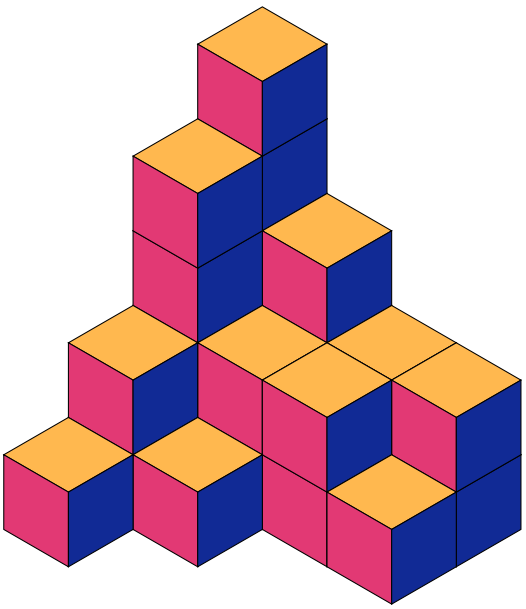
# CSE 4407 Taskbook

Computer system analysis is like child-rearing; you can do grievous damage, but you cannot ensure success.

Tom DeMarco



Department of CSE
Islamic University of Technology

# Contents


**Lab 1 Systems, Roles, and Development Methodologies** **5**

1 Tasks List . . . 5
- 1.1 Task 1: Finalize Group and Select Organization . . . 5
- 1.2 Task 2: Preliminary Organizational Assessment . . . 5
- 1.3 Task 3: Assess Need for Systems Analyst Role . . . 5
- 1.4 Task 4: Evaluate and Select Development Methodology . . . 6

2 Presentation Requirements . . . 6

3 Presentation Rubrics . . . 6

4 Deliverables . . . 7

**Lab 2 Understanding and Modeling Organizational Systems** **8**

1 Tasks List . . . 8
- 1.1 Task 1: Revisit Organizational Context and Define System Scope . . . 8
- 1.2 Task 2: Develop a Context-Level Data Flow Diagram (DFD) . . . 8
- 1.3 Task 3: Develop a Preliminary Entity-Relationship (E-R) Diagram . . . 8
- 1.4 Task 4: Develop a High-Level Use Case Diagram . . . 9
- 1.5 Task 5: Analyze Management Levels and Organizational Cultures . . . 9

2 Presentation Requirements . . . 9

3 Report Requirements . . . 9

4 Presentation Rubrics . . . 10

5 Report Rubrics . . . 11

6 Deliverables . . . 11

**Lab 3 Project Management** **12**

1 Tasks List . . . 12
- 1.1 Task 1: Refine Problem Definition and Justify Project Selection . . . 12
- 1.2 Task 2: Conduct TEO Feasibility Analysis and Estimate Workloads . . . 12
- 1.3 Task 3: Forecast Costs and Benefits with Break-Even and Payback Analyses . . . 13

2 Presentation Requirements . . . 13

3 Report Requirements . . . 13

4 Presentation Rubrics . . . 14

5 Report Rubrics . . . 15

6 Deliverables . . . 15

**Lab 4 Project Management** **16**

1 Tasks List . . . 16
- 1.1 Task 1: Develop Initial Project Plan and Schedule . . . 16
- 1.2 Task 2: Consider Team and Risks . . . 16
- 1.3 Task 3: Draft Project Charter and Proposal Outline . . . 16

2 Presentation Requirements . . . 16

3 Report Requirements . . . 17

4 Presentation Rubrics . . . 17

5 Report Rubrics . . . 18

6 Deliverables . . . 18

7 Bonus Task: Complex Engineering Problem . . . 18
7.1 Overview . . . 18
7.2 What you have to do . . . 20
7.3 EA1: Range of Resources . . . 20
7.4 EA2: Level of Interactions . . . 22
7.5 EA3: Innovation . . . 23
7.6 Consequences to Society/Environment . . . 24
7.7 EA5: Familiarity . . . 25
7.8 Putting it all together . . . 27
7.9 tl;dr . . . 27

**Lab 5 Information Gathering: Interactive Methods 28**
1 Tasks List . . . 28
1.1 Task 1: Plan Stakeholder Interviews . . . 28
1.2 Task 2: Design a User Questionnaire . . . 28
1.3 Task 3: Perform Requirement Elicitation . . . 29
1.4 Task 4: Consider Other Interactive Methods . . . 29
2 Presentation Requirements . . . 29
3 Report Requirements . . . 29
4 Presentation Rubrics . . . 30
5 Report Rubrics . . . 30
6 Deliverables . . . 31

**Lab 6 Information Gathering: Unobtrusive Methods 32**
1 Tasks List . . . 32
1.1 Task 1: Sampling Strategy Design . . . 32
1.2 Task 2: Investigation Plan for Existing Data . . . 32
1.3 Task 3: Observation Planning – Analyst's Playscript . . . 33
1.4 Task 4: Observation Planning – STROBE . . . 33
1.5 Task 5: Synthesis with Interactive Methods . . . 33
2 Presentation Requirements . . . 33
3 Report Requirements . . . 33
4 Presentation Rubrics . . . 34
5 Report Rubrics . . . 35
6 Deliverables . . . 35

**Lab 7 Agile Modeling, Prototyping, and Scrum 36**
1 Tasks List . . . 36
1.1 Task 1: User Story Development and Prioritization . . . 36
1.2 Task 2: Prototyping / Mockup Creation . . . 37
1.3 Task 3: Agile Planning Simulation (Scrum Basics) . . . 37
1.4 Task 4: Agile vs. Structured Reflection . . . 37
2 Presentation Requirements . . . 38
3 Report Requirements . . . 38
4 Presentation Rubrics . . . 39
5 Report Rubrics . . . 39
6 Deliverables . . . 40

**Lab A Presentation Guidelines** **41**
1 Planning and Structure . . . . . 41
2 Slide Design . . . . . 41
3 Practice and Preparation . . . . . 41
4 Delivery and Professionalism . . . . . 41
5 Time and Q&A Management . . . . . 42
6 Technical and Logistical Preparation . . . . . 42
7 Engagement and Accessibility . . . . . 42

# Lab 1 Systems, Roles, and Development Methodologies

Welcome to your first lab task for CSE 4408! This lab serves as the crucial foundation for your semester-long project, where you will gradually design and propose an information system. Over the next week, your group will select a real-world organization, analyze the potential role and need for systems analysis within it, and apply the concepts learned in Chapter 1 to determine the most suitable systems development methodology for initiating a project in that context. This lab will help you think critically about how System Analysis and Design principles apply in practice before we dive into more technical analysis and design techniques in later chapters.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: Finalize Group and Select Organization

Form your project team of 3–4 members within your lab section. Because you will collaborate closely and present as one unit, it is essential to finalize your group composition early. Please submit the full list of members (names and student IDs) via this Google Form by Friday, May 09, 2025 if you have not already done so.

Once your team is set, choose one organization or a specific department/unit within a larger organization as the focus of your information-systems project. To facilitate data gathering, we recommend selecting a unit associated with IUT (for example, the Central Library, the Laundry, the Medical Center, the ICT Center, the Office of the Registrar, etc.). If you opt for an external organization (e.g., a local business or non-profit), be sure you can access enough preliminary information on its operations and challenges. In that case, include a brief justification in your presentation explaining your choice and how you anticipate gathering needed information.

### 1.2 Task 2: Preliminary Organizational Assessment

Begin by gathering basic descriptive information about your chosen entity. What is its core mission or primary function? Who are the key stakeholders (such as, end-users, customers, or managers) whose needs drive the system? Identify any high-level goals (e.g., improving service turnaround, ensuring regulatory compliance) or challenges (e.g., outdated processes, data silos) that you suspect the organization faces. Identify the size, scope of operations, and environment of the organization. You may need to make reasonable assumptions based on public information, preliminary interviews, or your own domain knowledge.

### 1.3 Task 3: Assess Need for Systems Analyst Role

Using your organizational assessment, pinpoint specific problems or opportunities where better information systems or process redesign could add value. For instance, is there an inefficient paper-based workflow, or are reporting delays hindering decision-making?

Investigate whether an explicit Systems Analyst role already exists within the organization (this may require a quick scan of organogram charts or job postings). If such a role is not readily identifiable, build a case for why systems-analysis expertise is needed in the context of your organization.

Next, map the organization's needs to the three primary Systems Analyst personas from the book:

1. Consultant: Advises on best practices and high-level strategy.
2. Supporting Expert: Provides technical know-how and documentation.
3. Agent of Change: Champions process improvement and user adoption.

Argue which of these roles would be most valuable in your context and why. Finally, highlight the critical personal qualities (e.g., strong problem-solving skills, clear communication, stakeholder empathy) that a successful analyst must bring to this engagement.

## 1.4 Task 4: Evaluate and Select Development Methodology

Consider the unique characteristics of your chosen organization:

- ▷ Size and stability: Is it a large, well-established unit or a small, agile team?
- ▷ Data sensitivity: Does it handle regulated or confidential information?
- ▷ Change dynamics: Are requirements likely to shift rapidly?
- ▷ Regulatory/documentation demands: Is heavy documentation mandated?
- ▷ Cost and reuse: Is budget a primary driver, and can existing components be leveraged?

Review the three major methodologies introduced in Chapter 1: SDLC, Agile, and O-O and evaluate each against your organization's profile. Then select the single methodology you believe best suits an initial systems project. Provide a strong, book-based justification linking specific methodology characteristics to your organization's needs.
Equally important, explain why the other two methodologies are less appropriate in this scenario. For example, if rapid iteration is critical, you might rule out a heavyweight SDLC; if documentation compliance is paramount, you might deem pure Agile too light. Your rationale should demonstrate clear alignment between theory and the real-world context you've researched.

