

**FPT STUDENT BLOG**

**Software Design Document**

**Prepared by Group 1 - SE1504**

**Nguyễn Thiện Quang**

**Nguyễn Dương**

**Lê Anh Khoa**

**Bùi Thị Thùy Linh**

**Trần Trọng Nhân**

– Ho Chi Minh, November 2021 –

**Table of Contents**

[**I. Overview**](#_heading=h.gjdgxs) **5**

[1. Code Packages/Namespaces](#_heading=h.30j0zll) 5

[2. Database Schema](#_heading=h.1fob9te) 6

[**II. Code Designs**](#_heading=h.w8adc8meybxo) **8**

[1. Login](#_heading=h.x8zk1dloz1sw) 8

[a. Class Diagram](#_heading=h.q0vyvza7wm70) 8

[b. Sequence Diagram(s)](#_heading=h.ofgew6ppy27l) 9

[c. Database queries](#_heading=h.7b08bdw7f4o0) 9

[2. Create post](#_heading=h.2s8eyo1) 9

[a. Class Diagram](#_heading=h.9fikre2g2i5c) 9

[b. Sequence Diagram(s)](#_heading=h.3trjffw5okmx) 10

[c. Database queries](#_heading=h.8anxb3lfbx3n) 10

[3. Comment](#_heading=h.nvot83hc1zdd) 10

[a. Class Diagram](#_heading=h.5ybfs8dfswvb) 10

[b. Sequence Diagram(s)](#_heading=h.6nquoyrn3khp) 11

[c. Database queries](#_heading=h.qdrb9tksd69y) 11

[4. Create Notification](#_heading=h.ql33ur6mtw51) 11

[a. Class Diagram](#_heading=h.l9qi6p4axv1i) 11

[b. Sequence Diagram(s)](#_heading=h.ugzd1hk970yh) 12

[c. Database queries](#_heading=h.h8t1nxipqpbb) 12

[5. Detail Post](#_heading=h.8s8kj7i0kqo4) 12

[a. Class Diagram](#_heading=h.7ckzkit0yy6p) 12

[b. Sequence Diagram(s)](#_heading=h.te1lm6jpkvo) 13

[c. Database queries](#_heading=h.16e79cjg2gky) 13

[6. Vote](#_heading=h.y7ceq8daza12) 13

[a. Class Diagram](#_heading=h.btn6n8gijnkd) 13

[b. Sequence Diagram(s)](#_heading=h.njkheby3d07v) 14

[c. Database queries](#_heading=h.h8pgoobsd4ml) 14

[7. Edit Post](#_heading=h.ovqqeac09h5p) 14

[a. Class Diagram](#_heading=h.i37ev3e62vcs) 15

[b. Sequence Diagram(s)](#_heading=h.v5be70qvd4uh) 15

[c. Database queries](#_heading=h.z7shtwnkgs9n) 15

[8. Search Post](#_heading=h.2lcjqj5vblt3) 15

[a. Class Diagram](#_heading=h.hcp9el33wr7f) 16

[b. Sequence Diagram(s)](#_heading=h.9zfyycst4twk) 16

[c. Database queries](#_heading=h.ajn2os1vf4ok) 16

[9. Edit Profile](#_heading=h.l9uq2hwzu1qw) 16

[a. Class Diagram](#_heading=h.lxcgheoz2yrh) 17

[b. Sequence Diagram(s)](#_heading=h.1k6ny16ah3c8) 17

[c. Database queries](#_heading=h.pikdmxumu4fw) 17

[10. Create Mentor](#_heading=h.1d08c9m4shfh) 17

[a. Class Diagram](#_heading=h.bye8gom2ym5q) 18

[b. Sequence Diagram(s)](#_heading=h.8uro1ayh9wgw) 18

[c. Database queries](#_heading=h.xdoey7cywsrx) 18

[11. Add Category](#_heading=h.56u788sm5k94) 18

[a. Class Diagram](#_heading=h.vg5ocxhjutlj) 19

[b. Sequence Diagram(s)](#_heading=h.82tmimgyqtrv) 19

[c. Database queries](#_heading=h.hq7hd3gftpx2) 19

[12. Update Category](#_heading=h.ppu4gk8fkpen) 19

[a. Class Diagram](#_heading=h.y6yp63rkgm86) 20

[b. Sequence Diagram(s)](#_heading=h.m717w7e60j3p) 20

[c. Database queries](#_heading=h.v4rtdtoqzyda) 20

[13. Ban User](#_heading=h.uhj8rb6a51gg) 20

[a. Class Diagram](#_heading=h.ionso84rnvvc) 20

[b. Sequence Diagram(s)](#_heading=h.xutnewz75vi2) 21

[c. Database queries](#_heading=h.z0yve590i3f4) 21

[14. Approve Post](#_heading=h.wvebmzsfsxas) 21

[a. Class Diagram](#_heading=h.8v6phjm134g) 22

[b. Sequence Diagram(s)](#_heading=h.gfdql1cp7172) 22

[c. Database queries](#_heading=h.upqb1kx7pfny) 22

[15. Deny Post](#_heading=h.d60bk33k507m) 22

[a. Class Diagram](#_heading=h.uckhl15nns55) 22

[b. Sequence Diagram(s)](#_heading=h.off4re39z9st) 23

[c. Database queries](#_heading=h.pmaft3kqjdns) 23

[16. Sign up](#_heading=h.t6gu5un8fm6b) 23

[a. Class Diagram](#_heading=h.rdq0q0ri9dy) 23

[b. Sequence Diagram(s)](#_heading=h.21w66u4f97cl) 24

[c. Database queries](#_heading=h.9oc05oymgqf6) 24

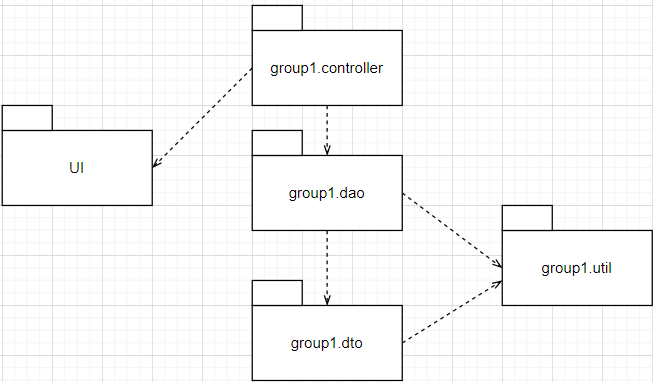
[17. Logout](#_heading=h.8f7c0viwfffv) 24

[a. Class Diagram](#_heading=h.d2l6cyephvw8) 24

[b. Sequence Diagram(s)](#_heading=h.8wx8185ht9u1) 25

# I. Overview

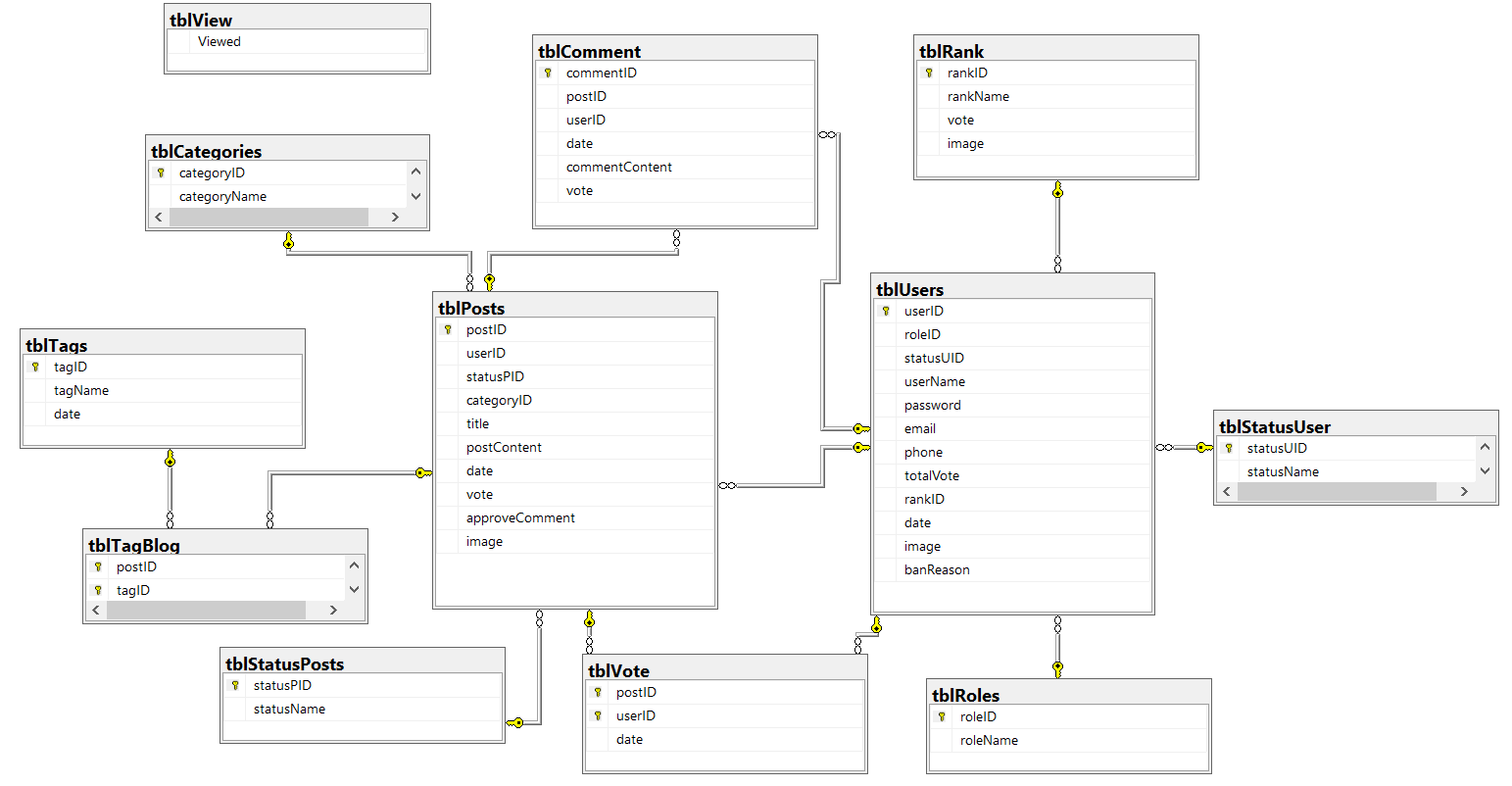
## 1. Code Packages/Namespaces



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

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 01 | group1.controller | Controllers are used to handle user requests and then communicate with the model  - AddCategoryController  - ApproveDenyPostController  - CategoryController  - CommentController  - CreateAccountController  - CreateNotificationController  - CreatePostController  - CreateMentorController  - EditPostController  - EditProfileController  - GetPostByTagController  - GoogleController  - HideUserController  - HomeController  - LoginController  - LogoutController  - MainController  - ManagerController  - ProfileController  - SearchController  - ShowDetailPostController  - SubmitAgainController  - UpdateCategoryController  - UpdateRankController  - ViewPostController  - ViewYourPostController  - VoteController |
| 02 | group1.dao | Set of queries to get data from the database and send it back to the controller  - CategoryDAO  - CommentDAO  - PostDAO  - RankDAO  - ReportDAO  - StatusDAO  - TagDAO  - UserDAO |
| 03 | group1.dto | Variables that store data retrieved from the database  - CategoryDTO  - CommentDTO  - PostDTO  - RankDTO  - ReportDTO  - StatusDTO  - TagDTO  - UserDTO  - UserError |
| 04 | group1.util | Open connection with database  - DBUtils |

