

ExclusivePic Project Report

Members

Michael Xi
Vinson Lai (1371193)
Michael Rijlaarsdam (1470061)

Introduction

ExclusivePic is a web-based photo sharing database system. It let users to store, view, and search pictures posted by user himself and other user; specifying viewing privilege, and generate data analysis for admin.

Bootstrap templates are used as frontend template to provide a better user interface.

User Management Module(Vinson Lai)

This module helps user to register and login into the system. A single PHP file is used to show

both register and login form to the user. The PHP file which shows this module is "main.php" .

Registration

In case, user wants to register, the form is filled and submitted. After submission of the form, it is

validated used JQuery and shows errors in case any field is blank and do not meet the

requirements like contact number cannot have alphabets. After successful submission of the

form, user goes to "registration.php" where unique username is checked. If username is not unique, user is redirected to the register page.

Query for Checking Unique user name

```
"select count(*) from users where password='$password' AND  
user_name='$username'"
```

Query for Saving data in table users

```
'insert into Users (user_name, password, date_registered)  
values (\''.$user.'\'',\''.$pswd.'\'',SYSDATE)';
```

Query for Saving data in table persons

```
'insert into persons (user_name, first_name, last_name,
address, email, phone) values
(\''.$user.'\'','\''. $fn.'\'','\''. $ln.'\'','\''. $address.'\'','\''. $e
mail.'\'','\''. $phone.'\'')';
```

Security module(Vinson Lai/Michael R)

Login

In case, user wants to login, the form is filled and submitted. After submission of the form, it is validated. After successful submission of the form, user goes to "login_button.php" where username and password is checked. If username and password both matches, user is logged into the system by setting session and redirecting to main.php.

Query for Checking username and password in table users

PHP Code \$databaseQuery>verifyUser(\$username,\$password)

```
"select count(*) from users where password='$password' AND
user_name='$username'"
```

To change groups information the user must be logged In and click the "Groups" tab; where they can create a group and add or remove members from a group.

A user cannot create two groups of the same name. The group is created with PHP CODE>

```
'begin CREATEGROUPPROC(:username_bv,:groupname_bv); end;'
```

Which looks like SQLCODE:

```
CREATE OR REPLACE PROCEDURE CREATEGROUPPROC
```

```
(username IN varchar2, groupname IN varchar2 )
```

```
IS newId INT;
```

```
BEGIN
```

```
newId := seq_group_id.nextVal;
```

```
INSERT INTO groups values(newId,username,groupname,sysdate);
```

```

INSERT INTO group_lists VALUES(newId,username,sysdate, null);

END;

/

```

Which also adds the owner to group_lists.

To add a member a user must provide the name of the group and the name of the member spelled correctly. The php executes a procedure which finds the groupId and checks that the user owns the group, then deletes the member.

PHP CODE:

```

'begin ADDGROUPMEMBERPROC(:username, :memname, :groupname);
end;'

```

SQL CODE:

```

/*Add Group Member*/

CREATE OR REPLACE PROCEDURE ADDGROUPMEMBERPROC

(username IN varchar2, membername IN varchar2, groupname IN
varchar2 )

IS groupId INT;

BEGIN

    select group_id into groupId

    from groups g

    where g.group_name=groupname

        and g.user_name= username;

INSERT INTO group_lists VALUES(groupId,membername,sysdate,
null);

END;

/

```

Similar to add a member; to delete a member from a group the user must specify the group and member name in text, spelled correctly. The php executes a procedure which finds the groupId and checks that the user owns the group, then deletes the member.

PHP CODE:

```
'begin DELGROUPEMEMBERPROC(:username, :memname, :groupname);  
end;'
```

SQL CODE:

```
CREATE OR REPLACE PROCEDURE DELGROUPEMEMBERPROC  
(username IN varchar2, membername IN varchar2, groupname IN varchar2 )  
IS groupId INT;  
BEGIN  
    select group_id into groupId  
    from groups g  
    where g.group_name=groupname  
    and g.user_name= username;  
    DELETE FROM group_lists gl  
    WHERE gl.group_id = groupId  
    AND gl.friend_id = membername;  
END;  
/
```

In all cases a successful operation will return the user to the groups page and an unsuccessful one will print the error message on a blank white screen.

The viewing permissions are handled in the WHERE sections of individual queries with some variance but they are basically

```
SQL CODE: AND ((i.permitted = 1)  
                OR (i.owner_name=$username)  
                OR (gl.friend_id = $username AND i.permitted = gl.group_id)  
                OR ($username='admin')) '';
```

where i is images and gl is group_lists, which checks if the image is public or you're the owner or in the group or the admin

Uploading module--Michael X

Uploading module consists of front end *uploadfrontend.php* and backend *upload.php*. It is capable of uploading one or multiple pictures at the same time.

