

---

# Software Requirements Specification

Version 1.0

May 9,2021

Application For Exam Results

Alaa AL-Othman

Kareen AL-soket

Rawan Ebrahim

Submitted in partial fulfillment  
Of the requirements of  
CS 310 Software Engineering

---

---

## Table of Contents

Table of Contents.....	1
List of Figure.....	1
1.0. Introduction.....	2
1.1. Purpose .....	2
1.2. Scope of Project.....	2
1.3. Glossary .....	3
1.4. References .....	3
1.5. Overview of Document .....	3
2.0. Overall Description.....	4
2.1 System Environment .....	4
2.2 Functional Requirements Specification .....	5
2.2.1 student Use Case .....	5
Use case: Login To The App .....	5
2.2.2 Students Affair Officer Use Case.....	6
Use case: Modify DataBase.....	6
2.3 User Characteristics .....	6
2.4 Non-Functional Requirements .....	7
3.0. Requirements Specification.....	8
3.1 External Interface Requirements .....	8
3.2 Functional Requirements .....	8
3.2.1 Login To The App.....	8
3.2.2 Modify DataBase .....	9
3.3 Detailed Non-Functional Requirements .....	10
3.3.1 Logical Structure of the Data .....	10
3.3.2 Security .....	11

## List of Figures

Figure 1 - System Environment .....	4
Figure 2 - Login To The App Use Case .....	5
Figure 3 - Modify Database Use Case .....	6
Figure 4 - Logical Structure of the Exam Results App Data.....	10

---

## **1.0. Introduction**

### ***1.1. Purpose***

The purpose of this document is to present a detailed description of the Exam Results App System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

### ***1.2. Scope of Project***

This software system will be a the Exam Results App for a students and a student affairs officer in the IT Engineering Faculty. This system will be designed to provide the possibility of automating the grading entry process for the student affairs officer , which would otherwise have to be performed manually and the ease of accessing the marks for the students. the system will meet the student 's needs while remaining easy to understand and use.

The system also contains a relational database containing a list of students and their grades.

### ***1.3. Glossary***

<b>Term</b>	<b>Definition</b>
Students	Person studying in it Engineering faculty.
student affairs officer	Person who enters marks in the database.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Students and student affairs officer
Database	Collection of all the information monitored by this system.

