

REVISED COURSE OUTLINE

CSE-2204: SYSTEM ANALYSIS AND DESIGN

Item	Details
Course Code	CSE-2204
BNQF Code	06112204
Course Title	System Analysis and Design
Credits / Contact Hours	3 Credits / 3 Hours per Week
Total Marks	100
Year / Semester	2nd Year / 2nd Semester
Course Type	Departmental Course
Pre-requisite	Data Structures (CSE-1201)
Teaching Duration	14 Weeks
Number of Classes	28 Classes, 2 Classes per Week

1. Course Description

This course introduces the principles, methods, tools, and techniques used to analyze, design, develop, test, implement, maintain, and evolve information systems. The course covers the System Development Life Cycle (SDLC), business process analysis, feasibility studies, requirements engineering, system modeling, database design, user interface design, system architecture, Agile and Scrum, object-oriented analysis and design, implementation, software testing, security and privacy, maintenance, configuration and change management, documentation, and emerging technologies. The course emphasizes practical system analysis and design through case studies, modeling exercises, and a system design project.

2. Course Objectives

- Introduce fundamental concepts of systems, system analysis, system design, and the SDLC.
- Develop students' ability to analyze business processes and identify system requirements.
- Develop practical skills in requirements elicitation, analysis, validation, and documentation.
- Develop competence in DFD, ERD, UML use case, and class diagram modeling.
- Introduce system architecture, database design, UI design, performance, scalability, security, and privacy.
- Explain Waterfall, iterative, Agile, and Scrum-based development approaches.
- Develop knowledge of implementation, software testing, quality assurance, UAT, maintenance, and change management.
- Enable students to evaluate an existing system and recommend maintenance, modification, replacement, or redesign.

3. Detailed Course Contents

No.	Topic	Major Contents
1	Introduction to System Analysis and Design	System concepts; information systems; system analyst role; system analysis and design; SDLC stages; stakeholders.
2	Understanding Business Processes	Business processes; workflows; process mapping; process analysis; bottlenecks; process improvement.
3	Project Initiation and Feasibility Study	Project initiation; scope; objectives; stakeholders; technical, economic, operational, legal, and schedule feasibility; cost-benefit analysis.
4	Requirements Elicitation and Analysis	Functional and non-functional requirements; interviews; surveys; observation; workshops; prototyping; prioritization; validation.
5	Requirements Documentation	SRS; use cases; user stories; acceptance criteria; requirements traceability matrix; requirements validation and change control.
6	Data Flow Diagrams	Context diagram; level-0 and lower-level DFDs; data flows; data stores; external entities; balancing; modeling errors.
7	Entity-Relationship Diagrams	Entities; attributes; relationships; cardinality; participation; conceptual data modeling; ER-to-relational mapping.
8	Use Case and Class Diagrams	UML fundamentals; actors; use cases; include/extend; classes; attributes; operations; associations; aggregation; composition; inheritance.
9	System Architecture and Design Principles	Layered, client-server, and three-tier architectures; hardware; software; network; modularity; cohesion; coupling; maintainability.

10	User Interface Design	User-centered design; usability; consistency; navigation; accessibility; prototyping; usability testing.
11	Database Design	Conceptual, logical, and physical design; keys; relational schema; normalization up to 3NF; integrity constraints; database security.
12	Performance and Scalability	Performance requirements; response time; throughput; capacity planning; bottlenecks; caching; load balancing; horizontal and vertical scaling.
13	System Development Methodologies	Waterfall; V-Model; iterative; incremental; prototyping; spiral; Agile; methodology selection.
14	Agile Development and Scrum	Agile principles; Scrum roles; backlog; sprint planning; daily Scrum; review; retrospective; user stories.
15	Object-Oriented Analysis and Design	OOAD principles; abstraction; encapsulation; inheritance; polymorphism; UML-based analysis and design.
16	System Implementation	Implementation planning; deployment; data migration; training; changeover methods; release planning.
17	Quality Assurance and System Testing	Software quality; QA versus QC; test planning; test cases; black-box and white-box testing; regression testing; defect management.
18	Test-Driven Development	TDD concept; red-green-refactor cycle; test-first development; benefits and limitations.
19	User Acceptance Testing	UAT planning; scenarios; acceptance criteria; user involvement; execution; defect resolution; acceptance decision.
20	System Maintenance and Evolution	Corrective, adaptive, perfective, and preventive maintenance; bug fixing; enhancements; upgrades; legacy systems; replacement decisions.
21	System Security and Privacy	Threats; vulnerabilities; risk assessment; authentication; authorization; access control; data protection; privacy; secure design.
22	Ethical and Social Considerations	Professional ethics; privacy; fairness; transparency; accessibility; social impact; data protection; legal requirements.
23	Configuration and Change Management	Configuration items; baselines; version control; change requests; impact analysis; approval; release management.
24	System Documentation	Requirements; analysis and design; technical documentation; user manuals; installation and maintenance documentation; standards.
25	Emerging Technologies and Trends	Cloud computing; IoT; AI and ML; mobile/web systems; microservices; low-code/no-code; DevOps; modern system architecture.

