

Course Code	:	MCS-213
Course Title	:	Software Engineering
Assignment Number	:	MCAOL(I)/213/Assign/2025
Maximum Marks	:	100
Weightage	:	40%
Last Dates for Submission	:	30th April 2025 (for January Session) 31st October 2025 (for July Session)

This assignment has one question for 80 marks. 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding assignments given in the Programme Guide for the format of presentation.

Q1:

Assume that you are assigned responsibility of developing an **Online Date Sheet Generation System (ODSGS)** for a University. **ODSGS** should run both on PCs and Mobile Devices. **ODSGS** will have fields such as Course Code, Course Title, Date of Exam, Time of Exam etc. The user of ODSGS will input the list of course codes and course titles and duration of examination. Sunday will be holiday. Starting date of exams will also be input by the user. There may be other information that is needed by ODSGS to generate the date sheet. So, you can list them and make provision for them in database structure. Make assumptions wherever necessary. Based on the inputs received, ODSGS should generate the Date Sheet.

For developing **ODSGS** as specified above,

- (a) Which SDLC paradigm will be selected. You may also suggest a SDLC paradigm that is proposed by you and non-existent as on date. Justify your answer. **(10 Marks)**
- (b) List the functional and non-functional requirements. **(10 Marks)**
- (c) Estimate the cost. **(15 Marks)**
- (d) Estimate the efforts. **(15 Marks)**
- (e) Develop SRS using IEEE format. **(15 Marks)**
- (f) List queries for whom Reports can be generated. **(5 Marks)**
- (g) List specific requirements which enables **ODSGS** to run on both PCs and Mobile Devices. **(10 Marks)**

MCS-213

SOLVED ASSIGNMENT 2025

Disclaimer/ Note: These Sample Answers/Solutions are prepared by Private Teacher/Tutors/Authors for the help and guidance of the student to get an idea of how he/she can answer the Questions given the Assignments. We do not claim 100% accuracy of these sample answers as these are based on the knowledge and capability of Private Teacher/Tutor. As these solutions and answers are prepared by the private teacher/tutor so the chances of error or mistake cannot be denied. Please consult your own Teacher/Tutor before you prepare a Particular Answer and for up-to-date and exact information, data and solution. Student should must read and refer the official study material provided by the university.

Q.1 - Assume that you are assigned responsibility of developing an Online Date Sheet Generation System (ODSGS) for a University. ODSGS should run both on PCs and Mobile Devices. ODSGS will have fields such as Course Code, Course Title, Date of Exam, Time of Exam etc. The user of ODSGS will input the list of course codes and course titles and duration of examination. Sunday will be holiday. Starting date of exams will also be input by the user. There may be other information that is needed by ODSGS to generate the date sheet. So, you can list them and make provision for them in database structure. Make assumptions wherever necessary. Based on the inputs received, ODSGS should generate the Date Sheet. For developing ODSGS as specified above,

(a) - Which SDLC paradigm will be selected. You may also suggest a SDLC paradigm that is proposed by you and non-existent as on date. Justify your answer.

ANS.- Selection of SDLC Paradigm for Online Date Sheet Generation System (ODSGS)

For developing the **Online Date Sheet Generation System (ODSGS)**, we must choose a **Software Development Life Cycle (SDLC) paradigm** that ensures **efficient planning, smooth execution, and flexibility for modifications**. Below are the possible SDLC paradigms that could be considered:

1. Waterfall Model

- The **Waterfall Model** follows a linear sequential approach where each phase (Requirement Gathering, Design, Implementation, Testing, Deployment, and Maintenance) must be completed before moving to the next.
- **Why not Waterfall?**
 - It lacks flexibility in accommodating changes after development starts.
 - ODSGS may require **modifications based on user feedback**, which is difficult to implement in Waterfall.

2. Agile Model (Recommended)

- The **Agile Model** is an iterative and incremental approach, ensuring frequent feedback from users.
- **Why Agile?**
 - Allows continuous updates and enhancements based on feedback.
 - Ensures early and frequent releases of functional components of ODSGS.
 - Facilitates testing in smaller iterations, improving quality and reliability.
 - Supports **cross-platform compatibility** (PCs and Mobile Devices).
- **Justification:**
 - As ODSGS must handle **dynamic scheduling rules**, an iterative approach allows modifications without major overhauls.
 - Enables university administration and examination department to give feedback on usability and improve system logic.

