

**SOCIAL CELL NETWORK**

**Software Design Specification**

– Hanoi, January 2024 –

**RECORD OF CHANGES**

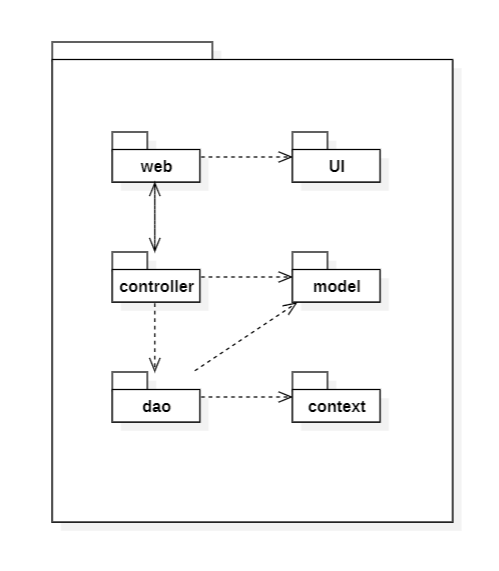
\*A - ADDED M - MODIFIED D - DELETED

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  | Phạm Hồng Phong |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

# **I. Overview**

## **1. Code Packages/Namespaces**



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

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | **controller** | Access users’ requests through the interface and process them by using package dao’s files’ functionalities |
| *02* | **dao** | Access and modify website’s objects’ data |
| *03* | **web** | Contains files that illustrate screen pages in website |
| *04* | **context** | Contains methods that initial connection to the database |
| *05* | **model** | Contains classes with properties and behaviors that describe objects in the website |
| *06* | **UI** | Contains web resources related to view |

## **2. Coding Conventions**

**Chosen language:** Java

1. **Naming:**

* Names of classes,interfaces, files have to be without space, have a capitalised letter in the beginning of words with the exception of variables and methods and describe their functionality. For example , the method used to insert a new user into database should be named insertUser
* Name of classes should be nouns, and of methods should be verbs. Example:

class User;

public void updateUser();

1. **Package and Import Statements:**

* The first non-comment line of most Java source files is a package statement. After that, import statements can follow. For example:

package java.awt;

import java.awt.peer.CanvasPeer;

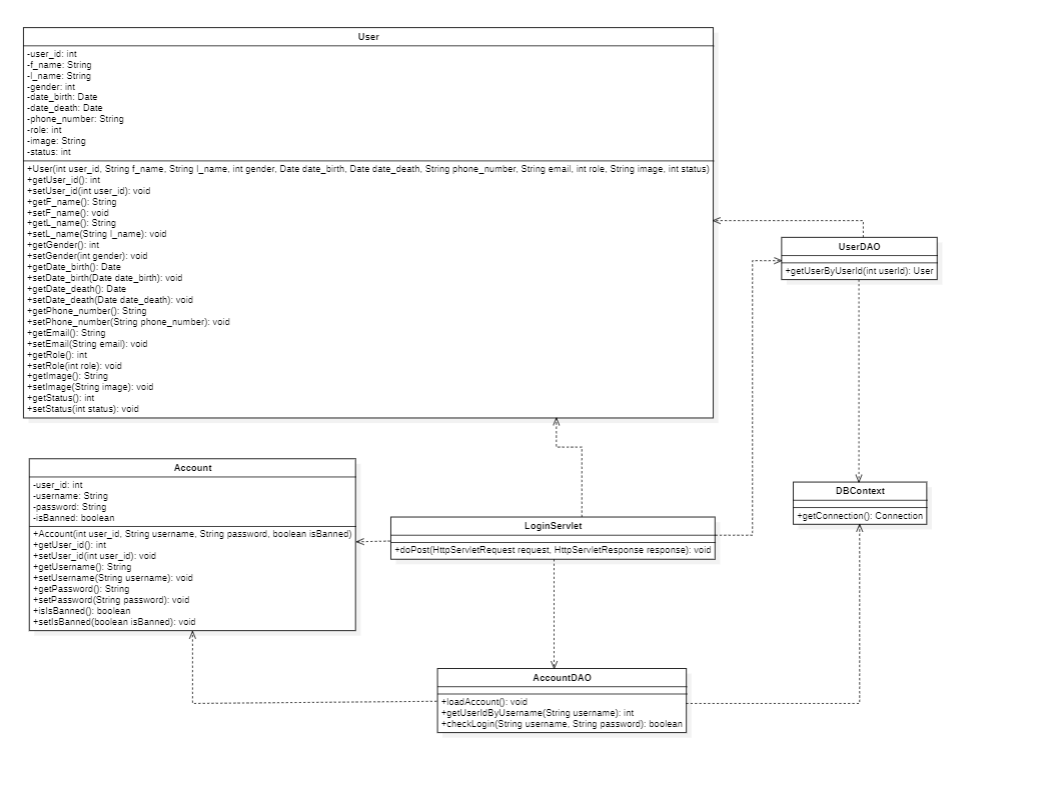
1. **Line length:**

* Avoid writing lines too long, since they’re not handled well by many terminals and tools, preferably before the vertical line in the IDE.

# **II. Code Designs**

## **Login**

### **a.Class Diagram**



### **b.Class Specifications**

##### ***Account***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *setUsername* | *Set the new value for username field* |
| *02* | *getUsername* | *get value of field username* |
| *03* | *getUser\_Id* | *get value of field id* |
| *04* | *setUser\_Id* | *set value of field id* |
| *05* | *getPassword* | *get value of field password* |
| *06* | *setPassword* | *set value of field password* |

##### ***DBContext***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### 

##### ***LoginServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *FindUsername* | *get request from UI and send username to DAO to search whether there’s an user with username inputted* |
| *02* | *checkLogin* | *get request from UI and send username and password to DAO to find user, if there is a user, set the return value to session scope* |

##### ***AccountDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUserIdByUsername* | *execute sql query to find user with given username* |
| *02* | *checkLogin* | *use the boolean function to check in accList to check if the username and password already exist* |

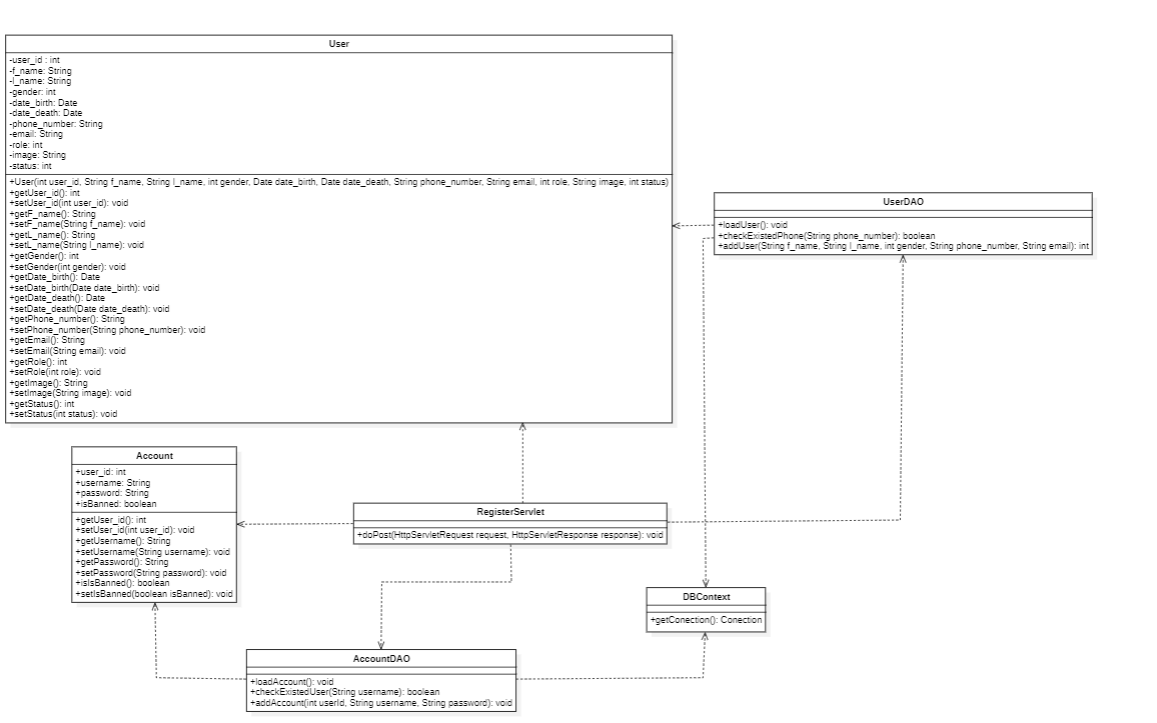
### **c.Sequence Diagram(s)**

### **d.Database queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *getUserIdByUsername* | *SELECT user\_id FROM account WHERE username = ‘@username’* |

