The Bagger Management Project

Michael Corder
Advisor: Dr. Zizhong Wang

Abstract

The Bagger Management Project is a research project undertaken with express intent of allowing greater management functionality within Armed Forces Commissaries. While cashiers are government employees, baggers are not. The independent contractor status these military dependents and spouses hold means that funding for any management software or equipment is nearly non-existent. As a result, most managers must use pen and paper to keep track of a plethora of employees. This research project streamlines management in these facilities allowing for a simplified work environment for all involved.

In order to accomplish this goal, many features were crafted and implemented in the project. The most prominent security feature is a login database that keeps track of users, admins, and superadmins as well as the various functions each has at their disposal. This ensures that any individual who accesses the site will see only the pages they were meant to see and minimizes potential harm to the scheduling.

Users are bagger employees who are able to check their schedule and request time off. Administrators, meanwhile, are similarly able to check their schedules, but they have access to other functions as well. First and foremost, the administrator is able to hire on new baggers and easily add the new employees to the database, setting the schedule of these employees as they see fit. Just as easily, administrators can also remove employees from the database upon employment termination. Administrators are also able to view requested time off, modify the schedules of employees, and alter any information concerning employees. One of the most useful features however, is the ability for administrators to check the daily schedule to see which employees are scheduled that day and each employee's associated "bagger fee." (An amount each employee pays to the manager based upon scheduled hours) The superadministrator, has access to these features as well as the ability to modify administrators.

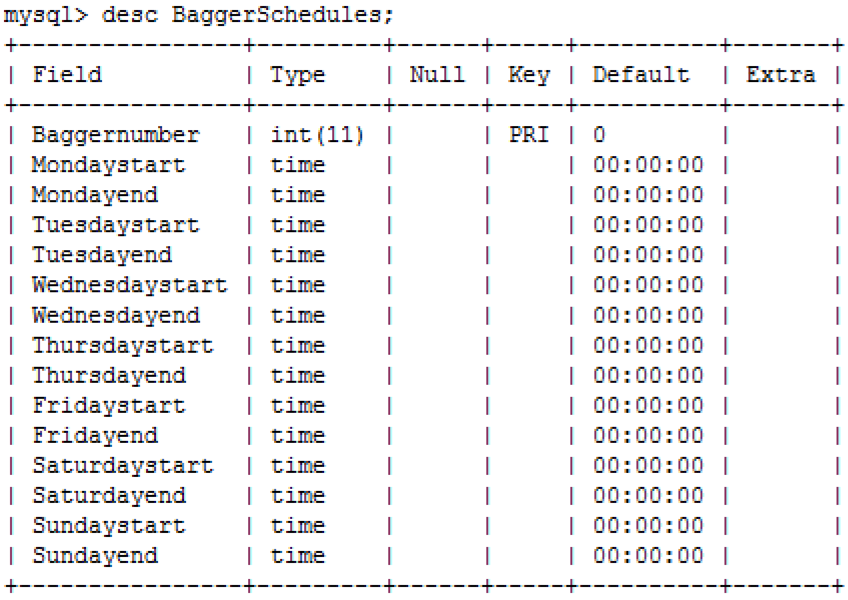
This project makes extensive use of the database technology in the MYSQL language as well as website logic and data manipulation through PHP and HTML. Through application of the principles of each programming language, a simplification of management has been created that has the potential to become the management tool of choice throughout armed forces establishments.

Design

In order to accomplish these goals I first had to realize the needs and inherent properties of bagger managers and baggers themselves. First, each employee is assigned a bagger number that is unique relative to the other baggers; however, baggers are often reassigned numbers as time goes by and turnover occurs. The goal then was to hold all of the bagger information but also allow any of it to be modified at any time. Second, each bagger employee pays a “bagger fee” for being scheduled relative to their hours. This bagger fee then, needed to be calculated using information the manager would input without requiring mathematical computation on their part. Finally, managers are elected which means there is a chance that an employee may become a manager or a manager revert to a bagger. Thus, any employee needs to have the capability of promotion while managers must maintain the possibility for demotion.

