**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 7](#_Toc134394775)

[Introduction 7](#_Toc134394776)

[Vision and Business Case 7](#_Toc134394777)

[Use-Case Model 8](#_Toc134394778)

[1. Functional Requirements: 8](#_Toc134394779)

[Supplementary Specification 8](#_Toc134394780)

[Glossary 10](#_Toc134394781)

[Risk List & Risk Management Plan 10](#_Toc134394782)

[CHAPTER 2 USE CASES 11](#_Toc134394802)

[2.1 Use Case Diagram 11](#_Toc134394803)

[2.2 Use Cases Distribution 12](#_Toc134394804)

[2.3 Brief Level Use Cases 13](#_Toc134394805)

[Eissa Masood (FA21-BSE-086) 13](#_Toc134394806)

[Use Case: Allocate Courses 13](#_Toc134394807)

[Eissa Masood (FA21-BSE-086) 13](#_Toc134394808)

[Use Case: Select Teacher 13](#_Toc134394809)

[Ebadat Nissa Khan (FA21-BSE-085) 13](#_Toc134394810)

[Use Case: View time table 13](#_Toc134394811)

[Ebadat Nissa Khan (FA21-BSE-085) 13](#_Toc134394812)

[Use Case: View time student table 13](#_Toc134394813)

[Ebadat Nissa Khan (FA21-BSE-085) 14](#_Toc134394814)

[Use Case: View teacher time table 14](#_Toc134394815)

[Ahmed Tariq (FA21-BSE-048) 14](#_Toc134394816)

[Use Case: Set semester courses. 14](#_Toc134394817)

[Ahmed Tariq (FA21-BSE-048) 15](#_Toc134394822)

[Use Case: Set Criteria 15](#_Toc134394823)

[Ahmed Tariq (FA21-BSE-048) 16](#_Toc134394824)

[Use Case: Set Constrains 16](#_Toc134394825)

[Waleed Rashid (FA21-BSE-162) 16](#_Toc134394829)

[Use Case: Manage slots. 16](#_Toc134394830)

[Fatima Aftab (FA21-BSE-088) 16](#_Toc134394831)

[Use Case: Generate Timetable 16](#_Toc134394832)

[Fatima Aftab (FA21-BSE-088) 16](#_Toc134394833)

[Use Case: Modify Timetable 16](#_Toc134394834)

[BASIT IQBAL (FA21-BSE-050) 17](#_Toc134394835)

[Use Case: Login 17](#_Toc134394836)

[BASIT IQBAL (FA21-BSE-050) 17](#_Toc134394837)

[Use Case: Logout 17](#_Toc134394838)

[2.4 Fully Dressed Use Cases 18](#_Toc134394839)

[BASIT IQBAL (FA21-BSE-050) 18](#_Toc134394840)

[Use Case UC1: Login 18](#_Toc134394841)

[BASIT IQBAL (FA21-BSE-050) 19](#_Toc134394842)

[Use Case UC2: Log out 19](#_Toc134394843)

[Waleed Rashid(FA21-BSE-162) 21](#_Toc134394844)

[Use Case UC1: Manage slots 21](#_Toc134394845)

[Eissa Masood (FA21-BSE-086) 24](#_Toc134394846)

[Use Case UC1: Allocate Course 24](#_Toc134394847)

[Eissa Masood (FA21-BSE-086) 26](#_Toc134394848)

[Use Case UC1: Select Teacher 26](#_Toc134394849)

[Ahmed Tariq (FA21-BSE-048) 28](#_Toc134394850)

[Use Case UC1: Manage slots 28](#_Toc134394851)

[Ahmed Tariq (FA21-BSE-162) 30](#_Toc134394852)

[Use Case UC1: Add courses 30](#_Toc134394853)

[Ahmed Tariq (FA21-BSE-162) 32](#_Toc134394854)

[Use Case UC1: Manage rooms 32](#_Toc134394855)

[FATIMA AFTAB (FA21-BSE-088) 34](#_Toc134394856)

[Use Case UC4: Generate Timetable 34](#_Toc134394857)

[FATIMA AFTAB (FA21-BSE-088) 36](#_Toc134394858)

[Use Case UC5: Modify timetable 36](#_Toc134394859)

[WALEED RASHID (FA21-BSE-162) 39](#_Toc134394860)

[Use Case UC5: Manage Slots 39](#_Toc134394861)

[2.4 Prototypes 41](#_Toc134394862)

[2.4.1. BASIT IQBAL 41](#_Toc134394863)

[2.4.2. Fatima Aftab 42](#_Toc134394864)

[Ebadat Nisa Khan- View timetable by student,View timetable by teacher. 43](#_Toc134394865)

[CHAPTER 3 DOMAIN MODEL 3.1 Individual Partial Domain Model 44](#_Toc134394866)

[UC 1,2 : Login & Logout (BASIT IQBAL ,FA21-BSE-050) 44](#_Toc134394867)

[UC 3,4 : Generate Timetable and Modify timetable (Fatima Aftab,Fa21-bse-088) 45](#_Toc134394868)

[CHAPTER 4 System Sequence Diagram (SSD) 48](#_Toc134394869)

[4.1. LOGIN (BASIT IQBAL : FA21-BSE-050) 48](#_Toc134394870)

[4.2. Logout (BASIT IQBAL : FA21-BSE-050) 49](#_Toc134394871)

[4.3. Generate Timetable Scenario (Fatima Aftab) 50](#_Toc134394872)

