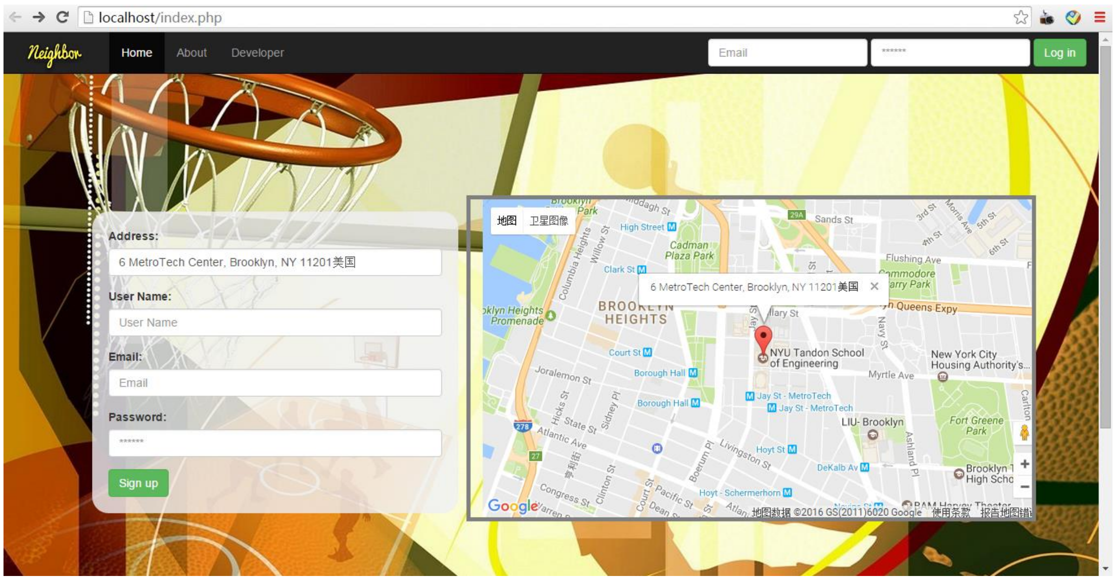
Principles of Database Systems

CS 6083 - INET

Final Report—Sports Fans Social Network

Lanlan Hu lh1981

Juexiu Wu jw4256



December 16, 2016

1. Database Design

1.1 Abstract

We choose the third Application Scenarios: Sports fans social network. By using this application, users can follow teams which they prefer.

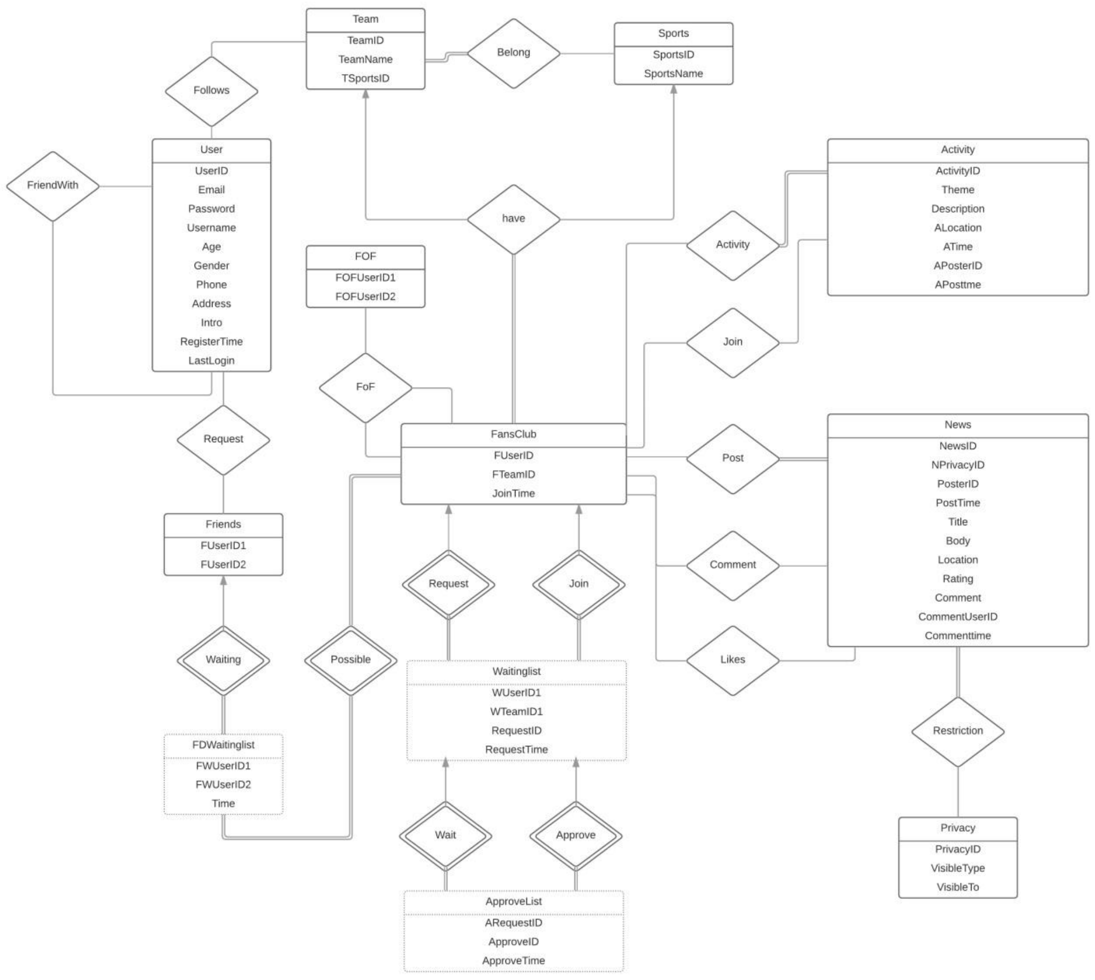
Each team must belong to one kind of sports. Each team has its own fans club. Users can apply to join a fans club by sending request. The request will be approved if there are at least three users in the fans group have approved it.

After users joining in a fans club, they can add other fans in the same fans club as friends. This kind of relationship is mutual, which means that two users are seen as friends only if each of them has selected the other as a friend. In addition, users in the same fans group can add Friends of Friends (FoF). However, this kind of relationship is unilateral.

Users can post diaries about their teams, sports, events, locations, etc. Besides, users can post activities, in which they can specify locations, such as a city they visited or a city for a conference they attended.

For aforementioned news and activities, users can like and make comments on them. The access permission to diaries can be restricted using privacy privileges. In "Privacy", users can define whether it could be visible only to their friends, or to friends of friends, or to everyone in the application.

1.2 E-R Model



1.3 Table Design

Table1	User(UserID, Email, Password, Username, Age, Gender, Phone, Address, Intro, RegisterTime, LastLogin, Notification, NotificationType)
Description	Each user has a unique UserID to be identified. Username, Email, and Password are stored when a user signs up. RegisterTime stores the time when a user signs up, and LastLogin is the last time when user log in. Other information such as NotificationType and Notification are set by user in System Setting, and can be modified by the user.

Table2	Team(TeamID, TeamName, TSportsID)
Description	Each team has a unique TeamID to be identified. One kind of sports may contain several teams. TSportsID is a foreign key referencing Sports(SportsID).
Table3	Sports(SportsID, SportsName)
Description	Each kind of sports has a unique SportsID to be identified.
Table4	FansClub(FUserId, FTeamID, JoinTime)
Description	FansClub table stores which team's fans club one user belongs to. Each user is identified via a unique pair of (FUserID, FTeamID), thus one user may belong to several different teams' fans clubs. And JoinTime is given to show the time when the request approved.
Table5	WaitingList(WUserID, WTeamID, RequestID, RequestTime)
Description	WaitingList table stores information when a user wants to join in one team, while the application has not been approved yet. RequestID is used to specify each request. WUserID and WTeamID displays the user and the team he/she wants to join in.
Table6	ApproveList(ARequestID, AUserID, ApproveTime)
Description	ApproveList table stores information when the request for joining in one team is approved by other fans in fans club. We assume that for a specific request, if there are more than three AUserIDs, this request will be approved and the user will be stored in the FansClub Table. RequestID and AUserID are used to identify each approval.
Table7	Friends(FUserID1,FUserID2)
Description	Friends table indicates that user1 and user2 are friends. Because being friends is a mutual relationship, we treat row (1,2) and row (2, 1) as redundant data, and we only store one row of them.
Table8	FDWaitingList(FWUserID1, FWUserID2,Time)
Description	Fans can add other fans in the same fans club as friends. This kind of relationship is mutual, which means that two users are friends only if each one has selected the other as a friend. Thus, before the request being

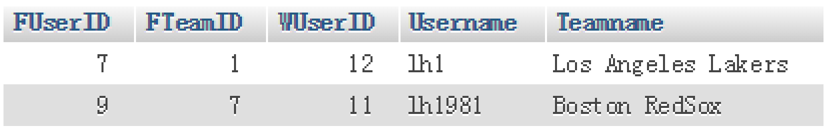
	approved, it needs to be stored in this table. And a pair of (FWUserID1, FWUserID2) is used to identify each request of being friend.
Table9	FOF(FOFUserID1, FOFUserID2)
Description	Users in fans group of the same sports can add Friends of Friends. This kind of relationship is unilateral. FoFUID2 is friend of FOFUID1. This relationship is done by FoFUID1.
Table10	News(NewsID, NPrivacyID, PosterId, PostTime, Title, Body, Location, Rating, Comment, CommentID, Commenttime)
Description	News table stores all postings, such as comments and diaries about their team, sports, events, places, etc. Each piece of news is identified via a unique NewsID. Each piece of news will choose a PrivacyID when being posted. Also each piece of news uses PosterId and PostTime to show who post this news and when the news is posted. Each piece of news is constituted of Title, Body, Location, and Rating. And each comment is constituted of Comment, CommentID, and Commenttime.
Table11	Privacy(PrivacyID, VisibleType, VisibleTo)
Description	Privacy table uses PrivacyID to identify different access permissions, and each PrivacyID has VisibleType and VisibleTo columns set by users to specify which user can read the news. For example, when VisibleType is “team”, fans in the same team can read the news. And if VisibleType is “sports”, users love the same sports can read the news. Besides, if VisibleType is “everyone”, every user can read the news.
Table12	Activity(ActivityID, Theme, Description, ALocation, ATime, APosterID, APosttime)
Description	Activity table stores all posted activities, and each activity is identified by a unique ActivityID.

