Exam Grades Administration System

Software Requirements Specification

Version 1

Nov. 18, 2015

Zhenyu Kuang

Student No.: 2013212122

Requirements Analysis Engineer

Introduction to Software Engineering

Fall 2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 11/18/2015 | Version 1 | Zhenyu Kuang | First revision. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 2

1.5 Overview 2

2. General Description 3

2.1 Product Perspective 3

2.2 Product Functions 3

2.3 User Characteristics 3

2.4 General Constraints 3

2.5 Assumptions and Dependencies 4

3. Functional Requirements 4

3.1 Context Analysis 4

3.1.1 Context Diagram 4

3.1.2 Data flow level 1 4

3.1.3 Data flow level 2 5

3.2 Functional Requirements 2

3.2.1 Functional Structure 2

3.2.2 Functional Requirement #1: Log In to the EGAS 2

3.2.3 Functional Requirement #2: Select Courses and Classes 3

3.2.4 Functional Requirement #3: Type in Grades 4

3.2.5 Functional Requirement #4: Make a Publishing Request 4

3.2.6 Functional Requirement #5: Send a Publishing Request 5

3.2.7 Functional Requirement #6: Read a Publishing Request 5

3.2.8 Functional Requirement #7: Approve a PR 6

3.2.9 Functional Requirement #8: Reject a PR 6

3.2.10 Functional Requirement #9: View a List of Rejections 7

3.2.11 Functional Requirement #10: View the Reasons of a Rejection 7

3.2.12 Functional Requirement #11: Select Grades to Edit 8

3.2.13 Functional Requirement #12: Edit Grades 8

3.2.14 Functional Requirement #13: Select and View Grades 9

3.3 Use Cases 10

3.3.1 User Role 10

4. Non-Functional Requirements 11

4.1 Quality Requirements 11

4.1.1 Performance 11

4.1.2 Reliability 11

4.1.3 Availability 11

4.1.4 Security 11

4.1.5 Maintainability 11

4.2 Engineering Requirements 11

4.2.1 Design Constraints 11

4.2.2 Logical Database Requirements 11

5. Change Management Process 12

# Introduction

This section gives a brief introduction to the EGAS and the SRS itself, including the purpose of this document, scope of the product, a table of all the definitions, acronyms and abbreviations used in this SRS, references to all the documents that are referenced in this SRS, and a general overview of the overall organization of this document.

## 1.1 Purpose

This document is intended for analyzing and describing the requirements of the Exam Grades Administration System. It is written for all the stakeholders of the system to both design and implement the EGAS and also make use of the system.

## 1.2 Scope

The Exam Grades Administration System is a system designed to help users typing in, viewing, editing and searching for grades of the exams of students in a university. More specifically, the EGAS provides different functions and features for different types of users. Users of teacher type can type in new grades, edit the existing grades, send publishing requests to the educational administrators and view the reasons if there are requests that has been rejected by the EAs. Users of EA type can view the list of publishing requests and determine whether to approve or to reject the requests. Users of student type can log in to the EGAS to view the grades of their own, and can set different filters to view grades of a single course or to view grades of the passed/failed courses.

## 1.3 Definitions, Acronyms, and Abbreviations

Table 1: Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SRS | Software Requirements Specification. |
| EGAS | Exam Grades Administration System, the system which this SRS is describing. |
| Grade | Grade of an exam a student has got. |
| Student | A current student of the university who has a valid student license and an active account of the student type on the university’s UIS. |
| Teacher | A teacher of the university who has a valid teacher license and an active account of the teacher type on the university’s UIS. |
| EA | Educational Administrator. An educational administrator of the university who has a valid working license and an active account of the educational administrator type on the university’s UIS. |
| Portal | A web application which receives input from the user and leads the user to another page or web system which presents the desired information for the user. |
| UIS | University Information System, a web system running on an internal server of a university, which stores and presents information of the university, schools, courses, teachers and students and provides an login web portal for the sub-systems including the Library Management System, the Intranet System and the EGAS. |
| PR | Publishing Request. A request sent by a teacher to publish a set of grades typed in or edited. |
| Approve | An action that an EA performs after viewing the content of a PR and publishes the grades from the PR. |
| Reject | An action that an EA performs after viewing the content of a PR and does not allow to publish the grades. One or more reasons must be added to describe the problems of the PR. |
| Rejection | A record generated after an EA rejects a PR, containing the ID of the rejected PR and the reasons for the rejection. |
| SCR | Specification Change Request. A request to make changes to the SRS to correct errors or change requirements. |
| RCR | Requirement Change Request. A request to change, add or remove one or more requirements. |

## 1.4 References

[1] IEEE Software Engineering Standards Committee. IEEE SA 830-1998, IEEE Recommended Practice for Software Requirements Specifications. October 20, 1998.