3. Proposed New SDLC Paradigm: Adaptive Dynamic Scheduling Model (ADSM)

- A **custom SDLC paradigm**, specifically designed for dynamic scheduling applications.
- **Key Features of ADSM:**
 - **AI-Driven Automation:** Automatically assigns exam slots based on course priorities and constraints.
 - **Real-time Adjustments:** Supports live updates to the schedule in case of administrative changes.
 - **Hybrid Approach:** Combines **Agile** for iterative feedback and **Prototyping** for rapid UI/UX development.
- **Why ADSM?**
 - Ideal for an exam scheduling system that requires **flexibility and automation**.
 - Ensures conflict-free scheduling by automatically resolving clashes.

- Provides **customization options** for universities with different scheduling policies.

Conclusion

For ODSGS, the **Agile Model** is the best existing SDLC approach due to its flexibility, iterative updates, and user involvement. However, a **newly proposed ADSM paradigm** could further enhance efficiency by integrating AI-driven scheduling and real-time adaptability, making it a more **tailored and innovative solution** for exam scheduling systems.

(b) - List the functional and non-functional requirements.

ANS.- Functional and Non-Functional Requirements for Online Date Sheet Generation System (ODSGS)

1. Functional Requirements (FRs)

These define the core functionalities of the system that ensure its proper working.

1. User Authentication and Authorization

- Admin login for authorized personnel (University Staff, Examination Controller).
- Secure access to modify, update, and generate date sheets.

2. Course and Exam Management

- Input Course Code, Course Title, and Examination Duration.
- Specify the **start date of exams** and automatically allocate exam dates.
- Ensure Sundays are **excluded** as exam days.
- Auto-adjust exam slots to avoid scheduling conflicts.

3. Date Sheet Generation

- Auto-generate the **exam timetable** based on input parameters.
- Provide options for **manual adjustments** if needed.
- Ensure that no two exams of the same course clash in timing.

4. Multi-Platform Compatibility

- System should be accessible on **PCs and mobile devices** (responsive design).

5. Notification and Alerts

- Notify students and faculty via email/SMS about exam schedules and changes.

6. Reporting and Printing

- Allow exporting the date sheet in **PDF, Excel, and print-friendly formats**.
- Generate custom reports for **departments and university administrators**.

7. Conflict Resolution

- Identify and highlight **scheduling conflicts** automatically.
- Provide suggestions for alternative exam slots.

2. Non-Functional Requirements (NFRs)

These define the quality attributes of the system.

1. Performance & Scalability

- The system should generate a **date sheet within seconds**, even for large universities.
- Support thousands of concurrent users without performance degradation.

2. Security

- **Encryption** for sensitive data like login credentials.
- **Role-based access control** to restrict unauthorized modifications.

3. Usability

- Simple **user interface (UI/UX)** for non-technical users.
- Accessible across different screen sizes (PCs, tablets, and mobile phones).

4. Reliability & Availability

- Ensure **99.9% uptime** so users can access the date sheet anytime.
- Implement **automated backups** to prevent data loss.

5. Maintainability & Flexibility

- Easy to update exam rules or add **new features** without major changes.

6. Compliance

- Follow **data privacy regulations** like GDPR for student data protection.

By ensuring these functional and non-functional requirements, ODSGS will be a **robust, efficient, and user-friendly** system.

(c) - Estimate the cost.

ANS.- Cost Estimation for Online Date Sheet Generation System (ODSGS)

The cost estimation of ODSGS depends on various factors such as **development team size, technology stack, hosting infrastructure, maintenance, and security**. Below is a detailed breakdown of estimated costs in Indian Rupees (INR).