2. Actions in System

2.1 View waiting

This view is created to store all requests of joining teams. By creating this view, it's much simpler to sort the requests that need to be dealt with for a specific user. And it is implemented in approve.php

```
CREATE VIEW waiting AS
SELECT distinct FUserID,FTeamID,WUserID,Username,Teamname
FROM fansclub NATURAL JOIN waitinglist NATURAL JOIN user NATURAL JOIN team
WHERE FTeamID=WTeamID
and WUserID=UserID
and FTeamID=TeamID
```

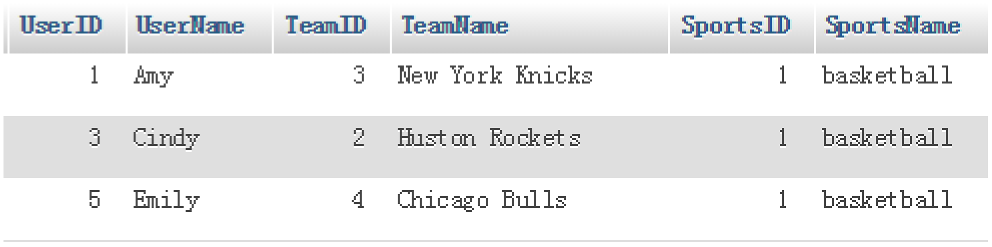


FUserID	FTeamID	WUserID	Username	Teamname
7	1	12	lh1	Los Angeles Lakers
9	7	11	lh1981	Boston RedSox

2.2 View neighbor

This view is created to store all information about which team and which kind of sports that users like. By creating this view, we can sort out the possible friends(FoF) for a specific user immediately. And it is implemented in neighbor.php

```
CREATE VIEW neighbor AS
SELECT UserID, UserName, TeamID, TeamName, SportsID, SportsName
FROM user natural join fansclub natural join team natural join sports
where UserID=FUserID
and TeamID=FTeamID
and TSportsID=SportsID
```

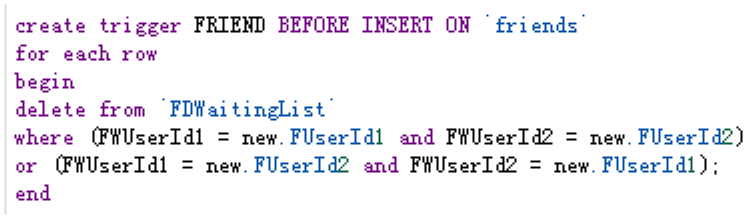


UserID	UserName	TeamID	TeamName	SportsID	SportsName
1	Amy	3	New York Knicks	1	basketball
3	Cindy	2	Huston Rockets	1	basketball
5	Emily	4	Chicago Bulls	1	basketball

2.3 Trigger Friend

1. User can view people who like the same kind of sports and have not been friends yet, and be able to add them as friends.
2. When adding a friend, we need to send a request and wait for approval. So at first we add the request to FDWaitingList table. If A and B already are friends, neither of them can send request to each other.
3. If the request is approved, a new friend relationship is added to the Friends table and corresponding request data FDWaitingList table is deleted. The deletion will be

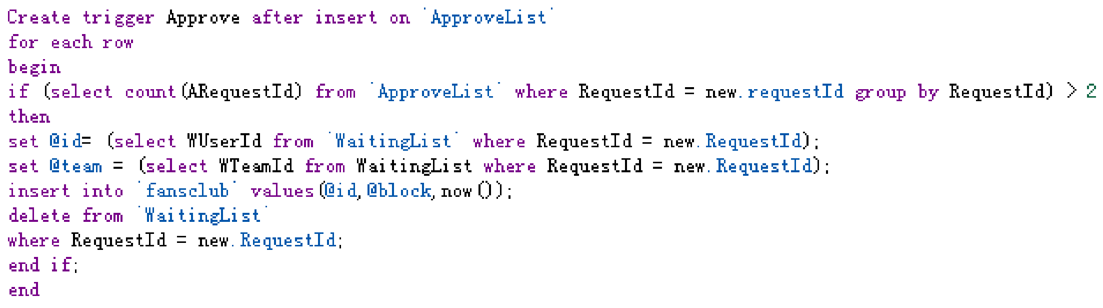
done by a trigger on Friends.
4. The friend relationship is symmetrical, which means we treat row(A,B) and row(B,A)
as same relation. So if we want to find friends of a user, we need to do search in both FUserID1 and FUserID2.
5. To show the possible users that we can add to be friends, all existing friends within the same sports should be filtered out in advance. In this process, both FUserID1 and FUserID2 need to be put into consideration.
The trigger Friend used for Friends table:



```
create trigger FRIEND BEFORE INSERT ON `friends`
for each row
begin
delete from `FDWaitingList`
where (FWUserId1 = new.FUserId1 and FWUserId2 = new.FUserId2)
or (FWUserId1 = new.FUserId2 and FWUserId2 = new.FUserId1);
end
```

2.4 Trigger Approve

1. If a user wants to join the fans club of a team, he or she should make a request and the request will be added into WaitingList table at first. If the request already exists, he or she is not able to make a repeated request. And we use WUserID and WTeamID to identify each request.
2. The request will be sent to every users belongs to the requested fans club.
3. Once a user in the fans club approves a request, the request will be added to ApproveList. If the number of the approvals reaches 3, then this request will be deleted from WaitingList, and a new member will be added in the fans club. The deletion will be implemented by a trigger on ApproveList.
4. When a user logs in, we have to judge whether the user belongs to a fans club. If not, he or she cannot view other information concerning the team. In general, the user will be guided to an empty page NoAuthority.php
The trigger Approve used for ApproveList table:



```
Create trigger Approve after insert on `ApproveList`
for each row
begin
if (select count(ARequestId) from `ApproveList` where RequestId = new.requestId group by RequestId) > 2
then
set @id= (select WUserId from `WaitingList` where RequestId = new.RequestId);
set @team = (select WTeamId from WaitingList where RequestId = new.RequestId);
insert into `fansclub` values(@id,@block,now());
delete from `WaitingList`
where RequestId = new.RequestId;
end if;
end
```

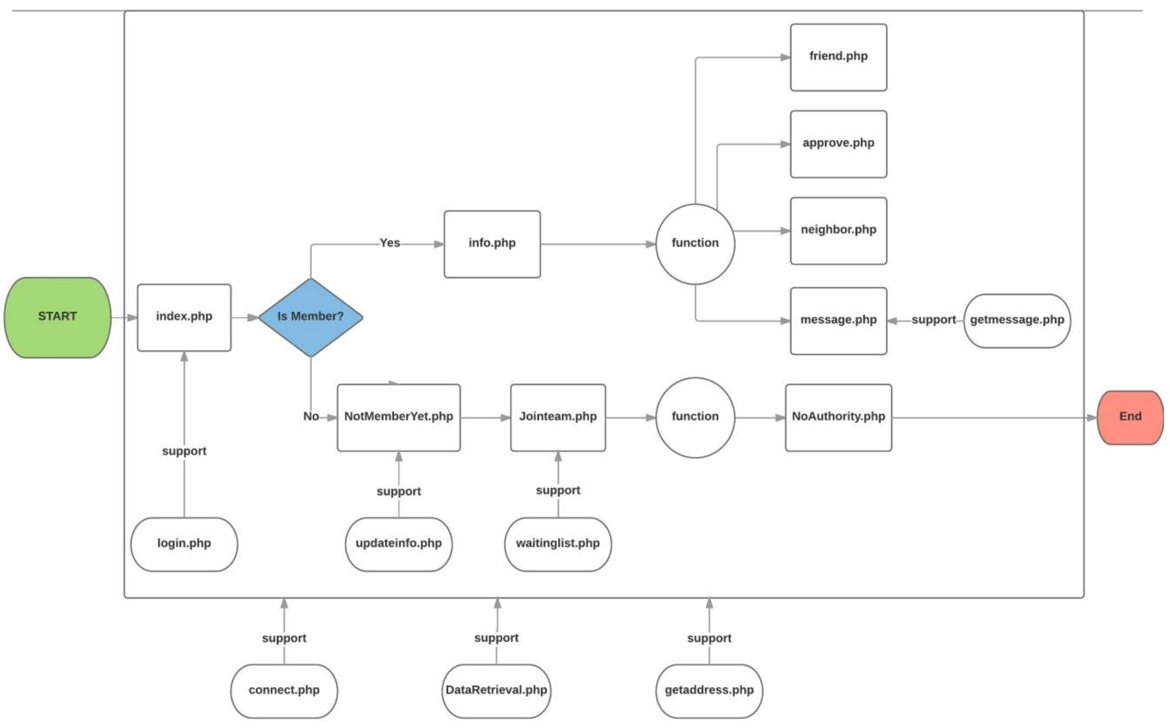
2.5 Bootstrap

In this project, the most popular frontend framework Bootstrap is used among all pages. The basic Bootstrap structure is like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

3. Back-End Design

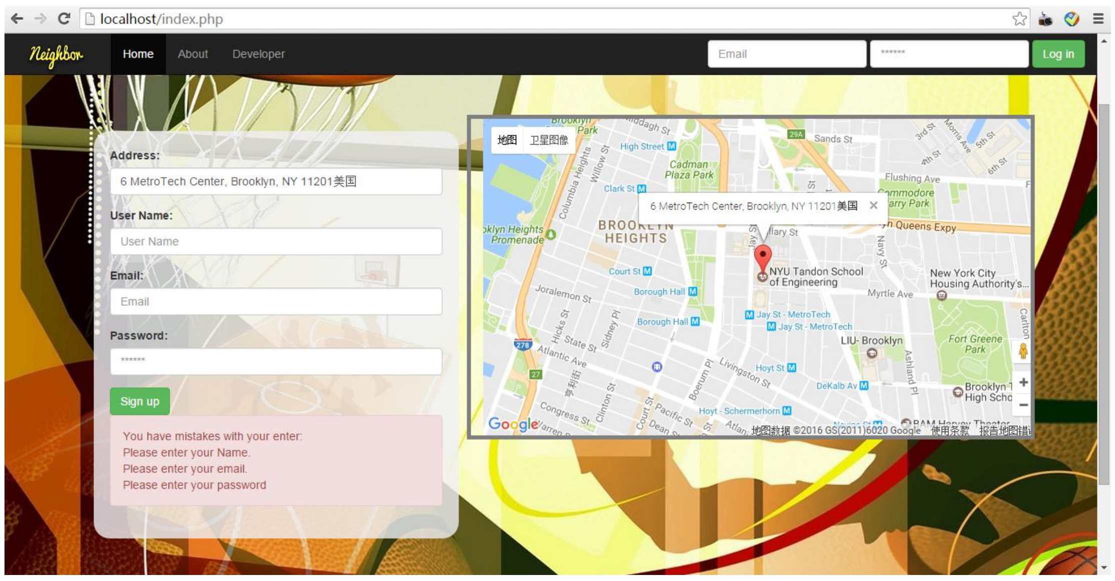
3.1 PHP Files Structure



3.2 PHP Files

3.2.1 index.php

The result of index.php and login.php with no valid entry.



Address:

6 MetroTech Center, Brooklyn, NY 11201美国

User Name:

User Name

Email:

Email

Password:

