Group Project Description
Fall-2023

Course name:	**Software Engineering 1**
Course code:	**CPCS351**
Syllabus topic area:	**Developing Software project**
Weighting % of the program:	**30%**

Outcomes:

CLO#5: Identify both functional and non-functional requirements in a given requirements specification for a software system by utilizing the knowledge of the requirements gathering

CLO#6: Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system.

CLO#8: Create appropriate design models for the structure and behaviour of software products from their requirements specification.

CLO#10: Depict UML diagrams that illustrate the structure and the dynamic behavior of a system (Class diagrams, sequence diagrams, activity diagrams and state diagrams).

CLO#13: Communicate orally and in writing the results of applying the acquired knowledge of software engineering phases of analysis and design and their relevant techniques to a small sized group project.

Objective:

You will be assigned to a group of students to identify and analyst a real-world problem. And as a group, you are required to design a good reasonable diagram in an attempt to tackle and solve that issue.

Instructions:

- Choose a group of **Four** students from your classmate to conduct this project with. Once you have established your group, you need as a group to appoint **a group leader** throughout these processes. (all the group members must agree on the appointed leader).
- You need to come up with a research question by collective effort by all the learn members. You can use any method that you agree on to reach a research question.

- Each group will be carrying out 2 presentations throughout the period of this semester.

Tasks:

Phase 1:

Each group is asked to choose an **idea of a project** from given ideas that is relevant to real world problems.

- Identify the system's stakeholders
- Project Name
- Project ID.
- Team Members
- Problem Description
- Project objectives
- Sources of Domain Analysis Information
- Scope of the system (main features) / Include and Exclude Features of the system

Phase 2:

Each group is required to demonstrate the following:

- Utilize Use Case Model
  - Complete Model (UML Use Case model)
- Four main use case descriptions (excluding log-in log-out)
  - Select Four main use cases and provide a detailed use case description (tables)
- Functional Requirements
  - Specify all your system functional requirements.
- Non-Functional Requirements
  - Specify all your system non-functional requirements that are important to your system

Phase 3: Static modelling

- Applying **Domain Model:** The diagram must model all the domain classes, required associations, attributes.
- **UML Class diagram:** The diagram must model all the domain classes, required associations, attributes, and methods. Apply design patterns when necessary.
- **System Architecture:** Depict the overall architecture of your system Make use of UML power to express your components, subsystems, hardware, and the architectural pattern of your system

Phase 4: Dynamic modelling

- **Sequence Diagrams for two main tasks:** - Analyze your system's functionalities then choose two main tasks which represents the main features of your system. - Represent them by sequence diagrams. - The actors, objects, and methods, steps in the sequence diagrams should already exist in your specification and your UML Domain Model.
- **Activity Diagram for two main tasks:** Illustrate the workflow of an object, system, or component of the system by drawing the corresponding activity diagram.
- **State Diagram for two main tasks:** Depict a state diagram to model the dynamic external behavior of an object in your system. You may alternatively model the behavior of your system or component of the system.
- **Test Design Techniques**

Report

Submit your report that contains the above deliverables items after enhancement from feedback you had in previous presentations.

- A Softcopy in Word Format must be submitted through blackboard.
- The file name should be written as:" CPCS351-Final Report-Project ID."

Project Deliverable	Main items to be covered	Grade
Presentation#1	• Phase 1 and 2	3
Final Presentation	• All Phases	5
Report	Submit your report.	22
Total	22(Group assessment) + 08 (Individual assessment)	

Rubric		
Phase 1:	**03**	**DeadLine**
Marking guidelines: • Introduction	0.5	11/9/2022
• Problem Description	0.5	
• Project objectives	0.5	
• Sources of Domain Analysis Information	0.5	
• Goal and Scope of the system	0.5	
• Include and Exclude Features of the system	0.5	
Phase 2: Business Requirements Specifications	**05**	
• UML Use Case Diagram	2	29/9/2022
• Use Case Descriptions (Table)	1.5	
• Functional Requirements & Non-Functional Requirements	1.5	
• Presentation 1	**03**	
Phase 3: Software Design and Structuring	**05**	20/10/2022
• Domain Model and class diagrams	05	
Phase 4:Modeling, Interaction & Behaviour	**09**	
		3/11/2022
• Sequence Diagrams	2.5	
• State Diagrams	2.5	
• Activity Diagrams	2.5	
• Testing	1.5	
• Final Presentation	5	
Total	**30**	

PROJECT TEMPLATE

Insert any logo if you want here

<Project Name>

Version 0.1
Status: Draft

Date	Version	Description	Authors
	0.1		
	0.2		

Phase 1: Project Description

1.1 Introduction

A general introduction to the topic you will be discussing. Many books recommend writing your introduction last, after you finish your project. This is to make sure that you introduce what you are actually going to say.
If your project changes in the creating process, it is important to make sure that your introduction accurately reflects what you will be saying. If, however, you have written a good outline and stick to it, then it is fine to start writing your introduction first. Just make sure in your proofreading that you have kept the thread consistent throughout the paper.
The length of your introduction depends on the length and complexity of your project, but generally it should not exceed one page unless it is a very long project or a book. The average length of an introduction is one half a page.

