Report of PhotoBar

V-Final

GROUP ID: 28

GROUP MEMBER:

HUANG ZILONG 1155026039

YI WEIYING 1155026053

HONG SHIQI 1155026038

WANG BOYU 1155026049

CSE@CUHK

1. INTRODUCTION

1.1 Project Overview

1.2 Objective

1.3 Highlights

1.4 Project statistics

2. SYSTEM ARCHITECTURE DESIGN

2.1 System Architecture

2.2 System Interface Description

3. DETAILED DESCRIPITION OF COMPONENTS

Server Side

User Side

4. USER INTERFACE DESIGN

4.1 Description of the User Interface

4.2 Screen Images

4.3 Objects and Actions

5. TEST

5.1 Test overview

5.n Case-n

1. INTRODUCTION

1.1 Overview

PhotoBar is a powerful Online photo share platform, whose services include image modification and perfection. It is a web application aimed to offer users a favorable environment of photo processing. The main advantages of PhotoBar is user-friendly interface and simplified image process method.

1.2 Objective

a) PhotoBar shall be able to provide service of more than 100 users concurrently
b) PhotoBar shall give every user good and fast online image-processing experience.
c) PhotoBar shall make the photo sharing a wonderful experience and help to connect remote friends, even to make new friends.

1.3 Highlights

a) High loading speed and easy to use
b) Large Photo storage ability (Cloud storage of photos)
c) Mature p2p communication
d) Beautiful User Interface with user friendliness.

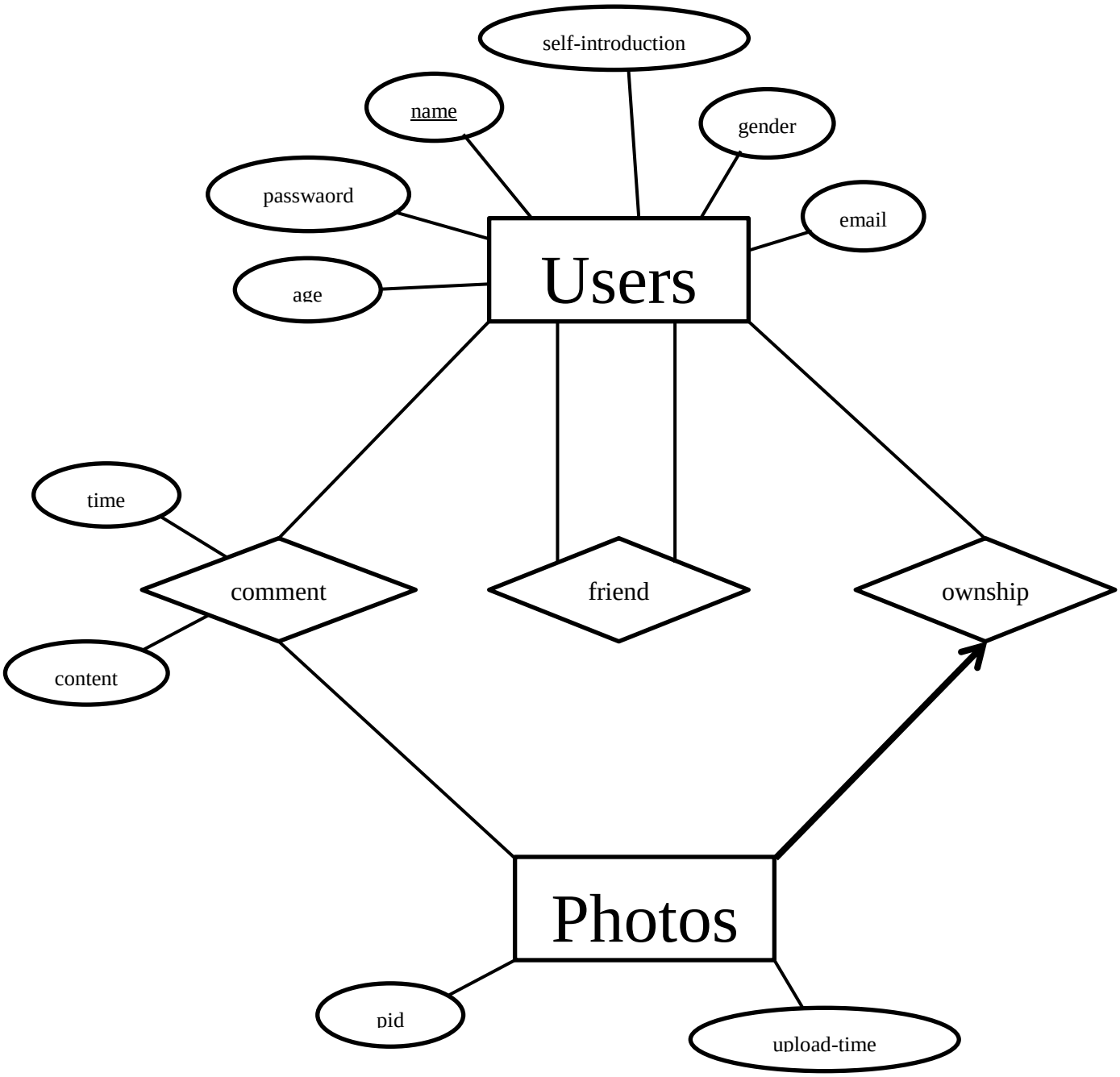
1.4 Project Statistics

LINES OF CODE

Client side has about purely 2,800 lines of javascript code and CSS is about 1,700 lines of code, but taking overlap into consideration, the total number of lines of code should be about 3,000.

Serve side has about 600 lines

ER DIAGRAM


self-introduction
name
gender
passwaord
email
Users
age
time
comment
friend
ownship
content
Photos
pid
upload-time

TEAM MEMBERS' CONTRIBUTION

HUANG ZILONG 1155026039

- He is responsible for designing the overview of the PhotoBar's architecture, the architecture of client side, the communication between server and clients, and writing javascript code.

YI WEIYING 115502605.

- He is responsible for designing the user interface, writing CSS code, finds the Image-Editing API, and doing testing.

HONG SHIQI 1155026038

- She is responsible for designing the user interface, writing PHP and CSS code, find a very useful PHP code to upload file, and doing testing.

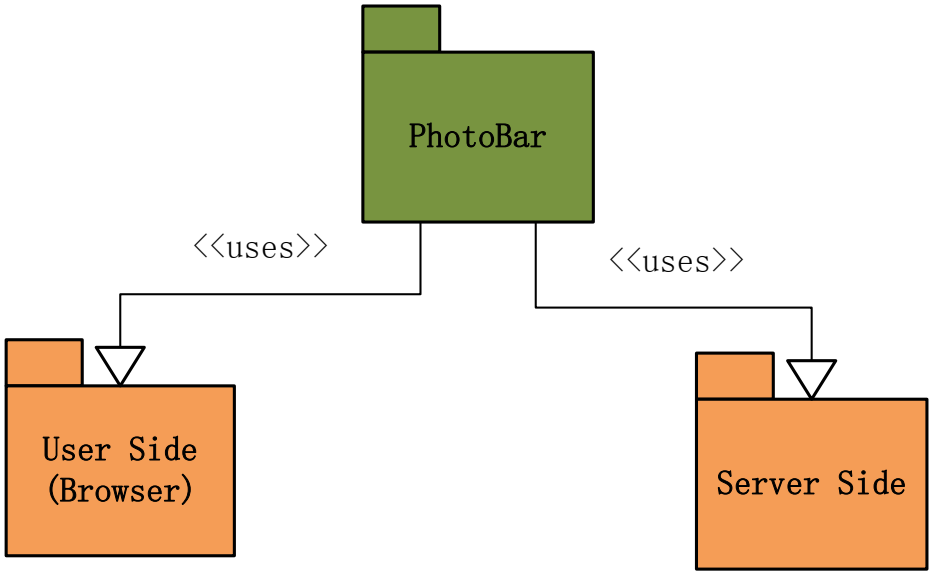
WANG BOYU 1155026049

- He is responsible for designing ER diagram for database and architecture of server side, writing PHP code.

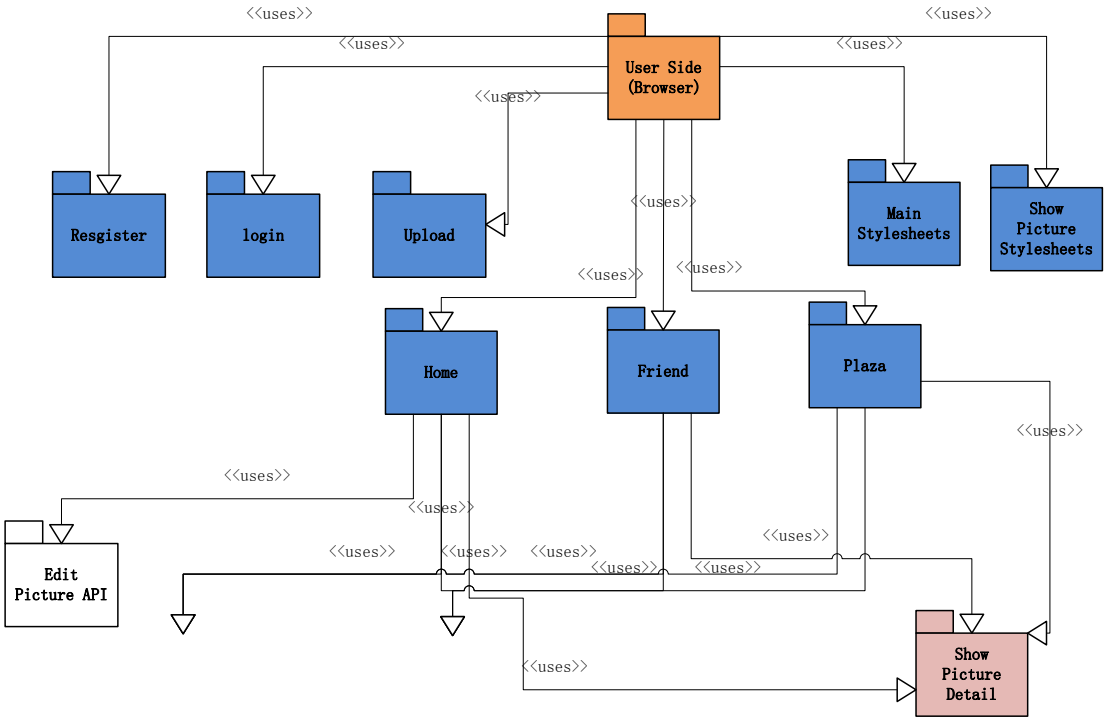
2. SYSTEM ARCHRITECTURE

2.1 System Architecture

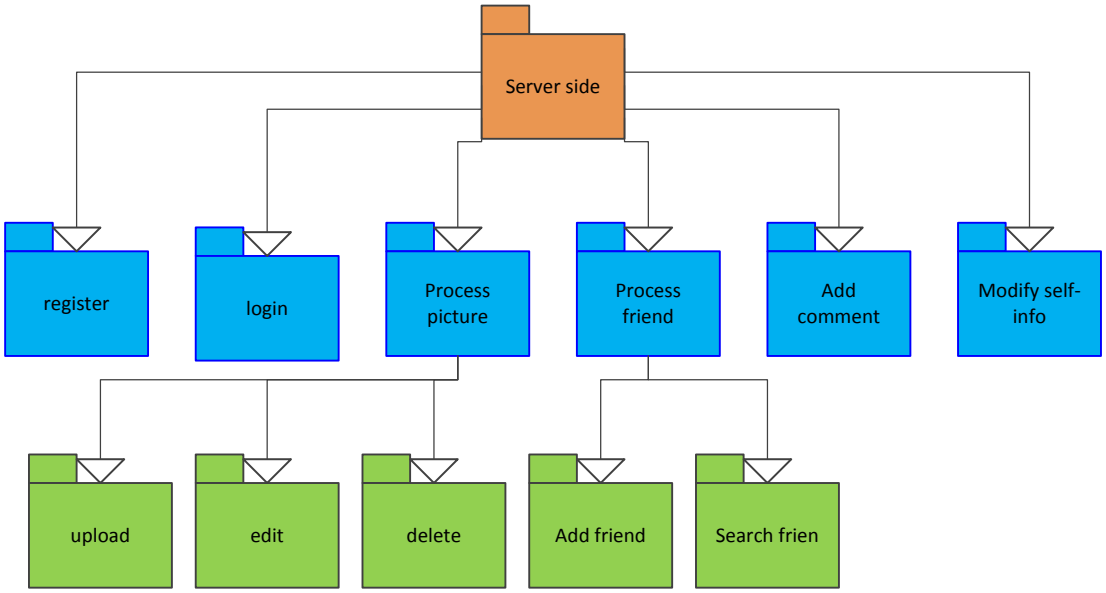
2.1.1 OVERALL VIEW



2.1.2.1 CLIENT SIDE OVERVIEW



2.1.2.2 SERVER SIDE OVERVIEW



2.2 System Interface Description

Key functions:

init()

This function is called in the very beginning to determine the size of main html elements.

SearchFriend()

This function uses AJAX to request a search of username in the server, and when result is returned from server, it display those result.

LoadMineInfo()

This function uses AJAX to request current user information in the server, and when result is returned from server, it display those result.

LoadFriendInfo()

This function uses AJAX to request current user's friends' information in the server, and when result is returned from server, it display those result.

ShowComment()

This function uses AJAX to request the comments about current picture in the server, and when result is returned from server, it display those result.

Upload()

This function uploads photos, changes the name of photo to username+photoname, automatically generates a thumb of these photos and saves them into the database respectively.

Plaza()

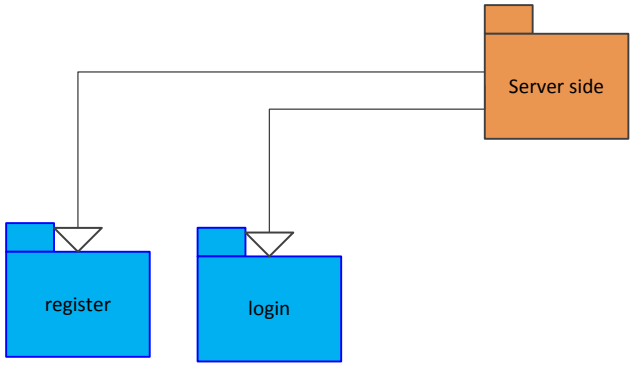
This function searches all database and display all the details (user, user's icon, comments, upload time) of the latest ten uploaded photo

3. DETAILED DESCRIPTION OF COMPONENTS

Server Side

Login and Resgister

Diagram



Functionality

Login component receives users input name and password, checks if they match. If match, save users session and return success, else return false.

Register component, much like the login component, it receives user's input information and store them into the database

Procedure and Function

login():

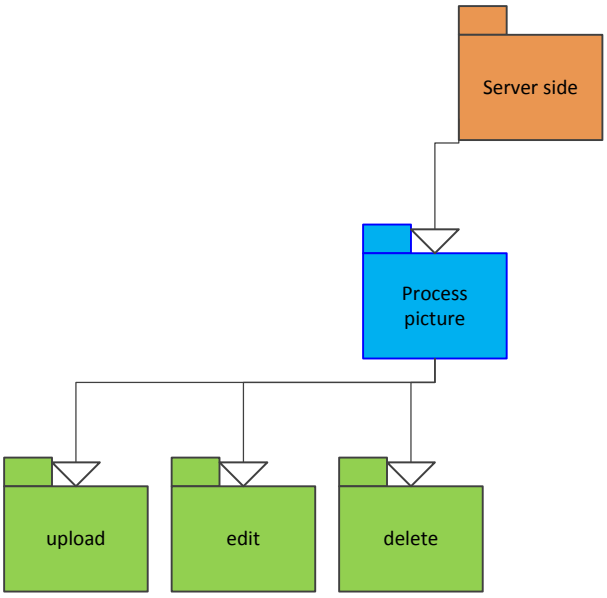
This function receives the user's username and password, and checks their match. If the password is correct and the user does exist, then login() function will send a correct message to

client, else return error.

register():

This function gets the user's basic information and inserts them to the database, if the username has not been used, return success, otherwise it will return error to client

Process picture



Functionality

Process picture part is the core of PhotoBar. This part take charge of photo uploading, editing and deleting.

Procedure and Function

Upload():

This function realizes photo uploading function. User can upload a photo by click upload button and select photo path or drag the target photo directly to uploading zone.

After that, photos will be stored in the user's correspondent folder. Also, a correspondent thumb will be generated automatically with file name as 'thumb + file name'.

Edit ():

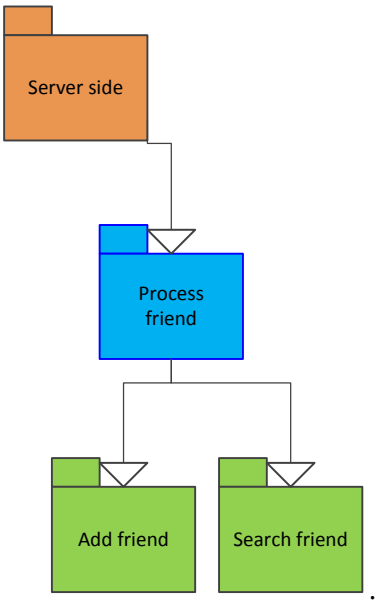
This function is the core of photo processing. We utilized a web API to obtain professional and less defected effect.

Users are able to edit their own photos using edit function. If he/she wants to save it, the edited photo will be stored to user folder with name 'new_filename' instead of replacing the old one directly.

Delete ():

This function implements photo deleting. If a user deleted a photo, all records of this photo, like comments, uploading time, will be removed from database. As a consequence, there will be no information of this photo showed in the website.