# 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ Title Slide: Course Name, Lab Title, Group Name, Members List, Date
- ▷ Introduction: Briefly introduce your chosen organization/department and its main purpose/context. Justify if choosing an external organization
- ▷ Need for SAD/Analyst Role: Discuss potential problems/opportunities justifying SAD. Argue for the need for analyst expertise for the organization. Identify the most relevant analyst roles and key analyst qualities needed for this context.
- ▷ Recommended Methodology and Justification: Clearly state your chosen primary methodology. Provide detailed justification based on Chapter 1 guidelines and organizational context.
- ▷ Argument against Other Methodologies: Briefly explain why *each* of the remaining two methodologies might not be suitable for this organization.
- ▷ Conclusion: Briefly summarize your key findings and the rationale for your methodology choice.
- ▷ Q&A Slide: A slide indicating you are ready for questions.

# 3 Presentation Rubrics

Your presentation will be evaluated based on:

- ▷ Clarity and sufficiency of the chosen organization's description and context.
- ▷ Depth and thoughtfulness of the analysis regarding the need for SAD and the relevant analyst roles/qualities.
- ▷ Strength, clarity, and logical consistency of the justification for the selected methodology (including why others were rejected).
- ▷ Thoroughness of the evaluation of the remaining *two* methodologies in the context of the chosen organization.
- ▷ Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 4 Deliverables

Convert your presentation slides to a PDF file and submit it in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L1_Slides.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

# Lab 2 Understanding and Modeling Organizational Systems

Welcome to Lab 2! Here, we will apply the concepts from Chapter 2 to gain a deeper, shared understanding of your chosen organization as a system. This lab focuses on creating initial graphical models (Context-Level Data Flow Diagrams (DFDs), Entity-Relationship (E-R) diagrams, and Use Case Diagrams) to define the preliminary scope of your potential system and visualize its key components and interactions. These models will serve as foundational artifacts for your ongoing project. Remember, these are *initial models*; they will evolve as your analysis deepens.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: Revisit Organizational Context and Define System Scope

Begin this lab by reviewing the organizational unit your group selected in Lab 1. Summarize the core function of the organization and revisit the key problem, need, or opportunity previously identified. This is essential for ensuring your models are grounded in a realistic context.

Next, come to a group consensus on the initial, high-level boundary of the system you intend to analyze and model. Define what aspects or processes will be in scope, and just as importantly, identify what is out of scope for your current system proposal. This boundary-setting exercise will help focus your models on what truly matters to the organization while avoiding unnecessary complexity. You need not include these findings in the presentation.

### 1.2 Task 2: Develop a Context-Level Data Flow Diagram (DFD)

With your system boundary defined, begin designing a Context-Level DFD to represent the system as a single process interacting with its external environment.

Start by identifying that one process which represents the entire proposed system. Then, determine the key external entities (people, departments, or other systems) that will provide inputs to or receive outputs from your system. These should be grounded in your earlier organizational assessment.

Next, determine the major data flows between these entities and your system. Focus on capturing essential inputs and outputs that support core functions. Finally, draw the Context-Level DFD using the standard notation: one process (a circle), multiple external entities (rectangles), and directional arrows showing data flows. This diagram should offer a clean, high-level snapshot of the system's interaction with its environment.

### 1.3 Task 3: Develop a Preliminary Entity-Relationship (E-R) Diagram

Shift your focus to the data model by brainstorming the core entities about which the system will need to store information. These may include people (e.g., Students), events (e.g., Requests), or objects (e.g., Equipment). From this session, select five to seven key entities that are central to the problem or opportunity you are addressing.

Identify the primary relationships among these entities reflecting actions or associations, such as an employee submitting a request or a user borrowing a resource. For each relationship, determine the appropriate cardinality (e.g., one-to-many, many-to-many) and use Crow's Foot notation to represent them accurately.

Construct the preliminary E-R diagram using standard conventions: entities as rectangles, relationships as labeled lines, and appropriate cardinality indicators. This model should begin to define how information will be structured within your proposed system.

### 1.4 Task 4: Develop a High-Level Use Case Diagram

Consider the functional perspective of your system. Identify 3 to 5 primary actors (roles, not individual people) who will interact directly with the system to achieve specific goals. Refer to your DFD to see if any external entities might also serve as actors.

For each actor, define their main goals or tasks they aim to accomplish using the system. These high-level functions will become your use cases. Use clear verb-noun naming (e.g., "Submit Request," "Generate Report") to describe each one. Aim to identify 5 to 7 use cases that capture the most critical system functionality.

Draw the Use Case diagram showing the system boundary, actors (as stick figures), use cases (as labeled ovals), and '`communicates`' relationships (solid lines). For this initial model, you may skip optional relationships like `«include»`, `«extend»`, and generalizations aside unless they naturally emerge.

### 1.5 Task 5: Analyze Management Levels and Organizational Cultures

Conduct a brief but thoughtful analysis of the management structure within your chosen organization. Identify which levels (Operational, Middle, or Strategic Management) are likely to use the system, be affected by it, or influence its design. For each, consider what types of information they may require (e.g., transaction logs for operational staff, performance metrics for middle managers, summary dashboards for executives).

Additionally, reflect on the organizational culture or subcultures you have observed or inferred. How open is the unit to technological change? Are communication channels formal or informal? Is there evidence of resistance to change or a strong preference for manual processes? These cultural insights are important for understanding potential risks and planning for system adoption and user training.

## 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ Briefly recap your organization and the potential system's focus.
- ▷ Present and explain your:
  - ◇ Context-Level DFD
  - ◇ Preliminary E-R Diagram
  - ◇ High-level Use Case Diagram
- ▷ Briefly discuss your analysis of relevant management levels and cultural factors.

## 3 Report Requirements

For this lab, your report must include the following information:

- ▷ Cover Page
  - ◇ Group member names and student IDs

- ◇ Course code and title (CSE 4408: System Analysis and Design Lab)
- ◇ Lab number (Lab 2)

▷ Organizational Context and System Scope

- ◇ A concise summary of your selected organizational unit's core function
- ◇ A restatement of the key problem, need, or opportunity identified in Lab 1
- ◇ A clear definition of the proposed system boundary explaining what is in scope (list of processes, functions, or data included) and out of scope (list of processes, functions, or data explicitly excluded)
- ◇ A brief rationale explaining how this boundary focuses on solving the organization's core issue

▷ Context-Level Data Flow Diagram

- ◇ The properly drawn diagram
- ◇ A short narrative explaining each external entity and the key inputs/outputs

▷ Preliminary E-R Diagram

- ◇ A list of the chosen entities with brief definitions
- ◇ The properly drawn diagram
- ◇ A narrative (4-5 sentences) describing the most critical relationships and cardinalities

▷ High-Level Use Case Diagram

- ◇ A list of the primary actors and their roles
- ◇ A list of use cases
- ◇ The properly drawn diagram
- ◇ A brief explanation of how the chosen actors and use cases support core system functionality

▷ Management Levels and Organizational Culture Analysis

- ◇ Identification of the management level(s) that will use or be affected by the system
- ◇ A table/list matching each level to its primary information requirements
- ◇ A reflective discussion on the organization's culture

▷ Appendices: Any supplementary notes, glossaries, or supporting artifacts

▷ Formatting

- ◇ Use consistent, professional formatting (A4 paper, 12pt font, 1.5 line spacing, 1-inch margins)
- ◇ Number all figures and tables sequentially; include captions

Ensure that each diagram is clearly labeled and accompanied by explanatory text so a reader unfamiliar with your group's discussions can understand your models and analyses.

## 4 Presentation Rubrics

Your presentation will be evaluated based on:

▷ Context-Level DFD: Correct use of symbols, clear definition of boundary, appropriate identification of key external entities and major data flows relevant to the scope.

▷ Preliminary E-R Diagram: Identification of relevant key entities, clear depiction of relationships, correct application of Crow's Foot notation for cardinality.

▷ High-Level Use Case Diagram: Identification of relevant primary actors and key use cases, correct use

of symbols, clear representation of user goals related to the scope.

- ▷ Scope, Management Levels and Organizational Cultures: Quality and insightfulness of the explanation of scope, analysis of management levels, and consideration of cultural factors.
- ▷ Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 5 Report Rubrics

Your report will be evaluated based on:

- ▷ Organizational Context and System Scope: Clarity and completeness in summarizing the organizational unit's core function, restating the key problem/opportunity, and defining what's in-scope vs. out-of-scope, with a rationale.
- ▷ Context-Level Data Flow Diagram (DFD): Quality of the context-level DFD and its narrative.
- ▷ Preliminary E-R Diagram: Identification of entities, relationships, and proper Crow's Foot notation.
- ▷ High-Level Use Case Diagram: Identification of actors and use cases, and diagram clarity.
- ▷ Management Levels and Organizational Culture Analysis: Insight into who uses/influences the system and cultural factors affecting adoption.
- ▷ Presentation and Formatting: Professional consistency, cover page accuracy, figure/table numbering, and overall readability.

## 6 Deliverables

Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L2_Slides.pdf` and `[WWW_XXX_YYY_ZZZ]_L2_Report.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

# Lab 3 Project Management

This lab marks the culmination of the initial project planning phases. Considering the amount of work involved, this topic will be covered in 2 separate labs. Building upon the organizational context and initial models developed in Labs 1 and 2, your group will now apply the project management fundamentals learned in Chapter 3. The primary goal is to solidify the project's justification, assess its feasibility, and create initial plans. This lab bridges the gap between identifying a need and formally proposing a feasible, planned systems project. You will synthesize your understanding of project initiation, feasibility analysis, and scheduling concepts.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: Refine Problem Definition and Justify Project Selection

Begin by revisiting the findings and observations you collected in Labs 1 and 2, particularly regarding current workflows, performance metrics, and stakeholder sentiments. To strengthen your understanding of the current problem space, conduct simple spot checks. For instance, you can measure the turnaround time of a typical report and compare it against established service-level expectations. Observe a few real-world process instances to identify inefficiencies or inconsistencies. Additionally, seek out external feedback, such as a customer comment or peer evaluation, that provides an outside-in perspective.

Synthesize these data points into a concise, two-paragraph Problem Statement that articulates the gap between current and desired performance or the opportunity for system improvement. Following the statement, distill your observations into three clearly aligned lists. First, identify 3 to 5 key issues encountered in the current environment (e.g., "manual data collation introduces errors and delays"). For each issue, define a corresponding project objective: a specific, actionable goal such as "reduce manual entry through automation." Then, articulate 3 to 5 high-level system requirements, focusing on critical features the future solution must deliver (e.g., "system must support secure API integration with existing HR software"). You should also identify relevant constraints, which might include budget limitations, technological expertise available within the organization, or the requirement to deliver a working prototype by the end of the semester.