4. 14-Week / 28-Class Teaching Plan

Week	Class	Topic	Coverage
1	Class 1	Introduction to System Analysis and Design	System concepts; information systems; role of system analyst; course overview.
1	Class 2	System Development Life Cycle	SDLC stages; stakeholders; success and failure factors; overview of development approaches.
2	Class 3	Business Process Analysis	Business processes, workflows, process mapping, process improvement.
2	Class 4	Project Initiation and Feasibility	Project scope, objectives, stakeholders, feasibility types, cost-benefit analysis.
3	Class 5	Requirements Elicitation	Interviews, surveys, observation, workshops, document analysis.
3	Class 6	Requirements Analysis	Functional/non-functional requirements, prioritization, validation, conflict resolution.
4	Class 7	Requirements Documentation	SRS, use cases, user stories, acceptance criteria.
4	Class 8	Requirements Traceability and Case Exercise	RTM, requirement validation, change control, practical requirements exercise.
5	Class 9	Data Flow Diagrams I	Context diagram, external entities, processes, data flows, data stores.
5	Class 10	Data Flow Diagrams II	Level-0 and lower-level DFDs, balancing, common errors, practical modeling.
6	Class 11	ER Diagrams I	Entities, attributes, relationships, cardinality, participation.
6	Class 12	ER Diagrams II and Database Modeling	ER-to-relational mapping, keys, integrity constraints, practical ERD.
7	Class 13	UML and Use Case Diagrams	UML basics, actors, use cases, relationships, include and extend.
7	Class 14	Class Diagrams and OO Modeling	Classes, attributes, methods, associations, aggregation, composition, inheritance.
8	Class 15	System Architecture	Architectural concepts, layered, client-server, three-tier architecture, deployment considerations.
8	Class 16	Design Principles and Scalability	Modularity, cohesion, coupling, maintainability, performance, scalability, availability.
9	Class 17	User Interface Design	User-centered design, usability, navigation, accessibility, UI principles.
9	Class 18	Database Design and Normalization	Logical/physical design, relational schema, normalization up to 3NF.
10	Class 19	Development Methodologies	Waterfall, V-Model, iterative, incremental, spiral, prototyping.
10	Class 20	Agile and Scrum	Agile principles, Scrum roles, backlog, sprint, review, retrospective.
11	Class 21	Object-Oriented Analysis and Design	OOAD principles, UML, abstraction, encapsulation, inheritance, polymorphism.
11	Class 22	System Implementation and Deployment	Implementation planning, migration, training, deployment and changeover.
12	Class 23	Quality Assurance and Testing	QA/QC, test planning, unit, integration, system testing, defect management.
12	Class 24	TDD and User Acceptance Testing	TDD cycle, test cases, UAT planning, acceptance criteria, UAT execution.
13	Class 25	Maintenance and System Evolution	Corrective, adaptive, perfective, preventive maintenance; legacy systems; replacement decisions.
13	Class 26	Security, Privacy, Ethics and Change	Risk assessment, security controls, privacy, ethics, configuration and change

		Management	management.
14	Class 27	Documentation and Emerging Technologies	System documentation; cloud, IoT, AI/ML, microservices, DevOps, current trends.
14	Class 28	Integrated Case Study / Project Presentation	End-to-end analysis and design review; DFD, ERD, UML, architecture, requirements, testing and recommendations.

5. Course Learning Outcomes (CLOs)

CLO	Learning Outcome
CLO1	Define system, system analysis, system design, and other fundamental system concepts.
CLO2	Explain the System Development Life Cycle and major stages of system development.
CLO3	Demonstrate the processes and stages of system analysis and design using appropriate techniques and models.
CLO4	Analyze the factors that contribute to the success or failure of an information system.
CLO5	Evaluate an existing system and determine whether it should be maintained, modified, replaced, or redesigned based on technical, economic, operational, security, and user requirements.