## **Signup**

### **a.Class Diagram**



### **b.Class Specifications**

##### ***User***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUser\_Id* | *get value of field user\_id* |
| *02* | *setUser\_Id* | *set value of field user\_id* |
| *03* | *getF\_name* | *get value of field f\_name* |
| *04* | *setF\_name* | *set value of field f\_name* |
| *05* | *getL\_name* | *get value of field L\_name* |
| *06* | *setL\_name* | *set value of field L\_name* |
| *07* | *getGender* | *get value of field gender* |
| *08* | *setGender* | *set value of field gender* |
| *09* | *getDate\_birth* | *get value of field date\_birth* |
| *10* | *setDate\_birth* | *set value of field date\_birth* |
| *11* | *getDate\_death* | *get value of field date\_death* |
| *12* | *setDate\_death* | *set value of field date\_death* |
| *13* | *getPhone\_number* | *get value of field phone\_number* |
| *14* | *setPhone\_number* | *set value of field phone\_number* |
| *15* | *getEmail* | *get value of field email* |
| *16* | *setEmail* | *set value of field email* |
| *17* | *getRole* | *get value of field role* |
| *18* | *setRole* | *set value of field role* |
| *19* | *getImage* | *get value of field image* |
| *20* | *setImage* | *set value of field image* |
| *21* | *getStatus* | *get value of field status* |
| *22* | *setStatus* | *set value of field status* |

##### ***Account***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *setUsername* | *Set the new value for username field* |
| *02* | *getUsername* | *get value of field username* |
| *03* | *getUser\_Id* | *get value of field id* |
| *04* | *setUser\_Id* | *set value of field id* |
| *05* | *getPassword* | *get value of field password* |
| *06* | *setPassword* | *set value of field password* |

##### ***DBContext***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***RegisterServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *addUser* | *get request from UI and send inputted data to DAO to add information user* |
| *02* | *addAccount* | *get request from UI and send inputted data to DAO to register account* |

##### ***AccountDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkExistedUser* | *get request from UI and send username to DAO to to check if the username already exist* |
| *02* | *checkExistedPhone* | *get request from UI and send phone number to DAO to to check if the phone number already exist* |
| *03* | *addUser* | *get request from UI and send inputted data to DAO to add information user* |
| *04* | *addAccount* | *get request from UI and send inputted data to DAO to register account* |

### **c.Sequence Diagram(s)**

### **d.Database queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *addUser* | *INSERT INTO `social`.`user` (`first\_name`, `last\_name`, `gender`, `phone\_number`,`email`, `role`, `status`) VALUES (@first\_name, @last\_name, @gender, @phone\_number, @email, 0, 0)* |
| *02* | *addAccount* | *INSERT INTO `social`.`account` (`user\_id`, `username`, `password`, `is\_banned`) VALUES (@user\_id, @username, @password, 0)* |

## 

## **Change password**

## **Forgot password**

## **Logout**

### **a.Class Diagram**



### **b.Class Specifications**

##### ***LogoutServlet***

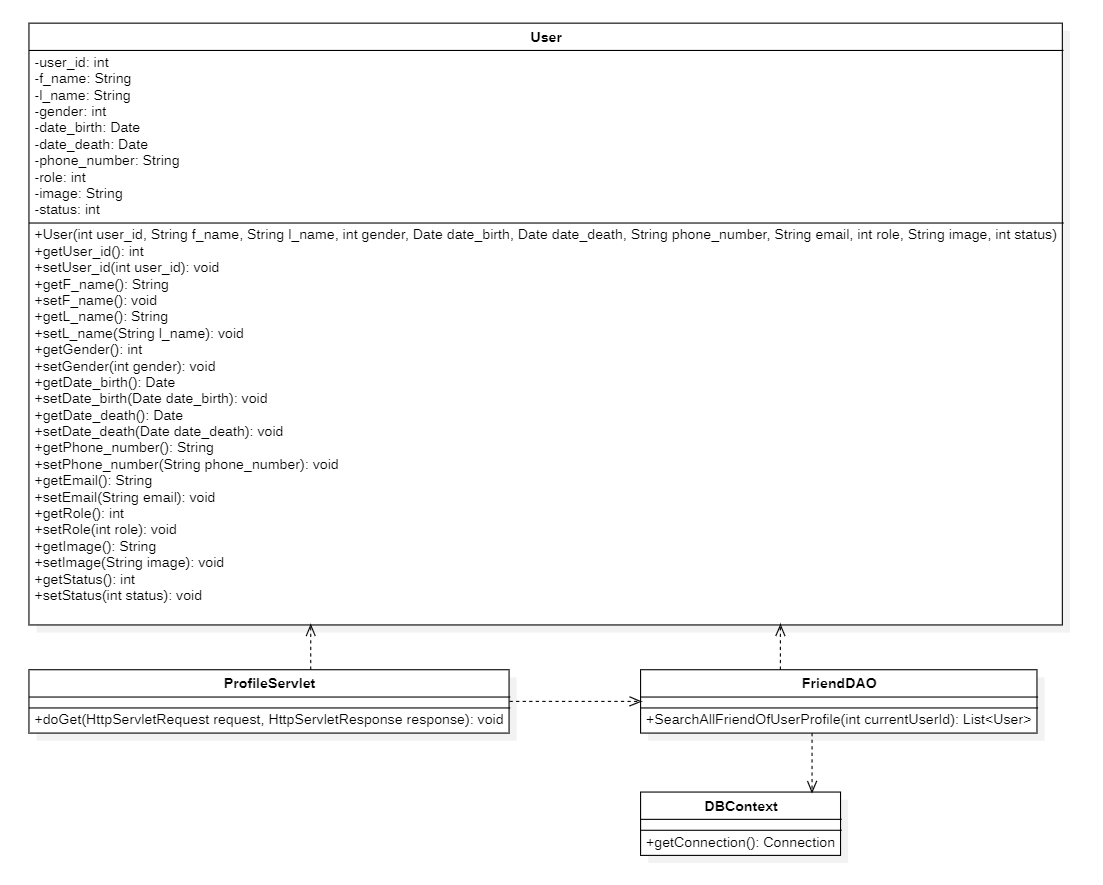
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *Remove all current session* |

### **c.Sequence Diagram(s)**

## **Edit User Profile**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### 

##### ***User***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUser\_Id* | *get value of field user\_id* |
| *02* | *setUser\_Id* | *set value of field user\_id* |
| *03* | *getF\_name* | *get value of field f\_name* |
| *04* | *setF\_name* | *set value of field f\_name* |
| *05* | *getL\_name* | *get value of field L\_name* |
| *06* | *setL\_name* | *set value of field L\_name* |
| *07* | *getGender* | *get value of field gender* |
| *08* | *setGender* | *set value of field gender* |
| *09* | *getDate\_birth* | *get value of field date\_birth* |
| *10* | *setDate\_birth* | *set value of field date\_birth* |
| *11* | *getDate\_death* | *get value of field date\_death* |
| *12* | *setDate\_death* | *set value of field date\_death* |
| *13* | *getPhone\_number* | *get value of field phone\_number* |
| *14* | *setPhone\_number* | *set value of field phone\_number* |
| *15* | *getEmail* | *get value of field email* |
| *16* | *setEmail* | *set value of field email* |
| *17* | *getRole* | *get value of field role* |
| *18* | *setRole* | *set value of field role* |
| *19* | *getImage* | *get value of field image* |
| *20* | *setImage* | *set value of field image* |
| *21* | *getStatus* | *get value of field status* |
| *22* | *setStatus* | *set value of field status* |

##### ***UserDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updateUser* | *Update new values for user* |

##### ***ProfileServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doPost* | *get request from UI and send inputted data to DAO to update user info* |