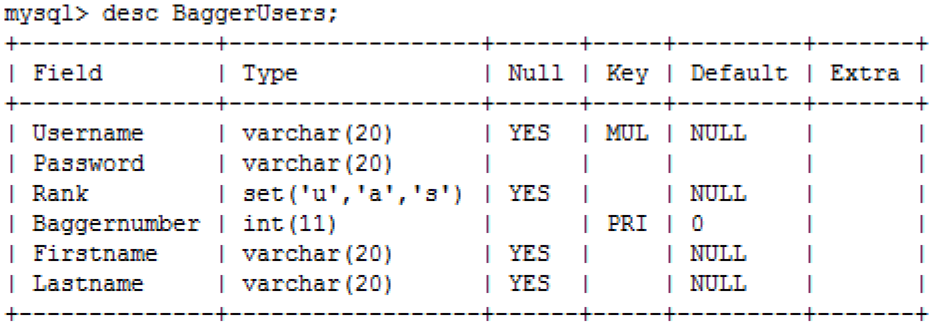
Tables

Armed with this information, I created three tables within my database in order to hold the relevant information. The first of which is “BaggerSchedules.” This tables makes of the TIME datatype in MYSQL in order to store the scheduled work hours for employees and managers alike. Employees and managers are scheduled on a weekly basis, in that they work the same set of shifts each week, by day of the week, not date. The bagger number of the employee is also tracked in order to properly keep track of to whom a schedule belongs.



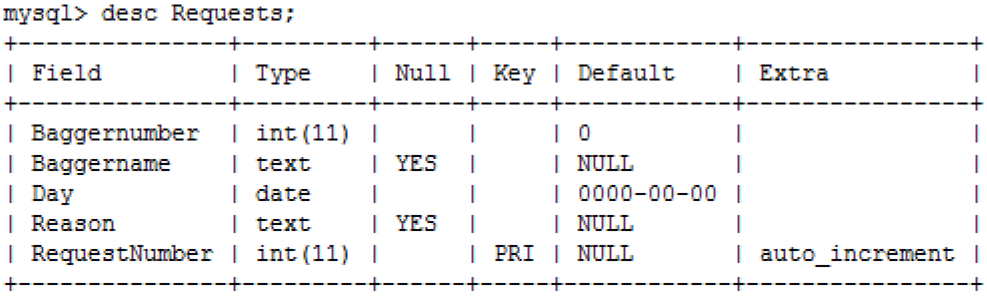
```
mysql> desc BaggerSchedules;
+-----------------+---------+------+-----+----------+-------+
| Field           | Type    | Null | Key | Default  | Extra |
+-----------------+---------+------+-----+----------+-------+
| Baggernumber    | int(11) |      | PRI | 0        |       |
| Mondaystart     | time    |      |     | 00:00:00 |       |
| Mondayend       | time    |      |     | 00:00:00 |       |
| Tuesdaystart    | time    |      |     | 00:00:00 |       |
| Tuesdayend      | time    |      |     | 00:00:00 |       |
| Wednesdaystart  | time    |      |     | 00:00:00 |       |
| Wednesdayend    | time    |      |     | 00:00:00 |       |
| Thursdaystart   | time    |      |     | 00:00:00 |       |
| Thursdayend     | time    |      |     | 00:00:00 |       |
| Fridaystart     | time    |      |     | 00:00:00 |       |
| Fridayend       | time    |      |     | 00:00:00 |       |
| Saturdaystart   | time    |      |     | 00:00:00 |       |
| Saturdayend     | time    |      |     | 00:00:00 |       |
| Sundaystart     | time    |      |     | 00:00:00 |       |
| Sundayend       | time    |      |     | 00:00:00 |       |
+-----------------+---------+------+-----+----------+-------+
```

The second table of interest is “BaggerUsers” which holds all of the information necessary for keeping track of users, their logins, and their rank. The user’s “Rank” is the most important part of this table, and implemented as follows. A bagger employee is rank “u” for user. A rank “u” user can only access a limited number of pages and functionality, none of which involve modifying other employees. An “a” rank user is a manager who is able to modify employee records, schedules, and view relevant manager information. A rank “s” user is a superadministrator, a rank which I reserve for myself. This allows me to use a set of convenient forms (as opposed to a MYSQL interface) in order alter the user records in the database. The other fields are fairly self-explanatory and used for more easily keeping track of employees.



```
mysql> desc BaggerUsers;
+--------------+------------------+------+-----+---------+-------+
| Field        | Type             | Null | Key | Default | Extra |
+--------------+------------------+------+-----+---------+-------+
| Username     | varchar(20)      | YES  | MUL | NULL    |       |
| Password     | varchar(20)      |      |     |         |       |
| Rank         | set('u','a','s') | YES  |     | NULL    |       |
| Baggernumber | int(11)          |      | PRI | 0       |       |
| Firstname    | varchar(20)      | YES  |     | NULL    |       |
| Lastname     | varchar(20)      | YES  |     | NULL    |       |
+--------------+------------------+------+-----+---------+-------+
```