[2] 王安生. 软件工程化. 北京: 清华大学出版社, 2014.

[3] Henry, N., Yockey, F., Solomon, J., Chen, P. and Duncan, B. Software Requirements Specification for Spinning Dragon Cruise Control. December 4, 2011.

## 1.5 Overview

The first section of this SRS gives a brief introduction to the EGAS and the SRS itself, including the purpose of this document, scope of the product, a table of all the definitions, acronyms and abbreviations used in this SRS, references to all the documents that are referenced in this SRS, and a general overview of the overall organization of this document.

The next section covers the general description of the EGAS, with a product perspective, functions of the product, analysis on the characteristics of the users, constraints, assumptions and dependencies of the EGAS product.

After that third and the fourth sections describe the specific and modeling requirements of the EGAS, both functional and non-functional. These two sections both analyze the requirements from multiple aspects.

The fifth section talks about the process of updating the system when the scope or requirements change. And the last section covers more additional and helpful information for the SRS that can be referred to when needed.

# General Description

This section gives an overview of the EGAS. The system will be explained in its context to display how the EGAS interacts with the other systems. Types of users for the EGAS and the different functions for each type of users will be described. At last, the constraints and assumptions for the EGAS will be presented.

## 2.1 Product Perspective

The EGAS usually forms part of a larger University Information System that runs on an internal server of a university. The UIS is a system which stores and presents the information of all the current students and teachers, and provides a login portal for the sub-systems including the Library Management System, the Intranet System and the EGAS.

## 2.2 Product Functions

A student user can log in with his student ID and password through the UIS of the university and then view his grades for different courses on the EGAS web portal. He can also search for the grade of a certain course, or view the grades of all passed/failed courses using a filter. The results will be viewed in a table providing the name, id and other information of the course and the teacher.

A teacher can log in using his teacher ID and password through the UIS. He can type in the grades of his own courses for the students using the EGAS web portal, and send a request to the educational administrators to publish the grades. When an error is found in the grades, the teacher can request the permission from the educational administrators to edit the grades.

An educational administrator can log in with his ID and password through the UIS and then view the unapproved requests for publishing grades or for the permission to edit the grades from the teachers. He can approve the requests, or refuse them with a reason given.

## 2.3 User Characteristics

There are mainly three types of users for the EGAS: students, teachers and educational administrators. All of the users are expected to have a certain valid license from the university, an account on the UIS and the basic knowledge of the interface and process of main-stream web services.

## 2.4 General Constraints

The EGAS must be under the same 2nd-level domain with the UIS to share cookies with the login portal of the UIS.

The EGAS must be deployed on the same server or a server in the same network with the database of UIS and grades to be able to connect to and interact with the database.

The web pages of the EGAS must be correctly rendered and work well on some out-of-dated operating systems like Windows XP and browsers like Internet Explorer 6.0 as there are lots of computers in the classrooms and laboratories that have not been upgraded for years. Students may using the EGAS quite often with these machines.

## 2.5 Assumptions and Dependencies

It is assumed that users of the EGAS have a valid license from the university and browse the EGAS website inside the Intranet or using methods like VPN to acquire the permission to access resources from the internal network of the university.

# Functional Requirements

This section contains the functional and quality requirements of the EGAS, with a detail description of the features provided by the system.

## 3.1 Context Analysis

### 3.1.1 Context Diagram

EGAS

Teachers

EA

Students

Picture 1: Context Diagram of the EGAS

Picture 1 shows the context of the EGAS. External entities of the EGAS consist of students, teachers and educational administrators. The students can fetch data from the EGAS, and the teachers and EAs can both obtain data from and write their data into the EGAS.

