

<Project Name>

System Architecture Document

Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Baseline Version** | **Approved Date** | **Reviewer** | **Approver** |
| <N.M> | DD-MMM-YY | <Full name> | <Full name> |

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| <N.M> | DD-MMM-YY | <Full name> | <Full name> |

Table of Contents

[I. Introduction 4](#_Toc296677850)

[1. Purpose 4](#_Toc296677851)

[2. Scope 4](#_Toc296677852)

[3. Definition, Acronyms and Abbreviation 4](#_Toc296677853)

[4. References 4](#_Toc296677854)

[II. System Architecture 4](#_Toc296677855)

[1. Overview Architecture 4](#_Toc296677856)

[2. Architectural Goals and Constraints 4](#_Toc296677857)

[3. Alternative Solutions 4](#_Toc296677858)

[III. Business Work Flow 4](#_Toc296677859)

[IV. Use Case View 4](#_Toc296677860)

[V. Data Model (Optional) 4](#_Toc296677861)

[VI. Deployment View 4](#_Toc296677862)

[VII. Implementation View 4](#_Toc296677863)

[1. Overview 4](#_Toc296677864)

[2. Layers 4](#_Toc296677865)

[2.1. Presentation Layer 4](#_Toc296677866)

[2.2. Business Logic Layer 4](#_Toc296677867)

[2.3. Data Access Layer 4](#_Toc296677868)

[VIII. Performance & Security 4](#_Toc296677869)

[1. Performance 4](#_Toc296677870)

[2. Security 4](#_Toc296677871)

[IX. Other consideration 4](#_Toc296677872)

[X. Non Functional List 4](#_Toc296677873)

# Introduction

## Purpose

## Scope

## Definition, Acronyms and Abbreviation

|  |  |  |
| --- | --- | --- |
| **Abbreviation** | **Description** | **Comment** |
| SAD | Software Architecture document |  |

## References

# Business Work Flow

# System Architecture

## Overview Architecture

## Architectural Goals and Constraints

## *Hardware Technical Platform*

## *Software Technical Platform*

## *Security*

## *Persistence*

## *Reliability/Availability*

## *Scalability*

## *Internationalization (i18n)*

## Alternative Solutions

# Use Case View

# Data Model (Optional)

# Deployment View

# Implementation View

## Overview

## Layers

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Component** | **Methods/Functional** | **ViewName** |
| 1 |  | GetSalesAssessmentByUser | UC011- Tổng Quan |
|  |  | GetRotersByUser | UC011- Tổng Quan |
|  |  | GetSalesmanByUser | UC011- Tổng Quan |
|  |  |  |  |

# Performance & Security

## Performance

[This subsection describes all major characteristics software that imparts the system performance.

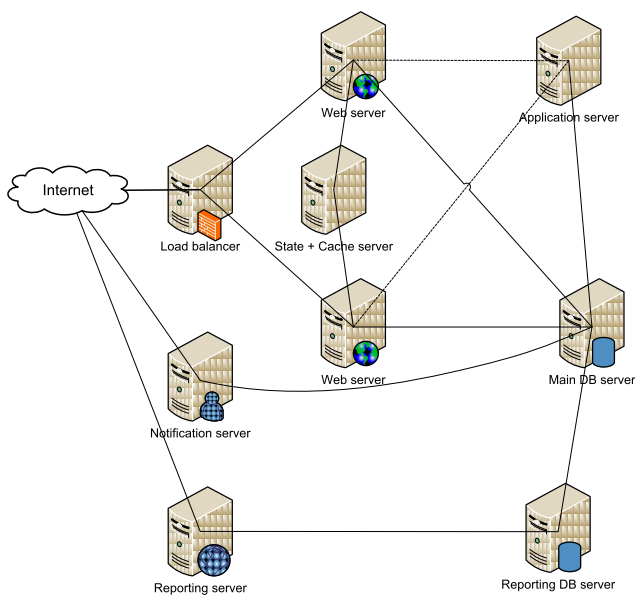
*For reference:*

*The chosen software architecture supports the key sizing and timing requirements, as stipulated in the Supplementary Specification [15]:*

