Software Requirements Specification

For

Course Enrollment System

Prepared by: Group 24

Course: CS340-Databases

Instructor: Naveed Arshad

Date created: 8 October 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Maryam Shakeel	14/10/2022	Writing the document	
Umer Malik	14/10/2022	Writing the use cases	

1. Introduction

• Purpose

This document describes the software and hardware requirements and the program features of the course enrollment system of Lahore University of Management Sciences (LUMS). It is intended to be used by the students, instructors and the departments of each school within the university. This document describes the constraints under which the system is designed and the cases under which the system must operate. This system is designed to be utilized by the course staff, university administration (departments in this case) and the students of LUMS.

• Document Conventions

The entire document is written in Arial using the following conventions:

1. The sections are written in bold
2. The sub-sections are written in bold
3. The important sub notes are written with an asterisk (*)
4. Sub-headings are written in bold and underlined for emphasis

• Intended Audience and Reading Suggestions

This document is prepared for the teaching staff of CS 340 - Databases. The overall Description Section explains in detail the features being used in the system and the user interaction with the system. The demonstration of these features in correspondence to the users is done in the Use Cases section which shows how different users (actors) can utilize this system effectively. The Software and Hardware Description in the external requirements are to be read along with security functionalities so that the client gets a detailed overview of how the system caters to the security concerns by deploying it in software.

• Project Scope

The function of this course enrollment system is to allow students to enroll in courses being offered at the university online. It will take into consideration the student's batch, major, department and total credits taken according to which the student can enroll in courses. The course staff will be able to choose the courses they want to offer and the capacity of each section. The departments will be able to see the records of the students and courses. The goal is to make enrollment as intuitive and simple as possible, with an easy way to search for courses based on the student's requirements and restrictions and add them to their weekly schedule.

The overall purpose of the system is to provide an online platform where the different departments of the university can work with the individual instructors and students and allow them to enroll into the courses that are offered at the university. Each actor will have a different level of authority.

• References

Detailed blog on what an SRS document is:

https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

Summarized overview of what an SRS document requires:
https://enisinanaj.medium.com/writing-a-software-requirements-specification-document-97d622805aef#:~:text=Document%20Conventions%20is%20an%20optional,used%20for%20and%20so%20on.

Overall Description

• Product Perspective

This product is a follow-up of a larger system of the LUMS software (Zambeel) that deals with student finances, academics and counselling, hostel registration but in our case, it is strictly limited to the course enrollment perspective.

The larger system has several components that will be linked to the course registration system. The financial system will be directly linked to the course enrollment making it unable for students who have not cleared their dues to be able to enroll in courses. Similarly, the academics and counselling system will be in direct correspondence with our system where students who have not attended their mandatory advisory meeting will have their course enrollment system blocked. The larger system will all be on the same interface while giving the client the option to choose the sub-systems they want to access. This system will allow for intuitive and easy enrollment of courses for students offered by instructors in the upcoming semester.

The stakeholders of this product are the clients for which the product is being designed, which are the teaching staff of Cs –340. The stakeholders are kept in the loop for any updates and changes made to the system.

• Product Features

The course enrollment system for LUMS allows all users to have these basic functionalities:

1. The user can login and sign out of the system. (MUST login to use the system?)
2. The user can search for courses.
3. The staff can update the list of courses, sections, time slots and instructors.
4. The students can add/drop the courses.
5. The user will be able to browse the course catalog.

The three main users of this system are the students, instructors and the department staff.
The students can login to the system and browse the course catalog, search for specific courses, add/drop courses according to the availability of shopping cart, enrollment time and credits available taking into consideration their batch, major chosen, total credits taken and the department they belong to. The department will offer a list of courses to the instructors that they can choose to teach from and decide the credit hours the courses will have as well as the time slot of each section of the course. The instructors in this design can choose the courses they want to teach from the list the department provides, keep certain pre-requisites for the courses and course capacity. In this schema the students cannot add classes that clash with each other.

After the enrollment for the students is done, the system will create a weekly schedule that is clash free for every course the student decides to take. The system will not allow sections to be in the same classroom at the same time and no student will have multiple courses at the same time slot.

*Students who do not fulfil course pre-requisites will not be able to enroll in the course.
*Students who have not cleared their dues will not be able to enroll in any course.

• User Classes and Characteristics

User characteristics:

1.The user should be literate with computers.
2.The users should know how to use the internet.
3. Must be able to use the system interface and complex features.