### 

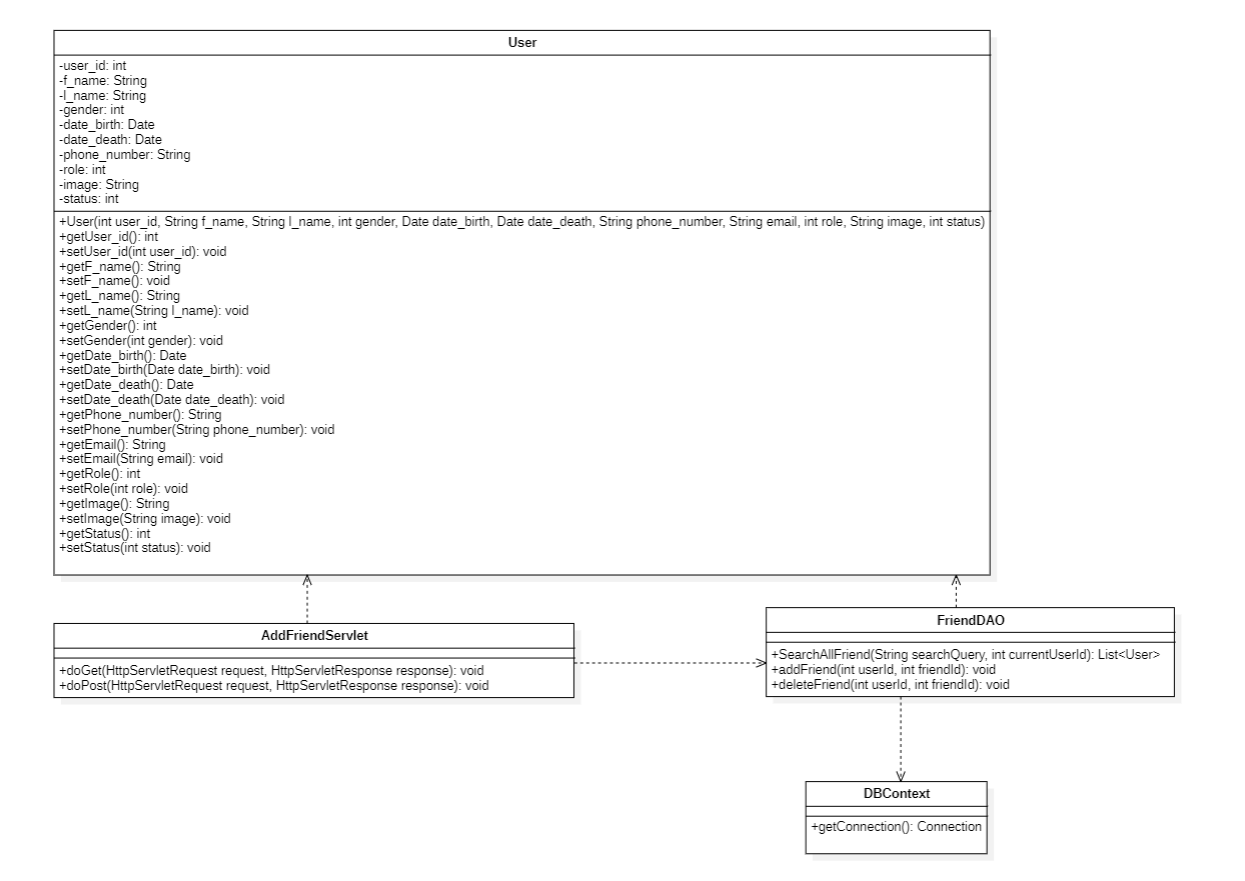
### **c.Sequence Diagram**

### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *updateUser* | *UPDATE `social`.`user` SET `first\_name` = @first\_name, `last\_name` = @last\_name, `gender` = @gender, `date\_birth` = @date\_birth, `phone\_number` = @phone\_number, `email` = @email, `role` = 0, `image`= @image WHERE `user\_id` = @user\_id* |

## **Add friend**

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***User***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getUser\_Id* | *get value of field user\_id* |
| *02* | *setUser\_Id* | *set value of field user\_id* |
| *03* | *getF\_name* | *get value of field f\_name* |
| *04* | *setF\_name* | *set value of field f\_name* |
| *05* | *getL\_name* | *get value of field L\_name* |
| *06* | *setL\_name* | *set value of field L\_name* |
| *07* | *getGender* | *get value of field gender* |
| *08* | *setGender* | *set value of field gender* |
| *09* | *getDate\_birth* | *get value of field date\_birth* |
| *10* | *setDate\_birth* | *set value of field date\_birth* |
| *11* | *getDate\_death* | *get value of field date\_death* |
| *12* | *setDate\_death* | *set value of field date\_death* |
| *13* | *getPhone\_number* | *get value of field phone\_number* |
| *14* | *setPhone\_number* | *set value of field phone\_number* |
| *15* | *getEmail* | *get value of field email* |
| *16* | *setEmail* | *set value of field email* |
| *17* | *getRole* | *get value of field role* |
| *18* | *setRole* | *set value of field role* |
| *19* | *getImage* | *get value of field image* |
| *20* | *setImage* | *set value of field image* |
| *21* | *getStatus* | *get value of field status* |
| *22* | *setStatus* | *set value of field status* |

##### ***Relation***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getRelation\_id* | *get value of field Relation\_id* |
| *02* | *setRelation\_id* | *set value of field Relation\_id* |
| *03* | *getUser1\_id* | *get value of field User1\_id* |
| *04* | *setUser1\_id* | *set value of field User1\_id* |
| *05* | *getUser2\_id* | *get value of field User2\_id* |
| *06* | *setUser2\_id* | *set value of field User2\_id* |
| *07* | *getType* | *get value of field Type* |
| *08* | *setType* | *set value of field Type* |

##### ***FriendDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *SearchAllFriend* | *Search friend* |
| *02* | *addFriend* | *Add new user to friend* |
| *03* | *deleteFriend* | *Delete friend in user’s list* |

##### ***AddFriendServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doPost* | *get request from UI and add/delete friend* |
| *02* | *doGet* | *get request from UI and search friend* |

### 

### **c.Sequence Diagram**

### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *SearchAllFriend* | *SELECT \* FROM user WHERE (first\_name LIKE @first\_name OR last\_name LIKE @last\_name ) AND user\_id != @user\_id* |
| *02* | *addFriend* | *INSERT INTO relation (user1\_id, user2\_id, type) VALUES (@user1\_id, @user2\_id, 1) ON DUPLICATE KEY UPDATE type=VALUES(type)* |
| *03* | *deleteFriend* | *DELETE FROM relation WHERE (user1\_id = @user1\_id AND user2\_id = @user2\_id ) OR (user1\_id = @user1\_id AND user2\_id = @user2\_id)* |

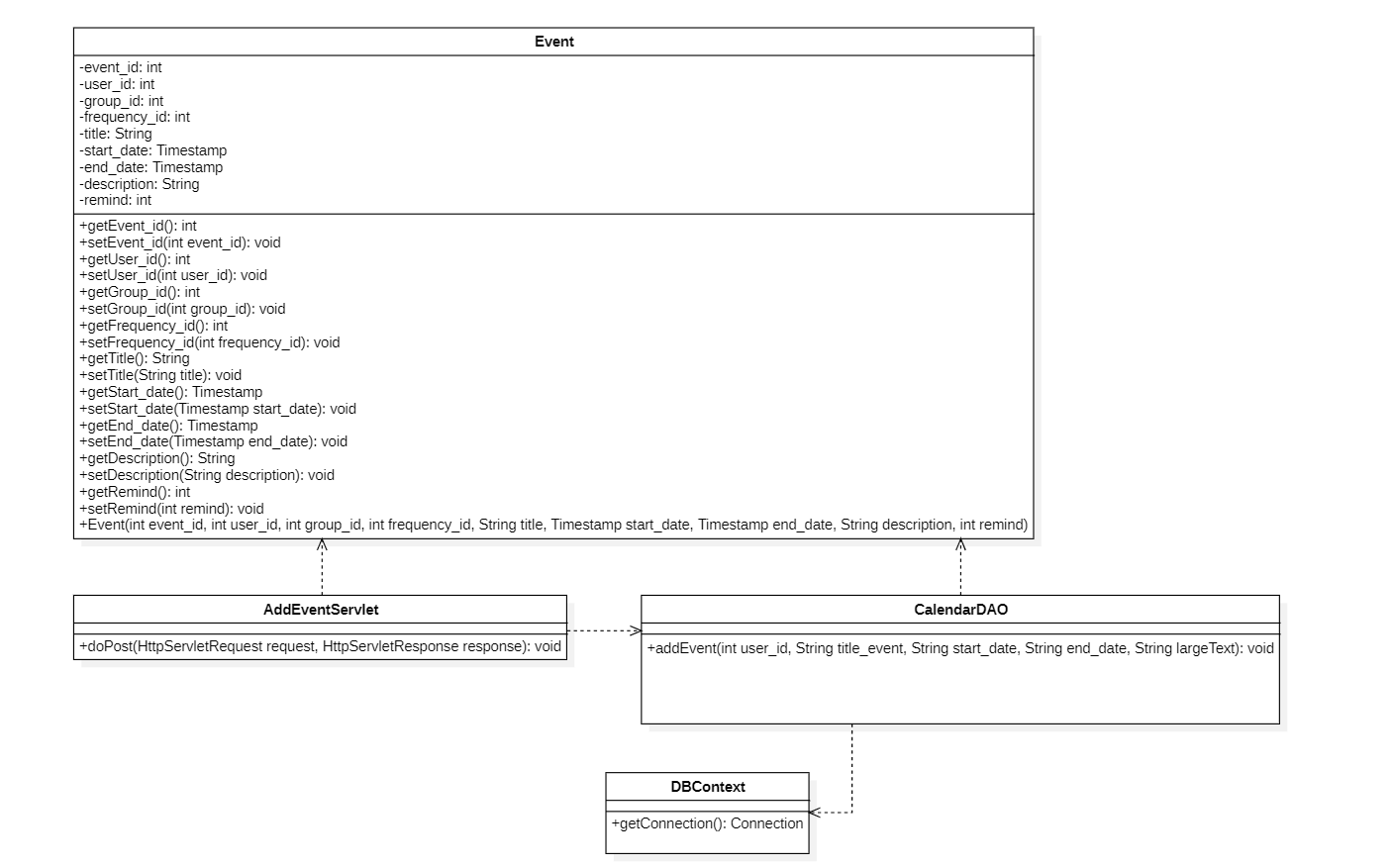
## 

## 

## 

## **Add event**

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***Event***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getEvent\_id* | *get value of field Event\_id* |
| *02* | *setEvent\_id* | *set value of field Event\_id* |
| *03* | *getUser\_id* | *get value of field User\_id* |
| *04* | *setUser\_id* | *set value of field User\_id* |
| *05* | *getGroup\_id* | *get value of field Group\_id* |
| *06* | *setGroup\_id* | *set value of field Group\_id* |
| *07* | *getFrequency\_id* | *get value of field Frequency\_id* |
| *08* | *setFrequency\_id* | *set value of field Frequency\_id* |
| *09* | *getTitle* | *get value of field Title* |
| *10* | *setTitle* | *set value of field Title* |
| *11* | *getStart\_date* | *get value of field Start\_date* |
| *12* | *setStart\_date* | *set value of field Start\_date* |
| *13* | *getEnd\_date* | *get value of field End\_date* |
| *14* | *setEnd\_date* | *set value of field End\_date* |
| *15* | *getDescription* | *get value of field Description* |
| *16* | *setDescription* | *set value of field Description* |
| *17* | *getRemid* | *get value of field Remid* |
| *18* | *setRemid* | *set value of field Remid* |