### ***1.4. References***

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

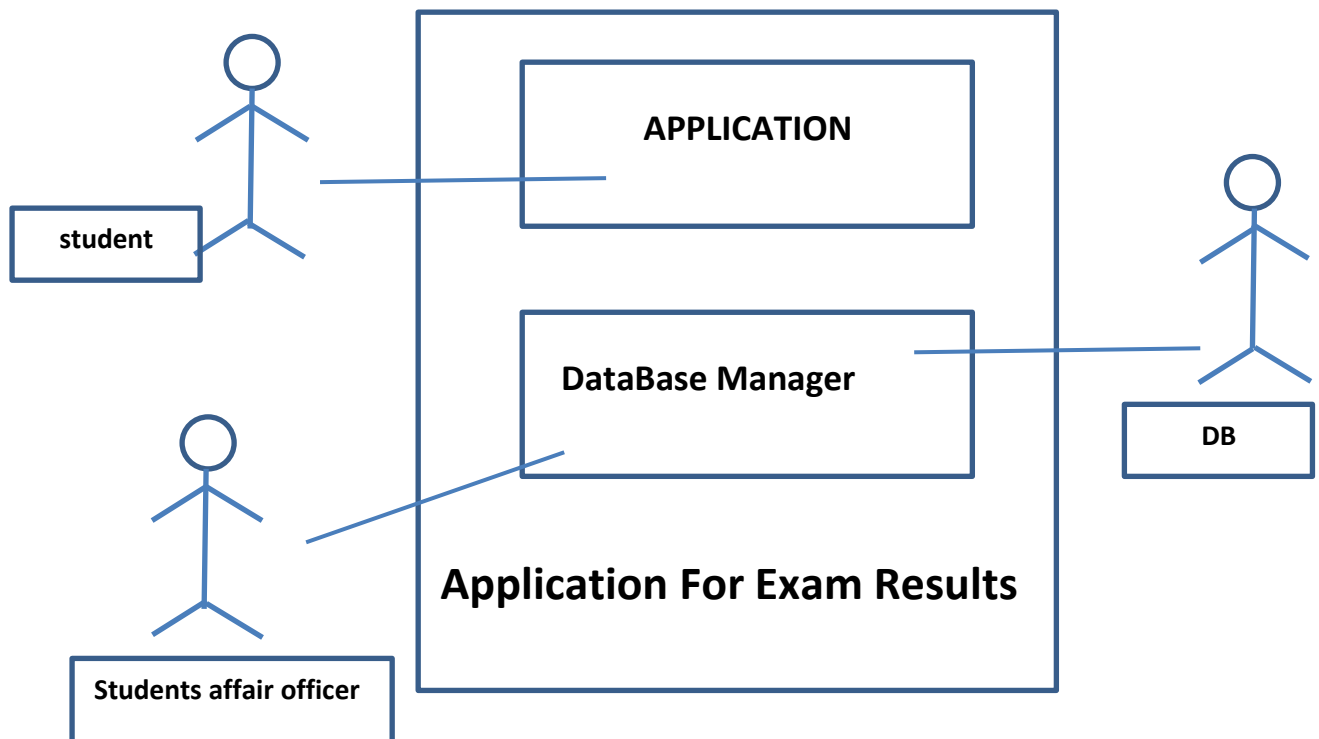
### ***1.5. Overview of Document***

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

## 2.0. Overall Description

### 2.1 System Environment



**Figure 1 - System Environment**

The Exam Results App System has two active actors and one cooperating system.

The Student accesses the app through the Internet. Students affair officer accesses the entire system directly.

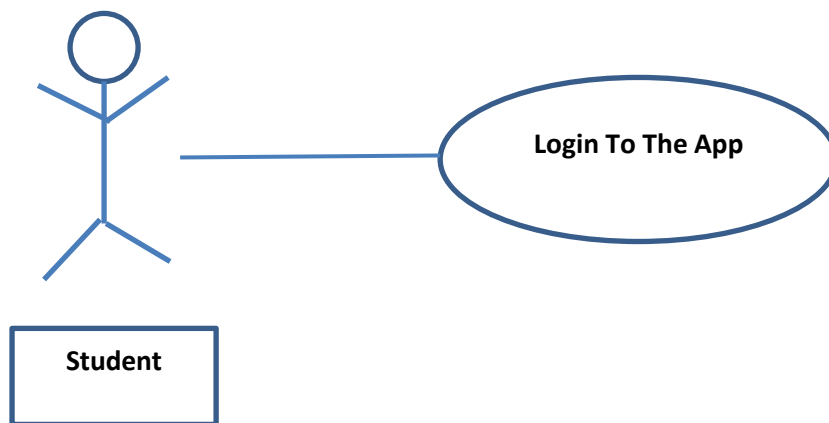
## ***2.2 Functional Requirements Specification***

This section outlines the use cases for each of the active user separately. The student has only one use case apiece and the student affair officer is main actor in this system.

### **2.2.1 Student Use Case**

Use case: **Login To The App**

**Diagram:**



#### **Brief Description**

The Student accesses the app, enters his information and sees his marks.

#### **Initial Step-By-Step Description**

Before this use case can be initiated, the Student has already accessed the Internet.