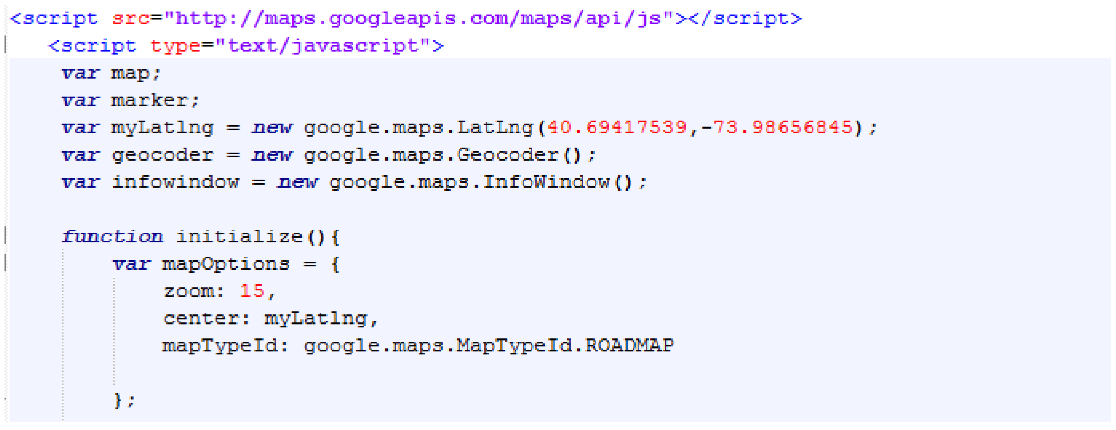
******

Sign up

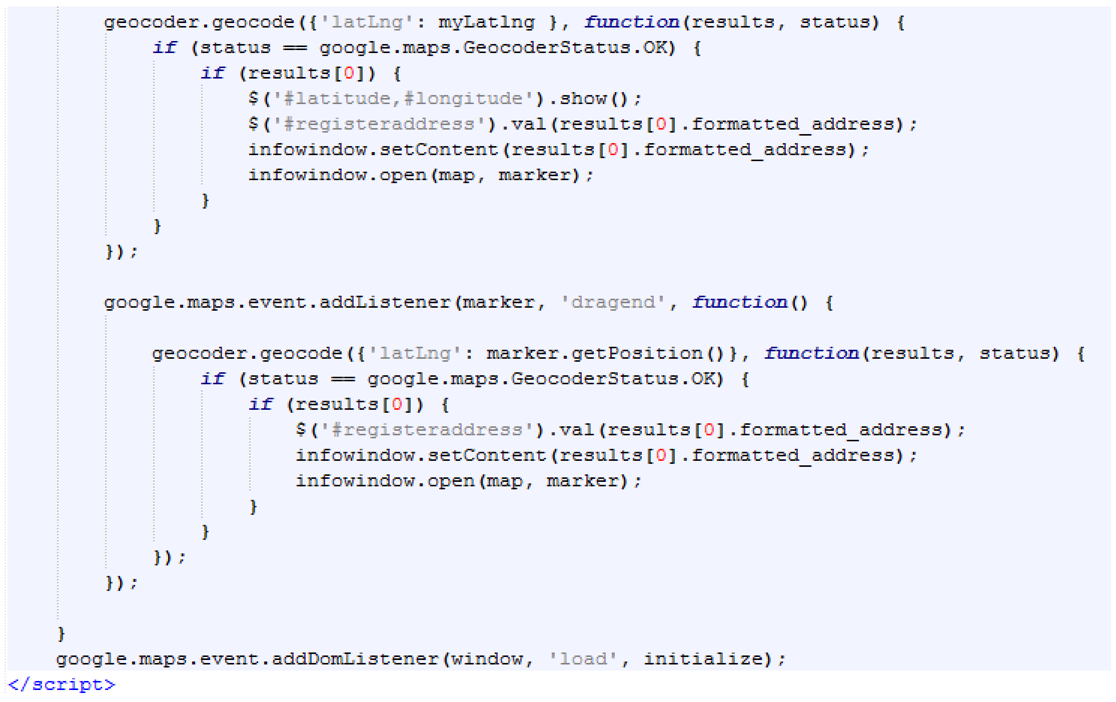
You have mistakes with your enter:
Please enter your Name.
Please enter your email.
Please enter your password

6 MetroTech Center, Brooklyn, NY 11201美国

In this file, the most important part is the script listed below, the function of this script is to load the google map when initializing the window, and also initialize the marker on the map which gives users the ability to drag the marker to update address information in the address input box.



```
<script src="http://maps.googleapis.com/maps/api/js"></script>
    <script type="text/javascript">
      var map;
      var marker;
      var myLatlng = new google.maps.LatLng(40.69417539,-73.98656845);
      var geocoder = new google.maps.Geocoder();
      var infowindow = new google.maps.InfoWindow();

      function initialize(){
          var mapOptions = {
              zoom: 15,
              center: myLatlng,
              mapTypeId: google.maps.MapTypeId.ROADMAP

          };
```



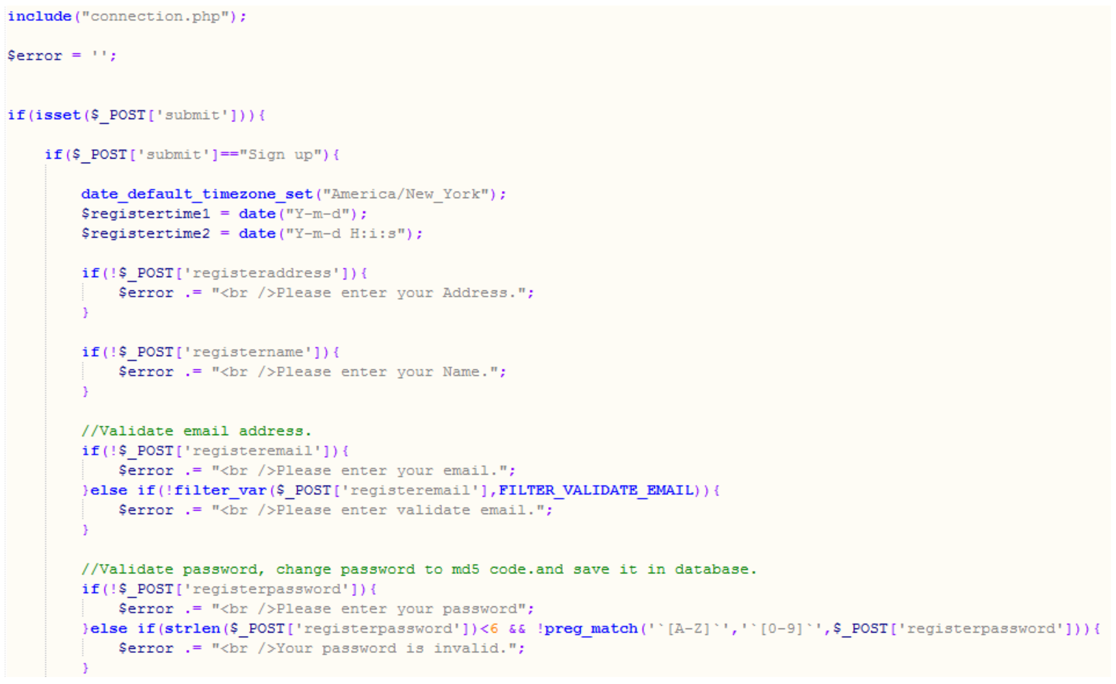
```
          geocoder.geocode({'latLng': myLatlng }, function(results, status) {
              if (status == google.maps.GeocoderStatus.OK) {
                  if (results[0]) {
                      $('#latitude,#longitude').show();
                      $('#registeraddress').val(results[0].formatted_address);
                      infowindow.setContent(results[0].formatted_address);
                      infowindow.open(map, marker);
                  }
              }
          });

          google.maps.event.addListener(marker, 'dragend', function() {

              geocoder.geocode({'latLng': marker.getPosition()}, function(results, status) {
                  if (status == google.maps.GeocoderStatus.OK) {
                      if (results[0]) {
                          $('#registeraddress').val(results[0].formatted_address);
                          infowindow.setContent(results[0].formatted_address);
                          infowindow.open(map, marker);
                      }
                  }
              });
          });

      }
      google.maps.event.addDomListener(window, 'load', initialize);
</script>
```

3.2.2 login.php

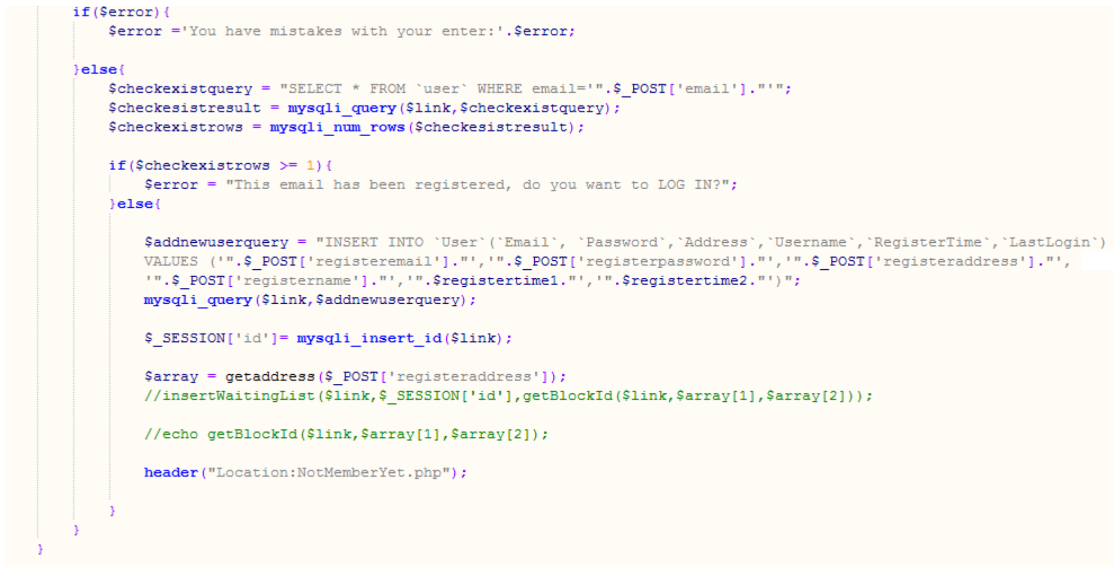
The main function of this php file is for a new user to sign up and enable old user to log in. This two different functions can be distinguished by get the value of $_POST['submit'].

At the beginning of this file, we have ***session_start ()_*** which make it possible to store UserID in the session, this variable needs to be used almost on every PHP file of this project.

The script used for new user to register:



```
include("connection.php");

$error = '';


if(isset($_POST['submit'])){

    if($_POST['submit']=="Sign up"){

        date_default_timezone_set("America/New_York");
        $registertime1 = date("Y-m-d");
        $registertime2 = date("Y-m-d H:i:s");

        if(!$_POST['registeraddress']){
            $error .= "<br />Please enter your Address.";
        }

        if(!$_POST['registername']){
            $error .= "<br />Please enter your Name.";
        }

        //Validate email address.
        if(!$_POST['registeremail']){
            $error .= "<br />Please enter your email.";
        }else if(!filter_var($_POST['registeremail'],FILTER_VALIDATE_EMAIL)){
            $error .= "<br />Please enter validate email.";
        }

        //Validate password, change password to md5 code.and save it in database.
        if(!$_POST['registerpassword']){
            $error .= "<br />Please enter your password";
        }else if(strlen($_POST['registerpassword'])<6 && !preg_match('`[A-Z]`','`[0-9]`',$_POST['registerpassword'])){
            $error .= "<br />Your password is invalid.";
        }
```



```
        if($error){
            $error ='You have mistakes with your enter:'.$error;

        }else{
            $checkexistquery = "SELECT * FROM `user` WHERE email='".$_POST['email']."'";
            $checkesistresult = mysqli_query($link,$checkexistquery);
            $checkexistrows = mysqli_num_rows($checkesistresult);

            if($checkexistrows >= 1){
                $error = "This email has been registered, do you want to LOG IN?";
            }else{

                $addnewuserquery = "INSERT INTO `User`(`Email`, `Password`,`Address`,`Username`,`RegisterTime`,`LastLogin`)
                VALUES ('".$_POST['registeremail']."','".$_POST['registerpassword']."','".$_POST['registeraddress']."',
                '".$_POST['registername']."','".$registertime1."','".$registertime2."')";
                mysqli_query($link,$addnewuserquery);

                $_SESSION['id']= mysqli_insert_id($link);

                $array = getaddress($_POST['registeraddress']);
                //insertWaitingList($link,$_SESSION['id'],getBlockId($link,$array[1],$array[2]));

                //echo getBlockId($link,$array[1],$array[2]);

                header("Location:NotMemberYet.php");

            }
        }
    }
```

