

**COVID CARE**

**Software Design Document**

**Table of Contents**

[I. Overview 3](#_Toc98340417)

[1. Code Packages/Namespaces 3](#_Toc98340418)

[II. Code Designs 3](#_Toc98340419)

[1. <Feature/Function Name1> 3](#_Toc98340420)

[a. Class Diagram 3](#_Toc98340421)

[b. Class Specifications 4](#_Toc98340422)

[c. Sequence Diagram(s) 4](#_Toc98340423)

[d. Database queries 4](#_Toc98340424)

[2. <Feature/Function Name2> 5](#_Toc98340425)

[III. Database Design 5](#_Toc98340426)

[1. Database Schema 5](#_Toc98340427)

[2. Table Description 5](#_Toc98340428)

# I. Overview

## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

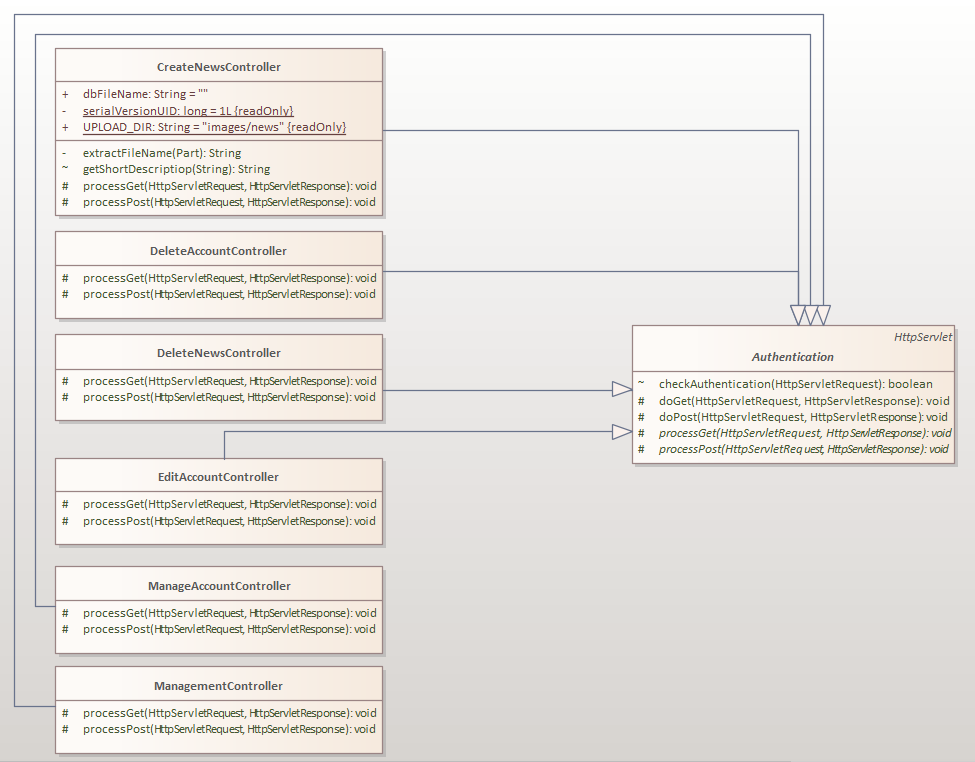
|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *Controller* | *Acting as an intermediary between the Model and the View, it is responsible for receiving requests from the client, then processing the request, loading the model and sending data through the corresponding view and then returning the results to the client.* |
| *02* | *Dal* | *A pattern in which objects are reserved for communication with the Data Layer.* |
| *03* | *Model* | *Are classes that contain object information (data), interact with database access.* |
| *04* | *WebSocket* | *Is a protocol that supports two-way communication between the client and the server to create a strong data exchange connection. WebSocket is often used in real-time requirements such as chat applications or graphs* |
| *05* | *utils* | *The package contains the definition of some useful classes that provide a lot of functionality. The package mainly contains collection classes useful for working with groups of objects. This package also contains definitions of classes that provide date and time utilities and many more* |
| *06* | *view* | *Is the place to receive data from the model, database and arrange them correctly thanks to the request from the controller and transmit it to the client.* |

