SCHOOL OF BUSINESS SCIENCES

MILESTONE 1

Software Requirements Specification-Revised

OMNILERTPRIME

PRIME TECH THAT EMPOWERS YOU

PROJECT *#OMNiConnect*

PREPARED BY

GROUP 8

[redacted]

INTRODUCTION :

Purpose of the Document:

The purpose of this document is to provide a comprehensive overview of the software requirements for the Consultation Reservation System. It outlines the goals, objectives, and scope of the project, along with the limitations of the current approach/system. This document serves as a foundation for the development team to design, implement, and test the system effectively.

PROJECT SCOPE:

- The design and implementation of a self-contained system that enables students to schedule consultations with available university lecturers are included in the scope of the Consultation Reservation System project. By using the system, lecturers will be able to schedule consultations, organize several calendars, reserve equipment and spaces, and give consultations different statuses (such as upcoming/pending or completed). The system is a stand-alone system because it lacks network connectivity.
- Students, instructors, and administrators will be the three main user types for the system. Based on available time slots, the student user can schedule a meeting with a lecturer. The lecturer user will be able to observe and effectively manage their consultations.
- The administrator user will oversee managing users (adding, deleting, and updating them), overseeing the system's database, and producing reports using the data the system provides. The system will feature a Graphical User Interface and be user-friendly in design (GUI). As not all users are computer proficient, some users may need training on how to utilize the system. The technology will be implemented at the university, and it will be geared toward giving students individualized support.

GOALS & OBJECTIVES:

- Through one-on-one discussions with academics and instructors, this project seeks to enhance the academic support given to university students. The project's goals could consist of the following:

- To offer students individualized support based on their unique academic requirements and aspirations.
- To make clear the expectations and objectives of students for their classes and extracurricular activities.
- To give students fresh approaches, methods, and resources to help them succeed in school.
- To assist students in striking a balance between their academic obligations and other elements of their lives, such as their social lives, personal care, and mental health.
- To raise persistence and retention rates for students.
- To raise the level of student satisfaction with the university's academic assistance services.
- To compile information on the program's efficiency and utilize it to guide improvement.
- To develop an intuitive interface that makes it simple for academics and lecturers to schedule appointments and access resources.
- To provide lecturers with a comprehensive view of their students' academic backgrounds, strengths, areas for improvement, goals, and objectives.
- To give lecturers the ability to monitor students' progress and offer personalized support and guidance.
- To provide administrators with data analytics and reporting capabilities to assess the effectiveness of academic support programs and inform decision-making.

The project's goal is to assist student performance and well-being by attaining these goals, which will also help to improve the university experience.

Limitations of the Current Approach/System

- ***User interface and user experience:*** The software should have a user-friendly
- interface that is intuitive and easy to navigate for both students and lecturers.
- ***Security:*** The app deals with personal information- student and faculty details- it must have security measures in place to protect sensitive data from unauthorized access. Adhering to data protection regulations could be complex.
- ***Support and maintenance:*** Once the app is developed, ongoing technical support and maintenance need to be provided i.e., regular updates, bug fixes, and improvements will be necessary to ensure the app remains functional and meets user expectations.
- ***Fairness:*** Ensuring equal access and opportunities for all students can be a challenge in individual consultation programs.
- ***Large student population:*** Universities have a large number of students enrolled. Providing individual time with lecturers for each student can be logistically challenging, time consuming and unrealistic.

Requirements:

System requirements (Hardware + Software):

- Hardware:
  - ***Desktop :*** the most vital hardware element is the desktop since the system being built is a local desktop application and thus it will run on a desktop. There is no specific type of desktop required as the system will run on any desktop (given compatible operating system).
  - ***Network infrastructure:*** The system should require network devices such as routers, switches, and network cables to facilitate communication between all the desktops, and other connected devices.
- Software:
  - ***Operating system :*** again, the most vital element regarding software, as no activities will take place on the desktop without an operating system, the preferred operating system for this system is the Windows operating system.
  - ***Microsoft access (DBMS) :*** The system will use Microsoft Access as its database management system, it will be the one that facilitates all databases related to the system, e.g., a database for storing students and all relevant details related to each student i.e., student number, faculty, etc.