The list of the favored user classes is as follows:

- **Department:**
  The department is the admin in our system. They have full access to the system. In this case the departments have the authority to change class and section timings, give grades to students after they have been finalized by the instructors, provide a list of courses to the instructors to choose from and finalize the instructor, set up timings of each course, decide the credit hours that each course will have and create login accounts for students and instructors. The department for each major within a school will be different, I.e., the computer science department will be different from the chemistry department.

- **Instructors:**
  The instructors are the employees of LUMS who will be teaching a course in the semester and even those who are on a sabbatical or not teaching a course in the current semester but have an instructor status from the department they are in. They will actively be able to login, logout, make edits to their courses and descriptions, add marks of their course components.

- **Students:**
  The students are the currently active and registered students of LUMS. They will not have access to the system features that allow them to change the environment or hardware or the software. The students can be from any batch that has not yet graduated I.e., has not completed their total credit hours requirement to graduate. They can only use the functionalities such as login, log-out, browse courses, search for specific courses, add or drop courses. There are approximately more than 4000 students who are expected to be using this system to enroll in courses. 70-80% of these individuals are expected to use this system 4-5 times in a semester.
  *Students will be able to login after providing their username and password and authentication is complete.

• Operating Environment

This system requires a stable internet connection that can favor the loading of the huge web application of this system. Any device and operating system that has access to the internet can use this online system. This system does not require fixed software or hardware, any up-to-date operating system can run this system such as Microsoft Windows 7 and above, macOS, Linux.

- **Design and Implementation Constraints**

The databases need to be regularly updated to avoid inconsistencies in data that is uploaded to the system via these databases and doing so results in a lot of storage being utilized as well as time constraints are there. The system is to be regulated by the clients and the developer needs to maintain modularity in such cases so that the clients can easily make changes to the software without needing to rebuild the entire system. This system uses network-based applications so the system should be developed in a way that it can be used on any web browser. The system is to be designed in a way that security and system validation are not of any concern when the system is handed over to the client. For maintenance and security issues the developers will need to be there for customer support.

- **User Documentation**

The primary actors and users of the system are the target audience for the user documentation. Documents such as user guides, user manuals, video tutorials, FAQ's (frequently asked questions) will be made available in PDF format to the client which can be shared to the respective users of the system.

- **Assumptions and Dependencies**

The software will require an internet service as it is a network-based system without which the system will not work. The assumptions made in this document are as follows:

1. All users of the system have a reliable internet connection and a device which supports web applications.
2. The users are registered in the university and have a valid user ID.
3. The students enrolling on the courses have all their finances cleared and no dues left to be paid.
4. The students fulfill the course pre-requisites, and they need not be stopped from enrolling in the course.
5. The number of students on a course does not exceed the capacity of the course or else they cannot be enrolled in the course.
6. The course registration system follows a first come first served basis.
7. The instructor has not scheduled a course in the same class with the same time slot.
8. It is assumed all instructors offering the courses are offering them from their own department.
9. It is assumed that all users have read the user documentation and know how to use the system.

2. Use Cases and System Features

Use Case List

Primary Actor	***Use Cases***
Student	1.Student course enrollment 2. Drop courses 3. Add courses 4. Browse courses catalog
Instructors	5. Instructors choosing courses 6. Defining course pre-requisites and credit hours 7. Update course information
Department	8. Offer course lists 9. Select time slots 10. Maintain student information
Students/Instructors	11. Login 12. Log out 13. Sign up 14. Update password

Use Case Template

Use Case ID:	1.0		
Use Case Name:	***Student Course Enrollment***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	**Students**
Description:	This use case is for students to select a course according to the conditions and enroll in it if it has capacity.
Trigger:	Beginning of Enrollment Phase, one month before beginning of the next semester.
Preconditions:	1. Students must not be on academic suspension or leave. 2. Students must fulfill course pre-requisites. 3. Student dues must be cleared before enrollment.
Postconditions:	1. Students must not be enrolled in more than the credit hour limit per semester. 2. A course must not clash, as in occur in the same timeslot as another course. 3. The course must have capacity for the student to enroll.
Normal Flow:	***1*.1 Adding enrolled course to weekly schedule**

