

**Quiz Practice System**

**Software Design Document**

**Project Code: QPS**

**Document Code: QPS– v1**

– Hanoi, July 2025 –

**Table of Contents**

[1 Introduction 3](#_Toc200092605)

[1.1 Purpose 3](#_Toc200092606)

[1.2 Definitions, Acronyms and Abbreviations 3](#_Toc200092607)

[1.3 References 3](#_Toc200092608)

[2 System Architecture 3](#_Toc200092609)

[2.1 System Overview 3](#_Toc200092610)

[2.3 Assumptions 4](#_Toc200092611)

[2.4 Design Constraints 4](#_Toc200092612)

[3 Software Architecture Design 5](#_Toc200092613)

[3.1 Architectural Representation 5](#_Toc200092614)

[3.2 Software Architecture 5](#_Toc200092615)

[4 Detailed Component Design 10](#_Toc200092616)

[4.1 Common Design 10](#_Toc200092617)

[4.2 <Use-Case ID/Feature ID> - <Use case Name/Feature Name> 10](#_Toc200092618)

[4.3 <Use-Case ID/Feature ID> - <Use case Name/Feature Name 2> 13](#_Toc200092619)

[4.3 <Name of Function Group > 14](#_Toc200092620)

[5. Database Design 14](#_Toc200092621)

[5.1 Database Design 14](#_Toc200092622)

[5.2 Data File Design 15](#_Toc200092623)

## 1 Introduction

### 1.1 Purpose

[Describe the document purposes; what information this document will contain; who readers that this document aims at…are]

### 1.2 Definitions, Acronyms and Abbreviations

*[Describe all common business/technical definitions, acronyms, and abbreviations that are used in the whole document]*

### 1.3 References

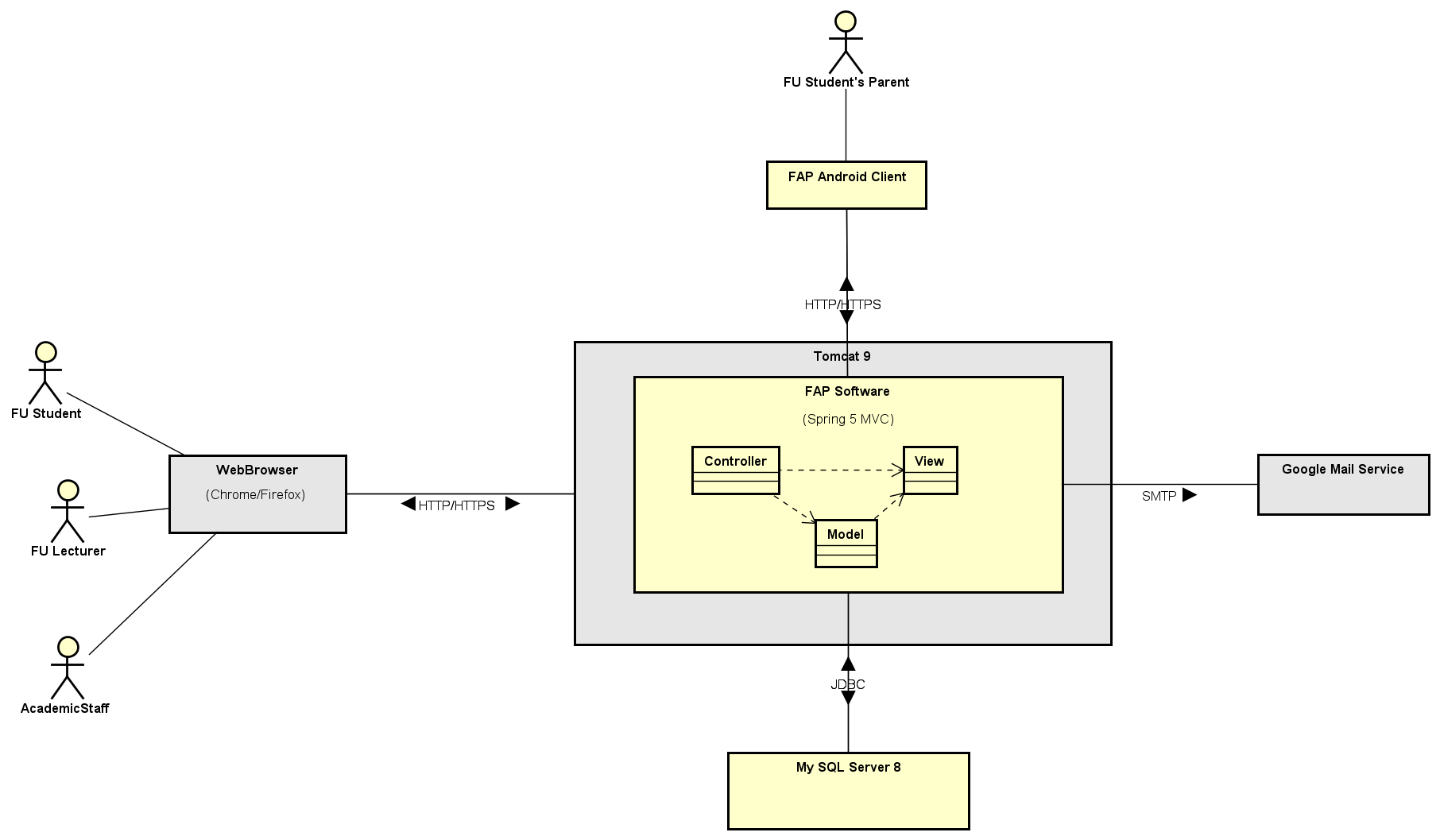
*[List any other documents or Web addresses to which this SDD refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.]*