User will be redirected to login page if the server detected that user is not logged in.

Frontend will collect all information regarding to picture info, and send them to backend for query processing.

SQL statement used:

```
SELECT g.group_id, g.group_name, g.user_name FROM groups
g;
```

```
// Used by admin to populate current group name options.
```

```
//e.g. private, public
```

```
SELECT g.group_id, g.group_name, g.user_name FROM groups
g left outer join group_lists l on g.group_id=l.group_id
WHERE g.user_name=' ' . $user . ' \' or l.friend_id=' ' .
$user . ' \'';
```

```
// Used by non-admin user to pull all the group names
associated with the user
```

```
INSERT INTO images VALUES '
```

```
    . '(' . $curr_id . ',\' ' . $user . '\',\' ' .
    $permitted . '\',\' ' . $subject . '\',\' ' . $place .
    '\','
```

```
    . 'TO_DATE(\' ' . $date . '\', \'yyyy/mm/dd\'),\' ' .
    $description . '\',empty_blob(),empty_blob()) '
```

```
    . 'RETURNING thumbnail, photo INTO :thumbnail,
    :photo';
```

```
// Used to insert pictures and regarding info into
database
```

Major function implemented

getThumbnail() is implemented to convert picture into thumbnail before inserting into database.

Display module--Vinson

In display module, the thumbnail of the images are shown at "main.php", the complete details of images are

shown with original image and top 5 most viewed images are shown after user click on the thumbnail, which brings user to 'display.php'. On homepage of the user logged in system, thumbnail of the uploaded image by user is shown and all images that can be shown to this user are also shown.

For displaying thumbnail image, photo id along with parameter thumb is passed to file getimage.php. In this file, the data of image is extracted using photo Id and according to parameter thumb. Similarly, in case of complete image details, display.php is called which extracts all the data corresponding to the photo id and passes to getimage. The username of logged in user is

checked with the owner of the image, if both are equal, option to delete and edit image is shown.

Query to retrieve photo data

```
"SELECT PHOTO FROM images WHERE photo_id = '$id'"
```

Query to retrieve all photo details

```
if (isset($_REQUEST['subj'])) $query.= " subject =  
'" . $_REQUEST['subj'] . "',";  
  
        if (isset($_REQUEST['date'])) $query.= " timing =  
TO_DATE('" . $_REQUEST['date'] . "', 'MM/DD/YYYY'),";  
  
        if (isset($_REQUEST['place'])) $query.= " place =  
'" . $_REQUEST['place'] . "',";  
  
        if (isset($_REQUEST['desc'])) $query.= " description  
= '" . $_REQUEST['desc'] . "',";  
  
        if (isset($_REQUEST['group'])) $query.= " permitted  
= '" . $_REQUEST['group'] . "'";
```

Whenever display.php is opened, it is checked if the user has seen the image before or not.

If the user has not seen the image before, a new entry into the table photo_visit is made. In this table, photo id and username is stored which tell that the user has seen the image with the

corresponding photo Id.

Query to check if user has seen the image before

```
select * from photo_visit where owner_name = '$user_name' and  
photo_id = '$id'"
```

Query to insert unique views

```
"INSERT into photo_visit(owner_name,photo_id) values (  
'$user_name','$id')"
```

On Main page of user, all of the user info is shown, the photos that can be seen by user is shown by default.

Query for extracting 5 top most viewed images

```
"select t1.photo_id from(select photo_id from (select photo_id,  
count(photo_id) as visits from photo_visit group by photo_id order by visits  
desc) where ROWNUM <=5)t1 inner join (select photo_id from images where  
permitted = '1' or owner_name = '$user_name' or permitted in
```

```
(select group_id from group_lists where friend_id = '$user_name' union  
select group_id from groups where user_name = '$user_name' ))t2 on  
t1.photo_id = t2.photo_id "
```

Search module--Michael R

The search module is composed of the searchPage.php and the searchFunction.php. The page needs keywords, and search order which is by radio button. The function page has two functions to parse the keywords and the date and a third to dynamically create the query based on the variables, start, end, and order.