## 2. Database Schema



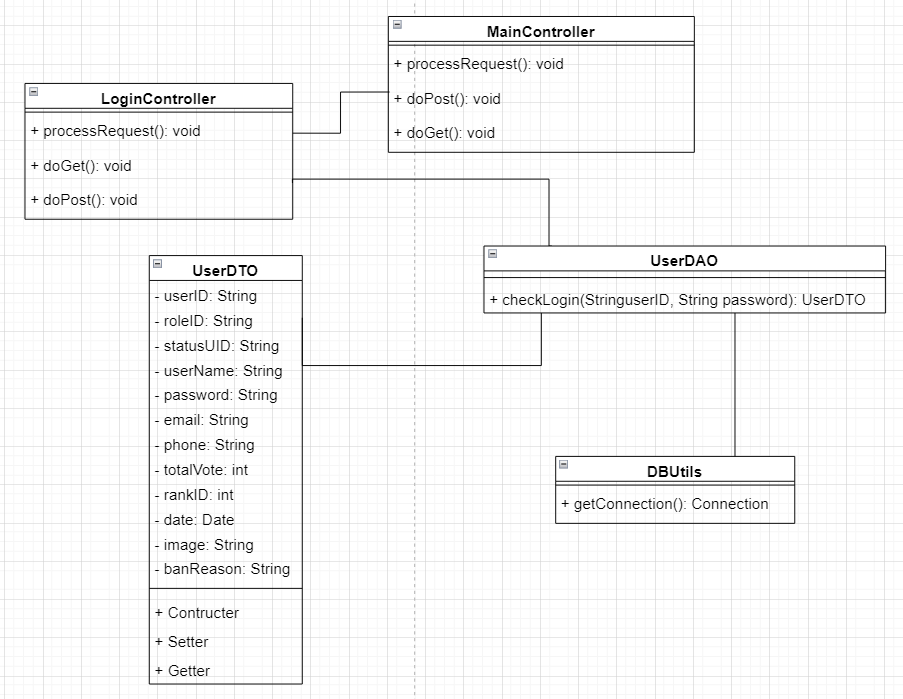
***Table descriptions & package class naming conventions are as below***

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | tblPosts | Table to store posts on the system  - Primary keys: postID  - Foreign keys: categoryID |
| 02 | tblUsers | The table that stores the users in the system  - Primary keys: userID  -Foreign keys: roleID, rankID |
| 03 | tblComment | Table that stores user comments for each post  - Primary keys: commentD  -Foreign keys: postID, userID |
| 04 | tblVote | Table that stores user votes for each article  -Foreign keys: postID, userID |
| 05 | tblCategories | Table to store article categories in the system  - Primary keys: categoryID |
| 06 | tblRank | The table stores the hierarchy of users in the system  - Primary keys: rankID |
| 07 | tblStatusUser | Table to save the state of users in the system  - Primary keys: statusID |
| 08 | tblRoles | Table to store user permissions in the system  - Primary keys: roleID |
| 09 | tblStatusPosts | The table stores the browsing, waiting, or rejecting statuses of articles in the system  - Primary keys: statusPID |
| 10 | tblTagBlog | The table stores the tags included in each article  -Foreign keys: postID, tagID |
| 11 | tblTags | Table to store tags in the system  - Primary keys: tagID |
| 12 | tblView | The table stores the number of users accessing the system |

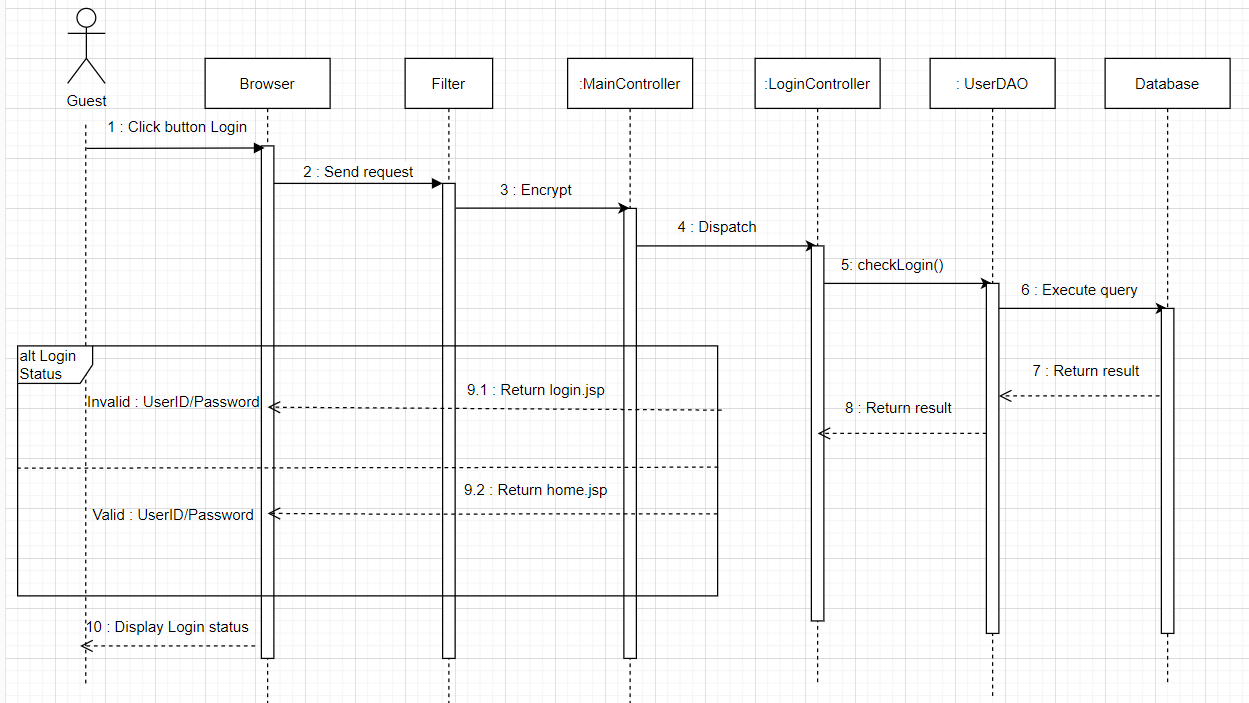
# II. Code Designs

## 1. Login

### a. Class Diagram

****

### b. Sequence Diagram(s)

****

### c. Database queries

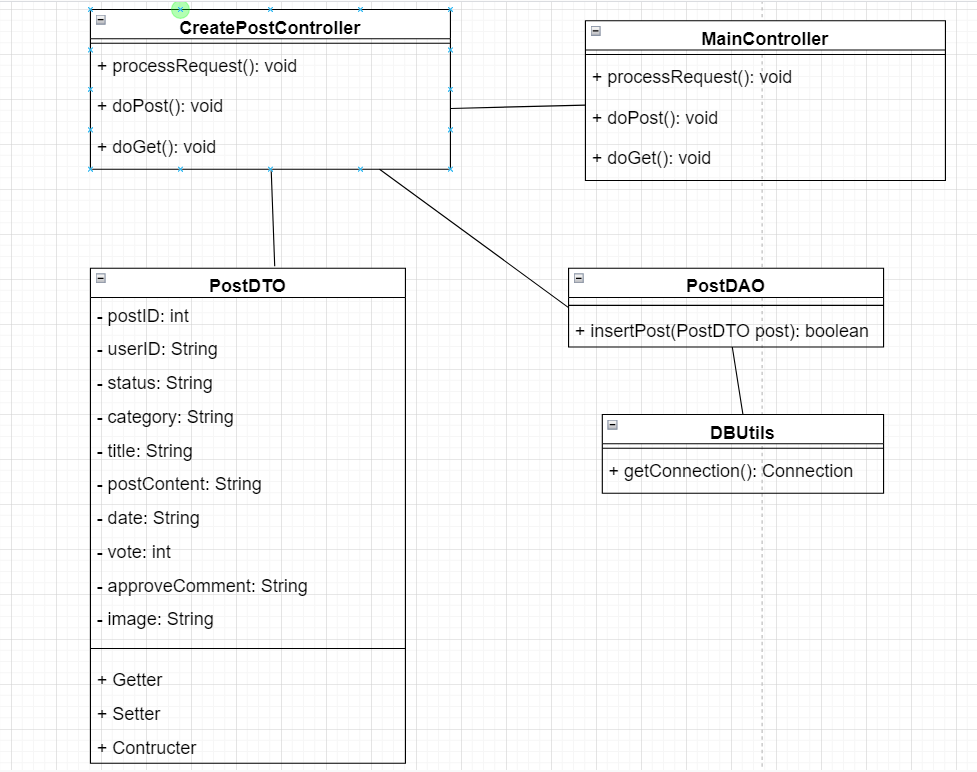
* SELECT roleID, statusUID, userName, email, phone, totalVote, rankID, date, image, banReason

FROM tblUsers

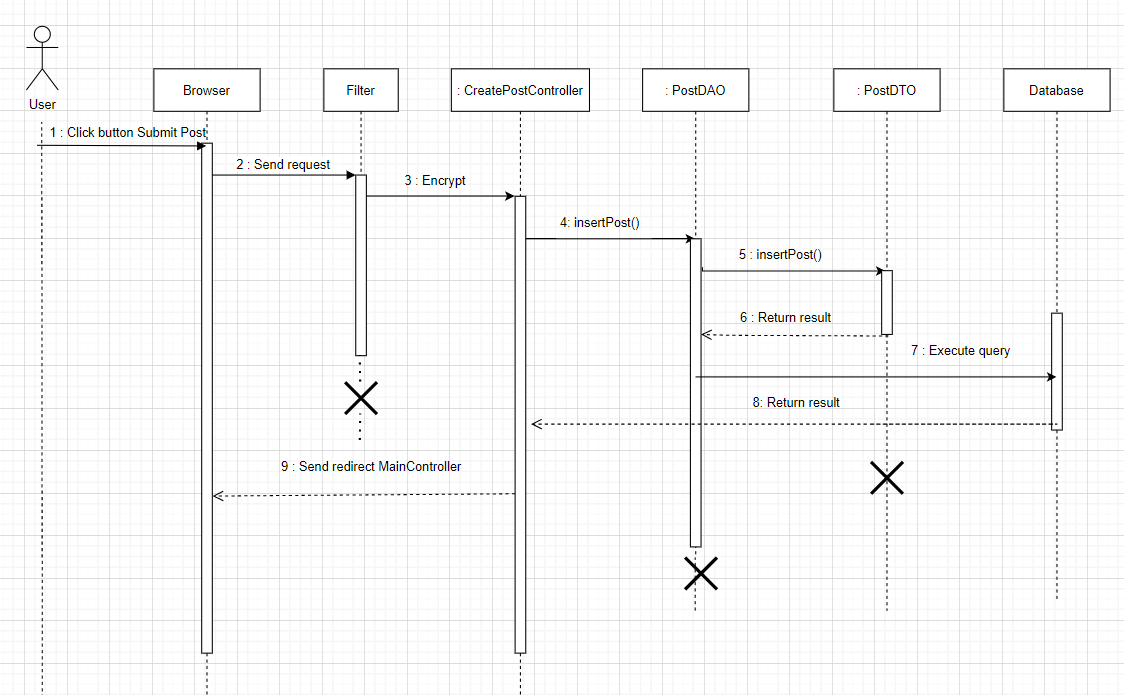
WHERE userID=? AND password=?