Process friend



Functionality

Friend component is a social element. It is can realize interaction between users. This functionality makes PhotoBar become more diverse and attractive since knowing friends' life is quite a interesting idea.

Procedure and Function

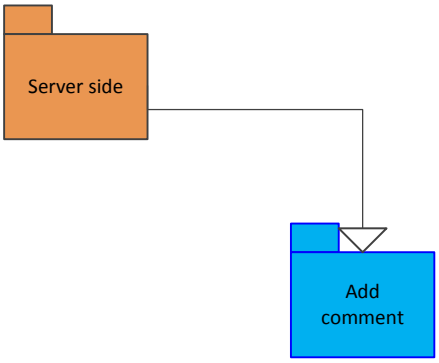
Search friend( ), Add friend( ):

User can search a friend by its username. If a user A wants to add another user B as a friend, user A can type B's name in search bar and after all the similar user names listed, user A can choose the one he/she wants to add.

Another place can add friend is in the plaza. If a user is interested in one photo, he/she may want to add the user who uploaded it as friend. Since we provide the add button under each photo, user is able to add the target user directly.

After adding friend successfully, the friend information will be linked to the user in database.

Add comment

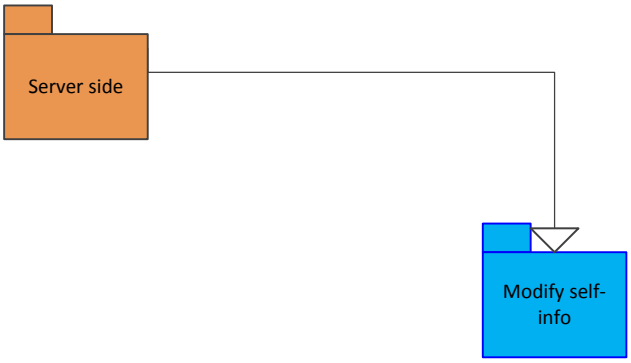


Functionality

This component realizes comments adding function. If a user wants to add comment on a photo, he/she can type a comment under a photo.

If a photo be commented, the comment will be added to the correspondent position of database, thus when other users explore this photo later, they will see the comments.

Modify Self-info



Functionality

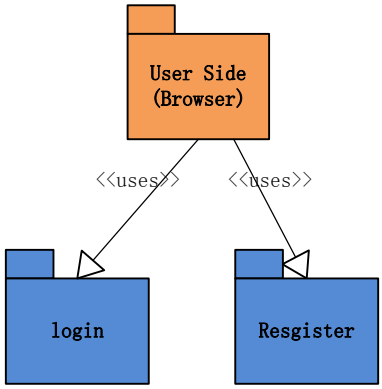
It is about self-information modification ability. When a user first register an account, he/she may not input the user-information. After they have been using PhotoBar for quite a while and add some friends, he/she may want to change his/her information to complete his/her profile. Then he/she can click modify self-info button to do it. The limited number of self-info is 1024 characters.

Also, if a user wants to change his/her icon, he can set an uploaded photo as his/her icon immediately since we have a ‘set as icon’ button under each photos.

Client Side

Login and Resgister

Diagram



Functionality

Login component gives clients the first impression with its ever-changing background. It's designed to check the current internet user's identity.

Register component, much like the login component, it's designed for new user to register their initial information like username, password, birthday etc.

Procedure and Function

login():

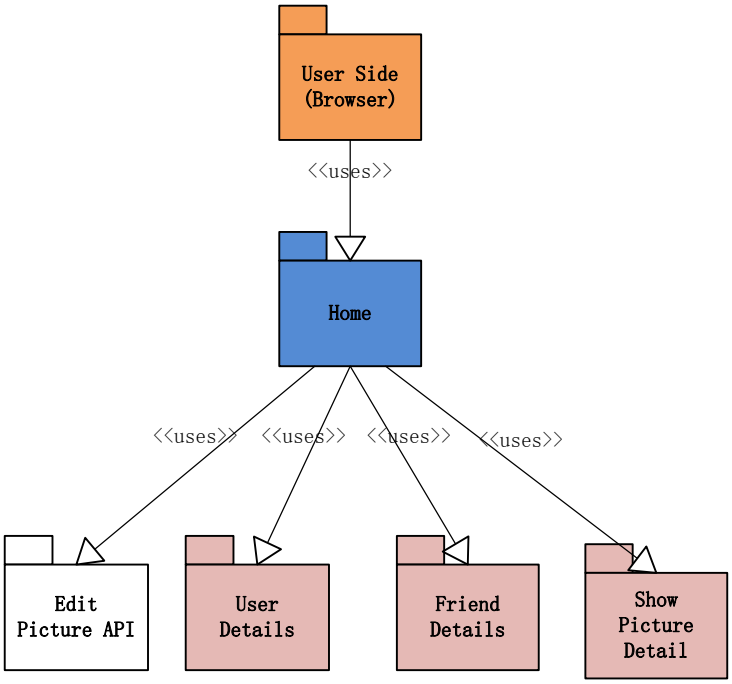
This function gets the user's username and password, and submits them to the server for verification. If the password is correct and the user does exist, then login() function will make the webpage jump to the client's home page, otherwise an alert will appear.

register():

This function gets the user's basic information and submits them to the server, if the username has not been used, a new account is created, otherwise it will display an alert.

Home

Diagram



Functionality:

Home component, much like an intermediate, will display the client home page after a client login successfully, and then interact with client, invoke basic function like ShowUserDetail(), SearchFriend(), ShowPictureDeatail() to retrieve data from the server and display the data accordingly and neatly.

Only through this component can a client invoke the Edit Picture API, because a client doesn't have the permission to edit others' pictures.

Besides using lower level component, it also has some special functions as listed below.

Procedure and Functions:

init()

This procedure will get the information about the current operating environment, like how wide the screen is, and then determined the size of the main html tag in order to display a ordered user interface.

Open()

This procedure will pop a webpage for client to upload picture, and the newly created tab is of special features like no scrollbar is allowed, no address bar, no menu bar etc, to focus on the upload feature.

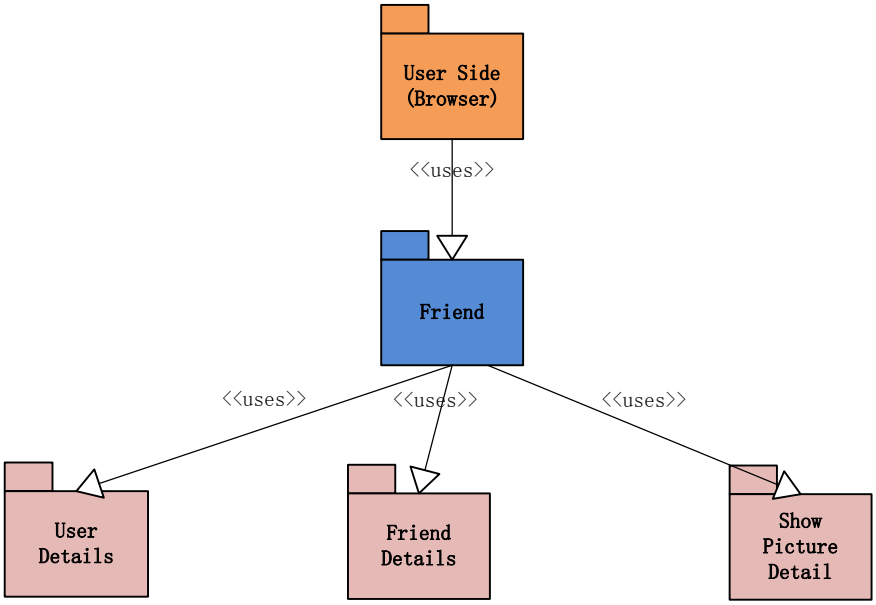
Jump()

This procedure will jump to another html page and pass current client information like ‘username’ and ‘friend’s username’ though the URL.

E.g, if a client want to visit a friend’s home page, them jump() procedure will be used and the browser will change the URL to “***friend.html?username=[username]&friend=[friend’s username]***”

Friend

Diagram



Functionality:

Friend Component is still like an intermediate.

What's special is that it reads friend's name from the URL, and pass it as the parameter to ShowUserDetail() so that in the '***Friend.html***' ,although the same client is using the ShowUserDetail() component, the webpage will not display current client's information but instead the targeted friend's information.

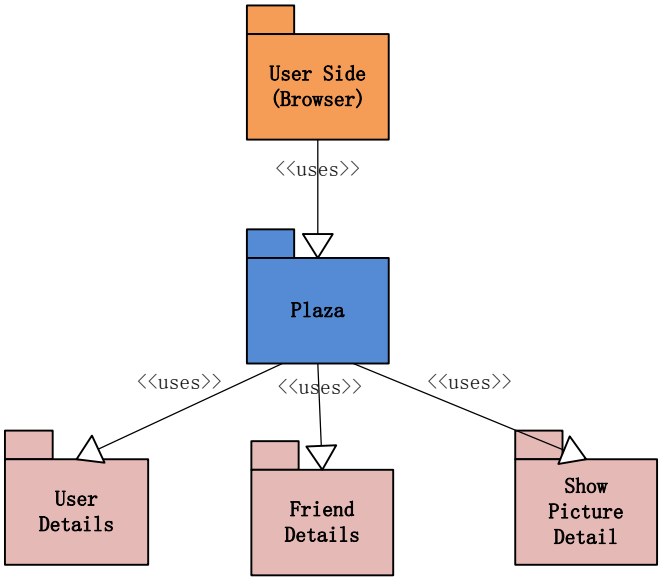
Procedure and functions:

LoadCurrentFriendInfo():

This function gets the friend's username and uses it with the ShowUserDetails() to read the friend's detailed information.

Plaza

Diagram



Functionality

Plaza Component is designed for clients to know more about strangers. The lastest uploaded 10 pictures will be shown with their uploaders in this component, and all clients are able to add those uploaders as his/her friend.

Procedure and functions

LoadPlaza():

This function will get the latest 10 pictures from the server and display in in the main frame, showing the upload time, uploader's basic information etc. And this function set some special feature different from Home Component and Friend Component when it uses the same component SearchFriend() and ShowPictureDetial().

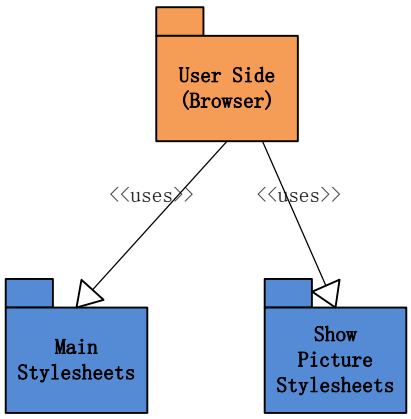
E.g:

If the client wants to add this uploader as a friend, he can click the button "add", which will make use of the SearchFriend()'s addFriend Function to submit the request to the server.

If the client wants to view more details of the picture, he can click on the picture, which will uses ShowPictureDetail() with different parameters when this component is used by the Home Component, to pop out a new tab.

MainStyleSheets and ShowPictureStyleSheets

Diagram



Functionality

These two components are actually two cascade stylesheets file, combined with the Home, Friend, Plaza, Login, Upload, Register Component, it can help to determine different display style, such as the look of some buttons, and scrollbars etc.

It makes use of some special functions supported by ‘webkit’ to give a more comfortable look on the webpage.

Procedure and Functions

These two components don’t have functions or procedure. Instead they defined some key class and id cascade stylesheet.

Eg:

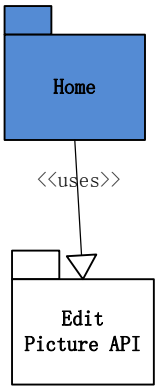
```
 .Images_Container
{
        background:#FFFAFA;
        border-top: 2px solid #45ada9;
        width: 98%;
        margin: 0 auto;
}
.Images_Container    img
{
        max-width: 100%;
```

```
        max-height: 100%;
        margin: 0px auto;
        -webkit-border-top-left-radius:12px;
        -webkit-border-top-right-radius:12px;
        -webkit-border-bottom-left-radius:12px;
        -webkit-border-bottom-right-radius:12px;
        border-style:inset;

}
.navi_button
{
        -webkit-border-bottom-left-radius:15px;
        -webkit-border-top-left-radius:15px;
        -webkit-border-top-right-radius:15px;
        border-width:0px 3px 3px 0px;
        border-style: solid;
        border-color: #FFB6C1;
        background-color:#fff6d4;
        font-family:Britannic Bold;
        font-size:x-large;

}
#left_list
{
        -webkit-border-top-left-radius:15px;
        -webkit-border-top-right-radius:15px;
        background: #FFF5EE;
        position:fixed;
        top:16%;
        bottom:0%;
        border:3px solid #45ada9;
        margin:0px 0px 0px 20px;
        float:left;
        overflow-y:auto;
        overflow-x:hidden;
        height:82%;
        width:300px;
}
#left_list::-webkit-scrollbar {
width:14px;
height:14px;
background:-webkit-gradient(linear,left                                                top,right
top,color-stop(0%,rgba(202,202,202,0.07)),color-stop(100%,rgba(229,229,229,0.07)));
background:-webkit-linear-gradient(left,rgba(202,202,202,0.07)    0%,rgba(229,229,229,0.07)
```

```
100%);
background-color:rgba(229,229,229,.3);
-webkit-box-shadow:0 0 1px 0 rgba(0,0,0,.15) inset,0 1px 0 0 #fff;
box-shadow:0 0 1px    0 rgba(0,0,0,.15) inset,0 1px 0 0 #fff;
overflow:visible;
border-radius:10px;
border:solid 1px #A6A6A6;
}
```

Edit Picture API

Diagram



Functionality

This is a API to an online picture editing application, it reads the URL of the picture on our server, and transfer to the online application. After editing, the application will store the picture on its server, so we should be able to retrieve the picture from its server to our server, and store the picture locally.

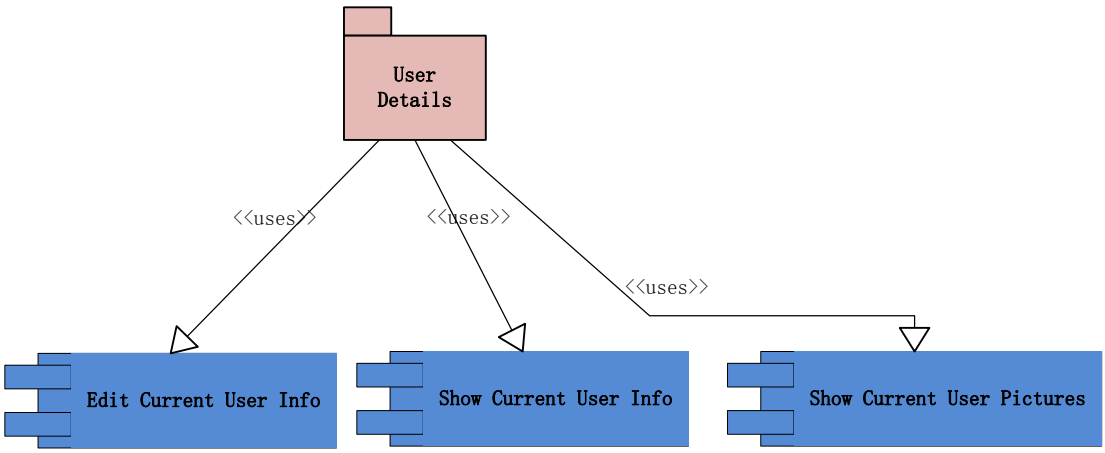
Function and Procedure

SaveNewPic():

This function reads the URL of the modified picture store on the online application's server, and submit it to our server in order to let our server store the modified picture locally.

UserDetails

Diagram



Functionality

This UserInfo can display user's information, user's all pictures and edit user's information

Procedure and Functions

EditCurrentUserInfo()

This function can create a html form tag according to the current user's information, and user is able to change his/her self-introduction, age etc. If the user decide to give up current changes, this function will restore the previous information, if the user decide to submit the change, this function will send a POST HTTP request to the server with new user information. And every time a new user information form is submitted, it will called ShowCurrentUserInfo() to update the user information in the HTML file.

ShowCurrentUserInfo()

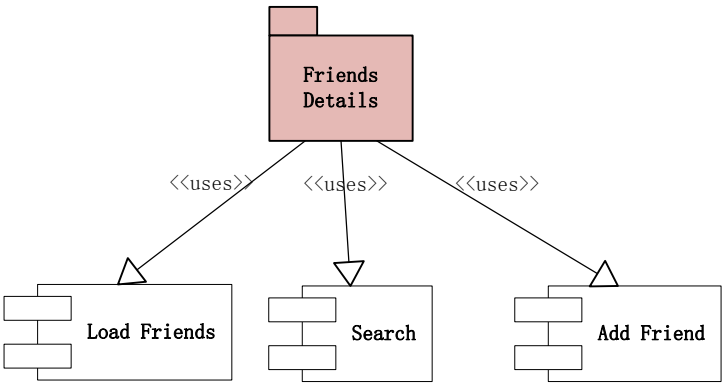
At the initializing stage of the webpage, this function sends a POST HTTP request to the server, and the server shall respond will read the current login username from PHP Session and return this function with current user information. Then this function will dynamically change the content of the current HTML file to display the user information.

ShowCurrentUserPicture()

This function sends a POST HTTP request to the server, and the server responds with all picture id of currently login user. Then this function will display all the picture with their thumb one in order to increase the loading speed. Also it add event listener to every <img> tag to ensure every time the <img> tag is clicked, details of the picture will be shown.

Friends Details

Diagram



Functionality

This component will load current user's friends' details when the webpage is initializing. When the search button at the left is clicked, this component will search for a user with input as username.When the Add button of a search result is clicked, this component will send a POST HTTP request to the server to establish a friendship with the targeted user in the search result.