Conclude this task with a persuasive justification for project approval. Using the five evaluation criteria from the textbook: note signs of management support, argue why this semester offers appropriate timing, demonstrate how the project aligns with strategic goals, assess its practicality given your group's skillset and resource availability, and make the case that it is a worthwhile investment relative to other competing initiatives. This holistic framing is essential for building organizational confidence in the project.

### 1.2 Task 2: Conduct TEO Feasibility Analysis and Estimate Workloads

With a clearer understanding of your system's objectives, proceed to conduct a TEO (Technical, Economic, Operational) feasibility analysis. Start by evaluating technical feasibility. Consider whether the existing system can be enhanced or if it must be replaced altogether. Examine the stability and maturity of any proposed technologies and assess whether your team possesses the necessary technical expertise or if third-party platforms or packaged solutions will be required. Present a brief comparison of potential hosting models (on-premise, IaaS/PaaS, and SaaS) to understand the trade-offs in control, scalability, cost, and complexity.

Next, assess economic feasibility by estimating whether the anticipated benefits will justify the required investments. Reflect on whether long-term value (such as increased productivity, reduced operational bottlenecks, or strategic flexibility) will offset initial expenses. Be specific about short-term savings too: for instance, could the system reduce staffing needs or overtime hours right away? Support your conclusions with logical reasoning or preliminary data extrapolation.

For operational feasibility, analyze how well your proposed solution fits within the workplace environment identified in Lab 2. Consider cultural readiness, resistance to change, and the adaptability of users. Identify any modifications to existing workflows the system may require and outline strategies for managing that change (such as training, pilot testing, or phased rollouts).

Close this task by estimating the expected workload improvements. Choose two representative tasks, such as "monthly financial report compilation" and "inventory restocking updates." For each, quantify the current resource consumption (e.g., staff-hours, system load) and project the post-implementation values based on your proposed solution. Present concrete estimates. For example, "inventory updates will drop from 12 staff-hours to 4 staff-hours per week." These figures provide an early signal of potential impact and efficiency gains.

### 1.3 Task 3: Forecast Costs and Benefits with Break-Even and Payback Analyses

The final task involves evaluating the project's financial viability through a structured cost-benefit forecast. Begin by identifying key categories: list the tangible benefits, such as reduced labor hours or licensing fees, and intangible benefits, such as better decision-making or enhanced data accuracy. Similarly, outline all tangible costs, including one-time implementation expenses (e.g., a $5,000 software license or $6,000 worth of development labor), and intangible costs or risks, such as staff training time or potential operational disruption during deployment.

Check whether historical data exists to support these forecasts. For instance, if you have six months of records on overtime hours, use that data to extrapolate future savings. If not, rely on industry benchmarks or vendor whitepapers. Each major figure should be backed by at least one cited source. For example, "according to XYZ Software's case study, onboarding time for new users decreased by 40%."

Finally, perform a Break-Even Analysis to determine how many months it will take before cumulative benefits match cumulative costs. Alongside this, conduct a Payback Period calculation to estimate how quickly the project will return its investment. Present both analyses with simple, transparent formulas and numerical results (e.g., "break-even in month 11, payback in month 13"). These calculations provide essential data points for justifying project approval.

## 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ Brief Project Overview and Problem Definition Document.
- ▷ Summary of Feasibility Assessment (TEO highlights)
- ▷ Summary of your forecasts showing your data source, cost vs. benefits analysis, and the outcome of the break-even and payback analyses.

## 3 Report Requirements

For this lab, your report must include the following information:

- Cover Page
  - Group member names and student IDs
  - Course code and title (CSE 4408: System Analysis and Design Lab)
  - Lab number (Lab 3)
- Problem Definition and Project Selection Justification
  - Problem Statement: One/two concise paragraphs synthesizing your spot-check measurements, direct process observations, and any external feedback (customer comment or peer evaluation).
  - Key Issues: Primary pain points
  - Project Objectives: An actionable goal for each issue
  - Requirements: Preliminary "must-have" features that address those objectives
  - Constraints: Foreseeable limits.
  - Project Approval Justification: A focused narrative addressing the five textbook criteria.
- Preliminary Feasibility Assessment (TEO) and Workload Estimates
  - Technical Feasibility: Enhancement vs. replacement evaluation, Comparison of hosting models (On-Premise, IaaS/PaaS, SaaS) with pros/cons, and Team expertise gaps or third-party dependencies
  - Economic Feasibility: Logical assessment of long-term value versus upfront costs and Reference to any preliminary data or industry benchmarks
  - Operational Feasibility: Fit with cultural readiness and workflow changes (from Lab 2) and Change-management strategies
  - Workload Improvement Estimates: Current resource consumption, projected post-implementation values, current vs. projected metrics
- Cost-Benefit Forecast, Break-Even & Payback Analyses
  - Tangible Benefits, Intangible Benefits, Tangible Costs, and Intangible Costs
  - Supporting Data and/or Citations
  - Break-Even Analysis
  - Payback Period Calculation
- Formatting
  - Use consistent, professional formatting (A4 paper, 12pt font, 1.5 line spacing, 1-inch margins)
  - Number all figures and tables sequentially; include captions

Ensure that each section is clearly labeled and accompanied by explanatory text so a reader unfamiliar with your group's discussions can understand your analyses and planning artifacts.

## 4 Presentation Rubrics

Your presentation will be evaluated based on:

- Problem Definition and Justification: Clarity of issues, objectives, requirements, constraints; strength of project selection justification.
- Feasibility Assessment: Thoroughness and realism of TEO analysis; logical connections made.
- Cost/Benefit and Diagrams: Identification of relevant tangible/intangible costs and benefits; thoroughness of Break-Even Analysis and Payback Analysis.

▷ Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 5 Report Rubrics

Your report will be evaluated based on:

▷ Problem Definition and Project Selection: Clarity and alignment of the problem statement, issues/objectives/requirements lists, constraints, and justification against the five textbook criteria.

▷ Preliminary Feasibility Assessment and Workload Estimates: Depth of TEO analysis (Technical, Economic, Operational) plus concrete before-and-after workload projections for representative tasks.

▷ Cost–Benefit Forecast, Break-Even and Payback Analyses: Completeness of benefits/costs lists with supporting data or citations, and correctness of break-even and payback calculations.

▷ Presentation and Formatting: Professional consistency, cover page accuracy, figure/table numbering, and overall readability.

## 6 Deliverables

Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L3_Slides.pdf` and `[WWW_XXX_YYY_ZZZ]_L3_Report.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

# Lab 4 Project Management

This lab is a continuation of Lab 3. This lab completes the culmination of the initial project planning phases. Building upon the project's justification, feasibility, and initial plans in Lab 3, your group will now structure a compelling project proposal. You will synthesize your understanding of cost-benefit considerations, and team/risk factors into a coherent proposal structure.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: Develop Initial Project Plan and Schedule

Translate your chosen methodology (from Lab 1) into a Work Breakdown Structure (WBS). At a minimum, include phases such as Planning, Analysis, Design, Development, Testing, Deployment, and Maintenance, and break each into two levels of activities (e.g., "Design→ UI Mockups, Database Schema").
Assign time estimates (in weeks or months) to each top-level phase and key activity, justifying these based on team size, assumed complexity, or analogies to similar projects.
Finally, produce a Gantt Chart, using any online/offline tool, to visualize your schedule. Show duration and sequence of phases. Opt for PERT chart if dependency is needed to be shown.

### 1.2 Task 2: Consider Team and Risks

Identify the key skills and roles your project will need (for example, Business Analyst, Programmer, Database Specialist, UI/UX Designer, Domain Expert). This ensures you can staff each phase appropriately.
Next, brainstorm at least three significant project risks, stating each risk succinctly and then its underlying cause using a fishbone-style phrasing (e.g., "Risk: Low user adoption. Cause: Insufficient user involvement during requirements gathering").

### 1.3 Task 3: Draft Project Charter and Proposal Outline

Conclude Lab 4 by drafting a Project Charter that answers the core questions from the textbook: What are your Objectives? What's the Scope? Which Methods will you use? Who are the Participants? What Deliverables and Evaluation criteria apply? Outline your Timeline, Training plan, and Maintenance approach succinctly but clearly.
Finally, prepare an outline for your full Systems Proposal, listing the ten main sections (such as Executive Summary, Background, Requirements, Alternatives Analysis, System Design, Implementation Plan, etc.) as described in the textbook. You are expected to create a preliminary proposal at this point, which is expected to change throughout the semester.

## 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ Brief summary of the analysis conducted in Lab 3
- ▷ High-Level Plan (Show WBS structure and Gantt Chart/PERT Chart)

- Team formation and key risks Identified
- Brief summary of the Project Charter and Proposal
- Concluding statement on project viability/recommendation (implicit)

## 3 Report Requirements

For this lab, your report must include the following information:

- Cover Page
  - Group member names and student IDs
  - Course code and title (CSE 4408: System Analysis and Design Lab)
  - Lab number (Lab 4)
- Initial Project Plan and Schedule
  - Work Breakdown Structure (WBS) mentioning the top-level phases and two levels of activities under each phase
  - Time Estimates listing the estimated duration for each phase/activity with brief justification
  - Gantt Chart/PERT Chart showing phase durations, sequence, and dependencies
- Team Composition and Risk Analysis
  - Project Team Roles and Skills listing the key roles and their associated skills
  - Risk Register containing a concise description of the risk and its underlying factor
- Project Charter and Proposal Outline
  - Project Charter containing objectives, scope, methodology, participants, deliverables, evaluation criteria, timeline, training plan, and maintenance approach
  - Systems Proposal Outline containing your preliminary proposal
- Formatting
  - Use consistent, professional formatting (A4 paper, 12pt font, 1.5 line spacing, 1-inch margins)
  - Number all figures and tables sequentially; include captions

Ensure that each section is clearly labeled and accompanied by explanatory text so a reader unfamiliar with your group's discussions can understand your analyses and planning artifacts.

