**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**UNIVERSITY TIMETABLE GENERATION SYSTEM**

**CSC392 Object Oriented Software Engineering**

Submitted on: <3/April/2023 >

Group Members:

BASIT IQBAL (FA21-BSE-050)

Fatima Aftab (FA21-BSE-088)

Essa Masood (FA21-BSE-086)

Ebadat Nissa (FA21-BSE-085)

Ahmed Tariq (FA21-BSE-048)

Waleed Rashid (FA21-BSE-162)

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 4](#_Toc131711987)

[Introduction 4](#_Toc131711988)

[Vision and Business Case 4](#_Toc131711989)

[Use-Case Model 5](#_Toc131711990)

[Use-Case Model 5](#_Toc131711991)

[1. Functional Requirements: 5](#_Toc131711992)

[2. Use Cases of System: 5](#_Toc131711993)

[3. 10% use cases: 6](#_Toc131711994)

[Supplementary Specification 6](#_Toc131711995)

[Glossary 7](#_Toc131711996)

[Risk List & Risk Management Plan 7](#_Toc131711997)

[CHAPTER 2 USE CASES 9](#_Toc131711998)

[Use Case Diagram 9](#_Toc131711999)

[Use Cases Distribution 10](#_Toc131712000)

[Brief Level Use Cases 11](#_Toc131712001)

[Eissa Masood (FA21-BSE-086) 11](#_Toc131712002)

[Use Case: Allocate Courses 11](#_Toc131712003)

[Eissa Masood (FA21-BSE-086) 11](#_Toc131712004)

[Use Case: Select Teacher 11](#_Toc131712005)

[Ebadat Nissa Khan (FA21-BSE-085) 11](#_Toc131712006)

[Use Case: View time table 11](#_Toc131712007)

[Ebadat Nissa Khan (FA21-BSE-085) 11](#_Toc131712008)

[Use Case: View time student table 11](#_Toc131712009)

[Ebadat Nissa Khan (FA21-BSE-085) 12](#_Toc131712010)

[Use Case: View teacher time table 12](#_Toc131712011)

[Ahmed Tariq (FA21-BSE-048) 12](#_Toc131712012)

[Use Case: Manage course 12](#_Toc131712013)

[Ahmed Tariq (FA21-BSE-048) 12](#_Toc131712014)

[Use Case: Add course 12](#_Toc131712015)

[Ahmed Tariq (FA21-BSE-048) 13](#_Toc131712016)

[Use Case: Manage rooms 13](#_Toc131712017)

[Waleed Rashid (FA21-BSE-162) 13](#_Toc131712018)

[Use Case: Manage slots 13](#_Toc131712019)

[Fully Dressed Use Cases 13](#_Toc131712020)

[Waleed Rashid(FA21-BSE-162) 13](#_Toc131712021)

[Use Case UC1: Manage slots 13](#_Toc131712022)

[Fully Dressed Use Cases 17](#_Toc131712023)

[Eissa Masood (FA21-BSE-086) 17](#_Toc131712024)

[Use Case UC1: Allocate Course 17](#_Toc131712025)

[Fully Dressed Use Cases 19](#_Toc131712026)

[Eissa Masood (FA21-BSE-086) 19](#_Toc131712027)

[Use Case UC1: Select Teacher 19](#_Toc131712028)

[Fully Dressed Use Cases 21](#_Toc131712029)

[Ahmed Tariq (FA21-BSE-048) 21](#_Toc131712030)

[Use Case UC1: Manage slots 21](#_Toc131712031)

[Fully Dressed Use Cases 23](#_Toc131712032)

[Ahmed Tariq (FA21-BSE-162) 23](#_Toc131712033)

[Use Case UC1: Add courses 23](#_Toc131712034)

[Fully Dressed Use Cases 25](#_Toc131712035)

[Ahmed Tariq (FA21-BSE-162) 25](#_Toc131712036)

[Use Case UC1: Manage rooms 25](#_Toc131712037)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction

As we know, one of the major challenges faced by universities is creating a timetable that meets the needs of both students and faculty. The current system of manual timetable generation is time-consuming, error-prone, and often results in scheduling conflicts that are difficult to resolve.

The proposed system will be a desktop application that automates the timetable generation process. The system will take into consideration the constraints and requirements of both students and faculty, such as slots, professor availability, room availability. The system will also be customizable, allowing for the inclusion of specific course preferences and restrictions.

The benefits of this proposed system are numerous. Firstly, it will eliminate the need for manual timetable generation, reducing the potential for errors and inconsistencies. Secondly, it will increase the efficiency of the timetable generation process, freeing up valuable time for administrators to focus on other important tasks. Lastly, it will improve the student **experience** by providing them with a timetable that is tailored to their specific needs, ensuring that they have access to the classes they need at times that are convenient for them.

## Vision and Business Case

Vision

The main objective of the Timetable Management System is to manage the details of Timetable, Faculty, Subject, Student, Course. It manages all the information about Timetable, Semester, Course, Timetable. The project is totally built at administrative end and thus only the administrator is guaranteed access.

Business cases:

Cost savings: The implementation of a University Timetable Management System can save the university significant costs by automating the scheduling process and reducing the need for manual labor. By streamlining the scheduling process, the university can reduce administrative costs, increase operational efficiency, and allocate resources more effectively.