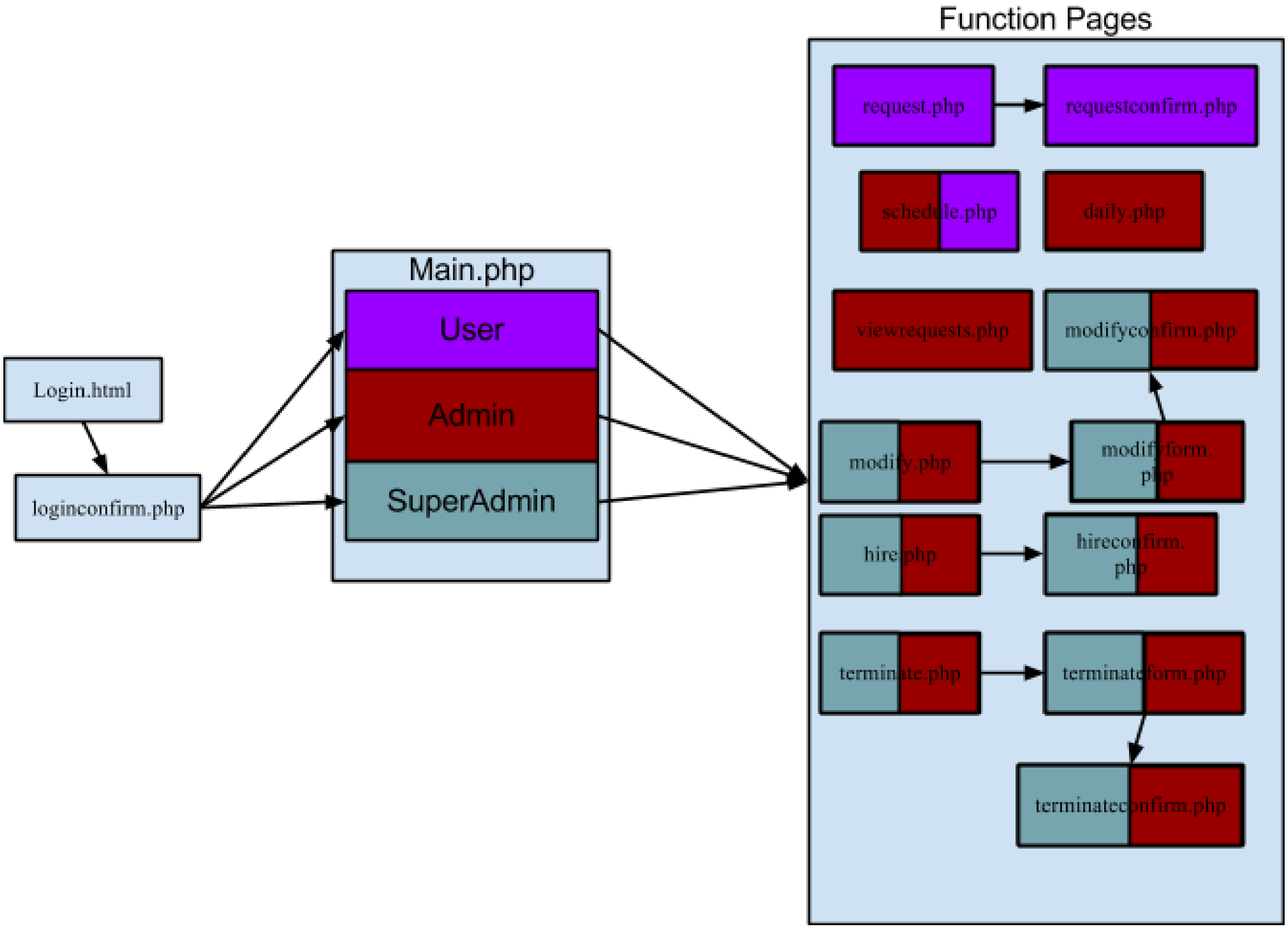
The final table in play is the "Requests" table which keeps track of employee requests for time off. This is the only table rank "u" users may add to and is viewed exclusively by managers. An employee's name and bagger number are tracked here in addition to the date they are requesting off and their reason for doing so. The "RequestNumber" field is used to index the requests so that one user could potentially have many requests in at the same time, in order to suit the employee needs.



```
mysql> desc Requests;
+---------------+---------+------+-----+------------+----------------+
| Field         | Type    | Null | Key | Default    | Extra          |
+---------------+---------+------+-----+------------+----------------+
| Baggernumber  | int(11) |      |     | 0          |                |
| Baggername    | text    | YES  |     | NULL       |                |
| Day           | date    |      |     | 0000-00-00 |                |
| Reason        | text    | YES  |     | NULL       |                |
| RequestNumber | int(11) |      | PRI | NULL       | auto_increment |
+---------------+---------+------+-----+------------+----------------+
```