## 2. Create post

### a. Class Diagram



### b. Sequence Diagram(s)

****

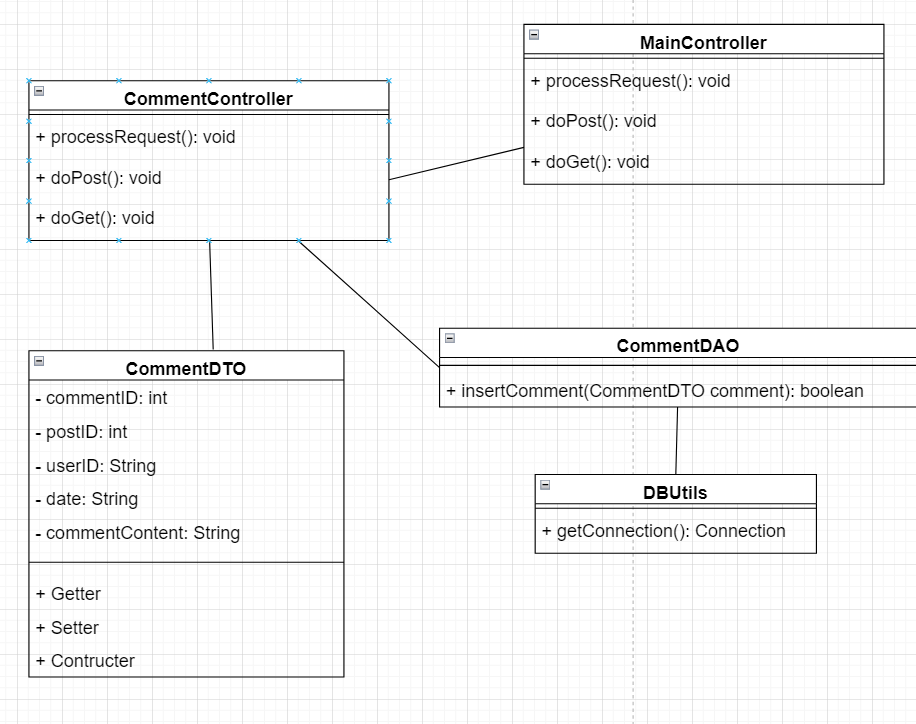
### c. Database queries

* Insert into tblPosts(userID, title, statusPID, categoryID, postContent ,date, vote, image )

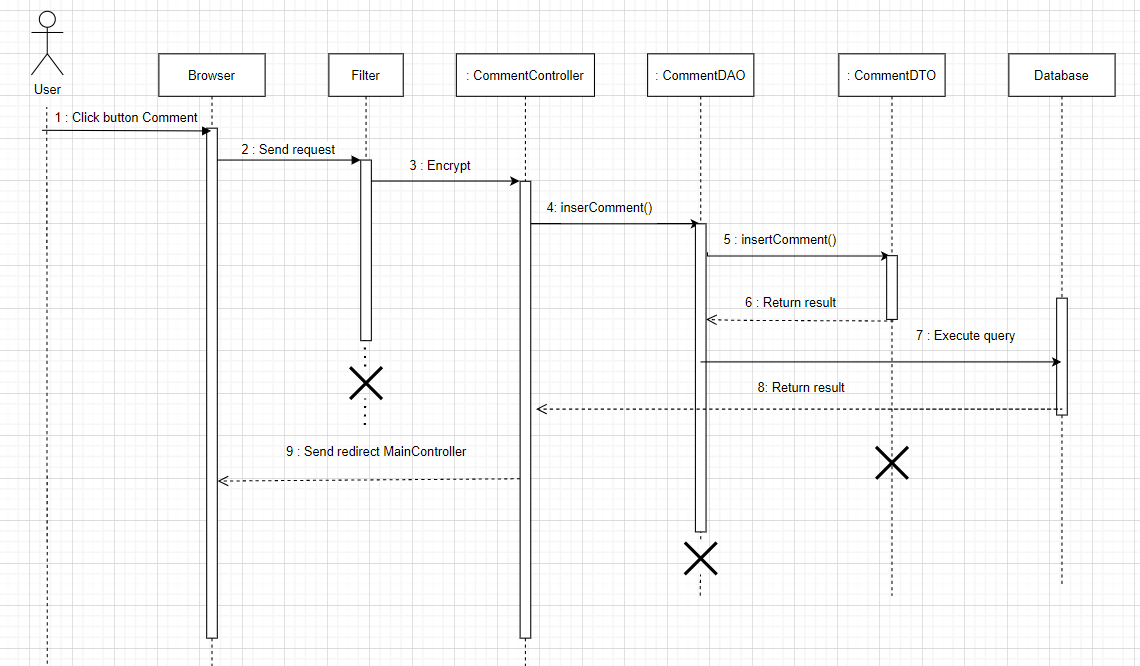
values(?,?,?,?,?,?,?,?)

## 3. Comment

### a. Class Diagram



### b. Sequence Diagram(s)

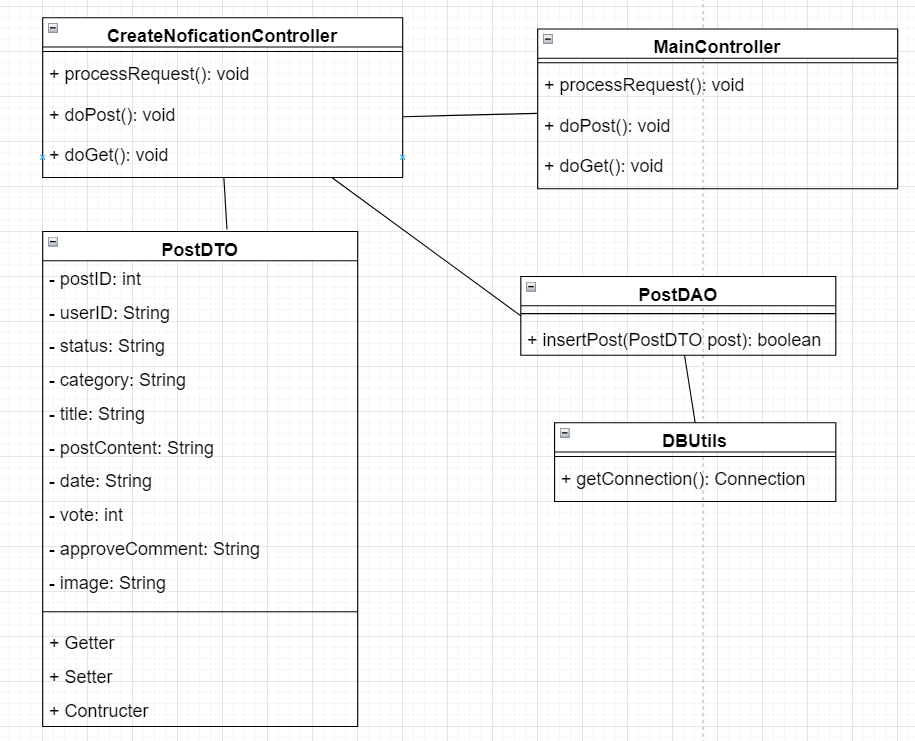
****

### c. Database queries

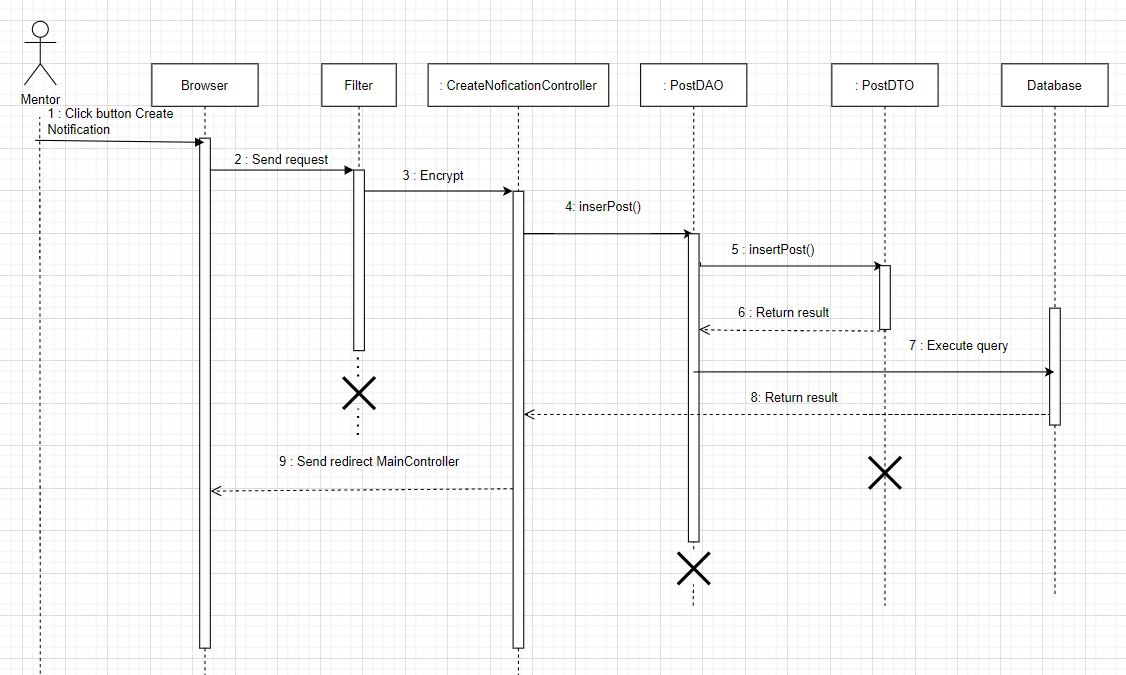
* INSERT INTO tblComment(postID, userID, date, commentContent) VALUES(?,?,?,?)