	1.When the student selects a course, the system will search the database for previously enrolled courses of the student, the course capacity and the credit hour limit. 2. If the conditions for each of these are met, the system will add the course to the weekly schedule of the student. 3. The database will be updated.
Alternative Flows:	**1.1 The course or student does not pass one of the conditions.** 1. The system prompts that this course cannot currently be enrolled in, and it is not added to the student's weekly course schedule.
Exceptions:	**1.0.E.1 Enrollment failed due to overdue finances** 1. System informs the student that enrollment failed due to overdue fees 2. A new fee challan is issued for the student 3. Student clears the dues and can enroll in courses provided enrollment is live **1.0.E.2 Enrollment time has passed** 1. System informs the student that they do not have a valid appointment time 2. Student waits for the next appointment to enroll in courses
Includes:	1. Login and logout of the student
Priority:	High
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	1.Student should be able to drop the courses they no longer require after enrollment.
Assumptions:	1. Student knows how to use the enrollment system
Notes and Issues:	None

Use Case ID:	2.0		
Use Case Name:	***Drop Courses***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students
Description:	Students can drop a course within two weeks of the start of the semester
Trigger:	Start of the semester
Preconditions:	1. Student must be currently enrolled in the course they choose to drop
Postconditions:	1. The students' total credit hours must not fall below 12
Normal Flow:	**2.0 Dropping a course** 1.When the student selects a course to drop, the system will search the database for whether the student is enrolled in the course and if they are, would removing the course put them below credit hour minimum. 2. If the conditions for each of these are met, the system will remove the course from the weekly schedule of the student.

	3. The database will be updated.
Alternative Flows:	**2.1 The course or student does not pass one of the conditions.** 1. The system prompts that this course cannot currently be dropped, and it is not removed from the student's weekly course schedule.
Exceptions:	**2.0.E.1 Enrollment time has passed** 1. System informs the student that they do not have a valid appointment time 2. Student waits for the next appointment to drop courses
Includes:	1. Login and logout of the student
Priority:	Medium
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	1. If the student wants to drop a course after the add/drop period has passed they cannot do so. However, the department can withdraw them from the course in this case.
Assumptions:	Student knows how to use the enrollment system
Notes and Issues:	None

Use Case ID:	3.0		
Use Case Name:	***Add Courses***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students
Description:	Students can add a course within two weeks of the start of the semester.
Trigger:	Start of the semester
Preconditions:	1. Students' total credit hours must not exceed the total credit hour requirement. 2. Students must fulfill the pre-requisites of the course. 3. Student dues must be cleared before adding a course.
Postconditions:	1. Students must not be enrolled in more than the credit hour limit per semester. 2. A course must not clash, as it occurs in the same timeslot as another course. 3. The course must have capacity for the student to enroll.
Normal Flow:	**3.0 Adding a course** 1.When the student selects a course to add, the system will search the database for previously enrolled courses of the student, the course capacity and the credit hour limit. 2. If the conditions for each of these are met, the system will add the course to the weekly schedule of the student. 3. The database will be updated.
Alternative Flows:	**3.1 The course or student does not pass one of the conditions.** 1. The system prompts that this course cannot currently be added, and it is not added to the student's weekly course schedule.
Exceptions:	**3.0.E.1 Add drop period has passed**

	1. System informs the student that they do not have a valid appointment time and course cannot be added to the schedule. **3.0.E.2 Overdue finances** 1. System informs the student that course could not be added due to overdue fees. 2. Student clears the dues and can enroll in courses provided add/drop period is live.
Includes:	1. Login and logout of the student 2. User authentication while logging in to the system.
Priority:	Medium
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	1.Student should be able to drop the course they have added if they no longer require it provided the add/drop period is still live.
Assumptions:	Student knows how to use the enrollment system
Notes and Issues:	None

Use Case ID:	4.0		
Use Case Name:	***Browse Course Catalog***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students
Description:	Students will be able to browse and run a customized search on all courses that they are allowed to enroll in according to their year and department. The search can be based on school, department, instructor, course id, section, time slot, classroom, credit hours and pre-requisites.
Trigger:	Beginning of Enrollment Phase, one month before beginning of the next semester.
Preconditions:	Students must be enrolled at the university.
Postconditions:	None
Normal Flow:	**4.0 Searching each category of courses** 1.Each category will have a text field the student can fill to filter the search. 2. The system will look through the database for the relevant attributes and the system will show results even if they only contain part of the string.
Alternative Flows:	**4.1 No search results** 1.If the selected filters result in no searches, the system will display a prompt that says no courses match
Exceptions:	**4.0.E.1 No valid appointment time** 1.The student will only be able to search for courses during enrollment period otherwise the system will give a 'No valid appointment time' error.