1.2 Project description and objectives

Objectives are statements that describe what the project will accomplish, or the business value the project will achieve. Objectives should refer to the deliverables of the project. An example of an objective statement might be to "upgrade the helpdesk telephone system by December 31 to achieve average client wait times of no more than two minutes".

Objectives are the things you and your team do to help you meet your goal. They are things that can be measured to tell you if your project was successful.

To work out the objectives of your project, ask yourself:

What actions will contribute to achieving of the goal?

What outcomes (results) do I expect? (Be realistic — think about the environment you are working in and the resources you are likely to have)

What can I measure to see if the project goals have been achieved? (eg. the number of people who have health checks)

What is the timeframe for the objective?

Write objectives of your project.

Objective 1

Objective 2

…

1.3 Project team

Write about your team, for example team leader and members details their work distribution etc.

Goals and Scope

1.4 Project Goals

A goal is often written using words like “to raise…to increase…to promote… to reduce”. The goal of your project identifies where you want to be or what you ultimately want to do.

Write different goals for your project.

1.5 Scope and Sources of Domain Analysis Information

The project scope of work is what is required to be delivered. It is important that the project scope statement is clear, unambiguous and easy to understand. It should also be as detailed as possible leaving the reader in no doubt what is being delivered as part of the project.

When documenting the project scope try to think by using the mnemonic SMART.

1. Specific – The project scope needs to be accurate and leave no doubt as to what the project will achieve.

2. Measurable – How will the sponsor know when the project is complete.
3. Achievable – Make the project as small as possible after all it is easier to eat an elephant one bite at a time. It is far easier to manage a few smaller projects than one big one.
4. Realistic – Make the project easy to deliver, if it is over complicated then it is likely to hit problems and run over budget, be delivered late or of poor quality.

5. Timely – Does the project have to be complete by a certain date? If so put it in the scope that the project has to be complete by that date.

1.6.1 Included

Write what you will be covering in your project.

1.6.2 Excluded

Write what you will NOT be covering in your project.

Phase 2: Business Requirements Specifications

The Business Requirements Specification is a document, which reports the finding of the project team after analysis of the client's requirements. The document specifies the project team's proposed solutions relating to the software and hardware requirements of the system that is to be developed. It should be written in terms that the customer could understand. An example document outline that can be used as a model for the Business Requirements Specification is presented as follows.

2.1 Domain analysis

2.2 Requirements & its types

2.3 Techniques for gathering data

2.4 Use case description

2.5 Use case diagram for given problems.

2.6 Difficulties & risk analysis in the domain

Phase 3: Design and Structuring

The software design document is prepared for programmers and future maintainers of the system, to specify the detailed architectural structure of the software. An example document outline that can be used as a model for the Design & Structuring is presented as follows.

3.1 Converting use cases to class diagram

3.2 UML class diagram

3.2.1 Association

3.2.2 Multiplicity

3.2.3 Generalization

3.2.4 Object diagrams

Phase 4: Modelling, Interaction & Behaviour

4.1 Interaction diagram

4.1.1 Sequence diagram

4.1.2 State diagram

4.1.3 Activity diagram

- **Sequence Diagrams for two main tasks:** - Analyze your system's functionalities then choose three main tasks which represents the main features of your system. - Represent them by sequence diagrams. - The actors, objects, and methods, steps in the sequence diagrams should already exist in your specification and your UML Domain Model.

- **Activity Diagram** Illustrate the workflow of an object, system, or component of the system by drawing the corresponding activity diagram. [minimum two activity diagram]

- **State Diagram** Depict a state diagram to model the dynamic external behavior of an object in your system. You may alternatively model the behavior of your system or component of the system. [minimum two activity diagram]

- **System Architecture diagram.**

TESTING

Objectives : Provide a concise summary of the test plan objective.

Testing Strategy : Specific test plan components include: Purpose for this level of test, Items to be tested, Features to be tested, Features not to be tested, technical approach, Pass / Fail criteria

Approach : For each level of testing there should be a test plan and the appropriate set of deliverables. Identify the inputs required for each type of test. Specify the source of the input. Also, identify the outputs from each type of testing and specify the purpose and format for each test output.