### 3.1.2 Data flow level 1

Picture 2 displays the basic internal functions and the flow of data of the EGAS. The external entities are the same as the context diagram of the EGAS.

The internal functions of the EGAS are concluded to six different parts. Teachers will type in the grades, send requests to the EAs for publishing the grades and edit the grades to fix some errors, etc. The EAs will review the requests sent by the teachers for both publishing new grades and committing changes to the existing grades. The student will mainly view their own grades.

6. Teachers

7. EA

8. Students

EGAS

1. Type in grades

5. Edit grades

4. View reject reasons

Grades

2. Request for publishing

3. Review requests

Publishing Request

Approved?

Reject Reason

N

Y

1. View grades

Picture 2: Level 1 Data Flow Diagram for the EGAS

Publishing requests, reject reasons and grades are the three major data types that are generated and used in the process of functions, which represents the status of the system.

### 3.1.3 Data flow level 2

Picture 2 shows the detailed internal functions and the flow of data of the EGAS. The internal functions of level 1 are elaborated in the level 2, with more detailed sub-functions added to each of the functions in level 1.

For the teachers, logging in is required as the first step to type in new grades for the students. They should also select the courses and classes of which the students’ grades will be typed in. Then they will do the typing work. After that, they should make a request to publishing the grades added. They will need to send the request to the EAs to acquire the permission to publish the grades. The teachers can also edit the existing grades by selecting the grades that need to be modified and editing them. After that, they should also make a request for publishing, as described above.

For the educational administrators, they can review the requests sent by the teacher via reading the requests and determining whether or not a request can be approved. If a request is approved, the grades will be added to the grades table for the students to view. If not, a reason should be added to the rejection for the teachers to know where was wrong. Teachers can view the reasons of rejections by viewing a list of rejections and selecting a single item to view its reason.

The students can view their own grades by first logging in to the EGAS, and selecting the grades they would like to view.

6. Teachers

7. EA

8. Students

EGAS

Grades

PR

Approved?

Reject Reason

N

Y

1. Type in grades

* 1. Log in to EGAS

1.2 Select course and class

1.3 Type in grades

2. Request for publishing

* 1. Make a request

2.2 Send request

3. Review requests

3.1 Read requests

3.2 Determine whether to approve a request

6. View grades

5. Edit grades

4. View reject reasons

4.1 View list of rejects

4.2 View a reason

5.1 Select grades to edit

5.2 Edit grades

6.1 Log in to EGAS

6.2 Select and view grades

Picture 3: Level 2 Data Flow Diagram for the EGAS

## 3.2 Functional Requirements

### 3.2.1 Functional Structure

EGAS

1. Type in grades

2. Request for publishing

3. Review requests

4. View reject reasons

5. Edit grades

6. View grades

1.1 Log in to EGAS

1.2 Select course and class

1.3 Type in grades

2.1 Make a request

2.2 Send request

3.1 Read Request

3.2 Determine whether to approve a request

4.1 View list of rejects

4.2 View a reason

5.1 Select grades to edit

5.2 Edit grades

6.1 Log in to EGAS

6.2 Select and view grades

System 1st Layer Functions 2nd Layer Functions

Picture 4: Functional Structure of the EGAS

### 3.2.2 Functional Requirement #1: Log In to the EGAS

3.2.2.1 Introduction

A user should be able to log in to the EGAS through a login portal of the outside UIS. The UIS shall store the information of the logged-in user into the cookies of his/her browser. The EGAS should be able to check whether a user has logged in and read his/her information out of the cookies. The EGAS must validate whether the information is provided by the UIS or fake data created by others.

3.2.2.2 Inputs

The identifier and basic information of the logged-in user should be stored inside the browser’s cookies for the EGAS to read as inputs.

3.2.2.3 Processing

The EGAS should check whether there is a user logged in by a validation value in the cookies, get his/her user ID, and validate the login state of the user by comparing the login token values from both the cookies and the server database. If the validation is ensured, the EGAS should continue to read the user’s basic information and use it to fetch data from the server.

3.2.2.4 Outputs

The desired web page and data should be shown if the login state has been validated.