## 2 System Architecture

### 2.1 System Overview

*[This section presents a high-level architecture of the system in which describe what components the system will have. The component may include user groups, developing software… and other systems/software that will integrate with the developing software]*

*<< Below is an example*

**

**Figure 2-1**: System Architectural Overview

As described in the figure 2-1, the system will have the following components

* **FAP Software**: this is a developing software which will be built on the Spring MVC framework and deployed on the Tomcat 9 application server.
* **Google Mail Service**: This is a mail server that the FAP software shall integrate with in order to authenticate the user and send email notification

*…>>*

### 2.3 Assumptions

[Describe any items, based on which to design future system. These may concern such issues as:

* Related software or hardware
* Operating systems
* End-user characteristics
* Possible and/or probable changes in functionality

Example:

This system is designed basing on these following assumptions:

* Windows XP
* Internet Explorer 6.0
* Acrobat Reader 6.0
* SQL Server 2000
* …
* ]

### 2.4 Design Constraints

[Describe any global limitations or rules that have a significant impact on the design of the system's software (and describe the associated impact). Such constraints may be imposed by any of the following (the list is not exhaustive):

* End-user environment.
* Availability or volatility of resources
* Standards compliance
* Interoperability requirements
* Interface/protocol requirements
* Data repository and distribution requirements
* Security requirements (or other such regulations)
* Memory and other capacity limitations
* Performance requirements
* Network communications
* Verification and validation requirements (testing)
* Other means of addressing quality goals
* Other requirements described in the requirements specification

Example:

This system should be complied with following items:

* Use AJAX to transfer data from browser to server.
* Every process in this system must be less than 10s.
* …]

## 3 Software Architecture Design

### 3.1 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.]*

### 3.2 Software Architecture

*[Select suitable architecture patterns for sub-systems. The content of this section include the overall diagram and the explanation for each of the diagram components]. For example: with the layer pattern, we need to provide the description for each of the layers and the relationship among the layers]*

#### 3.2.1 Process View

*[This section describe the main business process to illustrate the role of software component in the real world or internal processes of software.*

*The process view focuses on the dynamic aspects of the system, i.e., its execution time behavior. This view also derives from the logical view. It is an abstraction of processes or threads dealing with process synchronization and concurrency. It contributes to many nonfunctional requirements and quality attributes such as scalability and performance requirements.*

