# System Requirement Specifications (SRS)

University Student Management System (USMS)
Web Application

Josiah Prathaban
SE/2071/022

Table of Contents

## 1.Introduction

This section gives a scope description and overview of everything included in this SRS document.

### 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Student Management System” Software. It will illustrate the purpose and declaration for the development of system. It will also explain system 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 (myself).

### 1.2 Scope

The USMS is a web application that facilitates university admins to manage students and also facilitates students to enroll in courses.

The students can use web-portal to update their information and enroll for a course. Admins also use web-portal to view and manage students.

All system information is maintained in a database.

## 2. 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.

### 2.1 Product perspective

This system is fully a web application. Spring boot is used for the back-end application development and Angular11 is used for the front-end application development.

There will be a common login page for all types of users. And there will be separate pages for each users to interact with the system.

Scene this is a data-centric system it will need somewhere to store the data. For that, a relational database will be used. MySQL  is used for the database management.

### 2.2 Product functions

This is a simple web application that gives an interface for users to indirect with the subpart of the university management system called student management system.

User can log in and the system will show them the actions that which are available for that particular user and give them an interface to do the actions (view, edit, delete, etc…)

### 2.3 User characteristics

There are two types of users that interact with the system: students, and admins. Each of these three types of users has different use of the system so each of them has their own requirements.

The student can login to the system and he can update his information,can change his password, and can enroll for the course which are available for him.

Admin can login to the system and he can add, delete, update, view students in the system.

### 2.4 Assumptions and dependencies

This USMS is a sub system of University management system. So the Admin of University Management System will manage Lecturers, courses, and admins of USMS. USMS admins only have permission to interact and manage with student-related database entities, they can't make any changes in lecturers, course, or admin database entities.

## 3. Specific requirements

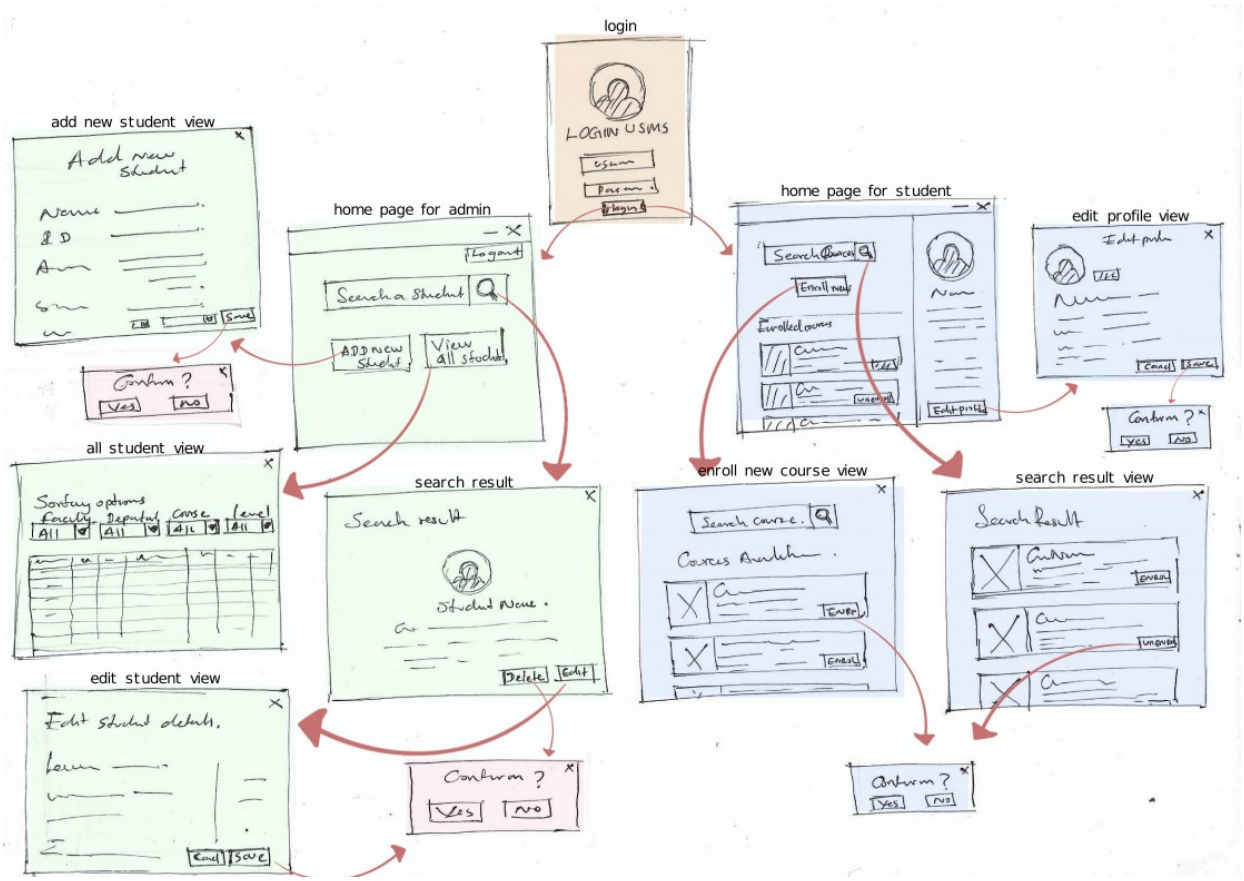
This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