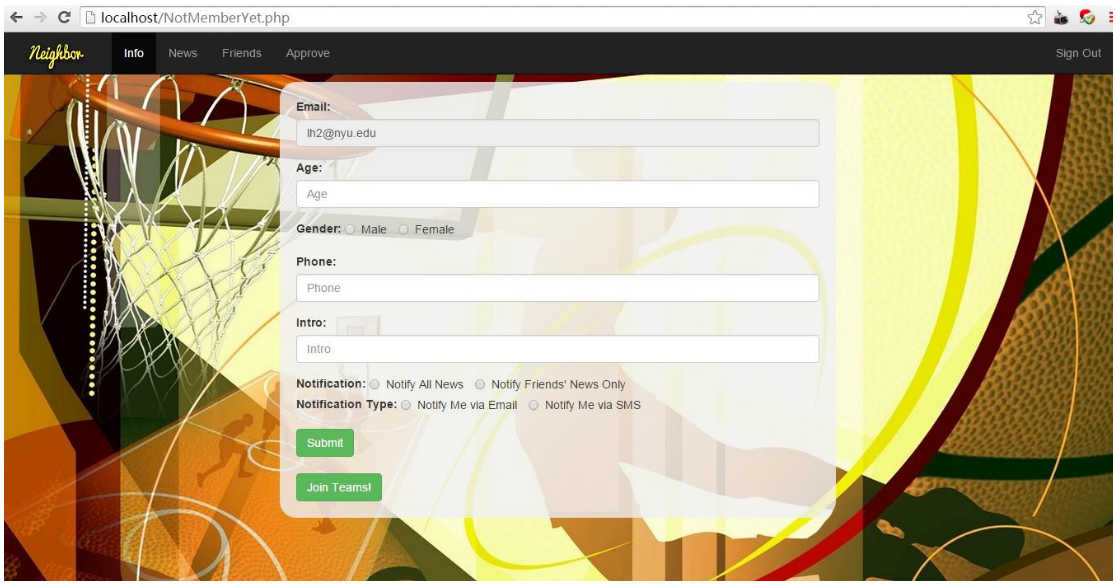
The script used for existing user to log in

```php
if ($_POST['submit']=="Log in") {

    date_default_timezone_set("America/New_York");
    $lastaccesstime = date("Y-m-d H:i:s");

    $loginquery= "SELECT * FROM `User` WHERE Email='".$_POST['loginemail']."' AND Password='".$_POST['loginpassword']."'";
    $loginresult = mysqli_query($link,$loginquery);
    $rows = mysqli_fetch_array($loginresult);

    if($rows){
        $_SESSION['id']=$rows['UserID'];
        //$_SESSION['blockid']=$rows['BlockId'];
        //$_SESSION['id']=$loginidresult;

        $updatelastaccesstime = "UPDATE `User` SET LastLogin='".$lastaccesstime."'WHERE UserId='".$_SESSION['id']."'";

        mysqli_query($link,$updatelastaccesstime);

        if(isMember($link,$_SESSION['id'])){
            header('Location:message.php');
        }else{
            header('Location:NotMemberYet.php');
        }

        //Redirect to logged in page
    }else{
        $error = "We could not find a user with that email and password!";
    }

}
```

3.2.3 updateinfo.php&NotMemberYet.php

In this page, user can type in user information, and press Submit button, then all the information will be updated in the database.

For a new user, NotMemberYet.php is used, and user can submit personal information and set notification preferences.



Email: lh2@nyu.edu

Age: Age

Gender: ○ Male ○ Female

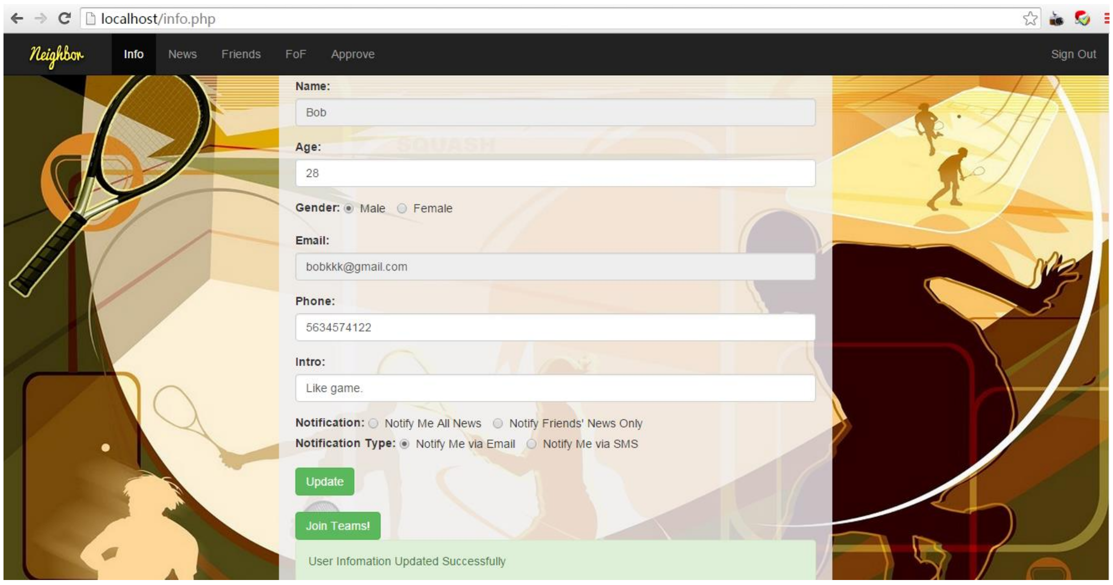
Phone: Phone

Intro: Intro

Notification: ○ Notify All News ○ Notify Friends' News Only

Notification Type: ○ Notify Me via Email ○ Notify Me via SMS

For an existing user, the user can modify his/her personal information in info.php. And the registered username and Email are displayed.


localhost/info.php

Name: Bob

Age: 28

Gender: [✓ Male] Female

Email: bobkkk@gmail.com

Phone: 5634574122

Intro: Like game.

Notification: Notify Me All News Notify Friends' News Only

Notification Type: [✓ Notify Me via Email] Notify Me via SMS

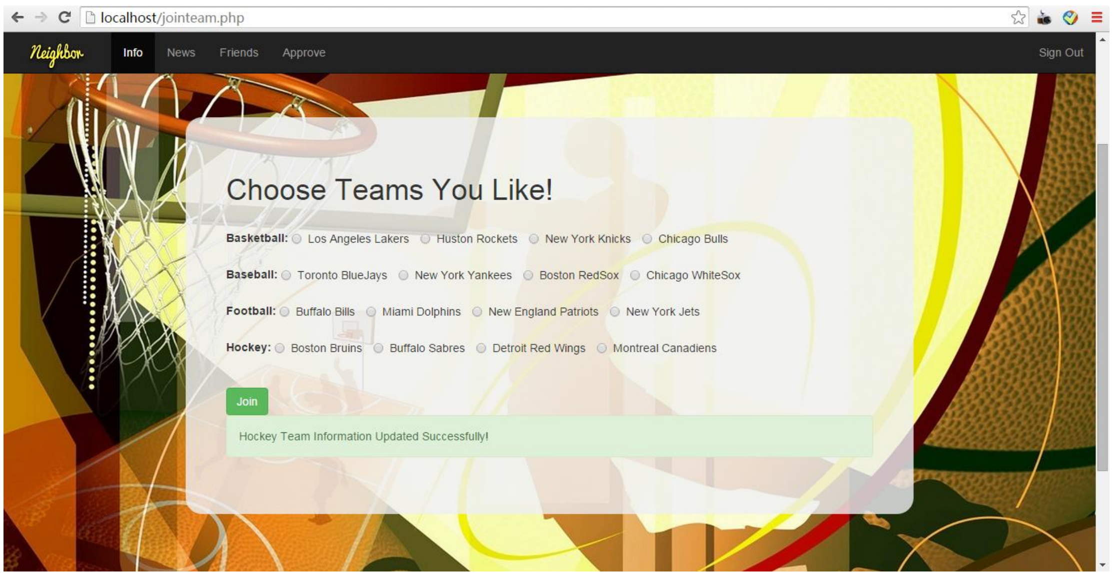
User Infomation Updated Successfully

The script used to update personal information:

```
session_start();
include("DataRetrieval.php");
include("connection.php");
error_reporting(0);

$success = '';
$error = '';

$getexsitinginfo = "SELECT * FROM `User` WHERE UserId = '".$_SESSION['id']."'";
$result = mysqli_query($link, $getexsitinginfo);
$results = mysqli_fetch_array($result);

if(isset($_POST['submit']) && $_POST['submit']=='Update'){


    $updateinfo = "UPDATE `User` SET Age='".$_POST['updateage']."',Gender='".$_POST['updategender']."',Phone='".$_POST['updatephone']."',
    Intro='".$_POST['updateintro']."',Notification='".$_POST['updatenotification']."',NotificationType='".$_POST['updatenotificationtype']."'
    WHERE UserId = '".$_SESSION['id']."'";

    if(mysqli_query($link,$updateinfo)){
        $success = "User Infomation Updated Successfully";
    }else{
        $error = "Update Failure";
    }

}
```

3.2.4 Jointeam.php&waitinglist.php

In jointeam.php, the user can choose team him/her likes and make request to join.


localhost/jointeam.php

Choose Teams You Like!

Basketball: Los Angeles Lakers Huston Rockets New York Knicks Chicago Bulls

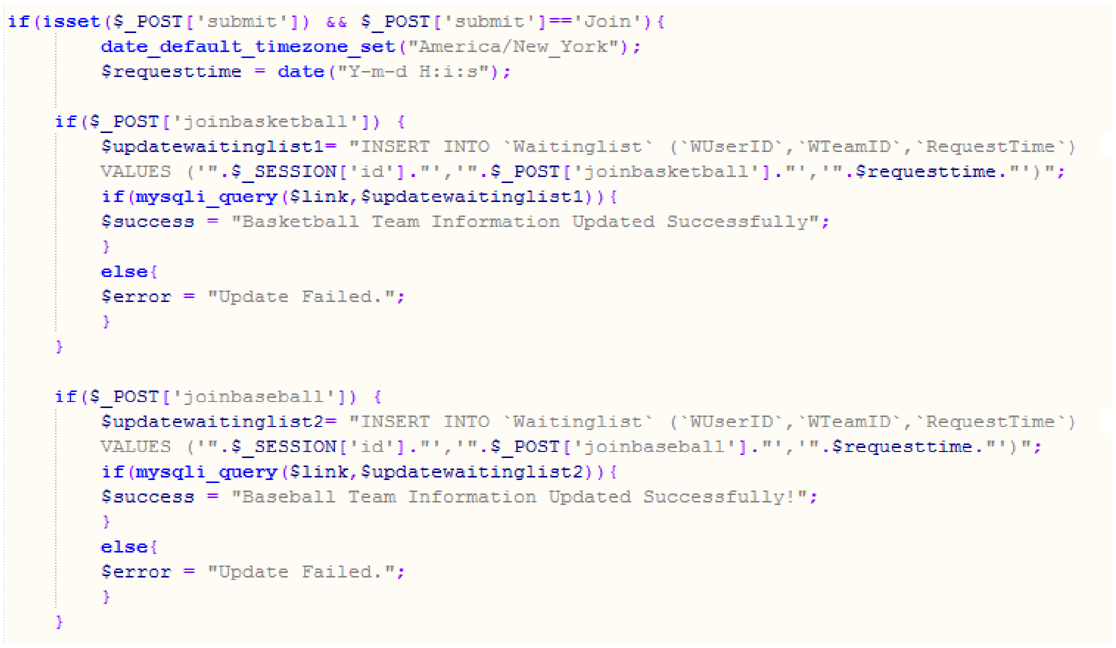
Baseball: Toronto BlueJays New York Yankees Boston RedSox Chicago WhiteSox

