

TUTOR ONLINE

Software Architecture

Project Code: TTO

Document Code: TTO – v1.0

**Hoa Lac, 31/5/2017**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 31/5/2017 | All | A | Create Architecture Design | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

|  |  |  |
| --- | --- | --- |
| AUTHOR: | Tran Viet Vuong  Project Manager (PM) | 31/05/2017 |
| REVIEWERS: | Nguyen Thi Khanh Huyen  Document Leader (DL) | 1/6/2017 |
| Nong Thi Hoai Thuong  Quality Assurance Officer (QA) | 1/6/2017 |
| APPROVAL: | Phan Truong Lam  Lecturer | 17/06/2017 |

TABLE OF CONTENTS

1 Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 4

1.5 Overview 4

2 Architectural Representation 4

3 Architectural Goals and Constraints 4

4 Use-Case View 4

4.1 Use-Case Realizations 4

5 Logical View 4

5.1 Overview 4

5.2 Architecturally Significant Design Packages 4

6 Process View 4

7 Deployment View 4

8 Implementation View 4

8.1 Overview 4

8.2 Layers 4

9 Data View (optional) 4

10 Size and Performance 4

11 Quality 4

12 Other Considerations 4

# Introduction

This document is a part of Tutor Online project, provides a high level overview and explains the whole architecture of the project; and this introduction section presents an over view of the document.

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions that have been made on the system.

This document is intended for both of supervisor and developers of Tutor Online project, submitted to instructor and stored in the team’s capstone project deliverables kit.

## Scope

This document applies to the overall design of the system. It contains information relating to the architectural design of the software, the structure of the system and the constraints.

This explains main parts of system that is architecture of application.

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Note |
| TTO | Tutor Online |  |
| SAD | Software Architecture Design |  |
| Admin | TTO’s administrator |  |
| Manager | TTO’s manager |  |
| Guest | TTO’s guest |  |
| Student | TTO’s student |  |
| Parent | TTO’s parent |  |
| Accountant | TTO’s accountant |  |
| Supporter | TTO’s supporter |  |

## References

|  |  |  |
| --- | --- | --- |
| No | Document | Source/ Web Address |
| 1 | Software Architecture Design Template | FPT Software Cop. |
| 2 | Software Requirements Specification | TTO |
| 3 | Software Architecture Design Illuminated Book | FPTU’s Library |
| 4 | Data Design | TTO |
| 5 | Screen Design | TTO |

## Overview

The Software Architecture Document contains the following subsections:

* **Section 1**: Provide an overview of entire Software Architecture Document.
* **Section 2**: Choice of Architecture Design
* **Section 3**: Architectural Representation
* **Section 4**: Architectural Goals and Constraints
* **Section 5**: Use-Case view
* **Section 6**: Logical View
* **Section 7**: Process View
* **Section 8**: Deployment view
* **Section 9**: Quality

# Architectural Representation

[This section describes what software architecture is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]

# Architectural Goals and Constraints

[This section describes the software requirements and objectives that have some significant impact on the architecture, for example, safety, security, privacy, use of an off-the-shelf product, portability, distribution, and reuse. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.]

# Use-Case View

[This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage - they exercise many architectural elements, or if they stress or illustrate a specific, delicate point of the architecture.]

## Use-Case Realizations

[This section illustrates how the software actually works by giving a few selected use-case (or scenario) realizations, and explains how the various design model elements contribute to their functionality.]

# Logical View

[This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities. You should introduce architecturally significant classes and describe their responsibilities, as well as a few very important relationships, operations, and attributes.]

## Overview

[This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.]

## Architecturally Significant Design Packages

[For each significant package, include a subsection with its name, its brief description, and a diagram with all significant classes and packages contained within the package.

For each significant class in the package, include its name, brief description, and, optionally a description of some of its major responsibilities, operations and attributes.]

# Process View

[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

# Deployment View

[This section describes one or more physical network (hardware) configurations on which the software is deployed and run. At a minimum for each configuration it should indicate the physical nodes (computers, CPUs) that execute the software, and their interconnections (bus, LAN, point-to-point, and so on.) Also include a mapping of the processes of the **Process View** onto the physical nodes.]

# Implementation View

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

## Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]

## Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

# Data View (optional)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

# Size and Performance

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

# Quality

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, for example safety, security or privacy implications, they should be clearly delineated.]

# Other Considerations

[This section provides a description of other approach/ solutions that were considered in selection process for the above architecture, i.e. a brief explanation of advantages and disadvantages of the selected architecture in comparison with others. It should be a clear answer to the question why the above architecture is selected for this system, not the others.]