*The process view looks at the system's processes and the communications among them. A software system can be decomposed into many runtime execution units. How to organize all execution units at runtime is presented in this view. The quality attributes such as performance, scalability, concurrency, synchronization, distribution, and system throughput are all addressed in the process view. This view maps functions, activities, and interactions onto runtime implementation with a focus on nonfunctional requirements as well as the implementation of the functional requirements.*

*The process view takes care of the concurrency and synchronization issues between subsystems. It can be described at several levels of abstraction, from independently executing logical networks of communicating programs to basic tasks running within the same processing node.*

*The process view must also address nonfunctional requirements such as multithreading and synchronous/asynchronous communications for performance and availability. The UML activity diagram and interaction overview diagram support this view.]*

##### 3.2.1.1 < Process name 1>

##### 3.2.1.2 Course Registration Process

**Figure 3-3**: Course registration process

As describe in the Figure 3-3, the course registration process will have the following steps:

**Step 1**: FU student start this process when he/she type URL of FAP website into location field of Web browser.

**Step 2**: UI XX component of FAP software receive a request contain user information, then

* Validate user information.
* If the user information is valid format, the UI XX component will send data to Service YY component to verify user information in database
* Displays the Login screen to require the user to enter account information

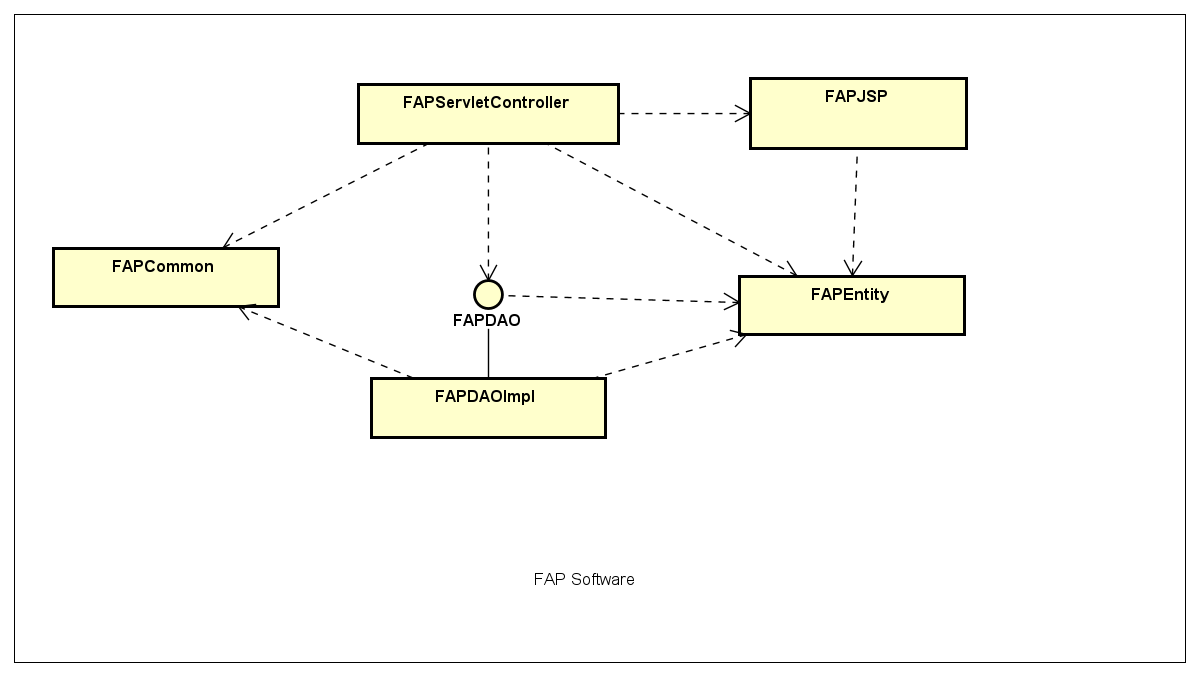
….