##### ***CalendarDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *addEvent* | *Add new event of user in calendar* |

##### ***AddEventServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doPost* | *get request from UI and add events of user in calendar at home page* |

### 

### **c.Sequence Diagram**

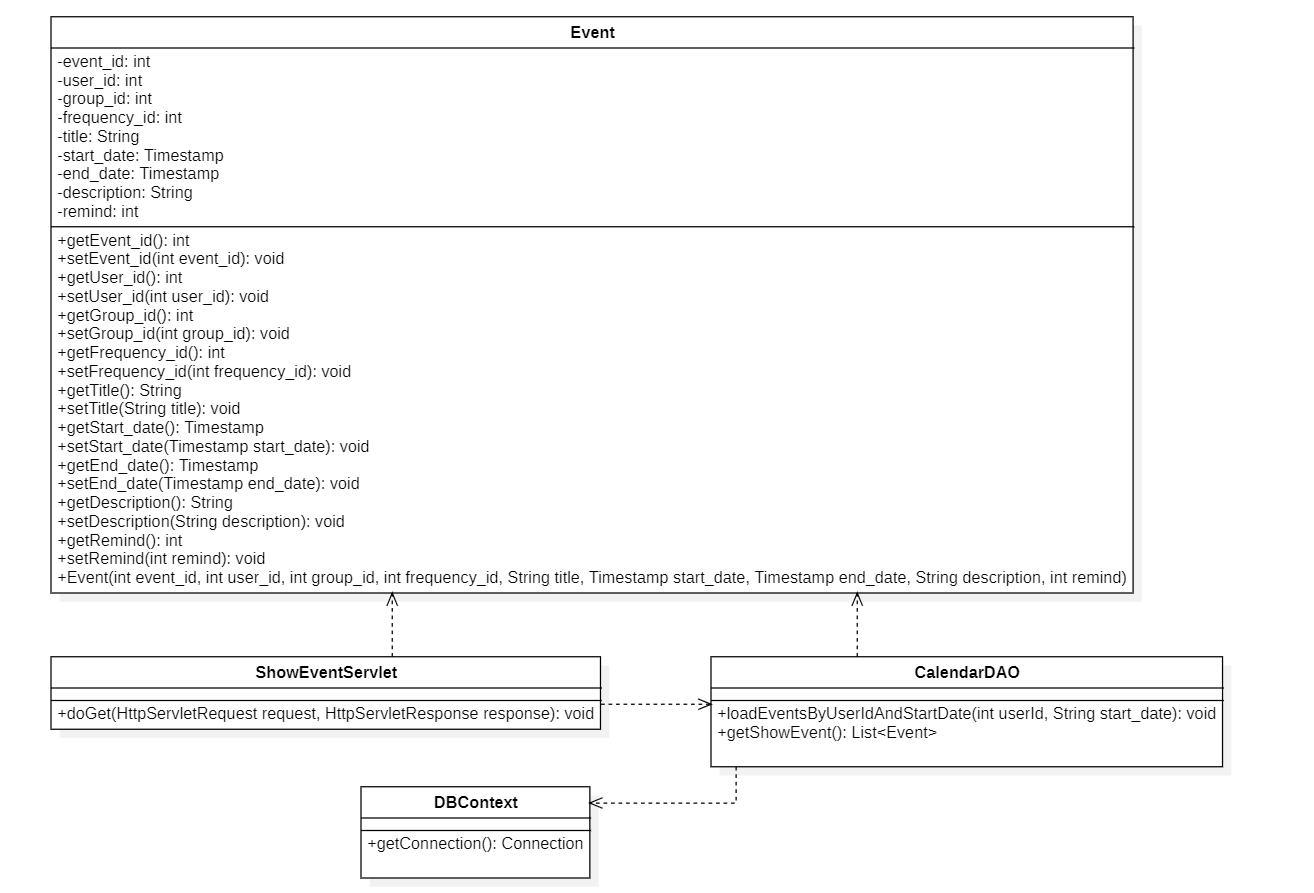
### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *addEvent* | *INSERT INTO `social`.`event` (`user\_id`, `name`, `start\_date`, `end\_date`, `description`) VALUES (@user\_id, @name, @start\_date, @end\_date, @description)* |

## **Show event in calendar**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***Event***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getEvent\_id* | *get value of field Event\_id* |
| *02* | *setEvent\_id* | *set value of field Event\_id* |
| *03* | *getUser\_id* | *get value of field User\_id* |
| *04* | *setUser\_id* | *set value of field User\_id* |
| *05* | *getGroup\_id* | *get value of field Group\_id* |
| *06* | *setGroup\_id* | *set value of field Group\_id* |
| *07* | *getFrequency\_id* | *get value of field Frequency\_id* |
| *08* | *setFrequency\_id* | *set value of field Frequency\_id* |
| *09* | *getTitle* | *get value of field Title* |
| *10* | *setTitle* | *set value of field Title* |
| *11* | *getStart\_date* | *get value of field Start\_date* |
| *12* | *setStart\_date* | *set value of field Start\_date* |
| *13* | *getEnd\_date* | *get value of field End\_date* |
| *14* | *setEnd\_date* | *set value of field End\_date* |
| *15* | *getDescription* | *get value of field Description* |
| *16* | *setDescription* | *set value of field Description* |
| *17* | *getRemid* | *get value of field Remid* |
| *18* | *setRemid* | *set value of field Remid* |

##### ***CalendarDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *CalendarDAO.INSTANCE.getShowEvent()* | *Show event of user in calendar* |
| *02* | *loadEventsByUserIdAndStartDate* | *Displays events created by the user and displays the start date* |

##### ***ShowEventServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *get request from UI and send events in calendar at home page* |

### 

### **c.Sequence Diagram**

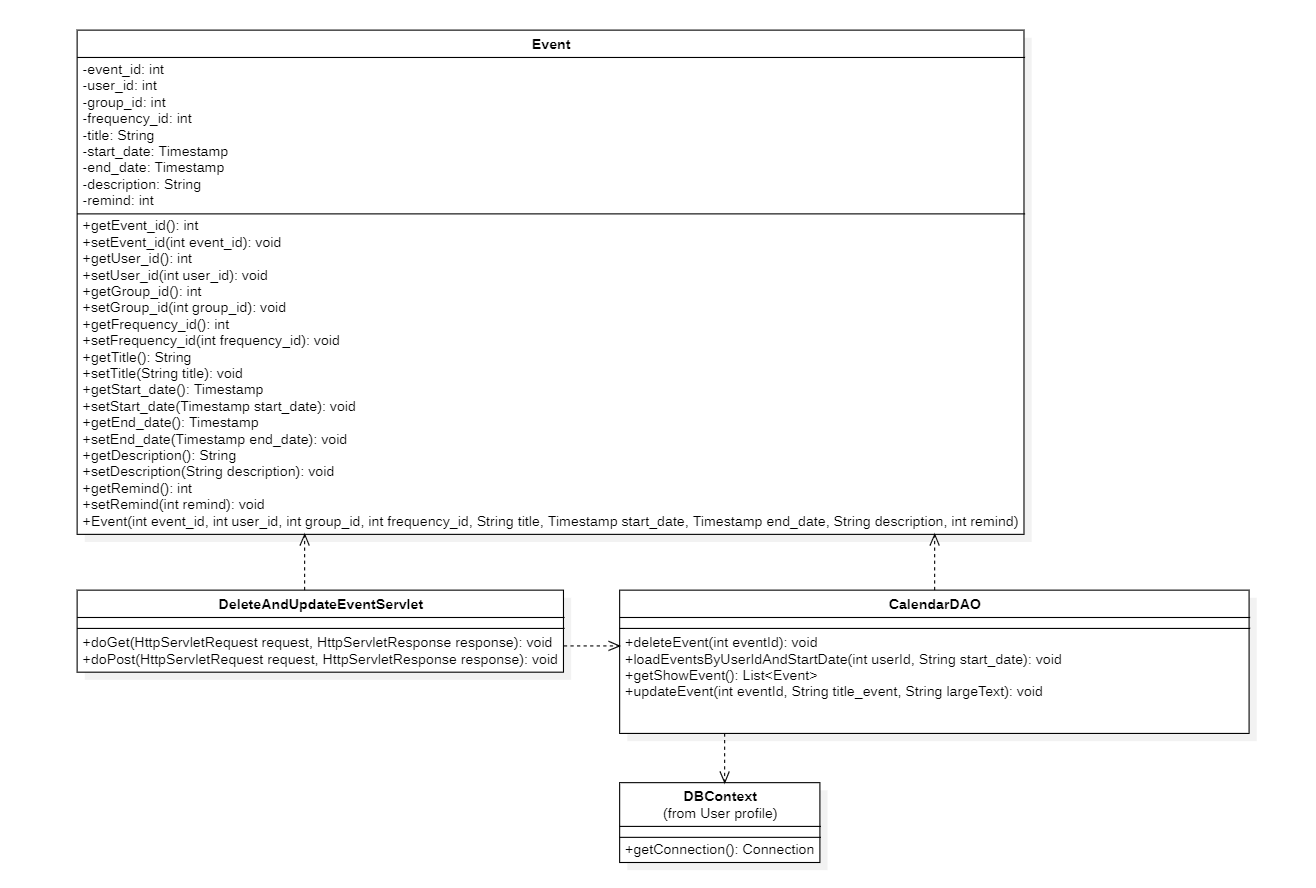
### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *loadEventsByUserIdAndStartDate* | *SELECT \* FROM event WHERE user\_id = @user\_id AND start\_date >= @start\_date AND start\_date < @start\_date* |