## 4. Create Notification

### a. Class Diagram



### b. Sequence Diagram(s)

****

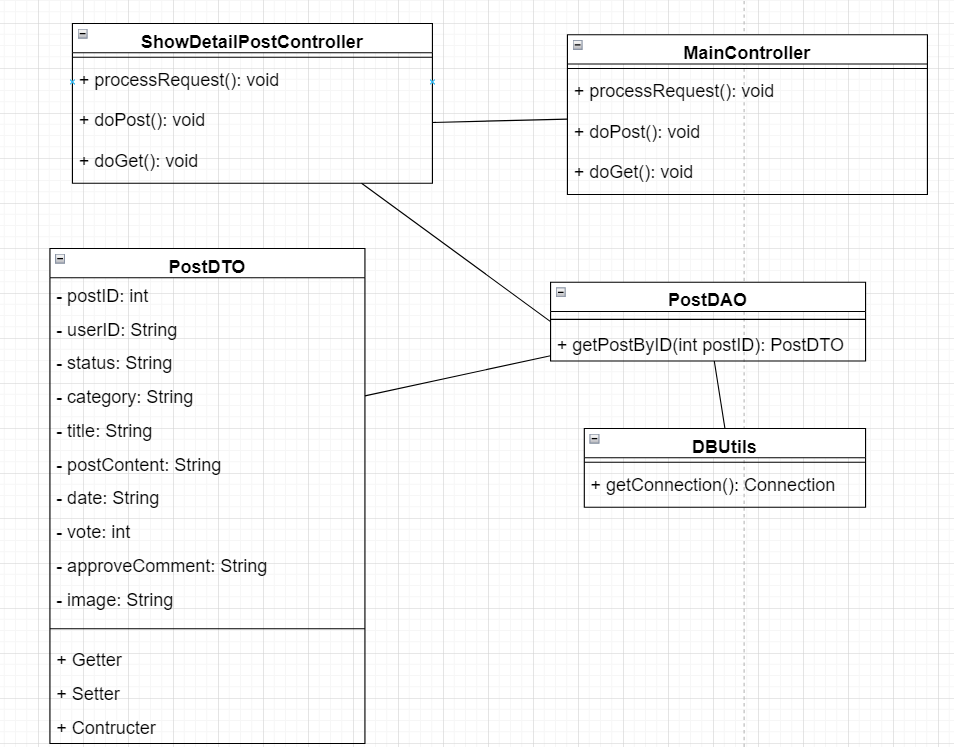
### c. Database queries

* Insert into tblPosts(userID, title, statusPID, categoryID, postContent ,date, vote, image )

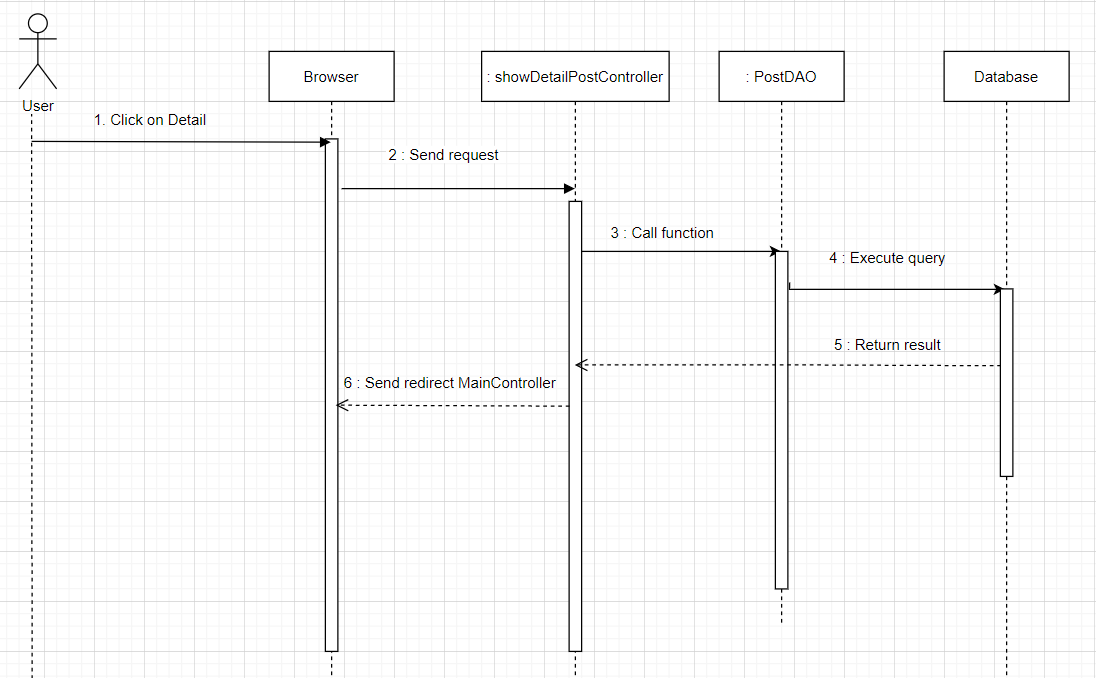
values(?,?,?,?,?,?,?,?)

## 5. Detail Post

### a. Class Diagram



### b. Sequence Diagram(s)

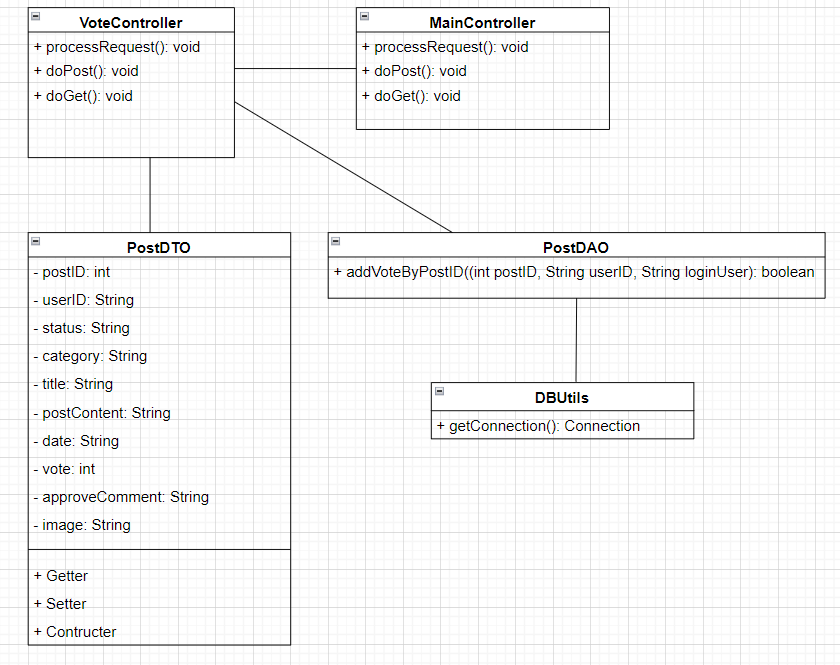
****

### c. Database queries

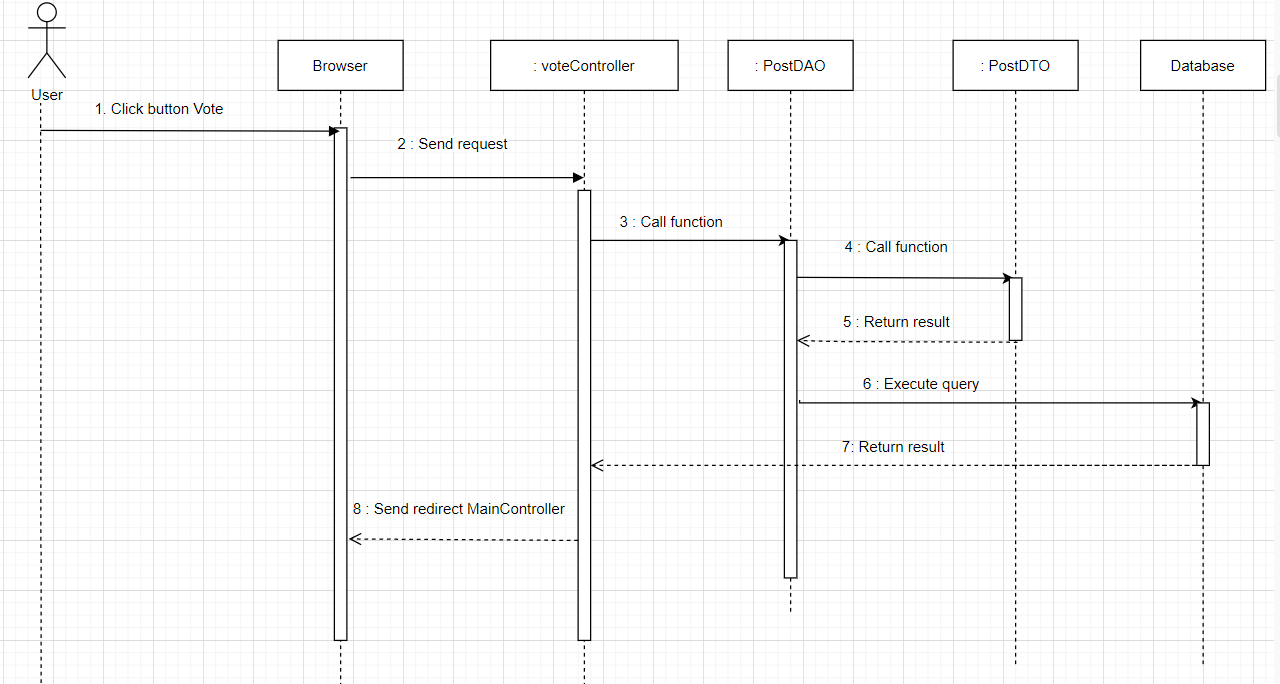
* select \* from tblPosts where postID=?

## 6. Vote

### a. Class Diagram



### b. Sequence Diagram(s)

****

### c. Database queries

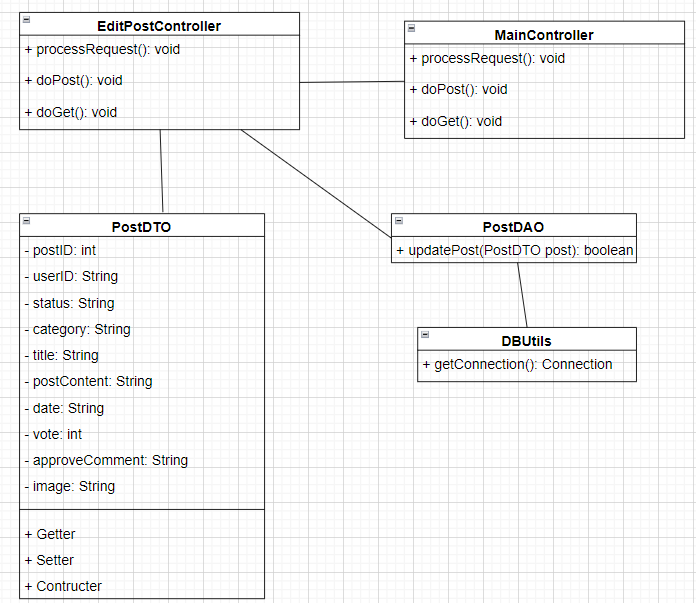
UPDATE tblPosts SET vote = vote + 1 WHERE postID = ?

UPDATE tblUsers SET totalVote = totalVote + 1 WHERE userID = ?