1. Development Costs

Component	Estimated Cost (INR)
Project Manager (1) – Rs.1,50,000/month for 3 months	Rs.4,50,000
Frontend Developer (1) – Rs.80,000/month for 3 months	Rs.2,40,000
Backend Developer (1) – Rs.90,000/month for 3 months	Rs.2,70,000
Database Engineer (1) – Rs.75,000/month for 2 months	Rs.1,50,000
UI/UX Designer (1) – Rs.70,000/month for 2 months	Rs.1,40,000
QA Tester (1) – Rs.60,000/month for 2 months	Rs.1,20,000
Deployment Engineer (1) – Rs.65,000/month for 1 month	Rs.65,000

Component	Estimated Cost (INR)
Total Development Cost	Rs.13,35,000

2. Technology and Software Costs

Component	Estimated Cost (INR)
Frontend Framework (React/Angular/Vue)	Rs.50,000
Backend Framework (Node.js/Django/ASP.NET)	Rs.60,000
Database (MySQL/PostgreSQL/MongoDB)	Rs.40,000
Hosting & Server (AWS/Azure/Google Cloud – 1st Year)	Rs.1,20,000
Domain Registration (University Subdomain or Custom Domain for 1 Year)	Rs.5,000
SSL Certificate (Security for 1 Year)	Rs.3,000
APIs & Third-Party Integrations (Email/SMS Alerts)	Rs.25,000
Total Software & Technology Cost	Rs.3,03,000

3. Infrastructure & Hardware Costs

Component	Estimated Cost (INR)
Development Machines (Laptops, Desktops, Testing Devices, etc.)	Rs.2,00,000
Cloud Storage for Database Backups (1 Year)	Rs.50,000
Networking & Security Tools (Firewall, VPN, etc.)	Rs.75,000
Total Infrastructure & Hardware Cost	Rs.3,25,000

4. Testing & Security Costs

Component	Estimated Cost (INR)
Penetration Testing (To Identify Security Risks)	Rs.50,000
Performance & Load Testing (Ensuring System Scalability)	Rs.40,000
Bug Fixing & Regression Testing	Rs.30,000
Total Testing & Security Cost	Rs.1,20,000

5. Maintenance & Support (Annual Cost)

Component	Estimated Cost (INR)
Regular System Updates & Bug Fixes	Rs.1,00,000
Customer Support & Helpdesk (Technical Support Staff Salary for 1 Year)	Rs.2,00,000

Component	Estimated Cost (INR)
Hosting Renewal & Cloud Costs (2nd Year Onwards)	Rs.1,00,000
Total Maintenance & Support Cost	Rs.4,00,000

6. Miscellaneous Costs

Component	Estimated Cost (INR)
Training & Documentation for University Staff	Rs.75,000
Legal & Compliance Fees (GDPR, Data Protection Laws, etc.)	Rs.50,000
Marketing & Awareness (If Needed for User Adoption)	Rs.50,000
Total Miscellaneous Cost	Rs.1,75,000

Total Cost Estimation

Category	Total Cost (INR)
Development Costs	Rs.13,35,000
Technology & Software Costs	Rs.3,03,000
Infrastructure & Hardware Costs	Rs.3,25,000
Testing & Security Costs	Rs.1,20,000
Maintenance & Support (Annual)	Rs.4,00,000
Miscellaneous Costs	Rs.1,75,000
Total Estimated Cost (First Year)	Rs.26,58,000

Conclusion

The estimated initial investment for developing ODSGS is **Rs.26,58,000**. The annual **maintenance cost from the second year** onwards will be around **Rs.4,00,000 to Rs.5,00,000**, depending on updates and hosting renewals.

This cost estimation ensures a **scalable, secure, and efficient** system that meets the university's requirements for online date sheet generation.

(d) - Estimate the efforts.

ANS.- Effort Estimation for Online Date Sheet Generation System (ODSGS)

Effort estimation in software development helps determine the time and resources required to build the Online Date Sheet Generation System (ODSGS). It is typically calculated using **person-months (PM)**, which represents the effort a single developer puts in over a month.

1. Effort Estimation Approach

We use the **Work Breakdown Structure (WBS)** and **Function Point Analysis (FPA)** to estimate efforts.