## **Update/Delete event in calendar**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***Event***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getEvent\_id* | *get value of field Event\_id* |
| *02* | *setEvent\_id* | *set value of field Event\_id* |
| *03* | *getUser\_id* | *get value of field User\_id* |
| *04* | *setUser\_id* | *set value of field User\_id* |
| *05* | *getGroup\_id* | *get value of field Group\_id* |
| *06* | *setGroup\_id* | *set value of field Group\_id* |
| *07* | *getFrequency\_id* | *get value of field Frequency\_id* |
| *08* | *setFrequency\_id* | *set value of field Frequency\_id* |
| *09* | *getTitle* | *get value of field Title* |
| *10* | *setTitle* | *set value of field Title* |
| *11* | *getStart\_date* | *get value of field Start\_date* |
| *12* | *setStart\_date* | *set value of field Start\_date* |
| *13* | *getEnd\_date* | *get value of field End\_date* |
| *14* | *setEnd\_date* | *set value of field End\_date* |
| *15* | *getDescription* | *get value of field Description* |
| *16* | *setDescription* | *set value of field Description* |
| *17* | *getRemid* | *get value of field Remid* |
| *18* | *setRemid* | *set value of field Remid* |

##### ***CalendarDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *deleteEvent* | *Delete event of user in calendar* |
| *02* | *loadEventsByUserIdAndStartDate* | *Displays events created by the user and displays the start date* |
| *03* | *updateEvent* | *Update event of user in calendar* |

##### ***DeleteAndUpdateEventServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *get request from UI and delete event in calendar* |
| *02* | *doPost* | *get request from UI and update information for event in calendar* |

### 

### **c.Sequence Diagram**

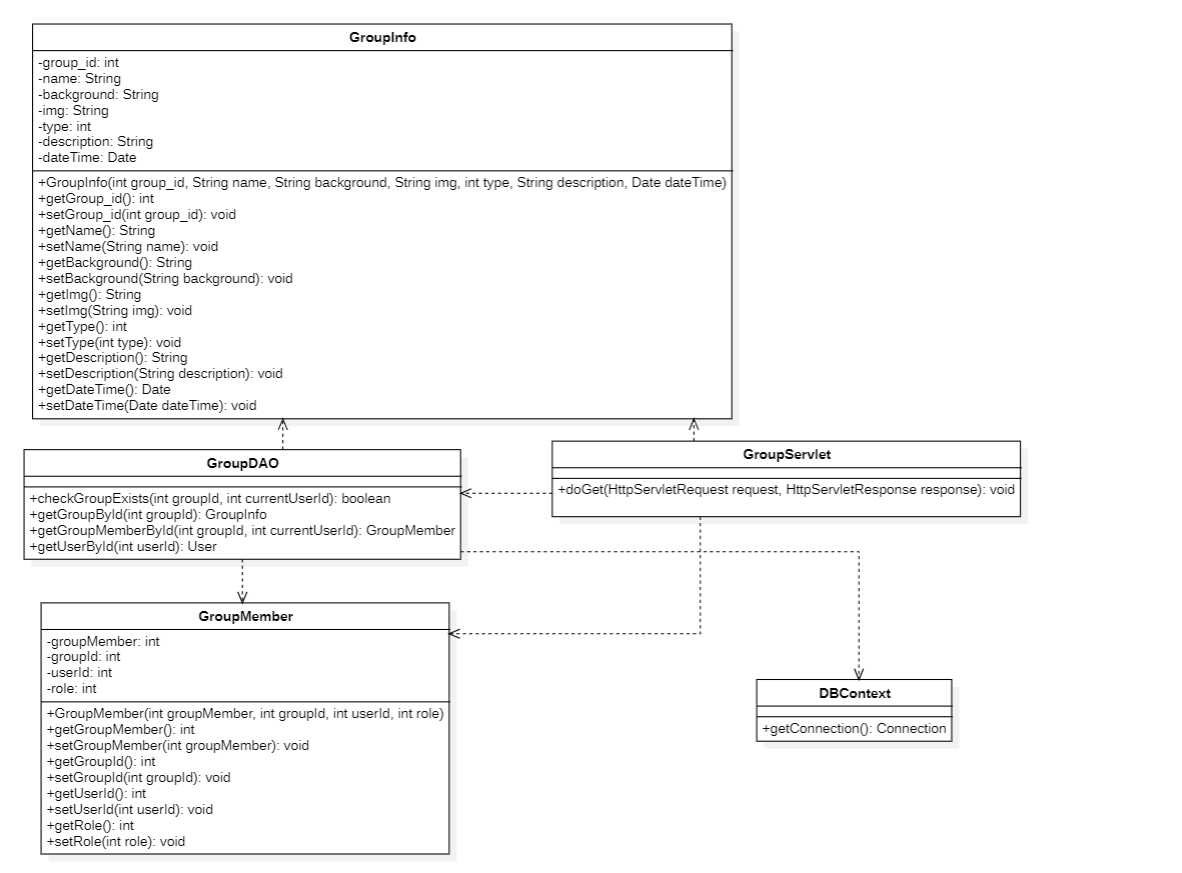
### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *loadEventsByUserIdAndStartDate* | *SELECT \* FROM event WHERE user\_id = @user\_id AND start\_date >= @start\_date AND start\_date < @start\_date* |
| *02* | *deleteEvent* | *DELETE FROM event WHERE event\_id = @event\_id* |
| *03* | *updateEvent* | *UPDATE event SET name = @name, description = @description WHERE event\_id = @event\_id* |

## **Group**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***GroupInfo***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getGroup\_id* | *get value of field Group\_id* |
| *02* | *setGroup\_id* | *set value of field Group\_id* |
| *03* | *getName* | *get value of field Name* |
| *04* | *setName* | *set value of field Name* |
| *05* | *getBackground* | *get value of field Background* |
| *06* | *setBackground* | *set value of field Background* |
| *07* | *getImg* | *get value of field Img* |
| *08* | *setImg* | *set value of field Img* |
| *09* | *getType* | *get value of field Type* |
| *10* | *setType* | *set value of field Type* |
| *11* | *getDescription* | *get value of field Description* |
| *12* | *setDescription* | *set value of field Description* |
| *13* | *getDateTime* | *get value of field DateTime* |
| *14* | *setDateTime* | *set value of field DateTime* |

##### ***GroupDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *checkGroupExists* | *Check group is existed or not* |
| *02* | *getGroupById* | *Find group by id* |
| *03* | *getGroupMemberById* | *Find member in group by id* |
| *04* | *getUserById* | *Find user by id* |

##### ***GroupServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *get request from UI to check group is existed or not* |

