**BUG MICROBLOG**

**Problem Statement:**

**The basic information of our system is as follows-**

The system/software to be developed is basically a social application, which creates a virtual social world and thus helps to interact among the social world thus establishing a knowledge sharing as well as an entertainment platform.

**The managing services required are-**

* Sign-Up/Registration
* Login
* Forgot Password
* My Profile
* Change Password
* Create Post
* View Posts
* Trends
* Update Post
* Delete Post
* Search
* Private Notes
* AutoLogout
* Chat

**SOFTWARE REQUIREMENTS SPECITFICATION**

**1. Introduction**

The BUG Microblog is a microblogging application for the social and healthy interaction among the society as well as an online chatting application to enhance an on the go social network.

**1.1 Purpose**

This software package is developed from scratch exclusively for the society as a whole to:

* Provide a real-time interaction amongst the people belonging to different fields.
* To enable the registered users to post blogs as under certain categories.
* To search for the trending news, events, timelines, etc. to stay updated.
* To provide information via blogs publicly to keep in touch with the general as well as technical knowledge.
* To provide a multimedia support including audios, videos, GIFs etc. to provide a better view of the surroundings.
* To give a virtual support with the technical experts using the chatting application.
* To maintain a personal information of individuals like notes, timelines, etc. as well in addition to the social attachment.
* To provide a free tour for the unregistered explorers (users) as well.

**1.2 Project Scope**

**Benefits:**

* The software provides a virtual access to the world under one roof.
* It helps to develop a healthy social environment.
* It facilitates a private access also in addition to social network.
* A helpful chat is ensured with techicals belonging to various fields.
* Is an better alternative to the existing softwares in terms of services provided.

**Objective:**

* The objective of the to be developed ' BUG Microblog’ software package is to provide a real time web based software. Hence the scope of the project is over a wide range of the social world at an educational level only.
* Also this software provides a one stop access to the various applications like chats, news, trends, private notes-area, etc.

Future Extension:

Project can be extended to a business level enterprise software and can be deployed to a business oriented company, non-profit organization or can be open sourced.

**2.Overview:**

**2.1 Product Perspective**

BUG Microblog is meant to serve as a common platform where the various information regarding to the different sections like politics, education, sports, etc. are shared as well as a chatting application is embedded too. Our goal is to develop a replacement to the pre-existing microblogging sites and social networking site and bring them both to a common platform along with a private storage for the users as well and make it more users friendly.

**2.2 Product Features**

**Modules-**

**Registration:**

This module helps to register a new user into the Software Database, and check is made for duplicate entries and generates ID and Password for future access.

To achieve this, a secure verification process is followed. The user interested to sign-up is asked for mail-id and password and as the user submits it, they are temporarily registered to the database as “Archive Users”. The user receives a verification link to verify his/her id and fill other required details, within a predefined time of 20:00 hours from registration. If not verified within time limit the email-id is deleted from “Archive Users”.

**Login / Logout:**

This module take input as user Id and password for logging into the system and user can easily logout as well.

**Forgot Password/ Change Password:**

The user is provided with a randomly generated password on his/her mail-id in case user forgets his/her password which could further be changed to a new password.

Change password is facilitated by the software by proper validations as well.

**Profile Management:**

The user can fill up various additional details (revealed publicly) like user-pic, status, city, state, etc. Also, user can look up for the followers and following user’s details and posts specific to them.

**Posts Management:**

Create Post:

The user can create a post of his/her interested category, can link up a youtube-link, add photos, text related to the post as well as can add taglines too.

In addition to it an optional “Share Location” facility is also provided to get the actual location of post creation.

View Posts:

The user relevant posts are displayed as per the trends(sorted as per a unique sort algorithm). User can like/unlike a post and can add comments on them. The geo-location of the post can also be viewed. The user can also view particular user’s post also and can follow or unfollow them. Also the user can view the profile, followers, following of other users.

The user’s home page are then updated with the post of the followed user as well.

Update Post/ Delete Post: Posts can be updated by the user as well as can be deleted.

Trends: The user can also see the post category wise even if the category is not included in the user’s interests.

**Search Blogs:**

This module is for the various search techniques to generate the relevant results based on category, type, date, followers, timeline, trending etc.

The query for search fetches the relevant posts as well as the username are also displayed.

As a local search area, the user’s post are searched and displayed on the user’s homepage for further updations, if any.

**Manage Note:**

This is a private module belonging to individual users which is for making sticky notes on the user homepage which include their to-do tasks, to-remember tasks, notify deadlines and expiry date for tasks, mark as done.

Notes could be created along with photos, and can be used as a private storage space for user access and can be deleted as well.

**Chatting:**

This module is a micro social network for an online chatting with the experts and followers so as to keep in touch with them and establish a strong virtual network in addition to the blog posts.