The effort is divided among key development phases:

1. **Requirement Analysis & Planning**
2. **System & Database Design**
3. **Frontend & Backend Development**
4. **Integration & Testing**
5. **Deployment & Maintenance**

2. Breakdown of Efforts per Phase

Phase	Team Members Involved	Estimated Effort (Person-Months - PM)	Duration (Weeks)
Requirement Analysis & Planning	Project Manager (1), Business Analyst (1)	2.5 PM	3 Weeks
System & Database Design	System Architect (1), Database Engineer (1)	3.0 PM	4 Weeks
Frontend Development	Frontend Developer (1), UI/UX Designer (1)	4.5 PM	6 Weeks
Backend Development	Backend Developer (1), Database Engineer (1)	5.0 PM	6 Weeks
Integration & Testing	QA Engineer (1), DevOps Engineer (1)	3.5 PM	4 Weeks
Deployment & Maintenance	Deployment Engineer (1), Support Staff (1)	2.5 PM	3 Weeks

Total Estimated Effort: 21 PM (21 person-months)

Total Duration: 18-20 Weeks (Approx. 4.5 - 5 Months)

3. Detailed Breakdown of Efforts

1. Requirement Analysis & Planning (2.5 PM, 3 Weeks)

- Identifying functional and non-functional requirements.
- Discussing user needs, UI design, and system workflows.
- Documenting project scope and risk analysis.

2. System & Database Design (3.0 PM, 4 Weeks)

- Designing system architecture (client-server model).
- Database schema design for course codes, exam schedules, and constraints (like holidays).
- Creating API structures for seamless communication between frontend and backend.

3. Frontend Development (4.5 PM, 6 Weeks)

- Building user interface for PC and mobile compatibility.
- Implementing forms for inputting course details and exam duration.
- Designing dashboards and notification systems.

4. Backend Development (5.0 PM, 6 Weeks)

- Implementing core business logic (date sheet generation algorithm).
- Integrating backend APIs with the frontend.
- Ensuring role-based access and security measures.

5. Integration & Testing (3.5 PM, 4 Weeks)

- Unit testing and bug fixing.
- Performance testing for handling multiple users.
- Ensuring proper scheduling without clashes.

6. Deployment & Maintenance (2.5 PM, 3 Weeks)

- Deploying the system on cloud servers.
- Training university staff to use the system.
- Ongoing maintenance and security updates.

4. Factors Affecting Effort Estimation

1. **Complexity of Scheduling Logic** – If the date sheet generation algorithm requires additional constraints, it may increase development efforts.
2. **Scalability Requirements** – If the university expects large-scale usage, additional cloud configurations and performance optimizations may be needed.
3. **Security & Compliance** – Implementing advanced security features like two-factor authentication (2FA) and GDPR compliance may extend the timeline.
4. **Third-Party Integrations** – If external services like SMS/email notifications are used, additional integration efforts will be required.

5. Final Effort Estimation Summary

- **Total Effort: 21 person-months (PM)**
- **Total Duration: 18-20 weeks (~5 months)**
- **Team Size: 6-7 members working simultaneously**

By following this effort estimation, ODSGS can be developed efficiently within the given timeframe while ensuring **quality, performance, and scalability**.

(e) - Develop SRS using IEEE format.

ANS.- Software Requirements Specification (SRS) for Online Date Sheet Generation System (ODSGS)

IEEE Standard Format

1. Introduction

1.1 Purpose

The Online Date Sheet Generation System (ODSGS) is designed to automate the scheduling of university exams. It allows users to input course details, exam duration, and starting dates while ensuring no exams are scheduled on Sundays. The system will generate a clash-free date sheet that can be accessed via PCs and mobile devices.

1.2 Document Conventions

- Bold text is used for section headings.
- Italicized text is used for emphasis.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- University Administrators
- Software Developers
- Quality Assurance (QA) Engineers
- Project Managers

1.4 Product Scope

The system will:

- Generate an automatic date sheet.
- Prevent scheduling conflicts.
- Be accessible via web and mobile platforms.
- Store exam-related information securely.

1.5 References

- IEEE 830-1998 Software Requirements Specification Standard
- Database Management System (DBMS) Guidelines
- Web & Mobile Application Development Guidelines

2. Overall Description

2.1 Product Functions

- Input and store course details.
- Schedule exams automatically based on provided inputs.
- Ensure no exams are scheduled on Sundays.
- Display and export the generated date sheet.

2.2 User Characteristics

- **University Administrators:** Manage course and exam details.
- **IT Support Team:** Maintain and update the system.
- **Students:** View and download the date sheet.

2.3 Constraints

- Must run on both PCs and mobile devices.
- Exams cannot be scheduled on Sundays.
- System should support multiple universities.

