|  |
| --- |
| **E:\My Documents\Desktop\Logo_FPT_University_doc.jpgMinistry of education and training** |
| Software Design Description |
| Place Sharing Network |
|  |
| |  |  | | --- | --- | | **PSN Team** | | | **Group Members** |  | | **Supervisor** |  | | **Ext Supervisor** |  | | **Capstone Project code** |  | |
| **Hanoi, 17th Apr, 2011** |

**Record of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Change Item** | **Description** | **By** | **Version** |
| 18/2/2011 | All | Create the document | DungTQ | 0.1 |
| 19/2/2011 | Detail Design  (1 to 10, 18 to 20) | Edit content | DungTQ | 0.2 |
| 20/2/2011 | Detail Design  (11 to 17) | Edit content | DungTQ | 0.3 |
| 21/2/2011 | - Design Overview  - System Architectural Design  - References | Create | DungTQ | 0.4 |
| 23/2/2011 | Database Design | Create | QuocNM | 0.5 |
| 24/2/2011 | - UI Design  - References | - Remove content  - Edit content | DungTQ | 0.6 |
| 24/2/2011 | Database design | Some description | QuangVT | 0.7 |
| 12/4/2011 | Change design | Change all cases | DungTQ | 0.8 |
| 13/4/2011 | Change database design | Add new and edit tables in Database design | DungTQ | 0.9 |
| 17/4/2011 | All | Add detail designs for new use cases | DungTQ | 1.0 |

# Table of Contents

[Table of Contents 2](#_Toc353572058)

[I. Design Overview 5](#_Toc353572059)

[II. System Architectural Design 5](#_Toc353572060)

[1. Overall System Architecture 5](#_Toc353572061)

[2. Web Application Architecture 6](#_Toc353572062)

[2.1 Psn.Common Component 7](#_Toc353572063)

[2.2 Psn.Resources Component 7](#_Toc353572064)

[2.3 Psn.Business Component 7](#_Toc353572065)

[2.4 Psn.Web Component 8](#_Toc353572066)

[III. Detailed Design 9](#_Toc353572067)

[A. Class Diagram 9](#_Toc353572068)

[1. Post 9](#_Toc353572069)

[1.1 Class Diagram 10](#_Toc353572070)

[1.2 Class Explanations 10](#_Toc353572071)

[2. Search Post 11](#_Toc353572072)

[2.1 Class Diagram 12](#_Toc353572073)

[2.2 Class Explanations 12](#_Toc353572074)

[3. LogOn\_Register\_LogOut 13](#_Toc353572075)

[4.1 Class Diagram 13](#_Toc353572076)

[4.2 Class Explanations 13](#_Toc353572077)

[5. ChangePassWord 14](#_Toc353572078)

[5.1 Class Diagram 14](#_Toc353572079)

[5.2 Class Explanations 15](#_Toc353572080)

[6. User Control Panel 15](#_Toc353572081)

[6.1 Class Diagram 16](#_Toc353572082)

[6.2 Class Explanations 16](#_Toc353572083)

[7. Manage Categories 17](#_Toc353572084)

[7.1 Class Diagram 17](#_Toc353572085)

[7.2 Class Explanations 17](#_Toc353572086)

[6. Admin Manage Bad Post 18](#_Toc353572087)

[8.1 Class Diagram 19](#_Toc353572088)

[8.2 Class Explanations 19](#_Toc353572089)

[9. Admin Manage Bad Words 20](#_Toc353572090)

[9.1 Class Diagram 20](#_Toc353572091)

[9.2 Class Explanations 20](#_Toc353572092)

[10. Admin Manage Bad Word Type 21](#_Toc353572093)

[10.1 Class Diagram 21](#_Toc353572094)

[10.2 Class Explanations 21](#_Toc353572095)

[11. Admin Manage District 22](#_Toc353572096)

[11.1 Class Diagram 22](#_Toc353572097)

[11.2 Class Explanations 23](#_Toc353572098)

[12. Admin Manage Post Status 23](#_Toc353572099)

[12.1 Class Diagram 24](#_Toc353572100)

[12.2 Class Explanations 24](#_Toc353572101)

[13. Admin Manage Provinces 25](#_Toc353572102)

[13.1 Class Diagram 25](#_Toc353572103)

[13.2 Class Explanations 25](#_Toc353572104)

[14 Admin Manage Roles 26](#_Toc353572105)

[14.1 Class Diagram 26](#_Toc353572106)

[14.2 Class Explanations 26](#_Toc353572107)

[15 Admin Manage Payments 27](#_Toc353572108)

[15.1 Class Diagram 27](#_Toc353572109)

[15.2 Class Explanations 27](#_Toc353572110)

[16 Admin Manage Report Post Status 28](#_Toc353572111)

[16.1 Class Diagram 28](#_Toc353572112)

[16.2 Class Explanations 28](#_Toc353572113)

[B. Sequence Diagram 29](#_Toc353572114)

[1. Search 29](#_Toc353572115)

[2. Advanced Search 29](#_Toc353572116)

[3. Delete Post 30](#_Toc353572117)

[4. View Post 30](#_Toc353572118)

[5. Add Post 30](#_Toc353572119)

[6. Edit Post 31](#_Toc353572120)

[7. Register 31](#_Toc353572121)

[8. Login 32](#_Toc353572122)

[9. View User Profile 32](#_Toc353572123)

[10. Edit User Profile 33](#_Toc353572124)

[11. Manage Post 33](#_Toc353572125)

[12. Manage Favorites 33](#_Toc353572126)

[13. Add Post to Favorite 34](#_Toc353572127)

[14. Remove Post from Favorite 34](#_Toc353572128)

[15. Add new Categories 35](#_Toc353572129)

[IV. Database Design 35](#_Toc353572130)

[1. ERDs 35](#_Toc353572131)

[**2.** **Tables** 35](#_Toc353572132)

[V. References 42](#_Toc353572133)

# Design Overview

This document describes the technical and UI design of the PSN system. It includes the architectural design and the detailed design of common functions and business functions. It also includes the design of database model, and overall design of user interface.

The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem. It will describe the patterns being used, the role of each component and the role of the system in the working environment.

The detailed design describes static and dynamic structure for each component and function. It includes class diagrams, class explanations, and sequence diagrams of the main use cases. The detailed design uses notations of .NET framework 4.0 and C# 4.0 as they are the framework and language for developing the system.

The database design describes the relationship between entities, and details of each entity. It uses notations of SQL Server 2008 as it is the database server for developing the system.

The user interface design describes the layout of the system, and some design for the screens.

# System Architectural Design

## Overall System Architecture



The system will use the Client-Server and SOA architectural patterns as the main architecture. It makes use of the Internet and common web protocols to deliver its services to the users.