* + 1. *The system shall support up to 2000 simultaneous users against the central database at any given time, and up to 500 simultaneous users against the local servers at any one time.*
    2. *The system shall provide access to the legacy course catalog database with no more than a 10 second latency.*
    3. *The system must be able to complete 80% of all transactions within 2 minutes.*
    4. *The client portion shall require less than 20 MB disk space and 32 MB RAM. The selected architecture supports the sizing and timing requirements through the implementation of a client-server architecture. The client portion is implemented on local campus PCs or remote dial up PCs. The components have been designed to ensure that minimal disk and memory requirements are needed on the PC client portion.*

*Solution to have powerful system with great performance capability:*

* 1. *Use caching system to avoid repeating the same task for multiple requests.*
  2. *Use load-balancing system with multiple server to serve a thousand of simultaneous users  
     Sample of system architecture for this kind of system*

**

]

## Security

[This section list out all security functional requirements and describe briefing of technology solution for each security requirement.

*Below are most of essential security concerns when developing software application:*

*1:* [*Injection*](http://www.owasp.org/index.php/Top_10_2010-A1)*: Web apps may be vulnerable to LDAP, OS, and SQL injection attacks when untrusted, potentially malicious data is sent as part of a command or query, causing a vulnerable interpreter to run commands or access data in an unexpected manner.*

*2:* [*Cross-Site Scripting*](http://www.owasp.org/index.php/Top_10_2010-A2-Cross-Site_Scripting_%28XSS%29)*: Apps that accept and send untrusted potentially malicious data to a Web browser without proper validation and escaping may be vulnerable to XSS attacks that exploit these flaws to execute hostile scripts.*

*3:* [*Broken Authentication and Session Management*](http://www.owasp.org/index.php/Top_10_2010-A3-Broken_Authentication_and_Session_Management)*: Apps that do not correctly authenticate and manage sessions may be vulnerable to attacks that exploit compromised passwords, keys, tokens, etc., in order to falsely assume another users’ identity.*

*4:* [*Insecure Direct Object References*](http://www.owasp.org/index.php/Top_10_2010-A4-Insecure_Direct_Object_References)*: Apps that make references to internal implementation objects such as files, directories, or indices may be vulnerable to attacks that manipulate these exposed references in order to reach unauthorized data.*

*5:* [*Cross-Site Request Forgery*](http://www.owasp.org/index.php/Top_10_2010-A5-Cross-Site_Request_Forgery_%28CSRF%29)*: CSRF attacks occur when a browser is forced to send a forged HTTP request containing a logged-in session user's cookie or other authentication data, causing a vulnerable Web app to treat the forged request as legitimate.*

*6:* [*Security Misconfiguration*](http://www.owasp.org/index.php/Top_10_2010-A6-Security_Misconfiguration)*: Web apps, frameworks, associated servers, and underlying platforms may all be vulnerable to a wide variety of attacks if not kept up-to-date and correctly configured with secure settings.*

*7:* [*Insecure Cryptographic Storage*](http://www.owasp.org/index.php/Top_10_2010-A7-Insecure_Cryptographic_Storage)*: Sensitive data, such as credit cards, social security numbers, passwords, and other credentials must be protected against unauthorized access through correct use of appropriate cryptographic techniques, such as encryption and hashing.*

*8:* [*Failure to Restrict URL Access*](http://www.owasp.org/index.php/Top_10_2010-A8-Failure_to_Restrict_URL_Access)*: Web apps must verify URL access rights not only before rendering protected links and buttons, but every time the underlying "hidden" Web pages are accessed to prevent unauthorized access using forged URLs.*

*9:* [*Insufficient Transport Layer Protection*](http://www.owasp.org/index.php/Top_10_2010-A9-Insufficient_Transport_Layer_Protection)*: Web apps that fail to properly authenticate, encrypt, and protect sensitive traffic as it transits a network may be vulnerable to a variety of confidentiality and integrity attacks.*

*10:* [*Unvalidated Redirects and Forwards*](http://www.owasp.org/index.php/Top_10_2010-A10-Unvalidated_Redirects_and_Forwards)*: Web apps that redirect or forward users to other URLs without proper validation of input data used to make such decisions may be vulnerable to attacks that redirect users to phishing or malware sites.*

]

# Other consideration

[This section describes all other consideration such as multi language.]

# Non Functional List

[This section describes all other functional list. Refer to requirement document for content]

*- End of the document -*