2.4 Assumptions and Dependencies

- The university provides accurate input data.
- The system is hosted on a reliable server.

3. Specific Requirements

3.1 Functional Requirements

- **FR1:** The system shall allow users to input course codes, titles, and exam duration.
- **FR2:** The system shall generate a date sheet without clashes.
- **FR3:** The system shall ensure exams are not scheduled on Sundays.
- **FR4:** The system shall provide export options (PDF/Excel).
- **FR5:** The system shall send notifications to students.

3.2 Non-Functional Requirements

- **Performance:** The system should generate a date sheet within 5 seconds.
- **Scalability:** Must support up to 10,000 concurrent users.
- **Security:** Encrypted data storage and role-based access.
- **Availability:** 99.9% uptime.

4. Appendices

- Database schema details
- UI wireframes
- API documentation

This SRS ensures the structured development of ODSGS, meeting both functional and performance expectations.

(f) - List queries for whom Reports can be generated.

ANS.- Queries for Report Generation in Online Date Sheet Generation System (ODSGS)

The system will allow university administrators to generate various reports for efficient exam scheduling and management. Below are some key queries for which reports can be generated:

1. **Exam Schedule Report** – Displays the complete date sheet, including course codes, course titles, exam dates, and times.

2. **Course-Wise Exam Report** – Lists all exams scheduled for a specific course.
3. **Student-Specific Exam Report** – Generates an individual student's exam schedule based on their enrolled courses.
4. **Department-Wise Exam Report** – Provides a list of exams scheduled for a particular department.
5. **Faculty-Wise Exam Report** – Shows exams assigned to specific faculty members.
6. **Date-Wise Exam Report** – Lists all exams happening on a specific date.
7. **Clash Detection Report** – Highlights any scheduling conflicts or overlapping exams.
8. **Room Allocation Report** – Displays exam halls assigned for different exams.
9. **Examiner Duty Report** – Lists faculty members assigned as invigilators for specific exams.
10. **Rescheduled Exams Report** – Shows any exams that were rescheduled due to conflicts or other issues.

These reports will help administrators and faculty manage and track exam schedules efficiently, ensuring smooth examination processes.

(g) - List specific requirements which enables ODSGS to run on both PCs and Mobile Devices.

ANS.- Specific Requirements for Running ODSGS on Both PCs and Mobile Devices

To ensure the **Online Date Sheet Generation System (ODSGS)** runs smoothly on both **PCs and mobile devices**, several technical and functional requirements must be implemented. Below are the key requirements categorized under different aspects:

1. Platform Compatibility Requirements

- The system must be a **responsive web application** using HTML5, CSS3, and JavaScript frameworks like **Bootstrap** to ensure smooth rendering across different screen sizes.
- It should be developed using **Progressive Web Application (PWA)** principles to function like a native app on mobile devices.
- The application should be cross-browser compatible and optimized for **Google Chrome, Mozilla Firefox, Safari, Microsoft Edge**, and mobile browsers.

2. User Interface (UI) Requirements

- The UI should support **adaptive layouts** for different screen resolutions using CSS media queries.
- The system should provide a **mobile-friendly navigation** menu with touch-friendly elements.
- Forms and input fields should be **optimized for touchscreens**, with appropriate font sizes and spacing.

3. Functional Requirements

- The system should allow **role-based access control** (admin, faculty, student) on both PC and mobile interfaces.
- Users should be able to **input and modify exam details** from any device.

- The generated **date sheet should be downloadable** in multiple formats (PDF, Excel) for both PC and mobile users.
- Notifications regarding **exam schedules and changes** should be sent via email and SMS, ensuring mobile accessibility.

4. Performance & Security Requirements

- The system should use a **cloud-based database** to ensure synchronization between PC and mobile devices.
- The system should support **offline access** for students and faculty using service workers (PWA feature).
- **Secure authentication mechanisms** (OTP-based login, Two-Factor Authentication) should be implemented for access on both platforms.

5. API & Integration Requirements

- The backend should provide a **RESTful API** to allow seamless data exchange between PC and mobile versions.
- Integration with **Google Calendar and iCal** should be available for users to sync their exam schedules.

By implementing these specific requirements, ODSGS will function efficiently on **both PCs and mobile devices**, ensuring flexibility and accessibility for all users.