

**Capstone Project Report**

**Report 4 – Software Design Document**

– Hanoi, Sep 2022 –

**Table of Contents**

[I. Record of Changes 3](#_Toc116588380)

[II. Software Design Document 4](#_Toc116588381)

[1. System Design 4](#_Toc116588382)

[1.1 System Architecture 4](#_Toc116588383)

[1.2 Package Diagram 4](#_Toc116588384)

[2. Database Design 6](#_Toc116588385)

[3. Detailed Design 6](#_Toc116588386)

[3.1 <Feature/Function Name1> 6](#_Toc116588387)

[3.2 <Feature/Function Name2> 8](#_Toc116588388)

[4. Database Tables 8](#_Toc116588389)

[1. Campuses 8](#_Toc116588390)

[2. UserRoles 8](#_Toc116588391)

[3. Addresses 8](#_Toc116588392)

[4. Users 8](#_Toc116588393)

[5. Houses 9](#_Toc116588394)

[6. Rooms 9](#_Toc116588395)

[7. Rates 10](#_Toc116588396)

[8. ImagesOfHouse 10](#_Toc116588397)

[9. ImagesOfRoom 11](#_Toc116588398)

[10. Districts 11](#_Toc116588399)

[11. Communes 11](#_Toc116588400)

[12. Villages 11](#_Toc116588401)

[13. Statuses 11](#_Toc116588402)

[14. RoomTypes 12](#_Toc116588403)

[15. Reports 12](#_Toc116588404)

[16. RoomHistories 12](#_Toc116588405)

# I. Record of Changes

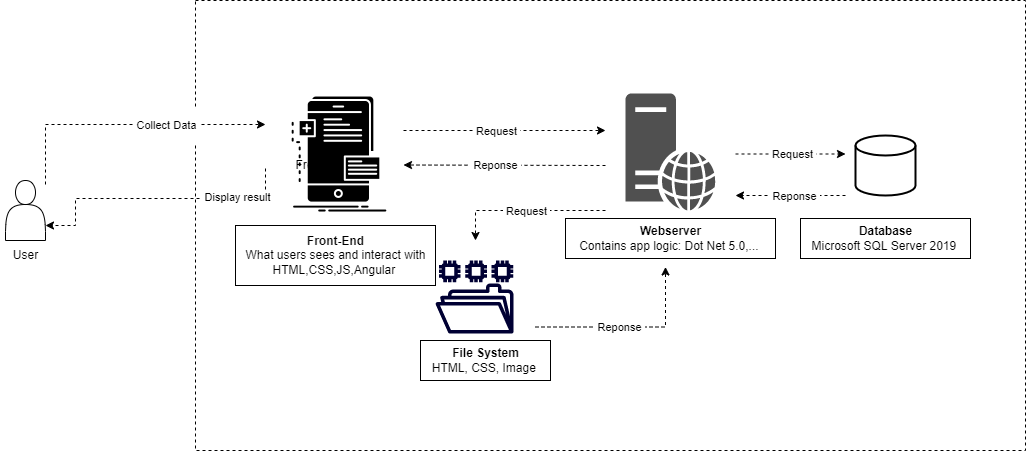
|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 08/10/2022 | A | KienNT | Database Tables |
| 09/10/2022 | M | GiangNT | Database Tables |
| 09/10/2022 | A | ThongPQ | System Architecture |
| 12/10/2022 | A | KienNT | Package Diagram, Package Description |
| 13/10/2022 | M | GiangNT | Package Diagram, Package Description |
| 17/10/2022 | M | ThongPQ | System Architecture |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Software Design Document

## 1. System Design

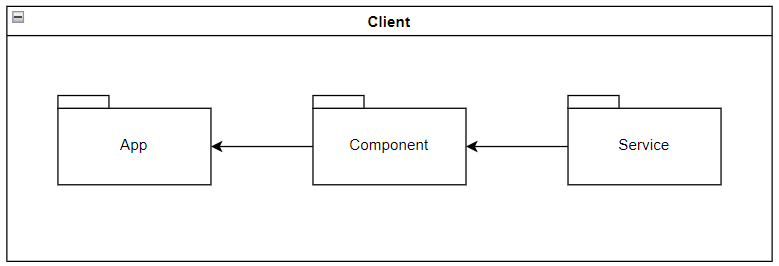
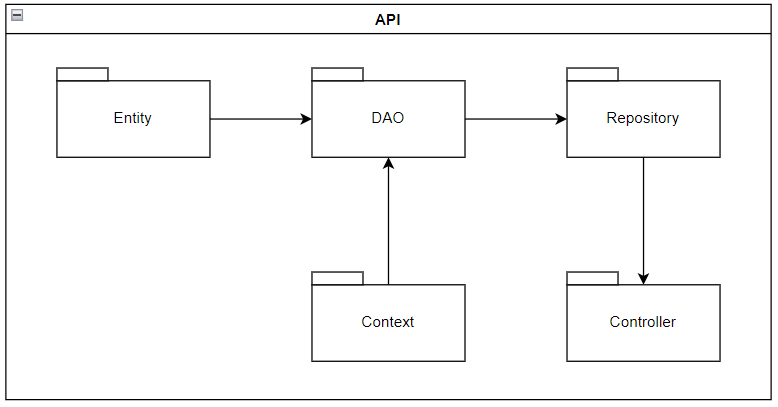
### 1.1 System Architecture