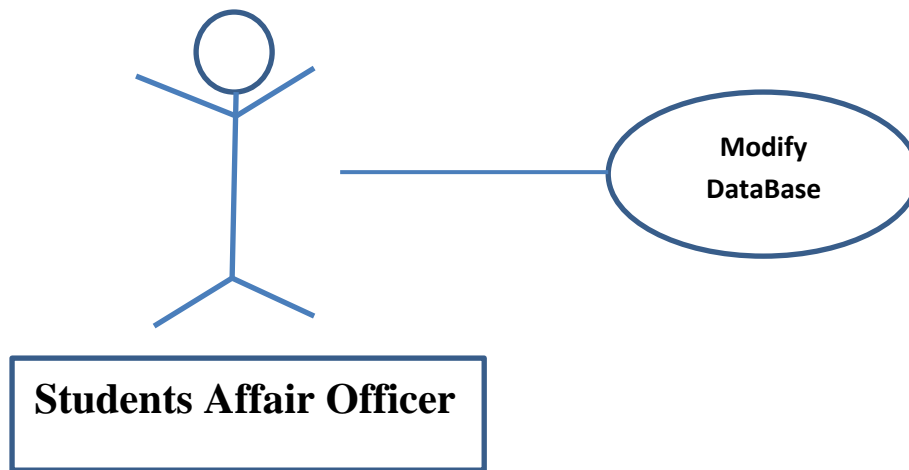
1. The Student enter his information in the app.
2. The Student select the year he want.
3. The system present his grades of the year he has selected.
4. The student can see his grades in any another year.

**Xref:** Section 3.2.1, Login To The App.

## 2.2.2 Students Affair Officer Use Case

Use case: Modify DataBase

**Diagram:**



### **Brief Description**

The Students Affair Officer accesses the DataBase, and modify students's marks.

### **Initial Step-By-Step Description**

Before this use case can be initiated, the Students Affair Officer has already connected to the DataBase.

1. The Students Affair Officer accesses to the database.
2. The Students Affair Officer add marks to the marks table in the database.
3. The Students Affair Officer submits the changes.

**Xref:** Section 3.2.2, Modify DataBase

## ***2.3 User Characteristics***

The Student is expected to be able to use the app and has to be a student in IT faculty.

The student affair officer has to be able to handle with sql phpMyAdmin database especially with enter a data.

---

## ***2.4 Non-Functional Requirements***

The database must be on a server with high speed internet capability.

There is a php page for connection between the app and the database.

The students affair officer's pc will contain an Access database. Access is already installed on this computer and is a Windows operating system.



### 3.0. Requirements Specification

#### 3.1 External Interface Requirements

The only link to an external system is the link to the students's marks database query about the marks of students.

The students's marks Database tables are students, lectures and marks.

The students table fields are student id, student name, student year and his password.

The lectures table fields are lecture id, lecture name and lecture year.

The marks table fields are student id, lecture id and his mark in this lecture.

The *login to the app* use case sends the student name, his id and the password that he entered in the interface's fields.

The *modify database* use case add record to the database tables.

#### 3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

##### 3.2.1 Login To The App

<b>Use Case Name</b>	Login To The App
<b>XRef</b>	Section 2.2.1, Login To The App
<b>Trigger</b>	The student access to the application login screen.
<b>Precondition</b>	The App displays input fields for student information at the login screen.
<b>Basic Path</b>	1. Student enters his name , id and password and clicks login button.

	2. The app displays different tabs about his study years. 3. Each tab displays the student marks in year's lectures and his average.
<b>Alternative Paths</b>	None.
<b>Postcondition</b>	The requested marks have been displayed.
<b>Exception Paths</b>	The Student may abandon the login screen at any time
<b>Other</b>	None.

### 3.2.2 Modify DataBase

<b>Use Case Name</b>	Modify DataBase
<b>XRef</b>	Section 2.2.2, Modify DataBase
<b>Trigger</b>	The Students Affair Officer accesses the Database, and modify students' marks.
<b>Precondition</b>	the students done new exam.
<b>Basic Path</b>	1. The Students affair officer connects to the Database. 2. He adds the new marks to DB.
<b>Alternative Paths</b>	None.
<b>Postcondition</b>	Close the connect with the Database
<b>Exception Paths</b>	The Students Affair Officer shouldn't abandon the Database without finish his work .
<b>Other</b>	None.

### 3.3 Detailed Non-Functional Requirements

#### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the student's marks database is given below.