Procedure and functions

LoadFriends()

This function will be invoked at the beginning and it will send a POST method to the server to retrieve current user's friends' details. Then it will display the return data in the left list orderly.

Search()

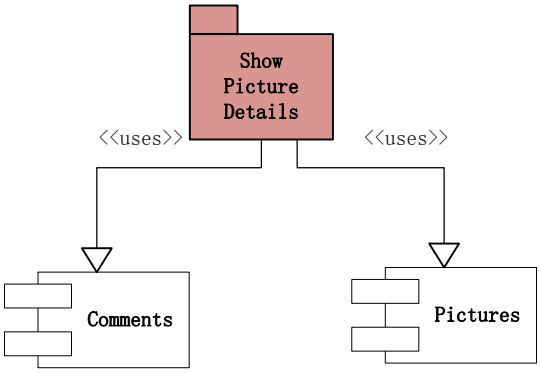
This function will read the input of the <input> tag and use AJAX to send a POST HTTP request to the server to check whether there is corresponding user with relative username. Then it will display the result in the left list.

AddFriend()

This function will read which button is clicked and determine whom the current user wants to become friend of. Then it uses AJAX POST HTTP request to tell the server about current client's make friends' decision.

Show Picture Detail

Diagram



Functionality

This component will show the targeted picture details differently according to different setting when it's used.

For the Home Component, this component will show the picture with comments and some editing button in the original webpage in the middle.

For the Friend Component, this component will show the picture with comments but without editing button in the original webpage in the middle.

For the Plaza Component, this component will show the picture with comments but without editing button in a newly created tab.

Procedure and functions

ChangePic()

This function will read the id of the targeted picture and display it in the middle or open a new tab to show detailed information.

showComment()

This function will read the id of the targeted picture and uses AJAX to send a POST HTTP request to the server in order to retrieve comments' information from the server. Then it displays the comments on the screen.

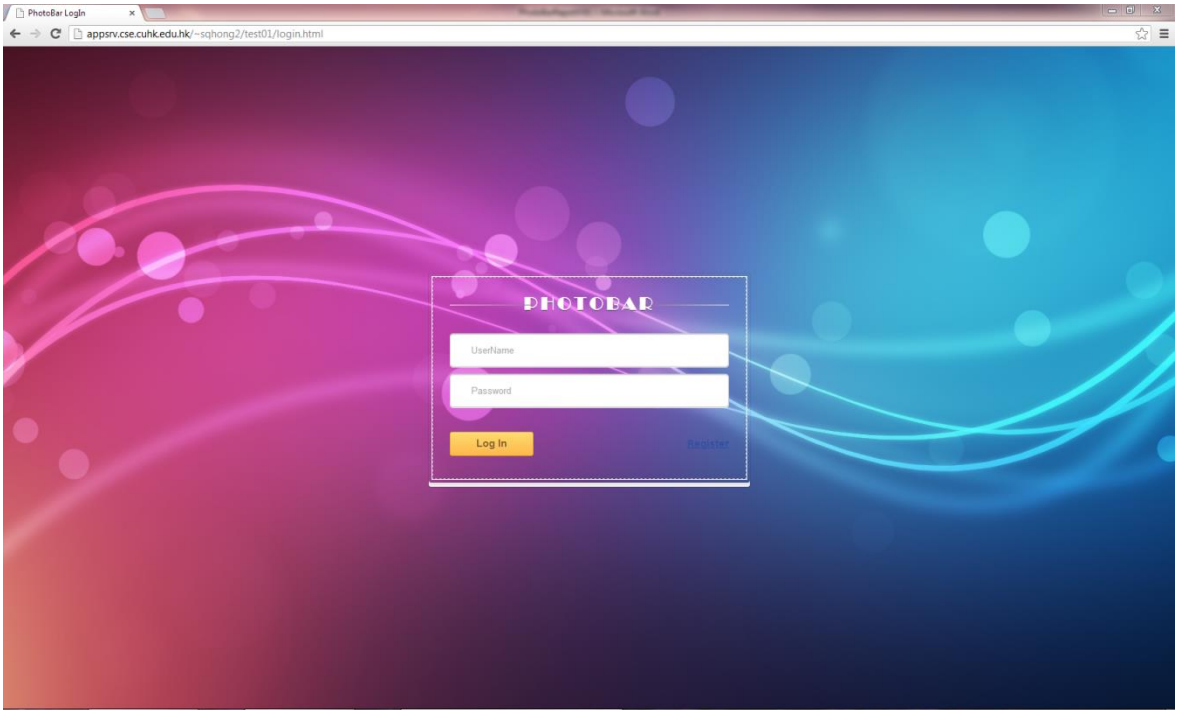
AddComment()

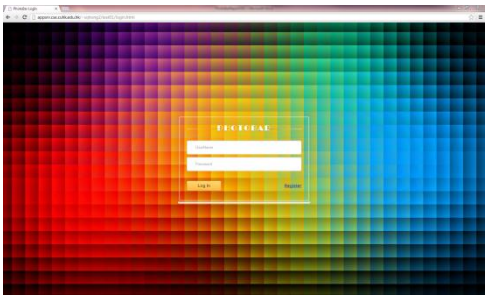
This function will read the input of the client and check whether is valid, if so, a POST HTTP request is sent to the server to add a new comment to the currently displayed picture.
In the end, it will invoked showComment() to keep the local comments updated with the server.

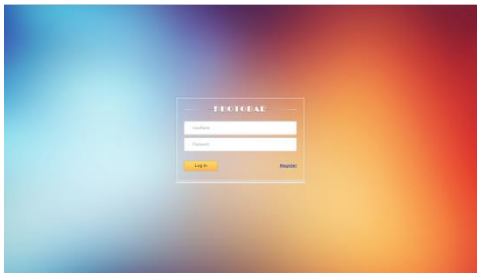
4. USER INTERFACE DESIGN

login.html

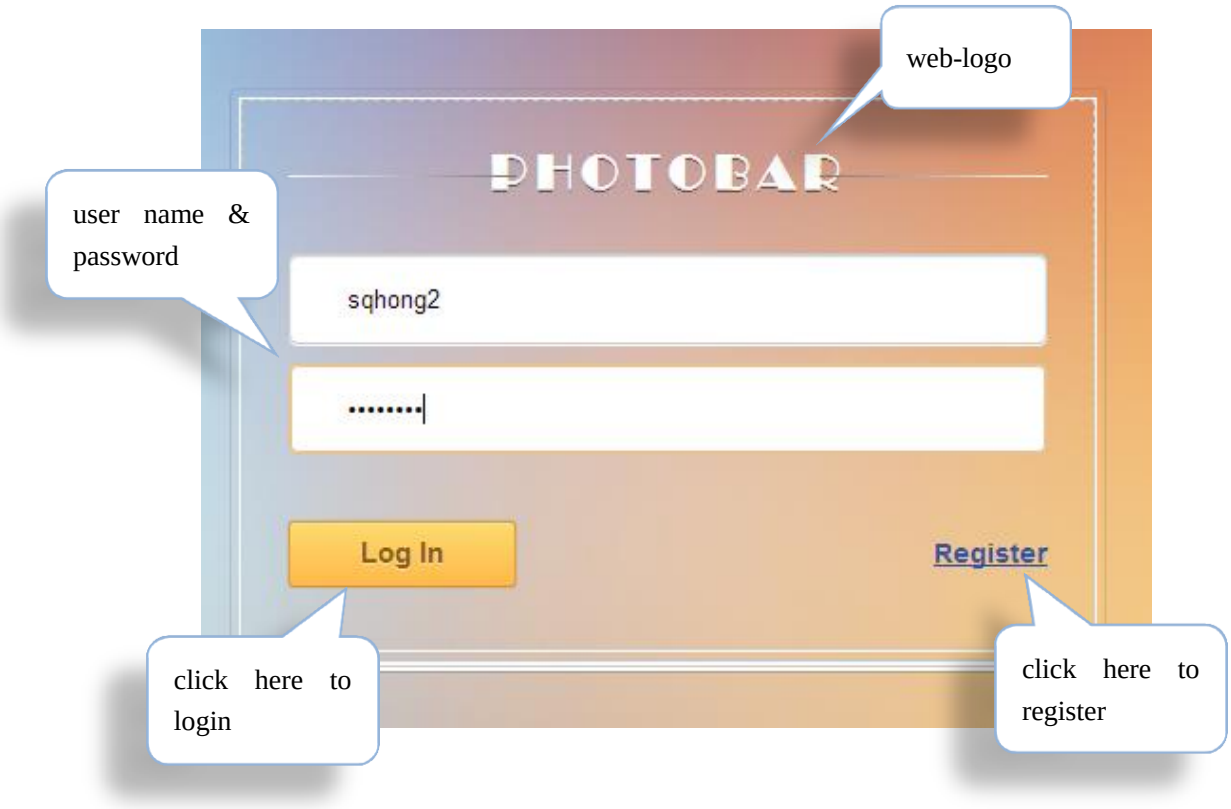
There are 5 random show up background of login page.



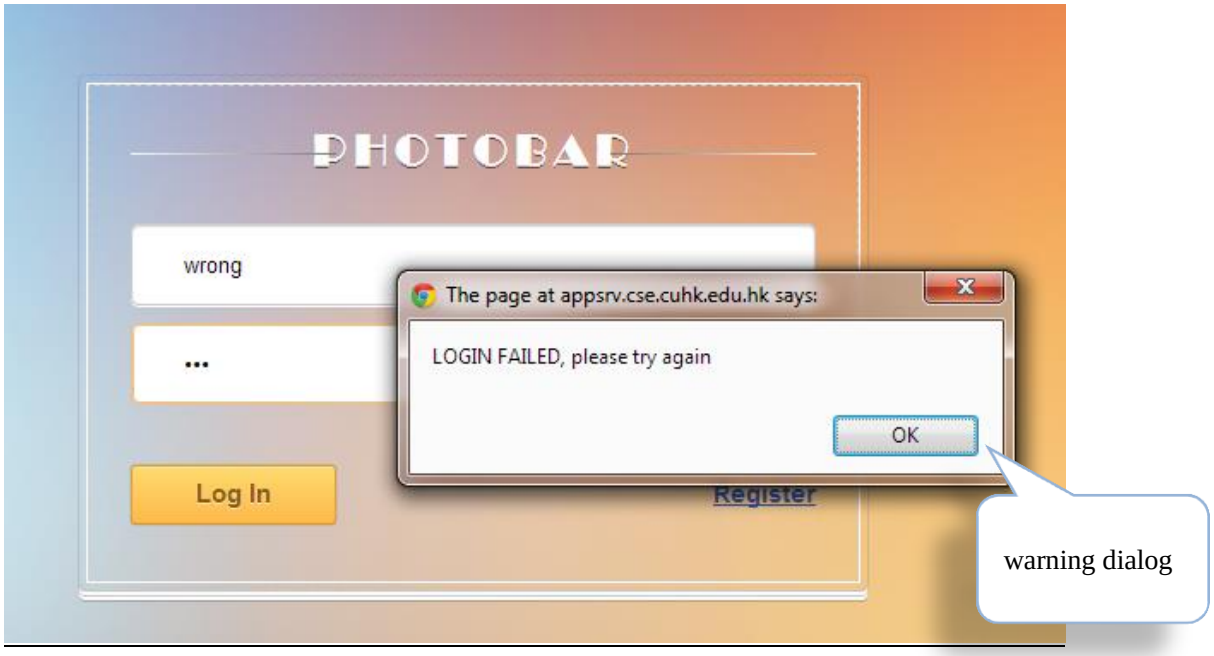




Detail login page

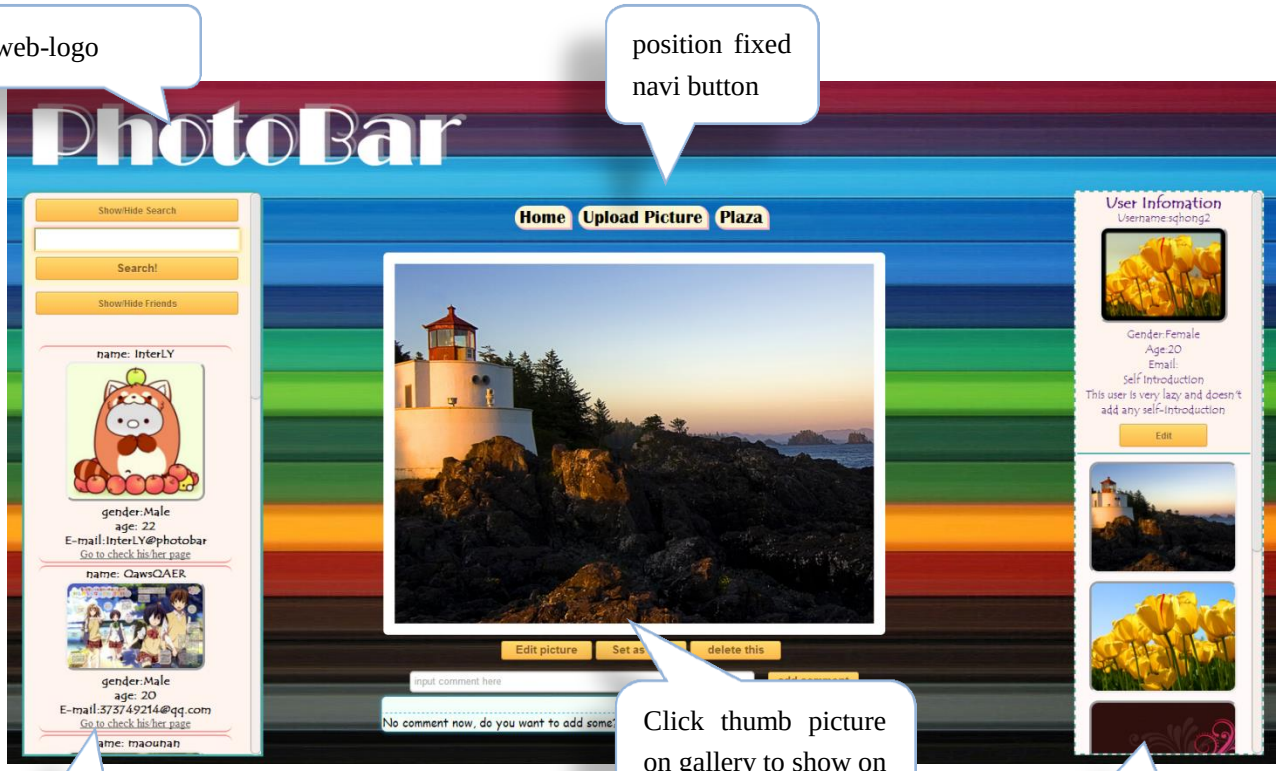


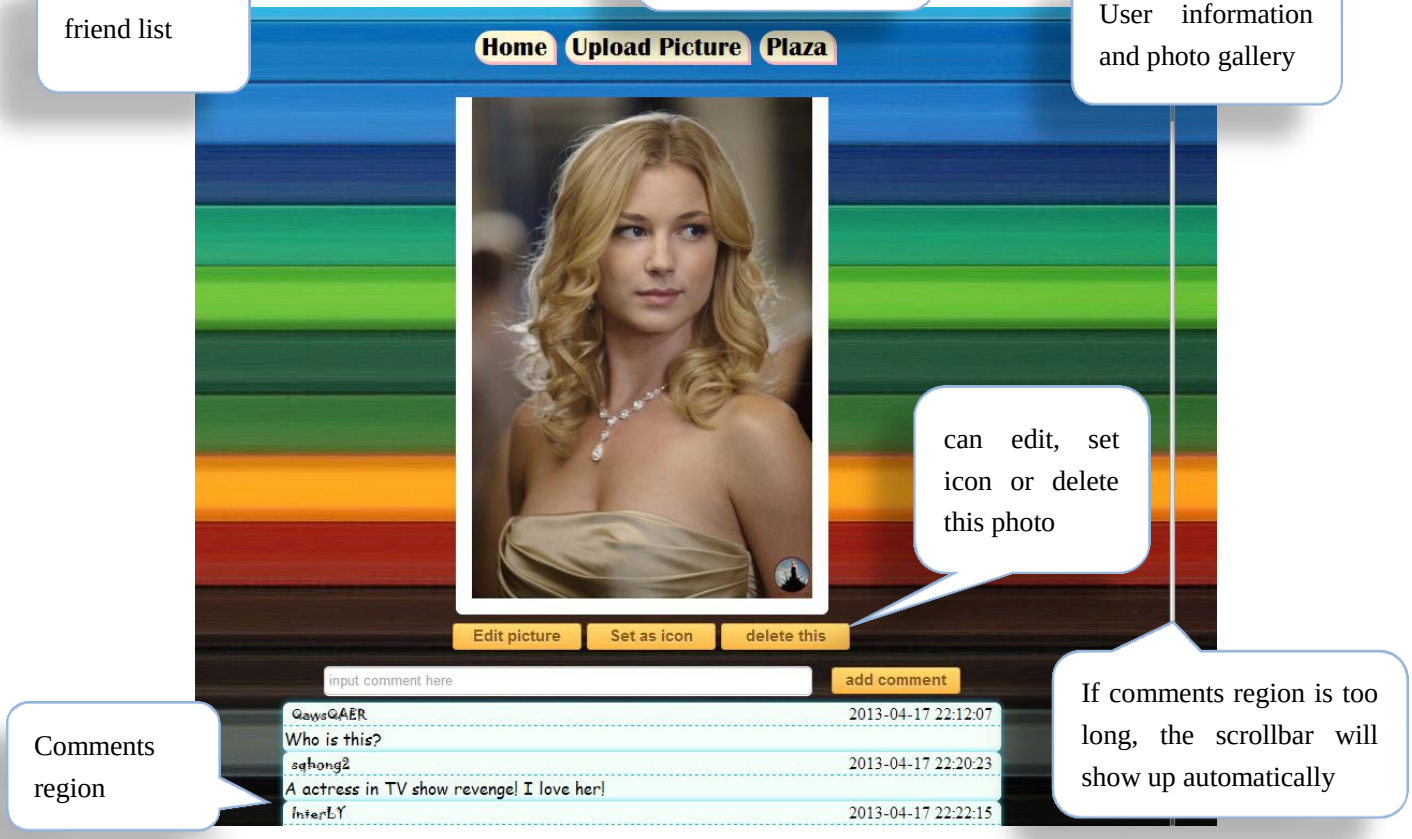
If user enters the wrong username or password, the web will show a warning dialog.