The main entities of the above diagram will be described as following:

* **Database store:** this is the central data storage for the system. It may be running on a separate server to the Application Server, or in the same machine.
* **Application Server:** this is the server of the system running on IIS web server. It includes two main components: Web Application and Web Services. Web Application is responsible for creating web pages. It is developed using **ASP.NET MVC 3** framework. Web Services provides the system’s API to the world and some parts of the application. The services are consumed by some mobile applications. It is noted that this component will not be included in this project’s scope. However the system architecture must be extensible for adding such a component in the future.
* **Service Providers:** these are third party service providers that the application may need services from. For example the application may need Google Map services from Google, or some API from Facebook and OpenID…
* **Clients:** these include internet browsers running on personal computers, or mobile applications targeting mobile devices. The web browsers will interact with the web application via normal protocols (HTTP, HTTPS…). The mobile applications use SOA pattern to interact with the system via web services. However it is also noted that they are not included in this project’s scope.

## Web Application Architecture

The architecture for the PSN web application will be depicted as the following diagram:



### Psn.Common Component

This component contains common logic that can be used throughout the application. It includes for example mechanism for logging, caching, cryptography, and other utilities.



### Psn.Resources Component

This component contains static resources for the system. It includes localized strings and messages.

### Psn.Business Component

This component contains all business logic for the system. It includes ORM classes, business classes, data access layer, business interface…

The aim and challenge for the design is that this component will be application type independent. That is this component is not tied only to web application but can be used for other .NET application types if necessary.

All the services of this component will be used by dependent components via its interfaces.



### Psn.Web Component

This component is responsible for delivering the front end to the users and accepting their inputs. It will be developed using ASP.NET MVC 3 framework. An MVC Application is designed and implemented using the following three attributes:



* **Model:** The model contains the core information for an application. This includes the data and validation rules as well as data access and aggregation logic.
* **View:** The view encapsulates the presentation of the application, and in ASP.NET this is typically the HTML markup.
* **Controller:** The controller contains the control-flow logic. It interacts with the Model and Views to control the flow of information and execution of the application.

The **PSN** application will make use another attribute: **View Model**. The view model encapsulates all the data properties that the view needs or provides. It is the view model that the view depends directly on, not the model. The model will be implemented by **Psn.Business** components, except those not related to business logics.



The overall architecture of Psn.Web component will be depicted as the following diagram:



* **Helpers:** These are utilities classes specific for this component
* **Framework:** Contain the core framework to build web application

# Detailed Design

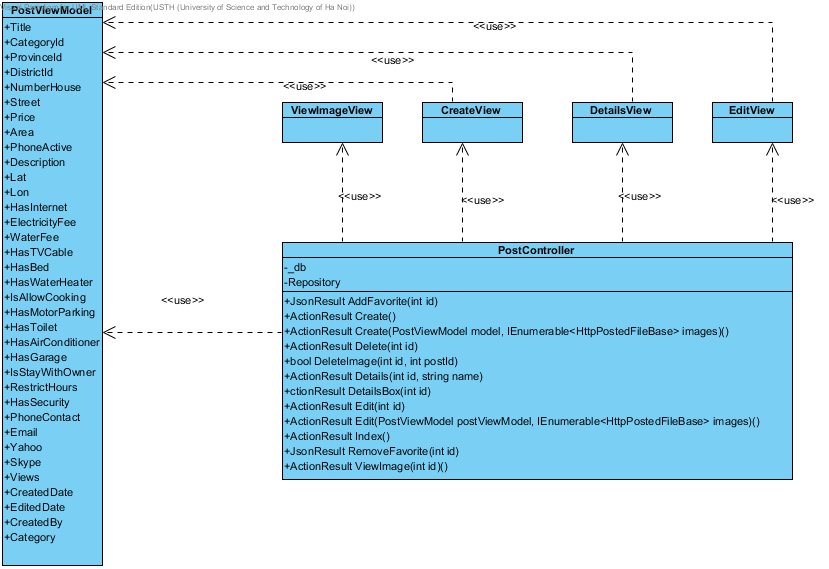
# Class Diagram

## Post

This part is the detailed design for the following use cases:

* Add Post
* Edit Post
* Delete Post
* View detail of post
* Add post to favorite
* Delete post from favorite

### Class Diagram



### Class Explanations

#### PostController class

##### **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| \_db | RentalHouseFindingEntities | Private |  |

##### **Operations**

###### ActionResult AddFavorite(int id)

Add post to favotite

###### ActionResult Create()

Render Create post screen

###### ActionResult Create(PostViewModel model, IEnumerable<HttpPostedFileBase> images)

Create a post

###### ActionResult Delete(int id)

Delete a post

###### ActionResult DeleteImage(int id, int postId)

Delete image of post

###### ActionResult Details(int id, string name)

Display details page

###### ActionResult DetailsBox(int id)

Display details popup

###### ActionResult Edit(int id)

Render Edit post page

###### ActionResult Edit(int id, IEnumerable<HttpPostedFileBase> images)

Edit post page

###### ActionResult RemoveFavorite(int id)

Delete favotite post

#### DetailsView class

Contain HTML markup and .NET code to render View screen

#### CreateView class

Contain HTML markup and .NET code to render View screen

#### ViewImageView class

Contain HTML markup and .NET code to render View screen

#### EditView class

Contain HTML markup and .NET code to render View screen

#### PostViewModel class

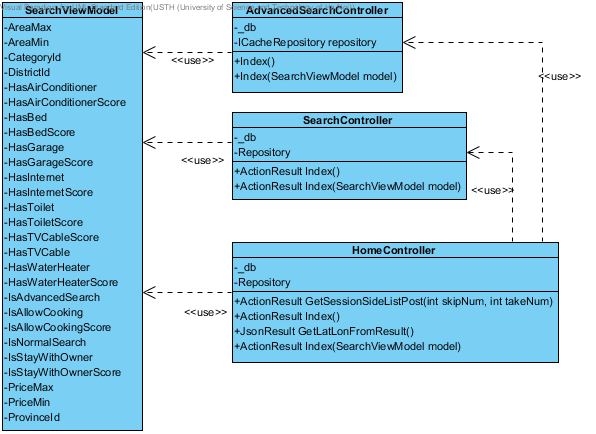
Model type of CreateView, DetailsView and EditView

## Search Post

This part is the detailed design for the following use cases:

* Search
* Advance Search

### Class Diagram



### Class Explanations

#### SearchController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |
| Repository | ICacheRepository |  |

##### **Operations**

###### ActionResult Index()

Render Search page

###### ActionResult Index(SearchViewModel)

Search

#### HomeController class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |
| Repository | ICacheRepository |  |

##### **Operations**

*JsonResult GetLatLonFromResult()*

Get Latitude and Longitude of Result.

*ActionResult GetSectionSideListPost(int skipNum, int takeNum)*

Get list post result

*ActionResult Index()*

Display result page.