User requirements:

Student Requirements

- The user shall be able to create a unique profile.
- The student shall be able to search required lecturer on a search bar.
- The student shall be able to see open time slots for consulting from chosen lecturer.
- The student shall be able to book and manage a consultation.

Lecturer Requirements

- The user shall be able to create a unique profile.
- The lecturer shall be able to create and manage which time slots are available for them to be consulted.

Administrator Requirements

- Administrator shall be able to manage users and generate reports.

System Requirements

- The system should keep a database of all users for future logins.
- The system must have registration and login screen for users.
- The system should have a screen to display different faculties to choose from and a search bar so students can navigate to desired lecturer.
- The lecturer shall be notified when a time slot has been booked for consultation.
- The system should notify students should a booked consultation be made unavailable.
- The system should allow administrators to create, remove, update or archive users from their database.
- System should keep updated details about the user's profile.

Non-functional requirements:

- ***Compatibility :*** The desktop application must be built in C# and should run on Windows and MacOS operating systems.
- ***Scalability :*** The application/system must handle large number of requests at a time without compromising performance.
- ***Usability:*** The application/system must be user-friendly and visually appealing to the users' interfaces which are easy to use.
- ***Reliability :*** The system must be available for the whole day, for the duration of the working week and on Sundays as well to cater to users' needs without interruption. The system should have a robust infrastructure and mechanisms in place to minimize downtime and ensure continuous availability.
- ***Accuracy and Data Integrity :*** The system must always provide accurate information based on what the user prompted. It must also validate the user inputs and perform all necessary data checks and ensure data integrity throughout the system. Data accuracy and consistency are essential for the trustworthiness of the system.
- ***Fault Tolerance :*** System faults should be able to be fixed without having the system shutdown and hence the system should be designed with fault tolerance in mind.
- ***Security :*** The system must prioritize security to protect sensitive information. i.e., When the user is entering the password, the characters must be hidden, and the password stored should be encrypted on the database.
- ***Maintainability :*** The system should prioritize maintainability, enabling developers to easily understand, modify, and enhance it. This involves employing well-structured and modular code, adhering to coding best practices, and providing thorough documentation for future maintenance and updates.
- ***Performance:*** The system needs to demonstrate efficient performance by maintaining fast response times and minimal delays, even during peak times. It should be scalable.

Use Case Descriptions:

Student :

1) Use Case Description for profile creation.

Name of Use Case	Create Profile
Description of Use Case	A student creates a profile on the OMNiConnect system.
Actor(s)	Primary actor: • Student Secondary actor: • System (OMNiConnect)
Preconditions	The student is registered
Trigger	Student's request to create a profile
Basic flow of events	
System	Student
1. The System prompts Students to create a profile by entering relevant information.	• The student enters the following information: - Student number - First name - Last name - degree
	• Students submit profile information.
2. The System saves the profile and notifies the student that profile creation was successful.	
Alternate flow	
Cancel profile creation	1. The student selects the option to cancel profile creation. 2. The System returns the student to the homepage.
Invalid information entered	1. The student selects the create profile option after entering information prompted by the System. 2. The System displays an error message notifying the student of invalid information entered and prompts the student to re-enter the information

	3. Student re-enters their information.
Post conditions	The student's profile is created in the system.

2) Use Case Description for Consultation Booking

Name of Use Case	Book consultations
Description of Use Case	A Student books a consultation with a Lecturer.
Actor(s)	Primary actor: ● Student Secondary actor: ● System (OMNiConnect)
Preconditions	The Students and Lecturers have accounts on the System.
Trigger	Student's request to book a consultation
Basic flow of events	
System	Student
1. The System displays a list of available courses/modules.	- The student selects a course/module they need to consult for.
2. The System displays a list of available lecturers.	- The student selects a Lecturer from the list.
3. The system presents the available time slots for the selected lecturer.	- The student selects a time slot for the consultation.
4. The System checks for conflicts and verifies the availability of the selected time slot.	
5. The System confirms the booking and notifies both the Student and Lecturer.	
Alternate flow	
No available time slots	1. The System notifies the student that there are no available slots and suggests alternative time slots if available.

