**Final Project: Object Design Document (Summer 2024)**

**Overview**

The Object Design Document (ODD) for the Final Project is the last document to be completed before implementation. It provides a detailed blueprint of how the project will be implemented, including UML class diagrams and thorough descriptions of each component. This document is essential for developers to execute the implementation accurately and efficiently.

**Purpose**

The purpose of this document is to outline the object-oriented design for the final project, ensuring that all components are well-defined and ready for the implementation phase. By providing detailed information, this document will guide developers in building the system according to the specified requirements.

**Contents**

1. **Introduction**
   * Project Overview
   * Objectives
   * Scope
2. **System Architecture**
   * Overview of the System
   * Deployment Diagram
   * Basic Architectural Design and Architectural Design Diagram
   * Description of Subsystems
3. **Class Diagrams**
   * UML Class Diagrams
   * Descriptions of Classes and Relationships
4. **Class Descriptions**
   * Detailed Description of Each Class
   * Attributes
   * Methods
   * Relationships
   * Responsibilities
5. **Sequence Diagrams**
   * Key Interactions
   * Workflow of the System
6. **Implementation Details**
   * Environment Setup
   * Libraries and Frameworks
   * Coding Standards and Conventions
7. **Use Case Realization**
   * Mapping of Use Cases to Class Diagrams
   * Example Scenarios
8. **Design Patterns**
   * Patterns Used in the Design
   * Justification for Use

**Detailed Sections**

**1. Introduction**

**Project Overview:** This section provides a brief description of the Puzzle Map project, its main features, and the intended user base.

**Objectives:** Outlines the goals of the project and what it aims to achieve upon completion.

**Scope:** Defines the boundaries of the project, including what is and what is not included.

**2. System Architecture**

**Overview of the System:** This section provides a high-level view of the Puzzle Map system architecture, illustrating how the different components interact to achieve the desired functionality. It includes the deployment diagram, basic architectural design, and a detailed description of each subsystem.

**Deployment Diagram:** The deployment diagram illustrates the physical deployment of artifacts on nodes. It shows the hardware and software components involved in the system, including servers, databases, and client devices.

**Basic Architectural Design and Architectural Design Diagram:** The architectural design outlines the logical structure of the system, showcasing the major components and their interactions. The architectural design diagram visually represents these relationships, highlighting the flow of data and control among the system components.

**Description of Subsystems:** View - UI Controller Model

**3. Class Diagrams**

**UML Class Diagrams:** Visual representations of the classes within the system, showing their attributes, methods, and relationships.

**Descriptions of Classes and Relationships:** Detailed explanations of each class, its purpose, and how it interacts with other classes.

**4. Class Descriptions**

**Detailed Description of Each Class:** For each class, provide:

* **Attributes:** List and describe each attribute.
* **Methods:** List and describe each method, including parameters and return types.
* **Relationships:** Describe how the class relates to other classes.
* **Responsibilities:** Outline the responsibilities of the class.

**5. Sequence Diagrams**

**Key Interactions:** Diagrams showing the sequence of interactions between objects for key use cases.

**Workflow of the System:** A step-by-step representation of how the system processes information.

**6. Implementation Details**

**Environment Setup:** Information on the development environment, including hardware and software requirements.

**Libraries and Frameworks:** A list of libraries and frameworks to be used, along with their versions.

**Coding Standards and Conventions:** Guidelines for writing consistent and maintainable code.

**7. Use Case Realization**

**Mapping of Use Cases to Class Diagrams:** How each use case is realized within the class structure.

**Example Scenarios:** Concrete examples of how the system handles specific scenarios.

**8. Design Patterns**

**Patterns Used in the Design:** A list of design patterns employed in the project.

**Justification for Use:** Reasons for choosing each pattern and how it benefits the design.

**Rubrics for Evaluation**

**Rubrics for Evaluation**

| **Criterion** | **Excellent (90-100)** | **Good (80-89)** | **Fair (70-79)** | **Poor (below 70)** |
| --- | --- | --- | --- | --- |
| **Introduction** | Comprehensive overview, clear objectives, and well-defined scope. | Clear overview and objectives, scope mostly defined. | Basic overview, some objectives unclear, limited scope. | Incomplete overview, unclear objectives, undefined scope. |
| **System Architecture - Overview** | Detailed high-level view of the system architecture, showing clear interactions. | Clear high-level view, mostly showing interactions. | Basic high-level view, some interactions unclear. | Incomplete or unclear high-level view of interactions. |
| **System Architecture - Deployment Diagram** | Detailed and clear deployment diagram, well-explained hardware and software components. | Clear deployment diagram, mostly well-explained components. | Basic deployment diagram, some components unclear. | Incomplete or unclear deployment diagram and components. |
| **System Architecture - Basic Architectural Design and Diagram** | Detailed logical structure, clear major components and interactions. | Clear logical structure, mostly clear components and interactions. | Basic logical structure, some components and interactions unclear. | Incomplete or unclear logical structure and interactions. |
| **Description of Subsystems (View)** | Comprehensive description of UI components, clear roles and interactions. | Clear description of UI components, mostly clear roles and interactions. | Basic description of UI components, some roles and interactions unclear. | Incomplete or unclear description of UI components, roles, and interactions. |
| **Description of Subsystems (Controller)** | Comprehensive description of controller components, clear roles and interactions. | Clear description of controller components, mostly clear roles and interactions. | Basic description of controller components, some roles and interactions unclear. | Incomplete or unclear description of controller components, roles, and interactions. |
| **Description of Subsystems (Model)** | Comprehensive description of model components, clear roles and interactions. | Clear description of model components, mostly clear roles and interactions. | Basic description of model components, some roles and interactions unclear. | Incomplete or unclear description of model components, roles, and interactions. |
| **Class Diagrams** | Detailed UML diagrams, clear relationships, and attributes/methods. | Clear UML diagrams, mostly clear relationships and attributes/methods. | Basic UML diagrams, some relationships unclear. | Incomplete or unclear UML diagrams and relationships. |
| **Class Descriptions** | Comprehensive class descriptions with attributes, methods, relationships, and responsibilities. | Clear class descriptions, mostly detailed. | Basic class descriptions, some details missing. | Incomplete or unclear class descriptions. |
| **Sequence Diagrams** | Detailed key interactions, clear workflow representation. | Clear key interactions, mostly clear workflow. | Basic interactions, some workflow steps unclear. | Incomplete or unclear interactions and workflow. |
| **Implementation Details** | Comprehensive environment setup, clear libraries/frameworks, and well-defined coding standards. | Clear environment setup, mostly clear libraries/frameworks, good coding standards. | Basic environment setup, some libraries/frameworks unclear. | Incomplete or unclear environment setup and libraries/frameworks. |
| **Use Case Realization** | Detailed mapping of use cases, clear example scenarios. | Clear mapping of use cases, mostly clear scenarios. | Basic mapping, some scenarios unclear. | Incomplete or unclear mapping and scenarios. |
| **Design Patterns** | Well-chosen patterns, clear justification for use. | Clear design patterns, mostly justified. | Basic patterns, some justifications unclear. | Incomplete or unclear design patterns and justifications. |