###### ActionResult Index(SearchViewModel model)

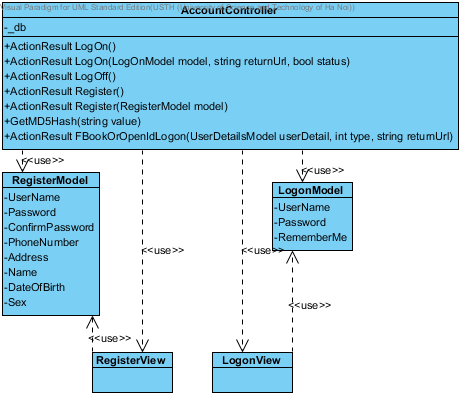
Search

## LogOn\_Register\_LogOut

This part is the detailed design for the following use cases:

* Login
* Register
* Logout

### Class Diagram



### Class Explanations

#### AccountController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult LogOn()

Render Login page

###### ActionResult LogOn(LogOnModel model, string returnUrl, bool status)

Login.

###### ActionResult Register()

Render Register page

###### ActionResult Register(RegisterModel model)

Register new account.

###### ActionResult FaceBookOrOpendIdLogOn(UserDetailModel model, int type, string returnUrl)

Login by using Facebook or OpenId

###### ActionResult LogOff()

Log out.

#### LogOnView class

Contain HTML markup and .NET code to render LogOn screen

#### RegisterView class

Contain HTML markup and .NET code to render Register screen

#### LogOnModel class

Model type of LogOnView

#### RegisterModel class

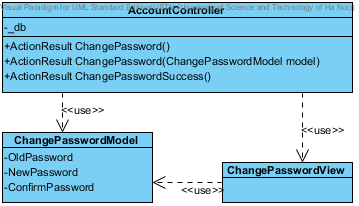
Model type of RegisterView

## Change PassWord

This part is the detailed design for the following use cases:

* ChangePassword

### Class Diagram



### Class Explanations

#### AccountController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult ChangePassword()

Render change password page

###### ActionResult ChangePassword(ChangePasswordModel model)

Change pasword

#### ChangePasswordView class

Contain HTML markup and .NET code to render Change password screen

#### ChangePasswordViewModel class

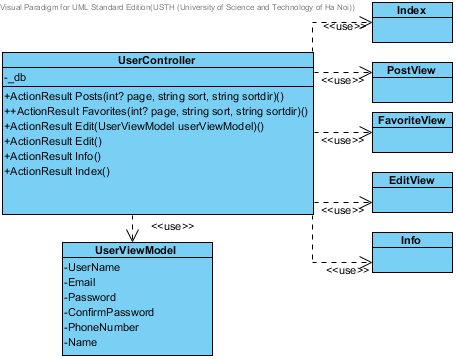
Model type of ChangePasswordView

## 6. User Control Panel

This part is the detailed design for the following use cases:

* Manage post
* Manage payment
* Manage favorite
* View User Info
* Edit User Info

### Class Diagram



### Class Explanations

#### UserController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Edit()

Render Edit User Info page

###### ActionResult Edit(UserViewModel userViewModel)

Edit User Info

###### ActionResult Posts(int? page, string sort, string sortdir)

Render User Manage Post

###### ActionResult Favorite(int? page, string sort, string sortdir)

Render User Manage Favorite

###### ActionResult Payments(int? page, string sort, string sortdir)

Render User Manage Post

###### ActionResult Index()

Render User Control Panel page

###### ActionResult Info()

Render User Manage Post

#### PostView class

Contain HTML markup and .NET code to render Post list screen

#### EditView class

Contain HTML markup and .NET code to render Edit User info screen

#### InfoView class

Contain HTML markup and .NET code to render Info screen

#### FavoriteView class

Contain HTML markup and .NET code to render Favorite List screen

#### InxdexView class

Contain HTML markup and .NET code to render User Control Panel screen

#### UserViewModel class

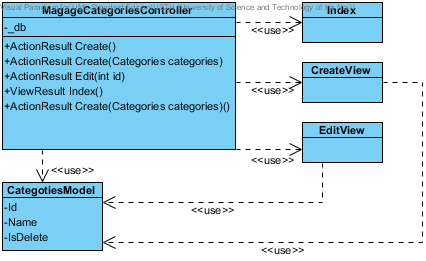
Model type of EditView and InfoView

## Manage Categories

This part is the detailed design for the following use cases:

* View Categories list
* Create a new category
* Edit category
* Delete category

### Class Diagram



### Class Explanations

#### ManageCategoriesController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create category page

###### ActionResult Create(Categories categories)

Insert a new category

###### ActionResult Edit(Categories categories)

Edit category

###### ActionResult Edit()

Render Edit page

#### IndexView class

Contain HTML markup and .NET code to render Manage Categories screen

#### CreateView class

Contain HTML markup and .NET code to render Insert a new Category screen

#### EditView class

Contain HTML markup and .NET code to render Edit Category screen

#### CategoryModel class

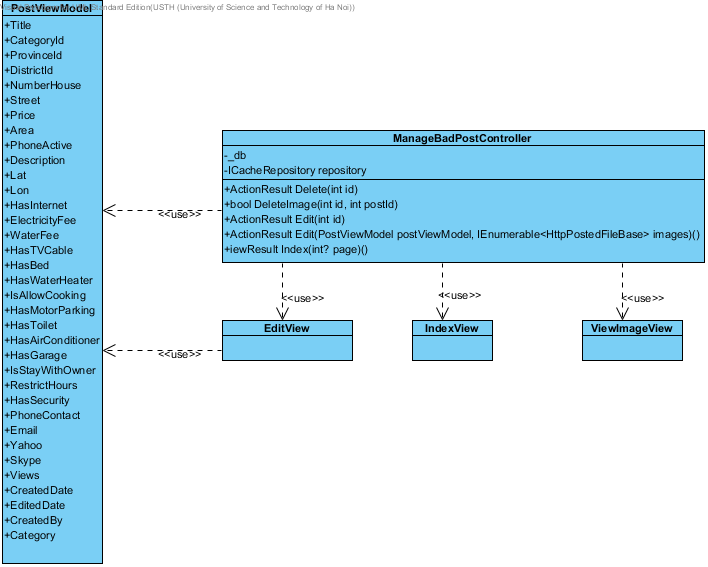
Model type of CreateView and EditView

## Admin Manage Bad Post

This part is the detailed design for the following use cases:

* Manage bad post

### Class Diagram



### Class Explanations

#### IPlaceRepository interface

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Delete(int id)

Delete bad post

###### ActionResult Edit(int id)

Get the reported details of place

###### ActionResult Index(int? page)

Display list bad post

###### ActionResult Edit(PostViewModel postViewModel, IEnumerable<HttpPostedFileBase> images)

Edit post and accept post

#### EditView class

Contain HTML markup and .NET code to render Edit bad post screen

#### Index

Contain HTML markup and .NET code to render List bad post screen

#### PostViewModel class

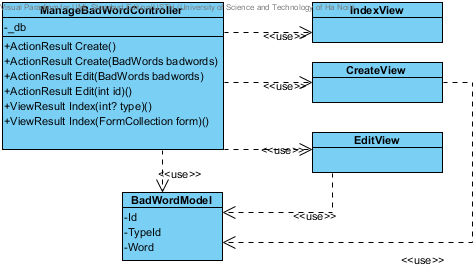
Model type of EditView

## Admin Manage Bad Words

This part is the detailed design for the following use cases:

* Manage bad words
* Add bad word
* Edit bad word
* Delete bad word

### Class Diagram



### Class Explanations

#### ManageBadWordsController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create page

###### ActionResult Create(BadWords badword)

Create new Bad Word

###### ActionResult Edit(int id)

Render Edit page

###### ActionResult Edit(BadWords badword)

Edit bad word

###### ActionResult Index()

Display list bad words

#### CreateView class

Contain HTML markup and .NET code to render Create new Bad word screen

#### EditView class

Contain HTML markup and .NET code to render Edit Bad word screen

#### Index class

Contain HTML markup and .NET code to render List Bad words screen

#### BadWordsModel class

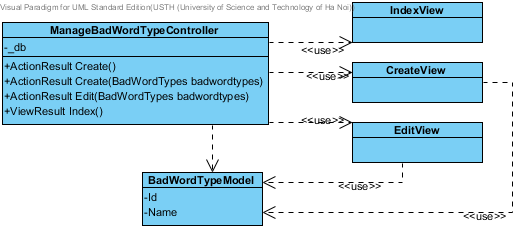
Model type of CreateView and EditView

## Admin Manage Bad Word Type

This part is the detailed design for the following use cases:

* Manage bad word type
* Add bad word type
* Edit bad word type
* Delete bad word type

### Class Diagram



### Class Explanations

#### 10.2.1 ManageBadWordTypesController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create page

###### ActionResult Create(BadWordTypes badwordType)

Create new Bad Word Type

###### ActionResult Edit(int id)

Render Edit page

###### ActionResult Edit(BadWordTypes badwordType)

Edit bad word type

###### ActionResult Index()

Display list bad words type

#### CreateView class

Contain HTML markup and .NET code to render Create new Bad word type screen

#### EditView class

Contain HTML markup and .NET code to render Edit Bad word type screen

#### Index class

Contain HTML markup and .NET code to render List Bad words type screen

#### BadWordTypesModel class

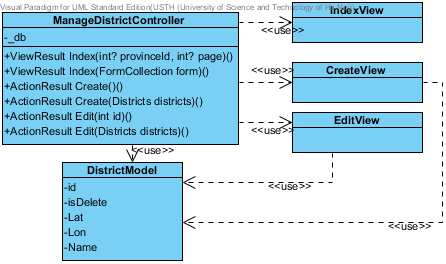
Model type of CreateView and EditView

## Admin Manage District

This part is the detailed design for the following use cases:

* Manage District
* Add District
* Edit District
* Delete District

### Class Diagram



### Class Explanations

#### 11.2.1 ManageDistictController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create page

###### ActionResult Create(Districts district)

Create new district

###### ActionResult Edit(int id)

Render Edit page

###### ActionResult Edit(Districts district)

Edit District

###### ActionResult Index()

Display list districts

#### CreateView class

Contain HTML markup and .NET code to render Create new District screen

#### 11.2.3 EditView class

Contain HTML markup and .NET code to render Edit District screen

#### Index class

Contain HTML markup and .NET code to render List Districts screen

#### DistrictsModel class

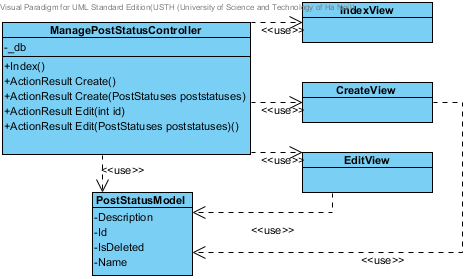
Model type of CreateView and EditView

## Admin Manage Post Status

This part is the detailed design for the following use cases:

* Manage Post Status
* Add Status
* Edit Status
* Delete Status

### Class Diagram



### Class Explanations

#### 12.2.1 ManagePostStatusController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create page

###### ActionResult Create(PostStatuses status)

Create new status

###### ActionResult Edit(int id)

Render Edit page

###### ActionResult Edit(PostStatuses status)

Edit status

###### ActionResult Index()

Display list status

#### 12.2.2 CreateView class

Contain HTML markup and .NET code to render Create new status screen

#### 12.2.3 EditView class

Contain HTML markup and .NET code to render Edit status screen

#### 12.2.4 Index class

Contain HTML markup and .NET code to render List status screen

#### PostStatusesModel class

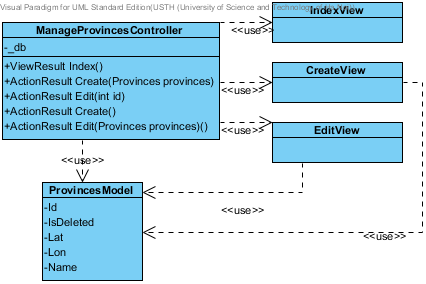
Model type of CreateView and EditView

## Admin Manage Provinces

This part is the detailed design for the following use cases:

* Manage Provinces
* Add Province
* Edit Province
* Delete Province

### Class Diagram



### Class Explanations

#### 13.2.1 ManageProvincesController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create page

###### ActionResult Create(Provinces province)

Create new province

###### ActionResult Edit(int id)

Render Edit province page

###### ActionResult Edit(Provinces province)

Edit province

###### ActionResult Index()

Display list province

#### 13.2.2 CreateView class

Contain HTML markup and .NET code to render Create new province screen

#### 13.2.3 EditView class

Contain HTML markup and .NET code to render Edit province screen

#### 13.2.4 Index class

Contain HTML markup and .NET code to render List province screen

#### ProvincesModel class

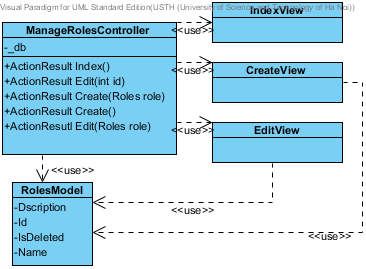
Model type of CreateView and EditView

## Admin Manage Roles

This part is the detailed design for the following use cases:

* Manage Roles
* Add role
* Edit role
* Delete role

### Class Diagram



### Class Explanations

#### 14.2.1 ManageRolesController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Create()

Render Create page

###### ActionResult Create(PostStatuses status)

Create new role

###### ActionResult Edit(int id)

Render Edit page

###### ActionResult Edit(PostStatuses status)

Edit role

###### ActionResult Index()

Display list role

#### 14.2.2 CreateView class

Contain HTML markup and .NET code to render Create new role screen

#### 14.2.3 EditView class

Contain HTML markup and .NET code to render Edit role screen

#### 14.2.4 Index class

Contain HTML markup and .NET code to render List role screen

#### RolesModel class

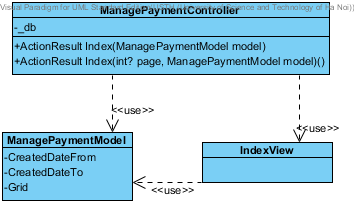
Model type of CreateView and EditView

## Admin Manage Payments

This part is the detailed design for the following use cases:

* Manage Payment

### Class Diagram



### Class Explanations

#### 15.2.1 ManagePaymentController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Index(int? page, string sort, string sortdir, ManagePaymentModel model)

Display list payment

#### 15.2.2 Index class

Contain HTML markup and .NET code to render List payment screen

#### 15.2.3 ManagePaymenModel class

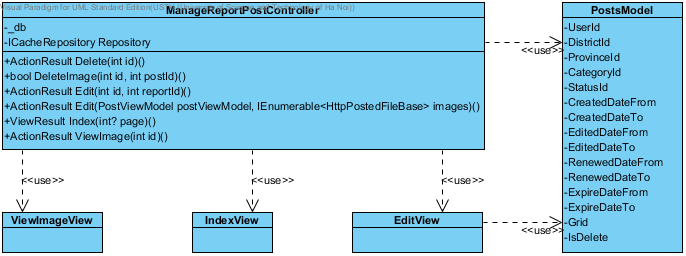
Model type of IndexView

## Admin Manage Report Post Status

This part is the detailed design for the following use cases:

* Manage Report Post

### Class Diagram



### Class Explanations

#### 16.2.1 ManageReportPostController Class

##### **Attributes**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| \_db | RentalHouseFindingEntities |  |

##### **Operations**

###### ActionResult Delete(int id)

Delete post

###### ActionResult DeleteImage(int id, int postId)

Create new status

###### ActionResult Edit(int id, int repostId)

Render Edit page

###### ActionResult Edit(PostViewModel postViewModel, IEnumerable<HttpPostedFileBase> images)

Edit Post

###### ActionResult Index()

Display list reported post

#### 16.2.2 ViewImageView class

Contain HTML markup and .NET code to render view image screen

#### 16.2.3 EditView class

Contain HTML markup and .NET code to render Edit post screen

#### 16.2.4 Index class

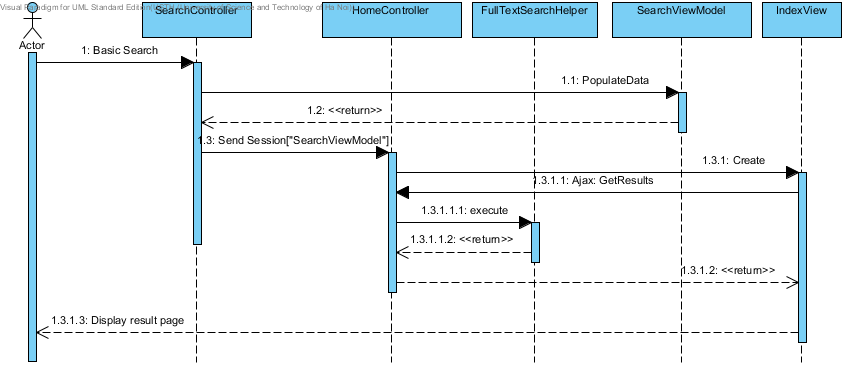
Contain HTML markup and .NET code to render List status screen