Improved student retention: A University Timetable Management System can contribute to improved student retention rates by providing students with an easy-to-use platform to view their schedules and adjust as needed. By giving students more control over their schedules, they are more likely to stay enrolled in courses and avoid conflicts that could lead to dropping out.

Enhanced student satisfaction: A University Timetable Management System can contribute to enhanced student satisfaction by providing them with a more user-friendly scheduling experience.

## Use-Case Model

## Functional Requirements:

* **Admin Requirements:**
* The only admin can select, remove, create, and update slot of timetable.
* The only admin can add, select, update, and remove rooms for timetable.
* The only admin can allocate courses to teachers to teach.
* The only admin can select courses and select teachers for semester.
* The only admin can view the timetable of both teachers and students.
* The only admin can request to system to generate timetable.
* **Teacher Requirements:**
* Teachers can only view their timetables.
* Teachers can give preferences for courses to be taught.
* **Student Requirements:**
* Students can only view their timetable.
* **System Requirements:**
* The University Timetable Generation System allows login and logout functionality to admins , students, and teachers.
* The University Timetable Generation System allows registration of new admins.
* The University Timetable Generation System generates timetables for admins, students, and teachers.
* The University Timetable Management System only accepts valid login details to enroll admins, students, and teachers in system.

## Supplementary Specification

**Introduction:**

The Timetable Management System is an application used by universities to manage the scheduling and allocation of courses and teaching resources. This Supplementary Specification outlines additional features and requirements for the system.

**User Interface:**

The user interface should be user-friendly and easy to navigate, with clear and concise instructions. It should allow for easy modifications to the assigned courses and teachers. The interface should also display the status of the courses and any potential conflicts with the assigned teachers.

**Performance:**

The system should be able to handle a large number of courses and teachers, and it should be able to generate and display the updated timetable in a timely manner. It should also be able to handle multiple users accessing the system simultaneously.

**Security:**

The system should have appropriate security measures in place to protect user data and prevent unauthorized access. User authentication and authorization should be implemented, and all data should be encrypted.

**Reports:**

The system should be able to generate various reports, including course and teacher schedules, conflicts, and availability. The reports should be easy to read and should allow for easy modifications to the schedules.

**Integrations:**

The system should be able to integrate with other university systems, including student information systems and learning management systems. The integration should allow for seamless transfer of data and information between the systems.

**Accessibility:**

The system should be accessible to all users, including those with disabilities. The interface should be designed with accessibility in mind, and the system should adhere to accessibility standards and guidelines.

**Support and Maintenance:**

The system should come with comprehensive support and maintenance services, including training for users and technical support. The system should be regularly updated and maintained to ensure optimal performance and usability.

**Performance Metrics:**

The system should be evaluated on various performance metrics, including response time, uptime, and user satisfaction. The system should be regularly tested and evaluated to ensure that it meets the required performance standards.

**Conclusion:**

The Supplementary Specification outlines additional features and requirements for the Timetable Management System. The system should be designed and developed with these specifications in mind to ensure that it meets the needs and requirements of the university.

## Glossary

***Key domain terminology, and data dictionary.***

7.8. NextGen Example: A (Partial) Glossary

## Risk List & Risk Management Plan

* **Risk List:**

The risk list is a comprehensive document that identifies all possible risks associated with the University Timetable management system. The following are some examples of potential risks:

**Technical Risks:**

This risk category includes issues related to the system's technology, such as software and hardware failures, compatibility issues, system crashes, and data loss.

**Personnel Risks:**

This category includes risks associated with the project team, such as staff turnover, lack of expertise, and poor communication.

**Time Risks:**

This category includes risks associated with project timelines and deadlines, such as delays in the development and implementation phases.

**Resource Risks:**

This category includes risks associated with the availability and allocation of project resources, such as insufficient funding, lack of equipment, and limited staff.

**Operational Risks:**

This category includes risks associated with the day-to-day operations of the system, such as user errors, security breaches, and system downtime.

**External Risks:**

This category includes risks associated with external factors that may affect the project, such as changes in government regulations, market fluctuations, and natural disasters.

* **Risk Management Plan:**

The risk management plan is a detailed document that outlines the strategies and actions that will be taken to mitigate and manage potential risks. The following are some examples of risk management strategies and actions that can be taken for each risk category:

**Technical Risks:**

To manage technical risks, the project team can implement robust software testing procedures, regular backups of data, and contingency plans in case of system failures.

**Personnel Risks:**

To manage personnel risks, the project team can ensure that the right people with the required skills and expertise are recruited, establish effective communication channels, and provide training and development opportunities.

**Time Risks:**

To manage time risks, the project team can develop realistic timelines, track progress regularly, and prioritize tasks based on criticality.

**Resource Risks:**

To manage resource risks, the project team can identify potential resource constraints early on, establish contingency plans, and seek additional funding and resources if required.

**Operational Risks:**

To manage operational risks, the project team can establish user access controls, implement robust security measures, and develop a disaster recovery plan.

