TERM 2510
CSE 6234 Software Design
Coursework Instruction Document

PROJECT (20%) AND ASSIGNMENT (30%)

PROJECT (20%) INSTRUCTIONS

Objectives:

This is an assessment to review your understanding of the fundamental concepts about Software Design.

Coursework Learning Outcome:

- Identify software design patterns in the construction of a software application.
- Exhibit applied knowledge of software design concepts and software design principles when designing a software application.

Deliverables:

- SOFTWARE DESIGN PROPOSAL & DESIGN PATTERN DOCUMENT (1 Report)
- A FUNCTIONAL PROTOTYPE

Submission Deadline:

- Submission deadline for Project is **4th May 2025 (Sunday), 11.59 pm.**
- Please strictly adhere to the important dates above.
- NOTE: A kind reminder for every students that **penalty for late submission** will be applied between 5 – 100% coursework mark deduction.

Group Creation:

Please form and create your group within the same Tutorial section ONLY.

File Submission Requirements:

- Provide your answers in a **PDF Format file**, in the correct sequence.
- Name your file in the following format: **Assgn_SD2510_TutorialClass_GroupName.pdf**

  Example:

  Prj_SD2430_TT1L_Venus.pdf

- Submit your final softcopy (PDF file) of your Assignment in **eBwise https://ebwise.mmu.edu.my/ in the placeholder provided**

Instructions:

1. Students are to form groups. Please work in a group of **4**. No individual submission is allowed. Please form your group within the **SAME TUTORIAL** section. *Please update your group member's details in the file provided in Teams*.

2. The final product should be a well-documented, **SOFTWARE PROPOSAL AND SOFTWARE DESIGN PATTERN DOCUMENTATION** that demonstrates the effective use of design patterns to solve the requirements and software attributes wanted.
3. Titles to choose from:

TITLE OPTIONS:
Develop a software/solution within SDG Goal 10, 11 and 12.
https://sdgs.un.org/goals

System Examples:

- Circular marketplace web applications
- Food banks, shelters, public services, etc.
- Community resource mapping application
- Resource sharing applications

4. Students have the freedom to define the business rules and determine the assignment scope. The software design should reflect the defined business requirements (functional and non-functional) and business rules.

Assignment Detail:

Prepare your report according to the sections as listed below. You can **add-in** other main sections or subsections according to the way your report (content) is being presented.

Cover Page
Table of Contents
Abstract (Executive Summary)
1. Introduction
1.1. Abstract
1.2. Problem Statement
1.3. Project Objectives
1.4. Literature Review – 3 reviews
1.5 Project and System Scope

2. **System Overview**
    2.1 Generic Use Case
    2.2 Class Diagrams
    2.3 Entity Relationship Model
    2.4 Object Diagrams
3. **Functional and Non-Functional Requirements**
    3.1 Software Design Concepts and Design Principles
4. **Proposed Design Patterns**
    a. **Design Pattern 1**
        i. **Function or Software Component Affected**
        ii. **Description of work flow or data flow**
        iii. **Sample Class Diagram**
        iv. **Sample Other UML Notations (ex State Diagram, WorkFlow Diagram)**
        v. **Sample Potential Code**
        vi. **Benefit**
        vii. **Limitations**
    b. **Design Pattern 2 (same as no 1)**
    c. **Design Pattern 3 (same as no 1)**
    d. **Design Pattern 4 (same as no 1)**
    e. **Design Pattern 5 (same as no 1)**
5. **Conclusion and suggestion**
6. **Bibliography/Reference**

In your report, you can write details explaining what is about the program, which include definitions. In addition, please also explain by showing screenshots about the program that it can run and explaining how the code work for the program files.

Warning:

- Each group is to submit one report (in .pdf) to your lecturer in eBwise based on their submission mode.
- Late submission, non-contributing members or plagiarism of content is to be penalized with **zero** mark.

PROJECT (20%) MARKING RUBRICS:

The requirements and marks allocations for ASSIGNMENT are defined below:

Items	Very High	High	Moderate	Low - 0
INTRODUCTION • Abstract • Literature Review • Problem Statement • Solutions and Justification • Objectives • Background of System • Project Scope	Effectively introduces the topic / occasion and provides information important for audience comprehension. Excellent abstract that concisely summarizes the entire project, clearly states the problem, approach, methodology, results, and significance related to the relevant theme. Language is precise and compelling.	Adequately introduces the topic / occasion and provides information important for audience comprehension. Good abstract covering most key elements (problem, approach, methodology, results, significance) with clear theme connections. Minor improvements in conciseness possible.	Basic abstract present but missing several key components or SDG connections. Lacks clarity or conciseness.	Fails to introduce the topic / occasion and does not provide information important for audience comprehension. Does not attempt to create interest for the rest of the speech.
SOFTWARE DESIGN CONCEPTS AND PRINCIPLES • Processes involved. • How would you design? • Considerations when design? • Software Design plan • Requirements and Task Plans • System design considerations • Architecture ideas	**Outstanding Design Documentation**: Comprehensive coverage of all design aspects with exceptional clarity. Includes detailed process models, thoughtful architectural decisions explicitly linked to the theme, well-justified design considerations, complete requirements with prioritization, thorough task plans with timeline, and innovative system architecture with clear rationale.	**Strong Design Documentation**: Clear and thorough coverage of most design elements. Process models are well-defined, design approach is thoughtful most considerations are addressed, requirements are clear with good task planning, and system architecture is appropriate with sustainability features. Minor improvements possible.	**Basic Design Documentation:** Minimal coverage of required design elements with underdeveloped process models, simplistic design approach, limited considerations, incomplete requirements, vague task plans, and basic architecture lacking detail. Weak theme connections.	**Unacceptable or Missing Design Documentation**: Completely inadequate or absent coverage of design concepts and principles. No evidence of understanding software engineering practices or theme integration.
PLANNED AND DETAILED CODE DESIGN • Design Pattern 1 • Design Pattern 2 • Design Pattern 3 • Design Pattern 4 • Design Pattern 5	All five design patterns are meticulously documented with outstanding clarity and purpose. Each pattern includes comprehensive UML diagrams, detailed implementation strategy, thorough justification for selection with explicit connections to the theme and thoughtful analysis of alternatives. Code examples demonstrate deep understanding of pattern principles.	**Strong Code Design**: All five design patterns are well-documented with clear UML diagrams and good implementation details. Strong justification for pattern selection with SDG connections. Most patterns show consideration of alternatives. Code examples demonstrate solid understanding	**Basic Code Design**: Some design patterns (3-4) documented with simplified UML diagrams. Implementation strategy lacks significant detail. Weak justification for pattern selections with minimal SDG connections. Little to no discussion of alternatives.	**Poor Code Design**: Few design patterns (1-2) documented with incorrect or missing UML diagrams. Implementation strategy is vague or inappropriate. Minimal justification for pattern selections No discussion of alternatives. Code examples show misunderstanding of patterns.
CONCLUSION, SUGGESTIONS AND REFERENCES	Conclusion is clear, concise and effective. Reflects the content and main idea in an original and interesting way. Makes the audience feel excited about future events / to find out more.	Conclusion is clear, concise and effective. Reflects the content and main idea in an original and interesting way.	Conclusion is weak and somewhat unrelated to the main idea.	Lacks a proper conclusion. Main idea is not restated. Left the audience feeling confused.
Total marks	**20**	**15**	**10**	**5**

ASSIGNMENT (30%) INSTRUCTIONS

Objectives:

This is an assessment to review your understanding of the fundamental concepts about Software Design.

Course Learning Outcome: Display the qualities of good software design in a prototype software with leadership skills.

Deliverables:

- Working Software Prototype (80-95% functions)
- Live Presentation with Presentation Slides - Full Class Diagram Improvised, Component Diagram, Software Architecture
- 3-Minute Individual Presentation on your contribution and mention the percentage of overall contributions.

Submission Deadline:

Submission deadline for Assignment is **16th June 2025 (Monday), 11.59 pm**

Submission Requirements:

GROUP SUBMISSION

- Working Prototype Software
- Slide Presentation. Content should include:
  - **System Objectives and deliverables**
  - **System Scope**
  - **System Overview**
  - **Component Level Diagram**
  - **Software Architecture**
  - **Input and Output**
  - **Solution achieved?**
  - **Work Segregation**
- **30-minute group presentation – well dressed and act like this is a business pitching**

INDIVIDUAL SUBMISSION

- 3-minute Presentation ONLY
- Slide Presentation on - **Self-Reflection Report:** Each student to share a brief report outlining their contributions, challenges faced, and areas for improvement.

Instructions:

7. Maintain the group combination from the first project. Please work in a group of **4**. No individual submission is allowed. Please maintain your group within the **SAME TUTORIAL** section. *Please update your group member's details in the file provided in Teams/ EbWIse.*

8. The final product should be a working prototype system, that demonstrates the effective use of design concepts, design principles, software design patterns to address the specified problem domain.

9. Students have the freedom to define the business rules and determine the assignment scope. The software design should reflect the defined business rules.

Warning:

Be fully aware of assignment deadline date.

Each group is to submit:

1. Working copy of software and all source codes and instruction to your lecturer in eBwise
2. Group Slide presentation
3. Live Recorded Presentation

- Individual Submission:
    1. 3-minute individual presentation
    2. Slide Presentation explaining contributions- **Self-Reflection Report:** Each student to share a brief report outlining their contributions, challenges faced, and areas for improvement.

