Ain Shams University
Faculty of Engineering
Computer Engineering and Software Systems Program

CSE242/CSE232: Advanced Software Engineering – Fall 2025

T E R M - P R O J E C T   R E Q U I R E M E N T S

This project is a group project with each group has 6 or 8 students. A detailed description that elaborates on the proposed project is needed. Each group can select from the ideas provided at the end of this document, extends and elaborates the ideas. Also, the group can provide their own ideas. The project required full analysis, design, and implementation. There is no restriction for the use of specific programming language, each group can agree upon the programming language they will use in the project implementation.

Project Proposal

Each group should deliver a one-page project proposal that contains the description of the project, its scope, and the names of the group members. Each group must get the TA approval for the proposal before the end of week 2 to be able to start the project.

Project Deliverable

The following are required to be delivered by the due date:

- Project source code using an object-oriented programming language. It must be submitted on LMS in a zip/rar archive. Full implementation of the project should be delivered by the due time.
- A presentation (in .pptx format) of the different phases of the project.
- Project document (in .docx format) that contains at least the sections shown hereafter.

Project Implementation

Each group must submit the source code of the project implementation using any framework or programming language but the project must function correctly.

Project Document

Each group must submit one project document by the due date that contains at least the following sections:

Cover Page

This page should contain university name, faculty name, and department name at the top of the page. In addition, course code and name, project name, your names, and your student IDs should be provided on the cover page.

Abstract: A brief description of the project.

Table of Contents

The table of contents of the whole document; include the page number for each section/sub-section.

1. Introduction

1.1 Purpose	The purpose of this document and its intended readership.
1.2 List of Definitions	The definitions of all used terms, acronyms and abbreviations.
1.2 Scope	Identify the product by name and explain what the software does.
1.3 List of References	References to all applicable documents.
1.4 Overview	Short description of the rest of the document and how it is organized.

2. General Description

2.1 Product Perspective	The relation to other systems that may be used by the client.
2.2 General Capabilities	The main capabilities of the system.
2.3 General Constraints	Constraints, background information, and justification.
2.4 User Characteristics	The characteristics of the different user roles.
2.5 Environment Description	A description of the operational environment of the system.
2.6 Assumptions and Dependencies	The assumptions upon which the specific requirements are based.

3. Specific Requirements

3.1 Capability Requirements	A list of all capability requirements.
3.2 Constraint Requirements	A list of all constraint and restrictions on user requirements.

4. Use-Case Diagram and the Narrative Description of all use cases

Use-case diagram with the narrative description of all the uses cases should be provided in this section.

5. Swimlane Diagrams of all use cases

Swimlane diagram for each use-case diagram should be provided in this section.

6. Interaction Diagrams

This section should contain the interaction diagrams (either sequence diagrams or collaboration diagrams could be adopted) for each use case / major scenrios in the project. Proper descriptions and/or comments on the provided diagrams should also be shown in this section.

7. Noun Extraction and CRC Cards

In this section you need to provide the detailed procedure in applying both noun extraction and CRC methods. Additionally, detailed CRC cards should be provided in addition to any updates in the use case diagram and its associated narrative description.

8. State Diagram

This section should contain the detailed state diagram for the whole system, all assumption you used in the state diagram should be stated explicitly in this section.

9. Client-Object Relation Diagram

This section should contain the client-object relation diagram for your project and its proper description. All objects that must be initiated by the main application object (or main function if C++ is adopted) should be clearly stated in this section.

10. Class Diagram

Class diagram that contains classes, their relationships, attributes, and operations should be provided in this section. Attribute types, operations arguments, argument types, and return types of each operation must be decided at this stage. If you use user-defined data types, you need to provide the detailed description of these types in terms of the built-in types of the language that you will use for implementing the project. You need to provide a proper description of this diagram in which you can show all assumptions used, any modifications done on the class diagram and the justifications of the modifications, and any other details that you cannot provide on the diagram itself.

11. Architectural Model

The architectural model of the system with the justification of the design decision should be detailed in this section.

12. Component Diagram

The component diagram of your system that shows the major components and the classes that represent each component, and how the components are interacted should be provided in this section.

13. Project Estimated Cost

Estimate the cost of your project using function points and COCOMO-II Model

14. Time Plan

A time plan of all phases of the project is going to be detailed in this section. A Gantt chart should be provided to show the actual time spent in each phase of the project. Also, the overlapping among the different activities of the project should be shown in the Gantt chart.

15. References

You need to list the references that you have adopted while preparing your report. All references must be written in IEEE format, and they must be cited inside the text. The order of the references should match the order of citing them inside the text. Do not number the title "References" in the report since it is not considered a section.

16. Appendices

This section might be needed if you have extra information that helps clarifying certain issues in your document that you don't need to distract the reader by adding them inside the text.