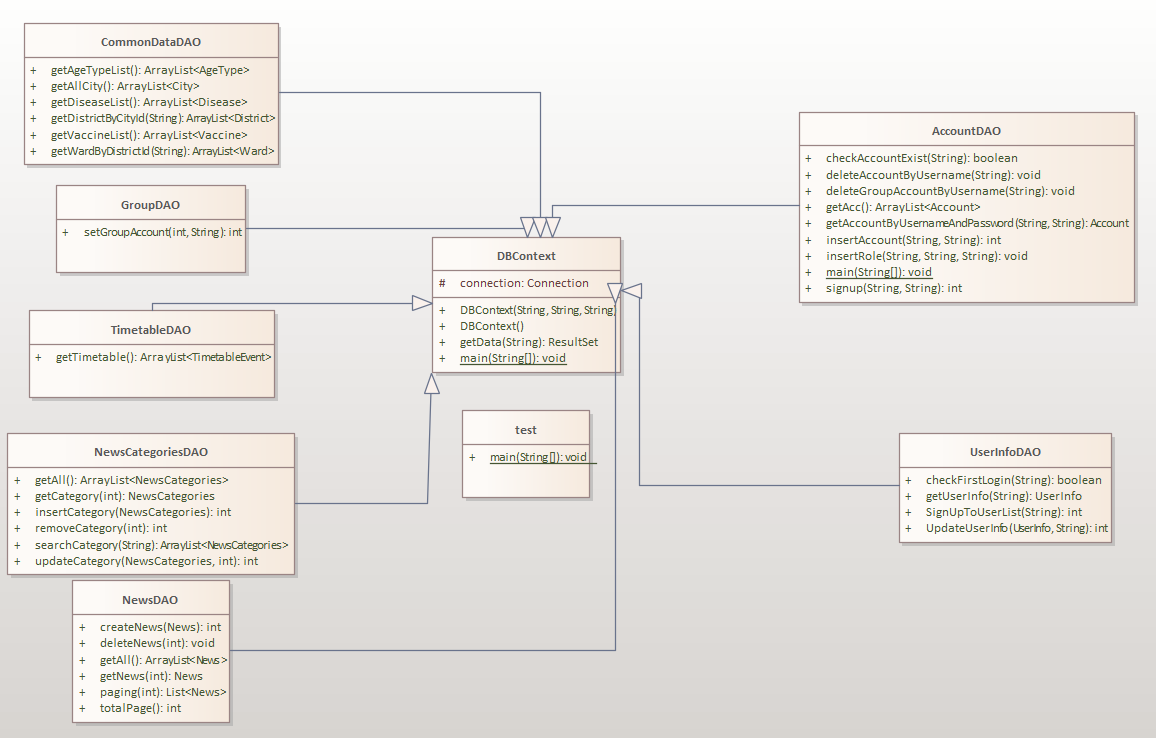
# II. Code Designs

## 1. <Feature/Function Name1>

*[Provide the detailed design for the function <Feature/Function Name1>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