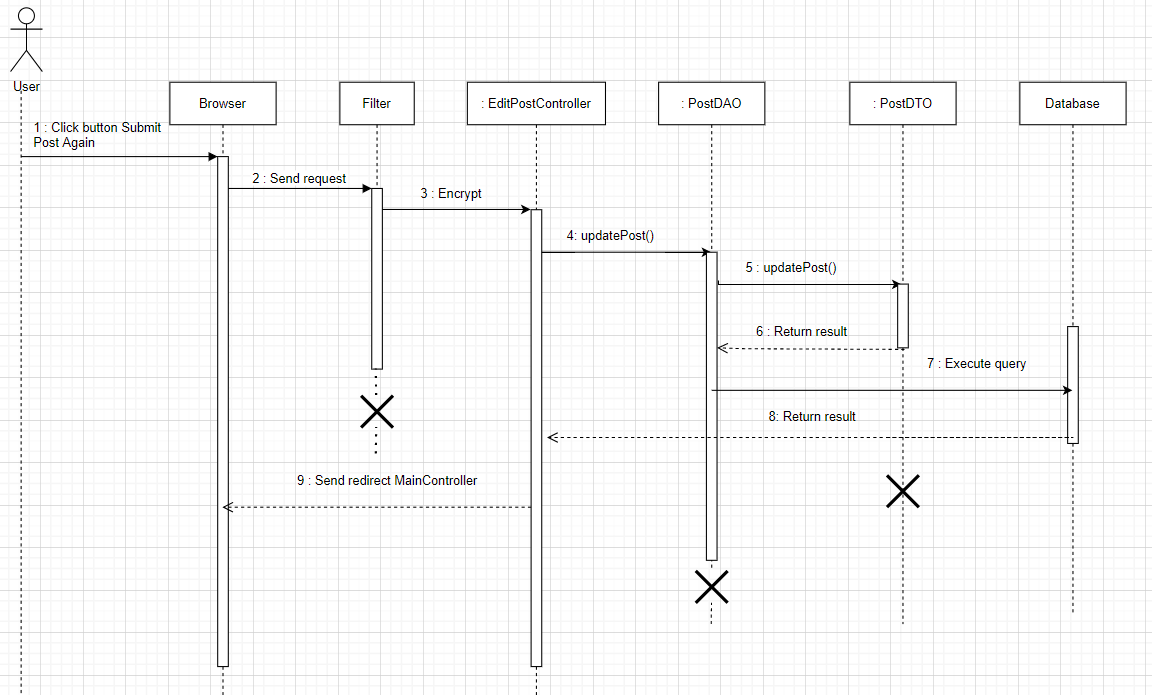
Insert into tblVote(postID, userID, date) values(?, ?, ?)

## 7. Edit Post

### a. Class Diagram

**

### b. Sequence Diagram(s)

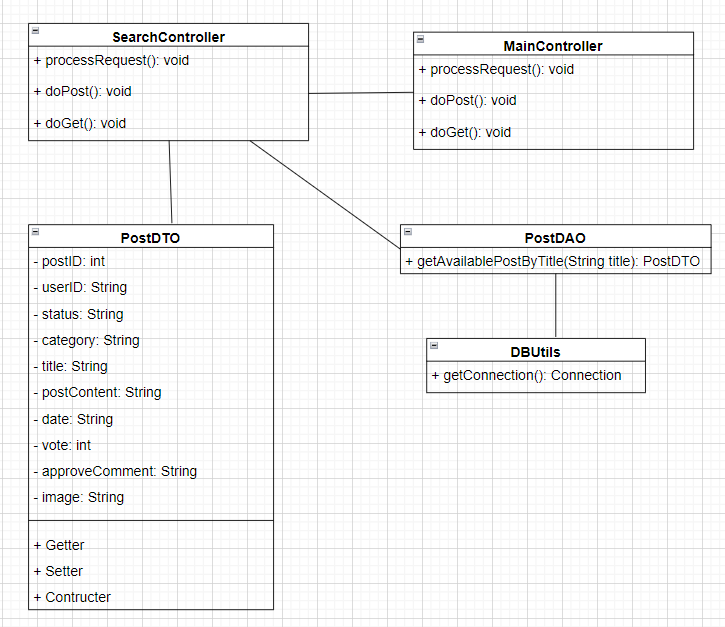
****

### c. Database queries

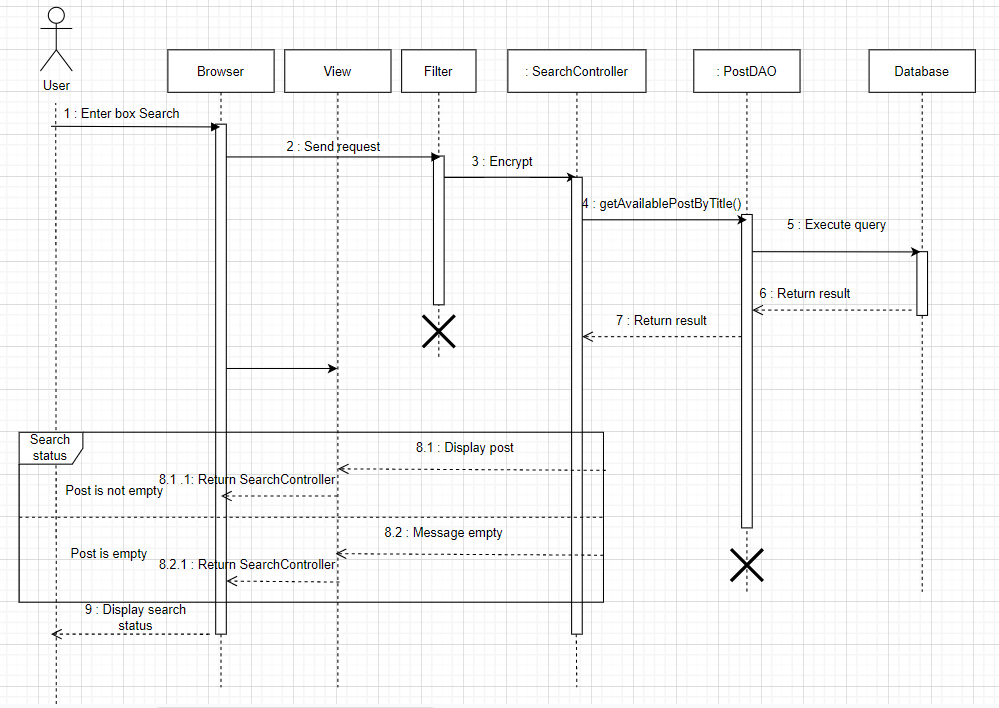
Update tblPosts Set userID=?, title=?, statusPID=?, categoryID=?, postContent=?, date=?, vote=?, image=?, approveComment=? Where postID=?

## 8. Search Post

### a. Class Diagram

**

### b. Sequence Diagram(s)

****

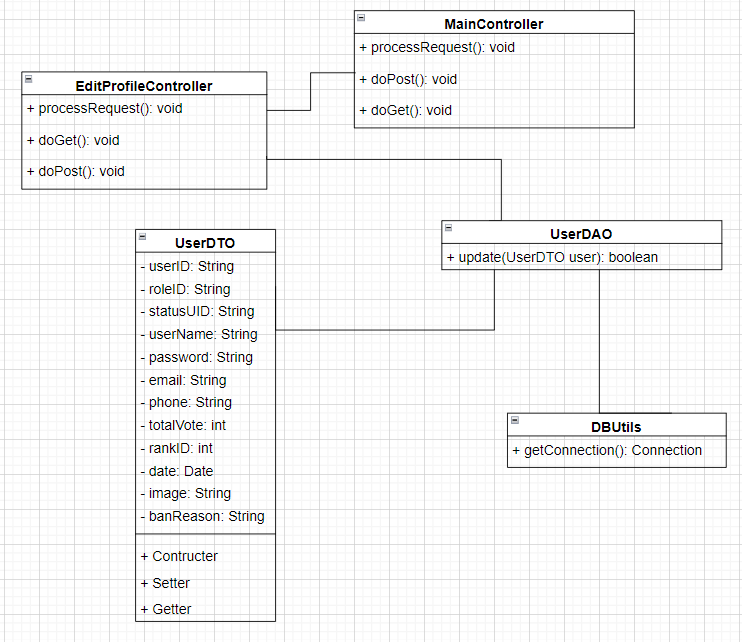
### c. Database queries

Select postID, userID, statusPID, categoryID, title, postContent, date, vote, image From tblPosts

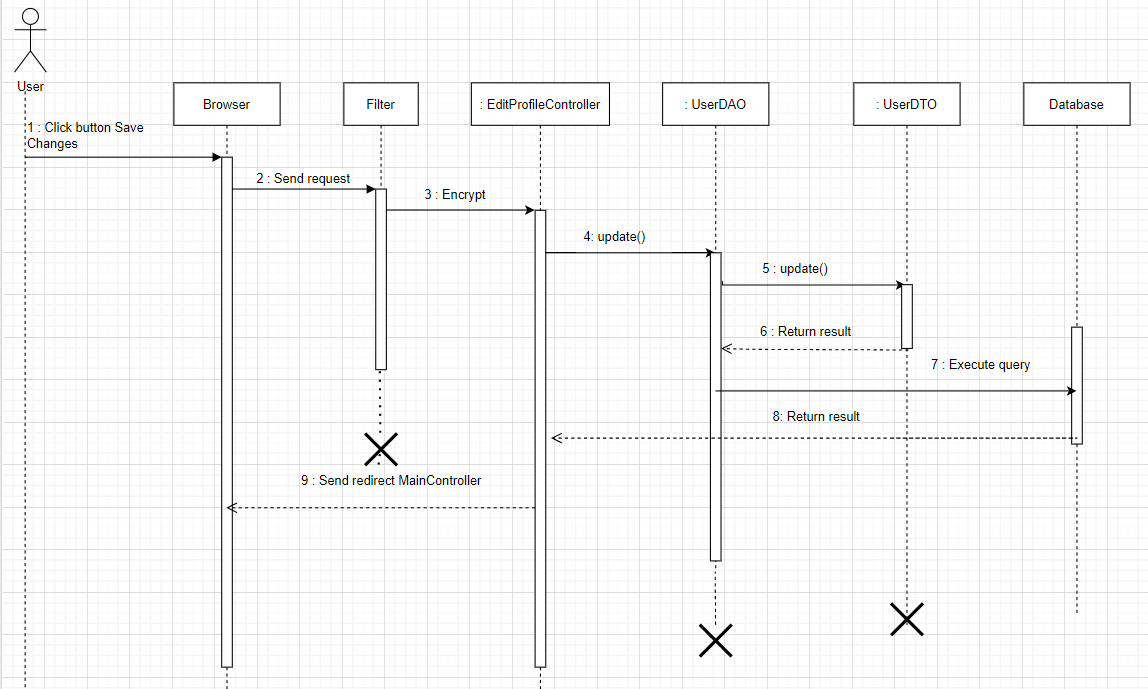
Where does the title LIKE ? and statusPID = ? order by date desc