- Late submission, non-contributing members or plagiarism of content is to be penalized with **<u>zero</u>** mark.
- Individual participation is taken into account by deducting or adding the base marks for the group

ASSIGNMENT (30%) ASSESSMENT RUBRICS:

The OVERALL requirements and marks allocations for Assignment 1 are defined below:

Items	Very High	High	Moderate	Low - 0	Weight
WORKING SYSTEM DURING PRESENTATION	Excellent on these: • Functionality & Working Software • Applied Software Design Patterns • User Interface Design • Proposed System Architecture	Very Good • Functionality & Working Software • Applied Software Design Patterns • User Interface Design • Proposed System Architecture	• Functionality & Working Software • Applied Software Design Patterns • User Interface Design • Proposed System Architecture	• Functionality & Working Software • Applied Software Design Patterns • User Interface Design • Proposed System Architecture	10
GROUP PRESENTATION & SLIDE PRESENTATION	Excellent presentation, critically is stated clearly and described comprehensively, delivering all relevant information necessary for full understanding.	Fair and High	Poor	Very Poor	10
INDIVIDUAL CONTRIBUTION AND PRESENTATION – • Software Component involved. • Base contribution • Value-added contribution •	Student demonstrates greater than required knowledge by answering all questions with explanations and elaborations. Clear fair share of wonderful work result	Student is at ease with expected answers to all questions but fails to elaborate. Fair contribution	Group is uncomfortable with information and can answer only basic questions. Poor Contributions	Group does not have a grasp of information, and cannot answer questions about the subject. Very Poor contribution	10
Total marks					

BREAKDOWN: SYSTEM & PROTOTYPE RUBRIC

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (50-74%)	Needs Improvement (0-49%)	Weight
Functionality & Technical Execution	Prototype works flawlessly with no bugs or crashes. Includes all key functionalities as per the assignment requirements	Prototype is functional with minor bugs, and most key features are implemented	Prototype has major bugs or missing functionalities, but the basic concept is there	Prototype is incomplete, crashes frequently, or lacks essential features	35%
User Interface (UI) Design	UI is intuitive, well-designed, and aesthetically pleasing. Adheres to user-centered design principles	UI is functional and mostly intuitive but lacks refinement or minor usability issues exist	UI is basic, with some confusing elements or poor design choices	UI is poorly designed, hard to navigate, or not usable	30%
Innovation & Creativity	Shows significant creativity and innovation in addressing the problem. Includes unique features or novel approaches	Some creative elements present, but relies on standard approaches	Limited creativity; prototype largely relies on basic, conventional methods	Lacks creativity or uniqueness; merely repeats standard examples without thought	20%
Code Quality	Code is clean, well-organized, and follows best practices (e.g., appropriate naming conventions, comments, modularity)	Code is readable but has some minor issues with organization or consistency	Code is functional but poorly organized, hard to follow, or lacking documentation	Code is unreadable, disorganized, or contains significant errors	15%
					100%

BREAKDOWN: GROUP PRESENTATION AND SLIDE PRESENTATION

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (50-74%)	Needs Improvement (0-49%)	Weight
Understanding of Design Patterns & Component Level Design	Demonstrates a thorough understanding of multiple design patterns (e.g., Singleton, Factory) and integrates them effectively into the prototype	Shows a good understanding of design patterns, with minor flaws in the integration	Demonstrates basic understanding, but has significant issues in the application of patterns	Shows little to no understanding of relevant design patterns or incorrectly applies them	35%
Presentation & Documentation	Clear and detailed documentation that explains design choices, functionality, and future improvements. Excellent presentation of the prototype	Documentation is clear but lacks depth in some areas. Presentation is good but could be more polished	Minimal documentation with gaps in explaining important aspects. Presentation is unpolished	No or very poor documentation. Prototype is poorly presented	35%
Team Collaboration	Evidence of strong collaboration with clearly defined roles. Team worked together seamlessly	Team collaboration is good but uneven workload distribution or minor issues	Team worked together but encountered some significant challenges	Lack of collaboration; project was completed by one member or poor communication evident	30%
					100%

BREAKDOWN: INDIVIDUAL CONTRIBUTION

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (50-74%)	Needs Improvement (0-49%)	Weight
Participation & Engagement	Actively participates in all project phases, consistently attends group meetings, and contributes valuable ideas	Participates regularly in most project phases, attends most group meetings, and provides useful input	Participates sporadically, attends meetings but does not contribute significantly	Rarely participates, misses meetings, or contributes little to the group effort	25%
Quality of Individual Contributions	High-quality contributions that significantly enhance the project (e.g., code, design elements, or documentation)	Good quality work that contributes positively to the project, but lacks some polish	Work is completed but with notable quality issues, requiring major revisions	Contributions are incomplete, incorrect, or of poor quality	25%
Initiative & Responsibility	Takes initiative, goes beyond assigned tasks, and proactively solves problems without needing to be prompted	Completes assigned tasks well and occasionally takes on additional responsibilities when needed	Completes assigned tasks, but needs reminders or encouragement to take initiative	Frequently avoids responsibility, does not complete assigned tasks, or relies on others to complete the work	25%
Communication Skills	Communicates effectively, keeps the team informed, and actively participates in discussions to move the project forward	Communicates well, though may occasionally miss updates or discussions, still contributes to team coordination	Communicates inconsistently, occasionally misses key updates or discussions, affecting team cohesion	Rarely communicates or fails to provide essential updates, which negatively impacts the team's progress	25%
					100%