### Class Diagram





### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

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

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

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

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

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

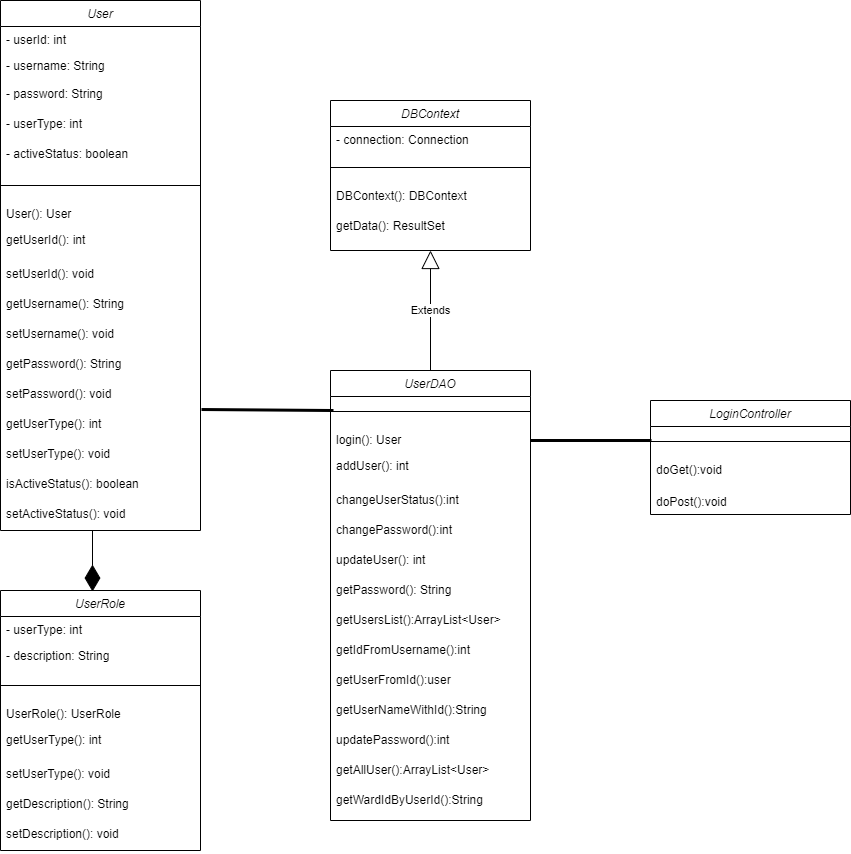


### d. Database queries

*[Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]*

## 2. Login

**a. Class Diagram:**

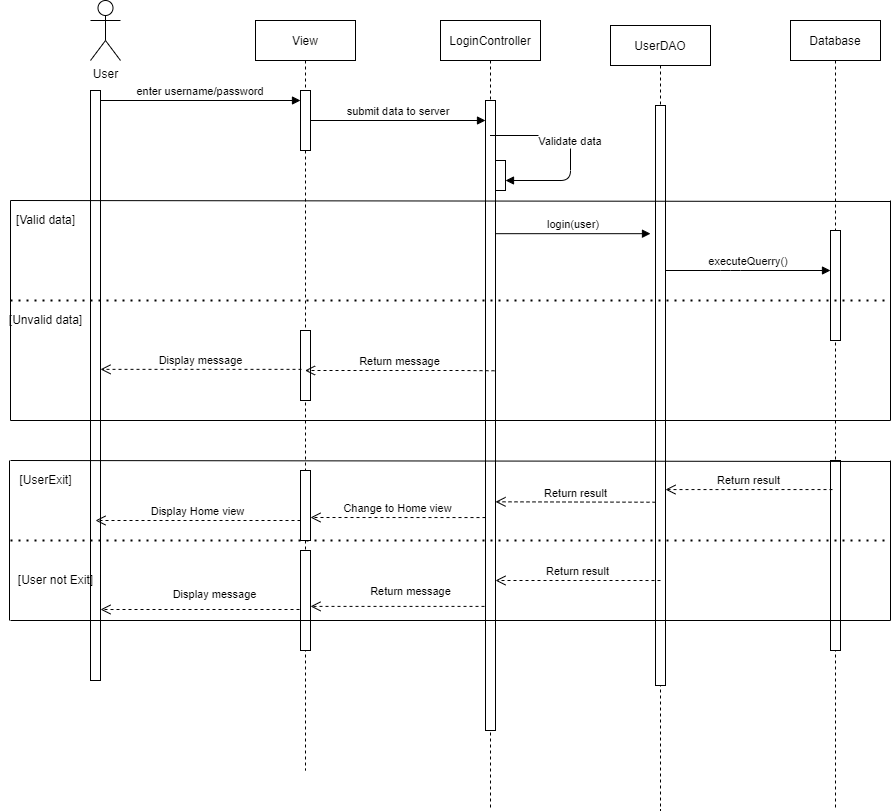


*Login\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

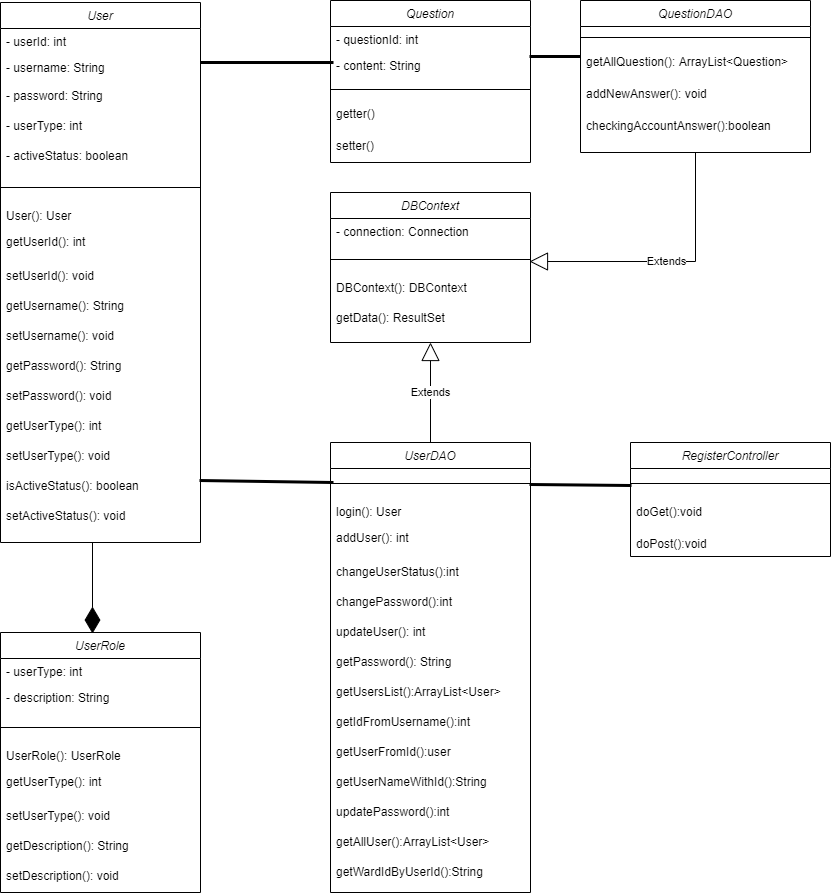
1. **Sequense Diagram**



*Login\_Sequense Diagram*

## 2. Register

**a. Class Diagram:**

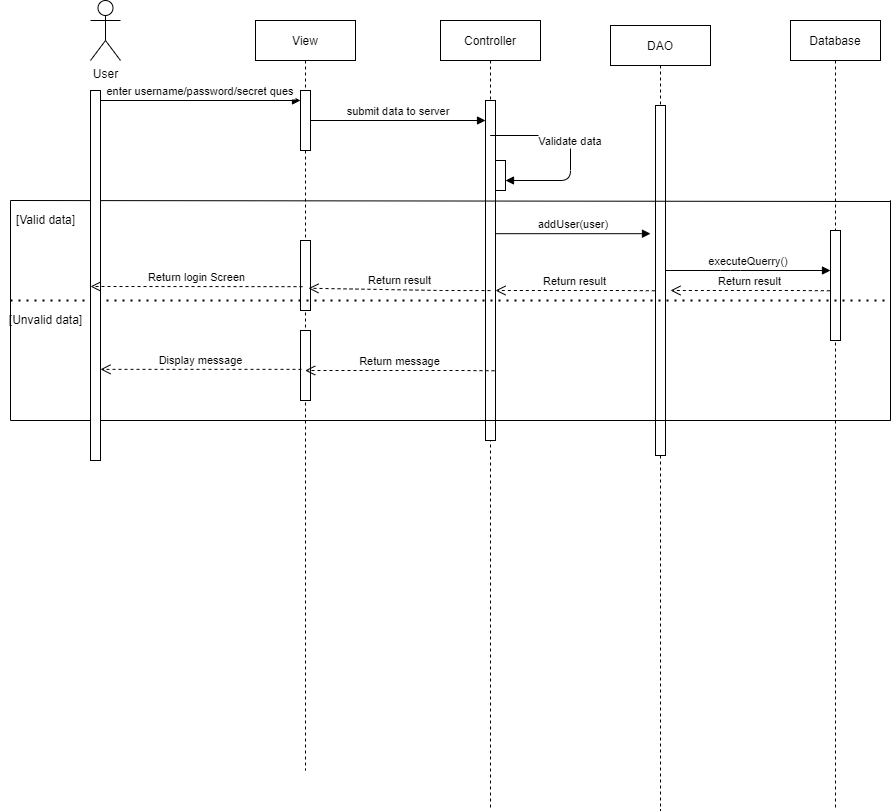