Football: Buffalo Bills Miami Dolphins New England Patriots New York Jets

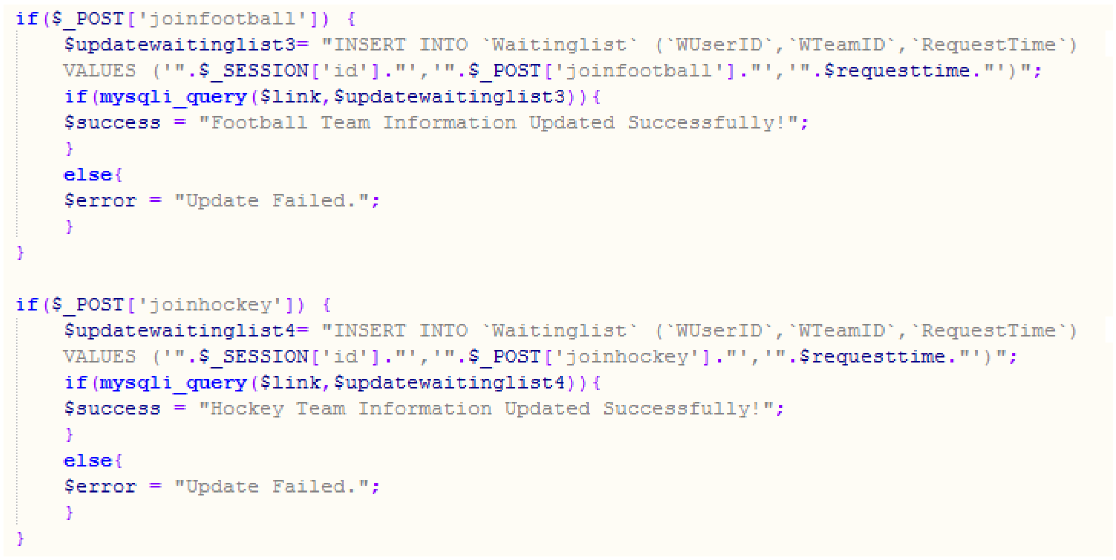
Hockey: Boston Bruins Buffalo Sabres Detroit Red Wings Montreal Canadiens

Hockey Team Information Updated Successfully!

In waitinglist.php, the requests and request time are recorded in the Waitinglist Table.



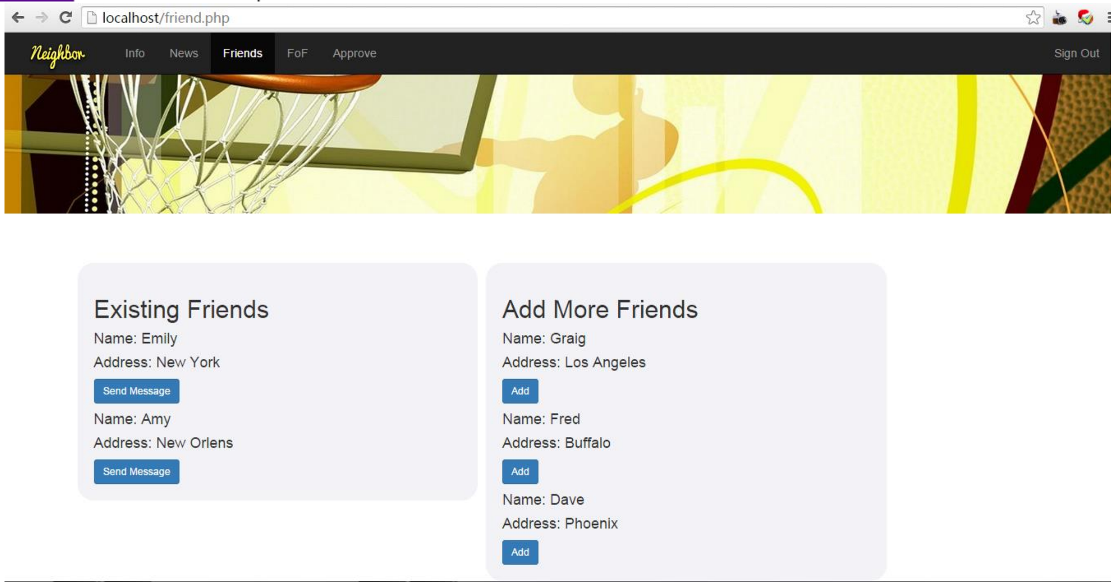
```
if(isset($_POST['submit']) && $_POST['submit']=='Join'){
        date_default_timezone_set("America/New_York");
        $requesttime = date("Y-m-d H:i:s");

    if($_POST['joinbasketball']) {
        $updatewaitinglist1= "INSERT INTO `Waitinglist` (`WUserID`,`WTeamID`,`RequestTime`)
        VALUES ('".$_SESSION['id']."','".$_POST['joinbasketball']."','".$requesttime."')";
        if(mysqli_query($link,$updatewaitinglist1)){
        $success = "Basketball Team Information Updated Successfully";
        }
        else{
        $error = "Update Failed.";
        }
    }

    if($_POST['joinbaseball']) {
        $updatewaitinglist2= "INSERT INTO `Waitinglist` (`WUserID`,`WTeamID`,`RequestTime`)
        VALUES ('".$_SESSION['id']."','".$_POST['joinbaseball']."','".$requesttime."')";
        if(mysqli_query($link,$updatewaitinglist2)){
        $success = "Baseball Team Information Updated Successfully!";
        }
        else{
        $error = "Update Failed.";
        }
    }
```



```
if($_POST['joinfootball']) {
    $updatewaitinglist3= "INSERT INTO `Waitinglist` (`WUserID`,`WTeamID`,`RequestTime`)
    VALUES ('".$_SESSION['id']."','".$_POST['joinfootball']."','".$requesttime."')";
    if(mysqli_query($link,$updatewaitinglist3)){
    $success = "Football Team Information Updated Successfully!";
    }
    else{
    $error = "Update Failed.";
    }
}

if($_POST['joinhockey']) {
    $updatewaitinglist4= "INSERT INTO `Waitinglist` (`WUserID`,`WTeamID`,`RequestTime`)
    VALUES ('".$_SESSION['id']."','".$_POST['joinhockey']."','".$requesttime."')";
    if(mysqli_query($link,$updatewaitinglist4)){
    $success = "Hockey Team Information Updated Successfully!";
    }
    else{
    $error = "Update Failed.";
    }
}
```

3.2.5 friend.php

Friend page lists all existing friends the user has and all possible friends, i.e. users in the same fans club that can be friends.



Existing Friends

Name: Emily

Address: New York

Send Message

Name: Amy

Address: New Orlens

Send Message

Add More Friends

Name: Graig

Address: Los Angeles

Add

Name: Fred

Address: Buffalo

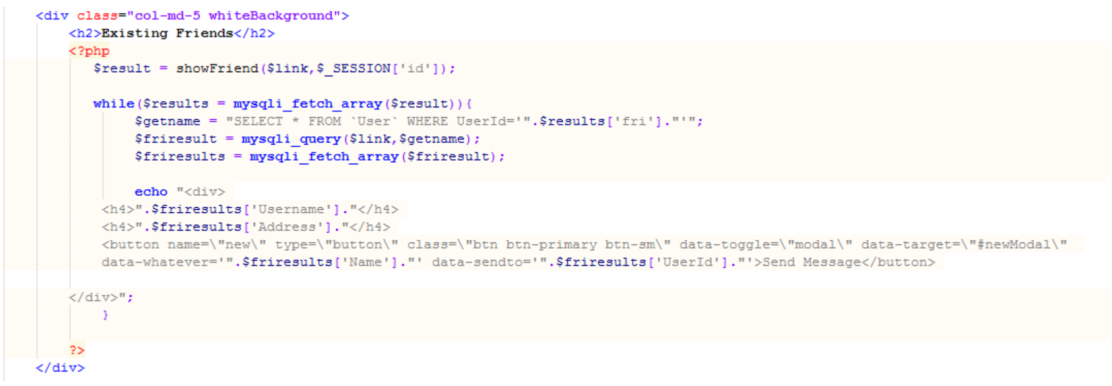
Add

Name: Dave

Address: Phoenix

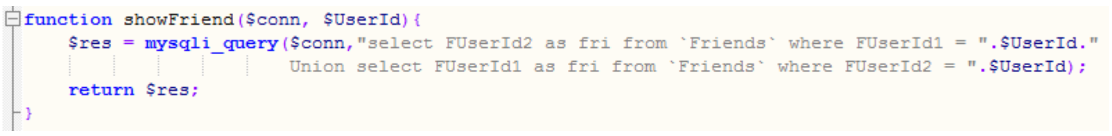
Add

The script used to show existing friends



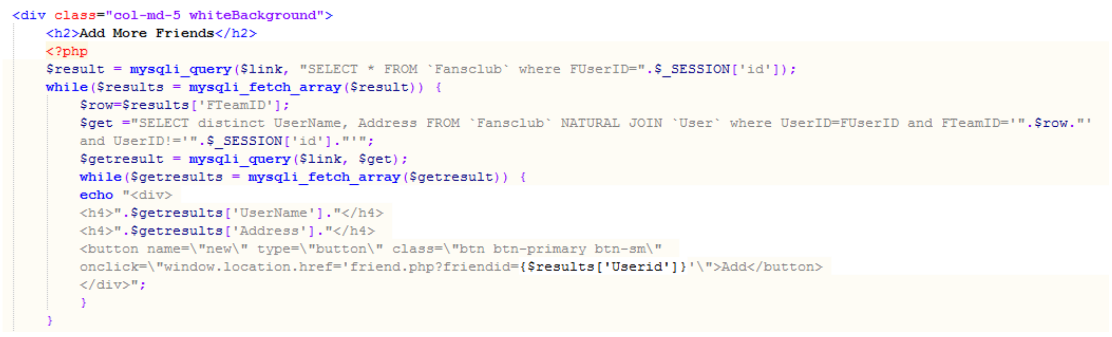
```
<div class="col-md-5 whiteBackground">
    <h2>Existing Friends</h2>
    <?php
      $result = showFriend($link,$_SESSION['id']);

      while($results = mysqli_fetch_array($result)){
          $getname = "SELECT * FROM `User` WHERE UserId='".$results['fri']."'";
          $friresult = mysqli_query($link,$getname);
          $friresults = mysqli_fetch_array($friresult);

          echo "<div>
        <h4>".$friresults['Username']."</h4>
        <h4>".$friresults['Address']."</h4>
        <button name=\"new\" type=\"button\" class=\"btn btn-primary btn-sm\" data-toggle=\"modal\" data-target=\"#newModal\"
        data-whatever='".$friresults['Name']."' data-sendto='".$friresults['UserId']."'>Send Message</button>

    </div>";
        }

    ?>
</div>
```

Function showFriend in DataRetrieval.php



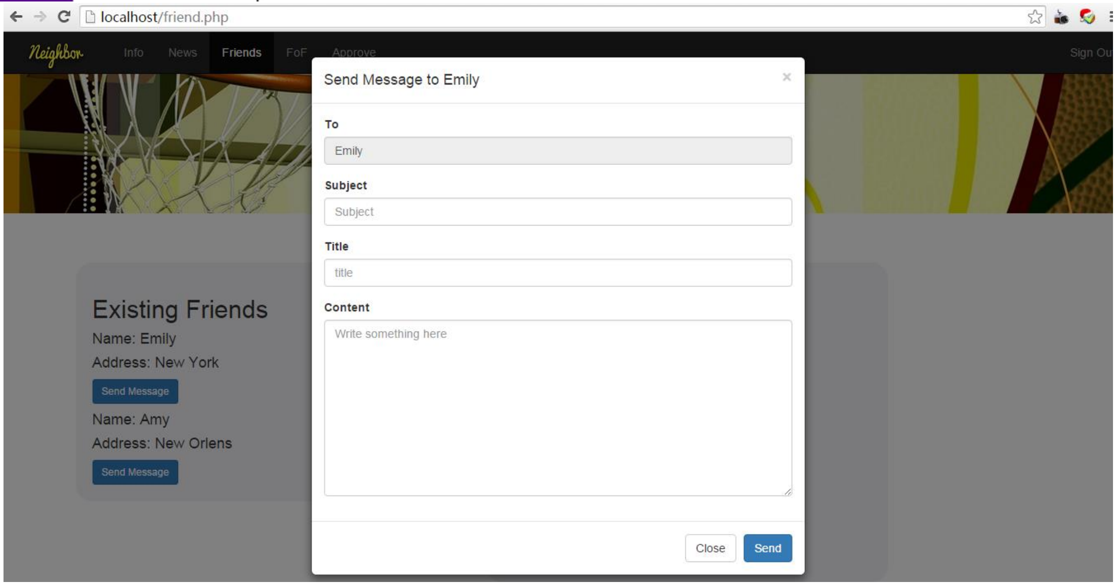
```
function showFriend($conn, $UserId){
    $res = mysqli_query($conn,"select FUserId2 as fri from `Friends` where FUserId1 = ".$UserId."
                        Union select FUserId1 as fri from `Friends` where FUserId2 = ".$UserId);
    return $res;
}
```

