

# Software Requirement Specification - Version 1.0

<u>Group Members:</u>	Zakiya Safi	1319070
	Ahmed Ali Karani	1036074
	Zubair Ahmed Bulbulia	1249593
	Kyle Morris	1112649



# **Table of contents:**

<b>1.Project definition:</b>	<b>3</b>
1.1 Scope:	3
<b>2. Specific Project requirements:</b>	<b>4</b>
2.1 External interface requirements:	5
2.1.1 User Types:	5
2.1.1.1 Student:	5
2.1.1.2 Course Coordinator:	5
2.1.1.3 School Administrator:	6
2.2 Interfaces:	6
2.2.1 Hardware Interfaces:	6
2.2.2 Software Interfaces:	6
2.2.3 Communication Interfaces:	7
2.3 System Features:	7
2.3.1 Interaction with the portal:	7
2.3.1.1 Description and Priority:	7
2.3.1.2 Stimulus/Responses:	7
2.3.2 Access Control features:	8
2.3.2.1 Description and Priority:	8
2.3.2.2 Stimulus/Responses:	8
2.3.3 Storage features:	8
2.3.3.1 Description and Priority:	8
2.3.3.2 Stimulus/Responses:	8
2.4 Performance requirements:	9
2.4.1 Response time:	9
2.4.2 Scalability:	9
2.4.3 Workload:	10
2.5 Constraints:	11
2.5.1 Technological constraints:	11
2.5.2 Design Constraints:	11
2.5.3 General Constraints:	12
2.6 Assumptions:	12
2.6.1 Student:	12
2.6.2 Course Coordinator:	12
2.6.3 School Administrator:	12
2.6.4 Database Administrator:	13
2.6.5 Systems Administrator:	13
2.7 Software system attributes:	13
2.7.1 Reliability:	13
2.7.2 Availability:	13

2.7.3 Security:	13
2.7.4 Maintainability:	14
2.8 Other requirements:	14
2.8.1 Documentation:	14
2.8.2 Long-term:	14
2.9 Proposed System Architecture:	14

---

## 1. Project definition:

The project requirement that the group is required to fulfill is that of the *Student Mark Management System*. The end result will be a fully-functioning web portal where students are able to view their marks for a number of courses.

### 1.1 Scope:

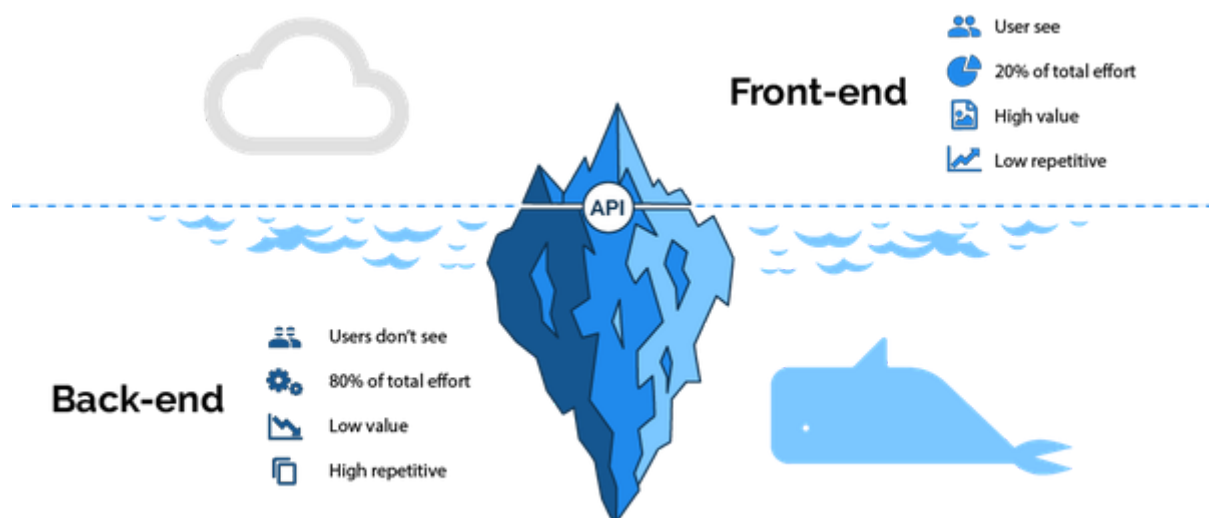
The task requirement is to develop a web-based system with the capabilities of being able to record marks of various assessments, for a multitude of students. It will thus function as a database for all student marks, across various courses. On the front-end, it is required to implement separate functionality and limitations for three different categories of user - student, course coordinator and school administrator. There must exist the capability for Course Coordinator and School Administrator to be able to access, edit and update marks on the database, as well as being able to view student marks. From the student side, they need to be able to view their marks from the 'portal'. They must also have the ability to query a certain assignment mark if they believe there to be an error with it. Students must have the limitation of not having access to other students' marks, and not being able to make modifications to their own marks.