*Register\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | addUser(user) | Add new User information to Database |
| 02 | getAllQuestion() | Get all question |

1. **Sequense Diagram**



*Register\_Sequense Diagram*

## 2. View list Account

**a. Class Diagram:**

*View list Account\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

1. **Sequense Diagram**

*View list Account \_Sequense Diagram*

## 2. Ban Account

**a. Class Diagram:**

*Ban Account\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

1. **Sequense Diagram**

*Ban Account \_Sequense Diagram*

## 2. Edit Account

**a. Class Diagram:**

*Edit Account\_Class Diagram*

1. **Class Specification**

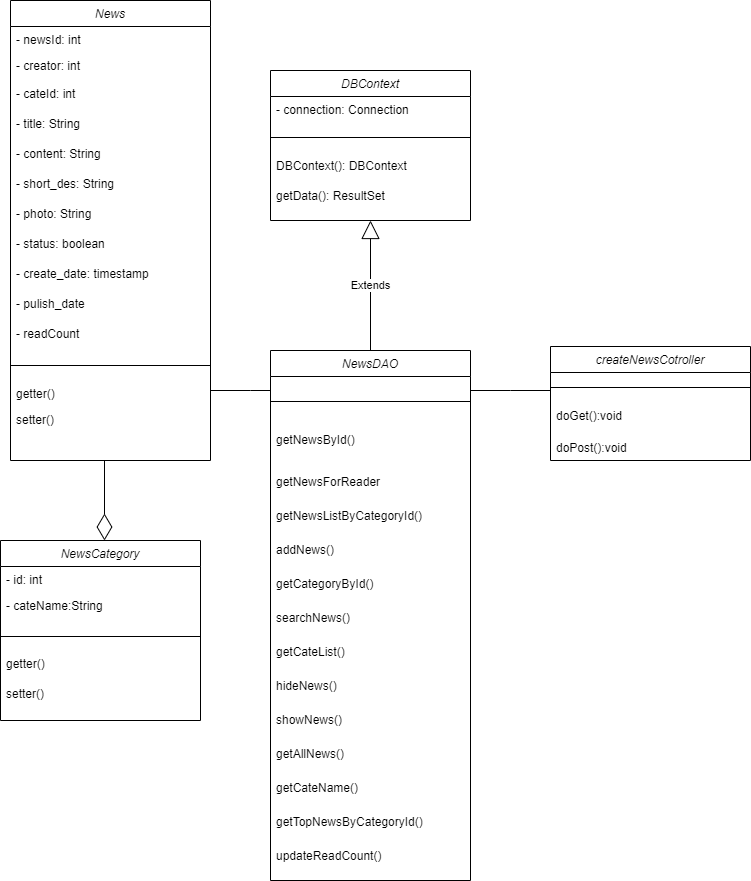
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