#### PostsModel class

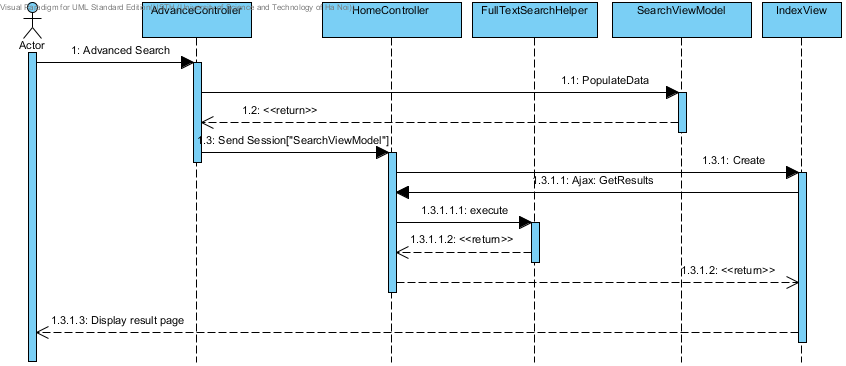
Model type of EditView

# Sequence Diagram

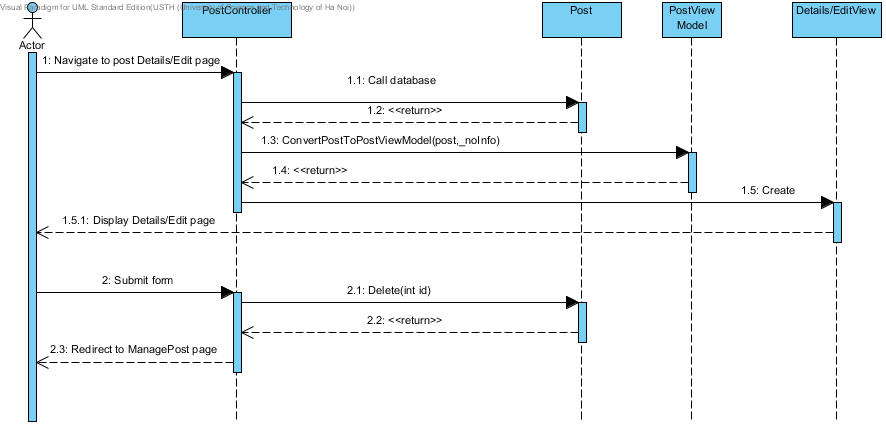
### Search



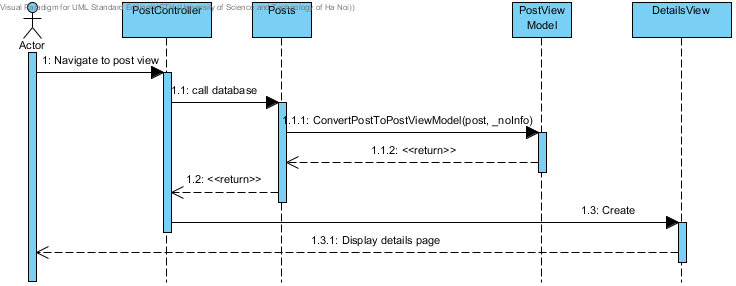
### Advanced Search



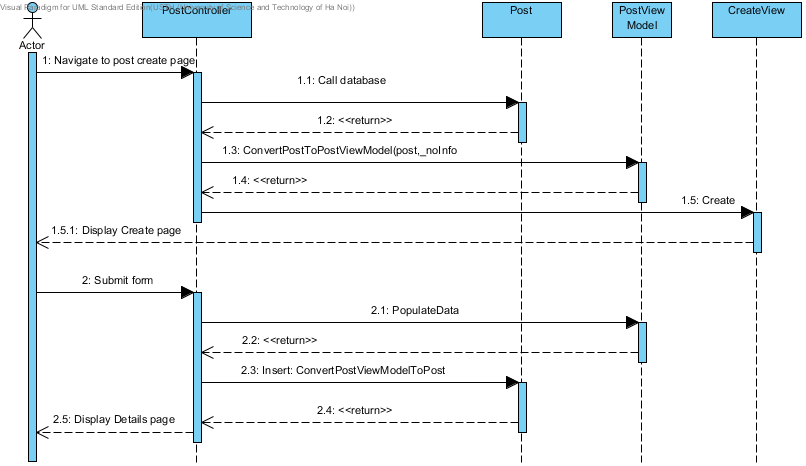
### Delete Post



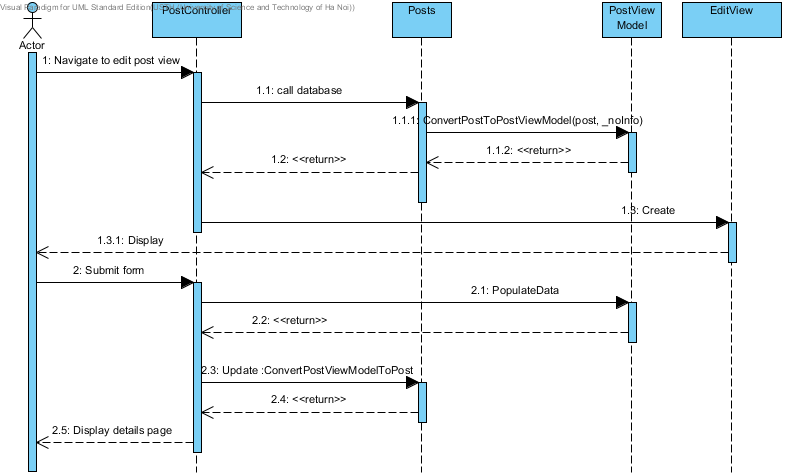
### View Post



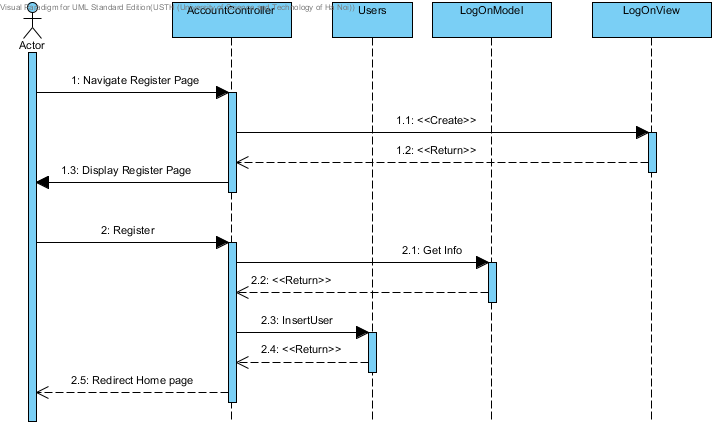
### Add Post



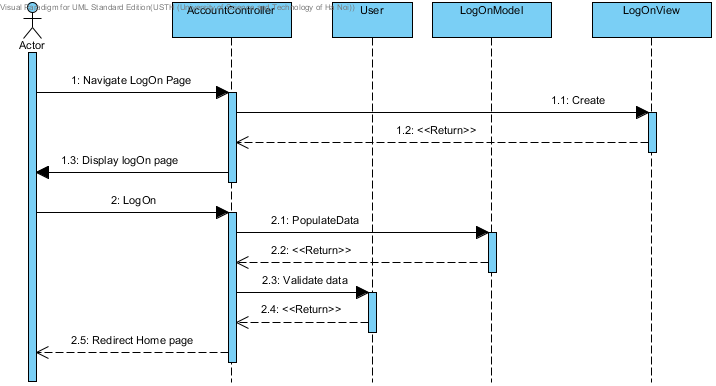
### Edit Post



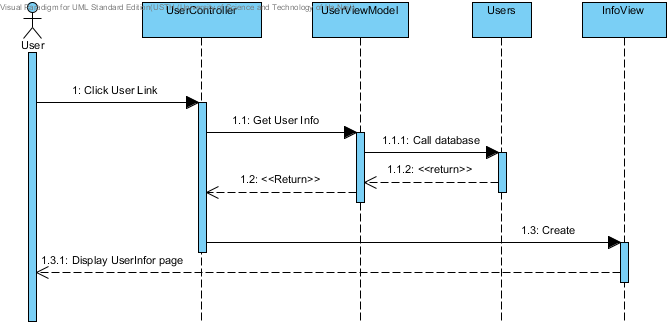
### Register



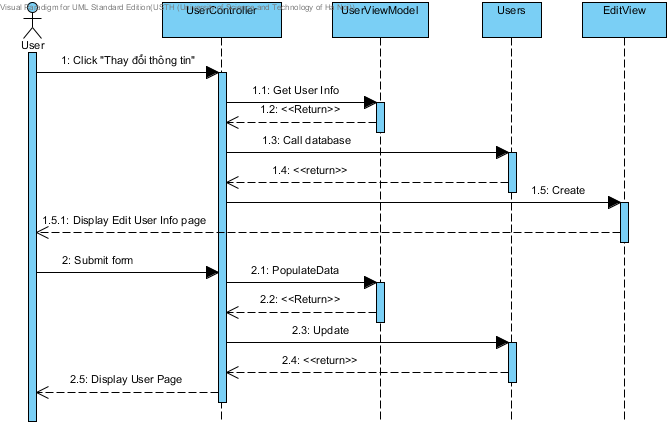
### Login



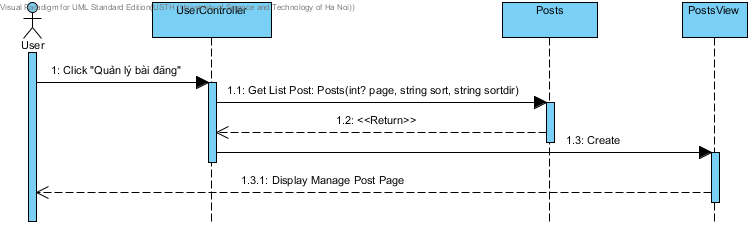
### View User Profile



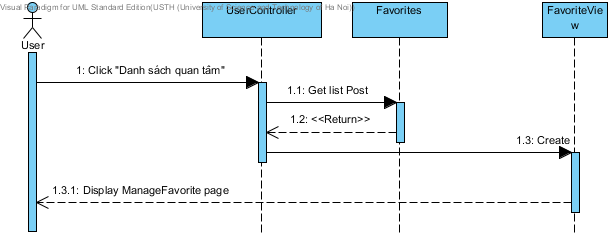
### Edit User Profile



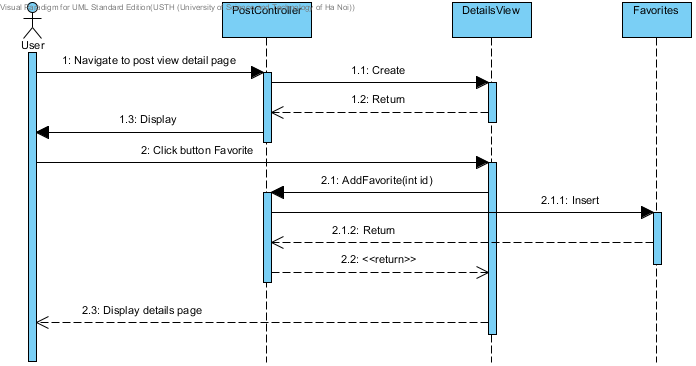
### Manage Post



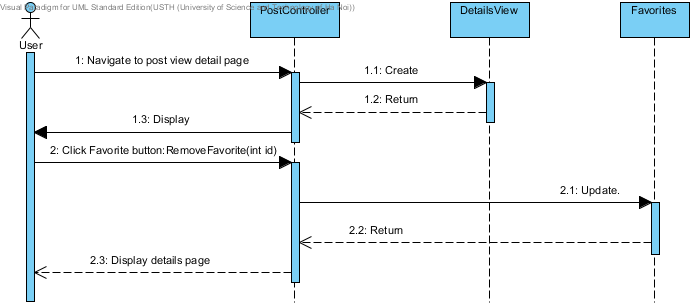
### Manage Favorites



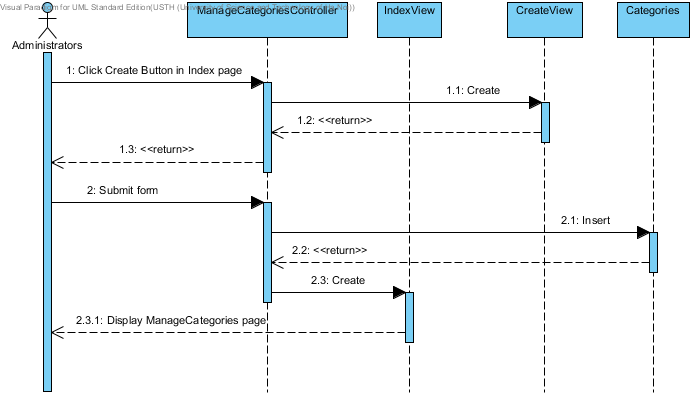
### Add Post to Favorite



### Remove Post from Favorite



### Add new Categories



# Database Design

## ERDs

Refer to attached ERD design

1. **Tables**

|  |  |
| --- | --- |
| Table Name | Description |
| AdvanceSearchScore | Store all score for advance search |
| Answer | Stores all answer |
| Badwords | Stores all badwords |
| BadWordTypes | Stores types of badwords |
| Categories | Store all categories |
| ConfigurationRHFs | Stores all configurations setting |
| Contacts | Stores all contacts information of post |
| Districts | Stores all districts |
| EmailTemplates | Stores all email templates |
| Facilities | Stores facilities information of post |
| FacilityTemplates | Store facilities template for advanced search algorithm |
| Favorites | Store all favorites |
| Locations | Stores locations for nearby place |
| Payments | Stores all payments information |
| PostImages | Store images of a post |
| PostLocations | Store relationship of post and location |
| Posts | Store information of post |
| PostStatus | Store status of post |
| Provinces | Store information of province |
| Questions | Store information of questions |
| ReportedPosts | Store information of reported post |
| Roles | Store roles of user |
| UserLogs | Store user log for messaging |
| Users | Store information of user |

## AdvanceSearchScore

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of AdvanceSearchScore |
| Name | nvarchar(MAX) |  |  |  |  | Name of Score |
| Score | int |  |  |  |  | Score of Title |

## Answer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of Answer |
| Content | nvarchar(MAX) |  |  |  |  | Content of Answer |
| CreatedDate | datetime |  |  |  |  | Created date |
| IsDeleted | bit |  |  |  |  | Is deleted |
| QuestionId | int |  |  | FK |  | Id of questions  FK to Questions.Id |
| IsRead | bit |  |  |  | False | Is receiver read this answer |