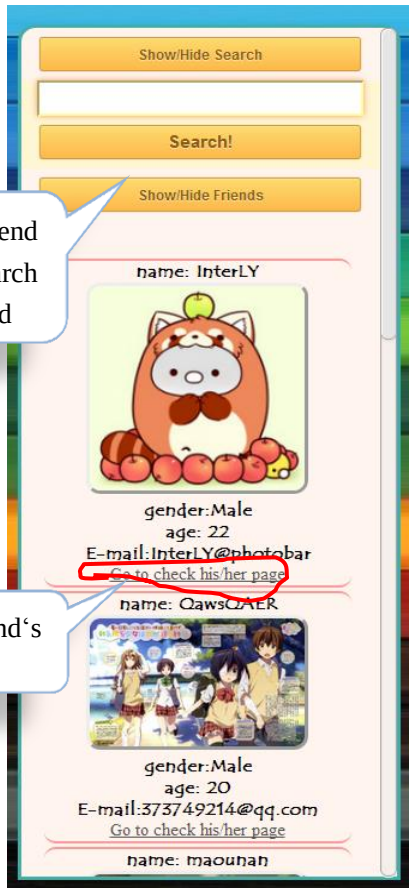


After user click login or type 'enter' on keyboard with correct username and pass word, it will show the main page of the user.

index.html

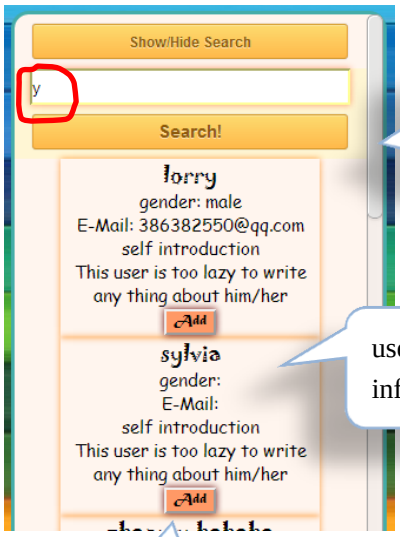






can hide friend list and search for new friend

link to friend's homepage



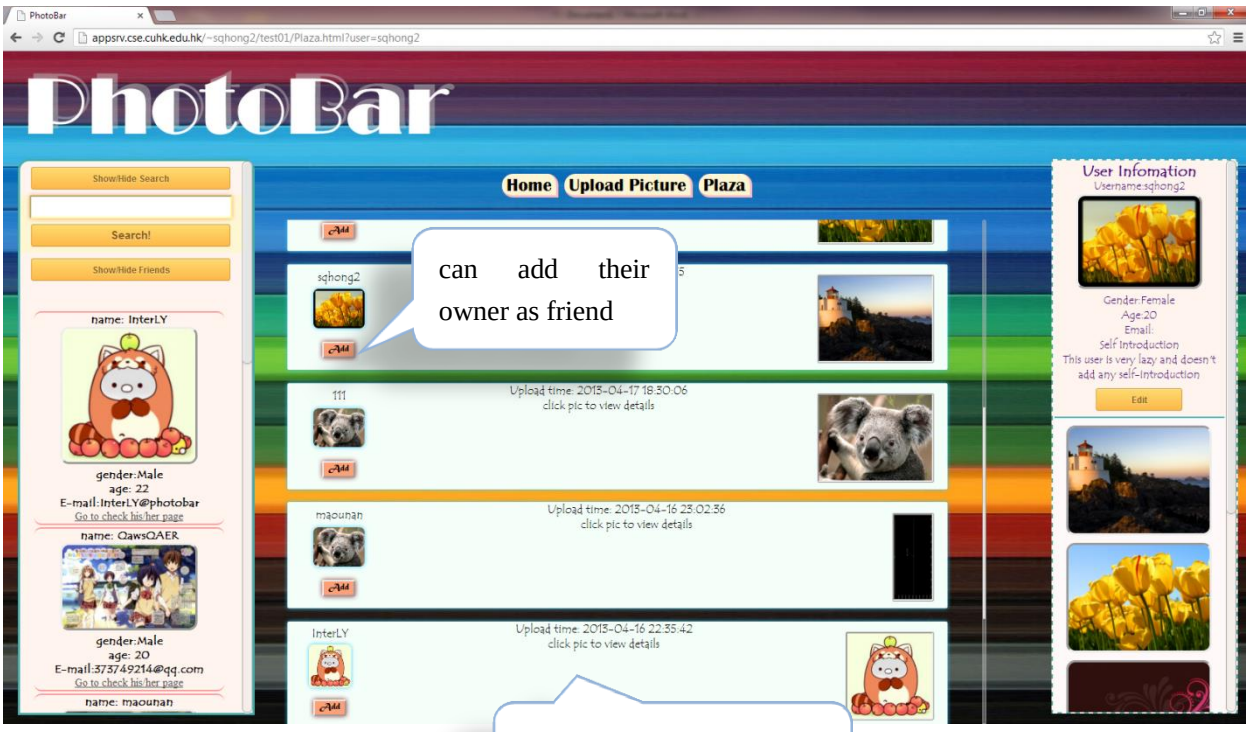
auto-show-up scroll bar

user information

search result for keyword 'y'. click add to add friend.

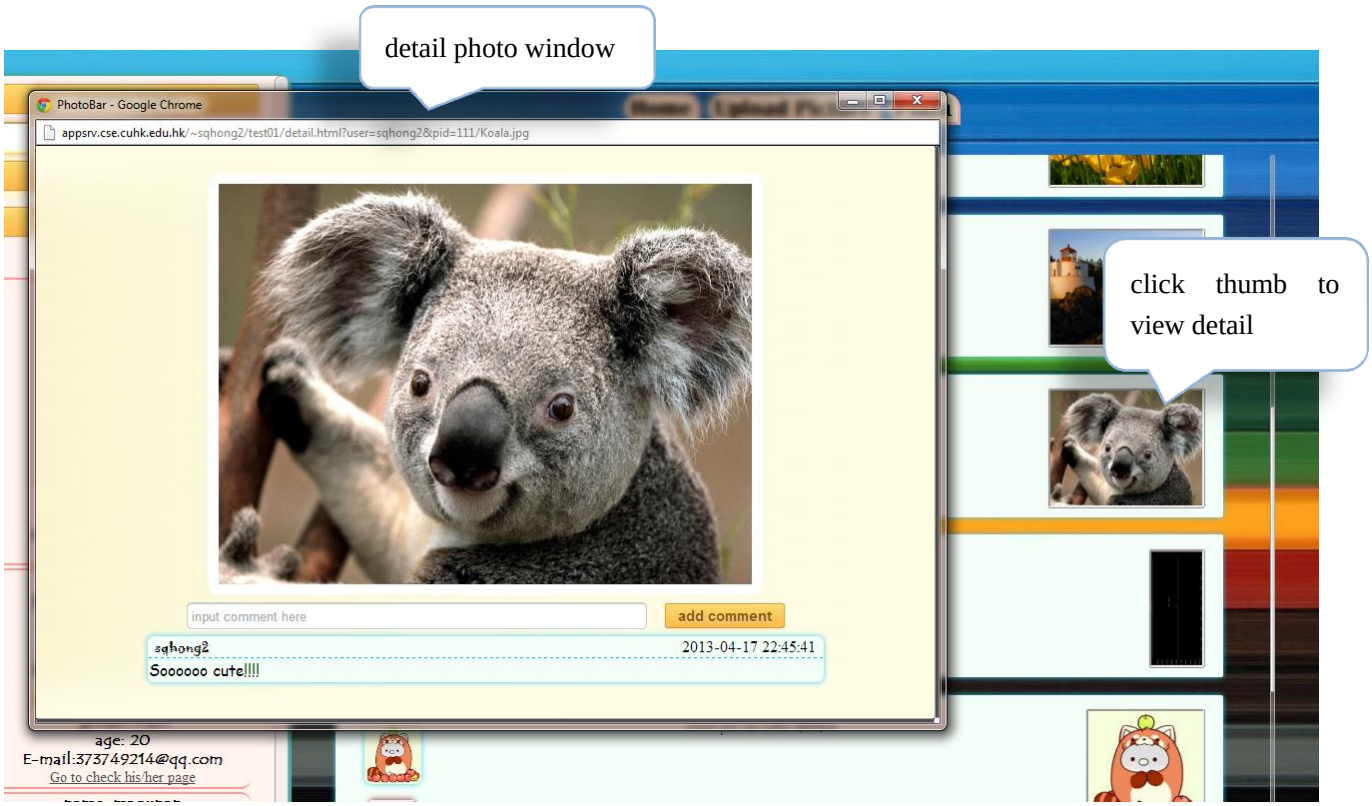
user can see the most new uploaded photos of all photobar users on plaza page.

plaza.html



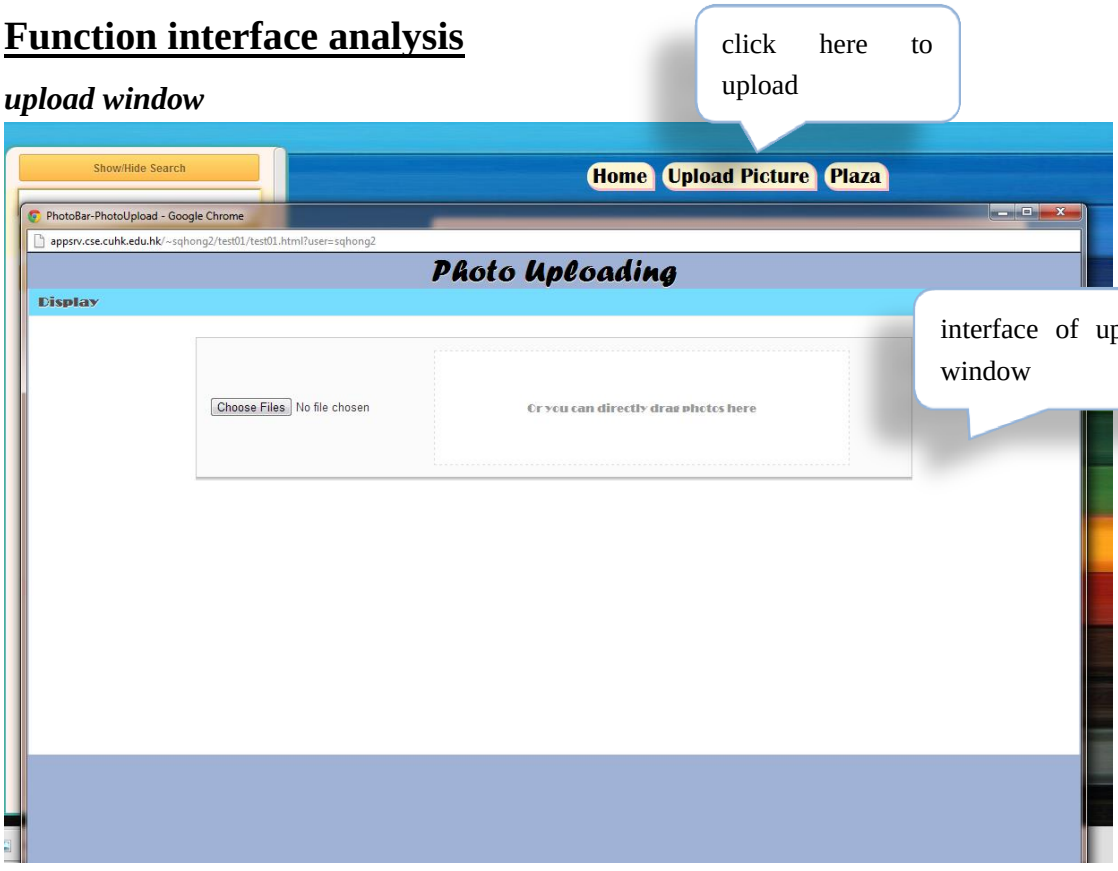
can add their owner as friend

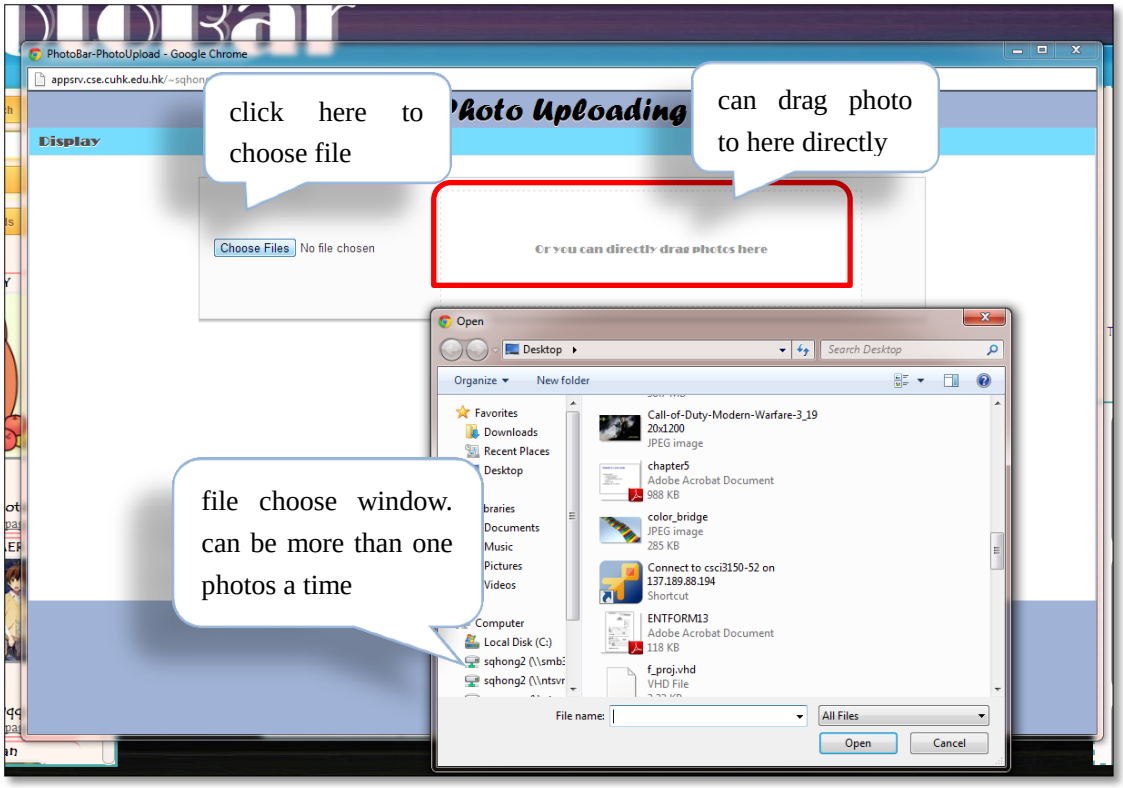
newest uploaded photos and their information



Function interface analysis

upload window




PhotoBar-PhotoUpload - Google Chrome
Photo Uploading
Display
Choose Files
No file chosen
Or you can directly drag photos here
click here to choose file
can drag photo to here directly
file choose window. can be more than one photos a time
Open
Desktop
Search Desktop
Organize
New folder
Favorites
Downloads
Recent Places
Desktop
Libraries
Documents
Music
Pictures
Videos
Computer
Local Disk (C:)
Call-of-Duty-Modern-Warfare-3_19
20x1200
JPEG image
chapter5
Adobe Acrobat Document
988 KB
color_bridge
JPEG image
285 KB
Connect to csci3150-52 on
137.189.88.194
Shortcut
ENTFORM13
Adobe Acrobat Document
118 KB
f_proj.vhd
VHD File
File name:
All Files
Open
Cancel