1. **Sequense Diagram**

*Edit Account\_Sequense Diagram*

## 2. Create News

**a. Class Diagram:**

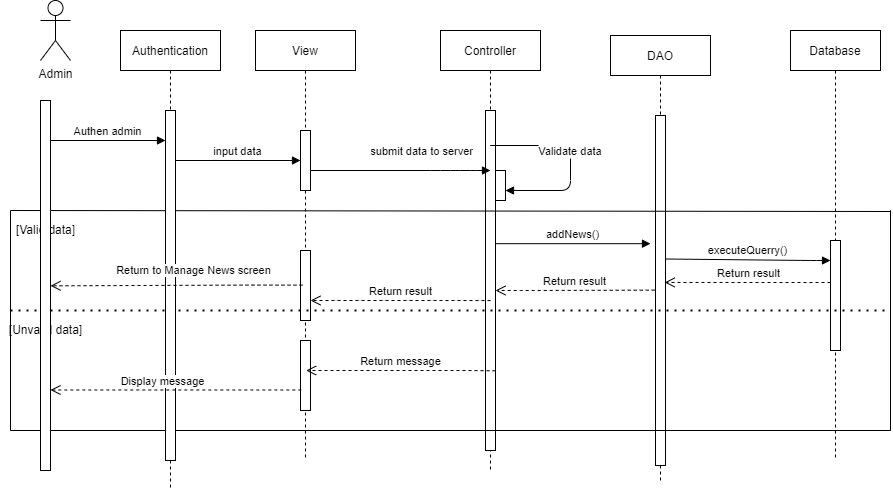


*Create News\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getNewsById() | Get News by News id |
| 02 | getNewsForReader() | Get News with news status equad to 1 |
| 03 | getNewsListByCategoryId() | Get News with cate id |
| 04 | addNews() | Insert new News |
| 05 | updateNews() | Update News |
| 06 | getCategoryById() | Get name of category by category id |
| 07 | searchNews() | Search News contain input text |
| 08 | getCateList() | Get list category |
| 09 | hideNews() | Update status News equad to 0 |
| 10 | showNews() | Update status News equad to 1 |
| 11 | getAllNews() | Get all news |
| 12 | getCateName() | Get category name |
| 13 | getTopNewsByCategoryId() | Get top 4 news with closest publish date |
| 14 | updateReadCount | Update time of read |

1. **Sequense Diagram**



*Create News \_Sequense Diagram*

## 2. View Detail News

**a. Class Diagram:**

*View Detail News\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

1. **Sequense Diagram**

*View Detail News \_Sequense Diagram*

## 2. Update News

**a. Class Diagram:**

*Update News\_Class Diagram*

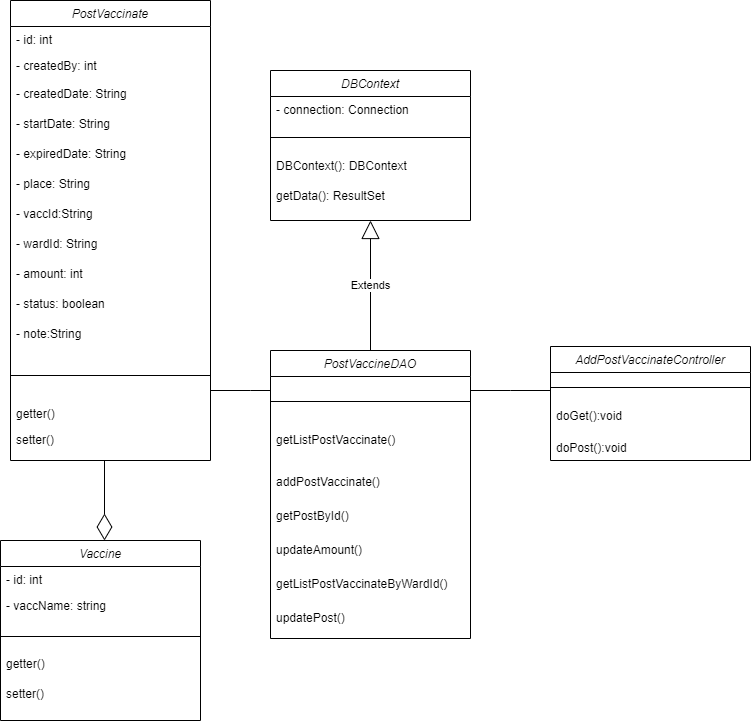
1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

1. **Sequense Diagram**

*Update News \_Sequense Diagram*

## 2. Create Post Vaccinate

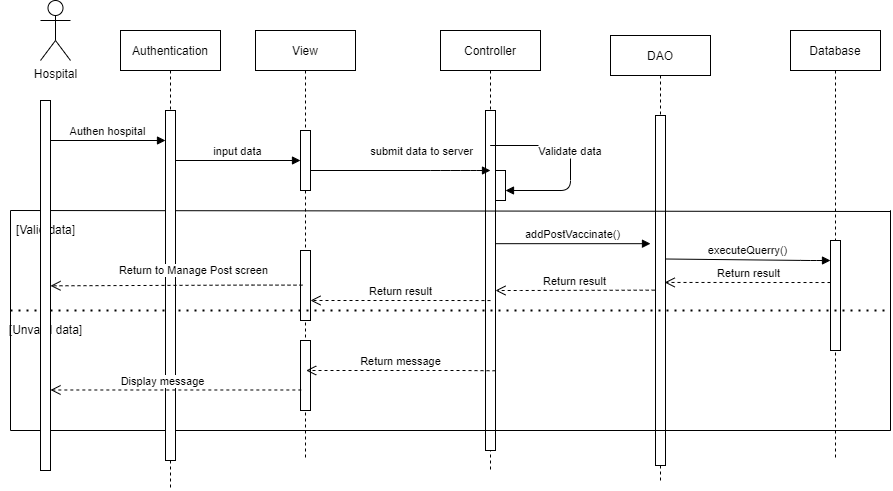
**a. Class Diagram:** 

*Create Post Vaccinate\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getListPostVacinate() | Get all Post in Database |
| 02 | addPostVaccinate() | Add new Post into database |
| 03 | getPostById() | Get Post by id of post |
| 04 | updateAmount() | Update amount number vaccin. |
| 05 | getListPostVaccinateByWarId() | Get list post with wardId |
| 06 | updatePost() | Update all attributes of Post |