|  |  |  |
| --- | --- | --- |
| **No** | **Component** | **Description** |
| 1 | User | As a user, impact on the site |
| 2 | Front-End | What users sees and interact with HTML, CSS, JS, Angular |
| 3 | Webserver | Contains app logic: Dot Net 5.0, ... |
| 4 | File System | Contains HTML, CSS, Image in client. |
| 5 | Database | Microsoft SQL Server 2019, this contains all data of website. |

### 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>

…

## 4. Database Tables

### 1. Campuses

*Store the information of all campus for users*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | CampusId | int |  |  | Yes | PK |  |
| 2 | CampusName | nvarchar | 100 |  |  |  |  |
| 3 | AddressId | int |  |  | Yes |  | To Table *Addresses* |
| 4 | CreatedDate | datetime |  |  |  |  |  |

### 2. UserRoles

*Store the information of all roles of users*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | RoleId | int |  |  | Yes | PK |  |
| 2 | RoleName | nvarchar | 100 |  |  |  |  |
| 3 | CreatedDate | datetime |  |  |  |  |  |

### 3. Addresses

*Store the information of addresses for Users (Landlords) and Houses*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | AddressId | int |  |  | Yes | PK |  |
| 2 | Addresses | nvarchar | 1000 |  | Yes |  |  |
| 3 | GoogleMapLocation | nvarchar | MAX |  |  |  | Location on Google Map |
| 3 | CreatedDate | datetime |  |  |  |  |  |
| 4 | LastModifiedDate | datetime |  |  |  |  |  |

### 4. Users

*Store the information of all users*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | UserId | nchar | 30 |  | Yes | PK | Include student id, host id and admission id;  Has specific convention to make |
| 2 | FacebookUserId | nchar | 300 |  |  |  | For people login with Facebook |
| 3 | GoogleUserId | nchar | 300 |  |  |  | For people login with Google |
| 4 | Email | nvarchar | 100 |  |  |  |  |
| 5 | Password | nvarchar | 100 |  |  |  |  |
| 6 | DisplayName | nvarchar | 500 |  |  |  | For Landlord and Staff |
| 7 | Active | bit |  |  |  |  |  |
| 9 | ProfileImageLink | nvarchar | 500 |  | No |  | For Landlord and Staff |
| 10 | PhoneNumber | nvarchar | 50 |  | No |  | Only for Landlord |
| 11 | FacebookURL | nvarchar | 300 |  | No |  | Only for Landlord |
| 12 | IdentityCardFrontSideImageLink | nvarchar | 500 |  | No |  | Only for Landlord |
| 13 | IdentityCardBackSideImageLink | nvarchar | 500 |  | No |  | Only for Landlord |
| 14 | AddressId | int |  |  | No |  | Only for Landlord;  To Table *Addresses* |
| 15 | RoleId | int |  |  |  | FK | To Table *Roles* |
| 16 | CreatedDate | datetime |  |  |  |  |  |
| 17 | LastModifiedDate | datetime |  |  |  |  |  |
| 18 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 19 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 5. Houses

*Store the information of all houses*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | HouseId | int |  |  | Yes | PK |  |
| 2 | HouseName | nvarchar | 100 |  |  |  |  |
| 3 | Information | nvarchar | MAX |  |  |  |  |
| 4 | AddressId | int |  |  | Yes |  | To Table *Addresses* |
| 5 | VillageId | int |  |  |  | FK | To Table *Villages* |
| 6 | LandlordId | nchar | 30 |  |  | FK | To Table *Users* |
| 7 | PowerPrice | money |  |  | Yes |  |  |
| 8 | WaterPrice | money |  |  | Yes |  |  |
| 9 | CreatedDate | datetime |  |  |  |  |  |
| 10 | LastModifiedDate | datetime |  |  |  |  |  |
| 11 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 12 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 6. Rooms