## 4 Presentation Rubrics

Your presentation will be evaluated based on:

- Initial Plan and Schedule: Quality and logic of WBS; reasonableness of time estimates; clarity and correctness of Gantt/PERT chart.
- Team and Risk Factors: Thoughtfulness in team composition and risk analysis
- Project Charter and Outline: Completeness of charter elements and proposal outline.
- Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 5 Report Rubrics

Your report will be evaluated based on:

- Initial Project Plan and Schedule: Quality of WBS, time estimates with justification, and visual schedule (Gantt or PERT) with clear dependencies.
- Team Composition and Risk Analysis: Appropriateness of team-role mappings and depth of risk identification with causes.
- Project Charter and Proposal Outline: Completeness and coherence of the charter elements and preliminary proposal structure.
- Presentation and Formatting: Professional consistency, cover page accuracy, figure/table numbering, and overall readability.

## 6 Deliverables

Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L4_Slides.pdf` and `[WWW_XXX_YYY_ZZZ]_L4_Report.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

## 7 Bonus Task: Complex Engineering Problem

*(tl;dr at the end)*
As a part of this lab, you will learn how to describe your information-system project as a Complex Engineering Problem (CEP) by aligning it with BAETE's outcome based framework:

- Engineering Knowledge Levels (WK): the depth and breadth of domain knowledge you must apply
- Complexity Attributes (WP): the dimensions that make a problem "complex"
- Complex Engineering Activities (EA): the real-world tasks and processes that embody those complexity attributes

### 7.1 Overview

A Complex Engineering Problem (CEP) is, by definition, a challenge that:

1. Requires in-depth engineering knowledge (WK)
2. Exhibits one or more Complexity Attributes (WP)

Formally, BAETE states:

> A problem becomes "complex" if it hinges on WP1 (depth of knowledge) and at least one of WP2 through WP7.

In other words, you cannot solve a CEP simply by following a cookbook or reusing existing modules. You must dig into engineering fundamentals (WK3), possibly consult specialist knowledge (WK4), consider resource and environmental design (WK5), apply engineering practice (WK6), and often perform research-based analysis (WK8). At the same time, you navigate one or more of these complexity attributes:

- WP1: Depth of Knowledge Required
- WP2: Range of Conflicting Requirements

- WP3: Depth of Analysis Required (no obvious solution)
- WP4: Novel or Infrequently Encountered Issues
- WP5: Lack of Established Codes or Standards
- WP6: Multi-Stakeholder Collaboration
- WP7: Interdependent Subsystems

In BAETE's framework, WK1-WK8 categorize the kind of knowledge you use when solving engineering problems. The most relevant for CEPs are:

- WK3: Engineering Fundamentals – Core principles such as data structures, algorithms, discrete math, and system architecture.
- WK4: Specialist Knowledge – In-depth expertise, such as healthcare-IT protocols (HL7) or secure payment standards (PCI-DSS).
- WK5: Resource and Environmental Design – Understanding hardware constraints, energy consumption, environmental impact, or regulatory compliance (e.g., GDPR, HIPAA).
- WK6: Engineering Practice – Real-world skills in team coordination, project management, DevOps pipelines, and version control.
- WK8: Research-Based Analysis – Applying scholarly research and new methodologies when no standard solution exists, such as designing a novel machine-learning model or customizing a message-broker for medical devices.

When tackling a CEP, you will typically draw on multiple WK levels. For example, building a secure e-commerce platform might require WK3 (cryptographic algorithms), WK6 (setting up CI/CD), and WK8 (researching anti-fraud heuristics).
WP1–WP7 describe what makes a problem "complex." Here is a brief summary:

- WP1 (Depth of Knowledge Required): You must apply at least one of WK3, WK4, WK5, WK6, or WK8, meaning you cannot solve the problem by simple recall of a known template or pattern.
- WP2 (Range of Conflicting Requirements): You have to juggle competing goals or constraints. Examples: performance vs. security; cost vs. usability; scalability vs. quick development.
- WP3 (Depth of Analysis Required): There is no obvious, off-the-shelf solution. You must build new abstractions or mathematical models, or perform deep "what-if" reasoning.
- WP4 (Novel or Infrequently Encountered Issues): The domain or technology is new for most practitioners. You may be pioneering a feature that rarely appears in textbooks.
- WP5 (Lack of Established Codes/Standards): There is no clear "standard" to follow, so you must decide on your own safety, security, or architectural guidelines.
- WP6 (Multi-Stakeholder Collaboration): You must coordinate with multiple groups, each with different interests (e.g., IT Security team, Legal department, End Users).
- WP7 (Interdependent Subsystems): Your system has many parts that depend on each other (e.g., front-end UI, back-end API, payment gateway, database), and you must manage how they interact.

A problem qualifies as "complex" only if WP1 is present (i.e., you need deep engineering knowledge) plus at least one of WP2–WP7.
While WP1–WP7 define why a problem is complex, EA1–EA5 define what you do when you solve a CEP. In other words, EAs are the real-world tasks and processes you undertake, each mapping to specific WPs and WKs. The 5 EAs are:

1. EA1 (Range of Resources)

2. EA2 (Level of Interactions)
3. EA3 (Innovation)
4. EA4 (Consequences to Society/Environment)
5. EA5 (Familiarity)

A CEP arises when you embark on one or more of these EAs, each of which inherently relies on specific WPs and WKs. Your job in this lab is to demonstrate, for your own project, how each EA is already in play, drawing explicit connections to WPs and the WK levels you must use.

## 7.2 What you have to do

We will align your project to five Complex Engineering Activities (EA1-EA5). Completing this task will strengthen your project proposal's CEP section (worth bonus marks). You are not required to produce a perfect answer; instead, focus on understanding each EA, unpacking the terminology, and mapping it to your own project deliverables so far.
We will use a fictional project "Digitalizing the University Medical Center (UMC)" to show how each EA is interpreted and mapped. Read through the example for guidance, then substitute details from your own project.

## 7.3 EA1: Range of Resources

EA1 means your project touches many different kinds of resources. These resources include:

- People with different expertise
- Data sets
- Physical/hardware resources
- Financial resources
- Specialized tools or technologies

In other words, EA1 emphasizes that your project cannot be built using just one type of input. Rather, it requires coordination across people, data, hardware, money, and specialized software.

### Maps to WPs

- WP1 (Depth of Knowledge Required): You need at least WK3–WK6 to understand and integrate all these resources.
- WP6 (Multi-Stakeholder Collaboration): You are working with different groups (end users, administrators, IT staff).
- WP7 (Interdependent Subsystems): Each resource (data, hardware, etc.) depends on the others.

### Required WKs

- WK3 (Engineering Fundamentals): Model data flows and system architecture to see how resources interact.
- WK5 (Resource and Environmental Design): Consider hardware constraints, energy usage, and budget.
- WK6 (Engineering Practice): Manage teams, scheduling, and procurement.

## UMC Example

UMC Example for EA1:

- People: Doctors, Nurses, Receptionists, Lab Technicians, Hospital Administrators all have different needs and knowledge domain.
- Data: The system must manage patient demographics, medical history, lab test data, billing information.
- Hardware: Servers to host the database, tablets for doctors to enter notes, barcode readers for medication tracking.
- Budget: The hospital has a limited IT budget; provisioning new servers or buying EHR licenses involves finance approval.
- Tools/Technologies: A secure EHR platform (e.g., OpenMRS), integration middleware to talk to lab instruments, HIPAA-compliant messaging services.

## Guide for your project

1. List all resource types that your project depends on. In your previous labs, you identified stakeholders and created a high-level DFD. Use that to extract:
   - People/resources: Which roles or departments will use or maintain the system? (From stakeholder analysis.)
   - Data types: What data flows did you model in Lab 2's DFD? List all data entities in your preliminary ERD.
   - Hardware/physical systems: What servers, devices, or networking equipment are required?
   - Budget constraints: Do you need to purchase new licenses, rent cloud hosting, or upgrade existing infrastructure?
   - Software/tools: What frameworks, libraries, or third-party modules will you rely on?
2. Explain, in a few sentences (3-5 lines), how coordinating these diverse resources makes your project more than a "simple" assignment. For instance, if you're building an e-commerce platform, note that you'll need to coordinate student accounts (people), inventory databases (data), payment gateways (software/tools), web servers (hardware), and possibly new funding for secure payment integration (financial).
3. Write your EA1 subsection (approx. 4-6 sentences) for your proposal, following the style below (replace bracketed text with your project's details):

   EA1: Range of Resources
   "Our [PROJECT NAME] system involves diverse resources: [LIST PEOPLE/ROLES] (people), [LIST DATA] (data), [HARDWARE/DEVICES] (hardware), [BUDGET DETAILS] (financial), and [SOFTWARE/-TOOLS] (specialized tools). Coordinating these elements, such as negotiating server budgets with IT (WK6), ensuring data schemas support both real-time queries and archival (WK3, WK5), and managing deployment pipelines. It also demonstrates WP6 (multi-stakeholder collaboration) and WP7 (interdependent subsystems). This goes beyond a simple coding assignment and requires us to integrate first-principles knowledge (WK3) with practical resource planning (WK6)."

## 7.4 EA2: Level of Interactions

EA2 means your project must manage conflicts or trade-offs between technical requirements and non-technical factors. In practice, this includes negotiating goals such as:

- Performance vs. Security
- Cost vs. Quality
- Usability vs. Compliance
- Short-term needs vs. Long-term maintainability

Put differently, EA2 captures that your project forces you to balance competing priorities and make informed decisions.

### Maps to WPs

- WP2 (Range of Conflicting Requirements): Directly corresponds to EA2; your system has competing goals.
- WP3 (Depth of Analysis Required): You must analyze options to find acceptable trade-offs.
- WP6 (Stakeholder Collaboration): You need input from different teams to rank priorities.

### Required WKs

- WK3 (Engineering Fundamentals): To measure performance trade-offs (e.g., algorithmic complexity vs. usability).
- WK6 (Engineering Practice): To run stakeholder workshops and negotiation sessions.
- WK8 (Research-Based Analysis): To evaluate novel solutions or compare frameworks (e.g., different encryption libraries).