The chatting is established between the user who follow each other, a term defined as “Binded Users”.

The user can also send images and pdf flies in addition to the text messages sent.

**Auto Logout:**

It is a module that keeps a track on the user’s blog access time by maintaining a timer as described by the user and as result the user gets automatically logged out as the timer returns zero.

**2.3 Design Constraints:**

Cardinalities-

* Multiple User can follow multiple User (another). m...n
* One User can post multiple microblogs(posts). 1…n
* Multiple User can message multiple (other)User. m…n
* One User can create multiple Notes. 1…n
* Multiple User are interested in multiple Category. m…n
* One Category can belong to multiple Microblog. 1…n
* Multiple User can add user\_pic which can have multiple Images. m…n
* Multiple Microblog can contain multiple Images. m…n
* Multiple Users can write multiple Comments. m…n
* Multiple Comments are commented on multiple Microblog. m…n

**2.4 Software System Attributes:**

**1. Security:-**

The system has the facility to provide security of data since login-password is maintained. It also ensures the integrity of system from accidental or malicious damage.

**2. Usability:-**

The system to be developed provides usability of certain data so that an effective memory management is achieved.

**3. Reliability:-**

The reliability is maintained in the to be developed software, so that the data as well as all other included operation provide the correct calculations in a reliable way.

**4. Supportability:-**

The system will be supported on various platform and thus ensure that it is platform independent.

**5. Adaptability:**

The system welcomes any new module which provides any new facility to the existing system.

**6. Familiar Interface:-**

The new system will have to be user friendly and familiar to users.

**7. Identification: -** The new system will be able to determine the class of user as registered users, unregistered, administration or software development team.

**3. Logical Database Requirements:**

# **ER diagram description:**

Entity Description-

* Registered Users are disjoint into Archive User and User.
* Comment is a weak entity depending on User by identifying relationship, writes.
* Images is a weak entity depending on User by identifying relationship, user\_pic.
* Category as well as User are strong entities and are related by relation, writes.
* Microblog is a weak entity depending on User and Category by identifying relationships, posts and belongs\_to respectively.
* Note is a weak entity depending on User by identifying relationship, creates.
* Microblog is related to Images and Comments by relations, contains and commented\_on respectively; and all are weak entities.

Attributes description-

* The Profile is a composite attribute comprising of Profession, State, Status and Hometown.
* Post\_count is a derived attribute that can be derived from Posts.

**5. Implementation:**

**SQL QUERIES**

**//change\_forgot\_password.php**

$query = "select username from users where

username='$email'";

$query = "update users set password='$hash\_pass' where

username='$email'";

**//changepass.php**

$query="select \* from users where

username='".$\_SESSION['session\_user\_id']."'";

$query="update users set password='".md5($new\_pass)."' where

username='".$\_SESSION['session\_user\_id']."'";

**//core.inc.php**

$query="SELECT `$field` FROM `users` WHERE

`username`='".$\_SESSION['session\_user\_id']."'";

**//delete\_note.php**

$delete\_query="DELETE FROM `bug\_notes` WHERE

`username`='".getuserfield('username')."' AND

`note\_id`='$delete\_note\_id';";

**//delete\_post.php**

$query = "update category set

count = count-1 where

category in

(select category from post where post\_id='$post\_id')";

$query = "delete from post where

post\_id='$post\_id' and

posted\_by='".$\_SESSION['session\_user\_id']."'";

**//edit\_post.php**

$query = "update post set

content='$content', posted\_on=posted\_on,title='$title', video\_link='$\_link', location='$location', tags='$tags', category='$category' where

post\_id='$post\_id' and

posted\_by='$posted\_by'";

$query\_1 = "select \* from post where

post\_id='$edit\_post\_id' and

posted\_by = '".$\_SESSION['session\_user\_id']."'";

$query = "select category from category";

$query\_2 = "select \* from images where

post\_id='".$row['post\_id']."'";

**//edit\_user\_info.php**

$query = "update users set

firstname='$fname', lastname='$lname' where

username = '$uidd'";

$query = "select \* from user\_profile where

email='$uidd'";

$query = "update user\_profile set

profile\_status = '$status', city='$city', state='$state', profession = '$job' where

email = '$uidd'";

$query = "insert into user\_profile values ('$uidd','$status','$city','$state','$job',null)";

**//follow\_to\_by.php**

$query = "select follow\_to from follows where

follow\_by = '$user\_id' limit $limit\_start, $limit\_end";

$query1 = "select concat(firstname,' ',lastname) 'name',DATE\_FORMAT(created\_on,'%h:%i %p %d- %b\'%y ') as created\_on, gender from users where

username = '$uid'";

