**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**TIMETABLE MANAGEMENT SYSTEM**

**CSC392 Object-Oriented Software Engineering**

Submitted on: <04-04-2023>

Group Members:

Saud Khan (FA21-BSE-033)

Usama Ijaz (FA21-BSE-186)

Zakariya Anwer (FA21-BSE-042)

Contents

[**CHAPTER 1 PROJECT PROPOSAL** 3](#_Toc131950839)

[**Introduction** 3](#_Toc131950840)

[**Vision and Business Case** 3](#_Toc131950841)

[**Use-Case Model** 4](#_Toc131950842)

[**Supplementary Specification** 4](#_Toc131950843)

[**Glossary** 5](#_Toc131950844)

[**Risk list and risk management plan** 5](#_Toc131950845)

# **CHAPTER 1 PROJECT PROPOSAL**

## **Introduction**

A timetable management system is a software application that helps users create, manage, and organize their schedules. It can be used for a variety of purposes, such as managing academic timetables.

The system allows users to input their schedules, and then generates a visual representation of the timetable that can be easily viewed and modified as needed. Admin can add, delete, or modify, and the system will automatically update the timetable accordingly.

## **Vision and Business Case**

The high-level goals of the timetable management system for a university are to improve scheduling efficiency, reduce conflicts, and increase overall productivity. The system should be able to handle a large number of users, including students, faculty, and staff, and be able to handle complex scheduling requirements, such as room and resource allocation.

The constraints of the system include the need to integrate with existing university systems, such as student and faculty databases, and the need to adhere to strict privacy and security protocols to protect sensitive data.

**Business Case:**

The timetable management system for a university can provide significant benefits, including:

1. Increased efficiency: The system can automate many of the scheduling processes, reducing the time and effort required to create and manage schedules.
2. Reduced conflicts: The system can detect and resolve scheduling conflicts, reducing the likelihood of double-bookings or other issues.
3. Improved communication: The system can provide a centralized location for all scheduling information, making it easier for students, faculty, and staff to stay informed and up-to-date.
4. Enhanced productivity: By streamlining scheduling processes and reducing conflicts, the system can help increase overall productivity and effectiveness.

**Executive Summary:**

The timetable management system for a university is a software application that helps improve scheduling efficiency, reduce conflicts, and increase productivity. The system should be able to handle complex scheduling requirements and integrate with existing university systems while adhering to strict privacy and security protocols. The system can provide significant benefits, including increased efficiency, reduced conflicts, improved communication, and enhanced productivity. By implementing a timetable management system, the university can improve its scheduling processes and better serve its students, faculty, and staff.

## **Use-Case Model**

The use-case model for a timetable management system for a university may include the following functional requirements:

1. **Login:** Allow users to log in to the system using their university credentials.
2. **Create Schedule:** Allow admin to create a new schedule for a particular semester or academic term.
3. **Modify Schedule:** Allow admin to modify an existing schedule by adding or removing classes or changing rooms.
4. **View Schedule:** Allow users (student, faculty & other staff) to view their own schedules or timetable.
5. **Room Allotting:** Allow authorized users to allot room or do changing in it.
6. **Resource Allocation:** Allow authorized users to allocate resources (such as projectors, microphones, or laptops) for classes or events.
7. **Notification:** Notify users of any changes to their schedules or scheduling conflicts.
8. **Report Generation:** Allow users to generate reports on scheduling information, such as attendance reports or room utilization reports.

## **Supplementary Specification**

1. **Performance:** The system should be able to handle a large number of users and data without significant delays or downtime.
2. **Scalability**: The system should be scalable, able to accommodate an increasing number of users and data as the university grows.
3. **Availability**: The system should be available to users at all times, with minimal downtime for maintenance or upgrades.
4. **Security**: The system should be designed with robust security features to protect sensitive data and prevent unauthorized access.
5. **Usability**: The system should be easy to use, with an intuitive interface and clear instructions for users.
6. **Accessibility**: The system should be accessible to users with disabilities, conforming to accessibility standards and guidelines.
7. **Reliability**: The system should be reliable, with minimal errors or failures.
8. **Compatibility**: The system should be compatible with various browsers and devices, to ensure all users can access the system.
9. **Compliance**: The system should comply with relevant laws, regulations, and policies, such as data protection laws and university policies on information security and privacy.
10. **Maintenance**: The system should be easy to maintain and upgrade, with minimal disruption to users.

During the inception phase, it is useful to have some idea of the key non-functional requirements that will have a major impact on the architecture. This will help ensure that the system is designed to meet these requirements, and that any trade-offs between different requirements are identified and addressed. As the project progresses, additional non-functional requirements may be identified and added to the supplementary specification as needed.

## **Glossary**

1. **Timetable Management System:** A software application that allows users to create, modify, and manage schedules for classes and events at a university.
2. **Schedule**: A list of classes or events that occur during a specific time period, such as a semester or academic term.
3. **Scheduling Conflict:** An issue that occurs when two or more classes or events are scheduled at the same time or in the same location.
4. **Resource**: Any physical item required for a class or event, such as projectors, microphones, or laptops.
5. **Room**: A physical space used for classes or events, such as a classroom or lecture hall.
6. **User**: Any individual who interacts with the timetable management system, including students, faculty, and staff.

## **Risk list and risk management plan**

**Risk List**

1. **Inadequate System Performance:** There is a risk that the timetable management system will not perform adequately, resulting in delays or downtime for users.
2. **Security Breach:** There is a risk that the system will be compromised by a security breach, resulting in unauthorized access to sensitive data.
3. **Insufficient Budget:** There is a risk that the project budget will be insufficient to meet all project requirements, resulting in incomplete or suboptimal system functionality.
4. **Lack of Skilled Personnel:** There is a risk that the project team will lack the necessary skills and expertise to complete the project successfully.
5. **Project Delays:** There is a risk that the project will be delayed due to unforeseen circumstances, such as technical issues or resource constraints.
6. **Resistance to Change:** There is a risk that users may resist using the new system, resulting in low adoption rates and limited system effectiveness.

**Risk Management Plan:**

1. **Inadequate System Performance:** Mitigation strategies could include performance testing and optimization during development, as well as ongoing monitoring and maintenance to ensure system performance meets expectations.
2. **Security Breach:** Mitigation strategies could include implementing robust security features such as encryption, firewalls, and user authentication, as well as ongoing monitoring and updating of security protocols.
3. **Insufficient Budget:** Mitigation strategies could include careful budget planning and prioritization, as well as seeking additional funding sources if necessary.
4. **Lack of Skilled Personnel:** Mitigation strategies could include identifying skill gaps early in the project and addressing them through training, hiring additional staff with necessary expertise, or outsourcing certain tasks to third-party providers.
5. **Project Delays:** Mitigation strategies could include developing a realistic project timeline with contingency plans, monitoring progress regularly, and adjusting timelines as needed to ensure the project stays on track.
6. **Resistance to Change:** Mitigation strategies could include involving users in the development process through user testing and feedback, providing clear communication and training materials, and highlighting the benefits of the new system to encourage user adoption.

Top of Form