

**Capstone Project Report**

**Report 4 – Software Design Document**

– Hanoi, Sep 2022 –

**Table of Contents**

[I. Record of Changes 3](#_Toc83349083)

[II. Software Design Document 4](#_Toc83349084)

[1. System Design 4](#_Toc83349085)

[1.1 System Architecture 4](#_Toc83349086)

[1.2 Package Diagram 4](#_Toc83349087)

[2. Database Design 4](#_Toc83349088)

[3. Detailed Design 5](#_Toc83349089)

[3.1 <Feature/Function Name1> 5](#_Toc83349090)

[3.2 <Feature/Function Name2> 6](#_Toc83349091)

# I. Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 08/10/2022 | A | KienNT | Database Tables |
| 09/10/2022 | M | GiangNT | Database Tables |
| 09/10/2022 | A | ThongPQ | System Architecture |
| 12/10/2022 | A | KienNT | Package Diagram, Package Description |
| 13/10/2022 | M | GiangNT | Package Diagram, Package Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

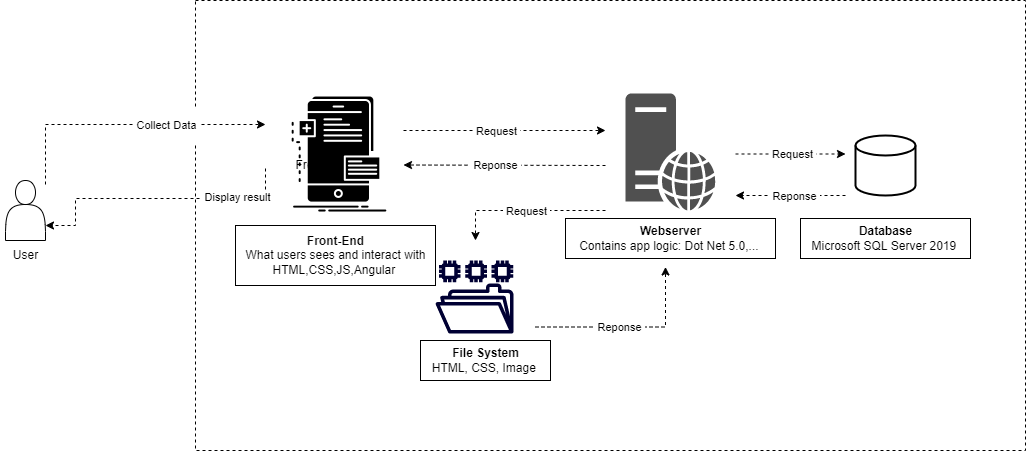
\*A - Added M - Modified D - Deleted

# II. Software Design Document

## 1. System Design

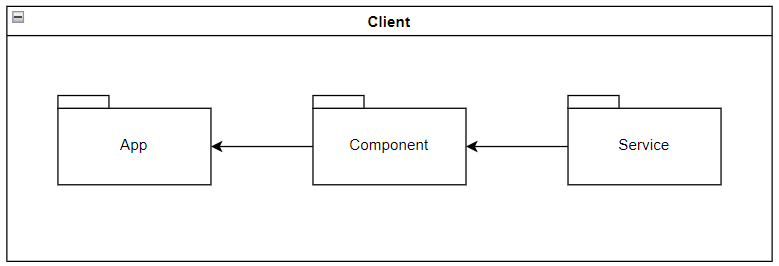
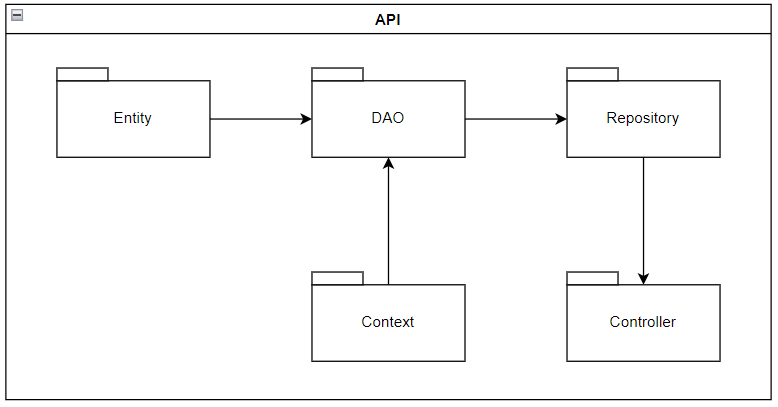
### 1.1 System Architecture

*[The content of this section includes the overall diagram which includes the sub-systems, the external systems, and the relationship/connection among them. You need also provide the explanation for each of the diagram components (modules, sub-systems, external systems, etc.)].*



### 1.2 Package Diagram

*[Provide the package diagram for each sub-system. The content of this section includes overall package diagram(s) and the explanation for each package (or namespace). Please see the sample and description table format below]*

***Package Descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | Entity | Define the logic of data extracted from Database |
| 02 | DAO | Provide the connection to the Database; Extract data for the system |
| 03 | Repository | Create an abstraction layer between the data access layer and the business logic layer of an application |
| 04 | Context | Query from a database and group together changes that will then be written back to the store as a unit |
| 05 | Controller | The controller takes the result of the model's processing (if any) and returns either the proper view and its associated view data or the result of the API call |
| 06 | App | Root component |
| 07 | Component | Components are the main building block for Angular applications. Each component consists of: An HTML template that declares what renders on the page. A TypeScript class that defines behavior |
| 08 | Service | Define code or functionalities that are then accessible and reusable in many other components in Angular project |

## 2. Database Design

*[Provide the files description, database table relationship & table descriptions like example below]*



***Table Descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *<Table name>* | *<Description of the table>*  *- Primary keys: <<list of primary key fields>>*  *- Foreign keys: <<list of foreign key fields>>* |
| *02* | *<Table name2>* | *…* |

## 3. Detailed Design

### 3.1 <Feature/Function Name1>

*[Provide the detailed design for the feature <Feature Name1>. It includes Class Diagram, Class Specifications, and Sequence Diagram(s);* ***For the features/functions with the same structure of class & sequence diagrams, you need to provide the diagrams once for one feature/function and refer to those diagrams from other features/functions****]*

#### 3.1.1 Class Diagram

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



***3.1.2 <Sequence Diagram Name1>***

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



#### 3.1.3 <Sequence Diagram Name2>

#### 3.1.4 …

### 3.2 <Feature/Function Name2>

…