### UMC Example

- Performance vs. Security: Doctors want quick access to patient charts, but strict encryption and two-factor authentication slow down logins.
- Cost vs. Quality: The hospital must decide between an expensive commercial EHR package (with full support) or an open-source solution that requires in-house customization.
- Usability vs. Compliance: Nurses need a simple touchscreen interface to record vital signs, but compliance officers demand audit trails for every data edit (more clicks and log entries).

### Guide for your project

1. Identify at least two sets of conflicting requirements in your project. Look back at your Lab 3 feasibility study: did you note any trade-offs? Examples:
   - For an e-commerce platform: balancing payment speed (students want instant checkout) versus fraud detection measures (which add delays).
   - For digitalizing the cafeteria: choosing between a simple web app (easier to develop) vs. a mobile app (higher user adoption), but mobile requires extra expertise and cost.
   - For carbon footprint tracker: high-precision emission calculations (complex algorithms, slower processing) vs. quick daily summaries (less accurate, more responsive).

2. Describe (in 3–5 sentences) how you will gather stakeholder input to clarify these conflicts and how you might negotiate them. For example, will you hold a workshop to rank priorities? Will you prototype two UI options and measure user satisfaction?
3. Write your EA2 subsection (approx. 4–6 sentences) in the proposal. For instance:

   EA2: Level of Interactions
   "Our [PROJECT NAME] must reconcile [CONFLICT A] (e.g., performance vs. security) and [CONFLICT B] (e.g., cost vs. usability). By engaging [STAKEHOLDER GROUPS] through [METHODS, e.g., workshops, surveys] (WK6), we will clarify which trade-offs are acceptable. We will prototype [COMPONENT OR WORKFLOW] (WK8) to evaluate alternative approaches and measure performance metrics (WP3). These steps address WP2 (range of conflicting requirements) and WP6 (multi-stakeholder collaboration), ensuring we do not adopt a one-size-fits-all solution."

## 7.5 EA3: Innovation

EA3 means your project calls for genuine creativity or research-based solutions, something beyond assembling off-the-shelf components. You may need to design new algorithms, integrate emerging technologies, or apply domain knowledge in novel ways. In a CSE context, this could be:

- Implementing a custom recommendation engine for student merchandise (not just a textbook "if-then" rule)
- Designing a dynamic load-balancing mechanism for cafeteria checkout during peak hours.
- Building a real-time carbon-footprint estimator that uses sensor data and machine-learning models.

Innovation is thus about doing something that is not routine; it requires you to think from first principles, research best practices, and propose an original solution.

### Maps to WPs

- WP3 (Depth of Analysis Required): Because no off-the-shelf fix exists, you must analyze and create new approaches.
- WP4 (Novel or Infrequently Encountered Issues): You are doing something that most practitioners haven't tried before.
- WP5 (Lack of Established Codes/Standards): You may have to set your own guidelines or benchmarks.

### Required WKs

- WK4 (Specialist Knowledge): For the domain where novelty is required. E.g., healthcare interoperability, advanced payment fraud detection, IoT systems.
- WK8 (Research-Based Analysis): To read academic papers, compare existing algorithms, and prototype new models.

### UMC Example

The UMC team realizes no existing EHR package supports native lab-equipment integration. They research HL7 standards, design a lightweight message-broker service to translate lab-instrument outputs into their EHR's format, and prototype a rule-based alert system to flag abnormal lab results in real time.

## Guide for your project

1. Pinpoint at least one area where your project cannot just "plug in" a standard library or template: you must innovate. Review your preliminary ERD or use-case diagram:
   - ▷ Does your e-commerce platform need a novel loyalty-points algorithm that current plugins don't support?
   - ▷ For the cafeteria system, is there a unique "predictive restocking" feature you plan to build from scratch?
   - ▷ If your carbon footprint tracker relies on IoT sensors, do you need to design a custom data-aggregation pipeline?
2. Explain (in 3–5 sentences) how you will perform research or apply domain knowledge to that component. Which resources (technical papers, online forums, domain experts) will you consult? How will you prototype or test viability?
3. Write your EA3 subsection (approx. 4–6 sentences). Example:

   EA3: Innovation
   "Standard open-source e-commerce frameworks lack a campus-wide loyalty algorithm tied to meal plans. We will research gamification literature and design a dynamic point-allocation service that adjusts in real time based on purchase patterns (WK8, WP4). We plan to prototype a microservice (written in Node.js) that ingests transaction streams and applies time-decay weighting (WK4). By benchmarking against existing loyalty engines (WP3, WP5), we will refine our algorithm to ensure it is both novel and scalable."

## 7.6 Consequences to Society/Environment

EA4 reminds us that engineering solutions sometimes have large or unpredictable impacts on people or the environment. In CSE projects, these impacts often translate into:

- ▷ Ethical concerns
- ▷ Societal implications
- ▷ Environmental cost

EA4 asks you to think: "If our system succeeds or fails, who is affected and how?" This includes both positive outcomes (improved care, efficiency) and potential harms (data breaches, exclusion).

## Maps to WPs

- ▷ WP2 (Conflicting Requirements): Ethical and societal goals can conflict with technical goals (e.g., privacy vs. data analytics).
- ▷ WP5 (Lack of Established Codes/Standards): Sometimes no clear ethical/legal guidelines exist, forcing you to set your own best practices.
- ▷ WP7 (Interdependent Subsystems): A design choice that improves one subsystem (e.g., caching for performance) may negatively impact another (e.g., real-time data accuracy, which affects patient care).

## Required WKs

- ▷ WK5 (Resource and Environmental Design): To measure environmental footprint (e.g., server energy usage).

- ▷ WK3 (Engineering Fundamentals): For implementing secure data storage and encryption.
- ▷ WK6 (Engineering Practice): To develop policies, run risk assessments, and engage with ethical review boards.

### UMC Example

- ▷ Positive: Digital records reduce medication errors, speeding up diagnosis and treatment, saving lives.
- ▷ Negative: If servers go down, emergency-room doctors lose access to patient allergies. Also, storing all data in a large database increases energy usage, raising the hospital's carbon footprint.
- ▷ Ethical: Patient records include sensitive information (taboo-disease status, mental-health notes). Improper access could violate privacy and trust.
- ▷ Mitigation Strategies: Implement data-anonymization and role-based access control, schedule computationally heavy processes (e.g., image compression, backups) during off-peak energy times, and conduct regular security audits and train staff on proper data handling.

### Guide for your project

1. List at least two possible positive and two possible negative impacts of your system. Consult Lab 3's feasibility and Lab 2's stakeholder notes.
   - ▷ For an e-commerce platform:
     - ◇ Positive: Simplifies ordering, reduces food waste by matching demand.
     - ◇ Negative: Could exclude students without credit cards or Internet access; transaction data might be mined for marketing (privacy).
   - ▷ For a carbon footprint tracker:
     - ◇ Positive: Raises awareness, motivates greener habits.
     - ◇ Negative: Data collection from personal devices could infringe on privacy; cloud processing might increase energy usage.
2. Describe (in 3–5 sentences) how you will mitigate negative impacts. Will you build data-anonymization features? Provide an offline ordering alternative? Use green-hosting services?
3. Write your EA4 subsection (approx. 4–6 sentences). Example:

   EA4: Consequences to Society/Environment
   "Our [PROJECT NAME] will reduce food waste by promoting seasonal menu items and enabling pre-ordering (positive environmental/societal impact). However, exposing student purchasing patterns could raise privacy concerns; therefore, we will implement AES-256 encryption and role-based access controls (WK3, WP5). Hosting our analytics on a green-energy-certified cloud provider will minimize carbon emissions (WK5). Finally, to ensure equitable access, we will provide an SMS-based ordering alternative for students without smartphones, addressing social-equity concerns (WK6, WP2)."

## 7.7 EA5: Familiarity

EA5 captures how "new" or "unfamiliar" the technical or contextual challenges are. If you are venturing into territory that you (the team) have little prior experience with (whether a new domain, a novel technology stack, or an uncharted integration) you must rely on first-principles thinking. In essence:

- Familiarity refers to how much of the solution you can directly borrow (if it's highly familiar, you'd reuse past code or patterns; if it's unfamiliar, you must learn, research, and invent from scratch)
- An EA5 situation arises when the domain is new (e.g., they've never built a healthcare system before), or the technology is new (e.g., a blockchain-based ledger for medical records).

Put differently, EA5 means you cannot just copy-paste: this is new ground.

## Maps to WPs

- WP1 (Depth of Knowledge): Tackling unfamiliar territory requires WK3 and maybe WK4/WK8.
- WP4 (Novel Issues): If the technology or domain is uncharted for your team, you truly are in "new ground."

## Required WKs

- WK3 (Engineering Fundamentals): To learn the basics of any new tool or domain.
- WK4 (Specialist Knowledge): To acquire domain-specific expertise (e.g., healthcare regulations, GIS mapping for disaster management).
- WK8 (Research-Based Analysis): To read documentation, tutorials, research papers, and consult experts.

## UMC Example

The UMC team has built basic CRUD apps before, but they have never worked with HL7 messaging or HIPAA-grade security. As a result, they must research healthcare interoperability standards and design their own message translators.

## Guide for your project

1. Identify which parts of your project involve new territory for your team. Ask:
   - Have you built an e-commerce site before? If yes, what is new this semester: campus-specific payment tokens, integrating with the university's ID system, etc.?
   - For the carbon footprint tracker: If you've never processed IoT sensor data or sleep-tracking inputs, that is unfamiliar.
   - For disaster management: If you haven't built a GIS component (maps, real-time location), you must learn those tools.
2. Explain (in 3–5 sentences) how you will acquire the necessary knowledge, e.g., online tutorials, consultation with a subject-matter expert, reading API docs, prototyping, etc. Emphasize that you can't simply reuse last semester's code.
3. Write your EA5 subsection (approx. 4–6 sentences). Example:

   EA5: Familiarity
   "Although our team has previously built simple web applications (creating tables, CRUD operations), none of us has experience with message-broker architectures or GIS mapping. To close this gap, we will follow official documentation on RabbitMQ (WK8) and build a small 'spike' project to stream location data (WK3). Additionally, because disaster-management workflows are new to us, we plan to consult domain experts and review open-source GIS libraries (WK4). This unfamiliar territory ensures we cannot rely on existing templates alone,

WP4 demands research and first-principles thinking."