Includes:	User authentication while logging in and logging out.
Priority:	High Priority
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	None
Assumptions:	1. Student knows the courses they need to enroll in to fulfill their requirements.
Notes and Issues:	None

Use Case ID:	5.0		
Use Case Name:	***Instructors choosing courses***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Instructors
Description:	Instructors will be offered a small list of courses by the department, and they will be able to select the courses they want to teach.
Trigger:	Beginning of Enrollment Phase, two months before beginning of the next semester.
Preconditions:	1. The instructor is not on sabbatical or leave. 2. The instructor is at least at the position of an Associate Professor.
Postconditions:	1. The instructor cannot teach more than 2 different courses. 2. They must select the course at a specific time (Before enrollment starts) 3. Instructor can only teach a course from their expertise.
Normal Flow:	**5.0 Instructor selects the courses according to their expertise** 1.After logging in, the instructor will be presented with the different courses within the department 2.There will be text fields that the instructor can fill to filter the search. 3.The system will look through the database for the relevant attributes and will show results even if they only contain part of the string. 4. Instructor will select the course they want to teach, and the system will process the result. 5. The instructor will be added to the course instructor list after the department confirms their expertise.
Alternative Flows:	**5.1 Instructor wants to teach a course that is not available** 1.If the selected filters result in no searches, the system will display a prompt that says no courses match. 2.The instructor will have the option of selecting another course
Exceptions:	**5.0.E.1 The instructor does not want to teach a course** 1.In this case the instructor will either do a research project or guide the senior year projects of students.
Includes:	User authentication.

Priority:	High
Frequency of Use:	Twice in a year
Business Rules:	None
Special Requirements:	The instructor can only teach two courses in one semester.
Assumptions:	The instructor knows the system well.
Notes and Issues:	None

Use Case ID:	6.0		
Use Case Name:	***Instructors selecting the Pre-requisites/credit hours***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Instructors
Description:	Once the instructor has been assigned a course, they can input any pre-requisites courses that the students need to take as well as the number of credit hours of the course.
Trigger:	The instructor must be assigned to teach that course.
Preconditions:	The instructor must not be on leave or sabbatical.
Postconditions:	The instructor cannot put more than 4 credit hours or more than 3 pre-requisites for a course. This check will be regulated by the department.
Normal Flow:	**6.0 Instructor chooses the pre-requisites and credit hours for the course** Once the instructor is assigned to a course, they can select the courses which will need to be taken by the students as pre-requisites, by using the search functions. They can add these courses as pre-requisites in the form of a weak entity set to the course they are teaching. Then from a selection of credit hours, they can select the credit hours the course would have.
Alternative Flows:	**6.1 The system gives an error while keeping the pre-requisites** 1.The system will need to be refreshed to give the instructor the authority to do so
Exceptions:	**6.0.E.1 The instructor does not keep a pre-requisite of the course that requires a pre-requisite** 1.The department will keep the pre-requisite in this case. **6.0.E.2 The instructor keeps a course with 5 credit hours** 1.The department checks the difficulty of the course and keeps it to 4 credit hours if required.
Includes:	User authentication
Priority:	High
Frequency of Use:	Twice a year
Business Rules:	None

Special Requirements:	The instructor can add special requirements like only third year students or students with an A- or above in the pre-requisite course can enroll in the course.
Assumptions:	The instructor himself has not taught the courses he is keeping as pre-requisite.
Notes and Issues:	None

Use Case ID:	7.0		
Use Case Name:	***Update course information***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Instructors
Description:	The instructor can edit the course description from when they accept the course until enrollment begins.
Trigger:	Onset from when the instructor selects the course.
Preconditions:	The instructor must be teaching that course.
Postconditions:	None
Normal Flow:	**7.0 Adding course information** 1.Once the instructor has accepted the course, they will be prompted to input a description of the course. 2. The system will be updated with the new course information.
Alternative Flows:	**7.1 The instructor is unable to change the information** 1.The system will need to be refreshed to give the instructor the authority to do so.
Exceptions:	**7.0.E.1 Change of course information** 1.The description cannot change once the enrollment appointment has passed
Includes:	User authentication.
Priority:	Medium
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	None
Assumptions:	The instructor gives an insightful description of the course
Notes and Issues:	None