[4.4. Modify Timetable Scenario (Fatima Aftab) 51](#_Toc134394873)

[51](#_Toc134394874)

[4.5. Manage Slots (Waleed Rashid) 52](#_Toc134394875)

[CHAPTER 5 Operation Contracts (OC’s) 56](#_Toc134394876)

[5.1. Login (BASIT IQBAL : FA21-BSE-050) 56](#_Toc134394877)

[5.2. Logout (BASIT IQBAL: FA21-BSE-050) 56](#_Toc134394878)

[3.1. Add slots (waleed rashid : FA21-BSE-162) 57](#_Toc134394879)

[View slots (waleed rashid : FA21-BSE-162) 58](#_Toc134394880)

[Select slots (waleed rashid : FA21-BSE-162) 59](#_Toc134394881)

# 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 the 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 many 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:

# Data loss due to system failure or user error.

# Inadequate system security leading to unauthorized access or data breaches.

# Incorrect data input by users.

# Insufficient capacity to handle large volumes of data.

# Software bugs or errors leading to system crashes or malfunctions.

# Inability to handle unexpected changes in scheduling or events.

# Resistance from users to adopt the new system or change their habits.

# Lack of resources (time, budget, personnel) to implement or maintain the system.

# Lack of integration with other systems, such as student information systems or other systems.

# Risk Management Plan:

# Regular backups and disaster recovery procedures should be implemented to mitigate the risk of data loss.

# Access controls, firewalls, and encryption should be used to ensure system security and protect against unauthorized access or data breaches.

# Data validation rules and error-checking mechanisms should be implemented to prevent incorrect data input by users.

# The system should be designed to handle anticipated volumes of data and scalable to accommodate future growth.

# Software testing and quality assurance processes should be in place to identify and fix bugs and errors before they cause significant problems.

# The system should be flexible and adaptable to handle unexpected changes in scheduling or events.

# User training and communication should be conducted to encourage user adoption and facilitate the transition to the new system.

# Adequate resources should be allocated to implement and maintain the system.

# CHAPTER 2 USE CASES

## 2.1 Use Case Diagram



## 

## 2.2 Use Cases Distribution

|  |  |  |
| --- | --- | --- |
| S#. | Group Member | Assigned Use Cases |
| 1 | <BASIT IQBAL>  <FA21-BSE-050> | UC 1 : Login  UC 2 : Logout |
| 2 | <Fatima Aftab>  <FA21-BSE-088> | UC 3 : Generate timetable.  UC 4 : Modify Timetable  UC 5 : Manage Courses |
| 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 Constrains.  UC 10 : Set criteria |
| 5 | <Eissa Masood>  <FA21-BSE-086> | UC 11 : Allocate courses.  UC 12 : select rooms.  UC 13 : select teachers. |
| 6 | <Ebadat Nissa Khan>  <FA21-BSE-085> | UC 14 : View Timetable  UC 15 : View Teachers Timetable  UC 16 : View Timetable by Student |

## 2.3 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 Timetable 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 Timetable 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: Set Criteria

In a timetable management system, set criteria are used to determine how the schedule is generated. These criteria can be defined based on various factors, such as student preferences, teacher workload, room availability, and overall program requirements.

Some common set criteria that might be used in a timetable management system include:

1. Student preferences: This criterion considers the preferences of individual students when scheduling their classes. For example, some students may prefer to have their classes scheduled in the morning, while others may prefer afternoon or evening classes.
2. Teacher workload: This criterion ensures that each teacher is assigned a workload that is appropriate for their level of experience and expertise. For example, new teachers may be assigned fewer classes than experienced teachers.
3. Room availability: This criterion ensures that each class is scheduled in an available classroom that is appropriate for the class size and equipment needs. For example, a class that requires a computer lab may be scheduled in a room with the necessary equipment.
4. Program requirements: This criterion ensures that the schedule meets the overall requirements of the academic program. For example, a program may require that certain classes be taken in a specific order, or that certain classes be completed before graduation.

By defining and applying set criteria in a timetable management system, administrators can generate schedules that are efficient, effective, and meet the needs of all stakeholders involved. Set criteria help to ensure that resources are utilized effectively, scheduling conflicts are minimized, and optimal learning outcomes are achieved for all students.

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

#### Use Case: Set Constrains

#### In a timetable management system, set constraints are used to ensure that the schedule meets certain conditions or requirements. These constraints can take various forms, including course availability, classroom availability, teacher availability, student availability, and room capacity.

#### Course availability constraints ensure that each course is scheduled during specific days and times. Classroom availability constraints ensure that each class is assigned to an available classroom. Teacher availability constraints ensure that each teacher is available to teach during the scheduled time slot. Student availability constraints ensure that each student can attend all their registered classes without conflicts. Room capacity constraints ensure that each class is assigned to a classroom with sufficient capacity.

#### By incorporating these and other set constraints, a timetable management system can generate a schedule that is efficient, effective, and meets the needs of all stakeholders involved, including students, teachers, and administrators. Set constraints help to prevent scheduling conflicts, ensure that resources are utilized effectively, and promote optimal learning outcomes for all students.

### 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 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.

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

#### Use Case: Set semester courses.

### In a timetable management system, set semester courses are the courses that are offered during a specific semester or academic term. These courses can be categorized based on various factors such as course code, course name, and number of credits.

### Set semester courses are important for a timetable management system as they form the basis for creating the class schedule for that semester. The system needs to ensure that all courses offered in a particular semester are scheduled in such a way that there are no conflicts in the course schedules.