Important Notes

- Use UML notations (when applicable) for the diagrams you provide in your document.
- Use a professional drawing tool (e.g., MS-VISIO) to draw the diagrams in your document.
- Use consistent document format (font sizes, titles, subtitles, captions, paragraph formatting … etc.). Recommended font sizes are: main title 14pt, subtitles 12pt, main text 12pt, and captions 10 pt. Recommended font type is "**Calibri**" bold for titles and subtitles, and "Calibri" for all other texts. Recommended spaces before and after paragraphs are 12pt before and 6pt after each paragraph, and single spacing is highly recommended for the main text. Justified paragraphs from both sides are also recommended.
- Figures and tables must be centred in the pages, and they should be numbered separately. Each figure/table must have a caption that appears below the figure / above the table.
- Pages must be numbered consistently except the cover page.
- All sections must be numbered and paragraphs should start with a tab.
- Avoid having consecutive titles/sub-titles without separating them by an introductory paragraph.
- Avoid using column ":" after the titles.
- All reports must be written in English, always avoid typos and grammatical errors.
- Always use the technical writing skills while preparing your report.
- Report will go plagiarism check.
- All project deliverable must be uploaded to the LMS, no hardcopy is accepted, and please do not send your project deliverables by email.
- A presentation, discussion, and demo are required by the end of the project.

Appendix 1: Proposed Project Ideas

The following represents a set of proposed project ideas, you can select from them, then extend and elaborate the idea. Also, you can provide your own ideas.

1. Enterprise Resource Planning (ERP) Lite System

A modular ERP system for small businesses with modules like inventory, sales, HR, and accounting. Each module should be loosely coupled but integrated via a central service bus or API gateway. The focus is on layered architecture, modularity, and integration testing.

2. Real-Time Collaborative Document Editor

A multi-user editor supporting concurrent edits, undo/redo, and conflict resolution. Students can implement algorithms like Operational Transformations or CRDTs to handle distributed edits. It emphasizes concurrency control and distributed synchronization.

3. Healthcare Appointment Management System

A patient–doctor appointment system with role-based access for patients, doctors, and admins. Features include appointment booking, cancellation, notifications, and scheduling conflict detection. The project emphasizes transactional consistency and usability.

4. Collaborative Bug Tracking System

A web-based issue tracker that supports bug reporting, assignment, duplicate detection (via metadata, not AI), and release milestone planning. It includes dashboards for project managers and developers. Focus areas: workflow modeling, versioning, and role-based access control.

5. Customizable Learning Management System (LMS)

An LMS where courses, quizzes, and assignments are configurable by instructors. Students can track progress, while admins manage roles and permissions. The focus is on system modularity, extensibility, and adherence to software engineering design principles.

6. Online Examination System with Proctoring Logs

A secure exam system where instructors create exams with configurable rules (time limits, question types, randomization). The system generates detailed proctoring logs (navigation, session activity) for audit. Key focus: security, data integrity, and test coverage.

7. Event-Driven Ticket Booking System

An online ticketing platform (for buses, concerts, etc.) that handles concurrency and ensures no double-booking. The project involves transactional consistency, event-driven architecture, and load handling.

8. Project Management Dashboard

A web-based system for managing projects, tasks, milestones, and resources. It should provide Gantt chart visualization and role-based permissions. The project emphasizes modular architecture, reporting, and usability engineering.

9. Multi-Branch Library Management System

A system that manages multiple library branches under one platform, supporting cataloging, user borrowing, inter-branch transfers, and fines. The project emphasizes database normalization, distributed transactions, and modular architecture.

10. Online Food Ordering and Delivery Platform

A web app that connects restaurants, delivery staff, and customers. It should support order tracking, restaurant dashboards, and real-time delivery status updates. Focus: modular design, concurrency handling, and RESTful API design.

11. Conference Management System

A platform for managing academic or business conferences: submission, peer review, scheduling sessions, and attendee registration. Students practice workflow modeling, role-based permissions, and complex entity-relationship management.

12. Multi-Currency Accounting System

An accounting software that supports transactions in multiple currencies with automatic conversion using exchange rate APIs. It emphasizes transactional consistency, modularity, and financial reporting.

13. Online Auction System

A platform for real-time auctions where multiple users bid simultaneously on items. The system must handle concurrency, ensure data consistency, and implement anti-sniping rules.

14. Event Registration and Ticketing Platform

A system where users register for events, purchase tickets, and receive QR-code-based e-tickets. The project emphasizes database schema design, integration with payment gateways, and role-based workflows.

15. Fleet Management System

A platform for logistics companies to manage vehicles, drivers, and trips. Features include route planning, vehicle maintenance logs, and reporting. The project emphasizes modular design and reporting systems.

16. Warehouse Inventory Management System

A system for tracking stock levels, supplier orders, and shipment dispatches across multiple warehouses. It should support reorder triggers and detailed reporting. The focus is on distributed database consistency and usability.

17. Multi-User Blogging and Commenting Platform

A blogging site where multiple users can post, comment, and manage content. Features include content moderation, versioning, and tagging. Students will practice content management architecture and role-based access.