6. Teaching-Learning Methods

- Interactive lectures and classroom discussion
- Case studies and real-world system examples
- Hands-on DFD, ERD, UML, database, and UI modeling exercises
- Requirements elicitation and business process exercises
- Group-based system analysis and design project
- Short quizzes, assignments, presentations, and peer review
- Use of appropriate diagramming, prototyping, database, and project management tools

7. Assessment Strategy

Component	Marks	Indicative Coverage
Continuous Assessment (CT, Attn., etc.)	10	Quizzes, assignments, class activities, case studies, modeling exercises
Mid-Term Examination, 2	20	Introduction, SDLC, feasibility, requirements, DFD, ERD, UML
Project / Presentation/Assignment	10	System analysis and design project, documentation, models, presentation
Final Examination	60	Comprehensive assessment of analysis, design, methodologies, testing, security, maintenance, and emerging trends
Total	100	

8. Suggested Practical / Project Work

Students will analyze and design a real-world information system. Examples include a university student management system, hospital management system, library management system, e-commerce system, banking system, transportation management system, or online examination system.

- Problem identification and project scope
- Stakeholder identification and business process analysis
- Feasibility study
- Requirements elicitation and SRS
- Use cases and user stories
- DFD and ERD
- UML use case and class diagrams
- System architecture and database design
- UI prototype
- Security and privacy considerations
- Test plan, test cases and UAT plan
- Final system analysis and design documentation and presentation

9. Suggested Software / Tools

- diagrams.net (Draw.io), Lucidchart, or equivalent tools for DFD, ERD and UML
- Figma or equivalent for UI prototyping
- MySQL Workbench or equivalent database modeling tools
- Git and GitHub/GitLab for configuration and version management
- Jira, Trello, or equivalent Agile/Scrum project management tools

10. Recommended References

1. Dennis, A., Wixom, B. H., & Tegarden, D. Systems Analysis and Design: An Object-Oriented Approach with UML.
2. Kendall, K. E., & Kendall, J. E. Systems Analysis and Design.
3. Satzinger, J. W., Jackson, R. B., & Burd, S. D. Systems Analysis and Design in a Changing World.
4. Pressman, R. S., & Maxim, B. R. Software Engineering: A Practitioner's Approach.
5. Sommerville, I. Software Engineering.
6. Object Management Group (OMG). Unified Modeling Language (UML) specifications and related standards.

11. CLO Assessment Alignment

CLO	Primary Evidence	Assessment
CLO1	System concepts and terminology	Quiz, Mid-Term, Final
CLO2	SDLC and development methodologies	Assignment, Mid-Term, Final
CLO3	System analysis and design processes and models	Project, Practical, Mid-Term, Final
CLO4	System success/failure analysis	Case Study, Project, Final
CLO5	Existing system evaluation and replacement/redesign decisions	Project, Case Study, Final

12. Alignment with the Provided Course Materials

The 28-class schedule has been organized so that the supplied course-content topics can be covered within 14 teaching weeks, with two classes per week. The sequence follows a logical progression from system concepts and SDLC to requirements, modeling, architecture, design, development methodology, testing, maintenance, security, documentation, and emerging technologies.

Google Drive course-material link provided by the instructor:

<https://drive.google.com/drive/folders/1wI53zT0uTDnvwdh8QuQ4Y79LZewDB92y?usp=sharing>

Note: The shared Drive folder could be opened as a Drive landing page, but its individual files and contents were not exposed to the document-generation environment. Therefore, this revision is aligned to the detailed course contents supplied in the prompt. If the actual lecture files are uploaded here, the 28 classes can be mapped precisely to the existing slides, chapters, examples, and exercises.

13. Expected Course Competencies

- Analyze an organizational problem and identify appropriate system requirements.
- Conduct a basic feasibility study and justify a proposed system.
- Elicit, analyze, validate, prioritize, and document requirements.
- Develop DFD, ERD, UML use case, and class diagrams.
- Design basic system architecture, database structures, and user interfaces.
- Select an appropriate development methodology for a given project.
- Plan testing, UAT, maintenance, security, privacy, and change management activities.
- Evaluate an existing system and recommend improvement, replacement, or redesign.