$query1 = "select category from interests where

user\_id='$uid'";

$query1 = "select count(\*) as no from follows where

follow\_by='$suid' and

follow\_to ='$uid'";

$query = "select follow\_by as follow\_to from follows where

follow\_to = '$user\_id' limit $limit\_start, $limit\_end";

$query1 = "select concat(firstname,' ',lastname) 'name',DATE\_FORMAT(created\_on,'%h:%i %p %d- %b\'%y ') as created\_on, gender from users where

username = '$uid'";

$query1 = "select category from interests

where user\_id='$uid'";

$query1 = "select count(\*) as no from follows where

follow\_by='$suid' and follow\_to ='$uid'";

**//follow\_unfollow.php**

$query = "insert into follows values('$follow\_by','$follow\_to');";

$query = "delete from follows where

follow\_by = '$follow\_by' and follow\_to = '$unfollow\_to';";

**//function\_image\_save.php**

$query = "select image\_path from images where

post\_id='$post\_id'";

$query = "delete from images where

image\_path='".$row['image\_path']."'";

$query = "delete from images where

image\_path='$image\_id'";

$query = "INSERT INTO `images`(`post\_id`, `image\_path`, `image\_name`, `image\_alt`, `relation`)

VALUES('$user\_id','$name','$name\_org','$tags','$relation')";

**//function.inc.php**

$query="select \* from online where

user\_ip = '".trim($\_SERVER['REMOTE\_ADDR'])."'";

$query="delete from online where

user\_ip = '".trim($\_SERVER['REMOTE\_ADDR'])."'";

**//get\_binded\_user.php**

$query = "select follow\_to from follows where

follow\_by='$user\_id' and

follow\_to in

(select follow\_by from follows where

follow\_to= '$user\_id');";

$query1 = "select concat(firstname,' ',lastname) as name from users where

username='$uid'";

**//get\_post.php**

$query = "select \*,date\_format(posted\_on,'%Y') 'year',date\_format(posted\_on,'%b/%d') 'date',date\_format(posted\_on,'%r') 'time' from post where

posted\_by in

(select follow\_to from follows where

follow\_by='$user\_id') or

(posted\_by='$user\_id') and

(datediff(now(),posted\_on) < 2)

order by posted\_on desc limit

$limit\_start,$limit\_end";

$query\_1 = "select \* from images where

post\_id='".$row['post\_id']."'";

$query\_1 = "select firstname from users where

username='".$row['posted\_by']."'";

$query\_1 = "select liked\_by from likes where

for\_post\_id = '".$row['post\_id']."' and

liked\_by ='".$\_SESSION['session\_user\_id']."'";

$query = "select \*,date\_format(posted\_on,'%Y') 'year',date\_format(posted\_on,'%b/%d') 'date',date\_format(posted\_on,'%r') 'time' from post where

category in

(select category from interests where

user\_id='$user\_id') and

(datediff(now(),posted\_on)<15) and

posted\_by not in

(select follow\_to from follows where

follow\_by='$user\_id') and

(posted\_by!='$user\_id')

order by (date\_format(posted\_on,'%d')\*10+likes+comments\*2) desc

limit $limit\_start,$limit\_end";

$query\_1 = "select \* from images where

post\_id='".$row['post\_id']."'";

$query\_1 = "select firstname from users where

username='".$row['posted\_by']."'";

$query\_1 = "select liked\_by from likes where

for\_post\_id = '".$row['post\_id']."' and

liked\_by ='".$\_SESSION['session\_user\_id']."'";

$query = "select \*,date\_format(posted\_on,'%Y') 'year',date\_format(posted\_on,'%b/%d') 'date',date\_format(posted\_on,'%r') 'time' from post where

$cat (datediff(now(),posted\_on) < 102)

order by (date\_format(posted\_on,'%d')\*10+likes+comments\*2) desc

limit $limit\_start,$limit\_end";

$query\_1 = "select \* from images where

post\_id='".$row['post\_id']."'";

$query\_1 = "select firstname from users where

username='".$row['posted\_by']."'";

**//get\_user\_msg.php**

$query = "select \*,date\_format(send\_on\_time,'%h:%i %p') as mtime from messages where

(sender\_id = '$rec' and reciver\_id ='$sender') or

(sender\_id = '$sender' and reciver\_id ='$rec')

order by send\_on\_time desc

limit 25";

$query1 = "update messages set

status = 2, send\_on\_time=send\_on\_time where reciver\_id='$reca' and

sender\_id ='$senda' and

send\_on\_time='$timeons' and

(status = 1 or status = 0)";

$query = "select \*,date\_format(send\_on\_time,'%h:%i %p') as mtime from messages where