Website Flow

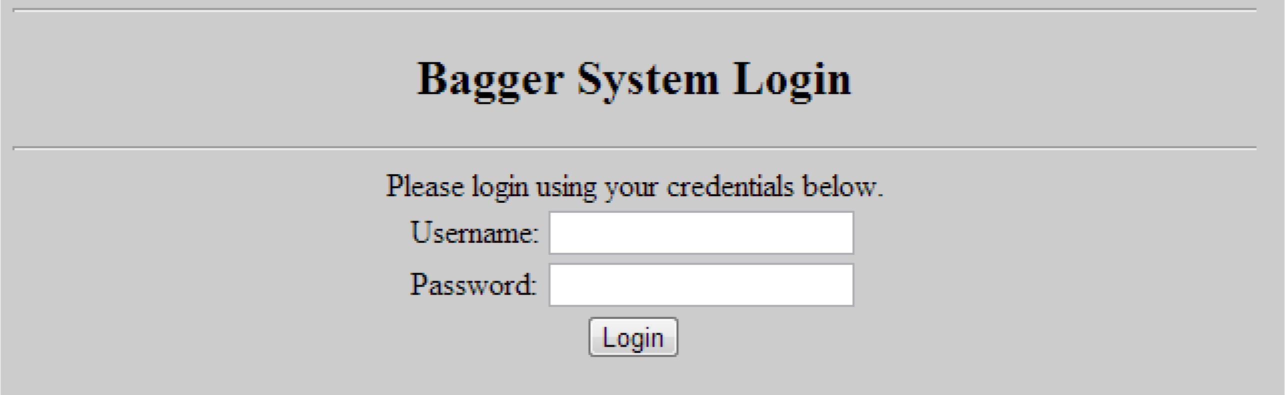
With these tables set up accordingly, the next task was to set about designing the framework of the website. The fundamental design decision of my approach was to create a "hub" page that changed depending on the login rank of the user and then provided links to all of their functions. This approach minimized the number of pages that needed to be created and resulted in an easy to use interface. The diagram below illustrates this concept. Any user may enter the login page, which then proceeds to "loginconfirm.php" a functionally hidden page that checks the user's entered information with the "BaggerUsers" table in order to ensure a proper login. It then uses the PHP header function in order to instantly place the user in "main.php," thus hiding the confirmation framework from the user.



In the diagram above, each of the pages are color-coded. Thus a page that has both a purple and red box in it can be accessed by both Users and Admins, however, the functionality is almost always altered in some fashion. The changes correspond to the needs of the rank of the currently logged in user. Additionally, all of the modification pages: “modifyconfirm.php,” “hireconfirm.php,” and “terminateconfirm.php” mimic the hidden nature of “loginconfirm.php.” These pages process the modification, hiring, and termination of employees within the tables, and then return the user to Main.php with a message notifying them of successful table modification.