3.2.2.5 Error Handling

If there is no login information in the cookies, the EGAS should redirect the browser to the login portal page provided by the UIS.

If the login information is fake or out-of-date, the EGAS should redirect the browser to the login portal page provided by the UIS.

### 3.2.3 Functional Requirement #2: Select Courses and Classes

3.2.3.1 Introduction

A user of teacher type should be able to view the list of his/her courses and classes. The user should be able to select one or more courses and classes to continue to type in the corresponding grades.

3.2.3.2 Inputs

The user should select one or more courses and classes from the list. The user’s ID and login token value should be added into the request parameters to send to the server.

3.2.3.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should fetch the list of students of the selected classes who have taken the exams of the selected courses, and allow the user to type in the grade of each student.

3.2.3.4 Outputs

A table of students who are in the selected classes and have taken exams of the selected courses should be shown, along with text fields for each student to allow the user to type in the grades.

3.2.3.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

### 3.2.4 Functional Requirement #3: Type in Grades

3.2.4.1 Introduction

A user of teacher type should be able to type in the grades of some students.

3.2.4.2 Inputs

The user should fill in all the text fields after the names of the students with the corresponding student’s grade for the course. The data of the table will be packaged with the user’s ID and login token value to send to the server.

3.2.4.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether grades of all the students have been typed in by the user. If the grades are all typed in, the EGAS should save the grades to a temporary grade table for the user to select when making a publishing request.

3.2.4.4 Outputs

A message will be shown to inform the user of the success.

3.2.4.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

If not all of the grades are filled in, the EGAS should be able to display the table with the user’s inputs and ask the user to filled in the missing fields.

### 3.2.5 Functional Requirement #4: Make a Publishing Request

3.2.5.1 Introduction

A user of teacher type should be able to make a request to publish the grades he/she typed in.

3.2.5.2 Inputs

The user should select a range of grades to be published and provide a short description of the grades to be published. The range of grades and description will be packaged with the user’s ID and login token value to send to the server.

3.2.5.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should save the description and the grades to be published into a database table.

3.2.5.4 Outputs

A message will be shown to inform the user of the success.

3.2.5.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

If the grades in the selected range have been published before, the EGAS should be able to alert the user and guide the user to select again.

### 3.2.6 Functional Requirement #5: Send a Publishing Request

3.2.6.1 Introduction

A user of teacher type should be able to send a publishing request to the EAs for reviewing.

3.2.6.2 Inputs

The user should click on the “Send” button on a publishing request page. The ID of the PR, the user’s ID and login token value will be sent to the server as the inputs.

3.2.6.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether or not the PR with the given ID exists. If the PR exists, the EGAS should add a “Sent” tag to the PR to inform the EAs to review it.

3.2.6.4 Outputs

A message will be shown to inform the user of the success.

3.2.6.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

If the PR with the given ID does not exist, the EGAS should alert the user of the error.

### 3.2.7 Functional Requirement #6: Read a Publishing Request

3.2.7.1 Introduction

A user of EA type should be able to view the description and the grades to be published of a PR sent by a teacher.

3.2.7.2 Inputs

The user should click on the “Read” button of a PR record in the list. The ID of the PR, the user’s ID and login token value will be sent to the server as the inputs.

3.2.7.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether or not the PR with the given ID exists. If the PR exists, the EGAS should fetch the content of the PR from the database and show it in the web page.

3.2.7.4 Outputs

A web page with the content of the PR inside.

3.2.7.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a EA, the EGAS should be able to alert the user and reject the user’s inputs.

If the PR with the given ID does not exist, the EGAS should alert the user of the error.

### 3.2.8 Functional Requirement #7: Approve a PR

3.2.8.1 Introduction

A user of EA type should be able to approve a PR sent by a teacher.

3.2.8.2 Inputs

The user should click on the “Approve” button on the detail page of a PR. The ID of the PR, the user’s ID and login token value will be sent to the server as the inputs.

3.2.8.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether or not the PR with the given ID exists. If the PR exists, the EGAS should add a “Approved” tag to the PR record in the database.

3.2.8.4 Outputs

A message will be shown to inform the user of the success.