There are two back-end roles that need to be fulfilled: The database administrator, who is responsible for ensuring the correct format of marks is recorded, that there are no erroneous values stored, and that backups exist and are frequently maintained, and the Systems administrator who is tasked with ensuring the correct system configuration is established, as well as the security perspectives of the portal.

## 2. Specific Project requirements:

The group has been tasked with specifically creating a web-based application (herein referred to as the portal). The portal will need to incorporate a fully functioning front-end (User interface), which is simple for all users to navigate (user friendly), aesthetically pleasing, produces the correct output, displays correctly formatted information on the student portal page, and have the correctly implemented security features to grant and restrict access to certain features and/or information, for particular users.

For the back-end (database) requirements, it is necessary to ensure the correct relationships exist between the various tables. It is imperative to select the correct identifying keys (primary and foreign), to ensure the correct relational organisation of database entries is upheld and maintained. Correct data types for the different attribute values must be selected (eg. to store marks/scores for assessments as numerical values rather than text values, to allow the creation of mathematical results, summaries and interpretive statistics). The database needs to be continuously maintained, backed-up and updated to ensure the best user experience on the portal and to rectify any bugs or glitches that may have resulted from incorrect data collection, storing or manipulation.



## 2.1 External interface requirements:

### 2.1.1 User Types:

The user will interact with the portal in different ways, depending on their administrative privileges.

2.1.1.1 Student:	The first type of user is the Student. The Student will be required to 'login' to the portal by using their ID - most likely their 'Student number' and a password (which will be set to their student number by default, and then changed once they have logged in for the first time). The portal will then load the 'student' home page. On this page, they will be provided with an overview of courses they are enrolled in. There will then be a clickable link for each enrolled course, which will take them to an assessment summary page (ASP) (for that course). On that page, all relevant marks will be able to be viewed, as well as the weighting to their overall course mark (a value out of 100). On each ASP, there will be a built-in tool that shows the mark(s) that need to be achieved on their remaining assessment(s) to achieve a pass. There must also be an option to 'report a mark' where a student believes a mark to be incorrectly stored.
2.1.1.2 Course Coordinator:	The next user is a Course Coordinator (CC). This type of user has a much greater administrative role and access on the portal than student. The CC is granted privileges of being able to sign-up to the portal. A password will be given to this type of admin in order to allow the sign-up to happen, and to prevent student from doing same. The CC needs to be able to add students to courses, enter marks for the students, as well as to add weighting to each assessment to define its contribution to the total course mark. All of these functions needs to be done in a restrictive manner. What is meant by this, is that the CC does not need to be aware of the data-storing conventions. Rather, error checking must exist in order to make sure data integrity remains intact when a CC is uploading marks and defining weighting. This needs to be done in the most user-friendly and efficient manner. (an example of such is that forcing marks to be entered by means of a scroll-box will ensure data integrity,

	but will not be efficient, and so to simply have an edit box that implements error checking afterwards is more efficient). Additional features can be present for the CC. This can include options such as statistical analysis to have access to both spreadsheets of data, and potentially even graphs to give a representative overview of students' marks.
2.1.1.3 School Administrator:	The final user type is School Administrator (SA). This type of user will have the same privileges as CC, as well as additional privileges. These include the ability of SA to be able to generate comparative information of courses for which they oversee, in order to identify which courses are performing well, and which are not. Also, as SA, they must be able to view any offenses of students (plagiarism, course exclusion, etc.) which would prevent them from being enrolled in the course.