### 

### **c.Sequence Diagram**

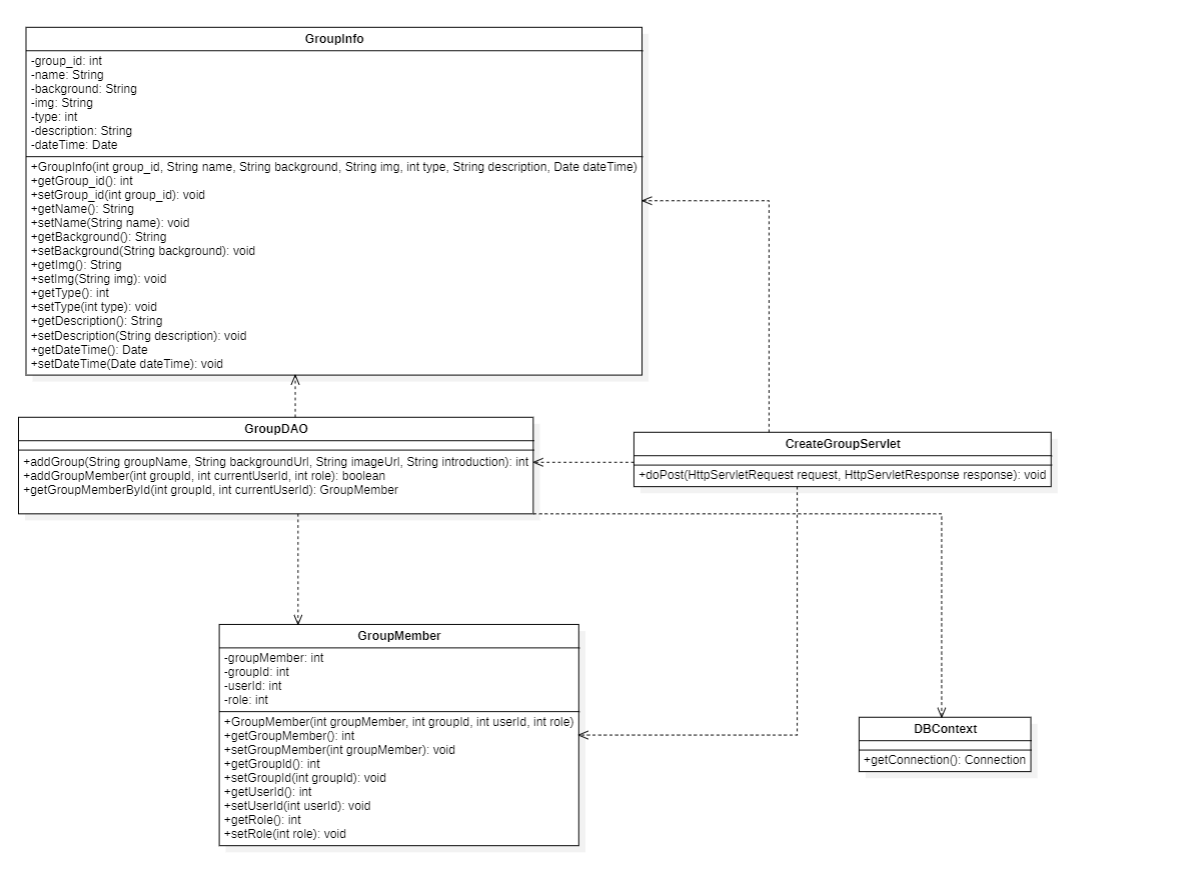
### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *checkGroupExists* | *SELECT \* FROM group\_member WHERE group\_id = @group\_id AND user\_id = @user\_id* |
| *02* | *getGroupById* | *SELECT \* FROM `group` WHERE group\_id = @group\_id* |
| *03* | *getGroupMemberById* | *SELECT \* FROM group\_member WHERE group\_id = @group\_id and user\_id = @user\_id* |
| *04* | *getUserById* | *SELECT user\_id, first\_name, last\_name, phone\_number, email, image FROM user WHERE user\_id = @user\_id* |

## **Create group**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***GroupInfo***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getGroup\_id* | *get value of field Group\_id* |
| *02* | *setGroup\_id* | *set value of field Group\_id* |
| *03* | *getName* | *get value of field Name* |
| *04* | *setName* | *set value of field Name* |
| *05* | *getBackground* | *get value of field Background* |
| *06* | *setBackground* | *set value of field Background* |
| *07* | *getImg* | *get value of field Img* |
| *08* | *setImg* | *set value of field Img* |
| *09* | *getType* | *get value of field Type* |
| *10* | *setType* | *set value of field Type* |
| *11* | *getDescription* | *get value of field Description* |
| *12* | *setDescription* | *set value of field Description* |
| *13* | *getDateTime* | *get value of field DateTime* |
| *14* | *setDateTime* | *set value of field DateTime* |

##### ***GroupDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *addGroup* | *Create new group* |
| *02* | *addGroupMember* | *Add member to group* |
| *03* | *getGroupMemberById* | *Find member in group by id* |

##### ***CreateGroupServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doPost* | *get request from UI and create new a group* |

### 

### **c.Sequence Diagram**

### **d.Database Queries**

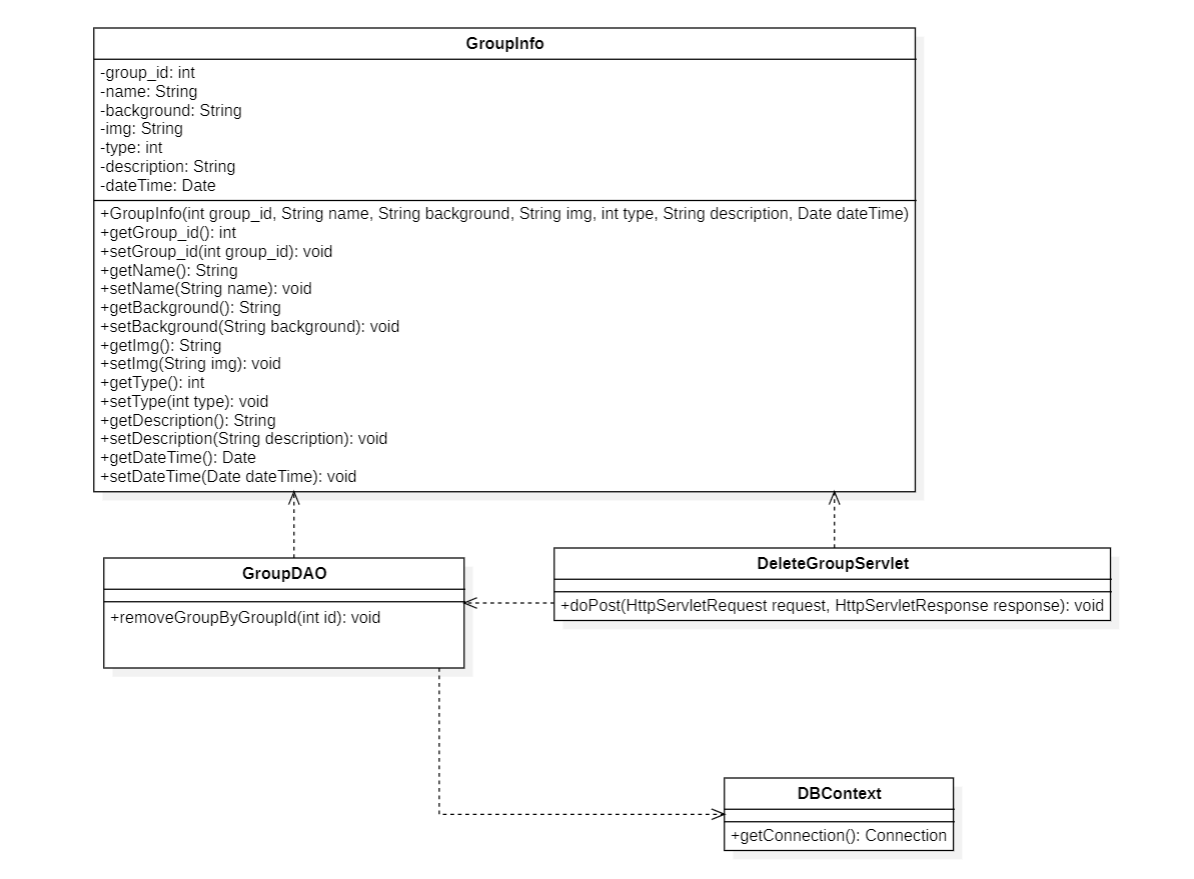
| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *addGroup* | *INSERT INTO `group` (name, background, img, type, description, date) VALUES (@name, @background, @img, 0, @description, CURRENT\_TIMESTAMP)* |
| *02* | *addGroupMember* | *INSERT INTO group\_member (group\_id, user\_id, Role) VALUES (@group\_id, @user\_id, @Role)* |
| *03* | *getGroupMemberById* | *SELECT user\_id, first\_name, last\_name, phone\_number, email, image FROM user WHERE user\_id = @user\_id* |

## 

## **Delete group**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***GroupInfo***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getGroup\_id* | *get value of field Group\_id* |
| *02* | *setGroup\_id* | *set value of field Group\_id* |
| *03* | *getName* | *get value of field Name* |
| *04* | *setName* | *set value of field Name* |
| *05* | *getBackground* | *get value of field Background* |
| *06* | *setBackground* | *set value of field Background* |
| *07* | *getImg* | *get value of field Img* |
| *08* | *setImg* | *set value of field Img* |
| *09* | *getType* | *get value of field Type* |
| *10* | *setType* | *set value of field Type* |
| *11* | *getDescription* | *get value of field Description* |
| *12* | *setDescription* | *set value of field Description* |
| *13* | *getDateTime* | *get value of field DateTime* |
| *14* | *setDateTime* | *set value of field DateTime* |

##### ***GroupDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *removeGroupByGroupId* | *Remove a group by id* |

##### ***DeleteGroupServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doPost* | *get request from UI and delete a group* |