((sender\_id = '$sender' and reciver\_id ='$rec'))

and status = 0

order by send\_on\_time desc

limit 25";

$query1 = "update messages set

status = 2, send\_on\_time=send\_on\_time where

reciver\_id='$reca' and

sender\_id ='$senda' and

send\_on\_time='$timeons' and

(status = 1 or status = 0)";

**//home.php**

$query = 'select category from category

order by count desc

limit 6';

$query = 'select category from category

order by count desc

limit 6';

**//Imageload.php**

$query="SELECT `note\_id`,length(note\_text),`note\_title`,`note\_text`,`deadline`, `priority`,

`task\_completed`,`note\_image\_name`,`note\_image\_path` FROM `bug\_notes` WHERE

`username`='".getuserfield('username')."'

AND `note\_id` like '".getuserfield('username')."%'

ORDER BY `note\_id`;";

$checkbox\_query="UPDATE bug\_notes SET

`task\_completed`=1 WHERE

`note\_id`='$checked\_note\_id';";

$checkbox\_query="UPDATE bug\_notes SET

`task\_completed`=0 WHERE

`note\_id`='$unchecked\_note\_id';";

**//index.php**

$query = "INSERT INTO `post` (`post\_id`, `posted\_by`, `content`, `title`, `video\_link`, `likes`, `location`, `tags`,`category`) VALUES

('$post\_id','$posted\_by','$content','$title','$\_link',0,'$location','$tags','$category')";

$query = 'select category from category

order by count desc

limit 6';

$query = "select category from category";

**//last\_user\_msg.php**

$query = "select \*,date\_format(send\_on\_time,'%h:%i %p') as time,date\_format(send\_on\_time,'%d- %m-%Y') as date\_on from messages m where

(m.sender\_id='$for\_user' or m.reciver\_id = '$for\_user')

and m.send\_on\_time in

(select max(send\_on\_time) from messages m1 where

(m1.sender\_id = m.sender\_id and m1.reciver\_id=m.reciver\_id) or

(m1.reciver\_id = m.sender\_id and m1.sender\_id = m.reciver\_id ))

order by send\_on\_time desc";

$query1 = "select concat(firstname) name from users where

username='$msg\_id'";

$query1 = "select concat(firstname) name from users where

username='$msg\_id'";

**//like\_comment.php**

$query = "insert into likes values('$liked\_by','$on\_post');";

$query = "update post set

likes=likes+1, posted\_on=posted\_on where

post\_id='$on\_post'";

$query = "delete from likes where

for\_post\_id='$on\_post' and

liked\_by='$liked\_by';";

$query = "update post set

likes=likes-1, posted\_on=posted\_on where

post\_id='$on\_post'";

$query = "select comment,comment\_time,comment\_by from comment where

post\_id='$post\_id'

order by comment\_time desc";

**//logout\_user.php**

$query = "delete from online where

user\_id='".$\_SESSION['session\_user\_id']."'";

**//no\_pic.php**

$query = "select profile\_img from user\_profile

where email='$user\_id'";

$query = "select firstname from users where

username = '$user\_id'";

**//note1.php**

$query\_count=mysql\_query("SELECT count(username) AS count from bug\_notes WHERE

`username`='".getuserfield('username')."';");

$query="INSERT INTO `bug\_notes` VALUES('$username','$note\_id','$note\_title','$note\_text','$start\_date','$deadline','$priority' ,'$task\_completed','$note\_image\_name','$target\_file')";

**//post\_comment.php**

$query = "insert into comment values

('$cmt\_id','$post\_id','$post\_content',now(),'$by\_user\_id')";

$query = "update post set

comments=comments+1, posted\_on=posted\_on where

post\_id='$post\_id'";

**//prof.php**

$query = 'select category from category

order by count desc

limit 6';

$query = "select count(\*) as count from follows where

follow\_to='$prof\_id' and

follow\_by='".$\_SESSION['session\_user\_id']."'";

$query = "select firstname,lastname from users where

username='$prof\_id'";

$query = "select count(\*) as count from follows where

follow\_to='$prof\_id'";

$query = "select count(\*) as count from follows where

follow\_by='$prof\_id'";

$query = "select \* from user\_profile where

email='$prof\_id'";

**//profile\_image.php**

$query = "select image\_path from images where

post\_id = 'prf\_pic\_".$user\_id."'

order by image\_path desc

limit 1";

$query = "select count(\*) as count from user\_profile where

email='$user\_id'";

$query = "insert into user\_profile (email,profile\_img) values ('$user\_id','$img\_path')";

$query = "update user\_profile set

profile\_img = '$img\_path' where

email='$user\_id'";

$query = "select count(\*) as count from user\_profile where

email='$user\_id'";

