To put in a client name select all (ctrl A) then f9

AhaTo put in a client name select all (ctrl A) then f9

**Architecture Design**

**Version – 0.1**

**25 July 2018**

TABLE OF CONTENTS

[1 INTRODUCTION 1](#_Toc444276080)

[1.1 Purpose 1](#_Toc444276081)

[1.2 Scope 1](#_Toc444276082)

[1.3 Definitions, Acronyms and Abbreviations 1](#_Toc444276083)

[1.4 References 1](#_Toc444276084)

[1.5 Overview 2](#_Toc444276085)

[2 ARCHITECTURAL REPRESENTATION 3](#_Toc444276091)

[2.1 System Architecture 3](#_Toc444276092)

[2.2 Integration 4](#_Toc444276093)

[2.2.1. Connect to Harvey Nash domain 4](#_Toc444276094)

[2.2.2. Connect to TMS Service 4](#_Toc444276095)

[2.2.3. Connect to AGS Service 5](#_Toc444276096)

[3 USE-CASE VIEW 5](#_Toc444276097)

[4 LOGICAL VIEW 5](#_Toc444276098)

[4.1 Overview 5](#_Toc444276099)

[4.2 Common Sequence Diagram 7](#_Toc444276100)

[5 DEVELOPMENT VIEW / IMPLEMENTATION VIEW 8](#_Toc444276101)

[6 PHYSICAL VIEW / DEPLOYMENT VIEW 9](#_Toc444276102)

[7 DATA VIEW 10](#_Toc444276123)

[8 OTHER CONSIDERATIONS 10](#_Toc444276124)

[8.1 Security 10](#_Toc444276125)

[8.2 Authentication 10](#_Toc444276126)

[8.3 Exception Handling 10](#_Toc444276127)

[8.4 Logging 11](#_Toc444276128)

[8.5 High Availability and Scalability 11](#_Toc444276129)

# INTRODUCTION

This document is to give the component as well as class diagram for Aha system.

## 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 software architecture document applies to the features that are defined in the requirement documents. It also affects the way the detailed design document is built.

## Scope

The document will define the architecture of the Aha system. The architecture of the system will be described by MVC architecture.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Reference** |
| MVC | Model View Controller |
| Entities | Data classes and their relationships show a data model for describing a database in an abstract way. |
|  |  |

## References

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Reference** | **Version** | File Name & Links |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Overview

Aha System:

The available main Use Cases are:

# ARCHITECTURAL REPRESENTATION

## System Architecture

<Architecture diagram here>

Figure System Architecture

|  |  |  |
| --- | --- | --- |
| **No** | **Item name** | **Description** |
|  | **Frontend** | JFX |
|  | **Backend** | Spring framework |
|  | **Database** | Aha database is MongoDB. |

# USE-CASE VIEW

This section depicts use cases which are significant with respect to the architecture of the system.

Refer to **Aha - Software Requirements Specification**

# LOGICAL VIEW

## Overview

The Aha application is built on main technologies:

* Frontend: JFX framework. This is presentation layer base on Model-View-Controller (MVC) architectural pattern
* Backend: Spring framework, Java

The System is divided into layers based on the N-Layers architecture



Figure Web Application Architecture Overview

Figure Integrated framework template



Figure Web App Flowchart

## Common Sequence Diagram



Figure Web Application Common Sequence Diagram

# DEVELOPMENT VIEW / IMPLEMENTATION VIEW



Figure Package Diagram

The component modules are described as below:

|  |  |  |
| --- | --- | --- |
| **Step** | **Project Name** | **Description** |
|  | AHA.Web | An JFX MVC package which exposes a Web Interface to user |
|  | AHA.Services | A package that implement the business logic behaviours based on Spring framework |
|  | AHA.DAO | A package for database access |

|  |  |  |
| --- | --- | --- |
| **NO** | **Item** | **Detail** |
|  | Development programming language | * Java |
|  | Development tool | * Eclipse |
|  | Framework/Library | * JFX * Spring framework * JQuery |
|  | Unit Test | * JUnit |
|  |  |  |

# PHYSICAL VIEW / DEPLOYMENT VIEW



Figure Deployment Diagram

The physical nodes in each boundary are described as below:

|  |  |
| --- | --- |
| **Module** | **Description** |
| Web Server | The computer that host the Aha web application running on Apache Tomcat 9.0 |
| Database Server | A MongooDB database |
|  |  |

# DATA VIEW

NA

# OTHER CONSIDERATIONS

## Security

* The system will only require ports for incoming and outbound HTTP/HTTPS to be enabled.

Network traffic will be encrypted utilizing SSL.

* The system shall be designed not to be prone to the OWASP top 10 Web Application Security Risks:
* A1: Injection
* A2: Cross-Site Scripting (XSS)
* A3: Broken Authentication and Session Management
* A4: Insecure Direct Object References
* A5: Cross-Site Request Forgery (CSRF)
* A6: Security Misconfiguration
* A7: Insecure Cryptographic Storage
* A8: Failure to Restrict URL Access
* A9: Insufficient Transport Layer Protection
* A10: Un-validated Redirects and Forwards

## Authentication

Aha system will use form authentication, integration with LDAPT server.

## Exception Handling

There are 2 main rules when handling exception in the project:

* Business or Data layer will catch all exceptions which can be generated by the common language runtime (CLR). Then, they are to be converted to exceptions and to be thrown to the upper calling methods/layers.
* Presentation layer will catch all exceptions and handle them correspondingly.

## High Availability and Scalability

NOT supported.