

Project Name:

Student Course Registration & Tracking System

Course Details:

Course Code: CS-322

Course: Software Construction

Instructor: Dr. Onaiza Maqbool

Submitted by:

Ahmed Nazir (Team Lead)

Ahsen Rasheed

Yamaan Bangash

Submission Date: 5th March 2025

Department of Computer Science
Quaid-i-Azam University Islamabad

This document is submitted as part of the **Software Construction** course and has been reviewed and approved by the team members.

This document represents our collective effort in planning and preparing **(Assignment 1)** for CS Office System for Student Course Registration and Tracking.

Name	Reg No.	Signature
Ahmed Nazir	04072312025	
Ahsen Rasheed	04072312012	
Yamaan Bangash	04072312021	

Instructor Name: Dr. Onaiza Maqbool

Signature: _____

Change History

Version	Date	Description	Name
1.0	4 th March 2025	Initial draft created	Ahmed Nazir
1.1	5 th March 2025	Added project details and refined objectives	Yamaan Bangash
1.2	5 th March 2025	Reviewed and finalized document	Ahsen Rasheed

Preface

This project plan has been prepared as part of our coursework for the Software Construction (SC) course at CS Department Quaid-i-Azam University, Islamabad. Our team has worked together to plan and design the Student Course Registration & Tracking System, detailing its objectives, development approach, and implementation strategies.

Table of contents

Sr No.	Task	Author	Page
1	Project Overview	Ahmed Nazir	5

1.1	Project Summary	Ahmed Nazir	5
1.2	Purpose, scope and objectives	Ahmed Nazir	5,6
1.3	Assumptions and constraints	Ahmed Nazir	7
1.4	Project deliverables	Ahmed Nazir	8
1.5	Schedule summary	Ahmed Nazir	8
2	References	Ahmed Nazir	9
3	Definitions	Ahmed Nazir	9
4	Project context	Ahsen Rasheed	10
4.1	Process model	Ahsen Rasheed	10
4.2	Methods, tools and techniques	Ahsen Rasheed	11
4.3	Product acceptance	Ahsen Rasheed	12
5	Project planning	Yamaan Bangash	14
5.1	Project work plans	Yamaan Bangash	14
5.2	Work activities	Yamaan Bangash	15
5.3	Schedule allocation	Yamaan Bangash	15
5.4	Resource allocation	Yamaan Bangash	16
8	Supporting process plans	Ahsen Rasheed	13
8.1	Risk management	Ahsen Rasheed	13,14

Project Plan for Student Registration and Tracking System

1. Project Overview

Project Summary:

The CS Office requires an automated system to efficiently track student course registrations, manage pass/fail status, and enforce prerequisite rules. This system will ensure that students register only for courses they are eligible for and will provide coordinators and timetable managers with a structured view of student progress. The system will also store the **scheme of study** for different batches, allowing seamless academic planning and monitoring.

Purpose, Scope, and Objectives

Purpose

The purpose of this project is to develop a **centralized course registration and tracking system** that improves efficiency, reduces manual workload, and ensures adherence to academic policies.

Scope

1. Student **course registration with prerequisite enforcement**.
2. **Automated pass/fail tracking** for student records.
3. **Study scheme management** for each batch.
4. Coordinator dashboard to **view student progress**.
5. **Timetable coordinator access** to plan course offerings.
6. Allow coordinators to **track students completed, ongoing, and skipped courses**.

Objectives

1. **Automation of course registration** to minimize manual workload.
2. **Accurate prerequisite enforcement** to prevent students from enrolling in courses they are not eligible for.

3. **Real-time tracking of student progress** for better decision-making.
4. **Timetable coordination support** for scheduling and planning courses effectively.
5. **Secure and scalable database** for long-term usage.

Assumptions and Constraints

Assumptions

- The system will be **web-based** and accessible from any device.
- The system will be used **by students, coordinators, and timetable managers**.
- Internet and **institutional server access** will be available for all users.
- **Coordinators and admins** will have different levels of access to the system.
- The study scheme for each batch will be **predefined and uploaded** into the system.

Constraints

- The system must **follow university course registration policies**.
- **Prerequisite enforcement must be accurate**, ensuring students register correctly.
- Student **data confidentiality and security** must be maintained.

- The system should **work within the university's existing IT infrastructure.**

Project Deliverables

The following deliverables will be produced:

- **A web-based student registration system** with role-based access.
- **A database containing student records, course details, and prerequisite data.**
- **A user-friendly interface** for students to register for courses.
- **An admin panel for coordinators** to track student progress and update records.
- **A reporting module** for generating student academic progress reports.
- **Comprehensive documentation** and user manual.