Page Analysis

login.html

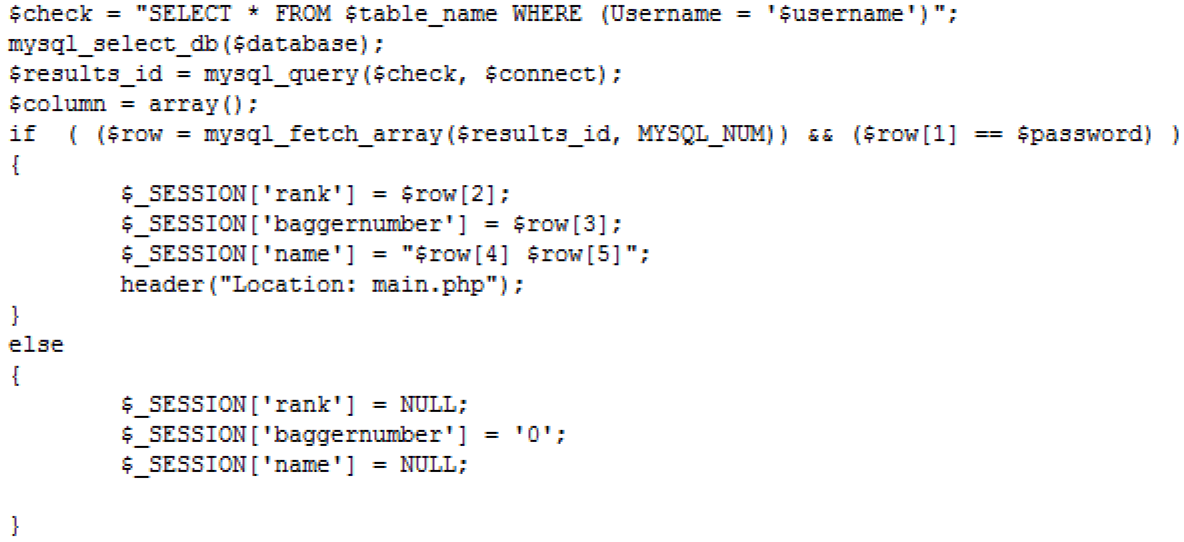


Bagger System Login

Please login using your credentials below.

This page uses a simple html prompt in order to get the user's login information. This information is then transferred via HTML form to "loginconfirm.php."

loginconfirm.php

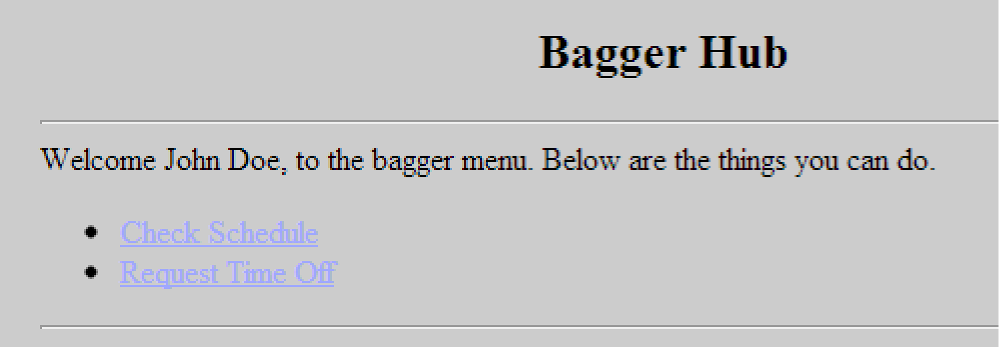


```
$check = "SELECT * FROM $table_name WHERE (Username = '$username')";
mysql_select_db($database);
$results_id = mysql_query($check, $connect);
$column = array();
if  ( ($row = mysql_fetch_array($results_id, MYSQL_NUM)) && ($row[1] == $password) )
{
        $_SESSION['rank'] = $row[2];
        $_SESSION['baggernumber'] = $row[3];
        $_SESSION['name'] = "$row[4] $row[5]";
        header("Location: main.php");
}
else
{
        $_SESSION['rank'] = NULL;
        $_SESSION['baggernumber'] = '0';
        $_SESSION['name'] = NULL;

}
```

This page checks the user's entered information with the MYSQL table "BaggerUsers." In order to do this, I imported the row information that matched the entered username as a numerically indexed array. Then I checked the password spot within the array to confirm that it matched the user's input. If a match was found, the header method was called, otherwise session variables are set to default values and the user is prompted to return to the login screen and try again. All pages after "loginconfirm.php" check the user's rank and determine if they qualify for access. An infiltrator will not have been assigned a rank, and thus cannot access the functionality of any pages other than "login.html" and "loginconfirm.html"

main.php

The page "main.php" serves as the "hub" page for users allowing them to view at a glance their accessible pages. As a result there are three variations to the page, displayed below.

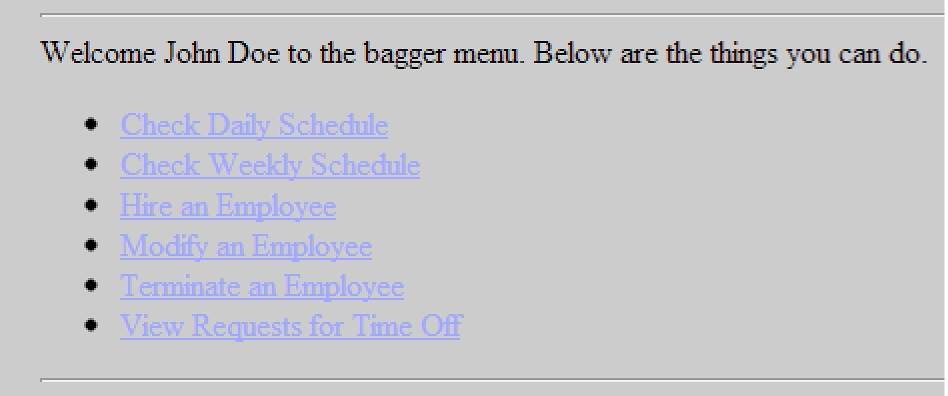


Bagger Hub

Welcome John Doe, to the bagger menu. Below are the things you can do.