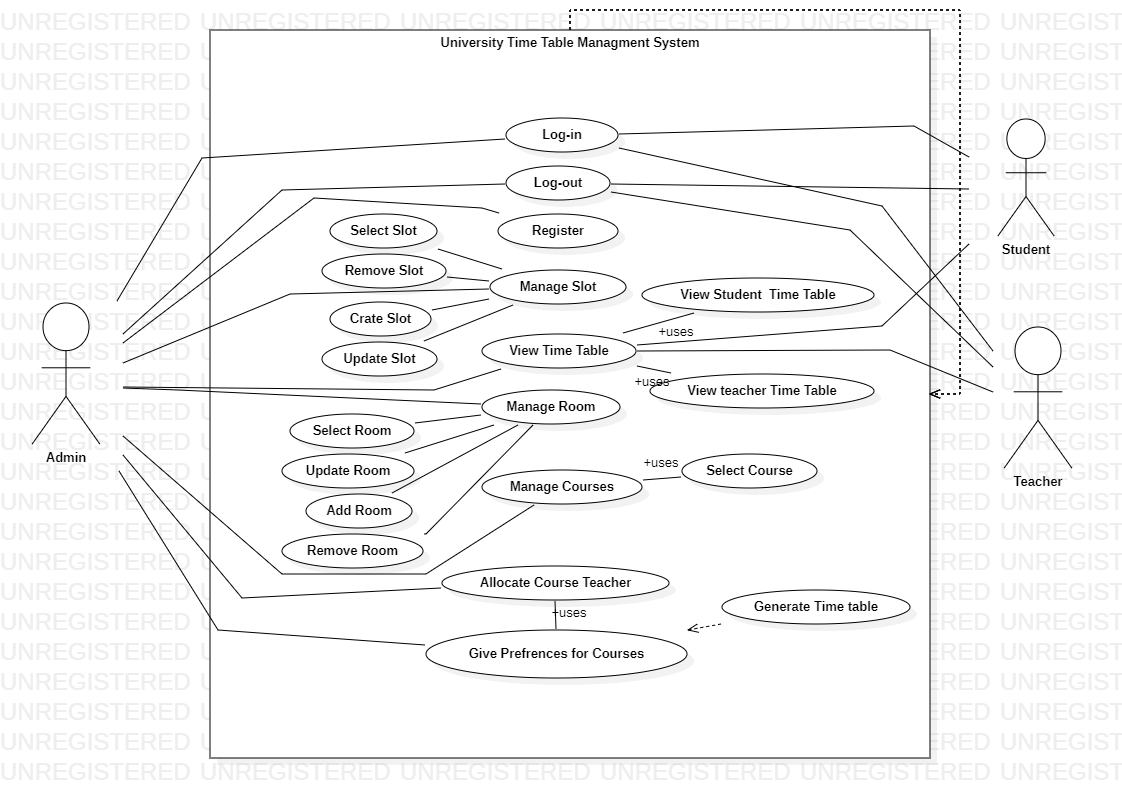
**External Risks:**

To manage external risks, the project team can monitor the external environment regularly, establish contingency plans, and develop alternative strategies to respond to unexpected events.

In summary, developing a risk list and risk management plan is an essential step in ensuring the success of the University Timetable management system project. By identifying potential risks and implementing appropriate risk management strategies and actions, the project team can minimize the impact of potential risks and ensure that the project runs smoothly and delivers the desired outcomes.

# CHAPTER 2 USE CASES

## Use Case Diagram



## 

## Use Cases Distribution

|  |  |  |
| --- | --- | --- |
| S#. | Group Member | Assigned Use Cases |
| 1 | <BASIT IQBAL>  <FA21-BSE-050> | UC 1 : Login  UC 2 :Set Constrains  UC 3: Logout |
| 2 | <Fatima Aftab>  <FA21-BSE-088> | UC 4: Generate timetable  UC 5: Modify Timetable |
| 3 | <Waleed Rashid>  <FA21-BSE-162> | UC 6 : Set slots.  UC 7 : Add slots.  UC 8 : View slots |
| 4 | <Ahmed Tariq>  <FA21-BSE-048> | UC 9 : Set semester’s courses.  UC 10: Send notification.  UC 11 : Set criteria |
| 5 | <Eissa Masood>  <FA21-BSE-086> | UC 12 : Allocate courses.  UC 13: select rooms.  UC 14: select courses. |
| 6 | <Ebadat Nissa Khan>  <FA21-BSE-085> | UC 15 : View Timetable  UC 16: View Teachers Timetable  UC 17: View Timetable by Student |

## 

## Brief Level Use Cases

### Eissa Masood (FA21-BSE-086)

#### Use Case: Allocate Courses

The Timetable Management System for a university is a crucial tool for allocating courses to the timetable for each academic term. The primary actor in this use case is the System Admin, who logs into the system and selects the academic term for which courses need to be allocated. The system displays the available courses, and the admin selects the courses to be allocated. The system checks for scheduling conflicts and suggests alternative options if necessary. The admin reviews and approves the suggested options, and the system assigns the courses to the timetable. If modifications are needed, the admin can modify the allocated courses by selecting the desired scheduling options. The updated timetable is generated and displayed in the system, and the courses are now allocated for the selected academic term.

### Eissa Masood (FA21-BSE-086)

#### Use Case: Select Teacher

The "Select Teacher" use case in a Timetable Management System allows admin to select teachers to teach the allocated courses. Once the courses have been allocated to the timetable, the admin can assign the courses to a teacher based on their availability and expertise in the subject area. The system displays a list of available teachers, along with their qualifications and availability for the selected time slots. The admin selects the desired teacher and assigns them to the course. The system updates the timetable accordingly, and the assigned teacher is notified of their teaching assignment. This use case ensures that each course is assigned to a qualified and available teacher, and it allows for efficient and effective management of the university's teaching resources.