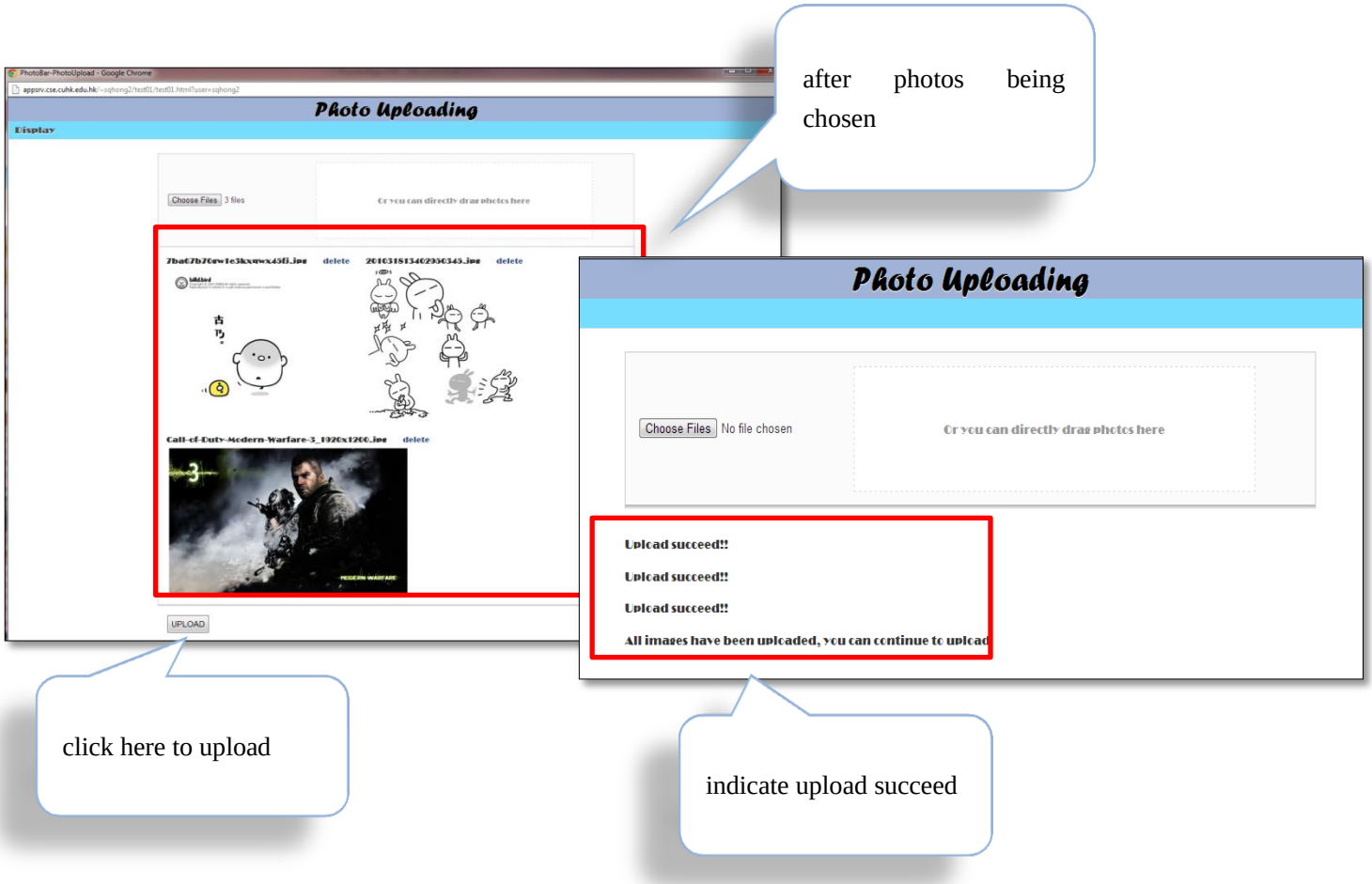
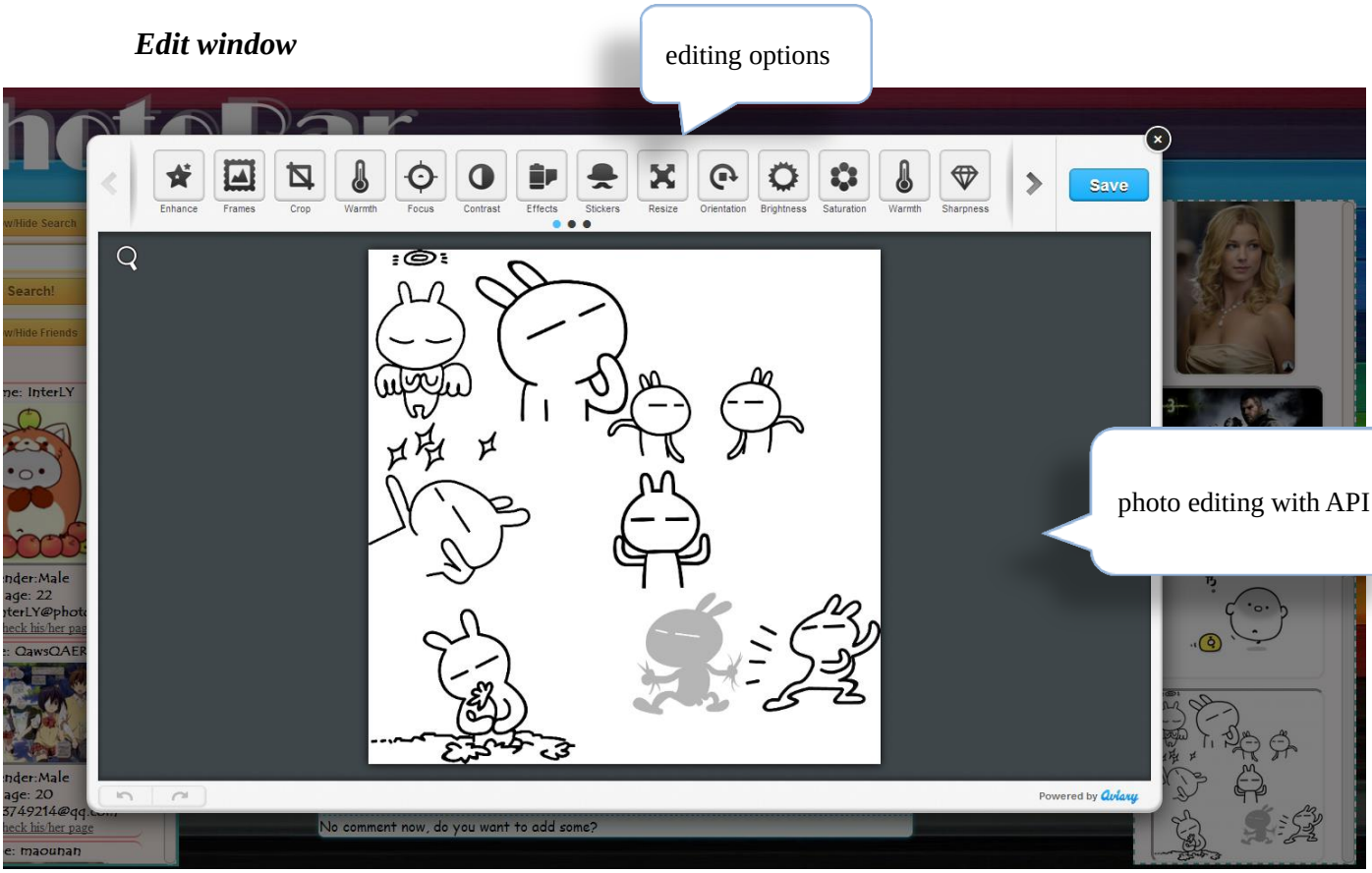
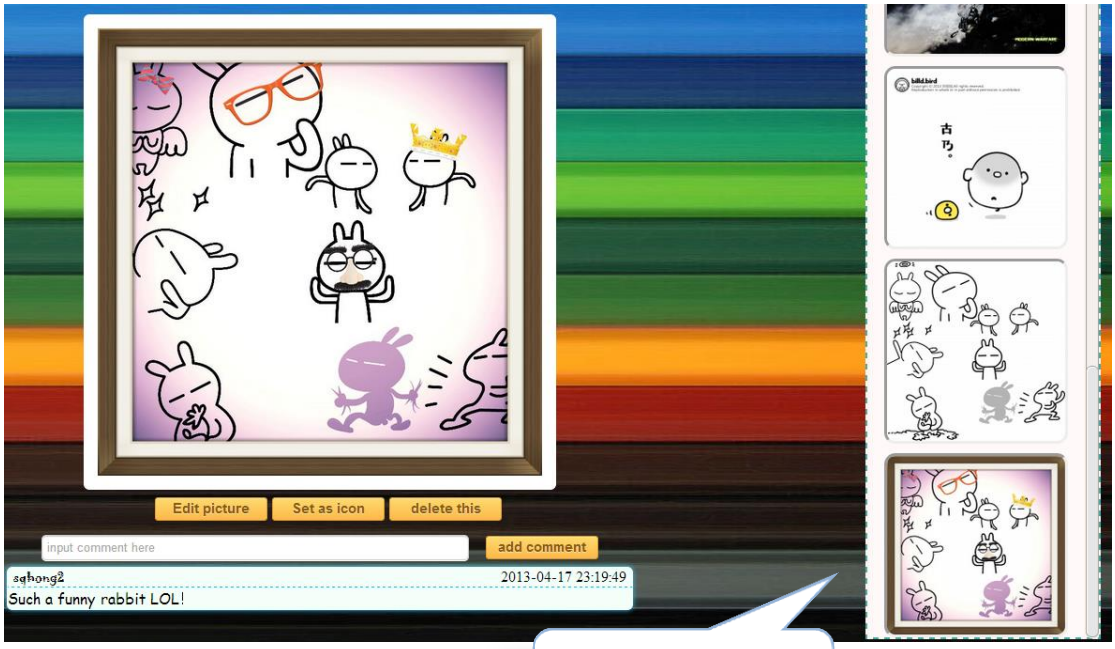


Photo Uploading
Display
Choose Files
3 files
Or you can directly drag photos here
delete
UPLOAD
after photos being chosen
Photo Uploading
Choose Files
No file chosen
Or you can directly drag photos here
Upload succeed!!
Upload succeed!!
Upload succeed!!
All images have been uploaded, you can continue to upload
click here to upload
indicate upload succeed

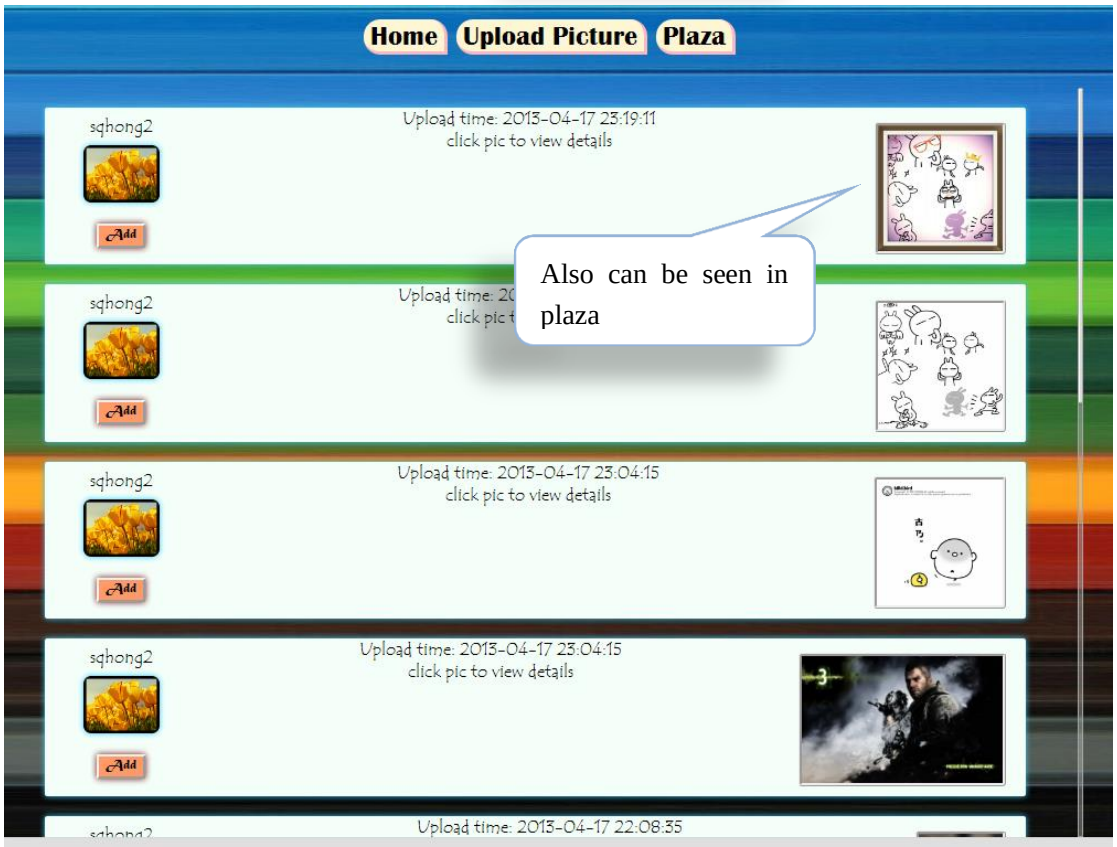
Edit window


editing options
Enhance
Frames
Crop
Warmth
Focus
Contrast
Effects
Stickers
Resize
Orientation
Brightness
Saturation
Warmth
Sharpness
Save
photo editing with API
Powered by Aviary
No comment now, do you want to add some?


Enhance
Frames
Crop
Warmth
Focus
Contrast
Effects
Stickers
Resize
Orientation
Brightness
Saturation
Warmth
Sharpness
Save
click here to save to gallery
Powered by Aviary



Already in the gallery



Also can be seen in plaza

5. TESTS

5.1Test Overview

The following test cases are design to test whether PhotoBar can work correctly.
It consists of four aspects:
1. Whether the home component can work properly.
2. Whether the friend component can work properly.
3. Whether the Plaza component can work properly.

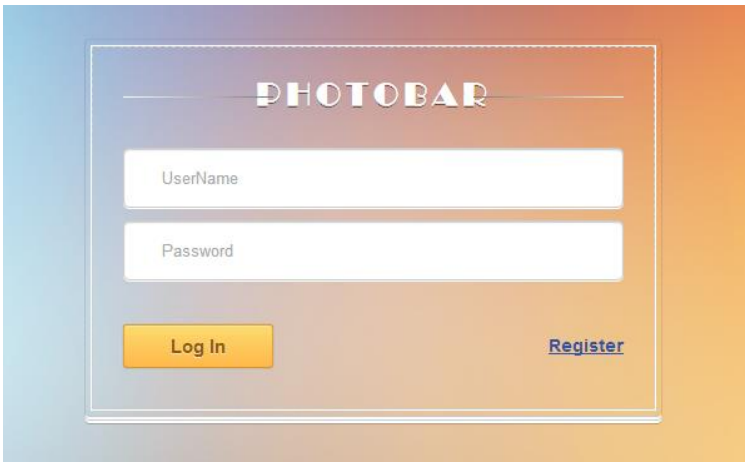
5.2 Case-1

5.2.1 Purpose

This test case is to test whether a blank Username can login our website and can the website give a proper warning windows output.

5.2.2 Inputs

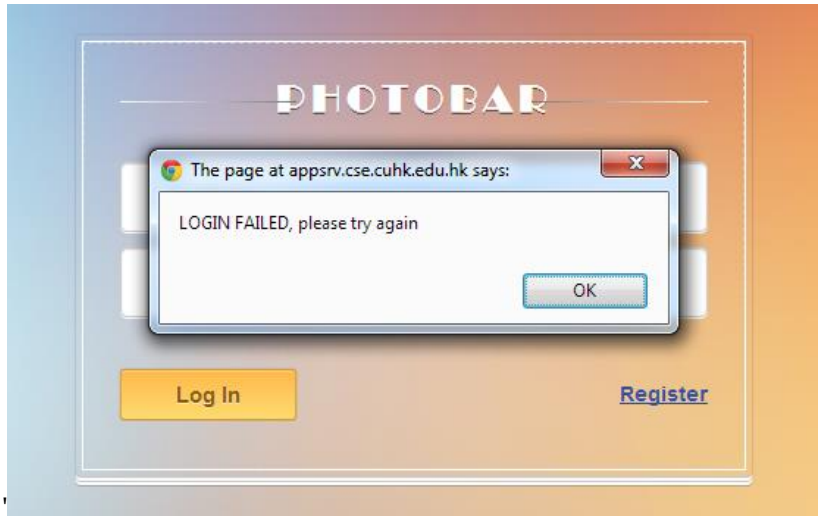
Blank Username and blank Password.



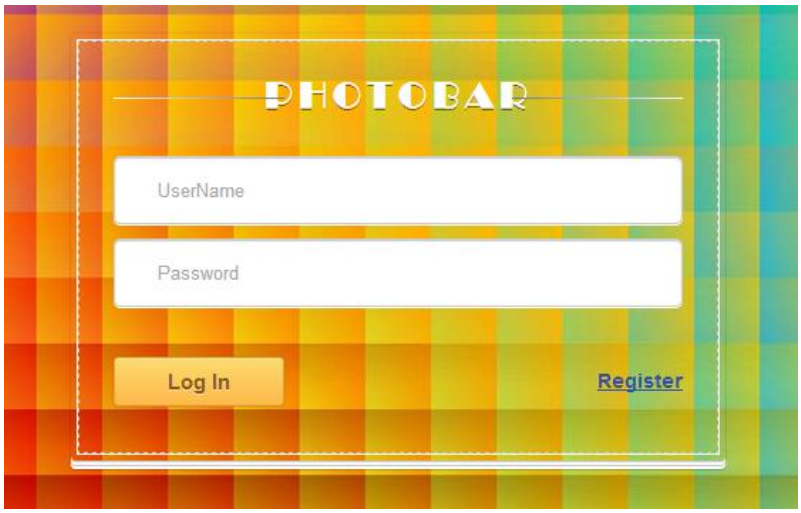
5.2.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected output is a warning window with content ‘LOGIN FAILED, please try again’. And we will return to the login page after we click the ‘OK’ button.

After we click ‘Log In’, the output is:



After we click ‘OK’ button, the output is:



Hence this test case PASSED.

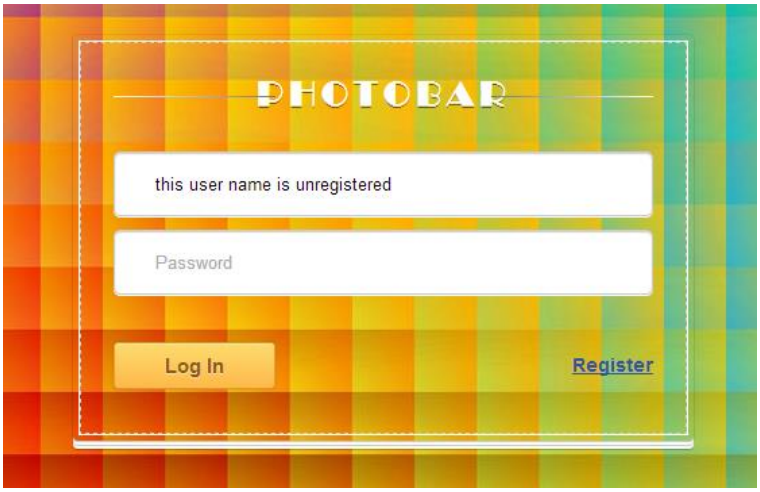
5.3 Case-2

5.3.1 Purpose

This test case is to test whether an unregistered Username can login our website and can the website give a proper warning windows output.