Schedule Summary

Phase	Duration	Key Tasks
Requirement analysis	1-2 weeks	Gather system requirements, finalize scope
System design	3-4 weeks	Define architecture, database structure, UI design
Development phase	5-9 weeks	Implement core functionalities, integrate modules
Testing and debugging	10-11 weeks	Identify and fix errors, ensure accuracy

Deployment and review	12-13 weeks	Deploy system, collect feedback, finalize documentation
-----------------------	-------------	---

2. References

- **Case Study Problem Description** provided by the instructor.
- **University academic policies** regarding course registration and tracking.
- Current University **CMS System**.
- **ISO/IEC/IEEE 16326-2019**: Standard for project management plans.
- **IEEE Std. 830-1998**: Guidelines for defining software scope.

3. Definitions

- **Student Registration**: The process by which students enroll in courses.
- **Prerequisite Checking**: Ensuring students meet required course conditions before registration.
- **Coordinator**: University staff responsible for managing course records.
- **Timetable Coordinator**: A staff member responsible for scheduling courses based on student enrollment.
- **Study Scheme**: A structured course plan for a specific student batch.
- **Pass/Fail Status**: The record of whether a student has successfully completed a course.
- **DBMS**: The system that stores all student records and course details securely.

- **Web-based System:** An online platform accessible via a browser, used for course registration and tracking.

4. Process Model

This system will follow the following road map.

- **Course Registration:** Students select courses that is offering by the department this list is provide by the course coordinator in which it will tell which course are offer for this semester. The system verifies that the prerequisites for the selected courses are met before allowing registration.
- **Scheme of Study Upload:** The coordinator uploads the scheme of study for each batch. In this scheme it will tell about the prerequisite courses .
- **Pre-Requisite Checking:** Whenever a student tries to register for a course, the system automatically checks if they have passed the prerequisite courses. If not then he can't register the course until it will pass previous.
- **Tracking student progress:** The system allows the CS office to track which courses each student has taken, which they have passed, and which they have skipped or failed.
- **Timetable Coordination:** The timetable coordinator can view course registrations and plan the course schedule accordingly to ensure optimal room allocation and avoid scheduling conflicts.
- **Reporting:** The system generates reports regarding the students' academic progress, registration history, and the courses they have successfully completed. It also tell about the grades of student at the end of each student.

Methods, Tools, and Techniques:

Methods:

- Agile Development: We will use the agile process because the agile process allows you to change the rules and you can also add feature after another like the rules of university change then it is easy to change if it is in agile process.
- Database Management: A relational database management system (RDBMS) will be used to store data about students, courses, and prerequisites.
- Data Validation: A validation method will be implemented to ensure that prerequisites are checked before course registration.

Tools:

Programming Languages: Java, Python, or C# for backend development.

Database Management Systems (DBMS): MySQL, or SQL Server for storing student and course data.

Web Development: HTML and CSS, for front-end development.

Team coordination: Git for managing code versions and collaborative development.

Techniques:

- Object-Oriented Programming (OOP): For oriented development and reuse of code.
- User Authentication: For securing access to the system for coordinators, students, and the timetable coordinator.
- Report Generation: For creating summary reports of student progress and course status.

Product Acceptance Plan:

The product acceptance plan outlines the criteria and procedures to determine whether the system meets the requirements and is ready for deployment:

1. Functional Testing:

- Ensure that the course registration process works, including the prerequisite checks.
- Verify that the system correctly tracks and displays students' courses, pass/fail status, and skipped courses.
- Confirm that the upload mechanism for the scheme of study works without errors.
- Test the accessibility and functionality of the system for the timetable coordinator to view student data.

2. Performance Testing:

- Test the system under load, especially during peak registration periods, to ensure that it can handle a large number of users.

3. Security Testing:

- Ensure that all user data is securely stored, and that access control measures are in place to restrict unauthorized access to sensitive information.

4. User Acceptance Testing (UAT):

- The CS office and the timetable coordinator will perform tests to confirm the system meets their expectations and provides the necessary features.
- Users will test the system with real-world scenarios (registering for courses, viewing progress, uploading the scheme of study).

5. Training:

- Provide training sessions for the CS office staff and timetable coordinator on how to use the system effectively.

6. Bug Fixes and Enhancements:

- Any bugs discovered during UAT will be addressed, and any additional features or changes based on feedback will be implemented.

8. Supporting Process Plans

This will ensure that every aspect of system has been checked and tested carefully. It indicates that now the system is ready to launch. This includes Project Management plan, Quality assurance, Deployment plan, Training the users etc.

Risk Management for the Course Registration System

This will manage that if a risk is accrued how we can meet with it and how to resolve it. It also checks the impact of the risk on the system.

Key Risk Management Steps:

1. Risk Identification:

Firstly you should identify which risk could accrue in the future you should have the planning for how you will resolve it.

2. Risk Assessment: After identifying the risks, assess the impact how much it can harm the system when it accrues. Impact could be varied like server failure will break down the whole system.

3. Risks Types:

- Technical Risks
- Data Integrity Risks
- Stakeholder Risks

4. Risk Monitoring:

- Track the identified risks throughout the project's life cycle.
- Use tools like **risk logs** to document, update, and assess the status of risks and mitigation efforts.