### 3.1 User interfaces

There will be a common login page for both users. The system will Identify the type of user using their username. And it will route to the appropriate home page (as shown in the figure).

Then from the Students' home page a student can enroll or unroll for a course, can search for a course or can edit his profile information. After student done an action there will be a dialog box to confirm the changes. Then only the system will update the database.

From the admins' home page a admin can search for a student and edit, add a new student, or view all student and edit. After admin done an action there will be a dialog box to confirm the changes. Then only the system will update the database.




### 3.2 Functional Requirements

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

#### 3.2.1 User Class 1 - The Student

ID: FR1
TITLE: User log-in
DESC: Given that a user has registered, then the user should be able to log in to the web application. The log-in information will be stored on the browser and in the future the user should be logged in automatically.

ID: FR2
TITLE: Home page - Search
DESC: Given that a user is logged in to the web application, then the first page that is shown should be the home page. The user should be able to search for a course that is eligible for him.

ID: FR3
TITLE: Home page – enroll for new course
DESC:  In home page the user should be able to enroll for a course that is eligible for him.

ID: FR4
TITLE: Enroll for new course result view
DESC: Should show all the available unroll coerces for the user.

ID: FR5
TITLE: Search result
DESC: If the student search for a course from home page the search result should show enrolled and new courses related to the search key. If the student search for a course from the enroll for new course result view then the  search result should show only unrolled new courses related to the search key.

ID: FR6
TITLE: Home page – view profile
DESC:  In home page the user should be able to view his profile details.

ID: FR7
TITLE: Home page – view enrolled courses
DESC:  In home page the user should be able to view the course details that he already enrolled.

ID: FR8
TITLE: Home page – edit profile
DESC:  In home page the user should be able to edit his profile details.

ID: FR9
TITLE: Home page – unroll courses
DESC: In home page the user should be able to unroll the course that he already enrolled.

ID: FR10
TITLE: Action Confirmation
DESC: Should confirm any action before make changes to the database.

### 3.2.2 User Class 2 – Admin

ID: FR11
TITLE: User log-in
DESC: Given that a user has registered, then the user should be able to log in to the web application. The log-in information will be stored on the browser and in the future the user should be logged in automatically.

ID: FR12
TITLE: Home page - Search
DESC: Given that a user is logged in to the web application, then the first page that is shown should be the home page. The user should be able to search for a student using  keywords like name, id, age or address etc.

ID: FR13

TITLE: Home page – view all students
DESC:  In home page the user should be able to view all the students.

ID: FR14
TITLE: View all students result view
DESC: Should show all the all the student details or sorted student details and sorting options.

ID: FR15
TITLE: Search result
DESC: Should show all the matched search results.

ID: FR16
TITLE: View all students result view - edit
DESC: The user should able to edit a students' information

ID: FR17
TITLE: Search result - edit
DESC: The user should able to edit a students' information

ID: FR18
TITLE: View all students result view - delete
DESC: The user should able to delete a student

ID: FR19
TITLE: Search result – delete
DESC: The user should able to delete a student

ID: FR20
TITLE: Home page – add new students
DESC:  In home page the user should be able to add a new student

ID: FR21
TITLE: Action Confirmation
DESC: Should confirm any action before make changes to the database.