## 2.2 Interfaces:

### 2.2.1 Hardware Interfaces:

This type of interface describes the physical components which are necessary to interact with the portal. As it is a web based portal, an internet connection needs to firstly be established to allow the user to access the portal, as well as to provide a good user experience by means of the database information being correctly and efficiently displayed on the portal pages. Any networking-enabled computing device will be able to interact with the portal. An input device (such as a physical keyboard) is required to login to the respective portal pages. A pointing device (like a mouse) will provide the best user experience as it provides ease of access with regard to navigating the portal pages. This method would be aimed at a more desktop computer approach. For a mobile approach, the user would likely make use of their touch screen to navigate the portal and make interactions.

### 2.2.2 Software Interfaces:

A web browser (such as Chrome, Safari, Mozilla Firefox, Internet Explorer, or Microsoft Edge) is required to access the system. Our main form of software interaction is done via the front end Graphical User Interface (GUI). This interface allows the user to interact with the

system using the above described pointing device. As this is the primary source of interaction between user and system, it is important to create a GUI which is easy to navigate, not confusing to use, and that the interactions produce the expected results. The system will also

### 2.2.3 Communication Interfaces:

Communication needs to exist on the portal between the back-end databases and the web-based application. Communication will be made possible through SQL.

## 2.3 System Features:

### 2.3.1 Interaction with the portal:

#### 2.3.1.1 Description and Priority:

The requirements of this feature set define how the system allows viewing, updating, modification and creation of mathematical statistics pertaining to specific students and courses within the portal. The system is a web-portal which functions as an electronic marks source, which allows students to view marks, and administrators to upload and edit marks.

#### 2.3.1.2 Stimulus/Responses:

**Stimulus:** a student wants to view their marks for a specific course.

**Response:** The system uses the web-portal to avail all marks pertaining to that student.

**Stimulus:** A user wants to see what mark they need to pass the course.

**Response:** A statistical analysis is made, tallying up their completed assignments and weightings, and what is still to be completed, to determine the mark required to pass.

**Stimulus:** Course Coordinator wants to upload new marks.

**Response:** The system allows the entry of marks, which are then stored on the database.

**Stimulus:** Student wants to report a mark error.

**Response:** An option exists to send an error to the course coordinator, which will automatically be associated with the mark they are currently accessing.

**Stimulus:** School administrator wants to remove students who has committed infringements.

**Response:** The system allows all infringements associated with students to be viewed, as well as the administrative privileges of removing a student.

### 2.3.2 Access Control features:

#### 2.3.2.1 Description and Priority:

The requirements for this feature set describe how users are granted privileges as well as having access to certain features limited, depending on their login-type.

#### 2.3.2.2 Stimulus/Responses:

**Stimulus:** A student wants to change one of their marks.

**Response:** They are restricted to read-only access for their marks. They can send a query to admin if they believe a mark to be incorrect.

**Stimulus:** A student wants to view one of their friends marks.

**Response:** The student is restricted from doing this, as their login provides them with access to the marks associated with their login details only.

**Stimulus:** A Course Coordinator wants to upload new marks for an assessment.

**Response:** The login details will grant admin the ability to modify and upload new assessment records.

### 2.3.3 Storage features:

#### 2.3.3.1 Description and Priority:

The requirements for this feature set describe how data is uploaded to the database, specifying data types, and how information is obtained from the portal.

#### 2.3.3.2 Stimulus/Responses:

**Stimulus:** A student logs in, wanting to see their latest marks.

**Response:** The system sends a query to the database, to load and display the marks associated with their login ID.



**Stimulus:** A Course coordinator wishes to upload marks, but accidentally records a character instead of a numerical value.

**Response:** Error checking will evaluate the input before sending the query to store the value in the database. This reduces complexity as it ensures no error data types can be stored for marks.

