Table of Contents

Activities

Add new activity

```
// Use dbinfo function to add new activity
$dbinfo->save_activity("Activity Description");

// The above function should do something like the following, which will be defined in dbinfo:
$user = $this->username();
$day = $this->day();
$hour = $this->hour();
$minute = $this->minute() + 1;
If $minute > 59
        $minute -= 60;
        $hour++;
If $hour > 23
        $hour -= 24;
        $day++;
insert into user_activity set username = '$user', day = $day, hour = $hour, minute = $minute, activity =
'$activity'
$dbinfo->update_all();
```

View activity for a specific user

```
select day, hour, minute, activity from user_activity where username = '$username' AND activity !=
'Current Time' order by day desc, hour desc, minute desc
```

Edit activity

```
//We will only allow changing of the activity attribute of user_activity
update user_activity set activity = '$activity' where username = '$username', day = '$day', hour =
'$hour', minute = '$minute'
```

Delete activity

```
// Determine if the activity we want to delete is the most recent
select username, day, hour, minute from user_activity order by day desc, hour desc, minute desc limit 1

// If ($username = $row['username'] && day hour minute...)
// must handle special

// Get a "random" user
select username from user limit 1
$randuser = $row['username']
```

```
// insert bogus activity because we need to not have the time “go backward”
insert into user_activity values ('$randuser', '$day', '$hour', '$minute', 'Current Time')

// remove the activity we are interested in
delete from user_activity where username = '$username', day ..... hour ....minute...
```

// **else**   from this: If ($username = $row['username'] && day hour minute...)

```
delete from user_activity where username = '$username', day ..... hour ....minute...
```

Registration

Add Registration (normal)

```
//1. See if username already exists. Count will = 1 or 0
select count(*) as count from user where username = '$username'

// 2. Insert user information into the db
insert into user values ('$username', '$password', false, '$realname', '$birthdate', '$street', '$city', '$state',
'$zip', '$phone', '$email', '$card_type', '$card_number', '$card_expire', 'default.jpg', 'N/A');

// 3. Save user activity
$dbinfo->save_activity(“Registered”);
```

Add Registration (using admin priv)

```
// 1 & 3 are the same, 2 is different:
insert into user values ('$username', '$password', $is_admin, '$realname', '$birthdate', '$street', '$city',
'$state', '$zip', '$phone', '$email', '$card_type', '$card_number', '$card_expire', 'default.jpg', 'N/A');
```

View Registration Info

```
select is_admin, realname, birth_date, shipping........., phone, email, card_type, card_number,
card_expire from user where username = '$username'
```

Edit Registration

```
// Cannot change username, all other fields are ok
update user set password = '$newpassword', ...etc
$dbinfo->save_activity(“Updated Registration Info”);
```

Delete Registration

Only way we will allow deletion of a registered user is if the following apply:

- User has no listed items (ie. Not the seller for any item)
- User is not the buyer of any items (ie. Not the winner of any auction)
- User is not the last user
- Current user is an Admin

```
if ($dbinfo->is_admin())
        // OK so far
// is there others users left beside this one?
```

```
select username from user where username != '$username'

if (mysql_num_rows($result) >= 1)

        // OK so far

// does user have any listed items or is the buyer of any listed items?

// First perform the following operation: Update all ItemListing buyer information for closed auctions
select count(*) as count from item_listing where seller = '$username' OR buyer = '$username'

if (count == 0)

        // OK to delete user

// Determine if the most recent activity is from current user
select username, day, hour, minute from user_activity order by day desc, hour desc, minute desc limit 1
// If ($username = $row['username'] ),
// Get a “random” user
select username from user where username != '$username' limit 1
$randuser = $row['username']

// insert bogus activity because we need to not have the time “go backward”
insert into user_activity values ('$randuser', '$day', '$hour', '$minute', 'Current Time')

// remove all activity of the desired user
delete from user_activity where username = '$username'
// remove all bids of the desired user

delete from bids_on where username = '$username'

// remove user

delete from user where username = '$username'

$dbinfo->save_activity(“Deleted user '$username'”); // remember, this activity is being saved for the
admin user
```

Profiles

Add Profile

// this is done at registration automatically and filled in with dummy values

Edit Profile

update user set profile_picture = '$pic', profile_description = '$desc' where username = '$username'
$dbinfo->save_activity("Updated Profile");

View a list of usernames for viewing a profile

select user.username as username, day, hour, minute from user, user_activity where user.username = user_activity.username AND activity = 'Registered' order by user.username;

// Iterate through results displaying a table that looks like:

Username	Registered	
jimbob	0	0:00
billbob	2	2:12
abob	1	13:54

View Profile

// First perform the following operation: Update all ItemListing buyer information for closed auctions
select profile_picture, profile_description from user where username = '$username'

select * from item_listing where (seller = '$username' AND buyerfeedbackforseller_rating != -1) OR (buyer = '$username' AND sellerfeedbackforbuyer_description != '') order by end_day desc, end_hour desc, end_minute desc;

//Show on profile page first the username, the picture, the description, then an area that has feedback and ratings from items sold, then an area that has feedback from sellers of items purchased.