$query = "insert into user\_profile (email,profile\_img) values ('$user\_id','$img\_path')";

$query = "select image\_path, image\_alt from images where

post\_id='prf\_pic\_".$user\_id."'";

**//search.php**

$query = "SELECT post\_id,posted\_by,content,title,video\_link FROM `post` where

concat(content,title,tags) like '%$q%'

order by (likes\*1 + comments\*2) desc

limit 3;";

$query1 = "select image\_path, image\_alt from images where

post\_id = '$post\_id'

limit 1";

$query = "SELECT \* FROM `users` e where

concat(firstname,' ',lastname) like '$q%'

order by

(select count(\*) from follows where

follow\_to=e.username)

desc limit 3;";

**//send\_user\_msg.php**

$query = "insert into messages values ('$from\_user','$for\_user','$msg',now(),'text',0)";

**//sign\_in\_up.php**

$query = "select \* from users where

(username='$username' and password='$hashed\_password')";

$query = "select \* from online where

user\_id='$username'";

$query = "insert into online values ('$username',1,'$user\_ip','$device\_info',now())";

$query = "delete from online where

user\_id = '$username'";

$query = "insert into online values ('$username',1,'$user\_ip','$device\_info',now())";

$query = "select \* from users where

username='$email'";

$query = "insert into archive\_user (username,password,gender,check\_code,status)

values('$email','$hashed\_password','$gender','$rand\_hash',0)";

**//user\_post.php**

$query = "select \*,date\_format(posted\_on,'%Y') 'year',date\_format(posted\_on,'%b/%d') date',date\_format(posted\_on,'%r') 'time' from post where

posted\_by = '$user\_id\_for\_post'

order by posted\_on desc

limit $limit\_start,$limit\_end";

$query\_1 = "select \* from images where

post\_id='".$row['post\_id']."'";

$query\_1 = "select firstname from users where

username='".$row['posted\_by']."'";

$query\_1 = "select liked\_by from likes where

for\_post\_id = '".$row['post\_id']."' and

liked\_by ='".$\_SESSION['session\_user\_id']."'";

**//verify\_mail.php**

$query = 'select \* from archive\_user where

username="'.$username.'" and

check\_code="'.$code.'" and

(timediff(now(),created\_on) < "20:00:00" or status = 1)';

$query = "update archive\_user set

status=1 where username='$username' and

check\_code='$code'";

$query = "select \* from users where

username='$username'";

**//welcome\_user.php**

$query = "insert into users (username,firstname,lastname,password,created\_on,gender) values

('$username','$f\_name','$l\_name','$hashed\_password','$created\_on','$gender')";

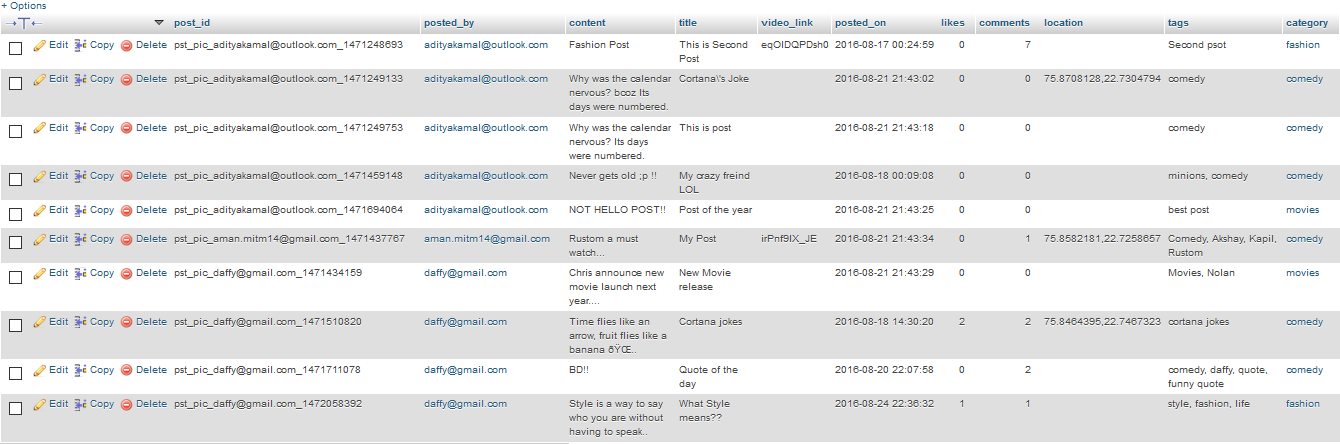
$query = "delete from archive\_user where