## 9. Edit Profile

### a. Class Diagram

**

### b. Sequence Diagram(s)

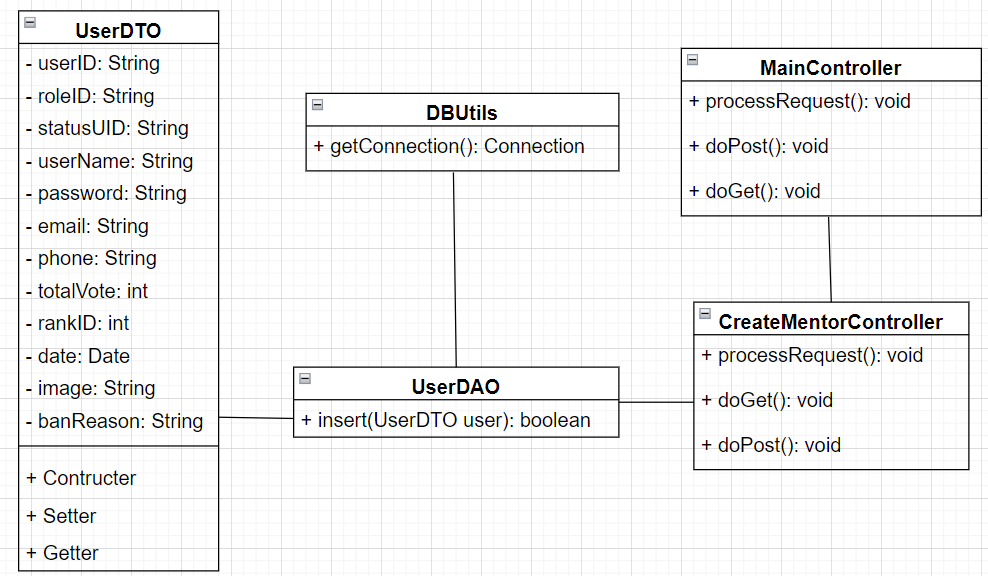
****

### c. Database queries

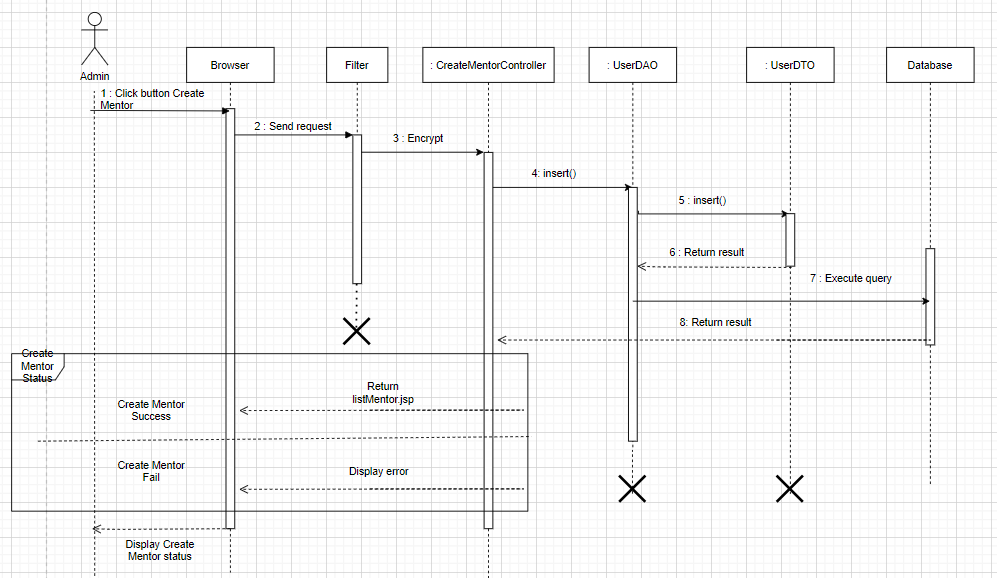
Update tblUsers set userName=?, password=?, email=?, phone=?, image=? Where userID=?

## 10. Create Mentor

### a. Class Diagram



### b. Sequence Diagram(s)

****

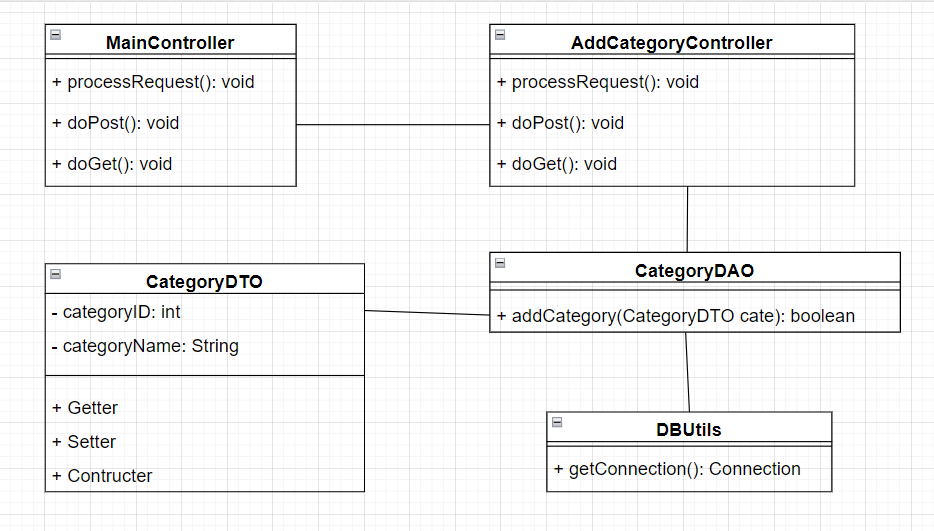
### c. Database queries

INSERT INTO tblUsers(userID, roleID, statusUID, userName, password, email, phone, totalVote, rankID, date, image)

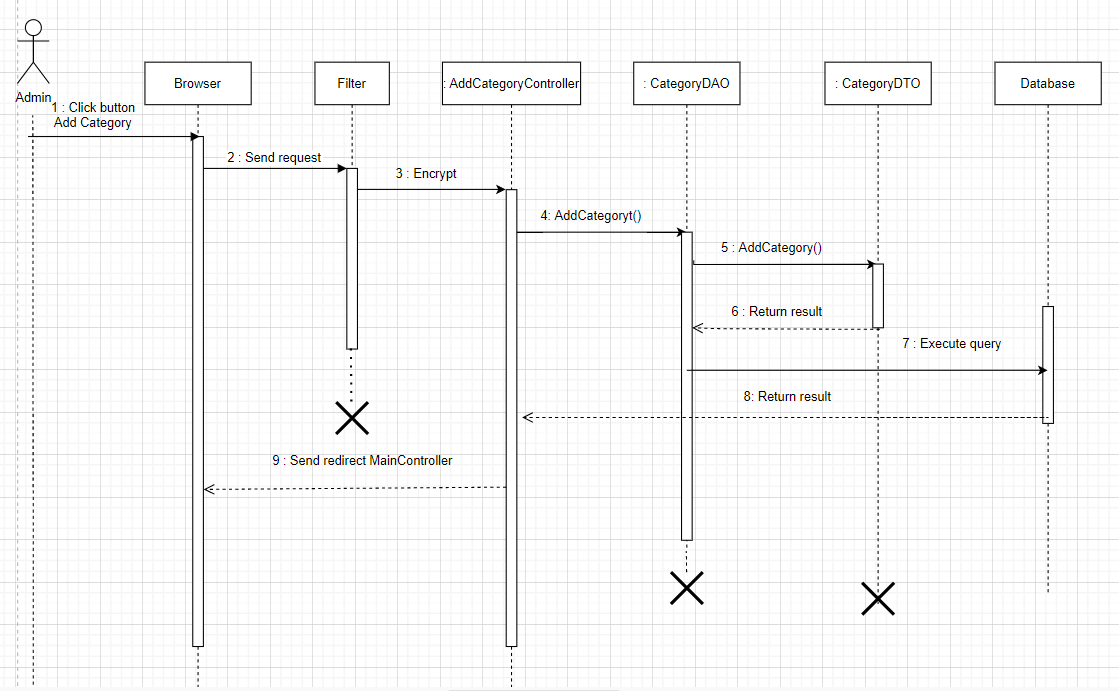
VALUES(?,?,?,?,?,?,?,?,?,?,?)

## 11. Add Category

### a. Class Diagram



### b. Sequence Diagram(s)

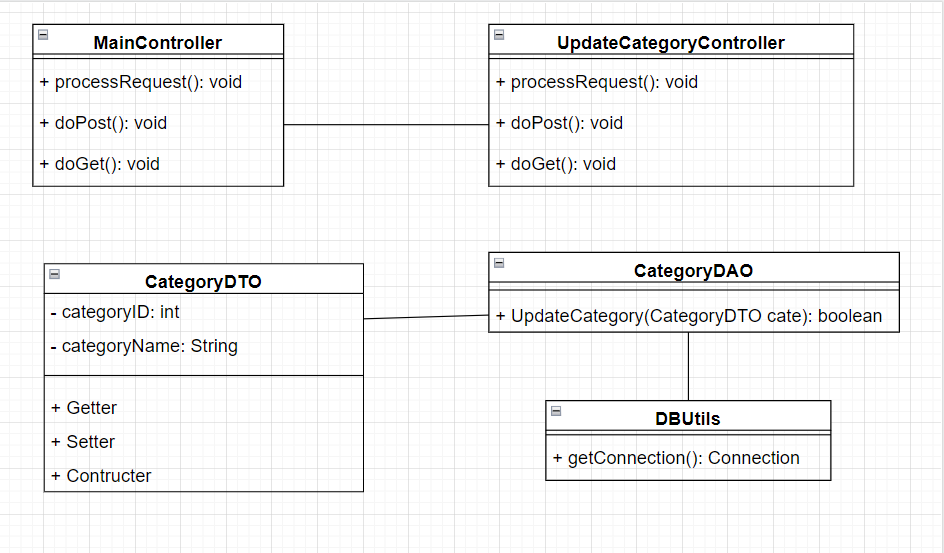
****

### c. Database queries

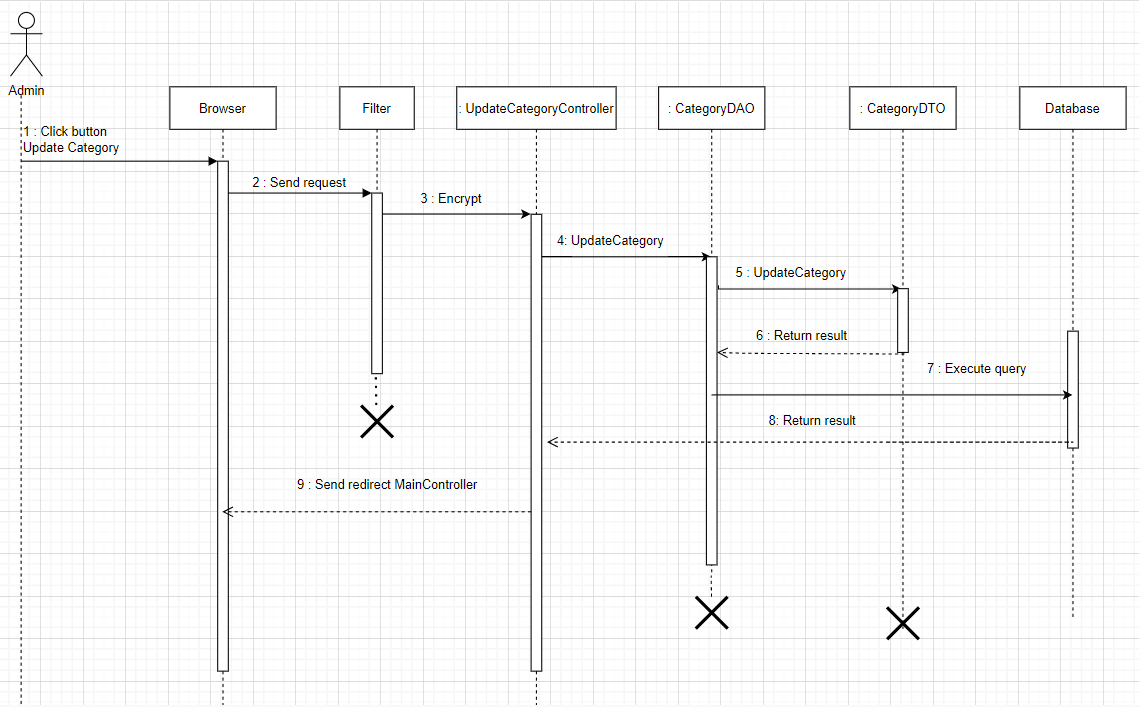
INSERT INTO tblCategories(categoryName) VALUES(?)