## 7.8 Putting it all together

1. Create a section (after "Detailed Results" section) in your project-proposal outline entitled:

```
7. Complex Engineering Problem
7.1 EA1: Range of Resources
7.2 EA2: Level of Interactions
7.3 EA3: Innovation
7.4 EA4: Consequences to Society/Environment
7.5 EA5: Familiarity
```

2. Under each subsection, write 4-6 sentences as shown in the examples above, replacing all bracketed or example language with your own project details. Each subsection must:
   - ▷ Start by restating the EA's definition in your own words (1–2 sentences).
   - ▷ Cite evidence from your prior labs (DFD, ERD, use cases, feasibility) to show how your project satisfies that EA (2–3 sentences).
   - ▷ Name the relevant WPs and WKs you are invoking.
   - ▷ Optionally, mention any next steps which may be a part of EA.

## 7.9 tl;dr

- ▷ A CEP = WP1 + (at least one of WP2–WP7), supported by applying WK3–WK8.
- ▷ EA1–EA5 are the concrete activities you perform when solving a CEP, each tied to specific WPs and WKs.
- ▷ By mapping your project to EAs and explicitly naming the relevant WPs/WKs, you show that your information-system project meets BAETE's standards for complexity and rigorous engineering practice.
- ▷ Use the Ideal Project Example (UMC) as a template, then replace its details with your own project's context.

# Lab 5 Information Gathering: Interactive Methods

Following the development and presentation of your initial project proposal in Labs 3 and 4, this lab focuses on planning the crucial next step: gathering detailed information requirements from stakeholders. While your proposal outlined the problem, feasibility, and high-level plan, effective system design requires a deep understanding of user needs, attitudes, current processes, and desired features. This lab requires you to apply the interactive information-gathering techniques discussed in Chapter 4 (Interviewing and Questionnaires) to your specific project. This planning bridges the gap between the high-level proposal and the detailed analysis to come.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: Plan Stakeholder Interviews

Begin by refining who you need to talk to in order to gather deep insights about the system. Review your project scope and previous models to finalize a list of key stakeholder groups, for example, end-users (e.g., front-line staff), managers (e.g., department heads), and technical personnel (e.g., IT support). Ensure you capture all roles whose input is critical to understanding requirements.
For 2-3 distinct stakeholder groups, clearly articulate what you intend to learn from each. For instance, for front-line users you might want to uncover hidden workflow pain points; for managers, understand high-level reporting needs; and for IT staff, assess integration constraints and infrastructure concerns. These objectives will guide your questions.
For each targeted group, draft a detailed interview guide including:

- ▷ Introduction script that explains the purpose and builds rapport.
- ▷ A curated mix of open-ended questions (to elicit narratives), closed questions (to confirm specifics), and probing follow-ups.
- ▷ A clear question flow using a recognized structure (e.g., Funnel: start broad, then narrow to specifics).
- ▷ Closing script that thanks the interviewee and outlines next steps.

Finally, conduct interviews, one for each stakeholders (real or simulated), and write the interview reports.

### 1.2 Task 2: Design a User Questionnaire

To reach a broader set of participants efficiently, craft a questionnaire. Decide what quantitative or comparative data you need—such as frequency of certain tasks, satisfaction levels with existing tools, or preferred communication channels. These objectives will ensure each question ties back to actionable insight.
Write approximately 10-15 questions, beginning with a brief introductory note on purpose and confidentiality. Use a mix of:

- ▷ Closed questions (multiple-choice, checkboxes, Likert scales) for easy analysis.
- ▷ 1-2 open-ended questions to capture unexpected feedback. Keep wording neutral, concise, and user-friendly to avoid bias.

Identify your target respondents (e.g., all departmental staff, specific role groups) and choose a delivery

method, such as an institutional survey platform or email with embedded form links. Briefly note timing and reminders strategy to maximize response rates.
Finally, conduct the survey among at least 10 participants (real or simulated).

### 1.3 Task 3: Perform Requirement Elicitation

Using insights from your interviews and questionnaires, you will now perform structured requirement elicitation. The objective is to identify, organize, and document the system requirements based on stakeholder needs and constraints.
Start by analyzing your collected data to extract key themes, pain points, desired features, and implicit expectations. Categorize the requirements into types such as Functional, Non-Functional, Business, User, and Technical Constraints.
Document each requirement clearly using standard syntax (e.g., "The system shall..") in your report, and where applicable, link them to your data sources for justification.
Optionally, validate your requirements with stakeholders (real or simulated) to ensure accuracy, completeness, and alignment with expectations.

### 1.4 Task 4: Consider Other Interactive Methods

Reflect on additional elicitation techniques beyond interviews and surveys.
Discuss whether inviting stakeholders to share real-world "stories" of how they currently work could reveal hidden requirements or pain points. For example, user stories might illuminate rare but critical exceptions that structured questions miss.
Assess if a Joint Application Design (JAD) workshop is feasible: weigh factors such as stakeholder scheduling constraints, organizational culture's openness to collaborative sessions (from Lab 2), and the complexity of system functionality. Conclude whether a JAD session would likely accelerate consensus or if more targeted methods are preferable.

## 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ A quick overview of the key stakeholders identified and the objectives for interviewing them.
- ▷ Examples of key questions from one of your interview guides (highlighting open/closed/probe use).
- ▷ The primary objective of your questionnaire and 1-2 sample questions.
- ▷ The key outcomes of the interviews and survey.
- ▷ The list of potential features extracted from the interview and survey.
- ▷ A brief statement on the potential role of Storytelling or JAD.

## 3 Report Requirements

For this lab, your report must include the following information:

- ▷ Cover Page
  - ◇ Group member names and student IDs
  - ◇ Course code and title (CSE 4408: System Analysis and Design Lab)

- ◇ Lab number (Lab 5)

▷ Stakeholder Interview Planning and Reports

- ◇ Final list of key stakeholder groups with role descriptions
- ◇ Interview objectives sating what you intend to learn from each group
- ◇ For each group, provide introduction script explaining the purpose and building rapport, question flow outline, open-ended, closed, and probing questions, and closing script thanking the interviewee and outlining next steps
- ◇ One report per interview (real or simulated) containing the brief context, summary of key findings organized by objective, and notable quotes or insights

▷ User Questionnaire Design and Results

- ◇ A short note on purpose, confidentiality, and instructions
- ◇ Set of questions with mix of closed-ended items and 1-2 open-ended questions, clear numbering, neutral wording, and response options defined
- ◇ Target respondent profile, chosen delivery method, and timing and reminder strategy to maximize responses
- ◇ Statement of how many participants were surveyed, with tabular summary of response rates for closed questions, and thematic synopsis of open-ended feedback

▷ Requirements Prioritization Planning: List of high-level system features derived from your interview, questionnaire, and scope/objectives along with justification.

▷ Other Interactive Elicitation Methods

- ◇ Discussion on how soliciting “real-world stories” can reveal hidden requirements or exceptions
- ◇ JAD Workshop Feasibility assessing scheduling constraints, cultural readiness, and complexity factors along with conclusion on whether a JAD session is advisable or if alternative methods are preferable

▷ Formatting

- ◇ Use consistent, professional formatting (A4 paper, 12pt font, 1.5 line spacing, 1-inch margins)
- ◇ Number all figures and tables sequentially; include captions

Ensure each section is clearly labeled, contains the required artifacts (guides, tables, summaries), and is accompanied by concise narratives so a reader unfamiliar with your group’s process can follow your planning and analyses.

## 4 Presentation Rubrics

Your presentation will be evaluated based on:

▷ Interviewing: Demonstrated understanding and appropriate application

▷ Questionnaire: Demonstrated understanding and appropriate application

▷ Requirement Elicitation: Identification of the appropriate features with proper justification

▷ Other Interactive Elicitation Methods: Thorough evaluation of other methods

▷ Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 5 Report Rubrics

You report will be evaluated based on:

- ▷ Stakeholder Interview Planning and Reports: Completeness of stakeholder selection, depth of interview guides, and quality of interview reports.
- ▷ User Questionnaire Design and Results: Quality of questionnaire design and clarity of results presentation.
- ▷ Requirement Elicitation: Thorough extraction of list of features from the interview and questionnaire.
- ▷ Presentation and Formatting: Professional consistency, cover page accuracy, figure/table numbering, and overall readability.

## 6 Deliverables

Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L5_Slides.pdf` and `[WWW_XXX_YYY_ZZZ]_L5_Report.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

# Lab 6 Information Gathering: Unobtrusive Methods

This lab focuses on applying the unobtrusive information gathering methods discussed in Chapter 5 to your ongoing semester project. In Lab 5, you planned your interactive information gathering strategy using interviews and questionnaires. Lab 6 complements that work by exploring how unobtrusive methods, such as, sampling, investigation of existing data, conceptual application of mining techniques, and structured observation, can provide additional valuable insights into system requirements, user behavior, and the organizational context without directly interrupting stakeholders. The goal is to develop a plan for how these methods could be used to enrich your understanding, building upon your initial interactive plans.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: Sampling Strategy Design

First, identify at least two potential populations relevant to your project, such as front-line user roles who perform daily transactions or system-generated error logs that capture operational issues. For example, front-line staff process end-user requests, while error logs record system failures automatically.
Next, select one population and choose an appropriate sampling method. If you target front-line staff to gain rich, experience-driven feedback, Purposive Sampling is ideal: it focuses on information-rich cases rather than random selection. Alternatively, Simple Random Sampling ensures every individual has an equal chance of being selected, promoting statistical representativeness.
Justify your choice by linking method to goals: purposive sampling aligns with qualitative depth and resource constraints, while simple random sampling suits quantitative generalization and minimizes selection bias.
Finally, outline practical steps to draw your sample. For purposive sampling of staff: (1) Define inclusion criteria (e.g., staff with $> 50$ transactions/week), (2) Obtain a roster from HR, (3) Select 10–12 staff covering peak/off-peak shifts, (4) Invite participants via email. These steps ensure clarity and traceability in your sampling process.