### To manage semester courses effectively, a timetable management system may use different techniques such as clustering courses based on similarity, grouping courses based on the availability of the required resources such as classrooms and lab equipment, and scheduling courses based on student demand and faculty availability.

### By effectively managing set semester courses in a timetable management system, administrators can ensure that students are able to complete the required courses within the specified time frame and are able to graduate on time. Moreover, a well-managed course schedule can reduce scheduling conflicts, maximize the use of resources, and improve the overall efficiency of the academic program.

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

#### Use Case: Login

The Login use case for the University Timetable Generation System allows users, including admins, teachers, and students, to efficiently log in to the system using their valid username and password. Upon navigating to the login page, the user enters their credentials, and the system verifies their identity. If the credentials are correct, the user gains access to the system, enabling them to perform actions within it. In case of any exceptions, such as incorrect login details, the system displays an error message and prompts the user to try again.

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

#### Use Case: Logout

The Log Out use case in the University Timetable Generation System allows the Admin, student, and teacher to log out of the system to ensure the privacy and security of their accounts. The user must be currently logged in to the system, and have access to the internet. The success guarantee of this use case is that the user is logged out of the system and redirected to the login screen. The main success scenario involves the user clicking on the "Log out" button or link in the user interface, the system logging the user out by invalidating their session token or cookie, and redirecting the user to the login screen. Exceptions include automatic log out due to session timeout, error during the log out process, and user already being logged out. The Timetable Management System should be user-friendly and regularly updated, and appropriate security measures should be in place to protect user data and prevent unauthorized access.

## 2.4 Fully Dressed Use Cases

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

| Use Case UC: 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. * Teachers: 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 UC: Log out |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Log Out |
| Scope | University Timetable 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. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BASIT IQBAL (FA21-BSE-050)Use Case UC: Mange Slots (View Slots)  |  |  | | --- | --- | | Use Case Section | Comments | | Use Case Name | View Slots | | Scope | University Timetable | | Level | User Goal | | Primary Actor | University Administrator | | Stakeholders and interest | * University Administrator : A person who is responsible for managing slots in University, can view slots. | | Preconditions | * User authentication : The user must be authenticated and authorized as an administrator with the necessary permissions to make changes to the timetable. * Slots Existence: slots must be existed in database. | | Success Guarantee | * Admin can view slots from database. | | Main Success Scenario | * The administrator selects the "Mange Slots" option from the system menu. * The administrator selects the view slots option. * The system displays the list of slots from the database. | | Exceptions | * The slots don’t exist in database. * The user is not connected to database. | | Special Requirements | * Usability: The system should have a user-friendly interface that is easy for administrators to navigate and view the courses. * Security: The system should ensure that only authorized administrators with the appropriate permissions can view the courses. * Reliability: The system should be reliable and available for administrators to always use, with minimal downtime for maintenance or updates. | | Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Java | | Frequency of occurrences | It depends upon the administrator that how many times he/she wants to view slots. | | 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 display course list correctly. | |
|  |

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

#### Use Case UC: Mange Slots (Delete Slots)

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Delete Slots |
| Scope | University Timetable |
| Level | User Goal |
| Primary Actor | University Administrator |
| Stakeholders and interest | * University Administrator : A person who is responsible for managing slots in University, can delete slots. |
| Preconditions | * User authentication : The user must be authenticated and authorized as an administrator with the necessary permissions to make changes to the timetable. * Slots Existence: Slots must be existed in database. |
| Success Guarantee | * Admin can delete slots from database. |
| Main Success Scenario | * The administrator selects the "Manage Slots" option from the system menu. * The system displays the list of slots from the database. * The administrator selects the delete slots option. * Admin selects the slots to be deleted. * Admin save the changes. |
| Exceptions | * The slots don’t exist in database. * The user is not connected to database. |
| Special Requirements | * Usability: The system should have a user-friendly interface that is easy for administrators to navigate and view the courses. * Security: The system should ensure that only authorized administrators with the appropriate permissions can view the courses. * Reliability: The system should be reliable and available for administrators to always use, with minimal downtime for maintenance or updates. |
| Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Java |
| Frequency of occurrences | It depends upon the administrator that how many times he/she wants to delete slots |
| 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 display course list correctly. |

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

#### Use Case UC: Mange Slots (Add Slots)

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Add Slots |
| Scope | University Timetable |
| Level | User Goal |
| Primary Actor | University Administrator |
| Stakeholders and interest | * University Administrator : A person who is responsible for managing slots in University, can add slots. |
| Preconditions | * User authentication : The user must be authenticated and authorized as an administrator with the necessary permissions to make changes to the timetable. |
| Success Guarantee | * Admin can add slots in database. |
| Main Success Scenario | * The administrator selects the "Manage Slots" option from the system menu. * The system displays the list of slots from the database. * The administrator selects the add slots option. * Admin enter the slots to be added. * Admin save the changes. |
| Exceptions | * The insertion of slots can be fail. * The user is not connected to database. |
| Special Requirements | * Usability: The system should have a user-friendly interface that is easy for administrators to navigate and view the courses. * Security: The system should ensure that only authorized administrators with the appropriate permissions can view the courses. * Reliability: The system should be reliable and available for administrators to always use, with minimal downtime for maintenance or updates. |
| Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Java |
| Frequency of occurrences | It depends upon the administrator that how many times he/she wants to add slots |
| 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 display course list correctly. |