**Stimulus:** A Course coordinator wishes to add a student to the system, but that student already exists.

**Response:** As student is defined by ID, this is the primary key, and so a prompt must show that this action is invalid as the student already exists on the database.

## 2.4 Performance requirements:

### 2.4.1 Response time:

The system needs to respond quickly regardless of device being used. Any delay will reduce the quality of the user experience with the portal. In order to minimise delays, scripting needs to be made as efficient as possible, to make the portal appear to have as instantaneous a response as possible. There are different response times to consider: when logging in, a user can sometimes expect to have a slight delay while their request is processed. However, once the page loads, the user expects to be able to view a fully rendered interface. Thus it would make more sense to force the system to wait slightly longer when logging in to achieve this, rather than a delay when logging in, and another delay to view their marks. With statistical analysis, as this is described as a 'feature', it is assumed that a user may be more accepting to wait slightly longer to receive the requested output.

### 2.4.2 Scalability:

The system should be able to handle increasing number of students and coordinators without major changes being made. The system should be developed in such a way that as the user base grows, storage and processing power can be increased without causing major disruptions to the day to day use of the platform. During the testing phase of the system, scalability will not be as much of a concern as ensuring the system functions optimally. However, new errors can arise under much higher traffic volumes and so it is important to try identify the limitations of the system during the testing phase in order to be better aware of how greatly resources need to be increased in the event of average traffic volumes growing rapidly. It is also important to ensure that if such a time occurs, the system has been

designed in a simplistic enough way to allow for the database to be migrated to a more suitable location.

#### 2.4.3 Workload:

The workload on the users device through the portal should be minimal as the device being used can vary from low-powered devices (smartphones) to high-performing devices (computers). The major processing tasks should therefore be performed on an external machine (server) in order to reduce processing load from the end users. Furthermore, a variety of transactions will exist with the portal: firstly are requests from the student (user) to view marks. This is a priority request as a large faster contributing to the efficiency of the portal is the user experience. Another transaction will be that of a Course Coordinator uploading marks for a course, for a number of students. A problem could arise when a student is trying to access their marks whilst a modification is being made to their portfolio, and so scheduling needs to be implemented for administrators to either upload marks during times of low network traffic, or for students to be restricted access during this time. Another transaction that needs to be prioritised is that of backups. It is vitally important to have up-to-date records of data. This however can be scheduled for low network traffic times and so will reduce the possibility of delays and corrupt data.



## 2.5 Constraints:

### 2.5.1 Technological constraints:

- The portal must be available for use by students and Course Coordinators 24/7. This means the server needs to be always active and accessible.
- The system must be usable on devices of varying performance. This means that the system must be optimised to ensure that it works regardless of the device being used to access the system.
- The system must be developed using an ASP.NET MVC framework and will include technologies such as C#, HTML, and CSS. This means the expansion of the system and design is restricted by the limitations of the coding language.
- The system will also integrate mathematical packages using R or Matlab for carrying out statistical calculations and data analytics, as well as plotting necessary graphs.
- The database must be able to handle multiple, simultaneous transactions without losing or corrupting data. This will be guided by the implementation of the database itself.
- Effective running of the system will be a direct result of the implementation and design.

### 2.5.2 Design Constraints:

- The user interface must be intuitive, and visually appealing.
- The GUI must be responsive to user inputs, and must produce correct outputs as a result of this. This will be restricted by providing a limited number of options which the user can access.
- The system must work on all modern day computing devices (Personal Computers, Laptops, Smartphones, Tablets).
- The user interface must be designed in a dynamic manner that allows for scalability. The user interface must render correctly according to the display size. All associated components must correctly resize and maintain the visual layout component of the portal.

### 2.5.3 General Constraints:

- The portal must be developed and completed in the given period of time (roughly 4 months)
- The system is limited by the server hardware specifications on which it will be run.
- The system will be developed and implemented by only 4 people, meaning task allocation needs to be carefully allocated, and deadlines need to be met.
- As it is a group effort, collaborative input is required. Decisions need to be made quickly and efficiently.
- The back-end of the system requires raw data in order to work correctly. This means that 'test' courses and marks either need to be created, or existing mark spreadsheets need to be obtained. In the case of the latter, confidentiality of information needs to be maintained.