3.2.8.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a EA, the EGAS should be able to alert the user and reject the user’s inputs.

If the PR with the given ID does not exist, the EGAS should alert the user of the error.

### 3.2.9 Functional Requirement #8: Reject a PR

3.2.9.1 Introduction

A user of EA type should be able to reject a PR sent by a teacher with reasons given.

3.2.9.2 Inputs

The user should click on the “Reject” button on the detail page of a PR, and type in the reasons for which the PR is rejected and click on the “Confirm” button. The ID of the PR, the reasons typed in, the user’s ID and login token value will be sent to the server as the inputs.

3.2.9.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether or not the PR with the given ID exists. If the PR exists, the EGAS should check whether the user has provided reasons for the rejection. If reasons are provide, the EGAS should add a “Rejected” tag to the PR record in the database to inform the teacher of the rejection.

3.2.9.4 Outputs

A message will be shown to inform the user of the success.

3.2.9.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a EA, the EGAS should be able to alert the user and reject the user’s inputs.

If the PR with the given ID does not exist, the EGAS should alert the user of the error.

If the user hasn’t provided some reasons for the rejection, the EGAS should alert the user of the error and ask the user to provide them.

### 3.2.10 Functional Requirement #9: View a List of Rejections

3.2.10.1 Introduction

A user of teacher type should be able to view a list of rejections of PRs sent by himself/herself.

3.2.10.2 Inputs

The user should click on the “Rejections” link on the home page of the EGAS. The user’s ID and login token value will be sent to the server as the inputs.

3.2.10.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should create a database query to fetch all the rejections of PRs sent by the user.

3.2.10.4 Outputs

A list of all the rejections of PRs sent by the user should be shown on the web page for the user to view. If there is no rejections for the user, a message should be shown to inform the user that there are no rejections.

3.2.10.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

### 3.2.11 Functional Requirement #10: View the Reasons of a Rejection

3.2.11.1 Introduction

A user of teacher type should be able to view the reasons of a rejection of a PR sent by himself/herself.

3.2.11.2 Inputs

The user should click on the “Detail” link of a rejection record in the list of rejections. The ID of the rejection, the user’s ID and login token value will be sent to the server as the inputs.

3.2.11.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether or not the rejection with the given ID exists. If the rejection exists, the EGAS should fetch the reasons of the rejection from the database.

3.2.11.4 Outputs

A web page with the reasons of the rejection should be shown.

3.2.11.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

If the rejection with the given ID does not exist, the EGAS should be able to inform the user of the error.

### 3.2.12 Functional Requirement #11: Select Grades to Edit

3.2.12.1 Introduction

A user of teacher type should be able to select a range of grades added by himself/herself to be edited.

3.2.12.2 Inputs

The user should select one or more grades from a list of grades. The range of grades, the user’s ID and login token value will be sent to the server as the inputs.

3.2.12.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether all of the selected grades are added by the user. If so, the EGAS should guide the user to the grades editing page.

3.2.12.4 Outputs

The user should be guided to the grades editing page to edit the grades the user has selected.

3.2.12.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

If any of the selected grades are not added by the user, the EGAS should inform the user of the error.

### 3.2.13 Functional Requirement #12: Edit Grades

3.2.13.1 Introduction

A user of teacher type should be able to edit the grades added by himself/herself listed in the table on the web page.

3.2.13.2 Inputs

The user should make the required changes to the grades listed in the table on the web page. All the text fields must be filled with valid grades. Data in the table will be packaged with the user’s ID and the login token value to be sent to the server as the inputs.

3.2.13.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should check whether grades of all the students have been typed in by the user. If the grades are all typed in, the EGAS should save the grades to a temporary grade table for the user to select when making a publishing request.

3.2.13.4 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a teacher, the EGAS should be able to alert the user and reject the user’s inputs.

If not all of the grades are filled in, the EGAS should be able to display the table with the user’s inputs and ask the user to filled in the missing fields.

### 3.2.14 Functional Requirement #13: Select and View Grades

3.2.14.1 Introduction

A user of student type should be able to select a range of grades of himself/herself and view the grades.