5.3.2 Inputs

Unregister Username and blank Password.

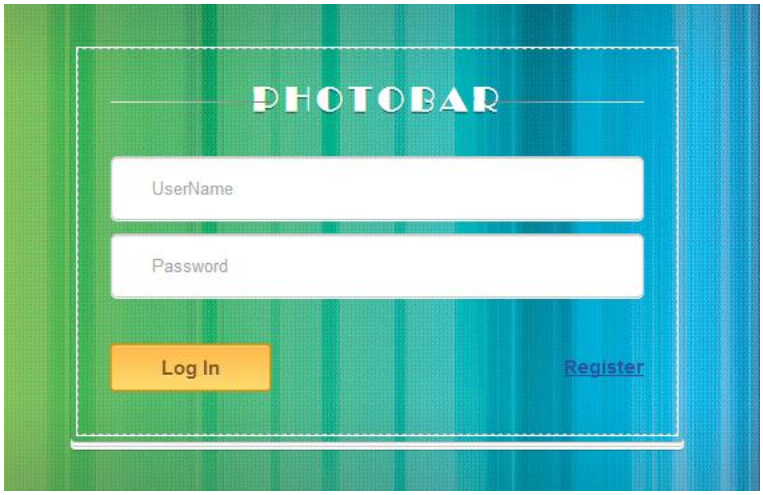


5.3.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected output is a warning window with content ‘LOGIN FAILED, please try again’. And we will return to the login page after we click the ‘OK’ button.

After we click ‘Log In’, the output is:

After we click ‘OK’ button, the output is:



Hence this test case PASSED.

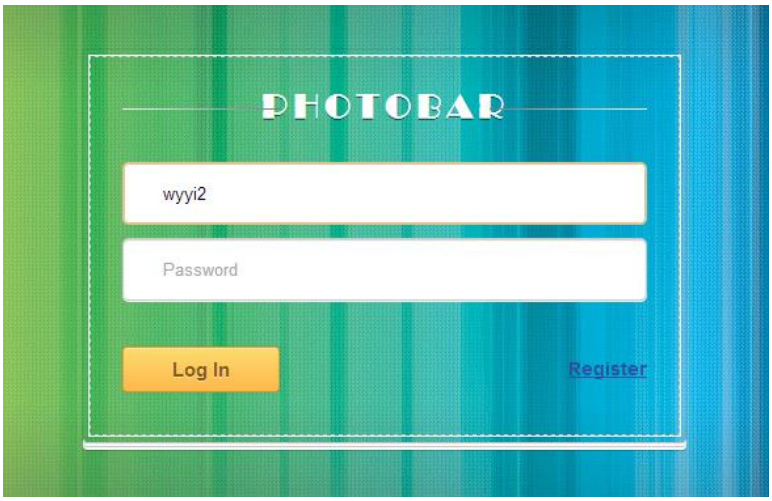
5.4 Case-3

5.4.1 Purpose

This test case is to test whether an registered Username with blank password can login our website and can the website give a proper warning windows output.

5.4.2 Inputs

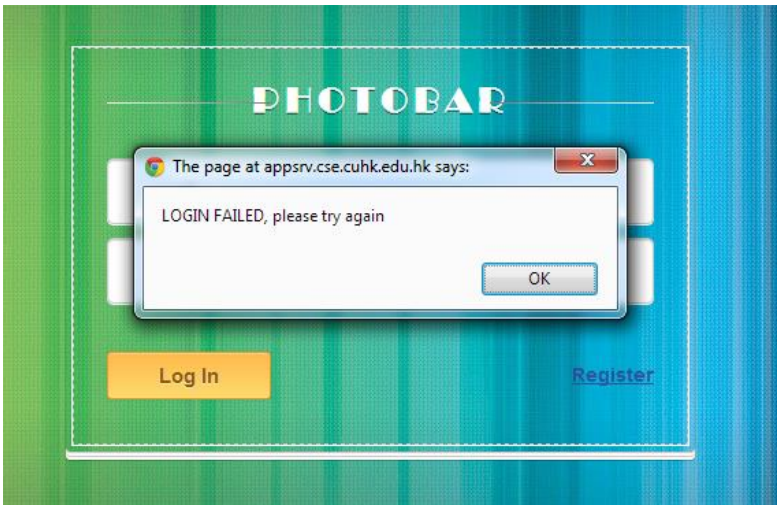
Registered Username and blank Password.



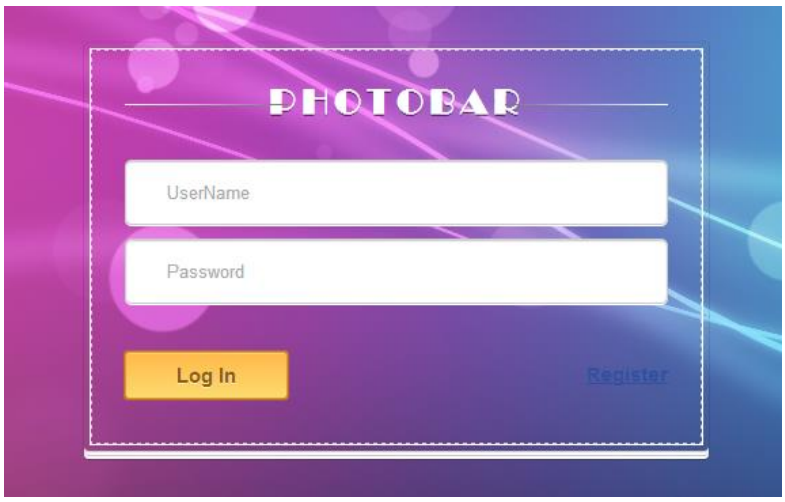
5.4.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected output is a warning window with content ‘LOGIN FAILED, please try again’. And we will return to the login page after we click the ‘OK’ button.

After we click ‘Log In’, the output is:



After we click ‘OK’ button, the output is:



Hence this test case PASSED.

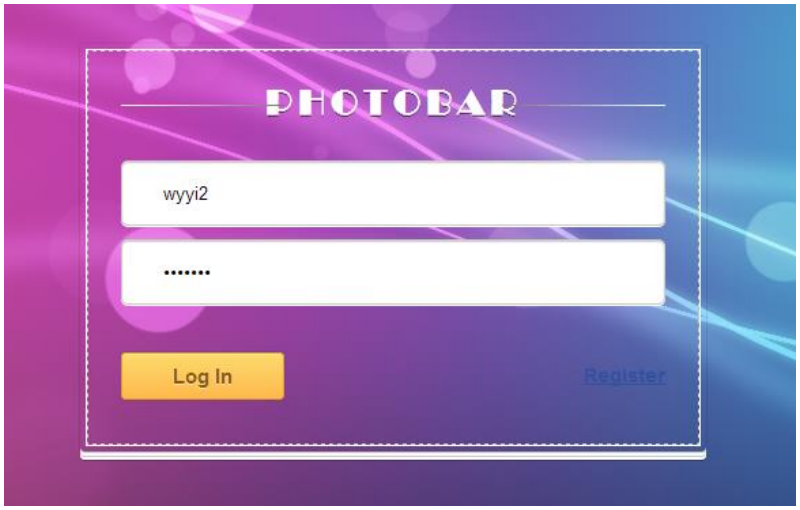
5.5 Case-4

5.5.1 Purpose

This test case is to test whether a registered Username with wrong password can login our website and can the website give a proper warning windows output.

5.5.2 Inputs

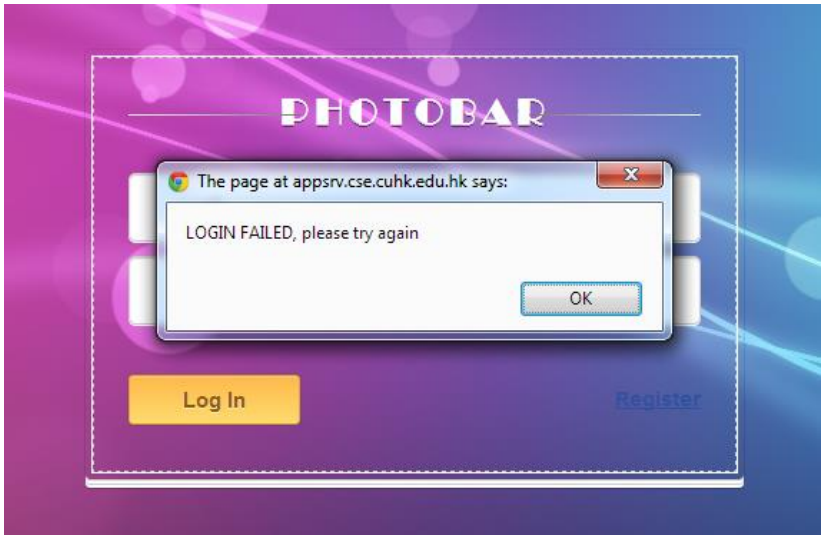
Registered Username and wrong Password with wrong password.



5.5.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected output is a warning window with content ‘LOGIN FAILED, please try again’. And we will return to the login page after we click the ‘OK’ button.

After we click ‘Log In’, the output is:



After we click ‘OK’ button, the output is:

Hence this test case PASSED.

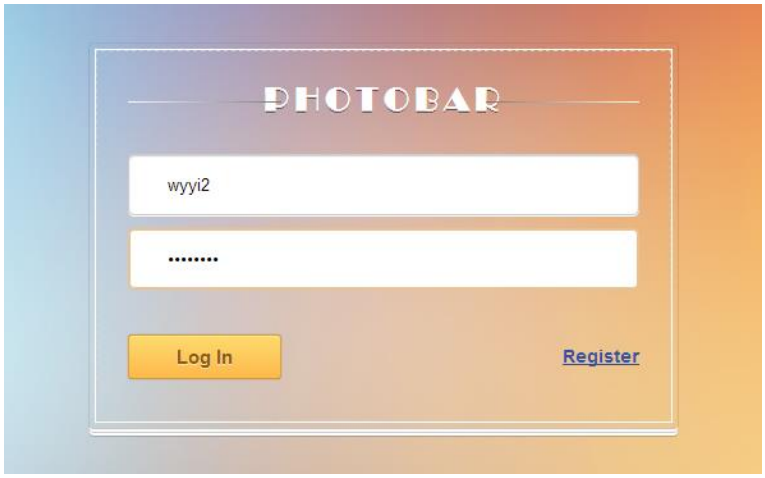
5.6 Case-5

5.6.1 Purpose

This test case is to test whether a registered Username with correct password can login our website.

5.6.2 Inputs

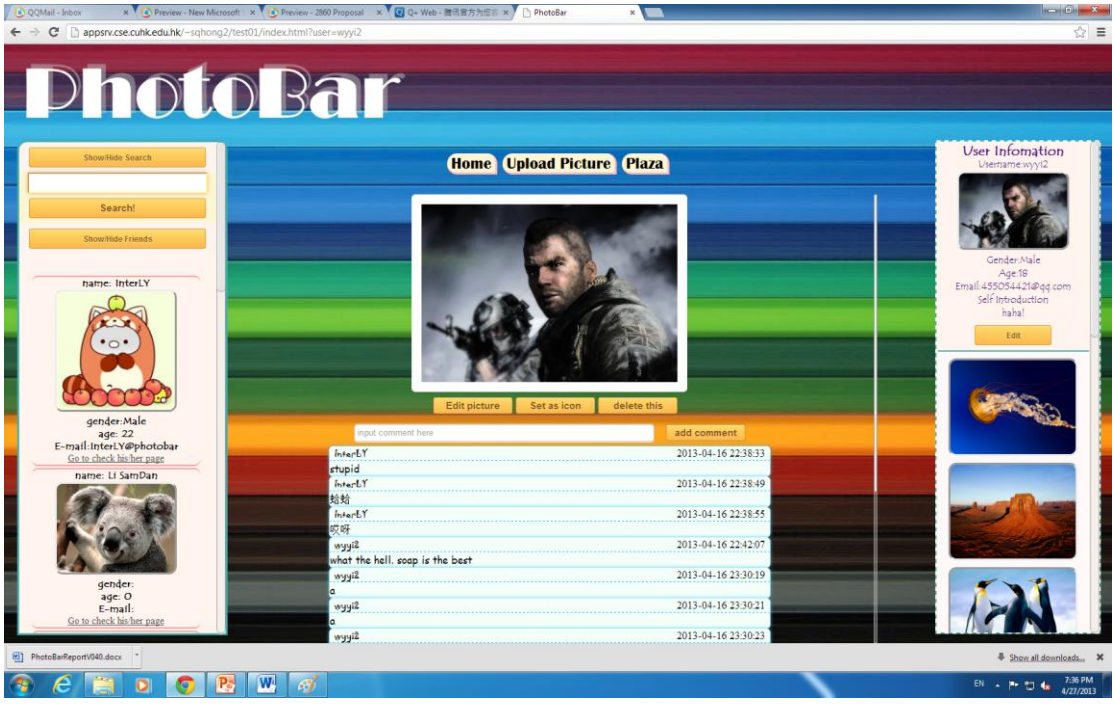
Registered Username and correct Password with correct password.



5.6.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected output is that we will jump to the index page of our project.

After we click ‘Log In’, the output is:



Hence, this test case PASSED.

5.7 Case-6

5.7.1 Purpose

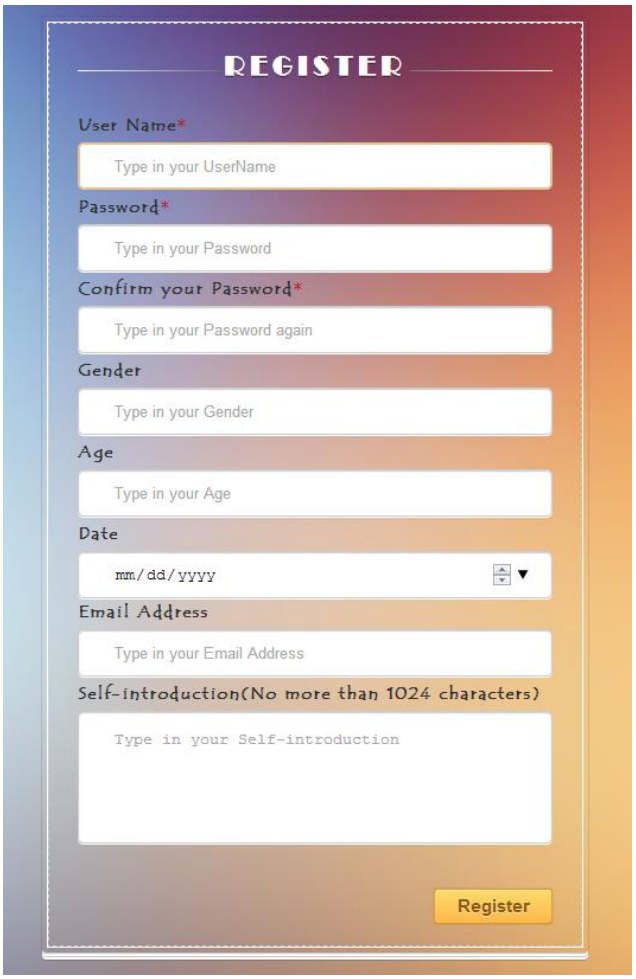
This test case is to test whether the ‘Register()’ component can respond correctly when

1) Blank Username
2) Blank Password
3) Blank Confirm Password

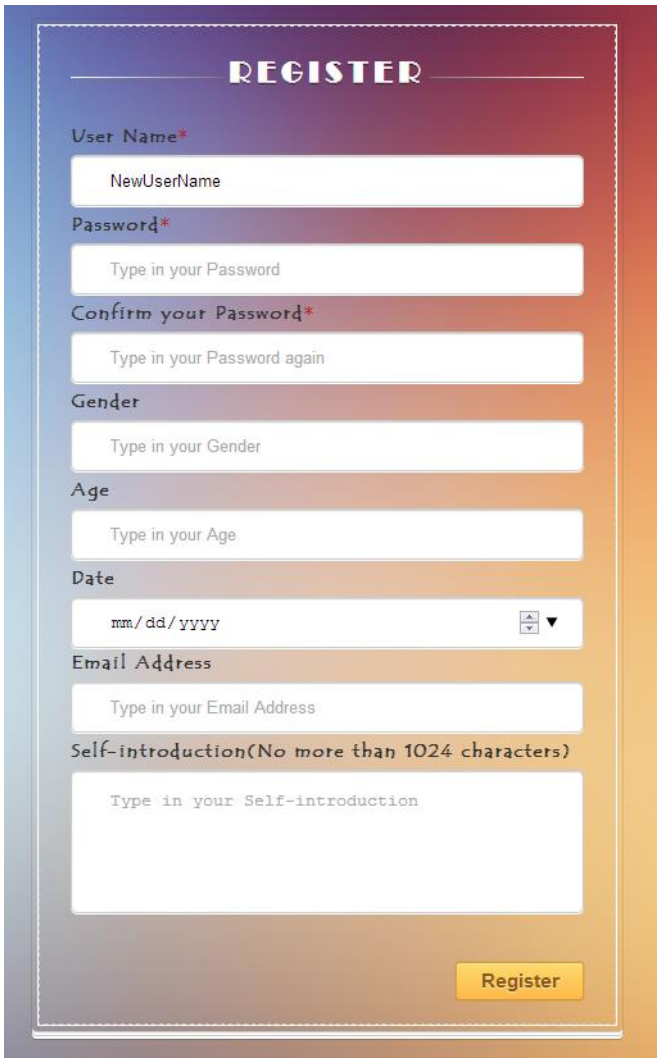
are the inputs.