## Badwords

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of badwords |
| Word | nvarchar(100) |  |  |  |  | Content of word |
| TypeId | int |  |  | FK |  | Id of badwordtype  FK to BadWordTypes.Id |

## BadWordTypes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of bad words type |
| Name | nvarchar(50) |  |  |  |  | Name of bad word types |

## Categories

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of category |
| Name | nvarchar(50) |  |  |  |  | Name of category |
| Description | nvarchar(MAX) | x |  |  |  | Description of category |
| IsDeleted | bit |  |  |  | False | Is it deleted |

## ConfigurationRHFs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of configuration |
| Name | nvarchar(100) |  |  |  |  | Name of configuration |
| Value | nvarchar(100) |  |  |  |  | Value of configuration |

## Contacts

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Phone | nvarchar(20) |  |  |  |  | Phone to contact |
| Email | nvarchar(100) | x |  |  |  | Email to contact |
| Yahoo | nvarchar(50) | x |  |  |  | Yahoo to contact |
| Skype | nvarchar(50) | x |  |  |  | Skype to contact |
| PostIdContacts | int |  |  | PK |  | ID of Posts.Id |
| NameContact | nvarchar(50) |  |  |  |  | Name to contact |

## Districts

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of district |
| Name | nvarchar(MAX) |  |  |  |  | Name of district |
| Lat | float |  |  |  |  | Latitude of district |
| Lon | float |  |  |  |  | Longitude of district |
| ProvinceId | int |  |  | FK |  | Id of province  FK to Provinces.Id |
| IsDeleted | bit |  |  |  | False | Is it deleted |

## EmailTemplates

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of EmailTemplate |
| Name | nvarchar(MAX) |  |  |  |  | Name of template |
| Template | nvarchar(MAX) |  |  |  |  | Template of email |

## Facilities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| HasInternet | bit |  |  |  |  | Has internet |
| ElectricityFee | float |  |  |  |  | Fee of electricity |
| WaterFee | float |  |  |  |  | Fee of water |
| HasTVCable | bit |  |  |  |  | Has TV cable |
| HasBed | bit |  |  |  |  | Has bed |
| HasWaterHeater | bit |  |  |  |  | Has water heater |
| IsAllowCooking | bit |  |  |  |  | Allow cooking or not |
| HasMotorParkingLot | bit |  |  |  |  | Has motor parking lot |
| HasToilet | bit |  |  |  |  | Has toilet |
| HasAirConditioner | bit |  |  |  |  | Has Airconditioner |
| HasGarage | bit |  |  |  |  | Has Garage |
| IsStayWithOwner | bit |  |  |  |  | Stay with owner or not |
| RestrictHours | nvarchar(50) | x |  |  |  | Restrict hours |
| HasSecurity | bit |  |  |  |  | Has security |
| PostIdFacilities | int |  |  | PK |  | Id of post |
| FacilityTemplateId | int |  |  | FK |  | Id of facility template  FK to FacilityTemplates.Id |