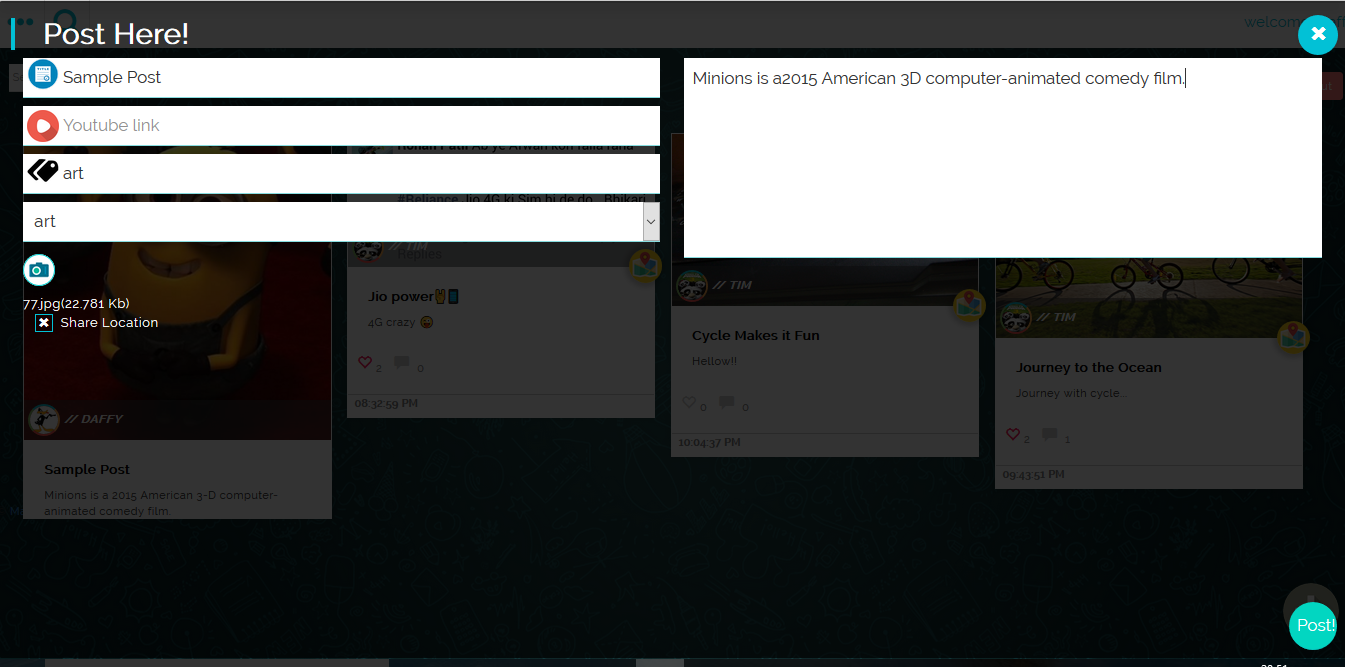
username='$username' or (timediff(now(),created\_on)>'20:00:00' and status = 0)";

$query = "select category from category";