## 2.6 Assumptions:

### 2.6.1 Student:

- A. Student marks will be available from the portal
- B. Students feel comfortable logging into and using the web portal
- C. Students have an understanding of a web-based application and so understand possible system delays.
- D. Students make effective use of the system.

### 2.6.2 Course Coordinator:

- A. CC will be responsible for uploading marks onto the database
- B. CC is comfortable with dealing with and making changes to the system
- C. CC understands how to apply a certain weighting to a course.

### 2.6.3 School Administrator:

- A. SA knows how to navigate between different courses
- B. SA is comfortable with creating comparative statistics on different courses.
- C. SA is able to identify students with offenses and can remove them from the course (unenrol).

#### 2.6.4 Database Administrator:

- A. The database is correctly created with the appropriate relationships between entities being implemented.
- B. The data types of respective fields are appropriately defined.
- C. Any erroneous data that appears during testing will be rectified.
- D. Backups will be done regularly, and stored appropriately.

#### 2.6.5 Systems Administrator:

- A. The security of the system will ensure no ungranted access is provided to respective users.
- B. The system will communicate between the front and back-end without error.

### 2.7 Software system attributes:

#### 2.7.1 Reliability:

To ensure the reliability of the system, all data contained in the database must be correct and up-to-date at all times, and the database itself must be functional and connected to the system at all times. The portal must have a mechanism to detect failures and other errors and alert administrators so these will be fixed in the shortest possible period of time.

#### 2.7.2 Availability:

The portal must be available for use at any time, on any web browser (including mobile browsers), and should it not be available for any particular reason, must recover from crashes in a short period of time (no longer than a day).

#### 2.7.3 Security:

Access to the portal must be restricted by means of user authentication. The system must have the ability to prevent SQL injections and encrypt all confidential data such as passwords and marks.

#### 2.7.4 Maintainability:

To ensure the maintainability of the system, the code must be written in a readable, structured and consistent manner. This will allow modifications to be made at any point without undesirable side effects.

### 2.8 Other requirements:

#### 2.8.1 Documentation:

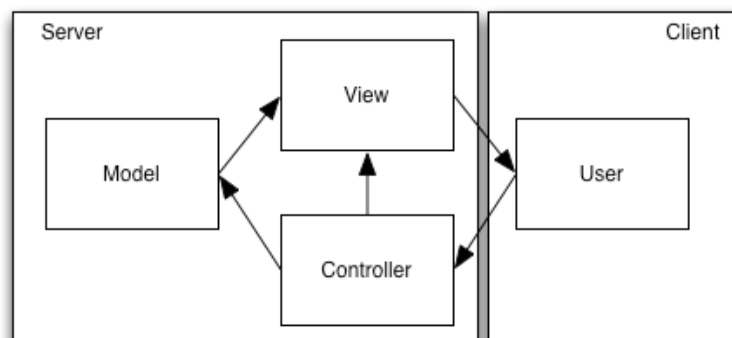
It is required to continually update the requirement specification as system implementation changes, or as constraints highlight features which cannot be implemented. The documentation must be easy to read, informative, and must provide a clear approach as to how the task is to be implemented.

#### 2.8.2 Long-term:

The aim of the project is to develop a fully working system that can be upgraded and/or modified where necessary. Once the testing phase is complete, and all initial errors are removed and solved, the system must move into a maintenance phase. This is where most focus is placed on the database, upkeep and maintaining of the portal, only making large, visible changes when vitally necessary, or during a version update release.

### 2.9 Proposed System Architecture:

The system needs to take into account all requirements and constraints, and based on these a web-based application will be implemented. Following the requirements, it should make use of the Model View Controller (MVC) pattern of software architecture. Additionally, this will also need to be implemented in a client-server setting due to users' access requirements as well as for better usability.



---

END OF DOCUMENT

---