18. Construction Project Monitoring System

A platform for tracking progress, budgets, and milestones of construction projects. It should support file uploads (blueprints, permits) and Gantt chart visualizations. Focus: modularity, workflow management, and reporting.

19. Hotel Reservation System

A system where guests can browse hotels, book rooms, and manage reservations. Features include cancellation policies, overbooking prevention, and billing. Emphasis: transactional integrity and usability design.

20. Online Course Registration Portal

A student course registration system with prerequisites, waitlists, and timetable conflict detection. Focus: rule enforcement, concurrency handling, and scalable design.

21. Employee Timesheet and Payroll System

A web application where employees submit timesheets, and payroll is automatically calculated. It includes role-based dashboards for HR and finance. Focus: modular architecture, validation, and reporting.

22. Multi-Tenant CRM System

A customer relationship management platform for multiple organizations. Each tenant manages contacts, leads, and reports independently. Focus: database design for multi-tenancy and modular extensions.

23. Travel Booking System

A unified platform for booking flights, hotels, and car rentals. It should manage availability, reservations, and payment. Focus: integration across services and transactional integrity.

24. Help Desk Ticketing System

A platform for IT support teams to handle tickets, assign priorities, and track resolution times. Features include SLA enforcement and dashboards for performance metrics. Focus: workflow modeling and reporting.

25. Sports Tournament Management System

A system that manages tournament registration, match scheduling, standings, and results. It should support knockout, round-robin, and league formats. Focus: algorithmic scheduling and modular extensibility.

26. Multi-Language Content Management System (CMS)

A CMS where content can be published in multiple languages, with role-based permissions for editors and reviewers. Focus: extensible design, modularity, and internationalization support.

27. Alumni Networking Portal

A university alumni platform for connecting graduates, posting jobs, and organizing events. The system emphasizes role management, modular social features, and secure data handling.

28. Vehicle Rental System

A system where users can book vehicles (cars, bikes, vans) for rental. Features include booking calendars, penalties for late returns, and billing. Emphasis: concurrency, transactional design, and modularity.

29. Community Forum Platform

An online discussion platform with categories, threads, moderation tools, and reporting features. Focus: modular content management and scalable architecture.

29. Document Management and Workflow System

A platform for organizations to store, version, and route documents through approval workflows. Focus: layered architecture, versioning, and workflow automation.

30. Subscription-Based Video Streaming Service

A Netflix-like system supporting user subscriptions, video catalogs, and streaming history. Focus: modular service design, session handling, and database scalability.

31. Parking Lot Management System

A system for managing parking slots in multi-level lots. It should support reservations, billing, and real-time availability. Focus: concurrency handling and real-time dashboards.

32. University Hostel Management System

A system to allocate hostel rooms to students, manage fees, and track maintenance requests. Emphasis: rule enforcement, modular design, and reporting.

33. Multi-Level Marketing (MLM) Management System

A system to manage agents, sales tracking, and commission calculations in an MLM structure. Focus: hierarchical data modeling, reporting, and modular extensibility.

34. Project-Based Learning Platform

A platform where students can register for projects, collaborate, and submit deliverables. Instructors can monitor progress and provide feedback. Focus: workflow management and modular roles.

35. Subscription and Billing Management System

A system for managing recurring subscriptions (e.g., SaaS products), handling billing cycles, and generating invoices. Focus: transactional consistency and modular extensibility.

36. Multi-Campus University Management System

A system that integrates student records, courses, and staff across multiple campuses, supporting centralized and campus-specific modules. Focus: distributed database design and modular extensibility.

37. Bank Loan Management System

A web-based platform to process loan applications, approvals, and repayments. Emphasis: layered architecture, validation workflows, and secure transactions.

38. Contract Management System

A system that stores, versions, and tracks business contracts through their lifecycle (draft, review, approval, renewal). Focus: versioning, workflow automation, and reporting.

39. Job Recruitment and Applicant Tracking System

A platform where companies post jobs, and applicants apply. HR staff can track candidates through stages (applied, interviewed, hired). Emphasis: workflow modeling and modular role-based design.

40. Restaurant Reservation and Table Management System

A system for restaurants to manage online reservations, seating layouts, and cancellations. Focus: concurrency handling and usability design.

41. Supply Chain Tracking System

A system for tracking goods through suppliers, warehouses, distributors, and retailers. Emphasis: workflow automation, audit trails, and modular reporting.

42. E-Library Digital Repository

A platform to store, categorize, and access e-books, journals, and research papers. It should support role-based access, search, and versioning. Focus: modular content management and scalability.

43. Online Carpooling and Ride-Sharing System

A system to match drivers and passengers for shared rides, with booking and cancellation policies. Emphasis: concurrency, real-time updates, and security.

44. Clinic and Pharmacy Management System

A system that integrates clinic appointments with pharmacy dispensing and billing. Emphasis: transactional consistency, modular design, and usability.

45. Online Real Estate Property Management System

A platform for listing, renting, and selling properties with owner, tenant, and agent roles. Focus: modular workflows, transactional integrity, and extensibility.