## 12. Update Category

### a. Class Diagram



### b. Sequence Diagram(s)

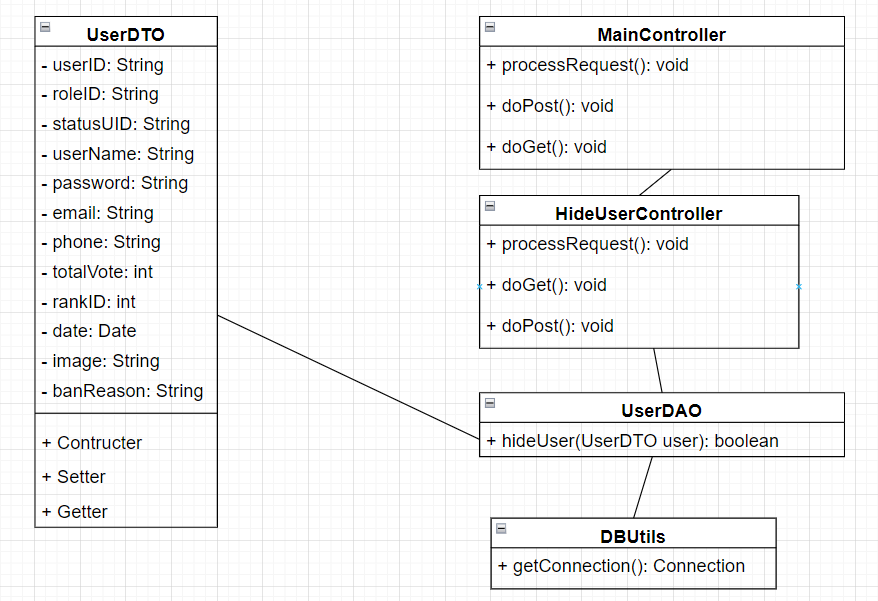
****

### c. Database queries

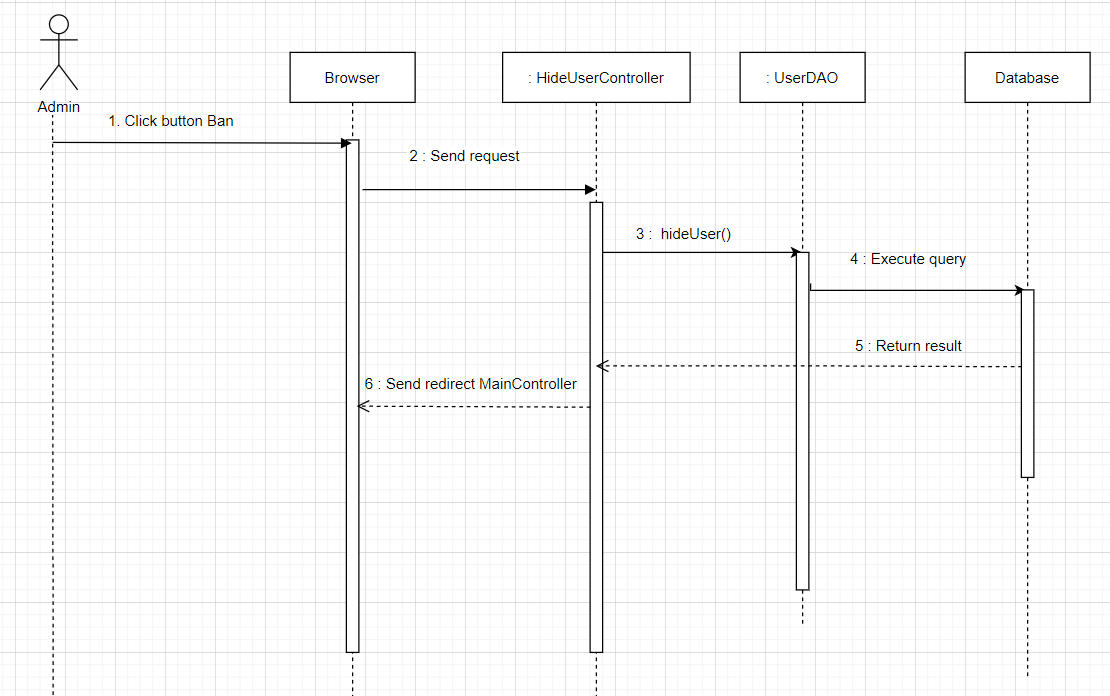
update tblCategories set categoryName = ? where categoryID = ?

## 13. Ban User

### a. Class Diagram



### b. Sequence Diagram(s)

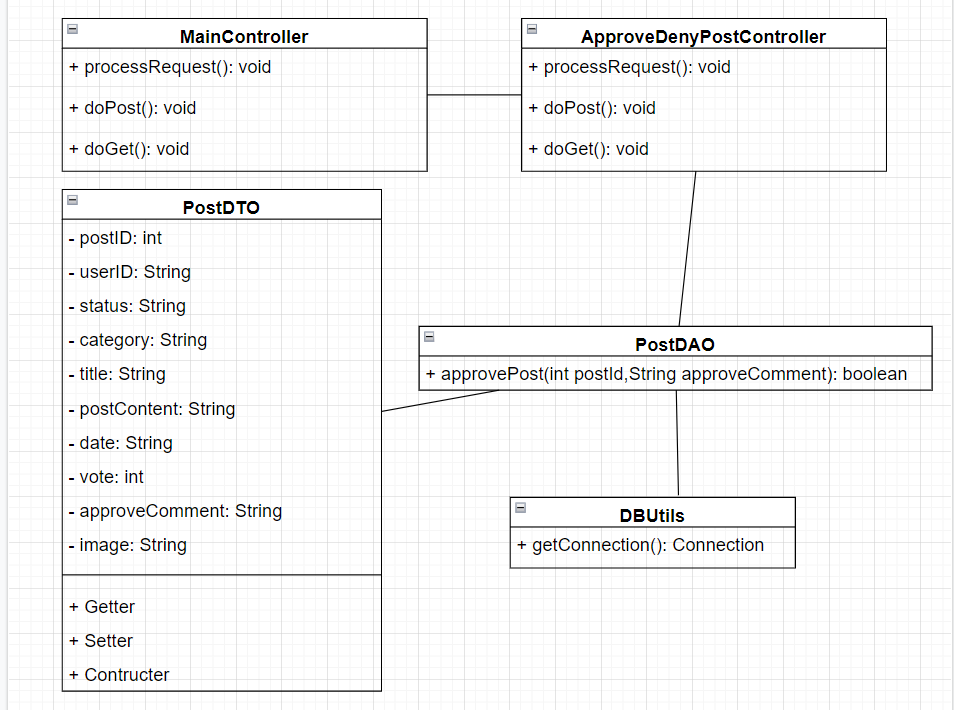
****

### c. Database queries

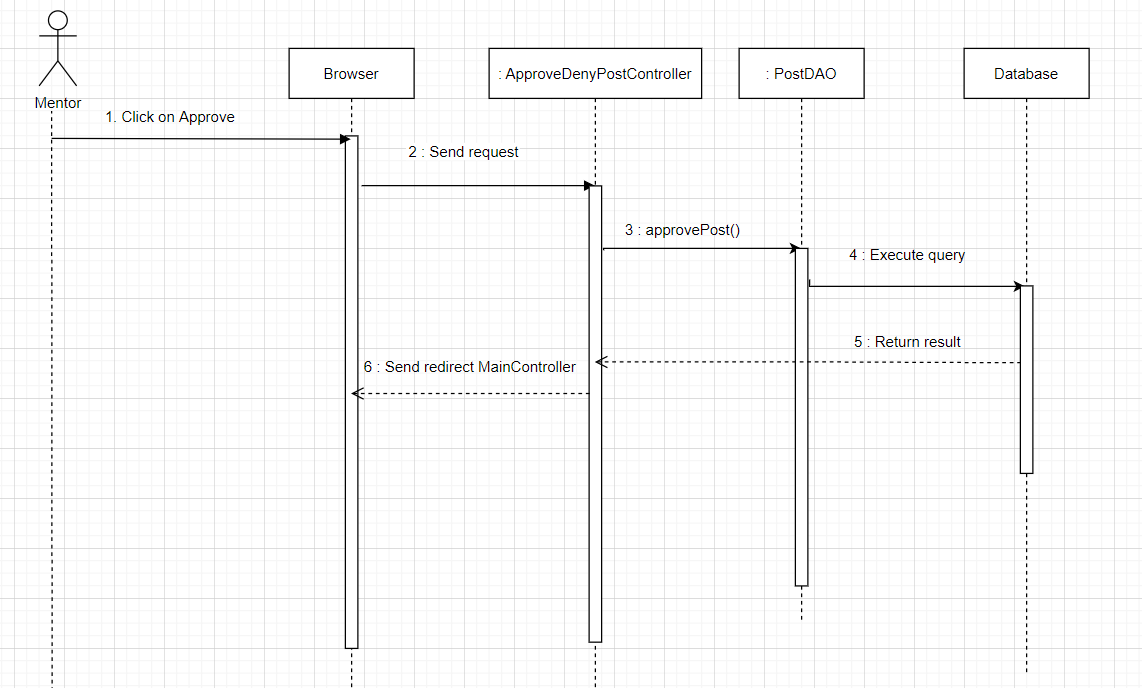
UPDATE tblUsers SET statusUID='0', banReason = ? WHERE userID=?