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

| Use Case UC: Generate Timetable |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Generate Timetable |
| Scope | University timetable |
| Main Success Scenarios | 1. The administrator selects the "Generate Timetable" option from the main menu. 2. The system displays a form where the administrator can enter the details of the timetable, such as the start and end date, the number of courses, the number of classrooms, and the number of students. 3. The administrator fills out the form and submits it to the system. 4. The system checks the validity of the input and generates a list of available time slots based on the input parameters. 5. The system uses an algorithm to generate a conflict-free timetable that allocates the available time slots to the courses and classrooms. 6. The system displays the generated timetable to the administrator. 7. The administrator reviews the timetable and verifies that it meets the requirements of the faculty and students. 8. If the timetable is satisfactory, the administrator approves it and saves it in the system. 9. If the timetable is not satisfactory, the administrator can modify and re-generate the timetable. |
| 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, faculty 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 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 UC: Modify timetable |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Modify timetable |
| Scope | University Timetable |
| Level | User Goal |
| Primary Actor. | University Administrator |
| Stakeholders and interest | * University Administrator : A person who is responsible for managing and wants to make changes to the timetable in a quick and efficient manner. |
| Preconditions | * User authentication : The user must be authenticated and authorized as an administrator with the necessary permissions to make changes to the timetable. * Timetable existence : The timetable must already exist in the system for the administrator to modify it. |
| Success Guarantee | * The changes made by the administrator are accurately reflected in the timetable. * The timetable for teachers, and students are updated to reflect the changes made in it. |
| Main Success Scenario | * The administrator selects the "Modify Timetable" option from the system menu. * The administrator selects the course, teacher, or classroom they wish to modify from the available options. * The administrator modifies the schedule information for the selected course, teacher, or classroom as necessary. * The system updates the timetable to reflect the changes made by the administrator. * Faculty and students can access the updated schedules and any changes made to the timetable. |
| Exceptions | * The timetable the administrator is attempting to modify does not exist in the system. * The course, faculty, or classroom the administrator is attempting to modify does not exist in the system. * The changes made by the administrator result in a conflict with an existing schedule for a course, instructor, or classroom. * The administrator inputs invalid or incomplete schedule information, such as an incorrect time or room number. * A system error occurs during the modification process and prevents the changes from being correctly updated in the system. |
| Special Requirements | * Usability: The system should have a user-friendly interface that is easy for administrators to navigate and use to modify the timetable. * Security: The system should ensure that only authorized administrators with the appropriate permissions can modify the timetable, and that user data is securely stored and protected from unauthorized access or tampering. * Reliability: The system should be reliable and available for administrators to always use, with minimal downtime for maintenance or updates. |
| Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Web technologies (html,css,react) * Java for GUI |
| Frequency of occurrences | It depends upon the administrator that how many times he/she wants to modify timetable after the timetable has been generated. |
| 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 modify timetable correctly and update timetable for teachers, and students. |

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

| Use Case UC: Manage Courses(View Courses) |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | View Courses |
| Scope | University Timetable |
| Level | User Goal |
| Primary Actor | University Administrator |
| Stakeholders and interest | * University Administrator : A person who is responsible for managing courses in University can view courses. |
| Preconditions | * User authentication : The user must be authenticated and authorized as an administrator with the necessary permissions to make changes to the timetable. * Course Existence: Courses must be existed in database. |
| Success Guarantee | * Admin can view courses from database. |
| Main Success Scenario | * The administrator selects the "Manage Courses" option from the system menu. * The administrator selects the view Courses option. * The system displays the list of courses from the database. |
| Exceptions | * The Courses don’t exist in database. * The user is not connected to database. |
| Special Requirements | * Usability: The system should have a user-friendly interface that is easy for administrators to navigate and view the courses. * Security: The system should ensure that only authorized administrators with the appropriate permissions can view the courses. * Reliability: The system should be reliable and available for administrators to always use, with minimal downtime for maintenance or updates. |
| Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Java |
| Frequency of occurrences | It depends upon the administrator that how many times he/she wants to view courses. |
| 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 display course list correctly. |

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

| Use Case UC: Manage Courses(Delete Courses) |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Delete Courses |
| Scope | University Timetable |
| Level | User Goal |
| Primary Actor | University Administrator |
| Stakeholders and interest | * University Administrator : A person who is responsible for managing courses in University can delete courses. |
| Preconditions | * User authentication : The user must be authenticated and authorized as an administrator with the necessary permissions to make changes to the timetable. * Course Existence: Courses must be existed in database. |
| Success Guarantee | * Admin can delete courses from database. |
| Main Success Scenario | * The administrator selects the "Manage Courses" option from the system menu. * The administrator selects the delete Courses option. * The admin selects the courses to be deleted. * Admin saves the changes. |
| Exceptions | * The Courses don’t exist in database. * The user is not connected to database. |
| Special Requirements | * Usability: The system should have a user-friendly interface that is easy for administrators to navigate and view the courses. * Security: The system should ensure that only authorized administrators with the appropriate permissions can view the courses. * Reliability: The system should be reliable and available for administrators to always use, with minimal downtime for maintenance or updates. |
| Technology and data  variation list. | * Object-Oriented Programming * Database Management Systems * Object Oriented Software Engineering * Java |
| Frequency of occurrences | It depends upon the administrator that how many times he/she wants to delete courses. |
| 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 display course list correctly. |