5.7.2 Inputs

1) We type nothing into the blocks but just click the ‘Register’ button.



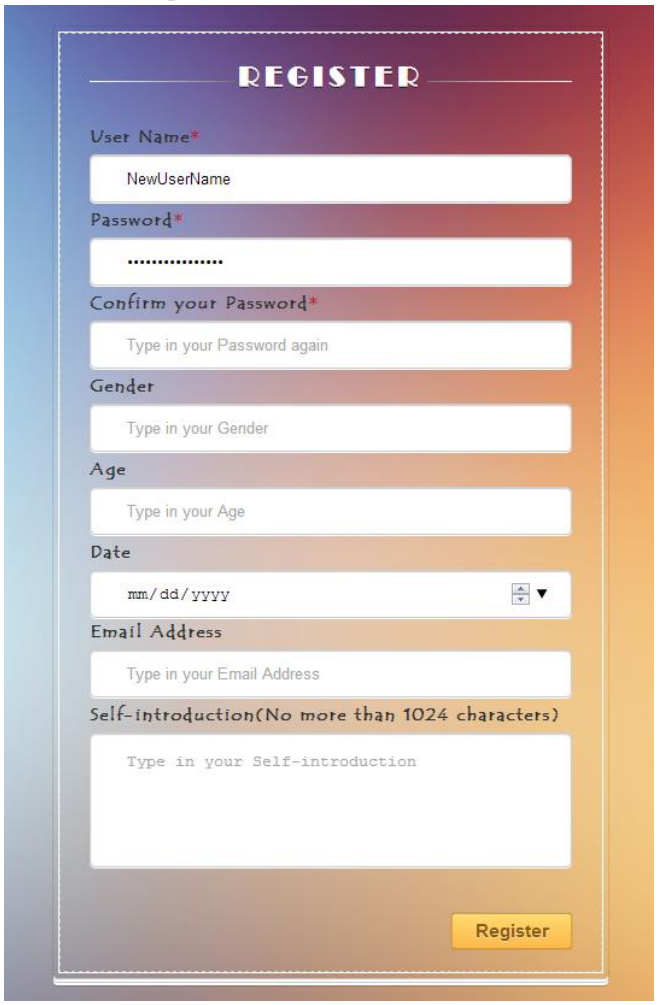
2) Or just type in the user name without password.
User Name: NewUserName



3) Or just type in the user name and password but without confirm password.

User Name: NewUserName

Password: newpassword



4) Type in the user name and password and with wrong confirm password.

User Name: NewUserName

Password: newpassword

Confirm your Password: wrongpassword

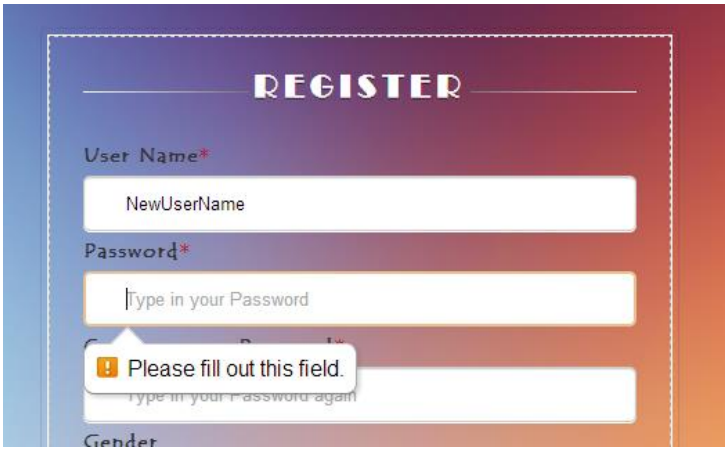
5.7.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected outputs are alerting warning with respective content.

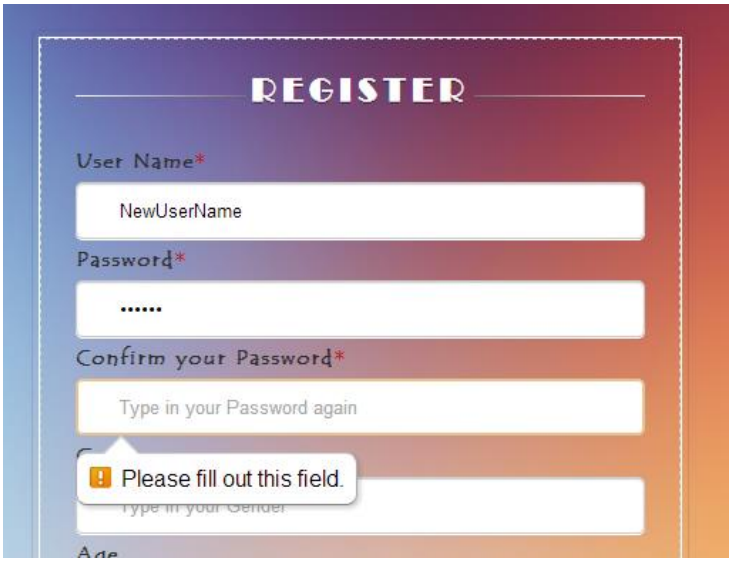
1)


REGISTER
User Name*
Type in your UserName
Please fill out this field.
Confirm your Password*

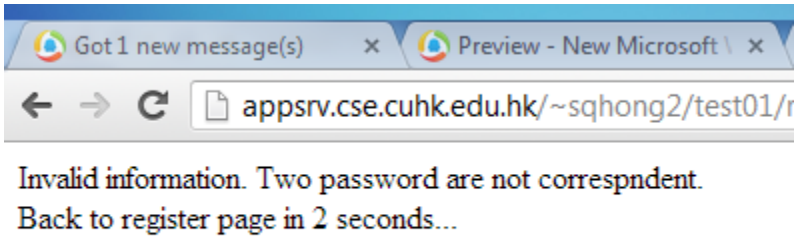
2)


REGISTER
User Name*
NewUserName
Password*
Type in your Password
Please fill out this field.
Gender

3)


REGISTER
User Name*
NewUserName
Password*
.....
Confirm your Password*
Type in your Password again
Please fill out this field.
Age

4)



Hence this test case PASSED.

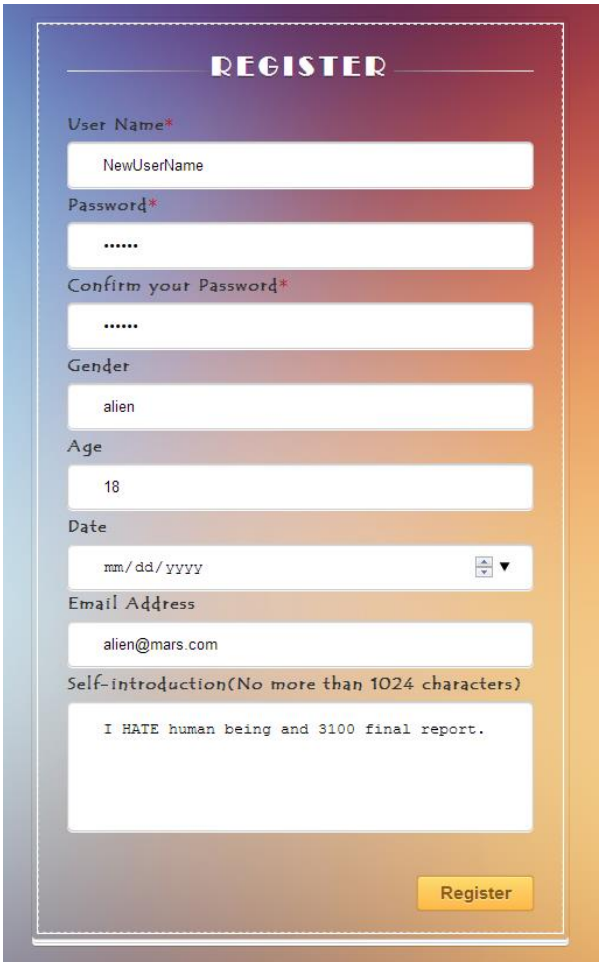
5.8 Case-7

5.8.1 Purpose

This test case is to test whether the 'Register()' component can respond correctly when all the blocks are filled with proper content. And register this new user name into the data base, and then jump back to the login page.

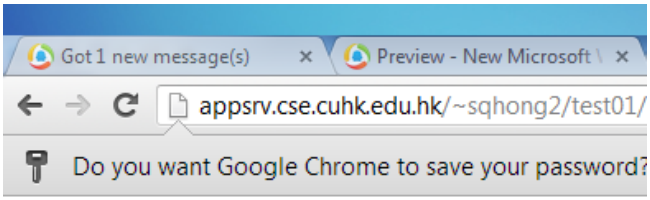
5.8.2 Inputs

User Name: NewUserName
Password: 123456
Confirm your Password: 123456
Gender: alien
Age: 18
Date:
E-mail Address: alien@mars.com
Self-introduction:
I HATE human being and 3100 final report.



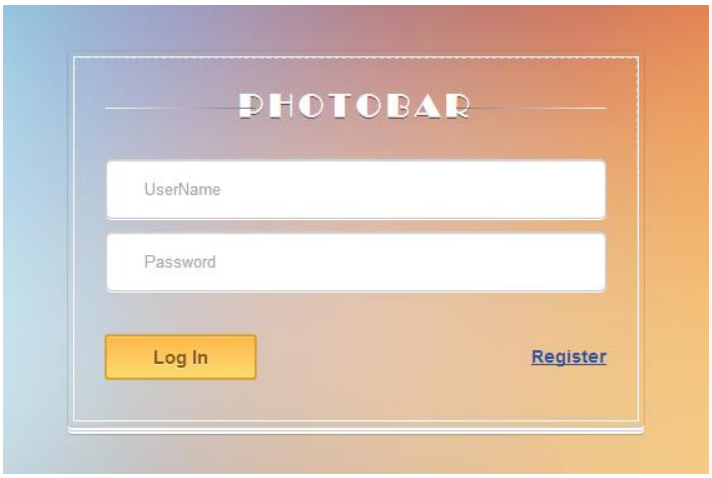
5.8.3 Expected Outputs & Pass/Fail criteria & Coverage

The expected output is that after we click the 'Register' button, we will jump to a new page with prompt information.



Register succeeded! Please log in.

And several seconds later, we will jump to the ‘Login’ page.



Hence this test case PASSED.

5.9 Case-8

5.9.1 Purpose

This test case is to test whether the Home component can work properly after login.

5.9.2 Inputs

PHP Session username

5.9.3 Expected Outputs & Pass/Fail criteria & Coverage

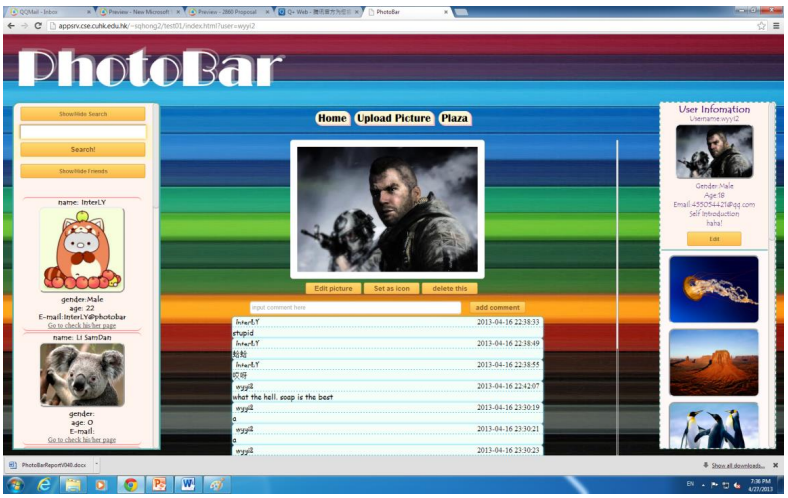
Expected Output:

1. Corresponding User information.
2. Corresponding User’s Pictures’ information.
3. Corresponding User’s Friends’ information.

Pass Criteria:

1. The information is shown in expected HTML elements and with expected style.
2. No javascript execution error.
3. No php execution error.

E.g



Fail Criteria:

1. Some information is displayed in wrong position.
2. Some information is displayed in wrong styles.
3. Some information is not in correct format.
4. Some HTML element is not working well. (E.g button element doesn't trigger specified javascript function)
5. Php code does not return data in correct JSON format.
6. Php code does not return correct data.

5.10 Case-9

5.10.1 Purpose

This test case is to test whether the Friend component can work properly after the user click into the friend's webpage.

510.2 Inputs

PHP Session username

HTML form friend's user name

5.10.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Output:

1. Corresponding User's Friends' Pictures' information.
2. Corresponding User's Friends' information.
3. Corresponding User's targeted Friend's detailed information.

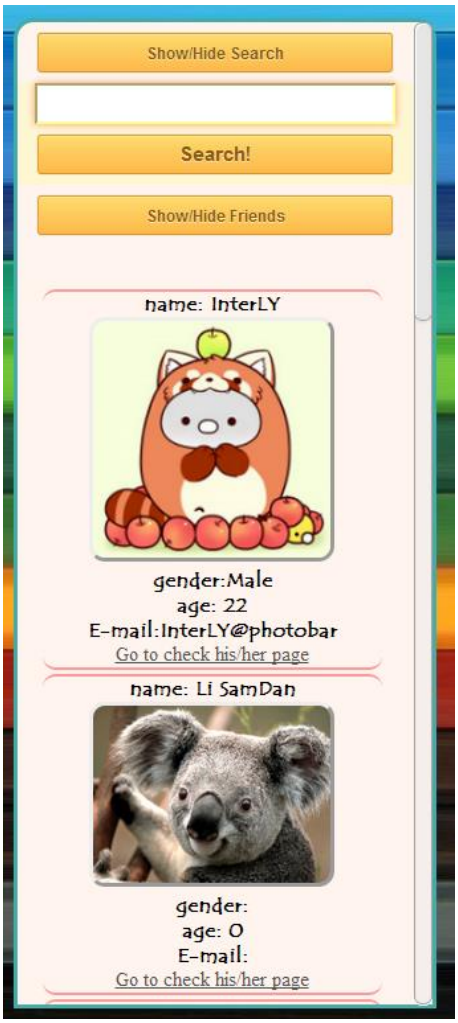
Pass Criteria:

1. The information is shown in expected HTML elements and with expected style.
2. No javascript execution error.
3. No php execution error.

Fail Criteria:

1. Some information is displayed in wrong position.
2. Some information is displayed in wrong styles.
3. Some information is not in correct format.
4. Some HTML element is not working well. (E.g button element doesn't trigger specified javascript function)
5. Php code does not return data in correct JSON format.
6. Php code does not return correct data.

The output of the friends' slot:



All sub-part of this friends' slot are shown properly. Hence this test case PASSED.

5.11 Case-10

5.11.1 Purpose

This test case is to test whether the Plaza component can work properly after the user click into the Plaza webpage.

511.2 Inputs

PHP Session name used to check whether the current user has login.

5.11.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Output:

1. Details of Latest 10 pictures inside the database of PhotoBar
2. Corresponding User's Friends' information.

Pass Criteria:

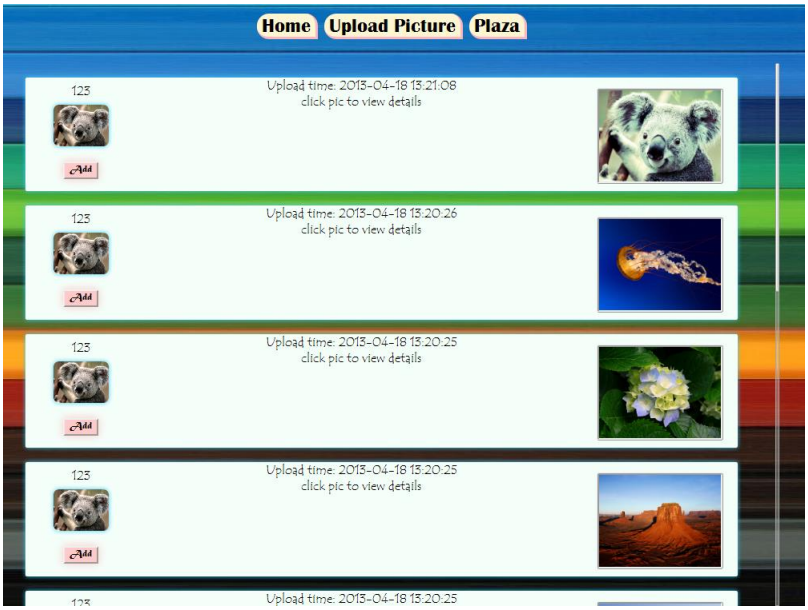
1. The information is shown in expected HTML elements and with expected style.
2. No javascript execution error.

3. No php execution error.

Fail Criteria:

1. Some information is displayed in wrong position.
2. Some information is displayed in wrong styles.
3. Some information is not in correct format.
4. Some HTML element is not working well. (E.g button element doesn't trigger specified javascript function)
5. Php code does not return data in correct JSON format.
6. Php code does not return correct data.

The output of the Plaza component is:



All the sub-part of the plaza are shown properly. Hence this test case PASSED.

5.12 Case-11

5.12.1 Purpose

This test case intends to test whether the upload component can work properly.