## 14. Approve Post

### a. Class Diagram



### b. Sequence Diagram(s)

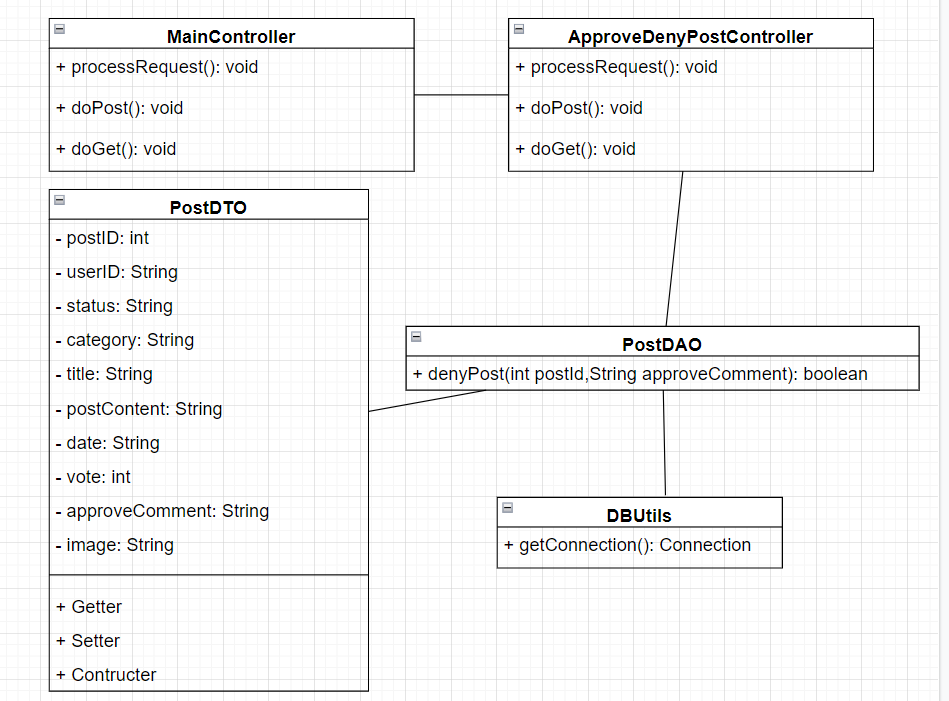
****

### c. Database queries

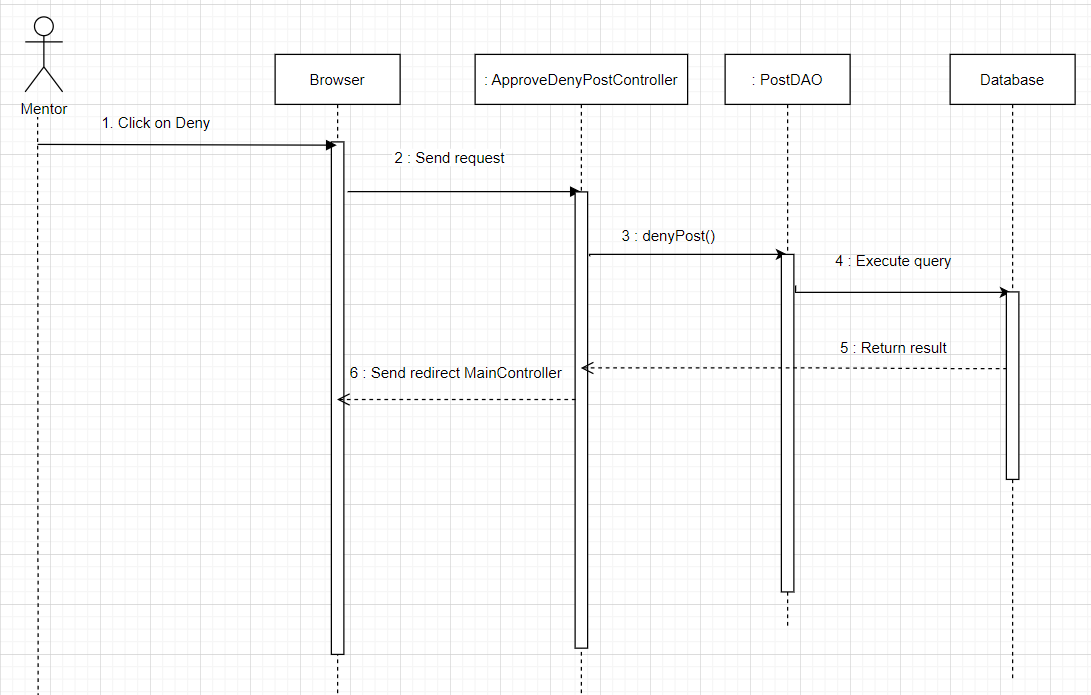
Update tblPosts set statusPID=1, approveComment=? where postID=?

## 15. Deny Post

### a. Class Diagram



### b. Sequence Diagram(s)

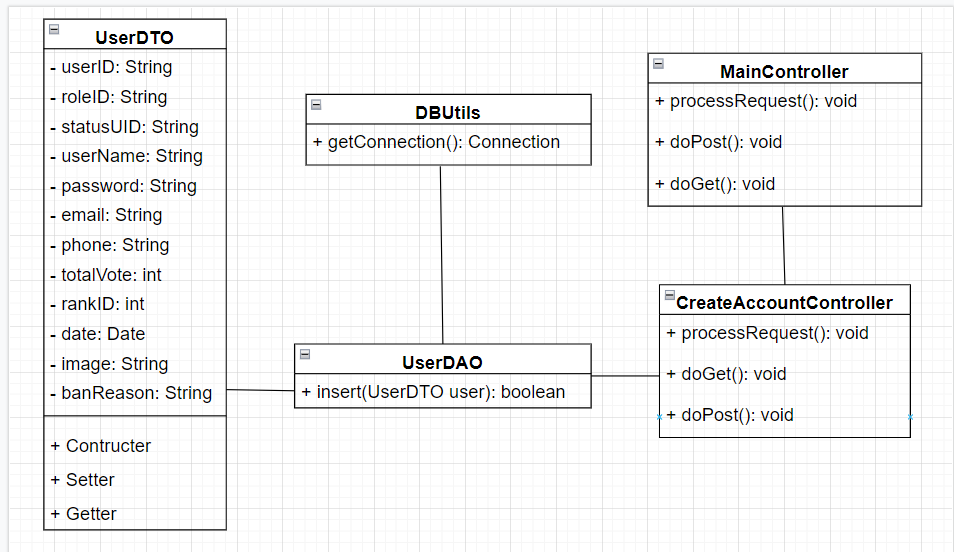
****

### c. Database queries

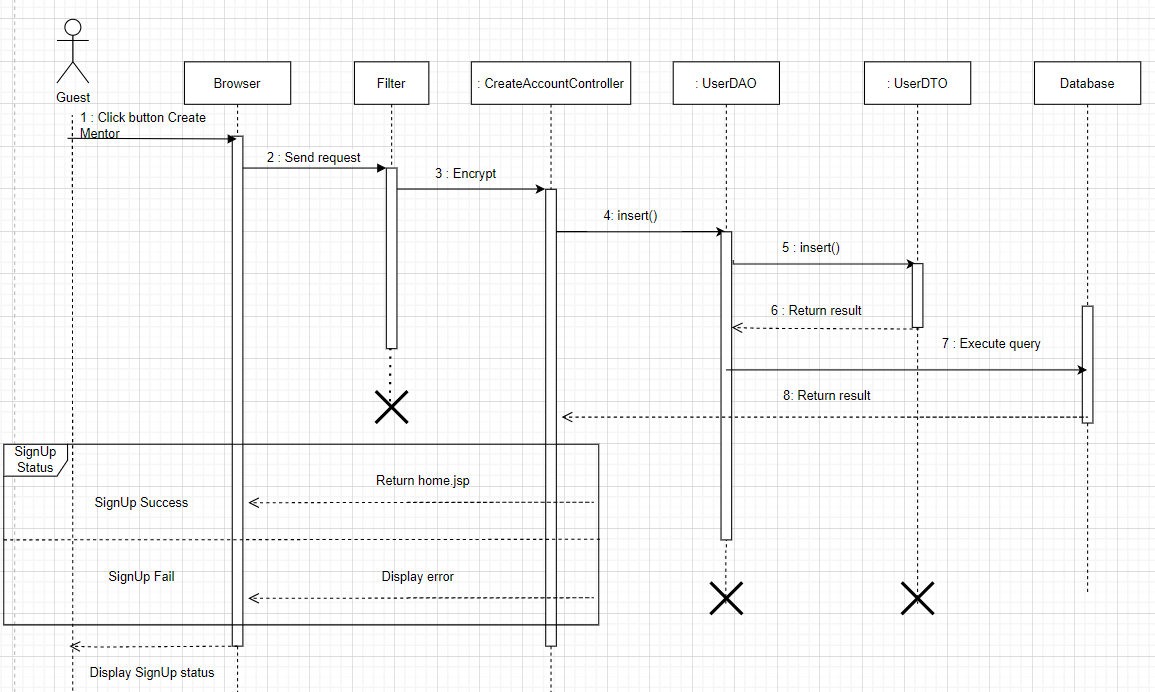
Update tblPosts set statusPID=0, approveComment=? where postID=?

## 16. Sign up

### a. Class Diagram



### b. Sequence Diagram(s)

****

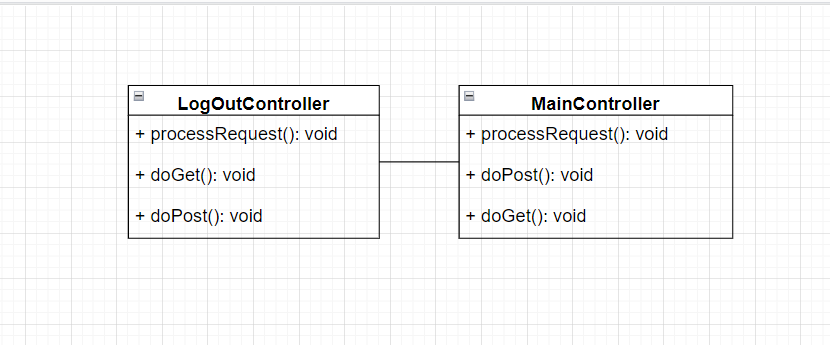
### c. Database queries

INSERT INTO tblUsers(userID, roleID, statusUID, userName, password, email, phone, totalVote, rankID, date, image)

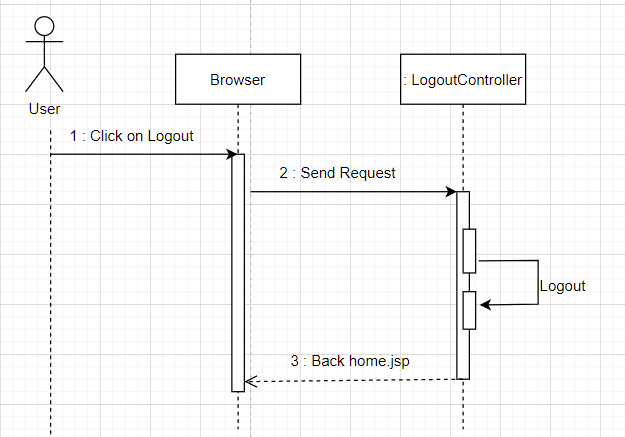
VALUES(?,?,?,?,?,?,?,?,?,?,?)

## 17. Logout

### a. Class Diagram



### b. Sequence Diagram(s)

****

### 