

## **SOFTWARE ENGINEERING (CS 3205) PROJECT:**

*Rubric 1: Create a software design document that reflects the design of a software project that answers an existing need and/or innovates existing software solutions.*

*Rubric 2: Apply the software engineering code of ethics and professional practice when designing the software project.*

A **software design document (SDD)** is a comprehensive document that outlines the design and architecture of a software system. It provides detailed information about the system's structure, components, interactions, and how it meets the specified requirements. While the exact content and format of an SDD can vary depending on the organization and project, please refer to the following outline for your software design document:

### **TITLE PAGE:**

Project Name

Document Title (Software Design Document)

Date of Creation

Version Number

Author's Name

Teacher and Subject

### **TABLE OF CONTENTS:**

To include sections and subsections

## **1. Introduction /Rationale**

Purpose of the document

Scope of the software project

Objectives and goals

Overview of the software system

## **2. System Architecture**

High-level system architecture diagram

Explanation of the architectural components

Interaction between components

Rationale behind the chosen architecture

## **3. Software Components**

Detailed description of software components/modules

Class diagrams, package diagrams, or other relevant diagrams

Purpose and responsibilities of each component

Dependencies and relationships

## **4. Data Design**

Database schema or data structure design

Explanation of data entities and relationships

Data access and storage methods

Data security and encryption, if applicable

## **5. User Interface Design**

Description of the user interface (UI) components

Wireframes, mock-ups, or UI design diagrams

Explanation of user interactions and flow

User experience considerations

## **6. System Behavior**

Use case diagrams or flowcharts illustrating system behavior

Description of key system behaviors and scenarios

Input and output data for each behavior

Error handling and exceptions

## **7. Security Design**

Security requirements and considerations

Authentication and authorization mechanisms

Data encryption and protection

Security policies and best practices

## **8. Performance Design**

Performance requirements and benchmarks

Scalability and load balancing strategies

Caching mechanisms, if applicable

Resource utilization and optimization techniques

## **9. Deployment**

Hardware and software requirements

Deployment architecture

Configuration management

Deployment procedures and scripts

## **10. Testing and Quality Assurance**

Test plan and strategy

Test cases and scenarios

Quality assurance processes and standards

Bug tracking and resolution procedures

## **11. Maintenance and Support**

Documentation and user guides

Future enhancements and feature requests

Support and maintenance procedures

Version control and release management

## **12. Conclusion**

Summary of the key design decisions

Any open issues or concerns

Sign-off section for project stakeholders

## APPENDICES:

Additional information, diagrams, or reference materials