### 

### **c.Sequence Diagram**

### **d.Database Queries**

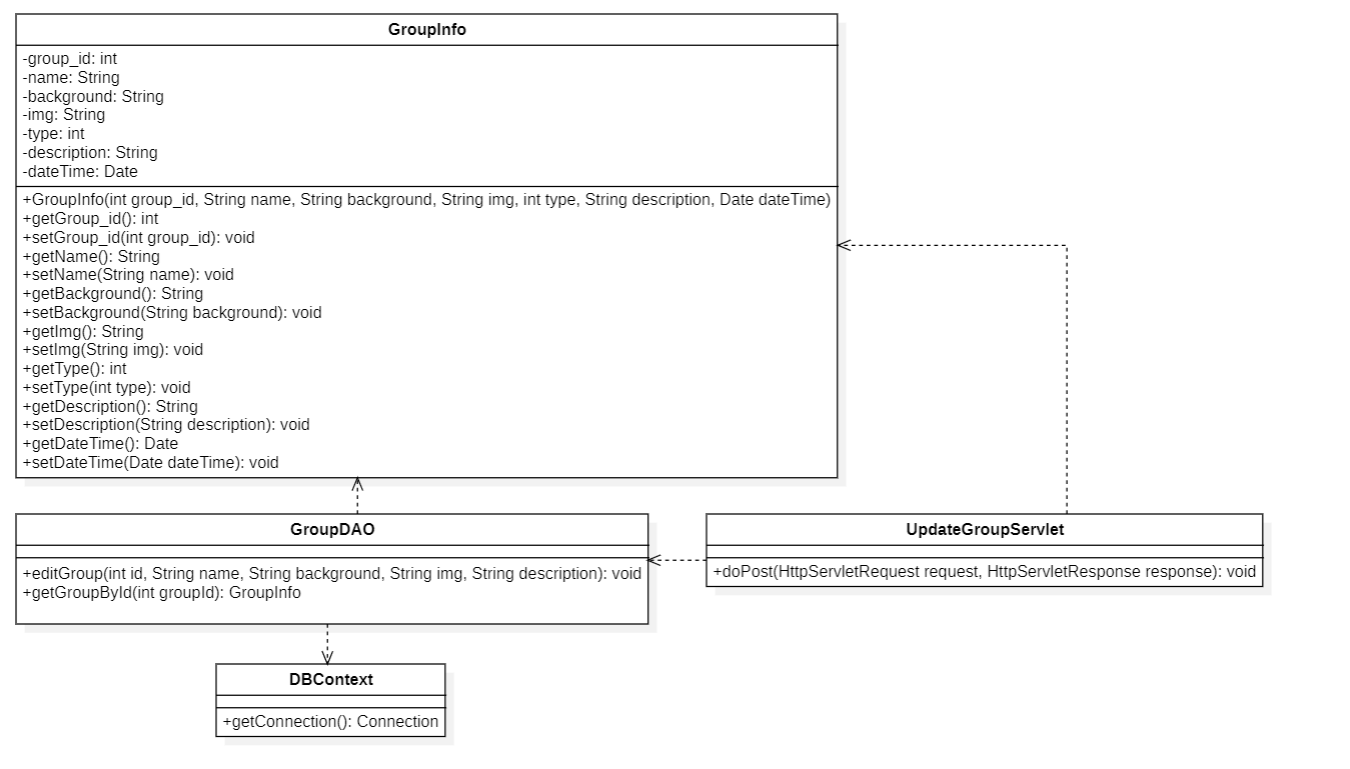
| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *removeGroupByGroupId* | *Delete from `social`.`group` Where group\_id = @group\_id* |

## 

## **Update group**

### 

### **a.Class Diagram**



### **b.Class Specification**

##### **DBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### ***GroupInfo***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getGroup\_id* | *get value of field Group\_id* |
| *02* | *setGroup\_id* | *set value of field Group\_id* |
| *03* | *getName* | *get value of field Name* |
| *04* | *setName* | *set value of field Name* |
| *05* | *getBackground* | *get value of field Background* |
| *06* | *setBackground* | *set value of field Background* |
| *07* | *getImg* | *get value of field Img* |
| *08* | *setImg* | *set value of field Img* |
| *09* | *getType* | *get value of field Type* |
| *10* | *setType* | *set value of field Type* |
| *11* | *getDescription* | *get value of field Description* |
| *12* | *setDescription* | *set value of field Description* |
| *13* | *getDateTime* | *get value of field DateTime* |
| *14* | *setDateTime* | *set value of field DateTime* |

##### ***GroupDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *editGroup* | *Edit information of group* |
| *02* | *getGroupById* | *Find group by id* |
| *03* | *getGroupMemberById* | *Find member in group by id* |

##### ***UpdateGroupServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doPost* | *get request from UI and update information of group* |

### 

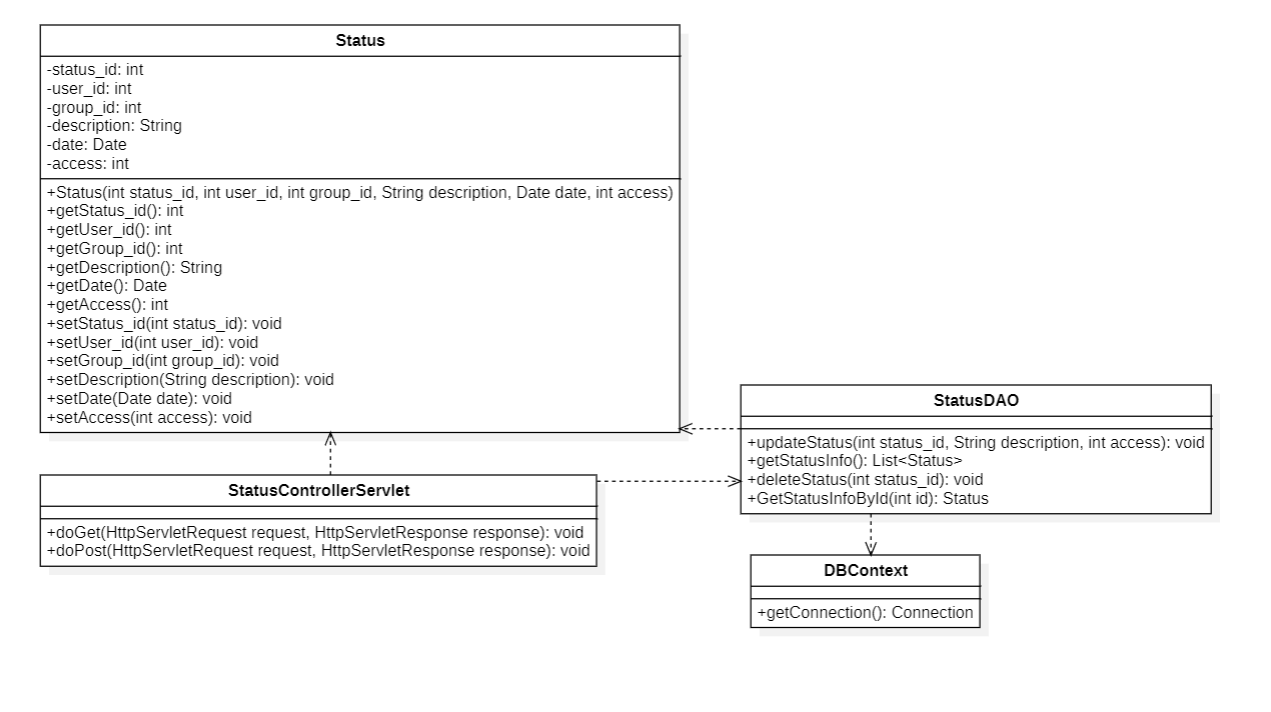
### **c.Sequence Diagram**

### **d.Database Queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *editGroup* | *UPDATE `social`.`group` SET `name` = @name, `background`= @background, `img` = @img,*  *`description` = @description WHERE `group\_id` = @group\_id* |
| *02* | *getGroupById* | *SELECT \* FROM `group` WHERE group\_id = @group\_id* |
| *03* | *getGroupMemberById* | *SELECT \* FROM group\_member WHERE group\_id = @group\_id and user\_id = @user\_id* |

## **Add/Delete Status**

### **a.Class Diagram**



### **b.Class Specifications**

##### ***Status***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getStatus\_id* | *get value of field Status\_id* |
| *02* | *setStatus\_id* | *set value of field Status\_id* |
| *03* | *getUser\_id* | *get value of field User\_id* |
| *04* | *setUser\_Id* | *set value of User\_Id* |
| *05* | *getGroup\_id* | *get value of field Group\_id* |
| *06* | *setGroup\_id* | *set value of field Group\_id* |
| *07* | *getDescription* | *get value of field Description* |
| *08* | *setDescription* | *set value of field Description* |
| *09* | *getDate* | *get value of field Date* |
| *10* | *setDate* | *set value of field Date* |
| *11* | *getAccess* | *get value of field Access* |
| *12* | *setAccess* | *set value of field Access* |