>>

#### 3.2.2 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.]*

*<<Sample class diagram*



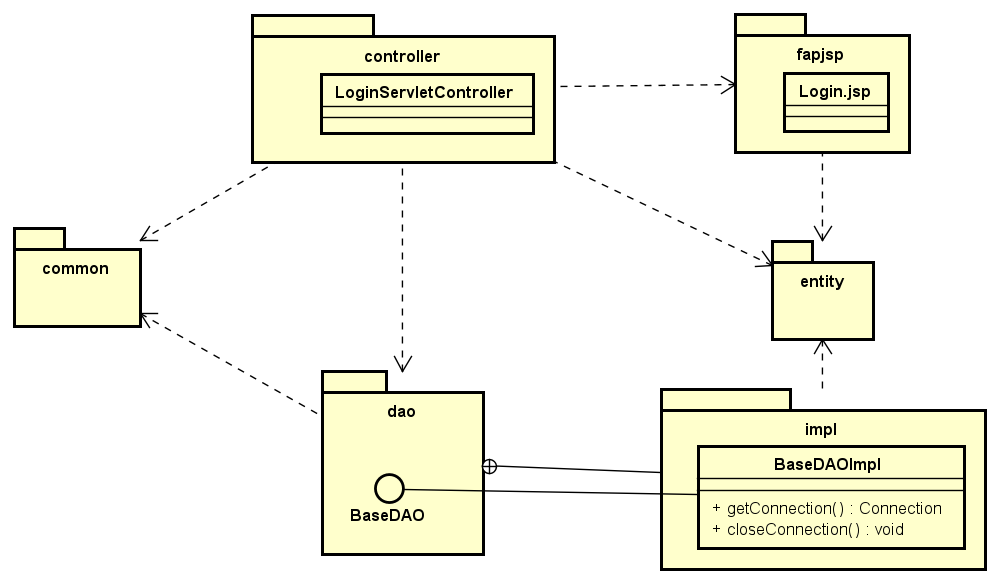
**Figure 3-1**: FAP Logical Overview

>>

#### 3.2.3 Development View

[*This section describes the source code structure of the software or software module when developing. The package or component diagram can be used to illustrate the source code structure*]

##### 3.2.3.1 Package Diagram

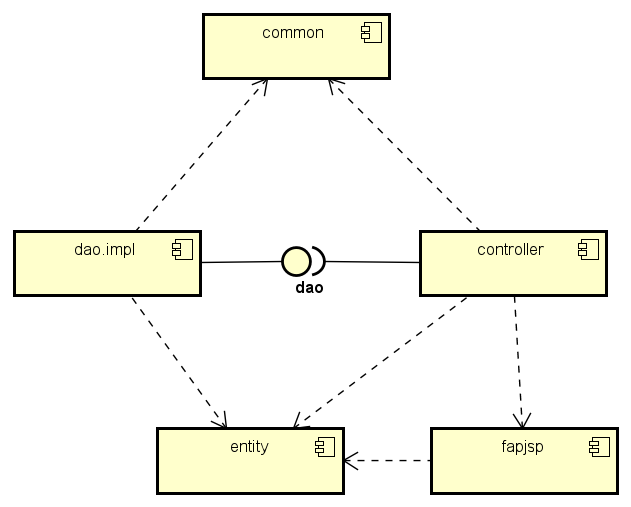


**Figure 3-1:** FAP Package Diagram

***Packages Description***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | <Package name> | <Description of the package> |
| 02 |  |  |

##### 3.2.3.2 Component Diagram [optional]



**Figure 3-2:** FAP Package Diagram

***Components Description***

|  |  |  |
| --- | --- | --- |
| **No** | **Component** | **Description** |
| 01 | < name> | <Description of the component> |
| 02 |  |  |