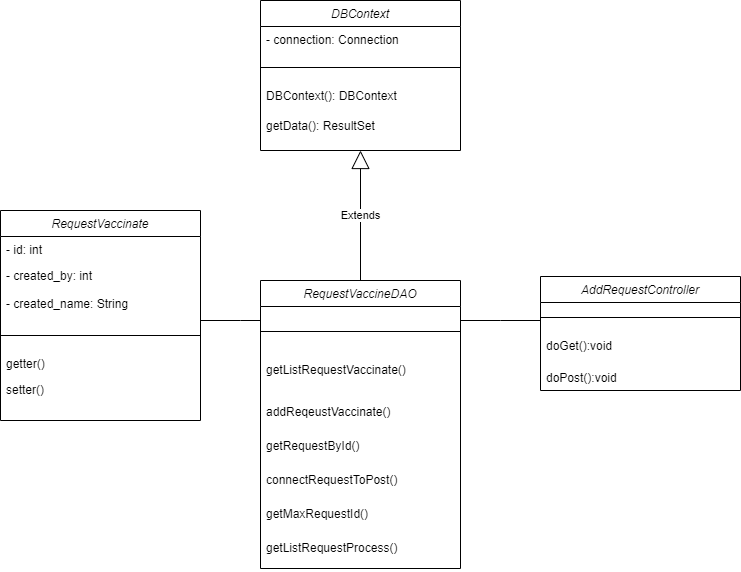
1. **Sequense Diagram**



*Create Post Vaccinate \_Sequense Diagram*

## 2. Create Request Vaccinate

**a. Class Diagram:**

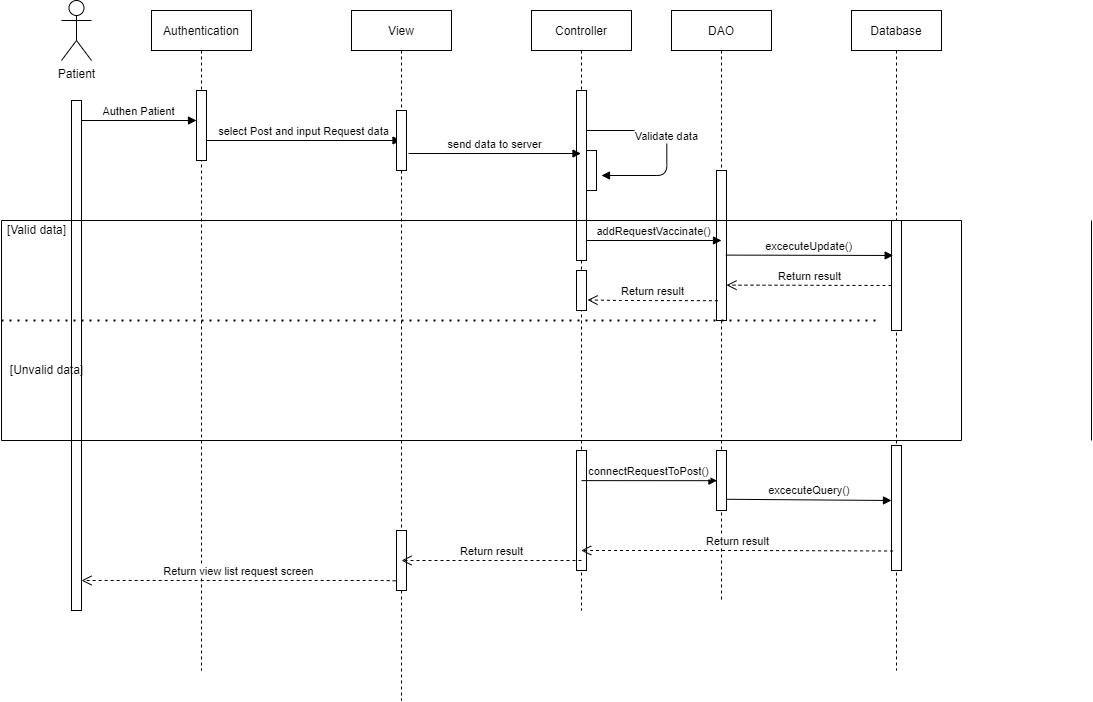


*Create Request Vaccinate\_Class Diagram*

1. **Class Specification**

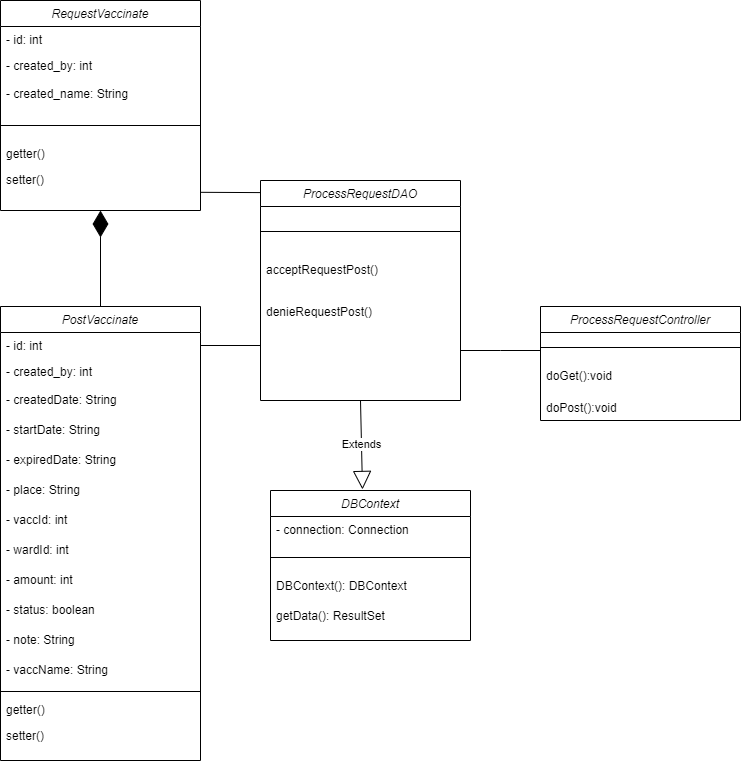
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getListRequestVaccinate() | Get list request vaccinate of user |
| 02 | addRequestById() | Add new request. |
| 03 | connectRequestToPost() | Add request to Post |
| 04 | getMaxRequestId() | Get closest request id |
| 05 | getListRequestProcess() | Get list request with status equal to 2 |

1. **Sequense Diagram**



*Create Request Vaccinate \_Sequense Diagram*

## 2. Appcept/ Denie Request

**a. Class Diagram:** 

*Accept Request\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | acceptRequestPost() | Set status request of post is 1 |
| 02 | denieRequestPost() | Set status request of post is 0 |

1. **Sequense Diagram**

*Accept Request \_Sequense Diagram*

## 2. Recover Password

**a. Class Diagram:**

*Recover Password\_Class Diagram*

1. **Class Specification**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | User() | Contructor of object User. |
| 02 | getUserId() | Return value of attribute userId |
| 03 | setUserId() | Set value to attribute userId |
| 04 | getUsername() | Return value of attribute userId |
| 05 | setUsername() | Set value to attribute userId |
| 06 | getPassword() | Return attribute password |
| 07 | setPassword() | Set value to attribute password |
| 08 | getUserType() | Return value of attribute userType |
| 09 | setUserType () | Set value to attribute userType |
| 10 | isActiveStatus() | Return value of attribute activeStatus |
| 11 | setActiveStatus() | Set value to attribute activeStatus |
| 12 | Login(User user) | Get data from database then create and return User object |
| 13 | addUser(User user) | Add data object User to database |
| 14 | ChangeUserStatus(userId, activeFlag) | Update data column *activeStatus* of table Users |
| 15 | changePassword(user) | Update data column *password* of table Users |
| 16 | updateUser(user) | Update data all column of table Users |
| 17 | getUsersList() | Get all data from table User, create object User then return list User. |
| 18 | getIdFromUsername(username) | Return Id of username input. |
| 19 | getUserFromId(Id) | Return User have Id same with input Id. |
| 20 | getUserNameWithId(Id) | Return username at object User have id same with input Id |
| 21 | getWardIdByUserId(userId) | Return ward id belong to user. |