### Ebadat Nissa Khan (FA21-BSE-085)

#### Use Case: View time table

The "View Timetable" use case is a fundamental feature of the Timetable Management System for University. This use case enables students and teachers to view their class schedules for the current week or multiple weeks in advance. The system displays the timetable information accurately and in a user-friendly format, including the course code, course name, class start and end time, and class location. The success of this use case guarantees that students and teachers can plan their schedules, accordingly, attend their classes on time, and avoid any scheduling conflicts. The View Timetable use case occurs multiple times per week for each student and teacher, and it is essential to the success of the Timetable Management System.

### Ebadat Nissa Khan (FA21-BSE-085)

#### Use Case: View time student table

The "View Timetable by Student" is a use case within the Timetable Management System for University, which allows students to access their class schedules. This use case is of high importance, as it ensures that students have a clear understanding of their class schedules and can attend their classes on time. The use case occurs at the user goal level, meaning it is a significant task that students need to accomplish to achieve their academic goals. The primary actor for this use case is the student, and the stakeholders are the faculty and university administration. The student must have an active account in the system and have already enrolled in the classes they wish to view on the timetable. The success of this use case guarantees that students can view their timetable with accurate information and plan their schedules accordingly. It occurs multiple times per week for each student and is critical to the success of the Time Table Management System.

### Ebadat Nissa Khan (FA21-BSE-085)

#### Use Case: View teacher time table

The "View Timetable by Teacher" is a use case within the Time Table Management System for University, which allows teachers to access their class schedules. This use case is of high importance, as it ensures that teachers have a clear understanding of their class schedules and can attend their classes on time. The use case occurs at the user goal level, meaning it is a significant task that teachers need to accomplish to achieve their academic goals. The primary actor for this use case is the teacher, and the stakeholders are the faculty and university administration. The teacher must have an active account in the system and have already been assigned to the classes they wish to view on the timetable. The success of this use case guarantees that teachers can view their timetable with accurate information and plan their schedules accordingly.

### Ahmed Tariq (FA21-BSE-048)

#### Use Case: Manage course

The "Manage course" use case of a University timetable management system involves the management of course-related information, such as creating, modifying, and deleting course details, assigning instructors to teach the courses, and scheduling classes. The objective is to provide an easy and efficient way for authorized personnel, such as academic coordinators and administrative staff, to manage the university's course catalog, course scheduling, and instructor assignments.

### Ahmed Tariq (FA21-BSE-048)

#### Use Case: Add course

The "Add course" use case of a University timetable management system involves the creation of a new course in the system. The main objective of this use case is to provide an easy and efficient way for academic coordinators or administrative staff to add new courses to the university's course catalog. The process starts with the user entering the course details, such as course code, title, description, prerequisites, and credit hours. The system should ensure that the course information is accurate and complies with the university's policies and regulations. Once the user submits the course details, the system should create a new course in the catalog and assign a unique course ID to the course. The system should also allow the user to assign an instructor to the course and specify the class schedule, such as the start and end dates, class times, and location. The system should ensure that the assigned instructor and class schedule do not conflict with any other courses or events. After the course is added, the system should update the course catalog and display the new course information to authorized users. The course can then be further modified or scheduled as needed using other use cases such as "Manage course" and "Schedule class"

### Ahmed Tariq (FA21-BSE-048)

#### Use Case: Manage rooms

The "Manage Rooms" use case of a University timetable management system involves the management of rooms or facilities available for scheduling classes, events, and meetings. The main objective of this use case is to provide an easy and efficient way for authorized personnel, such as administrative staff, to manage the university's room inventory and availability. The process starts with the user viewing the list of available rooms in the system. The system should display the room details, such as the room number, capacity, and availability. The user can then add new rooms, modify existing rooms, or delete rooms as needed. The user can also assign a room to a specific course or event schedule. The system should display a calendar with available time slots and allow the user to select the preferred time and location. The system should also ensure that there are no conflicts with other scheduled classes or events. The system should keep track of the room assignments and availability and provide alerts if there are any conflicts or double bookings. The user can also generate reports to view room usage and availability for a specific time period. In summary, the "Manage Rooms" use case plays a crucial role in the University timetable management system, enabling administrative staff to manage the university's room inventory and availability, and ensuring that classes and events are scheduled in appropriate locations and times.

### Waleed Rashid (FA21-BSE-162)

#### Use Case: Manage slots

|  |
| --- |

The academic administrator can add a new slot to the timetable by selecting a course or exam, assigning it to an available room, and choosing a suitable time slot. The academic administrator can modify an existing slot by changing the room or time slot, or by reassigning the course or exam to another faculty member. The academic administrator can cancel a slot if the course or exam is no longer required or needs to be rescheduled. The system should automatically update the timetable and notify all stakeholders of the change. The academic administrator can view all available slots, including those already assigned and those that are still open. The system should provide filters and search options to help the administrator locate specific slots.

### Fatima Aftab (FA21-BSE-088)

#### Use Case: Generate Timetable

The "generate timetable" use case involves the generation of a schedule for a specific time based on predefined criteria such as course offerings, class size, available classrooms, slots, and teacher availability. The Admin selects the criteria and inputs the necessary data, such as course names, class sizes, slots, and teacher schedules, into the system. The system then uses an algorithm to generate a timetable that meets all the specified criteria, while also considering any constraints, such as avoiding scheduling conflicts or ensuring that certain courses are offered at specific times. Once the timetable has been generated, the Admin can review and modify it as necessary before finalizing it. The generated timetable can then be printed or exported for distribution to students, and teachers.