Use Case ID:	8.0		
Use Case Name:	***Offer course lists***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Department

Description:	Each department will input a list of courses that will be selected by the students and instructors for that semester.
Trigger:	Before the enrollment period starts.
Preconditions:	1.The course list must be input at a specific time. A course cannot be added in the middle of the semester after the enrollment period has ended. 2.The department can only add courses from their school. (HSS school cannot input courses that are Computer Science Based)
Postconditions:	1. A course can be dropped if not enough students enroll in it. However, major cores or school core courses cannot be dropped.
Normal Flow:	**8.0 Adding courses to the system** 1.The department will input a list of courses, one by one. 2.If the course has not already been added, then it will be added into the system.
Alternative Flows:	**8.1 The course has already been added in the list** 1.If the course with the same name has been added then a prompt will show up "Course already added" and the system will not add the course to the list.
Exceptions:	None
Includes:	User authentication
Priority:	High Priority
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	None
Assumptions:	The instructors receiving the courses should have sufficient knowledge and experience to teach the course.
Notes and Issues:	None

Use Case ID:	9.0		
Use Case Name:	***Select time slots***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Department
Description:	Each department will input a range of timeslots, and the system would automatically allocate the courses to these timeslots based on a clash free algorithm
Trigger:	Before the start of the enrollment period
Preconditions:	1.The course list must be input at a specific time. 2. A timeslot cannot be added that is before 8:00am or after 8:00pm 2. A timeslot must not clash with another course that is a major core of that year.
Postconditions:	1. The time slot must be kept keeping in mind the number of credit hours of the course
Normal Flow:	**9.0 Adding time slots** 1.The department will input the range of timeslots, one by one.

	2. If the timeslot has not already been added, then it will be added into the system. 3. If the timeslot has been added for that course, then a prompt will show up "Timeslot already added".
Alternative Flows:	**9.1 Not enough time slots** 1.In the case where there are not enough timeslots to allocate the classes, a prompt will show up "Not enough timeslots available" and more entries would need to be taken.
Exceptions:	None
Includes:	User authentication
Priority:	High Priority
Frequency of Use:	Twice a year
Business Rules:	None
Special Requirements:	None
Assumptions:	The students will ensure themselves that the classes they enroll in do not have a clashing time with another they add.
Notes and Issues:	None

Use Case ID:	10.0		
Use Case Name:	***Maintain Student Information***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Department
Description:	Each department would be able to update the student's information like a major change or a year/semester off.
Trigger:	Department receives information from a separate system of the university's management.
Preconditions:	Students must belong to that department.
Postconditions:	None
Normal Flow:	**10.0 Updating student information** 1.The department would be able to edit the students' information that pertains to enrollment and update the database.
Alternative Flows:	**10.1 Student does not belong to the department** 1.In case the student does not belong to the department, the system would fail to update and return an error.
Exceptions:	None
Includes:	User authentication.
Priority:	Medium Priority
Frequency of Use:	When changes are made to student information
Business Rules:	None
Special Requirements:	Studeny can request change of information from the department like if he changes his address
Assumptions:	The department will make the appropriate and correct changes
Notes and Issues:	None

Use Case ID:	11.0

Use Case Name:	***Login***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students, Instructors, Department
Description:	Each actor would have their unique usernames and passwords and would dictate (depending on the actor) the authority they have.
Trigger:	On opening the website, they would be prompted to enter their login credentials.
Preconditions:	They must be either a student, teacher or department in the university
Postconditions:	Users will have to authenticate their information after logging in.
Normal Flow:	**11.0 Logging in** 1.There will be 2 text fields, one to input for the username and the other to input the password. 2. The student will then authenticate their identity.
Alternative Flows:	**11.1 Incorrect information entered** 1. If the user enters an incorrect username or password, the system will return an error message. 2. The system will take input again up to 5 more tries. 3. After 5 tries the system will block the user for a couple of minutes.
Exceptions:	None
Includes:	None
Priority:	High Priority
Frequency of Use:	Every time the user tries to login
Business Rules:	None
Special Requirements:	1. User will be able to know if someone tries to login with their account by the email they receive from the system.
Assumptions:	1. The department admin has created the accounts for the users.
Notes and Issues:	1. Third party users can try to login to the system.