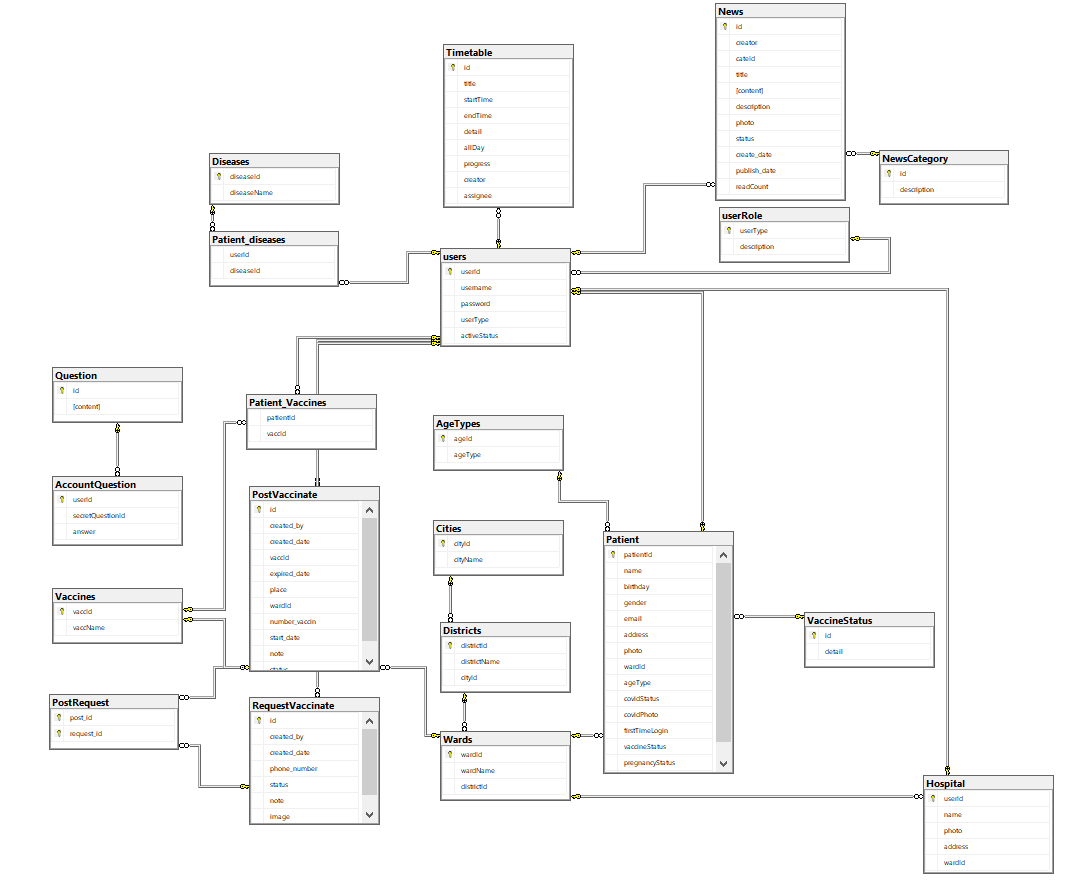
1. **Sequense Diagram**

*Recover Password \_Sequense Diagram*

# 

# III. Database Design

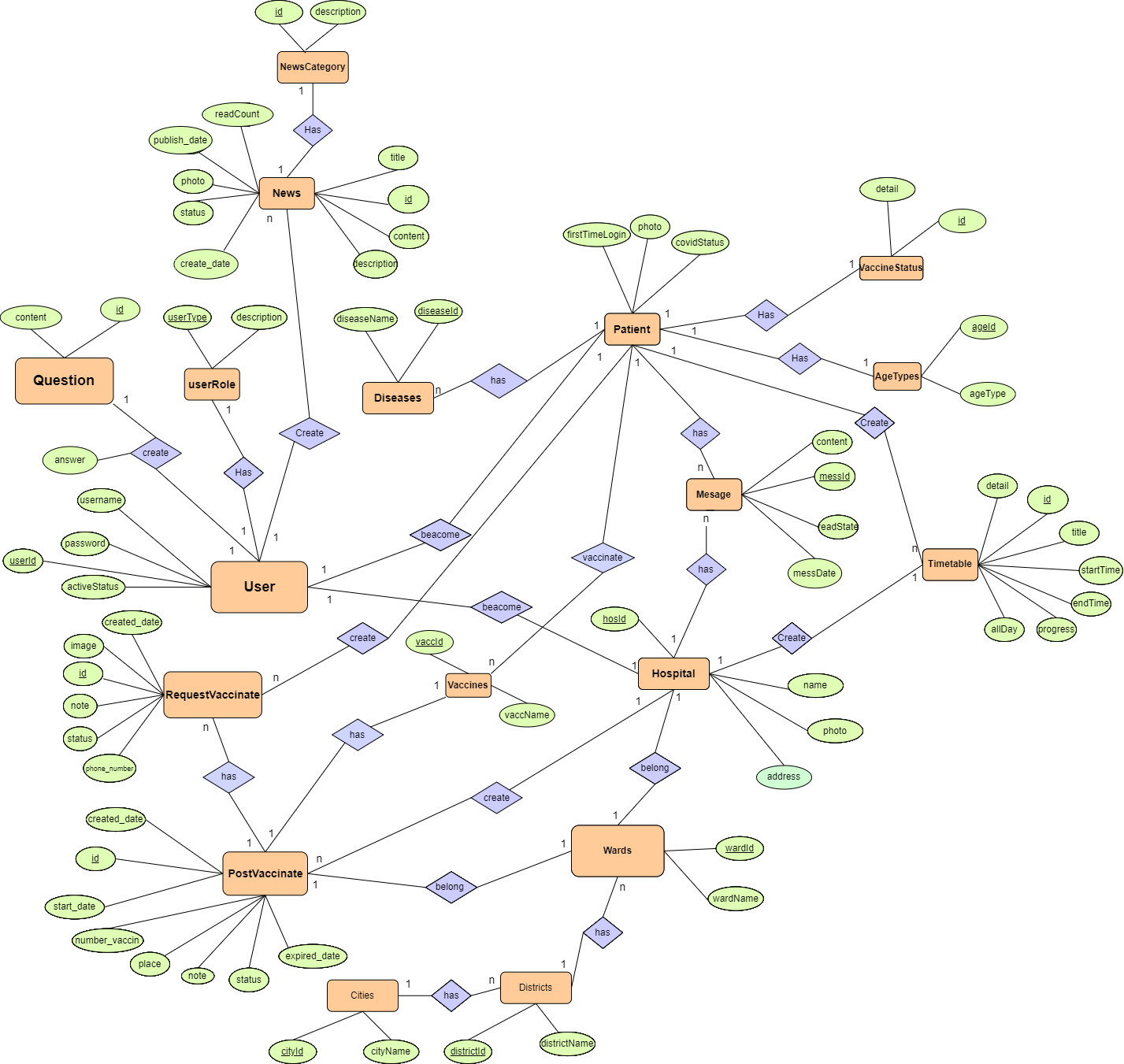
## Database Diagram:

1. Image:
2. *Img:Database Diagram.*

## 2. Table Description

|  |  |  |
| --- | --- | --- |
| **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* |  |  |

## 3: ERD:



*Img:Database Diagram.*