// The feedback will be an iteration through the previous query where you check that $row['seller'] == $username and display appropriate type of output or if $row['buyer'] == $username and then display appropriate output.

Delete Profile

// This will not delete bid history or feedback, only profile picture and profile description

update user set profile_picture = '', profile_description = 'N/A' where username = '$username'

$dbinfo->save_activity("Deleted Profile");

Item Listings

Add ItemListing

```
// check if unique
select count(*) as count from item_listing where title = '$title' AND category = '$category' AND
end_day = '$end_day' AND end_hour = '$end_hour' AND end_minute = '$end_minute'  AND seller =
'$username'

// if count = 0
// make sure we aren't entering older than current time.
$dbinfo->update_time();
if ($day >= $dbinfo->day() &&
        $hour >= $dbinfo->hour() &&
        $minute > $dbinfo->minute())

// put into db and save activity
// make certain that starting_price and current_price are currently the same value
// also make certain buyerfeedbackforseller_rating is set to

insert into item_listing set title = '$title', seller = '$username', category = '$category', end_day =
'$end_day', end_hour = '$end_hour', end_minute = '$end_minute', description = '$description',
shipping_cost = '$shipcost', shipping_method = '$shipmethod', starting_price = '$price', current_price =
'$price', picture = '$picture', buyer = '', buyerfeedbackforseller_description = '',
buyerfeedbackforseller_rating = -1, sellerfeedbackforbuyer_description = '';

$dbinfo->save_activity("Listed '$title' for auction.");
```

Edit ItemListing

```
// If editing description, shipping_cost, shipping_method, or picture, perform normal
// update query
update item_listing set <fields> where <pk criteria>

// Can't edit "seller"
// Can't edit "current_price"

// If editing starting_price, there must not be any bids on the item: you can do a query to get the
starting_price and current_price and see if they are equal, if they are you can edit starting price, if not
then you cannot – this is because whenever a bid is made the current_price is updated as well

// No editing of primary keys – the user must create a new item listing and delete the current one
(which is only allowed if there are no bids)

// Depending on if an admin or not, we will save activity:
```

```
// not admin
$dbinfo->save_activity("Item '$item_title' edited.");

// admin
$dbinfo->save_activity_for($username, "Item '$item_title' edited.");
```

Note to self: the save_activity_for function should check the following:

- current user is an admin
- $username is a valid user
- do similar things that save_activity does
- return success or failure

Update all ItemListing buyer information for closed auctions

```
// For example: This query will be run when the time is advanced
// Get all item listings that are closed and haven't been updated
select title, category, end_day, end_hour, end_minute, seller from item_listing where end_day <= $day
AND hour <= $hour and minute <= $minute and buyer = ''

while ($row = mysql_fetch_assoc($result)){
    select username, bid_amount from bids_on where item_title = $row['title'] AND <all other
    ItemListing Pks> order by bid_amount desc
    // get the row
    if !isset($row['username']) // then no winner
        update item_listing set buyer = 'None', set current_price = starting_price where
        <ItemListing Pks>
    else
        $buyer = $row['username']
        $price = $row['bid_amount']
        update item_listing set buyer = '$buyer', set current_price = '$price' where <ItemListing
        Pks>
}
```

Note: the updating of current_price will most likely already be done, since it is done at bid time – but I guess this way is not hurting things.