### Fatima Aftab (FA21-BSE-088)

#### Use Case: Modify Timetable

"Modify timetable" is a critical use case for a university timetable generation system. It involves making changes to the schedule to adapt to various circumstances, such as course offerings, student preferences, and resource utilization. This use case is important to avoid scheduling conflicts and ensure that students have access to the courses they need. By modifying the system, universities can create a schedule that meets the needs of students and faculty, maximizes the use of resources, and provides the best possible learning experience. Ultimately, a flexible and adaptable timetable generation system is crucial to the success of any academic institution.

## Fully Dressed Use Cases

### BASIT IQBAL (FA21-BSE-050)

| Use Case UC1: Login |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Login |
| Scope | University Timetable Generation System |
| Level | User Goal |
| Primary Actor. | Admin, student, teacher |
| Stakeholders and interest | * System Admin: Wants to efficiently login to the system to use the services being offered by the system. * Faculty Members: Wants to efficiently login to the system to use the services being offered by the system. * Students: Wants to efficiently login to the system to use the services being offered by the system. |
| Preconditions | The User must have a valid username and password (i.e Registration must have been completed). |
| Success Guarantee | The user is logged in to the system and can perform actions within it |
| Main Success Scenario | 1. The user navigates to the login page 2. The user enters their username and password 3. The system verifies the user's credentials. 4. If the credentials are correct, the system grants access to the system and the user can perform actions within it. |
| Exceptions | 1. If the user enters an incorrect username or password, the system displays an error message and prompts the user to try again. 2. If the user forgets their password, they can click on the "Forgot Password" link and follow the password recovery process. |
| Special Requirements | * The Timetable Management System should be user-friendly and easy to navigate. * The system should provide a clear overview of the available courses and their scheduling options. * The system should allow for easy modifications to the allocated courses. |
| Technology and data  variation list. | * The Timetable Management System can be accessed from a web browser or a dedicated application. * The system should be compatible with various operating systems and devices. * The list of available courses may vary depending on the academic term and the university's course offerings. |
| Frequency of occurrences | This use case occurs at the beginning when the user navigates to the website of the university timetable generation system or open the desktop application |
| Miscellaneous | * The Timetable Management System should be regularly updated and maintained to ensure optimal performance and usability. * The system should have appropriate security measures in place to protect user data and prevent unauthorized access. |

### BASIT IQBAL (FA21-BSE-050)

| Use Case UC2: Log out |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Log Out |
| Scope | University Time Table Generation System |
| Level | User Goal |
| Primary Actor. | Admin, student, teacher |
| Stakeholders and interest | * System Admin: Wants to logout off the system after using the services and wants privacy and security of the account. * Faculty Members: Wants to logout off the system after using the services and wants privacy and security of the account. * Students: Wants to logout off the system after using the services and wants privacy and security of the account. |
| Preconditions | * The user is currently logged in to the system. * The user has access to the internet. |
| Success Guarantee | * The user is logged out of the system. * The user is redirected to the login screen. |
| Main Success Scenario | * The user clicks on the "Log out" button or link in the user interface. * The system logs the user out by invalidating their session token or cookie. * The system redirects the user to the login screen. |
| Exceptions | 1. Automatic log out due to session timeout:   If the user's session has been inactive for a certain period of time, the system automatically logs the user out and invalidates their session token or cookie.  The system displays a message to the user indicating that they have been logged out due to inactivity and redirects them to the login screen.   1. Error during log out process:   If an error occurs during the log out process, such as the system being unable to invalidate the session token or cookie, the system displays an error message to the user and asks them to try again or contact support.   1. User is already logged out:   If the user has already been logged out, either by clicking the "Log out" button or automatically due to a session timeout, the system displays a message to the user indicating that they are already logged out and redirects them to the login screen. |
| Special Requirements | * The Timetable Management System should be user-friendly and easy to navigate. * The system should provide a clear overview of the available courses and their scheduling options. * The system should allow for easy modifications to the allocated courses. |
| Technology and data  variation list. | * The Timetable Management System can be accessed from a web browser or a dedicated application. * The system should be compatible with various operating systems and devices. * The list of available courses may vary depending on the academic term and the university's course offerings. |
| Frequency of occurrences | This use case occurs at the last when the user navigates to the website of the university timetable generation system or open the desktop application and has performed all the actions of his choice and wants to leave the system by clicking on logout. |
| Miscellaneous | * The Timetable Management System should be regularly updated and maintained to ensure optimal performance and usability. * The system should have appropriate security measures in place to protect user data and prevent unauthorized access. |