The script used to show all possible friends.



```
<div class="col-md-5 whiteBackground">
    <h2>Add More Friends</h2>
    <?php
    $result = mysqli_query($link, "SELECT * FROM `Fansclub` where FUserID=".$_SESSION['id']);
    while($results = mysqli_fetch_array($result)) {
        $row=$results['FTeamID'];
        $get ="SELECT distinct UserName, Address FROM `Fansclub` NATURAL JOIN `User` where UserID=FUserID and FTeamID='".$row."'
        and UserID!='".$_SESSION['id']."'";
        $getresult = mysqli_query($link, $get);
        while($getresults = mysqli_fetch_array($getresult)) {
        echo "<div>
        <h4>".$getresults['UserName']."</h4>
        <h4>".$getresults['Address']."</h4>
        <button name=\"new\" type=\"button\" class=\"btn btn-primary btn-sm\"
        onclick=\"window.location.href='friend.php?friendid={$results['Userid']}'\">Add</button>
        </div>";
        }
    }
```

And also in the Existing Friends lists, we can choose one friend and send him/her a message.


localhost/friend.php

Existing Friends

Name: Emily

Address: New York

Name: Amy

Address: New Orlens

Send Message to Emily

To: Emily

Subject: Subject

Title: title

Content: Write something here

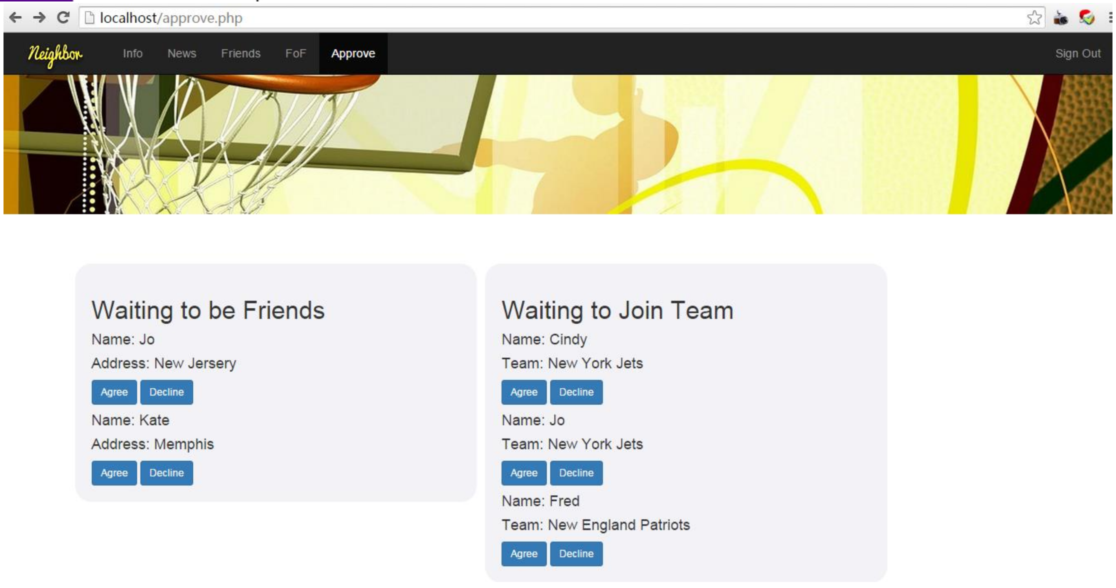
The script used to create the Send Message form:

```
<div class="modal fade" tabindex="-1" role="dialog" id="newModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                <h4 class="modal-title">Reply</h4>
            </div>
            <form method="post" >
                <div class="modal-body">
                    <div class="form-group">
                        <label>To</label>
                        <input class="form-control" type="text" name="sendto" id="sendto">
                    </div>

                    <div class="form-group">
                        <label>Subject</label>
                        <input class="form-control" type="text" name="sendsubject" placeholder="Subject">
                    </div>

                    <div class="form-group">
                        <label>Title</label>
                        <input class="form-control" type="text" name="sendtitle" placeholder="title">
                    </div>


                    <div class="form-group">
                        <label>Content</label>
                        <textarea class="form-control" name="sendcontent" rows="10" cols="80" class="center-block">Write something here</textarea>
                    </div>
                </div>

                <div class="modal-footer">
                    <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
                    <input type="submit" name="submit" class="btn btn-primary" value="Send">
                </div>
            </form>

        </div><!-- /.modal-content -->
    </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
```

3.2.6 approve.php

This page lists all users who want to get the approval to join in one team. If one joining request is approved by at least 3 users in the fans club, this request is considered to be approved. By the trigger Approve, this user will be removed from the Waitinglist table and added to the FansClub table.

Besides, this page lists all users who want to be friends. And the user can approve or decline their requests via button. If a friend request if approved, the trigger Friend is activated, and the corresponding record will be deleted from the FDWaitinglist table and added to the Friends table.



Waiting to be Friends

Name: Jo
Address: New Jersery
Agree Decline
Name: Kate
Address: Memphis
Agree Decline

Waiting to Join Team

Name: Cindy
Team: New York Jets
Agree Decline
Name: Jo
Team: New York Jets
Agree Decline
Name: Fred
Team: New England Patriots
Agree Decline

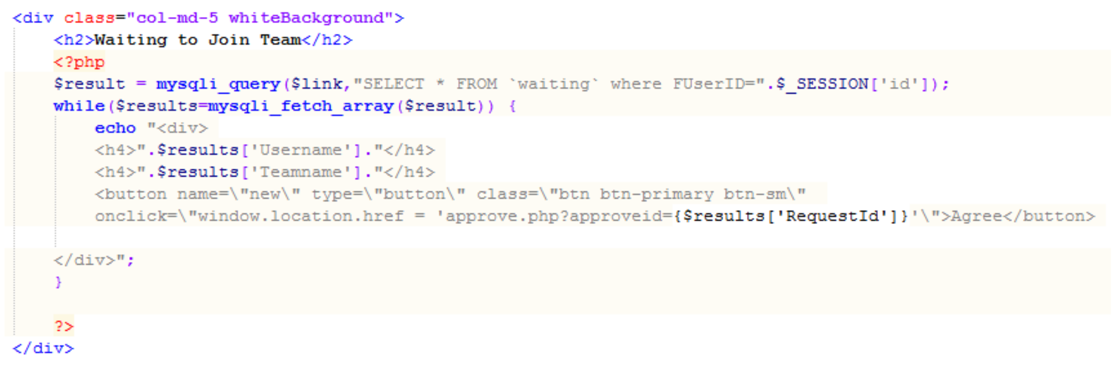
The script used to show Friends Waitinglist:

```
<div class="col-md-5 whiteBackground">
    <h2>Waiting to be Friends</h2>
    <?php
    $result = showFriendWaitingList($link,$_SESSION['id']);

    while($results = mysqli_fetch_array($result)){
        $getname = "SELECT * FROM `User` WHERE UserID='".$results['FWUserID1']."'";
        $friresult = mysqli_query($link,$getname);
        $friresults = mysqli_fetch_array($friresult);

        echo "<div>
        <h4>".$friresults['Username']."</h4>
        <h4>".$friresults['Address']."</h4>
        <button name=\"new\" type=\"button\" class=\"btn btn-primary btn-sm\"
        onclick=\"window.location.href = 'approve.php?friendid={$results['FWUserId1']}'\">Agree</button>

    </div>";
    }

    ?>
</div>
```

Function showFriendWaitingList in DataRetrieval.php

```
//be requested as a friend,so user here should be in FWUSERI2
function showFriendWaitingList($conn, $UserId){
    $res = mysqli_query($conn,"select * from `FdWaitingList` Where FWUserId2 = ".$UserId);
    return $res;
}
```

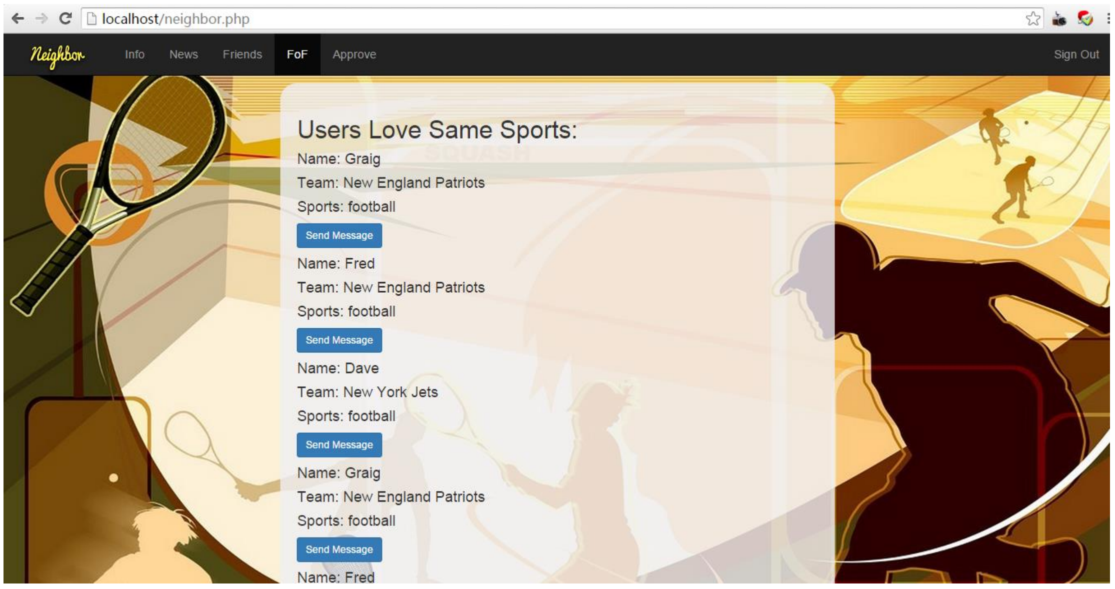
The script used to show Team Waitinglist, in which the view waiting is used:



```
<div class="col-md-5 whiteBackground">
    <h2>Waiting to Join Team</h2>
    <?php
    $result = mysqli_query($link,"SELECT * FROM `waiting` where FUserID=".$_SESSION['id']);
    while($results=mysqli_fetch_array($result)) {
        echo "<div>
        <h4>".$results['Username']."</h4>
        <h4>".$results['Teamname']."</h4>
        <button name=\"new\" type=\"button\" class=\"btn btn-primary btn-sm\"
        onclick=\"window.location.href = 'approve.php?approveid={$results['RequestId']}'\">Agree</button>

    </div>";
    }

    ?>
</div>
```

3.2.7 neighbor.php

This page lists all other users who love the same sports. And also we can choose one neighbor and send him/her a message. The view neighbor is used.