Delete ItemListing

```
// Assumption is that $username refers to the seller of the item, but not necessarily the person/admin
doing the deleting (though it is possible)
// Can only delete item listing if there are no bidders
select count(*) as count from bids_on where item_title = '$title' AND item_seller = '$username' AND
item_category = '$category' AND item_end_day = '$endday' AND item_end_hour = '$endhour' AND
item_end_minute = '$endminute'
```

```
if (count == 0)
        // there are no bidders, OK to delete
delete from item_listing where title = '$title' AND seller = '$username' AND category = '$category'
AND end_day = '$endday' AND end_hour = '$endhour' AND end_minute = '$endminute'
$dbinfo->save_activity("Item listing '$title' deleted.");
```

View a specific item listing

```
// First perform the following operation: Update all ItemListing buyer information for closed auctions
select title, seller, category, end_day, end_hour, end_minute, description, shipping_cost,
shipping_method, starting_price, current_price, picture, buyer, buyerfeedbackforseller_description,
buyerfeedbackforseller_rating from item_listing where title = '$title' AND seller = '$username' AND
category = '....'  AND end_day = '....' AND end_hour ....  AND end_minute

if (buyer != '') // auction open
        select bid_amount, username,  bid_day, bid_hour, bid_minute, from bids_on where item_title =
        '$title' AND <all other ItemListing Pks> order by bid_amount desc

        ~~// fetch first row and save bid_amount in variable for current price~~
        // This current_price value is now directly stored in the ItemListing

        // display title, seller, category, end_day, end_hour, end_minute, description, shipping_cost,
        shipping_method, starting_price, current_price, picture

        // iterate through query results and display username, bid amount, day, hour, minute

else if (buyer == 'None')
        // display title, seller, category, end_day, end_hour, end_minute, description, shipping_cost,
        shipping_method, starting_price, "Auction closed without a buyer", picture
else // buyer is set to a username
        q1 = select bid_amount, username,  bid_day, bid_hour, bid_minute, from bids_on where
        item_title = '$title' AND <all other ItemListing Pks> order by bid_amount desc

        ~~// fetch first row and save bid_amount in variable for current price~~
        // This current_price value is now directly stored in the ItemListing

        // display title, seller, category, end_day, end_hour, end_minute, description, shipping_cost,
        shipping_method, starting_price, sell price (current_price), "This auction listing is closed",
        picture

        // display (and link) to winning username

        if ( buyerfeedbackforseller_rating != -1)
                // display feedback rating
        if ( buyerfeedbackforseller_description != '')
                // display feedback desc
        // iterate through q1 results and display username, bid amount, day, hour, minute
```

View item listings in a category (by time remaining)

```
// First perform the following operation: Update all ItemListing buyer information for closed auctions
q1 = select title, seller, end_day, end_hour, end_minute, current_price from item_listing where category
= '$category' AND end_day >= '$day' AND end_hour >= '$hour' AND end_minute > '$minute' order by
end_day desc, end_hour desc, end_minute desc

for each entry in q1
        // display all info from q1
```

View item listings in a category (by title)

```
// First perform the following operation: Update all ItemListing buyer information for closed auctions
q1 = select title, seller, end_day, end_hour, end_minute, current_price from item_listing where category
= '$category' AND end_day >= '$day' AND end_hour >= '$hour' AND end_minute > '$minute' order by
title

for each entry in q1
        // display all info from q1
```

OR REVERSE LISTING

```
// First perform the following operation: Update all ItemListing buyer information for closed auctions
q1 = select title, seller, end_day, end_hour, end_minute, current_price from item_listing where category
= '$category' AND end_day >= '$day' AND end_hour >= '$hour' AND end_minute > '$minute' order by
title desc
```

View item listings in a category (by seller)

```
// First perform the following operation: Update all ItemListing buyer information for closed auctions
q1 = select title, seller, end_day, end_hour, end_minute, current_price from item_listing where category
= '$category' AND end_day >= '$day' AND end_hour >= '$hour' AND end_minute > '$minute' order by
seller
```

View item listings in a category (by current bid, lowest bid at the top)

```
// First perform the following operation: Update all ItemListing buyer information for closed auctions
q1 = select title, seller, end_day, end_hour, end_minute, current_price from item_listing where category
= '$category' AND end_day >= '$day' AND end_hour >= '$hour' AND end_minute > '$minute' order by
current_price
```

Bidding

Making a bid on an item

// check that current user isn't seller of the auction – ideally we won't show a bid on button if the user is
// viewing an item they are selling, but...
select count(*) as count from item_listing where username = '$curruser' AND <rest of Pks>
// if count = 0, then ok to bid

// check that user isn't already the highest bidder
select username, bid_amount from bids_on where item_title = '$title' AND <rest of ItemListing Pks in BidsOn> order by bid_day desc, bid_hour desc, bid_minute desc limit 1

// fetch the row and compare against the current user. If there is a match then don't allow!