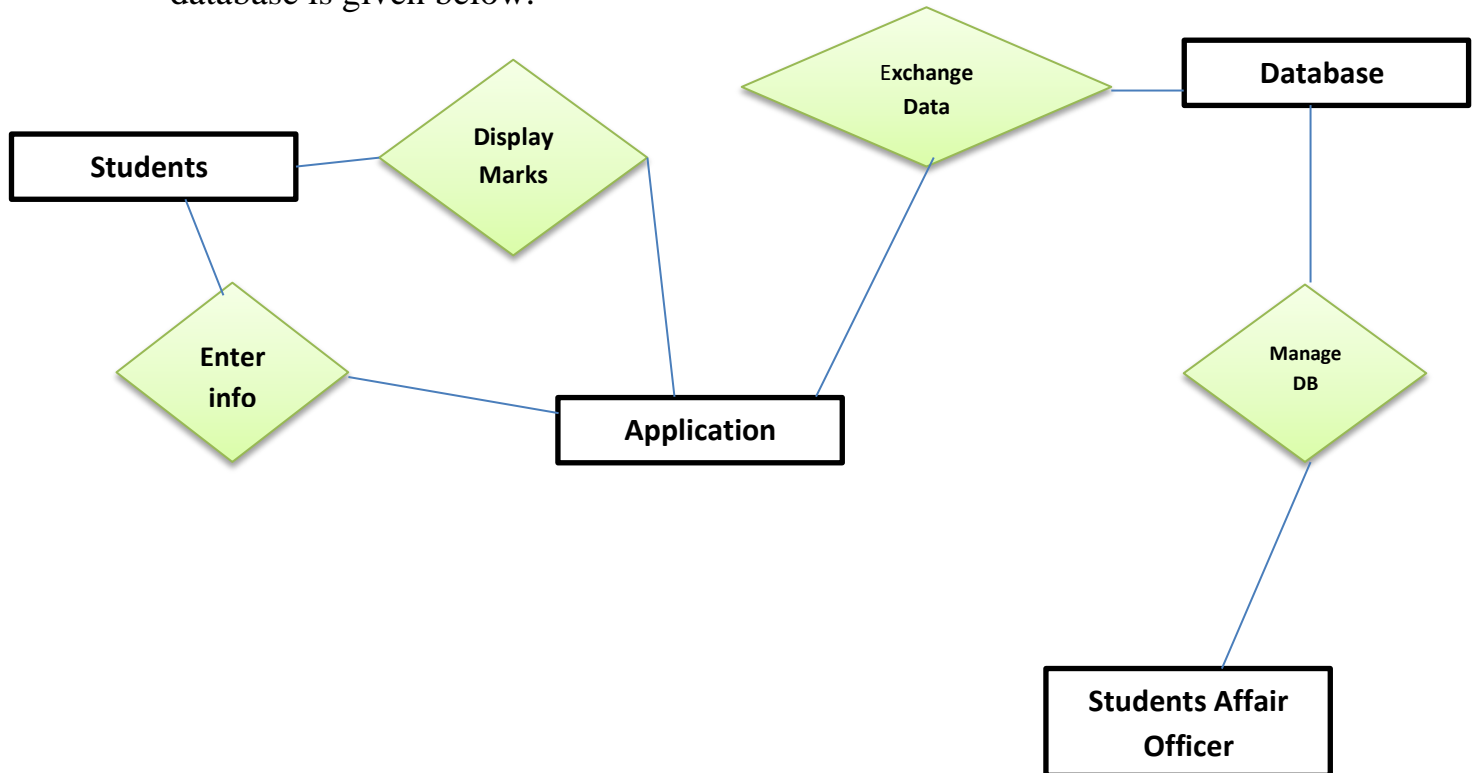


Figure 4 - Logical Structure of the Exam Results App Data

---

### 3.3.2 Security

The server on which the Database will have its own security to prevent unauthorized write/delete access. There is no restriction on read access.

The use

of email by an Student is on the client systems and thus is external to

the system.

Only the Students Affair Officer will have physical access to the Database and the program on it