##### ***DBContext***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### 

##### ***StatusControllerServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *get request from UI and show/delete group* |
| *02* | *doPost* | *get request from UI and add post to status* |

##### ***StatusDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *GetStatusInfoById* | *execute sql query to get status by status\_id* |
| *02* | *deleteStatus* | *delete status of user* |
| *03* | *addStatus* | *add new status user post to news* |

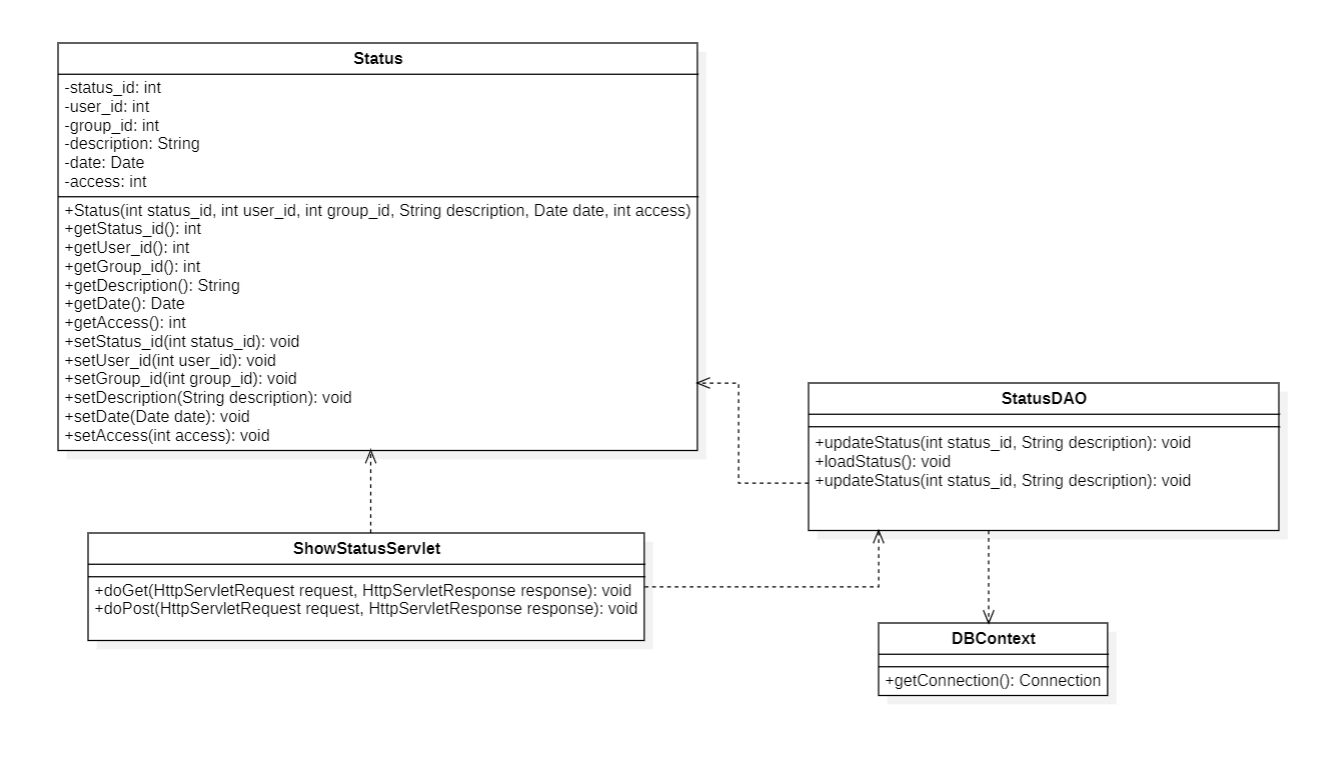
### **c.Sequence Diagram(s)**

### **d.Database queries**

| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *GetStatusInfoById* | *select \* from status where status\_id = @status\_id* |
| *02* | *deleteStatus* | *INSERT INTO `social`.`status` (`user\_id`, `description`, `date`, `access`) VALUES (@user\_id, @description, @date, @access)* |
| *03* | *addStatus* | *INSERT INTO `social`.`status` (`user\_id`, `description`, `date`, `access`) VALUES (@user\_id, @description, @date, @access)* |

## **Update/Show Status**

### **a.Class Diagram**



### **b.Class Specifications**

##### ***Status***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getStatus\_id* | *get value of field Status\_id* |
| *02* | *setStatus\_id* | *set value of field Status\_id* |
| *03* | *getUser\_id* | *get value of field User\_id* |
| *04* | *setUser\_Id* | *set value of User\_Id* |
| *05* | *getGroup\_id* | *get value of field Group\_id* |
| *06* | *setGroup\_id* | *set value of field Group\_id* |
| *07* | *getDescription* | *get value of field Description* |
| *08* | *setDescription* | *set value of field Description* |
| *09* | *getDate* | *get value of field Date* |
| *10* | *setDate* | *set value of field Date* |
| *11* | *getAccess* | *get value of field Access* |
| *12* | *setAccess* | *set value of field Access* |

##### ***DBContext***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getConnection* | *return a connection from the database* |

##### 

##### ***ShowStatusServlet***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *get request from UI and show/delete group* |
| *02* | *doPost* | *get request from UI and add post to status* |

##### ***StatusDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *StatusDAO.INSTANCE.getStatusInfo* | *to get status information* |
| *02* | *StatusDAO.INSTANCE.updateStatus* | *edit description of status* |

### **c.Sequence Diagram(s)**

### **d.Database queries**

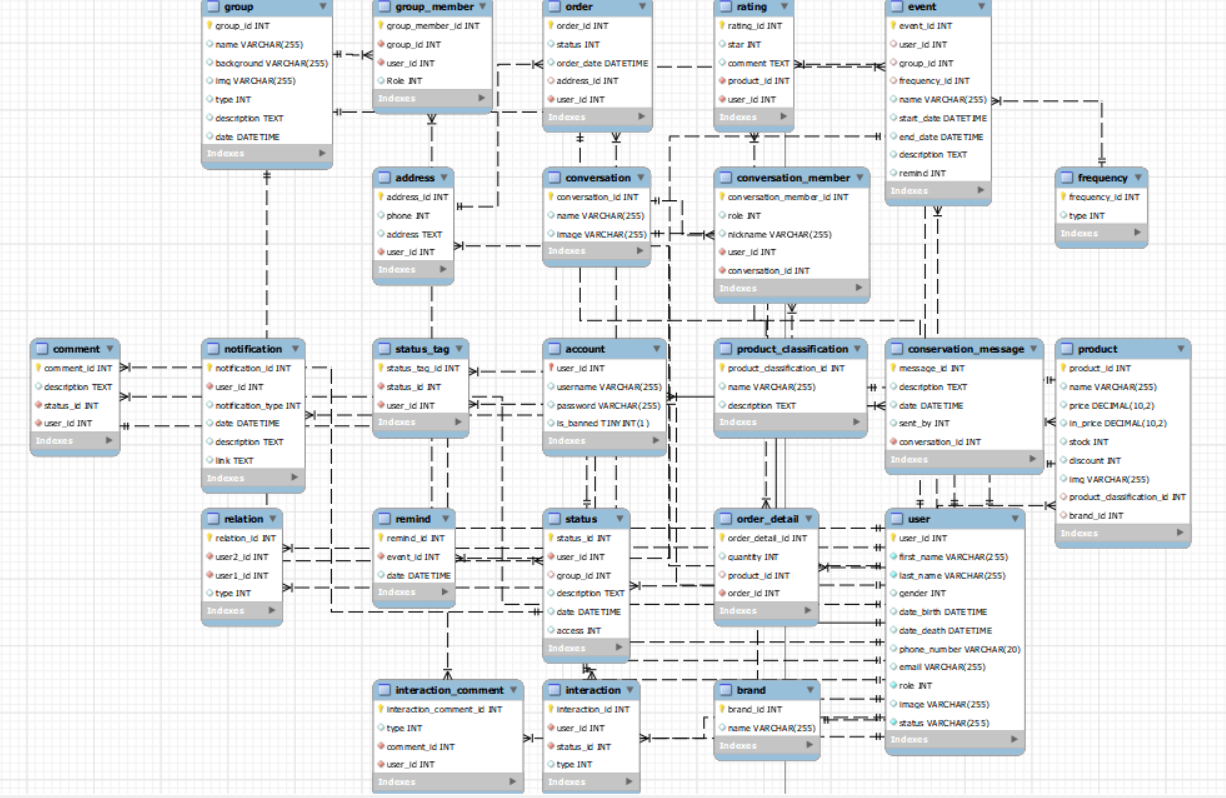
| **No** | **Method** | **Queries** |
| --- | --- | --- |
| *01* | *loadStatus* | *Select \* from status order by date desc* |
| *02* | *updateStatus* | *UPDATE `social`.`status` SET `description` = @description WHERE (`status\_id` = @status\_id)* |

# 

# **III. Database Design**

# 

## **1. Database Schema**



## **2. Table Description**

| **No** | **Table** | **Description** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 

## 