- Check Schedule
- Request Time Off

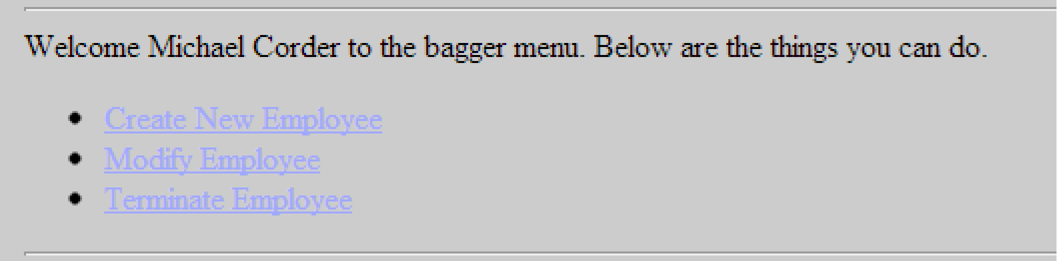
The rank "u" variant contains the most limited functionality as employees should not be allowed to modify schedules, hire, or terminate other employees. However, they are allowed to view the information regarding only their schedule and request time off.



Welcome John Doe to the bagger menu. Below are the things you can do.

- Check Daily Schedule
- Check Weekly Schedule
- Hire an Employee
- Modify an Employee
- Terminate an Employee
- View Requests for Time Off

The rank "a" variant, meanwhile sports the highest functionality of the three ranks. These users are able to view the weekly schedule, the daily schedule, perform any and all functions in regard to employee records, and view requests for time off.



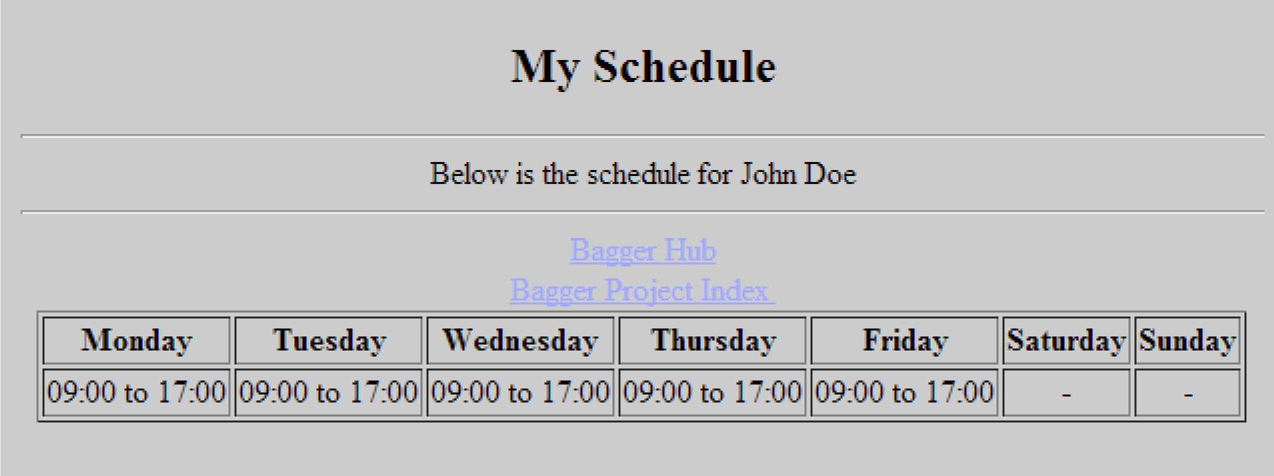
Welcome Michael Corder to the bagger menu. Below are the things you can do.

- Create New Employee
- Modify Employee
- Terminate Employee

The rank "s" variant contains all of the functions necessary for creating the first administrator, as well as modifying an employees due to error on the part of managers etc. It provides a handy way to alter the database from any device with a browser and internet access.

schedule.php

The page "schedule.php" also contains variations depending upon user ranking. It may be accessed by both rank "u" users and rank "a" users.