### Waleed Rashid(FA21-BSE-162)

| Use Case UC1: Manage slots |
| --- |

|  |  |
| --- | --- |
| Use Case Name | Manage Slot in Timetable Management System for University |
| Scope | Timetable Management System for the university |
| Level | User Goal Level |
| Primary Actor | Academic Administrator |
| Stakeholders and Interests | **Faculty**: Interested in having their classes scheduled at appropriate times and rooms  **Students**: Interested in having their classes scheduled without any conflicts and preferences considered  **Room** Administrators: Interested in ensuring that rooms are utilized efficiently and not overbooked.  **University** Administration: Interested in efficient scheduling to maximize resource utilization and minimize conflicts |
| Precondition | The administrator is authenticated and authorized to access the Timetable Management System, and the timetable is set up with the available courses, faculty, rooms, and other resources |
| Success Guarantee | Successful scheduling of classes, labs, and exams without any conflicts or overbooking of resources |
| Main Success Scenarios | The administrator selects the course or exam for which the slot needs to be scheduled.  The system presents the available time slots for the selected course/exam and checks for any conflicts.  The administrator selects the appropriate time slot and assigns it to the course/exam.  The system updates the timetable and notifies the faculty and students of the new schedule. |
| Exceptions | If there are no available time slots, the system alerts the administrator and suggests alternative solutions.  If there is a conflict with a previously scheduled course or exam, the system alerts the administrator and suggests alternative time slots or adjustments to the existing schedule.  if there are not enough resources (such as rooms or faculty) to schedule a course or exam, the system alerts the administrator and suggests alternative solutions. |
| Special Requirements | The system should be able to handle multiple constraints such as avoiding time conflicts, ensuring the availability of required equipment, and accommodating student preferences as much as possible. |
| Technology and Data Variation List | The system should be able to integrate with the university's existing information systems and databases, including course catalogs, faculty information, and room availability data. |
| Frequency of Occurrences | Multiple times per academic term, typically during the course scheduling and exam scheduling periods. |
| Prototype | A prototype of the system should be developed and tested with input from faculty, students, and administrators to ensure it meets their needs and expectations. |
| Miscellaneous | This use case is critical for the efficient operation of the university and can have a significant impact on the academic success of faculty and students. Therefore, it should be designed and implemented with care and attention to detail. |
| Use Case Section | Time Table Management |
| Primary Actors | Academic Administrator |
| Secondary Actors | Faculty, Students, Room Administrators, University Administration |
| Preconditions | The administrator is authenticated and authorized to access the Time Table Management System, and the timetable is set up with the available courses, faculty, rooms, and other resources |
| Post conditions | The system updates the timetable and notifies the faculty and students of the new schedule. |
| Related Use Cases | Manage Course, Manage Exam, Manage Room, Manage Faculty |
| Special Requirements | The system should be able to handle multiple constraints such as avoiding time conflicts, ensuring the availability of required equipment, and accommodating student preferences as much as possible. |

### Eissa Masood (FA21-BSE-086)

| Use Case UC1: Allocate Course |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Allocate Courses. |
| Scope | This use case describes the process of allocating courses to the timetable for a new academic term in the Timetable Management System. |
| level | User Goal |
| Primary Actor. | Admin |
| Stakeholders and interest | * System Admin: Wants to efficiently allocate courses to the timetable for each academic term. * Faculty Members: Want to ensure that their courses are allocated to the timetable at a convenient time and day. * Students: Want a well-structured and organized timetable that minimizes scheduling conflicts and allows for an optimal learning experience. |
| Preconditions | The Timetable Management System is operational.  The Admin has the necessary login credentials to access the system.  The list of available courses for the academic term is available in the system. |
| Success Guarantee | The courses are allocated to the timetable , and the updated timetable is generated and displayed in the system |
| Main Success Scenario | 1. The admin logs into the Timetable Management System. 2. The admin selects the academic term for which courses need to be allocated. 3. The system displays the list of available courses for the selected term. 4. The admin selects the courses that need to be allocated to the timetable. 5. The system checks for any scheduling conflicts with previously allocated courses. 6. If there are no conflicts, the system assigns the selected courses to the timetable. 7. If there are conflicts, the system suggests alternative scheduling options. 8. The admin approves the suggested scheduling options. 9. The system assigns the courses to the timetable according to the approved scheduling options. 10. The system generates and displays the updated timetable. |
| Exceptions | 1. If the Timetable Management System is not operational, the use case cannot proceed. 2. If the admin does not have the necessary login credentials, they cannot access the system. 3. If the list of available courses for the academic term is not available in the system, the admin cannot allocate courses to the timetable. |
| Special Requirements | * The Timetable Management System should be user-friendly and easy to navigate. * The system should provide a clear overview of the available courses and their scheduling options. * The system should allow for easy modifications to the allocated courses. |
| Technology and data  variation list. | * The Timetable Management System can be accessed from a web browser or a dedicated application. * The system should be compatible with various operating systems and devices. * The list of available courses may vary depending on the academic term and the university's course offerings. |
| Frequency of occurrences | This use case occurs at the beginning of each academic term, typically once a year. |
| Miscellaneous | * The Timetable Management System should be regularly updated and maintained to ensure optimal performance and usability. * The system should have appropriate security measures in place to protect user data and prevent unauthorized access. |