PHP CODE: function createsearchquery(\$keywords, \$order, \$start, \$end){

```
$username = "".$_SESSION['login_user']."";
```

```
//The big query
```

```
$sql = "SELECT DISTINCT photo_id, SCORE(1)*6+SCORE(2)*3+SCORE(3) AS  
SCORE, timing FROM images i, group_lists gl WHERE (CONTAINS (i.subject,  
".$keywords.", 1) > 0 OR CONTAINS (i.place, ".$keywords.", 2) > 0 OR CONTAINS  
(i.description, ".$keywords.", 3) > 0 ) AND ((i.permitted = 1) OR  
(i.owner_name=$username) OR (gl.friend_id = $username AND i.permitted =  
gl.group_id) OR ($username='admin')) ";
```

```
//Timing Constraints
```

```

if ($start!="") {
    $begin = reformatdate($start);
    $sql = $sql."AND (timing >= ".$begin.") ";
}
if ($end!="") {
    $stop = reformatdate($end);
    $sql = $sql."AND (timing <= ".$stop.") ";
}
//Ordering
switch ($order) {
    case 'norm': $sql = $sql . "ORDER BY SCORE desc"; break;
    case 'new': $sql = $sql . "ORDER BY timing desc"; break;
    case 'old': $sql = $sql . "ORDER BY timing asc"; break;
}return $sql;
}

```

The query is then run and the results displayed.

Data analysis module--Michael X

Admin may log into data analysis page to view current number of users, images, groups and persons stored in the oracle database. Admin may select specific subject, user, time period to view OLAP report.

User need to run **data_cube.sql** after creating all the tables to generate the view DATA_CUBE(OWNER_NAME, SUBJECT, TYEAR, TMONTH, TWEEK). Where TYEAR, TMONTH, TWEEK stands for this year, this month, this week.

cube.php function as a helper backend to process queries for generating table sorted by picture's owner_name, subject, and date.

admin.php function as the front end of data analysis module.

SQL statement used:

```

SELECT COUNT(*) AS NUMBER_OF_ROWS FROM images

// To retrieve current number of images

SELECT COUNT(*) AS NUMBER_OF_ROWS FROM groups

// To retrieve current number of groups

```



```
SELECT COUNT(*) AS NUMBER_OF_ROWS FROM users
```

```
// To retrieve current number of users
```

```
SELECT COUNT(*) AS NUMBER_OF_ROWS FROM persons
```

```
// To retrieve current number of persons
```

In cube.php, queries are dynamically generated by user's selection.
Part of the implementation are listed below.

```
$s = " select ";
```

```
$g = " group by ";
```

```
$w = " where ";
```

```
//Group by User
```

```
if ($user == "All") {
```

```
    $s.= " i.owner_name, ";
```

```
    $g.= " i.owner_name, ";
```

```
}
```

```
elseif ($user != "None"){
```

```
    $s.= " i.owner_name, ";
```

```
    $g.= " i.owner_name, ";
```

```
    $w.= " i.owner_name = '". $user."' and ";
```

```
}
```

```
//Group by Subject
```

```
if ($subj == "All") {
```

```
    $s.= " i.subject, ";
```

```
    $g.= " i.subject, ";
```

```
}
```

```
elseif ($subj != "None"){
```

```
    $s.= " i.subject, ";
```

```
    $g.= " i.subject, ";
```

```

        $w.= " i.subject = '$subj.'" and";
    }
    //Group by Date
    if ($date == "Year") {
        $s .= ' to_char(i.timing,\''yyyy\'') as tyear,';
        $g .= ' to_char(i.timing,\''yyyy\'') ,';
    }
    elseif ($date == "Month") {
        $s .= ' to_char(i.timing,\''yyyy-MON\'') as tmonth,';
        $g .= ' to_char(i.timing,\''yyyy-MON\'') ,';
    }
    elseif ($date == "Week") {
        $s .= ' to_char(i.timing,\''yyyy-MON-IW\'') as tweek,';
        $g .= ' to_char(i.timing,\''yyyy-MON-IW\'') ,';
    }
    else {
    }
    $g = rtrim($g," ,");
    $w = rtrim($w," and");
    $w = rtrim($w," where");
    $stp.= $s.' count(i.photo_id) as image_count ';
    $stp.= " from images i ";

```

Groups

To modify groups you must be signed and then click the groups link in the navigation bar. There are three forms; the top one is to create a group. Make it memorable and easy to type because you cannot create two groups with the same name and you have to type it whenever you add or remove a member

from it. To add or remove members just type the name of the group and the name of the member and click the button. If something goes wrong hit back and try again.

Search

To search be logged in and hit the search button on the navigation bar. The top bar must be filled out without punctuation. So if you want to search for cats and dogs just type “cats dogs” and every image about cats and dogs that you are permitted to see will be displayed. You can click any of them for more info.