

COLLEGE OF ENGINEERING
INFORMATION TECHNOLOGY PROGRAM
SWE 401: SOFTWARE ENGINEERING

PROJECT TOPICS AND GUIDELINE

▮ Learning outcomes covered on this assessment:

No.	Learning Outcome
1	Apply UML to create analysis and design models.
2	State measurable requirements for a software system.
3	Break down a large system into subsystems and do object design.
4	Translate system models into implementation code.
5	Compare a system against specific requirements.
6	Show effective teamwork skills.

▮ Topics

The centerpiece of this course is a semester-long project, in which you will manage, analyze, design, and implement a Software System following the Software Development Life Cycle (SDLC) main activities, techniques and processes. This is a set of guidelines for your final course project in SWE 401 course.

Your project should be an unconventional and innovative Business need Software System or an enhancement of an existing software system “is-as software system”. You should feel free to consult with me about anything related, but I'll also expect you to do the necessary background readings to justify your project choice on your own. *You must write a proposal (project description) and post it to discourse by the due date of Phase #1.*

▮ Project Guidelines

1. The project should address a real-life situation which can be implemented (preferably to be smart and using artificial intelligence). For example, the following are good sources but I you must avoid them and have something different and new:
 - a) Airline system (or train, cruise, or other) ticket purchase
 - b) Shopping sites on the internet (E-commerce)
 - c) Online banks and financial institutions
 - d) Online insurance system (life, medical, homeowner, and others)

- e) Blockchain Voting Application
- f) Rental's system (car, video, audio, and others)
- g) Student/ University Portal System
- h) Robotic Delivery System
- i) Clinic Online System (appointments, consulting)
- j) Personal Health Monitoring
- k) Pharmacy System
- l) Home Automation System
- m) Library System
- n) Restaurant System
- o) Parking Garage Automation
- p) Hotel Management System

q) Museum System

You can also use traditional firms if you can get the requisite permission from the appropriate people. Also, they should be able to provide you with the data to complete your project.

2. For the firm you have chosen, identify the major functions that a firm can perform using the appropriate requirements elicitation techniques you will study in the course.
3. Document each function. Figure out the process used in each function, the inputs used, and the outputs generated. However, web searching alone will be inadequate. You have to use your knowledge about the industry/firm/function to identify other data used by the system.
4. Identify at least **TWO** problems or limitations related to the functionality. The limitations could be new functions that could be useful to a potential customer or improvements over existing functions.
5. Analyze the UML models and propose a new system.
6. You should use and apply the concepts using the CASE tools.

Note: The project should be sufficiently multifaceted. Please consult with me in the beginning stage to determine whether your project is adequate for the course.

Project Deliverables

Phase 1:

Each group of students must think, search and decide about a business need and requirements that insists for a new software system, or enhance an existing system (at least two new or two enhanced functions). Identify and come up with a written *Problem Statement* you encounter during your search and the functional model of the proposed system. Note down any improvements that can be made. The problems and improvements should relate to the functionality provided by the system. Your documentation at this phase must include:

1. **Cover Page:**

This will include Department Name, University Name, Course No, Course Name, Title of the project, Student IDs & Names of the team members, Instructor Name, and Date of submission.

2. **Project Description (The Software System Proposal) (1-2 pages or 500 - 700 words):**

This will include:

- Introduction and definition of the problem – what the project is about and what problem it will solve
- General overview and main functionality of the system
- Stakeholders of the system and the intended users – how the user will benefit from using your system
- Description of the current system (if any) – “as-is system”
- Key problems and challenges that you may face during the project’s development
- Technologies that you will use (programming languages, frameworks, platforms, libraries, databases, etc)
- Team members brief CV (brief back ground information, technical skills, and the involvement in the project)

3. The Market Potential (between 200 and 400 words).

- User interest – why the user will be interested using your software system
- Any social and environmental impact (if any)
- Similar or related software systems (provide references)
- Novelty of your proposed idea – what makes your software system innovative and different than others existing (you can use references at this stage)

4. The Project Management Plan (between 2-3 pages).

- Define Activities and Sequence Activities (*Workplan*)
- Software Project Methodology Selection
- Develop the schedule (*Gantt Chart*)
- Estimate Costs (*if any*)
- Determine Budget (*if any*)

Phase 2:

1. Requirements Engineering, start with the requirements elicitation technique(s) which you need to use in order to collect the project requirements, and analyses the outcome. **(1-2 pages)**

2. System users (1/2 page)

This will include *all* types of users that may make use of the software system.

3. Functional Requirements: (1-2 pages)

This will include *all* functional requirements of the system in moderate detail.

4. Non-Functional Requirements: (1-2 pages)

This will include *all* non-functional requirements and quality constraints of the system in moderate detail.

5. Use Case Modeling:

- UML use case diagram for major use cases.
- Detailed text description uses cases

6. At least 2 Natural Language or Structured Natural Language Specifications

7. Traceability Matrix (without weight)

Phase 3:

In this phase, you will make the analysis object model and dynamic model of the system you proposed in Phase1. Make the following diagrams.

1) Activity Diagram:

Based on the UML use cases you modeled in Phase 2, draw an activity diagram for your system. One UML Activity Diagram is sufficient.

2) Class Diagram:

Based on the UML use cases and problem statement you created in Phase 2, create a class diagram to represent the entity, boundary, and control object types. Also show related attributes, operations, relationships and their types along with multiplicities between the object types.

3) Sequence Diagram and Communication Diagram

Based on the UML Use case modeled in Phase 2, and the UML Class diagram, develop UML sequence diagrams. One - Two UML Sequence Diagrams are sufficient.

Phase 4:

In this phase, you will make the system design model of the proposed system you used in Phase 2 and Phase 3.

- At least one system-level architectural diagram, e.g., your system's context model.
- First make an initial system design (software architecture) based on the functional requirements. Your design should include subsystems, their inter-dependencies, and a short description of each subsystem.
- If a database table is used, each table should have attributes showing the fields (columns) and its primary key(s). For instance, a table containing information on employees may look like the following one (note that the primary key, shown in bold, is SSN):

SSN	Last Name	First Name	Position	Department	Office	Telephone	Email
------------	-----------	------------	----------	------------	--------	-----------	-------

- Initial hardware design [optional, but required for projects which include hardware components]. The system hardware component organization can be shown here.
- Template of User Interface Design Patterns for the proposed software system. At least 6 – 8 snapshot designs with brief descriptions. These snapshots must contain details of the main components of the user interface. (e.g., panels, toolbars, menus, menu items, buttons, textboxes, etc.) should be presented, and the format of the output result, reports and/or statistics/ charts must be displayed.

Phase 5 – Implementation and Testing Phases:

You are encouraged to implement your software system as stated in all the phases using the appropriate language, frameworks, platforms, tools, and databases of your preference.