### 1.2 Task 2: Investigation Plan for Existing Data

Begin by listing at least three existing data sources. For instance, you might consider:

- ▷ System Error Logs (quantitative data capturing error codes and timestamps)
- ▷ Customer Feedback Forms (qualitative comments and suggestions)
- ▷ Monthly Performance Reports (quantitative KPIs on service levels)

Classify each source as quantitative or qualitative: error logs and performance reports yield numbers and statistics, while feedback forms provide narrative insights.
Select one source (say, Customer Feedback Forms) and describe the expected insights: thematic patterns in user satisfaction, recurring feature requests, and pain points in existing workflows. These insights help prioritize system requirements and enhance user-centered design.

## 1.3 Task 3: Observation Planning – Analyst's Playscript

Identify a key user role (for example, the Transaction Processor) whose actions directly affect system throughput and error resolution.
Draft a 5–7-step playscript outlining sequential tasks for this role. For the given example role:

1. Log In: Observe how the user authenticates and note any delays.
2. Capture Transaction: Record data entry steps and manual workarounds.
3. Monitor Validation: Note system feedback on errors or approvals.
4. Execute Corrections: Track user strategies for resolving validation failures.
5. Log Outcomes: Observe how results are documented and shared (e.g., via email).

This playscript emphasizes critical information-handling activities and highlights opportunities for automation or UI improvements.

## 1.4 Task 4: Observation Planning – STROBE

Prepare to apply the STROBE elements (Office location, Desk placement, Stationery, Props, External sources, Lighting/color, Clothing) to their (hypothetical, if needed) workspace. Each element provides clues about work habits, information access preferences, and environmental constraints, guiding UI design and support documentation strategies. Write down key organizational themes growing out of your findings from Lab 5. Observe and record the elements of STROBE from the workspace.

## 1.5 Task 5: Synthesis with Interactive Methods

Apply STROBE to create the Anecdotal List using the data from Task 4. Then compare the narrative and observations and use one of the five appropriate symbols to characterize the relationship.

# 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ Your chosen sampling strategy and justification
- ▷ Your selected existing data source and expected insights
- ▷ Your Playscript example
- ▷ Highlights from your STROBE observations and interpretations

# 3 Report Requirements

For this lab, your report must include the following information:

- ▷ Cover Page
  - ◇ Group member names and student IDs
  - ◇ Course code and title (CSE 4408: System Analysis and Design Lab)
  - ◇ Lab number (Lab 6)
- ▷ Sampling Strategy Design

- ◇ Population Identified: Brief description of *two* relevant populations (e.g., "Front-line support staff who process user tickets," "System-generated error logs documenting failures")
- ◇ Chosen Population and Sampling Method: Clearly state which population was selected, name and define the sampling method (e.g., Purposive Sampling, Simple Random Sampling), and provide justification linking method to your goal (qualitative depth vs. quantitative generalization; resource constraints; bias reduction)
- ◇ Sampling Procedure: Step-by-step outline on how you would draw your sample. Example: Define inclusion criteria (e.g., staff with $> 50$ transactions/week), request filtered roster from HR, select diverse participants across shifts, and send personalized email invitation.

▷ Investigation Plan for Existing Data

- ◇ Three Existing Data Sources: For each, provide name fo the data source, classification as quantitative or qualitative, and 1-sentence description of what it contains.
- ◇ Selected Data Source (Detailed): Choose one source and describe the expected insights, how it can inform requirements or UI/UX design, and any limitations in its current format.

▷ Observation Planning - Analyst's Playscript

- ◇ User Role Selected: Define one role with a key operational function
- ◇ Playscript: Sequential steps this user performs and use short action-descriptions in each step

▷ Observation Planning - STROBE Analysis

- ◇ Organizational Themes from Lab 5: Summarize 2-3 key cultural/operational themes
- ◇ Workspace STROBE Assessment: For each of the 7 STROBE elements, record observed details (hypothetical or simulated is acceptable) from office location, desk placement, stationary, props, external sources, lighting/color, and clothing.

▷ Synthesis with Interactive Methods

- ◇ Anecdotal List Derived from STROBE: List of narrative observations drawn from STROBE data
- ◇ Comparison with Interactive Methods: Align each anecdote with data from interviews/surveys (Lab 5) and use one of the five symbols to classify the relationship

▷ Formatting

- ◇ Use consistent, professional formatting (A4 paper, 12pt font, 1.5 line spacing, 1-inch margins)
- ◇ Number all figures and tables sequentially; include captions

Ensure each section is clearly labeled, contains the required artifacts (guides, tables, summaries), and is accompanied by concise narratives so a reader unfamiliar with your group's process can follow your planning and analyses.

## 4 Presentation Rubrics

Your presentation will be evaluated based on:

▷ Application of Concepts: Demonstrated understanding and appropriate application of sampling, playscript, and STROBE relevant to the project

▷ Justification and Clarity: Clear reasoning provided for choices made. Playscript and STROBE descriptions are clear and plausible.

▷ Synthesis: Thoughtful considerations of how unobtrusive methods complement interactive methods.

▷ Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 5 Report Rubrics

Your report will be evaluated based on:

▷ Sampling Strategy Design: Quality of population identification, sampling-method justification, and procedural clarity.

▷ Investigation Plan for Existing Data: Comprehensiveness of data-source mapping and depth of insight description.

▷ Observation Planning – Analyst's Playscript: Clarity and relevance of the playscript steps for the chosen user role.

▷ Observation Planning – STROBE Analysis: Depth of workspace assessment using STROBE elements and thematic linkage.

▷ Synthesis with Interactive Methods: Coherence of anecdotal list and its alignment with prior elicitation data.

▷ Presentation and Formatting: Professional consistency, cover page accuracy, figure/table numbering, and overall readability.

## 6 Deliverables

Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L6_Slides.pdf` and `[WWW_XXX_YYY_ZZZ]_L6_Report.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

# Lab 7 Agile Modeling, Prototyping, and Scrum

This lab transitions from planning information gathering to applying Agile principles covered in Chapter 6. You will begin translating the requirements you have started gathering for your chosen information system into an Agile format using User Stories. You will also create a simple prototype or mockup for a core part of your system, simulating the early stages of an Agile development cycle like Scrum. This lab aims to give you practical experience with user-centric requirements definition, iterative design through prototyping, and basic Agile planning concepts, setting the stage for more detailed system design in later phases. This practical application is crucial whether developing systems for local organizations in IUT, larger enterprises in your country, or any modern software context.

## 1 Tasks List

Complete the following tasks as a group:

### 1.1 Task 1: User Story Development and Prioritization

Begin by revisiting the stakeholder needs, process flows, and feature ideas you gathered in Labs 5 and 6. Consolidate insights from interviews, questionnaires, observations, and document analysis to identify the most critical system functions. This review ensures your user stories reflect real user pain points and opportunities uncovered so far.

Select 5-7 essential features that will deliver immediate value and form your Minimum Viable Product (MVP). These might include core transactional workflows, reporting dashboards, or notification modules. Choose features that address the highest-priority business needs (for example, “Process Transaction Requests,” “Generate Summary Reports,” or “Authenticate and Authorize Users”).

For each core feature, write at least one user story in the standard Agile format:

```
As a <role>,
I want <task/goal> so that <benefit/value>.
```

For example,

```
As a Transaction Processor,
I want to submit a new request through a guided form, so that I reduce data entry errors
and save time.


As a Manager,
I want to view a dashboard of daily transaction volumes, so that I can monitor performance
and reallocate resources promptly.
```

Ensure each story captures the user’s perspective and the business value it provides.

Assign a priority level (e.g., Must/Should/Could using MoSCoW or High/Medium/Low) to each user story based on factors such as business impact, technical dependency, and stakeholder urgency. For instance, user authentication is typically a “Must have,” whereas advanced analytics might be “Could have.” Briefly justify each prioritization, for example, “Transaction submission is foundational because no other feature can function until users can log in and create requests.”

## 1.2 Task 2: Prototyping / Mockup Creation

From the prioritized list, choose one or two stories that involve significant user interaction or key UI elements—often those marked as "Must have." For example, "Submit Transaction Request" and "View Transaction Dashboard" are strong candidates, as they represent both data entry and information visualization flows.

Using your preferred tool (e.g., Figma, Balsamiq, draw.io, or PowerPoint), build a wireframe or mockup for each selected story. Focus on illustrating the user flow: how the user arrives at the screen, what fields/buttons are visible, and how feedback is displayed. Annotate each screen with brief notes explaining interactive elements (e.g., "Clicking 'Submit' shows a confirmation banner").

In your deliverable, clearly state whether your prototype is a non-operational UI/UX prototype (focusing solely on layout and navigation) or a selected-features prototype (demonstrating partial functionality). For example, if you used Figma to simulate button interactivity but without backend logic, note that it is a "Non-operational UI/UX prototype." If you chose a No-Code tool like Bubble, explain that you built a clickable demo for form submission, highlighting your choice based on the need to quickly validate user flows.

## 1.3 Task 3: Agile Planning Simulation (Scrum Basics)

Assign three core Scrum roles within your team:

- **Product Owner:** Typically the group member with strong domain knowledge who can represent stakeholder priorities (e.g., the student who led stakeholder interviews).
- **Scrum Master:** Someone with a knack for process facilitation and keeping the team on track (e.g., the student who organized previous labs).
- **Development Team Member(s):** The remaining member(s) who will split tasks such as design, prototyping, and documentation. Briefly justify each assignment based on skills or interests, for example, "Alice has experience with UI design, so she'll be a Development Team member focusing on mockups."

List all the prioritized user stories from Task 1 in a Product Backlog table. For each story, include a brief description, its priority category, and (optionally) a rough size estimate (e.g., S, M, L). This backlog serves as your master list of work items, ordered by priority.

Assume a two-week Sprint for your first iteration. From the top of your Product Backlog, select 2-3 user stories that the team realistically believes it can complete in two weeks—often the highest-priority and smallest stories. Describe why these were chosen (e.g., "User authentication and basic dashboard display are foundational, low to medium complexity, and unblock other features"). If you decide to include story point estimates (e.g., assigning "3 points" to a form-submission story), note that as part of your Sprint Planning rationale.