### 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: Set Criteria |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Set Criteria |
| Scope | This use case involves setting criteria in the TimeTable Management System for the University. |
| Main Success Scenarios | 1. Academic Staff selects the "Set Criteria" option from the system menu. 2. System displays the current criteria for generating conflict-free timetables. 3. Academic Staff modifies the criteria according to their preferences. 4. System saves the new criteria and updates the system's conflict-free timetable generation algorithm. 5. System notifies the Academic Staff that the criteria have been successfully set. |
| Level | User goal |
| Primary Actor | University Administrator |
| Stakeholders | Professors, Students, University Administration, |
| Preconditions | * Academic Staff must be authenticated and have the necessary permissions to set criteria. * Course data for the semester must be available in the system. |
| Success Guarantee | Criteria for generating conflict-free timetables is successfully set. |
| Exceptions | 1. Academic Staff selects an invalid criteria value. 2. System encounters an error while updating the conflict-free timetable generation algorithm. |
| Special Requirements | None |
|  |  |
| Technology and Data Variations List | * System must have access to the current conflict-free timetable generation algorithm and criteria. * System must be able to update the conflict-free timetable generation algorithm with the new criteria. |
| Frequency of Occurrences | As needed. |
|  |  |
| Miscellaneous | None. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

| Use Case UC1: Set Constrains |
| --- |

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Set Constrains |
| Scope | This use case involves setting constrains in the TimeTable Management System for the University. |
| Main Success Scenarios | 1. Academic Staff selects the "Set Constraints" option from the system menu. 2. System displays the current constraints for generating conflict-free timetables. 3. Academic Staff modifies the constraints according to their preferences. 4. System saves the new constraints and updates the system's conflict-free timetable generation algorithm. 5. System notifies the Academic Staff that the constraints have been successfully set. |
| Level | User goal |
| Primary Actor | University Administrator |
| Stakeholders | Professors, Students, University Administration, |
| Preconditions | * Academic Staff must be authenticated and have the necessary permissions to set constraints. * Course data for the semester must be available in the system. |
| Success Guarantee | Constraints for generating conflict-free timetables is successfully set. |
| Exceptions | 1. Academic Staff selects an invalid constraint value. 2. System encounters an error while updating the conflict-free timetable generation algorithm. |
| Special Requirements | None. |
|  |  |
| Technology and Data Variations List | * System must have access to the current conflict-free timetable generation algorithm and constraints. * System must be able to update the conflict-free timetable generation algorithm with the new constraints. |
| Frequency of Occurrences | As needed. |
|  |  |
| Miscellaneous | None. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

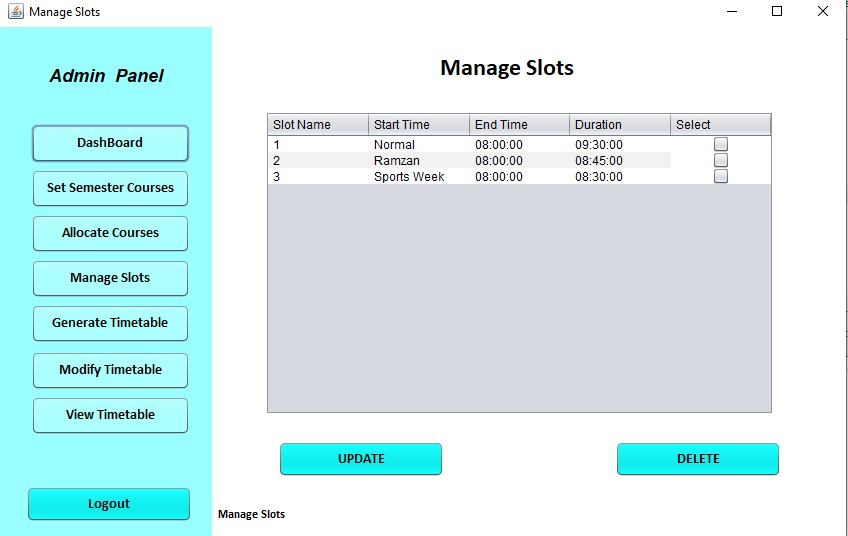
### WALEED RASHID (FA21-BSE-162)