	2. The student can either select an alternative time slot or exit the application.
Post Conditions	The consultation is successfully scheduled between the student and the selected Lecturer.

Lecturers:

3) Use Case Description for Profile Creation

Name of Use Case	Create profile	
Description of Use Case	This will allow the lecturer to create their profile on the OMNIConnect system.	
Actor(s)	Lecturer , System	
Preconditions	The lecturer is a registered lecturer at that university.	
Trigger	Lecturer's request to create a profile	
Basic flow of events	System 1) The System prompts lecturer to create a profile by entering relevant information.	Lecturer The lecturer enters the following information: • staff number • first name • last name • title • modules they teach.
		• Lecturer submits profile information.
	2) The system saves the profile and notifies the lecturer that profile creation was successful	
Alternate flow	Cancel profile creation: 1. The lecturer selects the option to cancel profile creation. 2. The System returns the lecturer to the homepage. Invalid information entered: 1. The lecturer selects the create profile option after entering information prompted by the System.	

	2. The System displays an error message notifying the lecturer of invalid information entered and prompts the lecturer to re-enter the information. 3. Lecturer re-enters their information.
Post Conditions	The Lecturer's profile is created in the system.

4) Use Case Description for Timeslot Creation

Name of Use Case	Create time slots	
Description of Use Case	This will allow lecturers to define their available slots for consultations with students.	
Actor(s)	Lecturer , System	
Preconditions	The lecturer profile already exists in the system	
Trigger	Lecturer's request to create time slots	
Basic flow of events	System	Lecturer
	1) The system presents the lecturer with options to manage their time slots.	• The lecturer selects the option to create time slots.
	2) The system prompts the lecturer to enter the date, start time, and end time for a time slot.	• The lecturer provides the necessary information.
	3) The system checks for conflicts with existing time slots and adds the new time slot if no conflicts are found.	

Alternate flow	Cancel time slot: If the lecturer wants to cancel the selected time slot the system will cancel the selected time and allow the lecturer to select a new time slot.
Post Conditions	The lecturer's available time slots are successfully defined in the system.

Administrator :

4) Use Case Description for Updating a Lecturer's Information

Name of Use Case	Update Lecturer Information	
Description of Use Case	This use case allows the administrator to update the lecturer and student's details on the system	
Actor(s)	Administrator , System	
Preconditions	The administrator is logged into the system. The lecture is registered in the system.	
Trigger	Admin selects the option to update student or lecturer information.	
Basic flow of events	Admin	System
	1) The admin selects the option to updates lecturer information.	• The system displays a list of existing lecturers.

	2) The admin selects a lecturer to update	• The system presents the following fields for modification : ▪ name ▪ degree ▪ contact information ▪ or office details.
	3) The admin modifies the necessary fields.	• The system validates the updated information and saves the changes.
	4) The system notifies the admin of the successful update.	
Alternate flow	If there are validation errors in the updated information: • The system prompts the admin to correct errors.	
Post Conditions	The lecturer information is successfully updated in the system	

CRUD TABLE :

DOMAIN CLASS	VERIFIED USE CASES	RELATED USE CASES
STUDENT	Create Profile	
	View Consultation History/Details	View Consultation History (includes)
	Update Profile	View Profile (includes)
	Archive Student Profile	
LECTURER	Create Profile	
	View Consultation Schedule	View Consultation Schedule (includes)
	Update Profile	View Profile (includes)
	Archive Lecturer Profile	
ADMINISTRATION	Create Accounts/Profiles	
	View System Statistics	View Statistics (includes)
	Update Account Information	View Profile (includes)
	Archive accounts	
APPLICATION	Create new time slots	View Lecturer Profile (includes)

	View available time slots	
	Modify/Update existing time slots	View Lecturer's time slots (includes)
	Delete/Remove existing time slots	View Lecturer's time slots (includes)

Use Case Diagram