## 1.4 Task 4: Agile vs. Structured Reflection

Write a short reflection (1–2 paragraphs) comparing your project's evolution under an Agile approach versus a traditional SDLC. Focus on how requirements discovery differs—Agile welcomes changing requirements through iterative feedback, whereas SDLC emphasizes complete requirement specification up front. Discuss how your team's early user stories and rapid prototyping in Agile contrast with a heavyweight requirements document that SDLC would produce.

Reflect on aspects such as user involvement (Agile engages stakeholders continuously, while SDLC might involve them primarily at milestone reviews), flexibility to change (Agile sprints enable quick pivots,

whereas SDLC change requests can be slow), delivery speed (Agile can deliver a working MVP in weeks, while SDLC typically yields a deliverable only at the end of a long development cycle), and project-specific challenges (e.g., if your stakeholders expect frequent new features, Agile's incremental releases will be advantageous; if regulatory compliance demands complete documentation up front, SDLC may be more comfortable). Conclude by noting which approach aligns better with your organization's culture and project constraints.

## 2 Presentation Requirements

The general guidelines for preparing for the presentation can be found in Appendix A. For this lab, your presentation must include the following sections:

- ▷ Brief reminder of your chosen project context
- ▷ Prioritized User Stories you developed
- ▷ Your prototype/mockups
- ▷ Simulated Sprint Planning (roles, backlog selections, etc.)
- ▷ Key points from your Agile vs. Structured reflection

## 3 Report Requirements

For this lab, your report must include the following information:

- ▷ Cover Page
  - ◇ Group member names and student IDs
  - ◇ Course code and title (CSE 4408: System Analysis and Design Lab)
  - ◇ Lab number (Lab 7)
- ▷ User Story Development and Prioritization
  - ◇ Context Review: Briefly summarize how stakeholder interviews, questionnaires, observations, and document analyses (Labs 5 and 6) informed your list of potential features.
  - ◇ MVP Feature Selection: List 5–7 essential features chosen for your Minimum Viable Product. For each feature, include a one-sentence rationale tying it to a high-priority business need.
  - ◇ User Stories: For each core feature, write at least one user story in the Agile format. Ensure each story captures both the user's perspective and the business value.
  - ◇ Prioritization: Assign a priority (Must/Should/Could or High/Medium/Low) to each user story. Provide a 1–2 sentence justification for each prioritization.
- ▷ Prototyping/Mockup Creation
  - ◇ Selected User Stories for Prototyping: Identify 1–2 user stories (preferably "Must have") that involve significant user interaction or key UI elements. Briefly explain why these stories were chosen for mockup.
  - ◇ Prototype Tool and Type: State which tool you used. Clarify whether it is a Non-operational UI/UX prototype (layout/navigation only) or a Selected-features prototype (clickable elements, partial functionality).
  - ◇ Wireframes/Mockups: Include one mockup per selected story, showing: the screen layout, user flow annotations, and any interactive element notes. Caption each mockup.

- ◇ Annotation: Under each mockup, add 2–3 sentences explaining how the design supports the user story's goal and expected behavior.

▷ Agile Planning Simulation

- ◇ Team Role Assignments: Name(s) and justification of Product Owner, Scrum Master, and Development Team Members.
- ◇ Product Backlog Table: A table listing all prioritized user stories from Task 1 with columns for story ID, description, priority category, optional size estimate. Order the table by priority.
- ◇ Sprint 1 Planning: Select 2–3 user stories from the top of the backlog that the team can realistically complete in a two-week sprint. Provide a 2–3 sentence explanation for your selections, referencing factors such as scope, estimated complexity, and business value. If used, include story point estimates next to each chosen user story and explain how those estimates informed your sprint capacity decision.

▷ Agile vs. Structured SDLC Reflection

- ◇ Comparison in your context: Contrast how Agile's iterative, feedback-driven approach (rapid user stories, continuous stakeholder involvement) differs from a traditional SDLC's comprehensive, up-front requirements documentation. Discuss Agile's continuous stakeholder engagement and adaptability to change versus SDLC's milestone-based reviews and formal change-control processes. Reflect on Agile's ability to deliver a working MVP in weeks versus SDLC's tendency to produce a deliverable only after a long, sequential cycle. Note which approach better suits your organization's culture and constraints (e.g., if stakeholders expect ongoing feature tweaks, Agile may be ideal; if compliance demands fully fleshed-out documentation early, SDLC might be more comfortable).

▷ Formatting

- ◇ Use consistent, professional formatting (A4 paper, 12pt font, 1.5 line spacing, 1-inch margins)
- ◇ Number all figures and tables sequentially; include captions

Ensure every section is supported by concise narratives and artifacts (mockups, tables) so that a reader unfamiliar with your team's internal discussions can readily follow your user-centered design reasoning and Agile planning.

## 4 Presentation Rubrics

Your presentation will be evaluated based on:

- ▷ User Stories Quality: Clarity, correctness of format, user-centricity, value proposition, and thoughtful prioritization
- ▷ Prototype/Mockup Quality: Clarity, relevance to user stories, usability of the concept presented, appropriate choice and explanation of prototype type
- ▷ Agile Planning and Reflection: Logical role assignment, clear backlog representation, justified sprint selection, insightful comparison of Agile vs. Structured for the specific project
- ▷ Clarity of explanation, professional quality of slides, effective communication by the group, adherence to time limit, and effective teamwork (evidence of collaboration, participation of all members).

## 5 Report Rubrics

Your report will be evaluated based on:

- ▷ User Story Development and Prioritization: Identify MVP features from prior elicitation, write clear user stories, and justify priorities.
- ▷ Prototyping/Mockup Creation: Produce one or two wireframes/mockups for top-priority user stories, with annotations.
- ▷ Agile Planning Simulation: Assign Scrum roles, build a product backlog, and select Sprint 1 stories with rationale.
- ▷ Agile vs. Structured SDLC Reflection: Compare your project's evolution under Agile vs. a traditional SDLC approach.
- ▷ Presentation and Formatting: Professional consistency, cover page accuracy, figure/table numbering, and overall readability.

## 6 Deliverables

Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: Convert your presentation slides the report into PDF files and submit them in the Google Classroom at the end of your presentation. File naming convention: `[WWW_XXX_YYY_ZZZ]_L7_Slides.pdf` and `[WWW_XXX_YYY_ZZZ]_L7_Report.pdf` (Replace `[WWW_XXX_YYY_ZZZ]` with the last three digits of each of your student IDs.).

# Appendix A Presentation Guidelines

Your presentations should be tightly structured, visually clear, professionally delivered, and fully aligned with the learning objectives and rubrics. Start strong with a clear roadmap, design slides that focus on one idea each, engage your audience through eye contact and interaction, manage time precisely, and be prepared both technically and substantively for questions. Consistent rehearsal, role distribution, and adherence to professional norms will maximize both your grade and the impact of your analysis. The following set of instructions is a compilation of presentation guidelines from various universities and organizations that seemed appropriate.

## 1 Planning and Structure

- **Know Your Audience:** Before drafting your slides, conduct a brief audience analysis: identify their roles, expertise level, and primary concerns so you can tailor terminology and examples appropriately.
- **Frame Your Story:** Use a frame-story approach: introduce a familiar "status quo," present a conflict (e.g., organizational pain point), then show how your system analysis resolves it. This narrative arc helps listeners follow and remember your key insights.
- **Three-Part Flow:** Organize into Introduction (purpose and outline), Body (main analysis artifacts), and Conclusion (key findings and next steps).
- **Single Message Focus:** Identify one central theme per presentation and tie each slide back to that theme.

## 2 Slide Design

- **One Idea per Slide:** Limit each slide to a single, clear takeaway that can be grasped in under 10 seconds.
- **Minimal Text:** Use phrases or keywords instead of full sentences to prevent cognitive overload.
- **Strong Visuals:** Incorporate high-quality images or diagrams to reinforce the message and boost retention by up to 400%.
- **Consistent Formatting:** Choose a professional template, use sans-serif fonts, and maintain high contrast for readability.

## 3 Practice and Preparation

- **Have Someone Review Your Speech:** Arrange one or two peer reviews of your full run-through. Fresh eyes catch jargon, logical gaps, or unclear visuals you may have missed. Incorporate their feedback into a second rehearsal.
- **Get a Good Sleep the Day Before:** Prioritize 7–9 hours of restful sleep to optimize cognition and voice control. Establish a consistent bedtime routine (limit caffeine, avoid screens an hour before bed) to ensure you wake up alert and confident.

## 4 Delivery and Professionalism

- **Distributed Roles:** Assign each member a distinct section to present, ensuring equitable speaking time.
- **Professional Attire:** Dress in professional attire (e.g., business casual, business professional, business formal, Smart casual, etc.).

- **Clear, Conversational Tone:** Speak at a measured pace, vary intonation, and avoid reading directly from slides.
- **Eye Contact:** Maintain regular eye contact to engage peers and instructors.
- **Gestures:** Integrate purposeful hand and body gestures to underscore key points—keep hands visible between waist and chest, avoid fidgeting, and match facial expressions to your message. Well-timed gestures make abstract concepts tangible and maintain audience interest.

## 5 Time and Q&A Management

- **Strict Timekeeping:** Target 5–7 minutes for presentation and reserve 1–2 minutes for questions.
- **Rehearsal:** Practice with Slide Show–Rehearse Timings or a stopwatch to refine pacing.
- **Prepare for Questions:** Anticipate queries about requirements, models, and design trade-offs; prepare concise answers.

## 6 Technical and Logistical Preparation

- **Equipment Check:** Arrive early to verify projector, audio, and cable connections; have backups on USB and cloud.
- **Handouts:** Provide concise printouts or digital links for data dictionaries, DFDs, or prototypes if you believe that might be helpful to understand your presentation.

## 7 Engagement and Accessibility

- **Opening Hook:** Start with a brief story, question, or statistic relevant to your client organization.
- **Interactive Elements:** When suitable, use quick polls or solicit brief audience input.
- **Accessible Design:** Use large fonts, clear language, and provide alternative text for visuals.

By weaving these elements into your preparation workflow, you'll further strengthen the alignment between your analysis content and audience expectations, polish your delivery through feedback and rest, and amplify your message with confident, expressive gestures. Note that while this is a comprehensive guideline, it does not cover all.