Example Risk Management Table:

Risk	Probability	Impact	Solution
Server Failure	Medium	High	Regular backup
Hacker Attack	Medium	High	Update the Antivirus
System Down/lacking	High	Medium	Improve the DBMS

5. Project Planning

Project Work Plans

The project will be executed in phases mirroring the semester schedule:

- Planning (Weeks 1-2): Finalize project plan, define team roles, set up the development environment.
- Analysis (Weeks 3-5): Gather detailed requirements, define use cases, develop the Software Requirements Specification (SRS).
- Design (Weeks 6-9): Design the system architecture, database schema, and user interfaces.
- Development (Weeks 10-13): Implement the system, focusing on backend logic, frontend design, and database connectivity.
- Testing (Week 14): Conduct unit and system testing, prepare test reports.
- Presentation (Weeks 15-16): Prepare and deliver project presentations.

Work Activities

- Requirement Gathering: Conduct interviews and surveys to gather detailed requirements from the CS office.
- Use Case Development: Define how different users will interact with the system.
- SRS Development: Document all functional and non-functional requirements.
- Database Design: Create an efficient database schema to store and manage data.
- UI/UX Design: Design user-friendly interfaces for students, coordinators, and timetable managers.
- Coding: Develop the system based on the design specifications.

- Testing: Perform thorough testing to identify and fix bugs.
- Documentation: Prepare user manuals and technical documentation.

Schedule Allocation


- Gantt Chart: A detailed Gantt chart, created using ProjectLibre, will be included here. It will visually represent the project timeline, task dependencies, and milestones.

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Introduction	7 days	3/5/25 8:00 AM	3/13/25 5:00 PM		
2		Project Planning and In...	7 days	3/5/25 8:00 AM	3/13/25 5:00 PM		
3		Planning	7 days	3/14/25 8:00 AM	3/24/25 5:00 PM	1	
4		Development of Project ...	7 days	3/14/25 8:00 AM	3/24/25 5:00 PM		
5		Analysis Phase	23 days	3/25/25 8:00 AM	4/24/25 5:00 PM	3	
6		Defining Usecases	6 days	3/25/25 8:00 AM	4/1/25 5:00 PM		
7		EID Break	5 days	4/2/25 8:00 AM	4/8/25 5:00 PM	6	
8		Development Analysis M...	5 days	4/9/25 8:00 AM	4/15/25 5:00 PM	7	
9		SRS Development and R...	7 days	4/16/25 8:00 AM	4/24/25 5:00 PM	8	
10		Design Phase	28 days	4/25/25 8:00 AM	6/3/25 5:00 PM	5	
11		Design Principle and Proj...	7 days	4/25/25 8:00 AM	5/5/25 5:00 PM		
12		Interface Design	7 days	5/6/25 8:00 AM	5/14/25 5:00 PM	11	
13		Detail Design	7 days	5/15/25 8:00 AM	5/23/25 5:00 PM	12	
14		Review of Design	7 days	5/26/25 8:00 AM	6/3/25 5:00 PM	13	
15		Development	28 days	6/4/25 8:00 AM	7/11/25 5:00 PM	10	
16		Database Connectivity	7 days	6/4/25 8:00 AM	6/12/25 5:00 PM		
17		Refining the Frontend	7 days	6/13/25 8:00 AM	6/23/25 5:00 PM	16	
18		Development of Classes	7 days	6/24/25 8:00 AM	7/2/25 5:00 PM	17	
19		Refinement of Classes a...	7 days	7/3/25 8:00 AM	7/11/25 5:00 PM	18	
20		Testing	21 days	7/14/25 8:00 AM	8/11/25 5:00 PM	15	
21		Testing of Software Syst...	7 days	7/14/25 8:00 AM	7/22/25 5:00 PM		
22		Project Presentation	14 days	7/23/25 8:00 AM	8/11/25 5:00 PM	21	

Resource Allocation

- Human Resources:
 - Team Lead (Member 1): Responsible for overall project management, planning, and documentation.

- Member 2: Focus on system design, database management, and backend development.
- Member 3: Handle frontend development, UI/UX design, and testing.
- Technical Resources:
 - Software: ProjectLibre, Visual Studio Code, XAMPP (or similar stack), Git.
 - Hardware: Laptops with necessary configurations for development and testing.
 - Database: MySQL or a similar database management system.
 - Platforms: GitHub for version control, MS Teams for communication and submission.

		Name	RBS	Type	E-mail Address	Material Label
1		DBMS		Material		
2		Team Leader		Work		
3		Manager		Work		
4		Testing Team		Work		
5		Analysis team		Work		
6		Coding team		Work		
7		Project Libre		Material		
8		Ms Word		Material		
9		My SQL		Material		
10		HTML		Material		
11		CSS		Material		
12		Netbeans		Material		
13		Designing team		Work		
14		Git/Github		Material		
15		Computers/Laptop		Material		
16		Backup storage		Material		