5.12.2 Inputs

Multiple Local picture files

5.12.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Outputs:

- An new entry in the database table
- An new image file in the server side

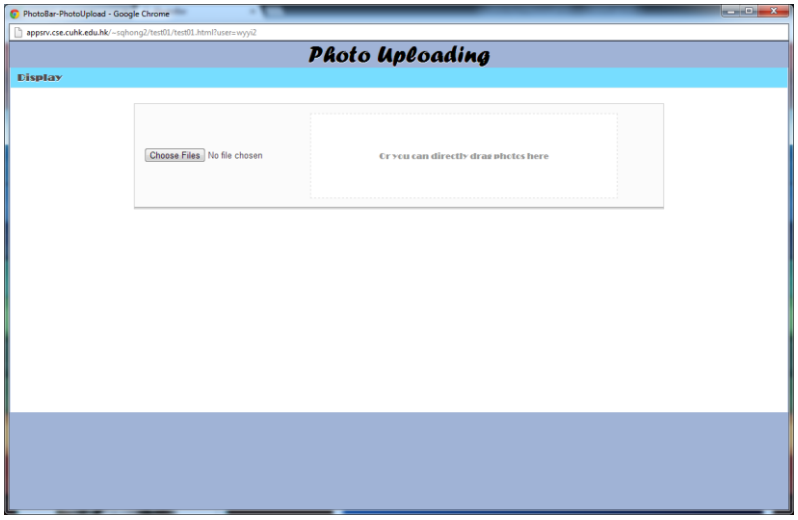
Pass Criteria:

- Corresponding image file is generated in server
- Picture information is recorded correctly in the database

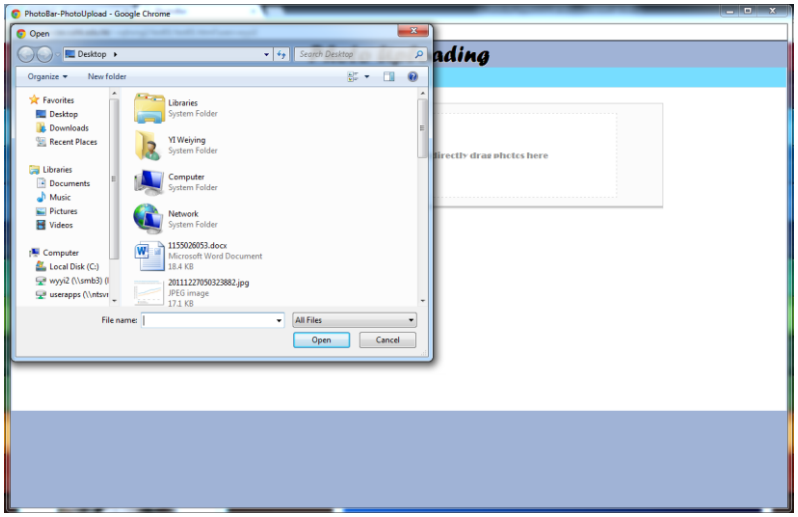
Fail Criteria:

- Corresponding image file is not generated in server
- Picture information is not recorded correctly in the database

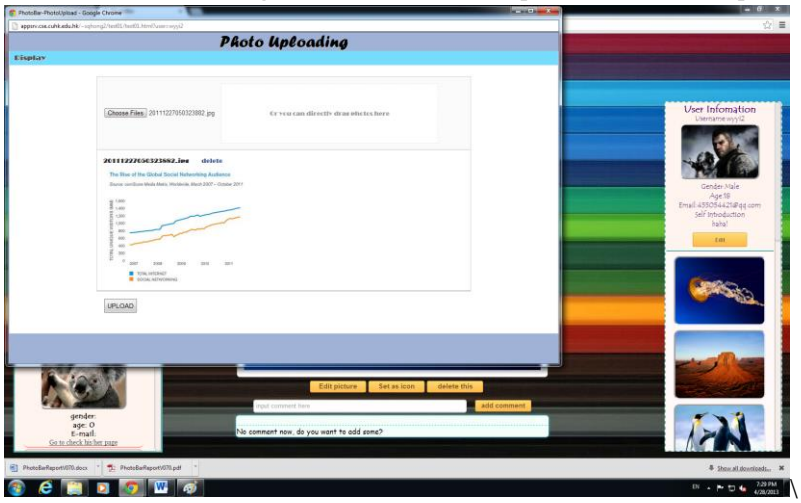
The upload page is shown as:



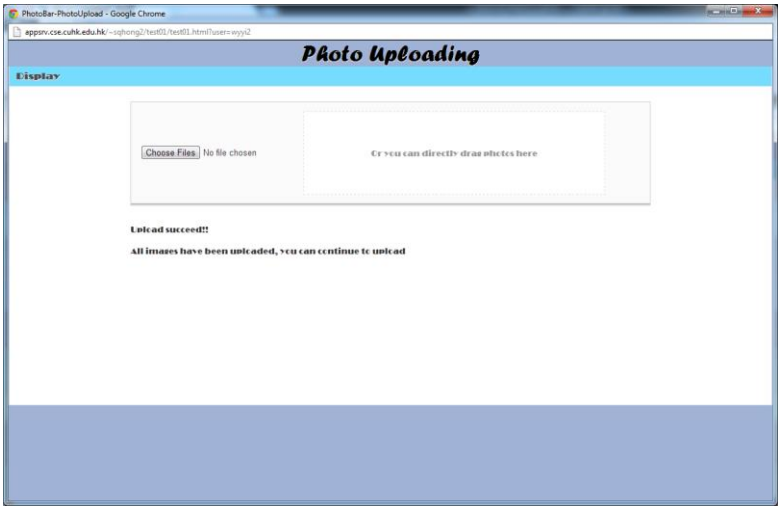
After we click the ‘Choose Files’, the output is:



After we choose some image files and click the ‘open’ button, the output is:



After we click the ‘UPLOAD’ button, the output is:



Hence this test case PASSED.

5.13 Case-12

5.13.1 Purpose

This test case intends to test whether the Search Friend Function can work properly

5.13.2 Inputs

Targeted friend’s username

5.13.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Output:

- Relative users’ information that whose username is similar to the targeted username.
- Search Result is displayed correctly on the screen.

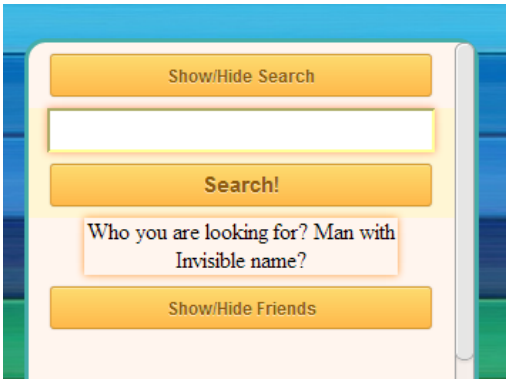
Pass Criteria:

- Relative users’ information that whose username is similar to the targeted username.
- Search Result is displayed correctly on the screen.

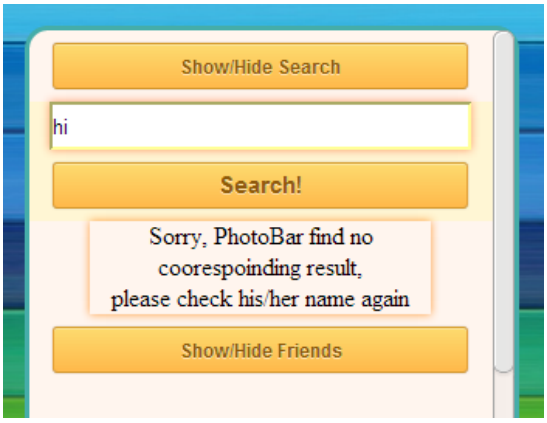
Fail Criteria:

- Output doesn’t match the expected output.

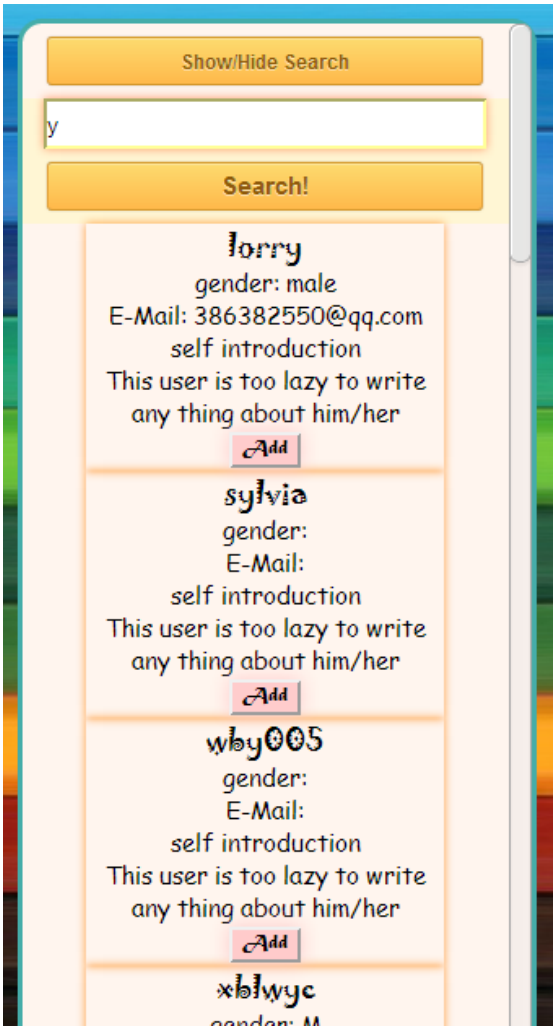
If the search content is blank and we click the ‘Search’ button, the output is:



If the search content is invalid, the output is:



If the search content is valid, the output is:



Hence this test case PASSED.

5.14 Case-13

5.14.1 Purpose

This test case intends to test whether the Editing API can work properly

5.14.2 Inputs

Local server image URL

5.14.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Output:

- The Edit API is called without error.
- The modified image stored in our server.

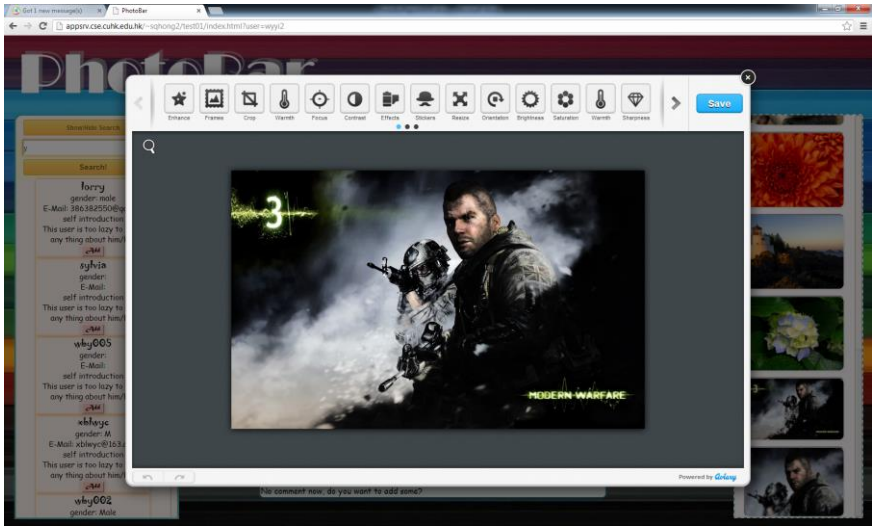
Pass Criteria:

- Output matches the Expected Output.

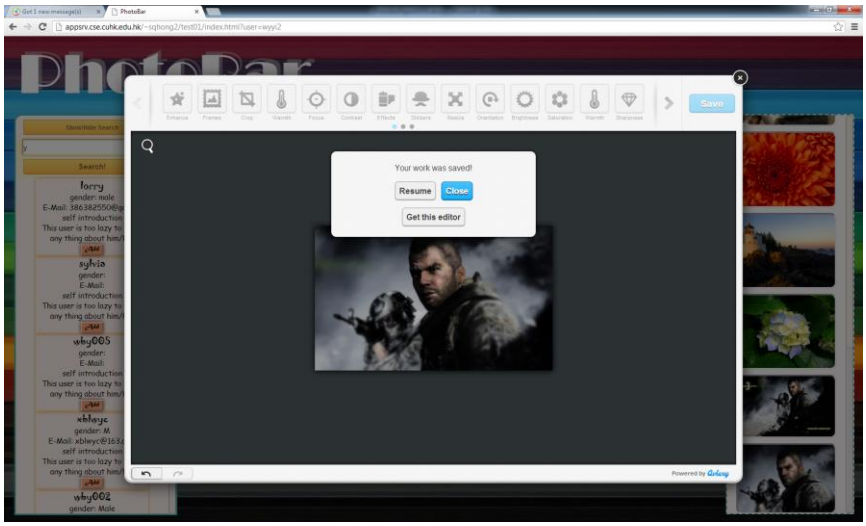
Fail Criteria:

- Output doesn't match the expected Output.

After we click the 'Edit picture' button, the output is:



After we do some editing and then click the 'Save' button, the output is:



And at last the picture is show is the ‘pictures show slot’:

Hence this test case PASSED.

5.15 Case-14

5.15.1 Purpose

This test case intends to test whether the comment component can work properly

5.15.2 Inputs

1. Picture id
2. Input comment content

5.15.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Output:

- New comment is shown on the screen with username, comment content and comment time.

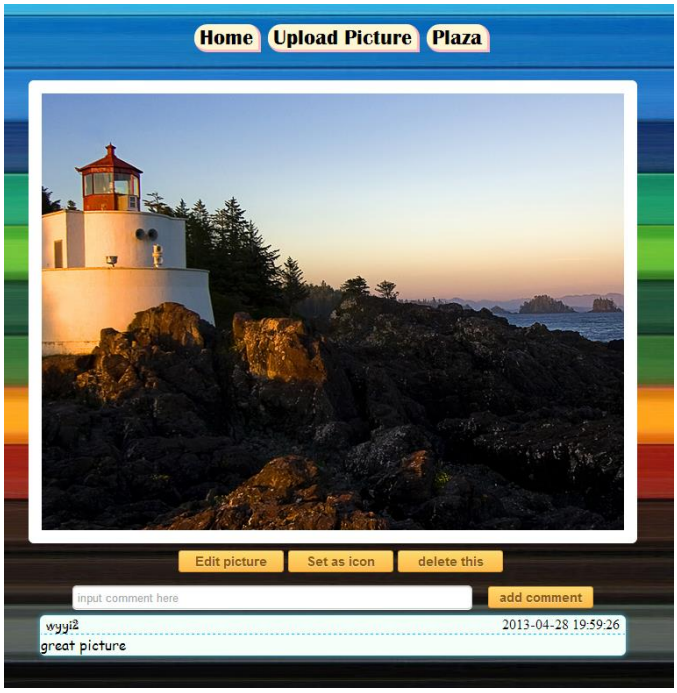
Pass Criteria:

- Output matches the Expected Output.

Fail Criteria:

- Output doesn't match the expected Output.

We input some comment into the comment block and click the 'add comment', the output is:



Hence this test case PASSED.

5.16 Case-15

5.16.1 Purpose

This test case intends to test ShowPictureDetail component.

5.16.2 Inputs

The picture id.

5.16.3 Expected Outputs & Pass/Fail criteria & Coverage

Expected Outputs:

- Picture Detailed information is shown in either the main image container or a newly created tab, according to different input parameters.
- Other HTML element is equipped with correct event listener.

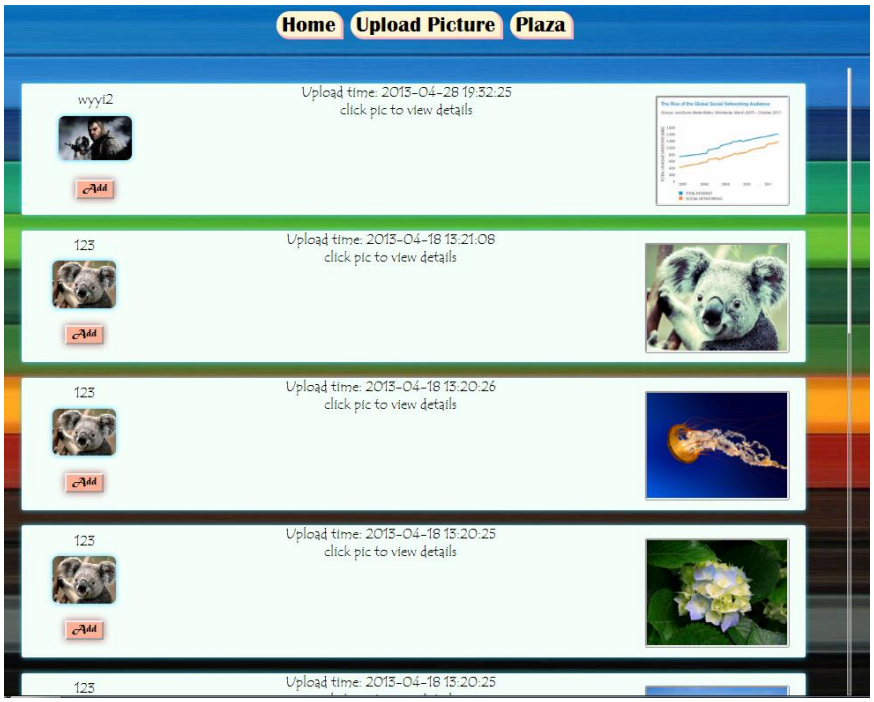
Pass Criteria:

- Output matches the Expected Output.

Fail Criteria:

- Output doesn't match the expected Output.

Before we click the picture in the Plaza, the small pictures are shown:



After we click the small picture, the detail of this picture is shown:

Hence this test case PASSED.

6. TO BE IMPROVED

1. After adding a user as a friend, the friend list should be able to refresh automatically.

2. When a user being added as a friend of another user, it should inform the user for acceptance.

3. If there are too many photos in a user's gallery which showed in the right list of the screen, it should be able to page automatically.

4. It should be able to change photo names for avoiding a user upload more than one photos with the same names.