Users Love Same Sports:

Name: Graig
Team: New England Patriots
Sports: football
Send Message

Name: Fred
Team: New England Patriots
Sports: football
Send Message

Name: Dave
Team: New York Jets
Sports: football
Send Message

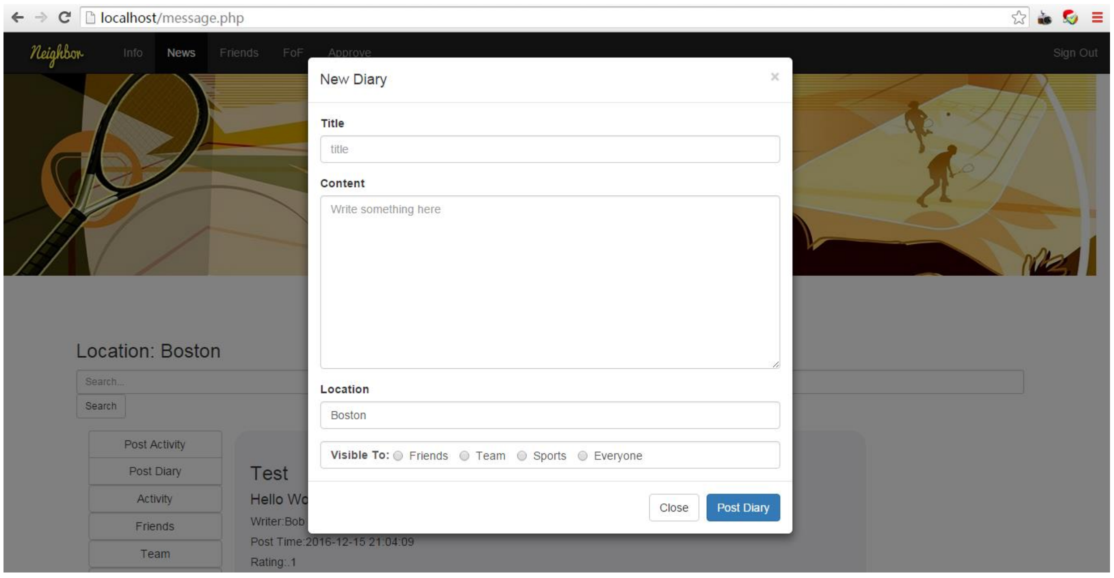
Name: Graig
Team: New England Patriots
Sports: football
Send Message

Name: Fred

3.2.8 message.php&getmessage.php

In this page, a user can post a new diary containing text and multimedia content, such as pictures and videos. And a user can like, dislike, or comment on others' posts. Besides, the user can choose a tag to sort out the news visible to certain users only. And also, the user can find the interested news via searching for a keyword.
In addition, a user can post activities and find activities posted by other users under the activity tag.

1) Post a new diary



New Diary

Title: title

Content: Write something here

Location: Boston

Visible To: ○ Friends ○ Team ○ Sports ○ Everyone

Location: Boston

Test

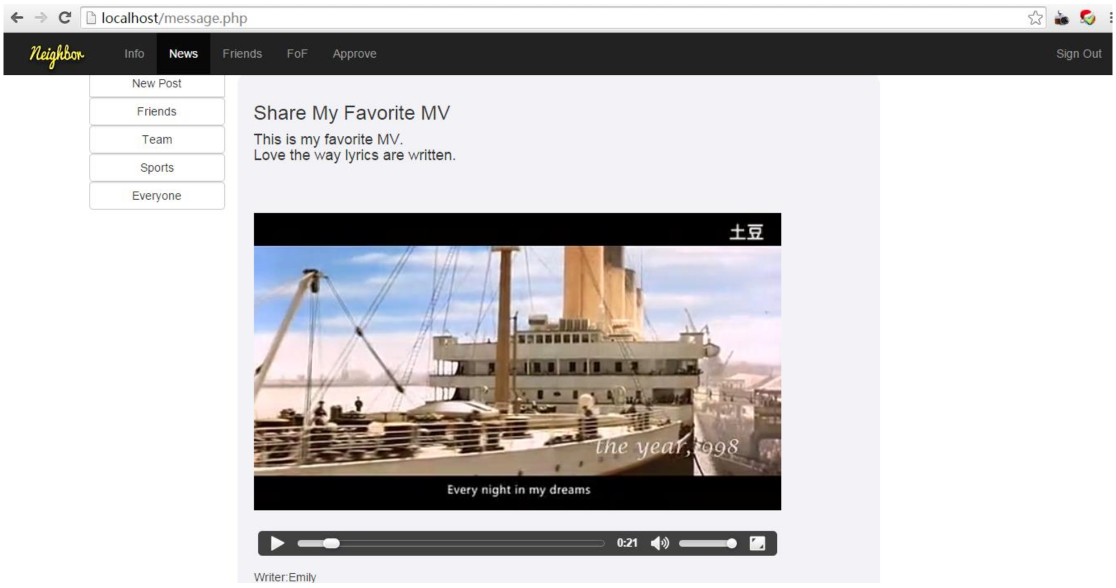
Hello Wo

Writer:Bob

Post Time:2016-12-15 21:04:09

Rating:.1

Diary with video:

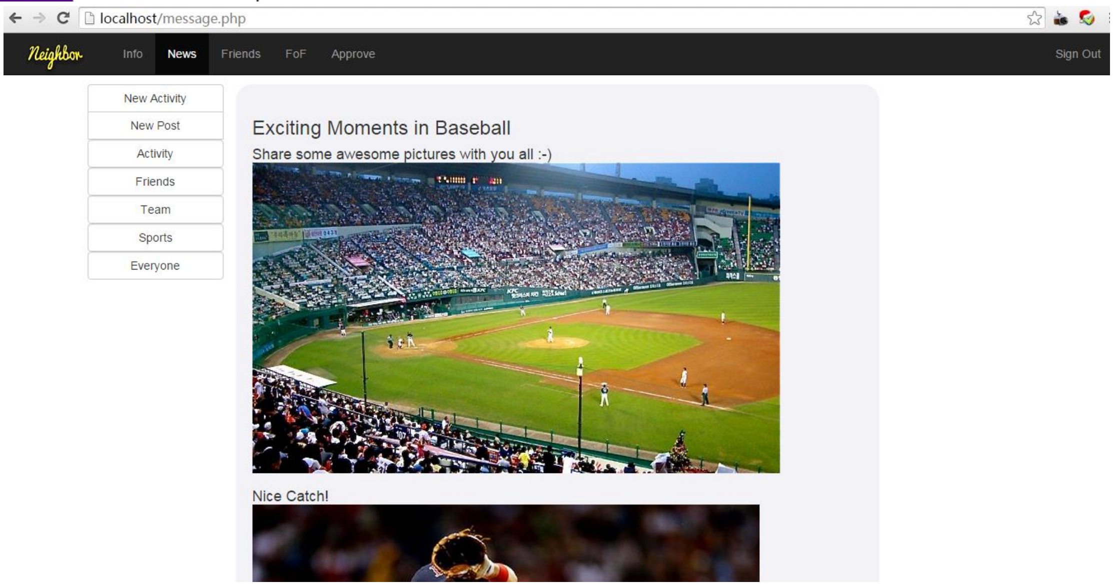


Share My Favorite MV

This is my favorite MV.
Love the way lyrics are written.

Writer:Emily

Diary with pictures



Exciting Moments in Baseball

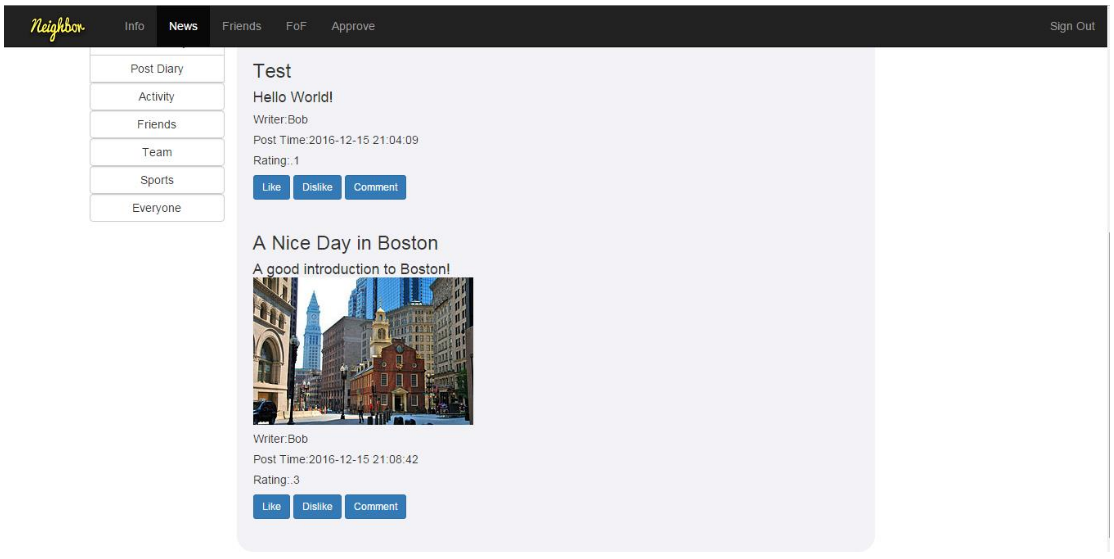
Share some awesome pictures with you all :-)

Nice Catch!

2) Like&Dislike&Sort by Visible Type

The attribute Rating reflects the number of Likes and Dislikes. If a user chooses to “like” a diary, the Rating of this diary is incremented by 1 in the News table. On the contrary, if a user chooses to “dislike” a diary, the Rating of this diary is decreased by 1 in the News table.

And also the user can click on the leftmost tag to view news posted to certain group of users only. For each diary, the title, content, writer name, and post time are displayed in the diary entry.



Test

Hello World!

Writer:Bob

Post Time:2016-12-15 21:04:09

Rating:.1

A Nice Day in Boston

A good introduction to Boston!

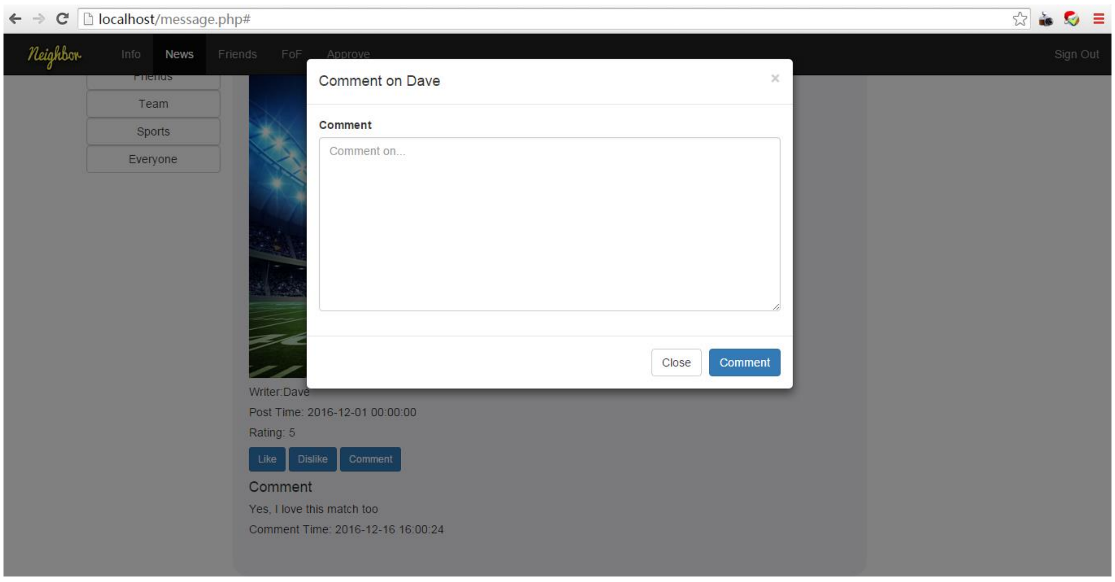
Writer:Bob

Post Time:2016-12-15 21:08:42

Rating:.3

3) Comment

The user can comment on an existing diary. And the comment will be inserted to the diary in the News table. If a diary has comments, the comment will be displayed as well.