#### 3.2.4 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.]*

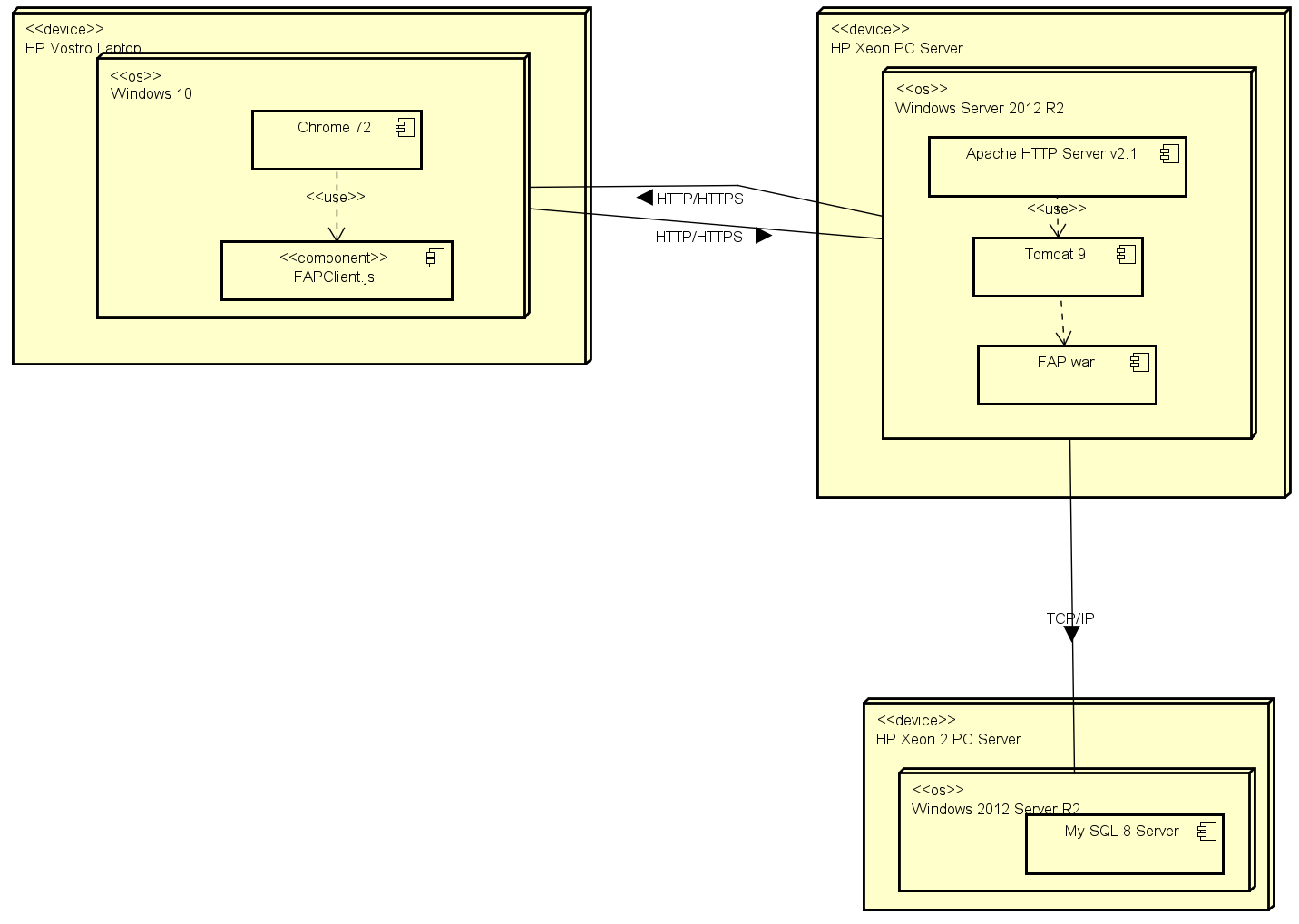
**

Figure 3-4: FAP Deployment structure

***Configuration Description***

|  |  |  |
| --- | --- | --- |
| **No** | **Component/Node** | **Description** |
| 01 | < name> | <Description of the component> |
| 02 |  |  |

## 4 Detailed Component Design

### 4.1 Common Design

*[Provide the common design for all the features/UCs. It include common design for interfaces and process components*

*- Class Diagram, Class Specifications, and Sequence//Communication Diagram (s)]*

*This section should contain the design spec of*

* *Error handling,*
* *Logging,*
* *Messages processing,*
* *UI structures, utility classes/components*
* *Transaction management*