| Use Case UC5: Manage Slots |
| --- |

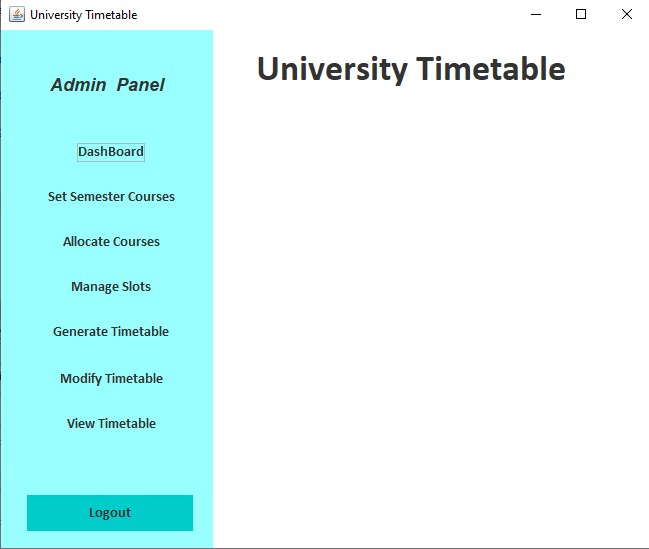
|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Manage Slots |
| Scope | University Timetable Generation System |
| Level | User Goal |
| Primary Actor. | Admin, student, teacher |
| Stakeholders and interest | * System Admin: Interested in managing the course slots efficiently and ensuring the best use of resources.. * Teachers: Interested in having clear slots for their courses and avoiding scheduling conflicts. * Students: Interested in having a clear and organized timetable for their courses. |
| Preconditions | * The Admin must be logged into the TimeTable Management System. * The system must have access to the course details, instructor information, and available resources such as classrooms and equipment. |
| Success Guarantee | The system will successfully update the timetable with the new slot information and notify the relevant stakeholders. |
| Main Success Scenario | 1. The Admin selects the option to manage slots. 2. The system displays the list of courses with their existing slots. 3. The Course Coordinator selects a course to modify. 4. The system displays the details of the selected course, including the existing slots and their corresponding instructors and resources. 5. The Course Coordinator modifies the slots by adding, deleting or changing the time, location or instructor. 6. The system validates the modifications and checks for any conflicts with existing slots or resources. 7. If no conflicts are found, the system updates the timetable with the new slot information and notifies the relevant stakeholders. |
| Exceptions | 1. I f there are conflicts with existing slots or resources, the system alerts the Course Coordinator and provides possible solutions. 2. If there are technical issues or errors, the system displays an error message and prompts the Course Coordinator to try again later or contact technical support. |
| Special Requirements | * The system should allow the Course Coordinator to view the entire timetable or filter by course, instructor, time, or location. * The system should have a user-friendly interface with clear instructions and feedback messages. * Technology and Data Variations List * The system should be accessible from any device with a web browser and internet connection. * The system should be able to handle different types of course data, instructor data, and resource data formats. |
| 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 | The Manage Slots use case can be performed multiple times per semester, as needed. |
| Miscellaneous | The Manage Slots use case is a critical function of the TimeTable Management System for University, as it enables efficient scheduling of courses, instructors, and resources. |

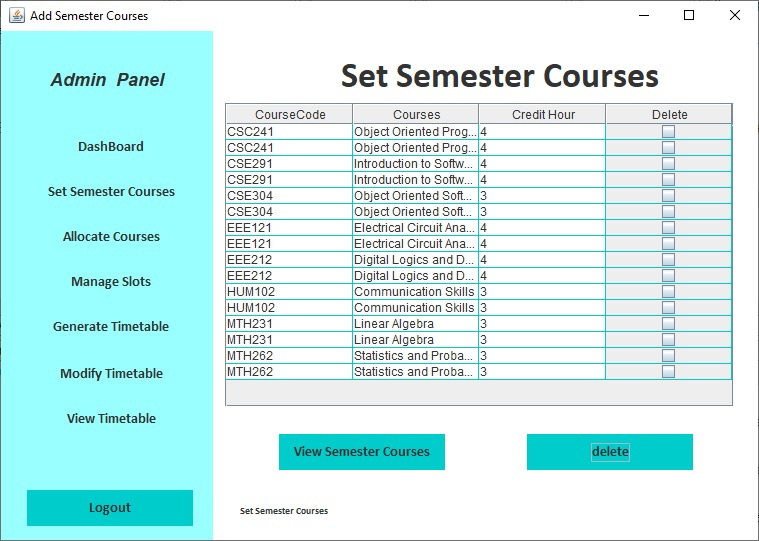
## Prototypes

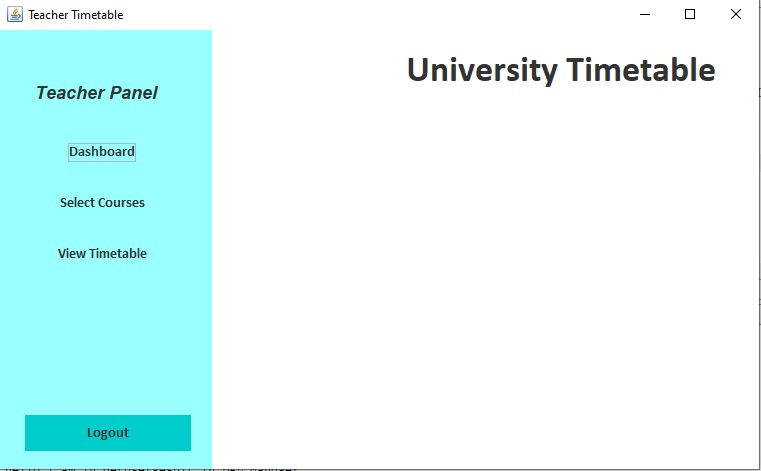
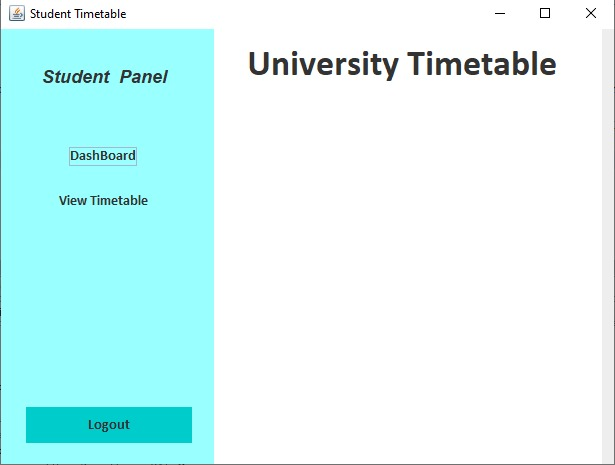
### BASIT IQBAL



