Software Design Document (SDD) NALABS Capella Integration

# 1. Introduction

## 1.1 Purpose

State the purpose of this Software Design Document and its intended audience.

## 1.2 Scope

Provide an overview of the software system being designed.  
List the major functionalities and objectives the software system aims to achieve.

## 1.3 Definitions, Acronyms, and Abbreviations

Provide definitions for terms and acronyms used throughout the document.

## 1.4 References

List any references used, such as related documents, architecture diagrams, external systems, etc.

## 1.5 Overview

Provide a brief description of what will be covered in the subsequent sections of this document.

# 2. System Overview

Provide a high-level description of the system's design.  
Include a system architecture diagram that shows the major components and how they interact.

# 3. Design Considerations

## 3.1 Assumptions

List any assumptions that influence the design (e.g., environment, hardware limitations).

## 3.2 Constraints

Describe any constraints such as performance limits, regulatory requirements, or design standards.

## 3.3 Dependencies

Describe any dependencies with other systems or software components.

## 3.4 Risks and Issues

Highlight potential risks and issues related to the design and implementation.

# 4. Architectural Design

## 4.1 System Architecture

Provide an architectural diagram(s) with a detailed explanation of each component.

## 4.2 Design Patterns

Explain any design patterns or methodologies that are employed in the architecture.

## 4.3 Data Flow

Provide data flow diagrams that explain the flow of information through the system.

# 5. Detailed Design

## 5.1 Component 1 (e.g., Database Layer)

Description: Briefly describe the purpose of the component.  
Responsibilities: Outline what this component does and how it contributes to the system.  
Interfaces: Describe how this component interacts with other components.  
Algorithms: Outline any algorithms used by the component.  
Dependencies: Highlight any dependencies this component has on other components or services.

## 5.2 Component 2 (e.g., API Layer)

Repeat the structure above for additional components.

# 6. User Interface Design

## 6.1 UI Overview

Provide an overview of the user interface design.  
Include wireframes or mockups if applicable.

## 6.2 Screen Designs

Detail specific screens and user interactions, including navigation flows.

# 7. Data Design

## 7.1 Data Structures

Define all key data structures used in the system, including database schemas and relationships.

## 7.2 Data Storage and Management

Describe the data storage mechanisms (e.g., database types, data redundancy, etc.).

## 7.3 Data Integrity and Security

Explain how data integrity is ensured, including backup strategies, validation, and access controls.

# 8. Security Design

## 8.1 Authentication and Authorization

Describe the authentication and authorization mechanisms used in the system.

## 8.2 Security Policies

Define the security policies and guidelines (e.g., encryption, secure communication).

## 8.3 Vulnerability Management

Detail how vulnerabilities will be managed and mitigated.

# 9. Performance Considerations

## 9.1 Performance Requirements

Define the performance requirements such as response times, throughput, etc.

## 9.2 Optimization Techniques

Describe techniques used to optimize performance, such as caching, load balancing, etc.

# 10. Testing and Validation

## 10.1 Test Strategy

Outline the testing strategies including unit testing, integration testing, and system testing.

## 10.2 Test Cases

Provide a list of important test cases with the expected inputs, outputs, and conditions.

## 10.3 Validation

Describe how the software will be validated against requirements.

# 11. Deployment and Maintenance

## 11.1 Deployment Architecture

Describe the deployment architecture, including how components are deployed in various environments.

## 11.2 Continuous Integration/Deployment (CI/CD)

Outline the CI/CD process, including automated builds, tests, and deployments.

## 11.3 Maintenance

Describe the maintenance strategy including bug fixes, updates, and monitoring.

# 12. Appendices

Include any additional information, diagrams, or technical details relevant to the software design.

# Sign-off

Include sign-off details to indicate final approval from the stakeholders.