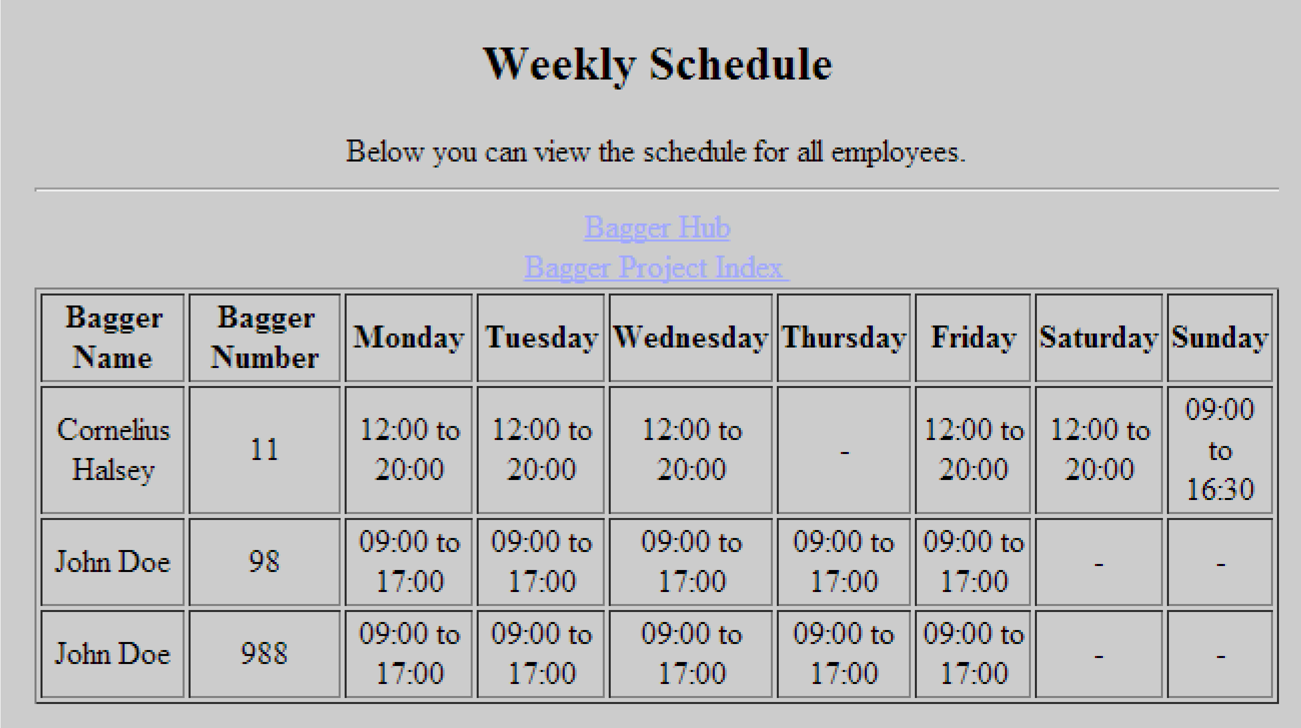
My Schedule

Below is the schedule for John Doe

Bagger Hub
Bagger Project Index

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-

The rank "u" variant uses MYSQL information in order to create a table that informs an employee of their workweek, formatted in military time. For days without hours a simple "-" is placed providing an easy read of the workweek.



Weekly Schedule

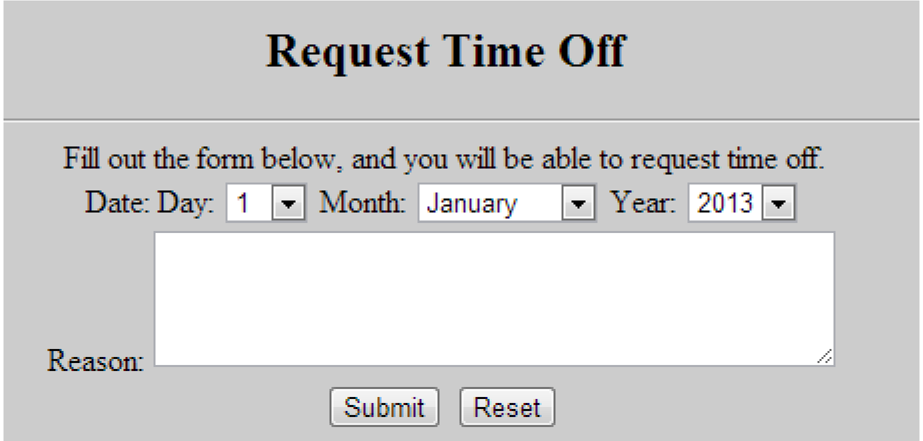
Below you can view the schedule for all employees.