// check that the current bid is the highest
Use PHP to compare bid_amount from previous query to $bid_amount coming from the bidder. If $bid_amount is > $row['bid_amount'], then it is OK

update bids_on set display_notification = 'y' where display_notification = 'n' AND <item_title, item_seller, item_category, item end "time" all match – this doesn't include "bid day", "bid hour", "bid minute">

insert into item_listing values ('$username', '$item_title', '$item_seller', '$item_category', '$item_end_day', '$item_end_hour', '$item_end_minute', '$curr_day', '$curr_hour', '$curr_minute', '$bid_amount', 'n');

update item_listing set current_price = '$bid_amount' where <match Pks of the ItemListing with current saved values>

$dbinfo->save_activity("Bid on item '$item_title'"); // which will also advance the time!

Editing a bid

// The only editing of a bid we should allow is whether the notification is displayed or not, otherwise any other values might violate the rules of the system

// Note: $notification_choice' is either 'y', 'n', or 'c'
// 'y' – display the notification, 'n' don't display the notification, 'c' – user canceled display of notification

update bids_on set display_notification = '$notification_choice' where <match PKs – username, title, seller, category, end day – hour -minute, bid day- hour- minute>

Deleting a bid

// A bid may only be deleted if it is not the highest bid
select username from bids_on where <match current values of the Pks of bids_on except username> order by bid_day desc, bid_hour desc, bid_minute desc limit 1

// If fetched username is the same as the username of the entry we want to delete (which may not be current user), disallow – otherwise wipe it out:
delete from bids_on where <match current values of the Pks>

Viewing all bids on an item listing

See: View a specific item listing

Viewing all bids made by a specific user

select item_title, item_seller, item_category, item_end_day, item_end_hour, item_end_minute, bid_day, bid_hour, bid_minute, bid_amount from bids_on where username = '$desired_user' order by bid_day desc, bid_hour desc, bid_minute desc
// display info fields from query

Notifications & Feedback

Show notification to winning buyer of an auction to leave feedback

// Assume $username is the current user and the one we are interested.
// First perform the following operation: Update all ItemListing buyer information for closed auctions
select * from item_listing where buyer = '$username' AND buyerfeedbackforseller_rating = -1 order by end_day desc, end_hour desc, end_minute desc

// It is assumed that rating and description will be updated simultaneously and that the rating is initially -1

// iterate through results which are sorted by time and display boxes to leave feedback (which will show relevant item information so the user knows what they are leaving feedback for, also let the user know they were the winner)

// If they fill out the box:
update item_listing set buyerfeedbackforseller_rating = $rating, buyerfeedbackforseller_description = '$desc' where buyer = '$username' AND <rest of Pks needed to uniquely identify>

Show all notifications to seller of all their closed auctions that have no bids

// First perform the following operation: Update all ItemListing buyer information for closed auctions
select * from item_listing where seller = '$username' AND buyer = 'None' order by end_day desc, end_hour desc, end_minute desc

Show all notifications to seller of all their closed auctions that there was a buyer

// First perform the following operation: Update all ItemListing buyer information for closed auctions
select * from item_listing where seller = '$username' AND (buyer != 'None' OR buyer != '') order by end_day desc, end_hour desc, end_minute desc

Leaving feedback to the buyer if you are the seller of a closed auction

// This will be on the home page and will show something to the effect of “Item <item title> sold to <username>” with a small box below to optionally leave feedback. To get this list:

// First perform the following operation: Update all ItemListing buyer information for closed auctions
select * from item_listing where seller = '$username' AND (buyer != '' OR buyer != 'None') AND sellerfeedbackforbuyer_description != '' order by end_day desc, end_hour desc, end_minute desc

// iterate through results which are sorted by time and display information described above

Deleting feedback

Only admin should be able to delete feedback.

If deleting feedback from buyer for seller make sure to set buyerfeedbackforseller_rating = -1, buyerfeedbackforseller_description = ''

If deleting feedback from seller for buyer make sure to set sellerfeedbackforbuyer_description = ''

In both cases, update activity:

```
$dbinfo->save_activity_for($username, “Feedback removed for seller $seller for the item $title”);
$dbinfo->save_activity(“Admin of $buyer: Feedback removed for seller $seller for the item $title”);
```

OR

```
$dbinfo->save_activity_for($username, “Feedback removed for buyer $buyer for the item $title”);
$dbinfo->save_activity(“Admin of $seller: Feedback removed for buyer $buyer for the item $title”);
```

Editing feedback

Only admin should be able to edit feedback.