## FacilityTemplates

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of facility templates |
| HasAirConditioner | bit |  |  |  |  | For advance search algorithm |
| HasBed | bit |  |  |  |  | For advance search algorithm |
| HasGarage | bit |  |  |  |  | For advance search algorithm |
| HasInternet | bit |  |  |  |  | For advance search algorithm |
| HasMotorParking | bit |  |  |  |  | For advance search algorithm |
| HasSecurity | bit |  |  |  |  | For advance search algorithm |
| HasToilet | bit |  |  |  |  | For advance search algorithm |
| HasTVCable | bit |  |  |  |  | For advance search algorithm |
| HasWaterHeater | bit |  |  |  |  | For advance search algorithm |
| IsAllowCooking | bit |  |  |  |  | For advance search algorithm |
| IsStayWithOwner | bit |  |  |  |  | For advance search algorithm |

## Favorites

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of favorites |
| UserId | int |  |  | FK |  | User who own favorite  FK to Users.Id |
| PostId | int |  |  | FK |  | Post of favorites  FK to Posts.Id |
| AddedDate | datetime |  |  |  |  | Day created |
| IsDeleted | bit |  |  |  | False | Is it deleted |

## Locations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of locations |
| Name | nvarchar(MAX) |  |  |  |  | Name of locations |
| DistrictId | int |  |  | FK |  | District id of location  FK to Districts.Id |
| IsCreatedByUser | bit |  |  |  |  | Is created by user |

## Payments

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of payments |
| PhoneNumber | nvarchar(20) |  |  |  |  | Phone number pay |
| CreatedDate | datetime |  |  |  |  | Created date |
| PostsId | int |  |  | FK |  | Post id to pay  FK to Posts.Id |

## Postimages

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of post image |
| PostId | int |  |  | FK |  | Post id of images  FK to Posts.Id |
| Path | nvarchar(MAX) |  |  |  |  | Path of images |
| IsDeleted | bit |  |  |  | False | Is it deleted |

## PostLocations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of postlocation |
| PostId | int |  |  | FK |  | Id of Post  FK to Posts.Id |
| LocationId | int |  |  | FK |  | Id of Location  FK to Locations.Id |

## Posts

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of Post |
| Title | nvarchar(MAX) |  |  |  |  | Title of post |
| CategoryId | int |  |  | FK |  | Category id of post  FK to Categories.Id |
| StatusId | int |  |  | FK |  | Status id of post  FK to PostStatus.Id |
| IsDeleted | bit |  |  |  | False | Is it deleted |
| CreatedDate | datetime |  |  |  |  | Created date |
| EditedDate | datetime | x |  |  |  | Edited date |
| RenewDate | datetime | x |  |  |  | Renew data ( latest date when user renew the post ) |
| DistrictId | int |  |  | FK |  | District Id of post  FK to Districts.Id |
| NumberAddress | nvarchar(MAX) |  |  |  |  | Detail address |
| Price | float |  |  |  |  | Price |
| Area | float |  |  |  |  | Area |
| PhoneActive | nvarchar(20) |  |  |  |  | Phone number use to active |
| Description | nvarchar(MAX) | x |  |  |  | Description of post |
| Lat | float |  |  |  |  | Latitude of post |
| Lon | float |  |  |  |  | Longitude of post |
| UserId | int | x |  | FK |  | User id who create post  FK to Users.Id |
| Street | nvarchar(50) |  |  |  |  | Street name of post |
| NearbyPlace | nvarchar(MAX) | x |  |  |  | List of nearby place |
| ExpiredDate | datetime |  |  |  |  | Expired date |
| Code | nvarchar(MAX) |  |  |  |  | Code use for pay or edit post |

## PostStatus

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of post status |
| Name | nvarchar(MAX) | x |  |  |  | Name of post status |
| Description | nvarchar(MAX) | x |  |  |  | Description of post status |
| IsDeleted | bit |  |  |  | False | Is deleted |

## Provinces

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of province |
| Name | nvarchar(MAX) |  |  |  |  | Name of province |
| Lat | float |  |  |  |  | Latitude of province |
| Lon | float |  |  |  |  | Longitude of province |
| IsDeleted | bit |  |  |  | False | Is deleted |

## Questions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of question |
| Content | nvarchar(MAX) |  |  |  |  | Content of question |
| SenderId | int | x |  | FK |  | Id of sender  FK to User.Id |
| CreatedDate | datetime |  |  |  |  | Created date |
| IsRead | bit |  |  |  |  | Is receiver read |
| IsDeleted | bit |  |  |  |  | Is deleted |
| PostId | int |  |  | FK |  | Post id of questions  FK to Posts.Id |
| Title | nvarchar(MAX) |  |  |  |  | Title of question |
| SenderEmail | nvarchar(50) |  |  |  |  | Email of sender ( in case not a user) |

## ReportedPosts

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of reported posts |
| PostId | int |  |  | FK |  | Post id of report  FK to Posts.Id |
| ReportedBy | int |  |  | FK |  | Reported by which user  FK to Users.Id |
| ReportedDate | datetime |  |  |  |  | Date when report submitted |
| Reason | nvarchar(MAX) |  |  |  |  | Reason to report |
| IsIgnored | bit |  |  |  | False | Admin approved or not |

## Roles

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of role |
| Name | nvarchar(20) |  |  |  |  | Name of role |
| Description | nvarchar(MAX) | x |  |  |  | Description of role |
| IsDeleted | bit |  |  |  |  | Is it deleted |

## UserLogs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of userlogs |
| Message | nvarchar(MAX) |  |  |  |  | Content of message user log |
| UserId | int |  |  | FK |  | User id  FK to Users.Id |
| IsRead | bit |  |  |  |  | Is read by user |
| CreatedDate | datetime |  |  |  |  | Created date |

## Users

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Type | Null | Unique | P/F Key | Default | Description |
| Id | int |  |  | PK |  | Id of User |
| Username | nvarchar(50) |  |  |  |  | Username |
| Password | nvarchar(MAX) | x |  |  |  | Password |
| Email | nvarchar(50) | x |  |  |  | Email of user |
| PhoneNumber | nvarchar(15) | x |  |  |  | Phone number |
| Name | nvarchar(MAX) | x |  |  |  | Real name |
| CreatedDate | datetime |  |  |  |  | Created date |
| LastUpdate | datetime |  |  |  |  | Last update |
| IsDeleted | bit |  |  |  | False | Is deleted |
| OpenIdURL | nvarchar(MAX) | x |  |  |  | Open id URL |
| RoleId | int |  |  | FK |  | Role of user  FK to Roles.Id |

# References

[1] Microsoft, ' ASP.NET MVC Framework ', viewed 18 February 2011

<http://www.asp.net/mvc/whatisaspmvc>

[2] MSDN, ' UML Class Diagrams: Guidelines ', viewed 18 February 2011

<http://msdn.microsoft.com/en-us/library/dd409416.aspx>

[3] MSDN, ' UML Sequence Diagrams: Guidelines ', viewed 18 February 2011

<http://msdn.microsoft.com/en-us/library/dd409389.aspx>

[4] Wikipedia, ' Dependency Injection ', viewed 18 February 2011

<http://en.wikipedia.org/wiki/Dependency_injection>