### 4.2 <Use-Case ID/Feature ID> - <Use case Name/Feature Name>

*[Provide the detailed design for the feature <Use case Name1>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

#### 4.2.1 Screen Design

*[Describe screen design and screen description]*

**Table 4-x:** **<Screen Name> Definition**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Object/Control Name** | **Type** | **Required** | **Length** | **Description** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

#### 4.2.2 Class/Component Design

##### 4.2.2.1 Classes/Component Structure

*[This part presents the class diagram for the relevant feature]*



##### 4.2.2.2 Classes/Component Description

*[Provide the description for each class, including both Class Attributes and Class Methods information. Those can be in the table format as below.*

*This section should contain the spec for application class/component logic only to avoid the large spec]*

###### 4.1.2.1 XYZ Class

*[Provide the detailed description for the class methods.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | <Class Name> | | | |
| **Description** |  | | | |
| **Base Class** |  | | | |
| **Constructor** |  | | | |
| **Prototype** |  | | | |
| **Source File** |  | | | |
| **Namespace** |  | | | |
| **Attributes** | Name | Type | Description | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |
|  |  |  |  |

###### 4.2.2.2 ABC Class

*[Provide the detailed description for the class methods]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | <Class Name> | | | |
| **Description** |  | | | |
| **Base Class** |  | | | |
| **Constructor** |  | | | |
| **Prototype** |  | | | |
| **Source File** |  | | | |
| **Namespace** |  | | | |
| **Attributes** | Name | Type | Description | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |
|  |  |  |  |

#### 4.2.3 Object Interactions (Sequence/Communication) Diagram(s)

*[Provide the sequence/communication diagram(s) for the feature, see the sample below]*



### 4.3 <Use-Case ID/Feature ID> - <Use case Name/Feature Name 2>

*[Provide the detailed design for the <Feature/ Use case Name2>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

#### 4.3.1 Screen Design

*[Describe screen design and screen description]*

#### 4.2.2 Class/Communication Design

##### 4.1.2.1 Class/Component Structure

*[This part presents the class diagram for the relevant feature/usecase]*

##### 4.2.2.2 Class/Component Description

*[Provide the description for each class, including both Class Attributes and Class Methods information. Those can be in the table format as below]*

#### 4.2.3 Object Interaction (Sequence/Communication Diagram(s)

*[Provide the sequence/communication diagram(s) for the feature, see the sample below]*

### 4.3 <Name of Function Group >

#### 4.3.1 Class/Component Design

##### 4.3.1.1 Class/component Structure

*[This part presents the class diagram for the relevant feature]*

##### 4.3.1.2 Class/component Description

*[Provide the description for each class, including both Class Attributes and Class Methods information. Those can be in the table format as below]*

#### 4.3.2 <Use-Case ID> - <Use case Name xx>

*[Provide the detailed design for the feature <Use case Name xx>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

##### 4.3.2.1 Screen Design

*[Describe screen design and screen description]*

##### 4.3.2.2 Objects Interaction (Sequence/communication Diagram(s))

*[Provide the sequence/communication diagram(s) for the feature, see the sample below]*

#### 4.3.3 <Use-Case ID> - <Use case Name xx>

*[Provide the detailed design for the feature <Use case Namexx>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

##### 4.3.3.1 Screen Design

*[Describe screen design and screen description]*

##### 4.3.3.2 Objects Interaction (Sequence/communication Diagram(s))

*[Provide the sequence/communication diagram(s) for the feature/usecase, see the sample below]*

## 5. Database Design

### 5.1 Database Design

*[Provide the tables relationship like example below]*



#### a. Table name 1

*[Give some lines about the table here>>*

*[Table fields, in the form of table format as below]*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | Field name1 |  |  |  |  |  |  |
| 2 | Field name2 |  |  |  |  |  |  |

#### b. Table name 2…

### 5.2 Data File Design

*[File List, in the table format as below]*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **File Name** | **Type** | **Notes** |
| 1 | Class name1 |  |  |
| 2 | Class name2 |  |  |
| 3 | .. |  |  |