Use Case ID:	12.0		
Use Case Name:	***Logout***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students, Instructors, Departments
Description:	Each actor would be able to securely log out of their account to prevent anyone else from accessing their account on the device they use.

Trigger:	The user presses the logout button that appears on the screen.
Preconditions:	The user has successfully logged in to their account.
Postconditions:	
Normal Flow:	**12.0 Logging out** 1. Once the logout button is pressed, the user's access to their account is removed and they are returned to the login page.
Alternative Flows:	**12.1 Error logging out** 1. User is unable to log out due to network issues. The user checks their network connection and retry to log out. **12.2 Browser closed** 1. The user automatically logs out of the system if they close the browser or shut down their system.
Exceptions:	**12.0.E.1 Inactive on the system** 1. After twenty minutes of inactivity, the user is automatically logged out of the system.
Includes:	Successfully logged in to the system and user authentication while logging in.
Priority:	Medium
Frequency of Use:	Every time the user logs in.
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	If the user does not log out themselves, someone else could access their account on that device.

Use Case ID:	13.0		
Use Case Name:	***Sign Up***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students, Instructors, Departments
Description:	A student, instructor or department will be able to create an account on the system.
Trigger:	The user clicks on the sing up button that appears on the log in page.
Preconditions:	1.The user must be a student, instructor or department member that is part of the university database for the current year. 2. User must either be a new admittee for that year or a new employee in the university. 3. The employee should be given a sign-up option if their role demands access to the system.
Postconditions:	1. The sign-up information required like ID, name, role, batch year for the students and department should match the university's database.
Normal Flow:	**13.1 Sign-up** 1. Once the user clicks sign-up, there will be a list of options to select their role between student, instructor or department.

	2. Once selected, it will ask the user for their ID/role number or employee number, name, batch year for the students, university email address and department name. 3. These will be matched with the university's database. If the match is successful, it will allow the user to create their own unique login username and password. 4. The username of the user will be their role number or employee number/ID and password created will need to be strong.
Alternative Flows:	**13.2 Errors in sign up** 1. If the user enters an ID, name, email address, role or department that do not match the university's database, it will not allow the user to proceed with creating an account and will show an error. 2. If the password is under 8 characters or not strong, the system will give an error and ask the user to input the password again.
Exceptions:	**13.0.E.1 Details do not match the database** \ 1. The administration member trying to authenticate the account will not approve it unless the user adds the correct details that match the database. 2. The user will have to create an account again and get it reapproved.
Includes:	There will be manual authentication by an administration member to ensure that the account the user is trying to create is valid according to the database of the university.
Priority:	High
Frequency of Use:	Once for each user.
Business Rules:	None
Special Requirements:	None
Assumptions:	The user knows how to sign up for the system.
Notes and Issues:	None

Use Case ID:	14.0		
Use Case Name:	***Update Password***		
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actors:	Students, Instructors, Departments
Description:	Any user will be able to change their password provided they authenticate that they are the actual owner of the account.
Trigger:	The user clicks on the update password button in the settings.
Preconditions:	1. The user must be logged in. 2. The user must remember their current password
Postconditions:	1. The new password must not be shorter than 8 characters and should be strong. 2. The new password must not be the same as their old password.
Normal Flow:	**14.1 Update password** 1. The user clicks on the update password button in the settings and the system asks them to re-enter their current password.

	2. Once the old password entered is valid, the user will be prompted to enter a new password. The user will be asked to enter the new password one more time for authentication purposes. 3. The database will update their old password with the new one.
Alternative Flows:	**14.2 Incorrect old/current password** 1. The user enters an incorrect current password. The system will give an error and ask to try again 2. If the new password is under 8 characters, the system will give an error and ask the user to input the password again. 3. If the new password is the same as the old password, the system will give an error and ask the user to input the password again.
Exceptions:	**14.0.E.1 Frequency of password change** 1. The password cannot be changed more than once in a single day.
Includes:	Logging and authentication of the user.
Priority:	Low
Frequency of Use:	Depends on the user
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	The user may want to create a new password due to a security breach.

3. External Interface Requirements

• User Interfaces

- The users will use the login information provided by the department. One registered the students will receive an email using which they will need to verify their account. Once the account is created the user will login using their roll number and password.

- The user will be able to browse the course catalog and select the desired courses, we are assuming the student can see only courses for which pre-requisites are met and capacity is available. The student will be able to enroll in the course.