### Fatima Aftab







# CHAPTER 3 DOMAIN MODEL 3.1 Individual Partial Domain Model

### UC 1,2 : Login & Logout (BASIT IQBAL ,FA21-BSE-050)



### UC 3,4 : Generate Timetable and Modify timetable (Fatima Aftab,Fa21-bse-088)

**View Timetable(Ebadat Nisa Khan, FA21-BSE-085)**



### Integrated Domain Model



# CHAPTER 4 System Sequence Diagram (SSD)

### 4.1. LOGIN (BASIT IQBAL : FA21-BSE-050)



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



### Text Description automatically generatedGenerate Timetable Scenario (Fatima Aftab)

### Modify Timetable Scenario (Fatima Aftab)

### Text Description automatically generated

### 4.5. Manage Courses(view Courses) (Fatima Aftab)

### A screenshot of a computer Description automatically generated4.6. Manage Slots(view Slots) (Basit Iqbal)

## 4.7. Manage Slots(Delete Slots) (Basit Iqbal)

## A diagram of a person Description automatically generated

## 4.8. Manage Slots(Add Slots) (Basit Iqbal)

A screenshot of a computer

Description automatically generated

## 4.9. Manage Courses(Delete Courses) (Fatima Aftab)

## 

# CHAPTER 5 Operation Contracts (OC’s)

## 5.1. Login (BASIT IQBAL : FA21-BSE-050)

|  |  |
| --- | --- |
| Id | 01 |
| Operation | displayDashboard( userType:String,session:AppSession) |
| Cross Reference | Login |
| Pre-Conditions | The user must have log in to the system by entering valid username and password. |
| Post Conditions | The dashboard was displayed to the user when he entered correct username and password.  If the username of password was incorrect then the error message was displayed and the user was asked to reenter the user name and password. |

## 5.2. Logout (BASIT IQBAL: FA21-BSE-050)

|  |  |
| --- | --- |
| Id | 02 |
| Operation | displayLoginPage( ) |
| Cross Reference | Logout |
| Pre-Conditions | The user must have paused or canceled all the running processes and must have clicked on logout button. |
| Post Conditions | If all the process have been paused then the user was redirected to login page.  If the process have not been paused then the user was asked to close or pause the process. |

## 5.3. Generate Timetable Use case Operation Contracts (Fatima Aftab)

|  |  |
| --- | --- |
| Id | 03 |
| Operation | SelectGenerateTimetableModule() |
| Cross Reference | None |
| Pre-Conditions | The admin was logged into the system and had the necessary permissions to access the generate timetable module. |
| Post Conditions | The system displayed a new screen where the admin could enter the details for the timetable. |

|  |  |
| --- | --- |
| Id | 04 |
| Operation | DisplayTimetableDetailsScreen() |
| Cross Reference | None |
| Pre-Conditions | The admin had selected the generate timetable module. |
| Post Conditions | The system presented a screen with input fields for the admin to enter the necessary details for generating the timetable. |

|  |  |
| --- | --- |
| Id | 05 |
| Operation | SetTimetableDetails(details: TimetableDetails) |
| Cross Reference | None |
| Pre-Conditions | The admin was on the timetable details screen and had entered all the required information for generating the timetable. |
| Post Conditions | The system stored the provided timetable details for generating the timetable. |

|  |  |
| --- | --- |
| Id | 06 |
| Operation | GenerateTimetable() |
| Cross Reference | None |
| Pre-Conditions | The admin had set all the necessary details for generating the timetable. |
| Post Conditions | The system applied a genetic algorithm to generate a timetable based on the provided details. |

|  |  |
| --- | --- |
| Id | 07 |
| Operation | DisplayTimetable(timetable: Timetable) |
| Cross Reference | None |
| Pre-Conditions | The timetable generation process had completed successfully. |
| Post Conditions | The system presented the generated timetable to the admin, displaying it on the screen for review and use. |

## 5.4. Modify Timetable Scenario Operation Contract (Fatima Aftab)

|  |  |
| --- | --- |
| Id | 08 |
| Operation | SelectModifyTimetableModule() |
| Cross Reference | None |
| Pre-Conditions | The admin is logged into the system and has the necessary permissions to access the modify timetable module. |
| Post Conditions | The system displays the semester timetable list to the admin. |

|  |  |
| --- | --- |
| Id | 09 |
| Operation | DisplaySemesterTimetableList() |
| Cross Reference | None |
| Pre-Conditions | The admin has selected the modify timetable module. |
| Post Conditions | The system presents a list of semester timetables to the admin for modification. |

|  |  |
| --- | --- |
| Id | 10 |
| Operation | SelectTimetableToModify(timetable: Timetable) |
| Cross Reference | None |
| Pre-Conditions | The admin is viewing the semester timetable list. |
| Post Conditions | The system displays a list of details to be modified for the selected timetable to the admin. |

|  |  |
| --- | --- |
| Id | 11 |
| Operation | DisplayModificationDetailsList(timetable: Timetable) |
| Cross Reference | None |
| Pre-Conditions | The admin has selected a timetable to modify. |
| Post Conditions | The system presents a list of details that can be modified for the selected timetable to the admin. |

|  |  |
| --- | --- |
| Id | 12 |
| Operation | EnterChanges(details: ModifiedDetails) |
| Cross Reference | None |
| Pre-Conditions | The admin is viewing the list of details to be modified for the selected timetable. |
| Post Conditions | The admin enters the desired changes for the selected details of the timetable. |

|  |  |
| --- | --- |
| Id | 13 |
| Operation | DisplayModifiedTimetable(timetable: Timetable) |
| Cross Reference | None |
| Pre-Conditions | The admin has entered the changes for the timetable details. |
| Post Conditions | The system displays the modified timetable to the admin for review. |

|  |  |
| --- | --- |
| Id | 14 |
| Operation | ApproveAndSaveChanges() |
| Cross Reference | None |
| Pre-Conditions | The admin has reviewed the modified timetable and is satisfied with the changes. |
| Post Conditions | The system saves the approved changes to the timetable and updates it accordingly. |