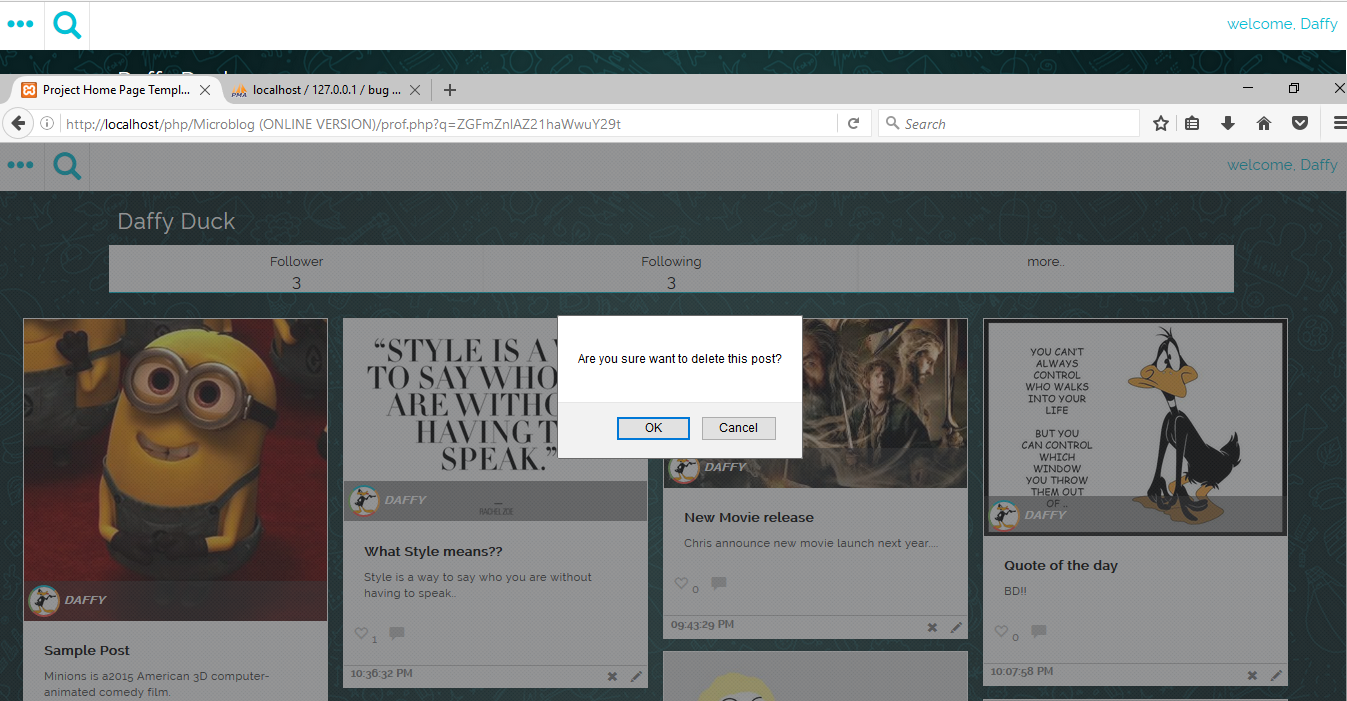
**6.Testing:**

**1. Add Post:**

(Database before post creation)

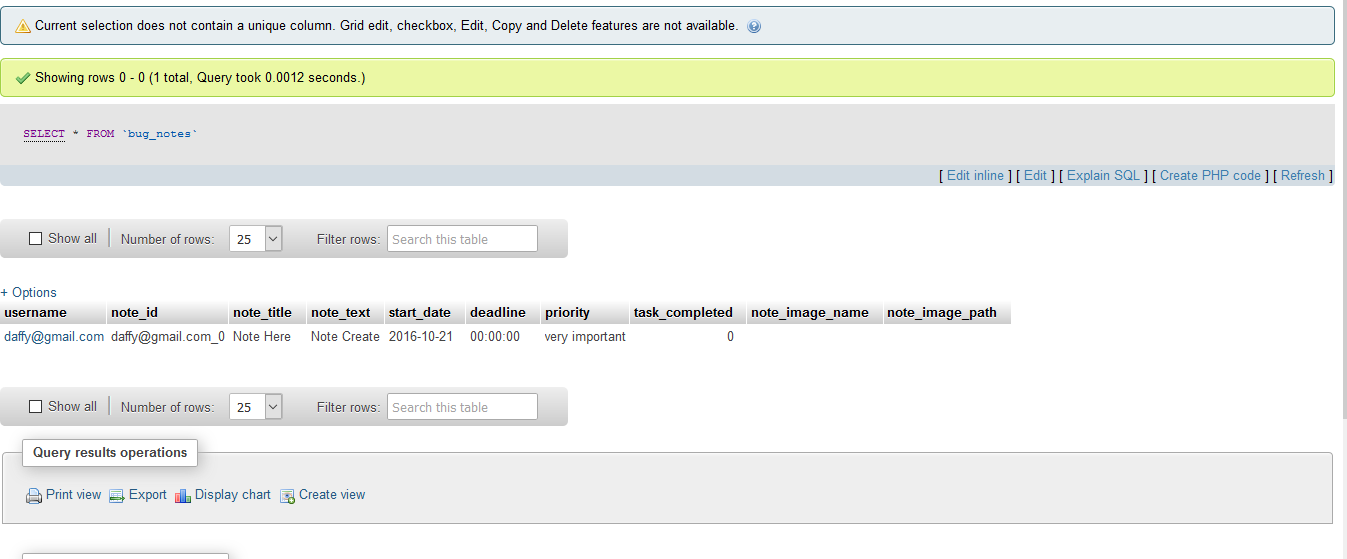
(Creating post)

(database after post creation)