### Eissa Masood (FA21-BSE-086)

| Use Case UC1: Select Teacher |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Select Teacher. |
| Scope | This use case describes the process of selecting and assigning a teacher to an allocated course in the Timetable Management System. |
| level | User Goal |
| Primary Actor. | Admin |
| Stakeholders and interest | * Admin: Wants to efficiently assign qualified and available teachers to the allocated courses. * Faculty Members: Want to ensure that the courses they are teaching are properly assigned to them in the timetable. * Students: Want to have a well-structured and organized timetable with qualified teachers teaching their courses. |
| Preconditions | * The Timetable Management System is operational. * The admin has the necessary login credentials to access the system. * The courses have been allocated to the timetable. |
| Success Guarantee | The selected teacher is assigned to the allocated course, and the updated timetable is generated and displayed in the system. |
| Main Success Scenario | 1. The admin logs into the Timetable Management System. 2. The admin selects the academic term and the allocated course for which a teacher needs to be selected. 3. The system displays a list of available teachers for the selected time slot and subject area. 4. The admin selects the desired teacher from the list. 5. The system checks for any scheduling conflicts with the teacher's existing assignments. 6. If there are no conflicts, the system assigns the teacher to the course. 7. If there are conflicts, the system suggests alternative scheduling options. 8. The admin approves the suggested scheduling options. 9. The system assigns the teacher to the course according to the approved scheduling options. 10. The system generates and displays the updated timetable. |
| Exceptions | 1. if the Timetable Management System is not operational, the use case cannot proceed. 2. If the admin does not have the necessary login credentials, they cannot access the system. 3. If the courses have not been allocated to the timetable, a teacher cannot be selected for the course. 4. If there are no available teachers for the selected time slot and subject area, the use case cannot proceed. |
| Special Requirements | * The Timetable Management System should provide a clear overview of the available teachers and their qualifications and availability. * The system should allow for easy modifications to the assigned teachers. * The system should have appropriate security measures in place to protect user data and prevent unauthorized access. |
| Technology and data  variation list. | * The Timetable Management System can be accessed from a web browser or a dedicated application. * The system should be compatible with various operating systems and devices. * The list of available teachers may vary depending on their availability and expertise in the subject area. |
| Frequency of occurrences | This use case occurs whenever a new academic term starts or when there is a change in the teaching assignments. |
| Miscellaneous | * The Timetable Management System should be regularly updated and maintained to ensure optimal performance and usability. * The system should be able to handle multiple teacher assignments for different courses and time slots. * The system should be able to send notifications to the assigned teacher regarding their teaching assignment. |

### Ahmed Tariq (FA21-BSE-048)

| Use Case UC1: Manage slots |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Manage Courses |
| Scope | This use case involves managing courses offered by the University within the Timetable Management System. |
| Main Success Scenarios | 1. The University Administrator creates a new course, specifying the course name, course code, prerequisites, credit hours, and the professor who will teach the course.  2. The University Administrator updates an existing course, changing information such as course name, prerequisites, or professor.  3. The University Administrator deletes a course from the system.  4. The University Administrator assigns a course to a classroom, specifying the time and day(s) of the week the course will be offered. |
| Level | User goal |
| Primary Actor | University Administrator |
| Stakeholders | Professors, Students, University Administration, |
| Preconditions | - The University Administrator is logged in to the Timetable Management System.  - The relevant course information (e.g., course name, course code, prerequisites, credit hours) is available. |
| Success Guarantee | The Timetable Management System accurately reflects the course offerings for the University. |
| Exceptions | 1. The course code already exists in the system, and the University Administrator must choose a different code. 2. The professor assigned to the course is not available during the specified time and day(s) of the week, and the Administrator must choose a different professor. 3. The classroom assigned to the course is not available during the specified time and day(s) of the week, and the Administrator must choose a different classroom. |
| Special Requirements | - The TimeTable Management System must be able to generate accurate course schedules based on the courses assigned to each classroom. |
|  |  |
| Technology and Data Variations List | - The TimeTable Management System should be accessible through a web interface.  - The course information must be stored in a database. |
| Frequency of Occurrences | This use case will be performed frequently, as courses are added, updated, and deleted each semester. |
|  |  |
| Miscellaneous | The University Administrator may also need to assign multiple professors to a course if it is a team-taught course. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Ahmed Tariq (FA21-BSE-162)

| Use Case UC1: Add courses |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Add Courses |
| Scope | This use case involves adding new courses to the TimeTable Management System for the University. It includes specifying course details such as course name, course code, prerequisites, credit hours, and assigning the course to a professor and classroom. |
| Main Success Scenarios | The University Administrator enters the course name, course code, prerequisites, credit hours, and other details into the TimeTable Management System.  The Administrator assigns the course to a professor who is qualified to teach the course.  The Administrator assigns the course to a classroom, specifying the time and day(s) of the week the course will be offered. |
| Level | User goal |
| Primary Actor | University Administrator |
| Stakeholders | Professors, Students, University Administration, |
| Preconditions | The University Administrator is logged in to the TimeTable Management System.  The relevant course information (e.g., course name, course code, prerequisites, credit hours) is available. |
| Success Guarantee | The new course is accurately reflected in the TimeTable Management System and can be assigned to a professor and classroom. |
| Exceptions | The course code already exists in the system, and the University Administrator must choose a different code.  The professor assigned to the course is not available during the specified time and day(s) of the week, and the Administrator must choose a different professor.  The classroom assigned to the course is not available during the specified time and day(s) of the week, and the Administrator must choose a different classroom. |
| Special Requirements | The TimeTable Management System must be able to generate accurate course schedules based on the courses assigned to each classroom. |
|  |  |
| Technology and Data Variations List | The TimeTable Management System should be accessible through a web interface.  The course information must be stored in a database. |
| Frequency of Occurrences | This use case will be performed frequently as new courses are added each semester. |
|  |  |
| Miscellaneous | The University Administrator may also need to assign multiple professors to a course if it is a team-taught course.  The course may have multiple sections that need to be assigned to different professors and classrooms. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### 