*Store the information of all rooms of houses*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | RoomId | int |  |  | Yes | PK |  |
| 2 | RoomName | nvarchar | 50 |  |  |  |  |
| 3 | PricePerMonth | money |  |  |  |  |  |
| 6 | Information | nvarchar | MAX |  |  |  |  |
| 7 | AreaByMeters | float |  |  |  |  |  |
| 8 | AirCon | bit |  |  | Yes |  | Air Conditioner |
| 9 | MaxAmountOfPeople | int |  |  |  |  |  |
| 10 | CurrentAmountOfPeople | int |  |  |  |  |  |
| 11 | BuildingNumber | int |  |  |  |  |  |
| 12 | FloorNumber | int |  |  |  |  |  |
| 13 | StatusId | int |  |  |  | FK | To Table *Statuses* |
| 14 | RoomTypeId | int |  |  |  | FK | To Table *RoomTypes* |
| 15 | HouseId | int |  |  |  | FK | To Table *Houses* |
| 16 | CampusId | int |  |  |  | FK | To Table *Campuses* |
| 17 | CreatedDate | datetime |  |  |  |  |  |
| 18 | LastModifiedDate | datetime |  |  |  |  |  |
| 19 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 20 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 7. Rates

*Store the information of all rates and comments of users*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | RateId | int |  |  | Yes | PK |  |
| 2 | Star | int |  |  |  |  |  |
| 3 | Comment | nvarchar | MAX |  |  |  |  |
| 4 | LandlordReply | nvarchar | MAX |  |  |  |  |
| 5 | HouseId | int |  |  |  | FK | To Table *Houses* |
| 6 | StudentId | nchar | 30 |  |  | FK | To Table *Users* |
| 7 | CreatedDate | datetime |  |  |  |  |  |
| 8 | LastModifiedDate | datetime |  |  |  |  |  |
| 9 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 10 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 8. ImagesOfHouse

*Store the information of all pictures of house*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ImageId | int |  |  | Yes | PK |  |
| 2 | ImageLink | nvarchar | 500 |  |  |  |  |
| 3 | HouseId | int |  |  |  | FK | To Table *Houses* |
| 4 | CreatedDate | datetime |  |  |  |  |  |
| 5 | LastModifiedDate | datetime |  |  |  |  |  |
| 6 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 7 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 9. ImagesOfRoom

*Store the information of all pictures of room*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  |  | Yes | PK |  |
| 2 | ImageLink | nvarchar | 500 |  |  |  |  |
| 3 | RoomId | int |  |  |  | FK | To Table *Rooms* |
| 4 | CreatedDate | datetime |  |  |  |  |  |
| 5 | LastModifiedDate | datetime |  |  |  |  |  |
| 6 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 7 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 10. Districts

*Store the information of all districts (for identifying house address)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | DistrictId | int |  |  | Yes | PK |  |
| 2 | DistrictName | nvarchar | 100 |  |  |  |  |
| 3 | CreatedDate | datetime |  |  |  |  |  |

### 11. Communes

*Store the information of all communes (for identifying house address)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | CommuneId | int |  |  | Yes | PK |  |
| 2 | CommunetName | nvarchar | 100 |  |  |  |  |
| 3 | DistrictId | int |  |  |  | FK | To Table *Districts* |
| 4 | CreatedDate | datetime |  |  |  |  |  |

### 12. Villages

*Store the information of all villages (for identifying house address)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | VillageId | int |  |  | Yes | PK |  |
| 2 | VillageName | nvarchar | 100 |  |  |  |  |
| 3 | CommuneId | int |  |  |  | FK | To Table *Communes* |
| 4 | CreatedDate | datetime |  |  |  |  |  |

### 13. Statuses

*Store the information of all status of a Room*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | StatusId | int |  |  | Yes | PK |  |
| 2 | StatusName | nvarchar | 300 |  |  |  |  |
| 3 | CreatedDate | datetime |  |  |  |  |  |

### 14. RoomTypes

*Store the information of all types of a Room*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | RoomTypeId | int |  |  | Yes | PK |  |
| 2 | RoomTypeName | nvarchar | 300 |  |  |  |  |
| 3 | CreatedDate | datetime |  |  |  |  |  |

### 15. Reports

*Store the information of all Reports of Students to Houses*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ReportId | int |  |  | Yes | PK |  |
| 2 | ReportContent | nvarchar | MAX |  |  |  |  |
| 3 | StudentId | nchar | 30 |  |  | FK | To Table *Users* |
| 4 | HouseId | int |  |  |  | FK | To Table *Houses* |
| 5 | CreatedDate | datetime |  |  |  |  |  |
| 6 | LastModifiedDate | datetime |  |  |  |  |  |
| 7 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 8 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |

### 16. RoomHistories

*Store the information of all records of Room usage for Landlords to manage if they want to*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | RoomHistoryId | int |  |  | Yes | PK |  |
| 2 | CustomerName | nvarchar | 800 |  |  |  | Name of the person stayed in the Room |
| 3 | RoomId | int |  |  |  | FK | To Table *Rooms* |
| 4 | CreatedDate | datetime |  |  |  |  |  |
| 5 | LastModifiedDate | datetime |  |  |  |  |  |
| 6 | CreatedBy | nchar | 30 |  |  | FK | To Table *Users* |
| 7 | LastModifiedBy | nchar | 30 |  |  | FK | To Table *Users* |