## 5.6. Manage Courses(View Courses) Fatima Aftab

|  |  |
| --- | --- |
| Id | 15 |
| Operation | viewCourses() |
| Cross Reference | None |
| Pre-Conditions | 1.The user must be logged in as an admin with appropriate access privileges.  2.The database connection must be established and functional. |
| Post Conditions | The system had displayed a list of courses from the database. |

|  |  |
| --- | --- |
| Id | 16 |
| Operation | DisplayList() |
| Cross Reference | None |
| Pre-Conditions | Admin must click on manage courses module. |
| Post Conditions | The system had displayed a list of courses from the database. |

## 5.7. Manage Slots(View Slots) Basit Iqbal

|  |  |
| --- | --- |
| Id | 17 |
| Operation | viewSlots() |
| Cross Reference | None |
| Pre-Conditions | Admin must click on manage slots module. |
| Post Conditions | The system had displayed a list of slots from the database. |

|  |  |
| --- | --- |
| Id | 18 |
| Operation | DisplaySlots() |
| Cross Reference | None |
| Pre-Conditions | Admin must click on view slots module. |
| Post Conditions | The system had displayed a list of slots from the database. |

## 5.8. Manage Slots(Delete Slots) Basit Iqbal

|  |  |
| --- | --- |
| Id | 19 |
| Operation | deleteSlots() |
| Cross Reference | None |
| Pre-Conditions | * 1. Admin must click on manage slots module.   2. Admin must select the slot to be delete, |
| Post Conditions | The system had deleted a slot from the database. |

|  |  |
| --- | --- |
| Id | 20 |
| Operation | displayUpdatedSlots() |
| Cross Reference | None |
| Pre-Conditions | * 1. Admin must click on manage slots module.   2. Admin must select the slot to be delete, |
| Post Conditions | The system had deleted a slot from the database.  The system displayed the updated slots from database |

## 5.9. Manage Slots(add Slots) Basit Iqbal

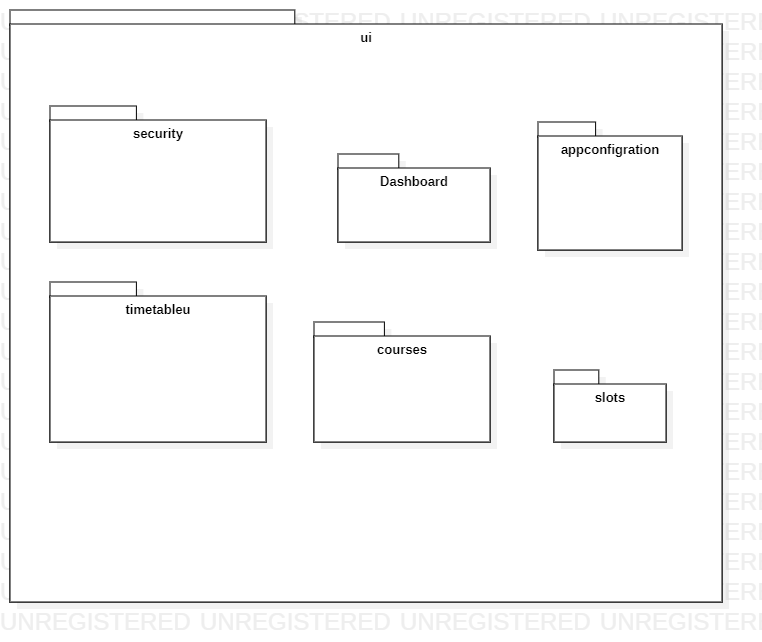
|  |  |
| --- | --- |
| Id | 21 |
| Operation | addSlots(data:slots) |
| Cross Reference | None |
| Pre-Conditions | * 1. Admin must click on manage slots module.   2. Admin must select add module.   3. Admin enters the new slot details. |
| Post Conditions | The slot had added to database |

|  |  |
| --- | --- |
| Id | 22 |
| Operation | displayNewSlots() |
| Cross Reference | None |
| Pre-Conditions | * 1. Admin must click on manage slots module.   2. Admin must select add module.   3. Admin enters the new slot details. |
| Post Conditions | The slot had been added to database.  The system had displayed new slots in list of slots |

## 5.10. Manage Courses(Delete Courses) Fatima Aftab

|  |  |
| --- | --- |
| Id | 23 |
| Operation | deleteCourse(id:string) |
| Cross Reference | None |
| Pre-Conditions | * 1. Admin must click on manage courses module.   2. Admin must select the slot to be delete, |
| Post Conditions | The system had deleted a slot from the database.  The system displayed the updated slots from database |

# CHAPTER 6 : Package Diagram



A screenshot of a computer screen

Description automatically generated

# CHAPTER 7 : CLASS DIAGRAM

# CHAPTER 8 : INTERACTION DIAGRAM

## 8.1 Communication Diagram:

### UC : Login (FA21-BSE-050)



### UC : Manage Courses [view courses](FA21-BSE-088)

## 8.2. Sequence Diagram:

### UC : Manage Courses [view courses](FA21-BSE-088)



### UC : Login (FA21-BSE-050)