### Ahmed Tariq (FA21-BSE-162)

| Use Case UC1: Manage rooms |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Manage Rooms |
| Scope | This use case involves managing classrooms in the TimeTable Management System for the University. It includes adding new classrooms, modifying existing ones, and removing classrooms. |
| Main Success Scenarios | The University Administrator adds a new classroom to the TimeTable Management System, specifying the room number, capacity, and any special equipment available in the classroom.  The Administrator modifies an existing classroom by updating its information (e.g., capacity, equipment).  The Administrator removes a classroom that is no longer available for use. |
| Level | User goal |
| Primary Actor | University Administrator |
| Stakeholders | Professors, Students, University Administration, |
| Preconditions | - The University Administrator is logged in to the TimeTable Management System.  The relevant classroom information (e.g., room number, capacity, equipment) is available.  Success Guarantee: The classroom information is accurately reflected in the TimeTable Management System, and the classroom can be assigned to courses as needed. |
| Success Guarantee | classroom information is accurately reflected in the TimeTable Management System, and the classroom can be assigned to courses as needed. |
| Exceptions | The room number already exists in the system, and the University Administrator must choose a different room number.  The Administrator attempts to remove a classroom that is currently assigned to a course, and the system prompts them to reassign the course to a different classroom before removing the original classroom. |
| Special Requirements | The TimeTable Management System must be able to generate accurate course schedules based on the classrooms available. |
|  |  |
| Technology and Data Variations List | The TimeTable Management System should be accessible through a web interface.  The classroom information must be stored in a database. |
| Frequency of Occurrences | This use case will be performed frequently as new classrooms are added or modified, and old classrooms are removed or replaced. |
|  |  |
| Miscellaneous | The University Administrator may also need to assign multiple classrooms to a course if it has multiple sections or requires specialized equipment.  Prototype: A prototype of the system can be created to test the usability of the interface and the accuracy of the classroom scheduling algorithm. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### FATIMA AFTAB (FA21-BSE-088)

| Use Case UC4: Generate Timetable |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Generate Timetable |
| Scope | University timetable Generation System |
| Main Success Scenarios | * The university administrator successfully generates a timetable that meets the requirements and constraints specified by the system. * The generated timetable is error-free and includes all necessary information, such as class times, room assignments, and instructor assignments. * Instructors and students can access their respective schedules and view the correct information. * The generated timetable is easily modifiable, allowing the administrator to make changes as necessary. |
| Level | User goal |
| Primary Actor | University Administrator |
| Stakeholders | * University administrator: Wants to generate an accurate and efficient timetable for the university to ensure that students and faculty can attend classes without conflicts. * Faculty: Need to know when and where their classes are scheduled. * Students: Need to know when and where their classes are scheduled. |
| Preconditions | * university administrator should login to the system. * The system has been set up with the necessary data, including course schedules, instructor availability, room availability, and other constraints. * The system has assigned courses to faculty members and recorded their preferences. |
| Success Guarantee | The system generates a timetable that meets the expectations and requirements of all stakeholders. |
| Exceptions | * If the system is unable to generate a conflict-free timetable, it should display an error message and prompt the Admin to modify the scheduling algorithm or parameters. * If there is a system error while generating the timetable, the system should display an error message and prompt the admin to try again later. * If there is a major change in faculty availability or course enrollment after the timetable has been generated, the admin may need to modify the schedule manually. |
| Special Requirements | The Timetable Management System must be able to generate accurate course schedules based on the classrooms available. |
|  |  |
| Technology and Data Variations List | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Web languages (html, CSS, react) * Course information may be updated between the time the timetable is generated and the start of the semester. * Faculty availability may change due to illness, scheduling conflicts, or other unforeseen circumstances. * Classroom availability may change due to maintenance or other scheduling conflicts |
| Frequency of Occurrences | * Once per semester for each university department. * If timetable is not accurate university administrator can also modify it. |
|  |  |
| Miscellaneous | * The system should be able to handle large amounts of data efficiently. * The system should be user-friendly and easy to navigate. * The system should be able to generate a timetable within a reasonable amount of time. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### FATIMA AFTAB (FA21-BSE-088)

| Use Case UC5: Give preferences for the courses |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Give preferences for the courses |
| Scope | University Timetable Generation System |
| Level | User Goal |
| Primary Actor. | Faculty members |
| Stakeholders and interest | * Faculty Members: Want to provide their preferences for when they can teach their assigned courses. * University administrator: Needs accurate and up-to-date information on faculty preferences to generate an efficient timetable. |
| Preconditions | * Faculty members must login to the system. * The system must have an updated form of courses. |
| Success Guarantee | * The system records the faculty member's preferences for when they can teach their assigned courses. |
| Main Success Scenario | * The Faculty Member logs into the system. * The system displays the courses assigned to the Faculty Member for the selected semester. * The Faculty Member submits their preferences. * The system updates the faculty member's preferences in the database. |
| Exceptions | * If there is a system error while submitting preferences, the system should display an error message and prompt the Faculty Member to try again later. * If faculty select the course of another department, then system should ignore that. |
| Special Requirements | * The system should be user-friendly to users. |
| Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Web languages (html,css,react) * Faculty members may need to update their preferences due to personal or professional conflicts. * Course assignments may change due to enrollment or faculty availability changes. |
| Frequency of occurrences | Once per semester for each faculty member. |
| Miscellaneous | * The system should be able to handle many faculty members and their preferences efficiently. * The system should be user-friendly and easy to navigate. * The system should be able to validate the preferences entered by faculty members to ensure they are within acceptable ranges. |

LOGIN FORM:

REGISTRATION FORM:

