the map view

DESC: The results displayed in the map view should be user friendly and easy to understand. Selecting a pin on the map should only take one click.

RAT: for a user to use the map view easily. DEP: none

* + 1. ***Usage of the information link***

**ID: QR5**

TITLE: Usage of the information link

DESC: The information link should be prominent and it should be evident that it is a usable link. Selecting the information link should only take one click.

RAT: for a user to use the information link easily. DEP: none

* + 1. ***Response time***

**ID: QR6**

TAG: Response Time

GIST: The fastness of the search

SCALE: The response time of a search

METER: Measurements obtained from 1000 searches during testing. MUST: No more than 2 seconds 100% of the time.

WISH: No more than 1 second 100% of the time.

* + 1. ***System dependability***

**ID: QR8**

TAG: SystemDependability

GIST: The fault tolerance of the system.

SCALE: If the system loses the connection to the Internet device or the system gets some strange input, the user should be informed.

METER: Measurements obtained from 1000 hours of usage during testing. MUST: 100% of the time.

## Design constraints

This section includes the design constraints on the software caused by the hardware.

* + 1. ***Hard drive space***

**ID: QR10**

TAG: HardDriveSpace GIST: Hard drive space.

SCALE: The application’s need of hard drive space. METER: MB.

MUST: No more than 20 MB. PLAN: No more than 15 MB. WISH: No more than 10 MB. MB: DEFINED: Megabyte

* + 1. ***Application memory usage***

**ID: QR11**

TAG: ApplicationMemoryUsage

GIST: The amount of Operate System memory occupied by the application. SCALE: MB.

METER: Observations done from the performance log during testing MUST: No more than 20 MB.

PLAN: No more than 16 MB WISH: No more than 10 MB

Operate System: DEFINED: The mobile Operate System which the application is running on. MB: DEFINED: Megabyte.

## Software system attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

* + 1. ***Reliability***

**ID: QR9**

TAG: SystemReliability

GIST: The reliability of the system.

SCALE: The reliability that the system gives the right result on a search. METER: Measurements obtained from 1000 searches during testing.

MUST: More than 98% of the searches. PLAN: More than 99% of the searches. WISH: 100% of the searches.

* + 1. ***Availability***

**ID: QR7**

TAG: SystemAvailability

GIST: The availability of the system when it is used.

SCALE: The average system availability (not considering network failing). METER: Measurements obtained from 1000 hours of usage during testing. MUST: More than 98% of the time.

PLAN: More than 99% of the time. WISH: 100% of the time.

**ID: QR22**

TITLE: Internet Connection

DESC: The application should be connected to the Internet.

RAT: For the application to communicate with the database. DEP: none

* + 1. ***Security***

**ID: QR12**

TAG: CommunicationSecurity

GIST: Security of the communication between the system and server.

SCALE: The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages.

METER: Attempts to get user-name and password through obtained messages on 1000 log-in session during testing.

MUST: 100% of the Communication Messages in the communication of a log-in session should be encrypted.

Communication Messages: Defined: Every exchanged of information between client and server.

**ID: QR13**

TAG: StaffMemberLoginAccountSecurity GIST: Security of accounts.

SCALE: If a staff member tries to log in to the web portal with a non-existing account then the staff member should not be logged in. The staff member should be notified about log-in failure. METER: 1000 attempts to log-in with a non-existing user account during testing.

MUST: 100% of the time.

**ID: QR15**

TAG: StaffMemberAccountSecurity GIST: Security of staff members accounts.

SCALE: A staff member and IP address should not be able to log-in for a certain time after three times of failed log-in attempts.

METER: 1000 attempts to log-in during the lock period after user account has been locked because of failed log-in attempts of three times.

MUST: The locking period should be half an hour, and during that period the log-in function is disabled.

**ID: QR17**

TAG: UserCreateAccountSecurity

GIST: The security of creating account for users of the system.

SCALE: If a user wants to create an account and the desired user name is occupied, the user should be asked to choose a different user name.

METER: Measurements obtained on 1000 hours of usage during testing. MUST: 100% of the time.

**ID: QR18**

TAG: StaffMemberCreateAccountSecurity

GIST: The security of creating account for staff members of the system.

SCALE: If a staff member wants to create an account and the desired user name is occupied, the staff member should be asked to choose a different user name.

METER: Measurements obtained on 1000 hours of usage during testing. MUST: 100% of the time.

* + 1. ***Maintainability***

**ID: QR19**

TITLE: Application extendibility

DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

RAT: For future functions to be implemented easily to the application. DEP: none

**ID: QR21**

TITLE: Application testability

DESC: Test environments should be built for the application to allow testing of the applications different functions.

RAT: To test the application. DEP: none

* + 1. ***Portability***

**ID: QR20**

TITLE: Application portability

DESC: The application should be portable with iOS and Android. RAT: The adaptable platform for the application to run on.

DEP: none

# Prioritization and Release Plan

To get a view of how to divide the requirements into different releases and what requirements should be included in which release, a prioritization of the requirements is needed.

This section discusses the choice of prioritization methods and gives a suggestion of how the release plan for these requirements could look like.

## Choice of prioritization method

When prioritizing the requirements, the ten most important ones were picked out first. This was done with a simple “1 to 10” ranking method, with one being “not important” and ten “very important”. Based on the elicitation meetings, and the perceived ideas of what was important to the different stakeholders, a number was set for each requirement. The numbers were then summed up for each requirement and the ten with the highest score were chosen to be prioritized with the cost value approach. The results, which are red-marked, can be seen in Appendix I and as shown, it turned out to be five functional requirements and five quality requirements. These requirements were then prioritized according to the cost value approach and the results can be viewed under Appendix II.

The remaining requirements were prioritized according to the “Five-Way Priority Scheme” as shown in Appendix III. This method was chosen since it gives the different stakeholders the same importance and has an enough wide range for determining which requirement is more important than the other [3].

However, in this prioritization process, the development team was not included as a stakeholder since the different features were not considered to be as important to them as for the other stakeholders.

Other methods for prioritization, such as the hundred-dollar test and the yes-no vote, were also considered. The hundred-dollar test is quite like the five-way priority scheme, since it also gives a wide range for ranking the requirements. However, it is more easily misused since someone could save all their money and put them on a requirement that they think is very important [3]. Others might not agree that this requirement is important but it might still get the most votes since one person cared about it [3].

The yes-no vote method might be fairly simple to carry out, however the range is too narrow. For instance, if two requirements are not very important it would be hard to determine which of those requirements that is more important than the other [3].

In conclusion, weighing the disadvantages and advantages of these methods against each other lead us to choose the five-way priority scheme.

## Release Plan

The requirements were divided into three releases based on the prioritization and their dependencies. The three different releases were assembled so that each would work as a fully functional application.

In the first release the requirements that build up the foundation of the application were included, together with the most highly prioritized requirements and their dependencies.

The second release also includes important requirements. However, these requirements are not vital for a functional application. They are more suited to act as additional features that can contribute to making the software product more attractive.

The third release includes the requirements that can be afforded to discard if the project gets delayed or overruns the budget.

For further details about the release plan, see Appendix IV.