3.2.14.2 Inputs

The user can add one or more filters to the grades using the input fields on the web page, including grade of a specific course, grades of passed/failed courses and the way of sorting. The selected filters, the user’s ID and the login token value will be sent to the server as inputs.

3.2.14.3 Processing

The EGAS should check the validation of the login state of the user using the user ID and login token from the inputs. If the login state is valid, the EGAS should convert the filters to corresponding query conditions used to fetch data from the database. The EGAS should acquire the desired grades and display in the web page.

3.2.14.4 Outputs

Information of the courses and grades should be listed in a table and be displayed in the result web page.

If there’s no record of grades for the user, the EGAS should inform the user of that there’s no grades to display.

3.2.14.5 Error Handling

If the login state is invalid, the EGAS should alert the user of the exceptional login state and guide the user to the login portal page.

If the user is not a student, the EGAS should be able to alert the user and reject the user’s inputs.

## 3.3 Use Cases

### 3.3.1 User Role

Extends

Extends

Extends

Extends

1. Type in grades

1.1 Log in to EGAS

1.2 Select course and class

1.3 Type in grades

2. Request for publishing

3. Review requests

4. View reject reasons

5. Edit grades

6. View grades

2.1 Make a request

2.2 Send request

3.1 Read Request

3.2 Determine whether to approve a request

4.1 View list of rejects

4.2 View a reason

5.1 Select grades to edit

5.2 Edit grades

6.1 Log in to EGAS

6.2 Select and view grades



Extends

Extends

Teachers



EAs



Students

Uses

Uses

Uses

Uses

Picture 5: Use Cases Diagram between the User Roles and the System Functions of the EGAS

# Non-Functional Requirements

## 4.1 Quality Requirements

### 4.1.1 Performance

The system must be interactive and response to a huge amount of users’ requests in a short delay. In case of opening the EGAS web site and loading the static resources, the delay must be shorter than 2 seconds. In case of processing user inputs and performing database queries, the delay must be shorter than 2 seconds.

### 4.1.2 Reliability

The EGAS must make sure that it is reliable in its operations and for securing the sensitive details.

### 4.1.3 Availability

The EGAS must be highly available, especially in peak seasons like the days after exams. The available rate must be higher than 98%.

### 4.1.4 Security

The EGAS must protect the security of the data stored in the database, preventing it from both the deliberate modifications or destructions by others and the damages due to the aging hardware and natural causes e.g. disasters.

### 4.1.5 Maintainability

The EGAS should be easy to maintain, both for the software and for the hardware. The software should be modularized and loosely coupled to make it easy to add new features and enlarge the scale of the system. It should have few restrictions on the hardware platform so that the hardware can be easy to upgraded and replaced.

## 4.2 Engineering Requirements

### 4.2.1 Design Constraints

This section describes the design constraints on the EGAS from different aspects.

4.2.1.1 Software and Framework Constraints

The EGAS should be developed using Java and the SSH architecture (Spring, Struts and Hibernate). The web pages of the EGAS should be written in standard XHTML 1.1 to provide better compatibility for different operating systems and browsers.

4.2.1.2 Server Architecture Constraints

The EGAS should be running on a server with CentOS or other GNU/Linux systems and with an Apache Tomcat server software running.

### 4.2.2 Logical Database Requirements

The EGAS should use a MySQL database running on the same server with the EGAS or another server in the local network with the EGAS. The grades, publishing requests and rejection reasons should be stored in separate data tables the database. Other data including user information, courses, etc. should not be stored in the database of the EGAS but in the database of the external UIS.

# Change Management Process

When one or more errors are found in this SRS, changes should be submitted in a Specification Change Request to describe the content to be changed and the reasons for doing so. Analysis and discussion should be make over the SCR to determine whether to approve or to reject it.

If the SCR is approved, changes should be made to the requirements, and several Requirement Change Requests should be written by assigned persons to summarize the detailed information on the requirement change. The RCRs should be reviewed by a review committee consisting of members from all types of stakeholders.

If an RCR is approved by the review committee, the RCR will be used in the procedures of the project to make changes to the designing, coding and testing. If an RCR is rejected, it should be edited and sent back to the review committee again.

If all the RCRs are closed, the SCR will be acknowledged and summarized, and here comes the end of the change management process.