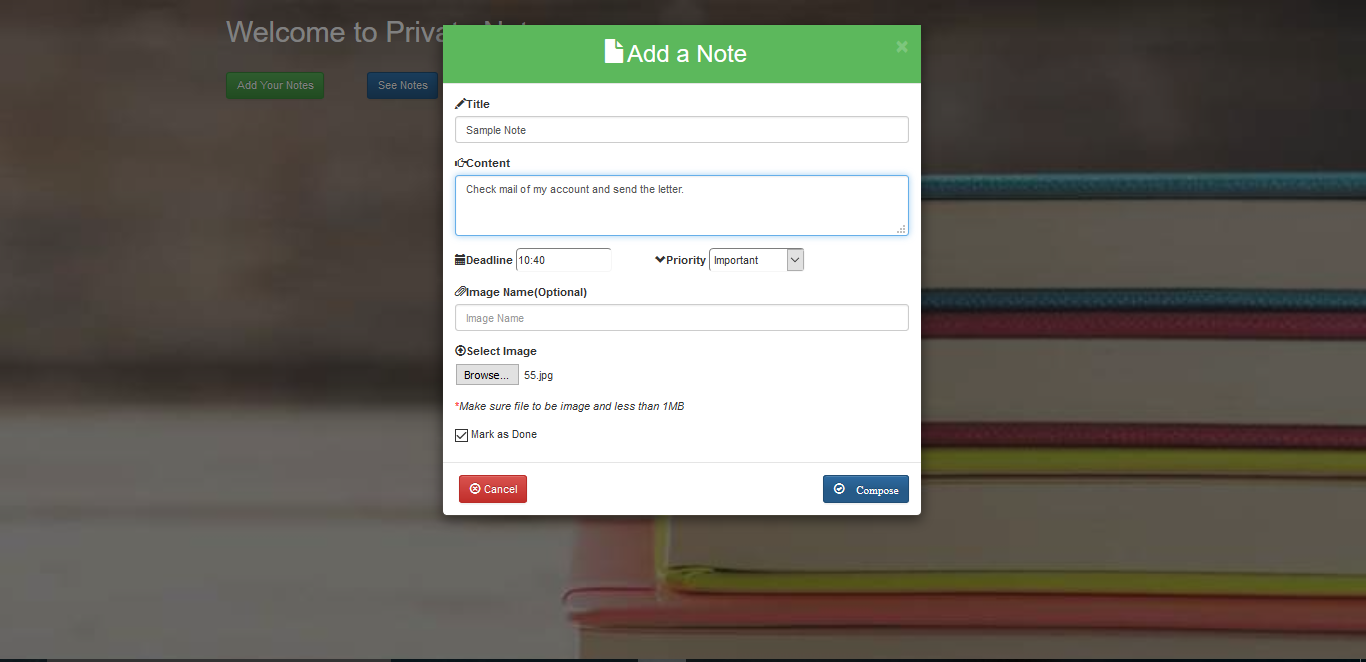
**2. Delete Post**

(database after post deletion)

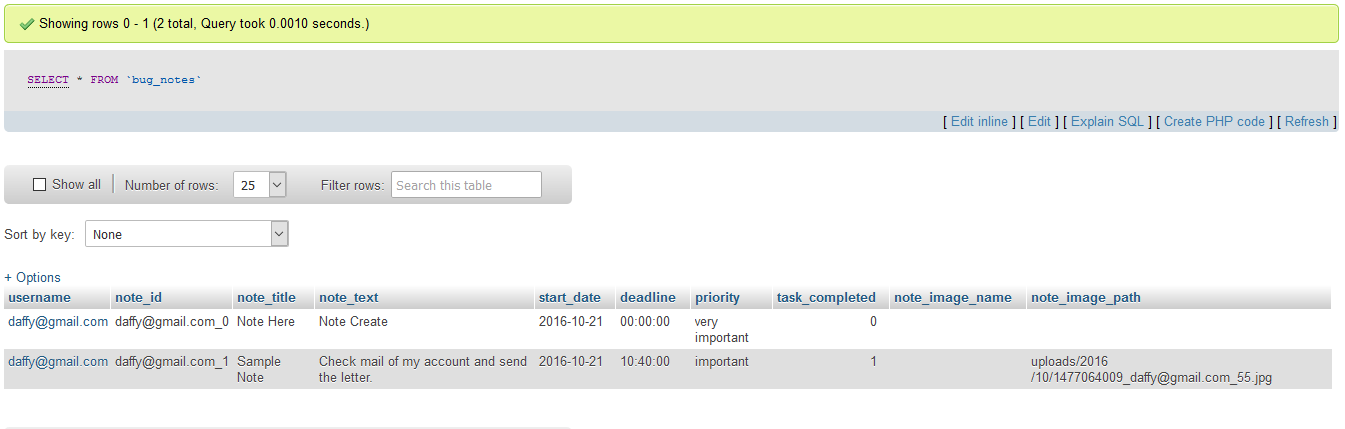
**3. Create Notes**

(database before note creation)

(Create Note)



(Database after Note creation)



**7. Conclusion:**

Hence, the provided document provides an overview of the whole Bug-Microblog system and represent all its functionalities (functional and non-functional) in detail. Also all the various constraints related to the various database concepts like, cardinalities, relational model, ER diagram, etc. are clearly depicted as is described in the software.

Thus the presented software meets all the design constraints as depicted in the SRS and also in strict accordance with the basic database rules.