Comment on Dave

Comment

Comment on...

Writer:Dave

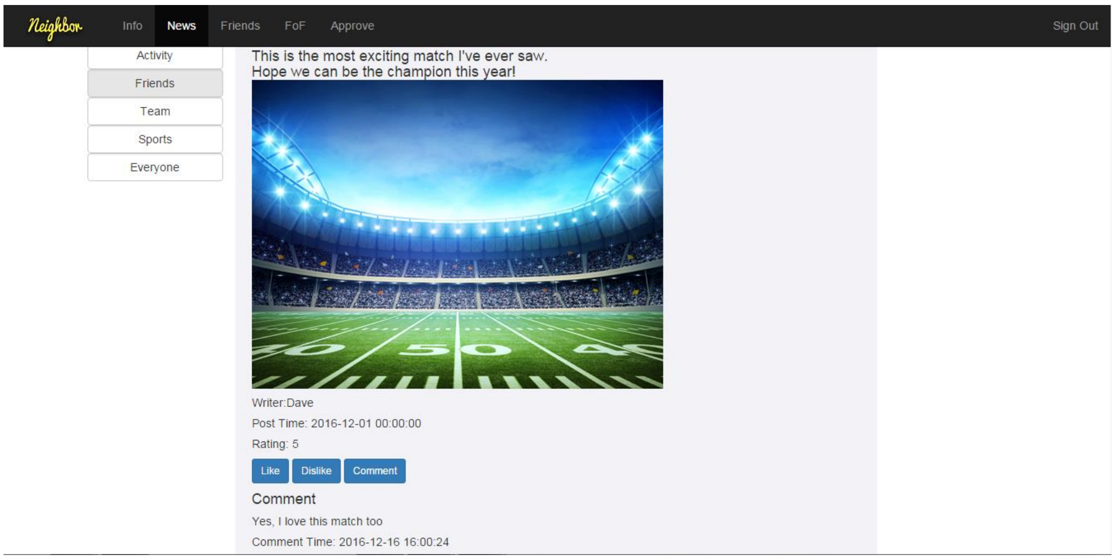
Post Time: 2016-12-01 00:00:00

Rating: 5

Comment

Yes, I love this match too

Comment Time: 2016-12-16 16:00:24



This is the most exciting match I've ever saw.
Hope we can be the champion this year!

Writer:Dave

Post Time: 2016-12-01 00:00:00

Rating: 5

Comment

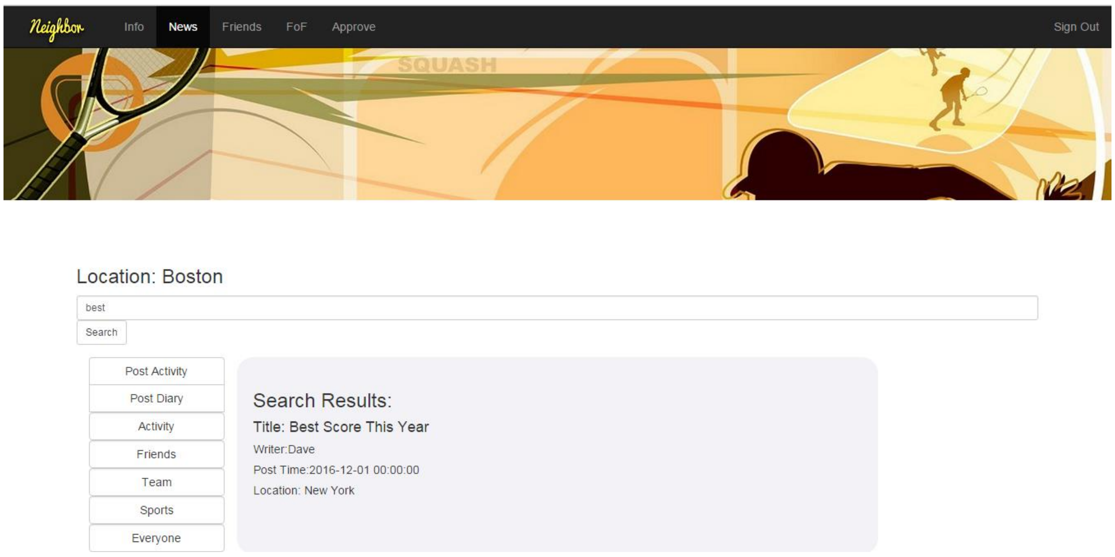
Yes, I love this match too

Comment Time: 2016-12-16 16:00:24

4) Search by keyword

The user can search for diaries containing certain keyword, whether it is in the title, diary content, writer’s name, or location.

We can find in the search result for “best” that a diary whose title contains “best” is returned:



Location: Boston

best

Search Results:

Title: Best Score This Year

Writer:Dave

Post Time:2016-12-01 00:00:00

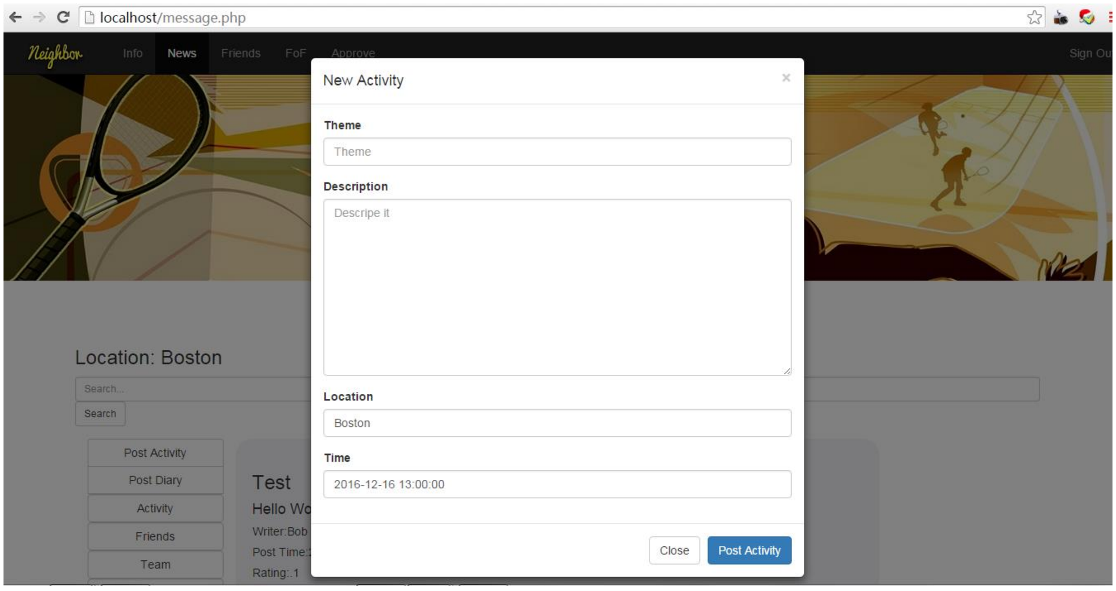
Location: New York

The script used to search in getmessage.php

```
//Keyword Search
if(isset($_POST['submit'])&&$_POST['submit']=="Search"){
    $search = "SELECT Title, Username, PostTime, Location FROM `News` NATURAL JOIN `User` WHERE User.UserID=News.PosterID and (Title LIKE
    '%".$_POST['search']."%' OR Body LIKE '%".$_POST['search']."%' OR Username LIKE '%".$_POST['search']."%' OR Location LIKE '%".$_POST['search']."%')";
    $searchresult = mysqli_query($link, $search);
}
```

5) Post new activity

The user can post a new activity containing activity theme, description, location, and start time. And a user can use the Activity tag to view activities posted by other users.



New Activity

Theme: Theme

Description: Descripe it

Location: Boston

Time: 2016-12-16 13:00:00

Location: Boston

Search...

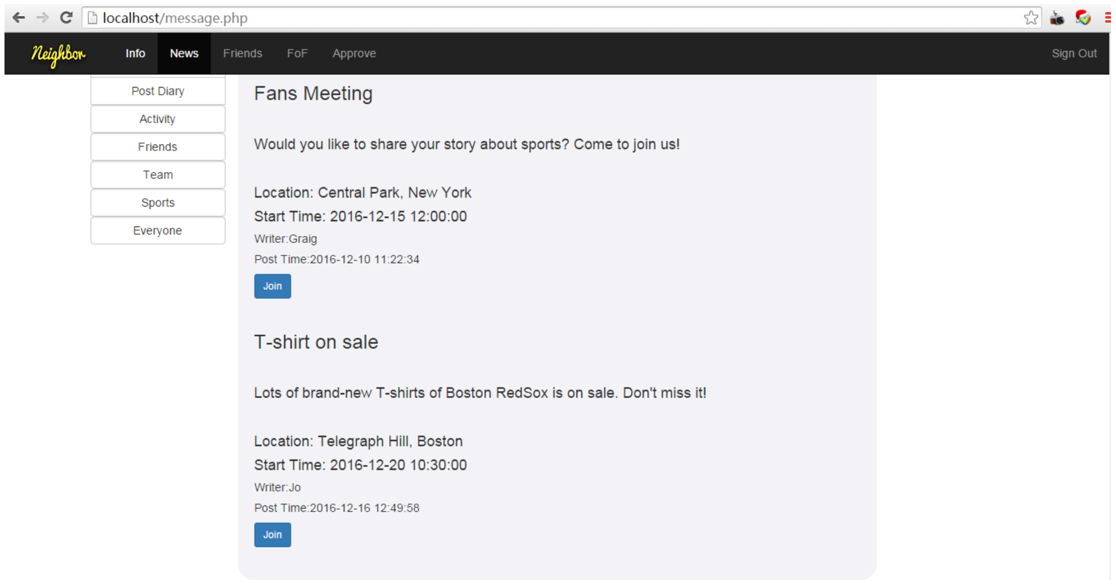
Test

Hello Wo

Writer:Bob

Post Time:

Rating: 1



Fans Meeting

Would you like to share your story about sports? Come to join us!

Location: Central Park, New York
Start Time: 2016-12-15 12:00:00
Writer:Graig
Post Time:2016-12-10 11:22:34

T-shirt on sale

Lots of brand-new T-shirts of Boston RedSox is on sale. Don't miss it!

Location: Telegraph Hill, Boston
Start Time: 2016-12-20 10:30:00
Writer:Jo
Post Time:2016-12-16 12:49:58

3.2.9 Others

There are some supportive files containing necessary functions: connect.php, DataRetrieval.php, getaddress.php.
And some information pages such as About.html and Developer.html are used to provide background information.

4. Group Work

For Project 1,
First, Lanlan Hu chose the third Application Scenarios for our project.
Then Juexiu Wu came up with the initial Relation Schema and ER diagram.
After that, we discussed, modified and decided the final Relation Schema and ER diagram for our project together.
Next Lanlan Hu create database schema, together with key, foreign key, and other constraints for our project.
Finally, Juexiu Wu finished the report.
For Project 2,
Lanlan Hu worked on Log in&Sign up, Update personal information, Join team&Approve, Add friend&Approve.
Juexiu Wu worked on Bootstrap, FoF, Post diary&Search$Post activity.
Lanlan Hu and Juexiu Wu finished the report together.