Fig 3.1.2: Course enrollment catalog

Standard buttons:
1. Back
2. Next

3. Help
4. Save

Screen layout constraints:
1. The buttons on the screen will be sized and shaped according to importance.
2. The resolution of the interface and screen will be in High definition to aid a user-friendly experience.

*Keyboard shortcuts used will be the general shortcuts utilized for websites and web pages and no extra features will be added.
**An error message will be generated if a command is entered that the web page does not recognize, and the user will be prompted to enter the correct information.

• Hardware Interfaces

The hardware used is the basic recommended configuration of the computer, and no external hardware is needed for this software. However, the minimum hardware requirements must be satisfied for this software to work properly:
1. Minimum 1GB RAM device
2. Minimum 2GB storage
3. 3.2GHz processor base frequency
4. 120GB Hard disk drives
5. Wi-Fi

• Software Interfaces

The operating systems used for this software will be:
1. Mac OS
2. Microsoft Windows
3. Linux

The databases used for this software will be:
1. MySQL
2. SQLite

The languages/software utilized will be:
1. HTML- Hyper Text Markup Language
2. JSON- JavaScript Object Notation
3. Python – Flask
4. React
5. Node js

*This list of software's in non-exhaustive, additions or subtractions will be made through the course of development.

• Communications Interfaces

The requirements of the internet transmission are:

1. Reliable network transmission
2. Web transfer
3. E-mail transfer

The communication protocols used will be:

1. Internet protocol mainly TCP - Transmission Control Protocol and Internet Control Message Protocol (ICMP)
2. HTTP- Hypertext Transfer Protocol
3. FTP- File Transfer Protocol
4. MIME- Multipurpose Internet Mail Extensions
5. Wi-Fi

4. Other Nonfunctional Requirements

• Performance Requirements

The system will require a great deal of processing speed since it must manage a database of 15000 records. There can be little to no lagging time for the system since there are going to be multiple users using the system simultaneously for various purposes. A lag in the processing of commands will result in an overall delay of the procedure since execution of one is directly linked to the other. The system should be able to open course catalogs and enroll courses without delay.

• Safety Requirements

The data or files transferred through the system will be ensured to having no loss of packets via transmission. The database will be ensured to having enough security that no third part can have access to it as a lot of personal information of the users will be in the database. The database cannot be accessed by the users themselves instead the users who will need the database will have to retrieve it from an authoritative person who can ensure that it is only available to required staff and not the students.
The users will have limited access to the database and the system depending on their role in the university.

• Security Requirements

Every user of the system cannot have access to everything. For example, the students cannot be given access to the database of the university because it can lead to privacy as well as security concerns. Several security requirements are:

1. Ensuring the encryption of passwords and other authentication features to avoid login attempts by hackers and third-party personnel.
2. Keeping the database secure so that only authenticated users have access to it.
3. The system will not allow students to modify their schedules or grades.
4. The instructors will not be allowed to modify other courses that they won't be teaching.
5. The departments will be the only ones who have full access of student's information.

- **Software Quality Attributes**

  - **Maintainability and portability**:
    The system will be designed in a way that the maintenance costs are reduced annually. The architectural implementation will ensure that if the system undergoes a fault, then the time required to recover from that is as less as possible. The system is designed in a way that it can be used on any operating system that supports internet and web applications. It can be used on phones, laptops and any portable device with an internet connection.

  - **Reliability, usability and availability:**
    The system will be available to the users 24 hours a week and it can be accessed at any time however the enrollment will be done on a given time. Reliability will be ensured by checking that no information is leaked or lost while data is being transmitted through the internet. The system will be kept up to date by the developers so that users can always have a user-friendly experience without lags and delays.

  - **Robustness:**
    When it comes to systems that have a large traffic and use, it is vital to ensure that any failures in the system are dealt with and the progress is saved automatically. This will ensure that when the user logs back in the work is still available and not lost during system crashes or failure. The system should have enough storage to accommodate the information from the database.

5. Other Requirements

Appendix A: Glossary

- **HTTP: Hypertext Transfer Protocol**
- **HTML: Hypertext Markup Language**
- **JSON: JavaScript Object Notation**
- **TCPL: Transmission Control Protocol**
- **ICMP: Internet Control Message Protocol**
- **FTP: File Transfer Protocol**
- **MIME: Multipurpose Internet Mail Extensions**

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>