If editing feedback from buyer for seller make sure to update buyerfeedbackforseller_rating, and buyerfeedbackforseller_description

If editing feedback from seller for buyer make sure to update sellerfeedbackforbuyer_description

In both cases, update activity:

```
$dbinfo->save_activity_for($username, “Feedback edited for seller $seller for the item $title”);
$dbinfo->save_activity(“Admin of $buyer: Feedback edited for seller $seller for the item $title”);
```

OR

```
$dbinfo->save_activity_for($username, “Feedback edited for buyer $buyer for the item $title”);
$dbinfo->save_activity(“Admin of $seller: Feedback edited for buyer $buyer for the item $title”);
```

Show notification to the user that an auction has closed since their last visit they they participated in

```
// This should only be displayed ONCE
// First perform the following operation: Update all ItemListing buyer information for closed auctions
```

// Get the last time the user was logged in
select day, hour, minute from user_activity where username = '$username' AND activity = 'Logged Out' order by day desc, hour desc, minute desc limit 1

// If the num rows returned is 0, then set vars for day hour minute to be 0 0 0, else..

// getting closed auctions they bid on
select * from bids_on where username = '$username' AND item_end_day <= '$curr_day' AND item_end_hour <= '$curr_hour' AND item_end_minute <= '$curr_minute' AND item_end_day >= '$lastvisit_day' AND item_end_hour >= '$lastvisit_hour' AND item_end_minute > '$lastvisit_minute' order by item_end_day desc, item_end_hour desc, item_end_minute desc
// iterate through them

// getting closed auctions they are the seller for
select * from item_listing where seller = '$username' AND end_day <= '$curr_day' AND end_hour <= '$curr_hour' AND end_minute <= '$curr_minute' AND end_day >= '$lastvisit_day' AND end_hour >= '$lastvisit_hour' AND end_minute > '$lastvisit_minute' order by end_day desc, end_hour desc, end_minute desc

Show all notifications to the user where they were outbid in an auction that is still going on that they haven't chosen to cancel

// First perform the following operation: Update all ItemListing buyer information for closed auctions
select * from bids_on where username = '$username' AND item_end_day >= '$curr_day' AND item_end_hour >= '$curr_hour' AND item_end_minute > '$curr_minute' AND display_notification = 'y' order by item_end_day desc, item_end_hour desc, item_end_minute desc

Hide a notification to the user where they were outbid in an auction that is still going on

// First perform the following operation: Update all ItemListing buyer information for closed auctions
update bids_on set display_notification = 'c' where username = '$username' AND item_title = '<passed in var>' AND item_seller = '<passed in var>' AND item_category = '<passed in var>' AND item_end_day = '$<passed in var>' AND item_end_hour = '$<passed in var>' AND item_end_minute = '$<passed in var>' AND bid_day = '<passed in var>' AND bid_hour = '<passed in var>' AND bid_minute = '<passed in var>'

Miscellaneous

Home Page Notifications – idea of how we will display all varieties together sorted by time

Perform all related queries for notifications. Specifically, the following are relevant:

- Show all notifications to the user where they were outbid in an auction that is still going on that they haven't chosen to cancel
- Show notification to the user that an auction has closed since their last visit they they participated in
- Show all notifications to seller of all their closed auctions that there was a buyer
- Show all notifications to seller of all their closed auctions that have no bids
- Show notification to winning buyer of an auction to leave feedback
- View activity for a specific user

While we are doing all of these queries, we will be getting results that we want to iterate over and produce some html output. Instead of directly echoing the output we will be doing the following:

- Maintain a global counter, say $x that is initially value 0.
- Maintain a global array, say $notifications which will be indexed using $x

As we are iterating through the query we will do the following things:

- Stuffing the time
  - A day will be stuffed from the front with zeroes so that it is 5 digits (kind of arbitrary, but ok)
  - An hour will be stuffed so it is 2 digits, same with minute
- For the current value of $x, we will set $notifications equal to the stuffing:
  - $notifications[$x] = $stuffed_day.$stuffed_hour.$stuffed_minute
- We will generate the appropriate html code for the query we are parsing (could be just a message with the date and time or could be code for a form for feedback, dependends on the notification)
- That html code will be appended to the $notifications array at the current position $x
  - $notifications[$x] = $notifications[$x].' '.$generated_html
- We reverse sort the array so it is time ordered by most recent time:
  - rsort($notifications)
- We now echo all the notifications:

```
for ($i = 0; $i < count($notifications); $i++){
        echo substr(    $notifications[$i],
                        strpos($notifications[$i], " "),
                        strlen($notifications[$i]
                );
}
```

Note: Prioritizing notifications based on type that happen at the same time

Ex: Auction closed and you are the winner.

We can stuff after the time a “priority code” such that the number with the highest priority will display highest than other notifications from the same time.