Bagger Hub
Bagger Project Index

Bagger Name	Bagger Number	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Cornelius Halsey	11	12:00 to 20:00	12:00 to 20:00	12:00 to 20:00	-	12:00 to 20:00	12:00 to 20:00	09:00 to 16:30
John Doe	98	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-
John Doe	988	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-

The rank "a" variant allows the manager to view the weekly schedule for all employees, including other "a" rank users. This allows the manager to easily ascertain days where under scheduling or over scheduling are occurring, thanks to the location of entries on the table.

request.php



Request Time Off

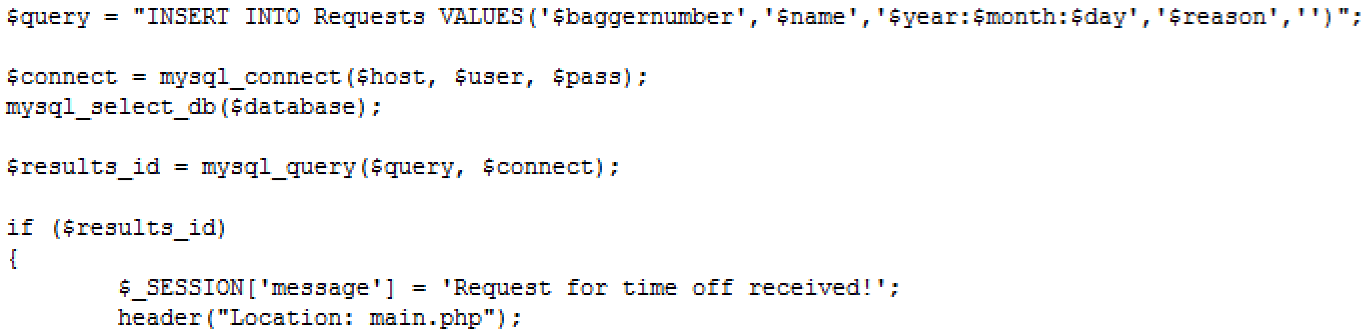
Fill out the form below, and you will be able to request time off.

Date: Day: 1 Month: January Year: 2013

Reason:

This page is accessed exclusively by rank “u” users and provides an HTML form for users to submit their requested day off. This information is then transmitted via HTML form to “requestconfirm.php.”

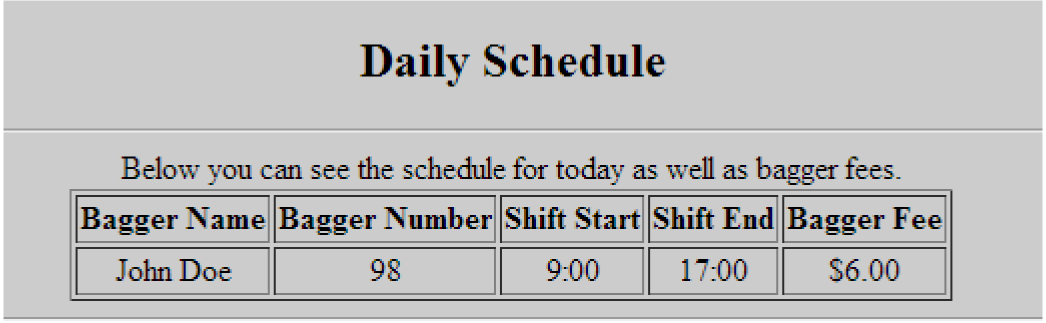
requestconfirm.php



```
$query = "INSERT INTO Requests VALUES('$baggernumber','$name','$year:$month:$day','$reason','')";

$connect = mysql_connect($host, $user, $pass);
mysql_select_db($database);

$results_id = mysql_query($query, $connect);

if ($results_id)
{
        $_SESSION['message'] = 'Request for time off received!';
        header("Location: main.php");
```

This page remains hidden to the user, but will store all of the information gathered in “request.php” into the MYSQL database. Additionally, a message will be created informing the user of the successful addition and it will be displayed at the top of “main.php” in blue font.

daily.php



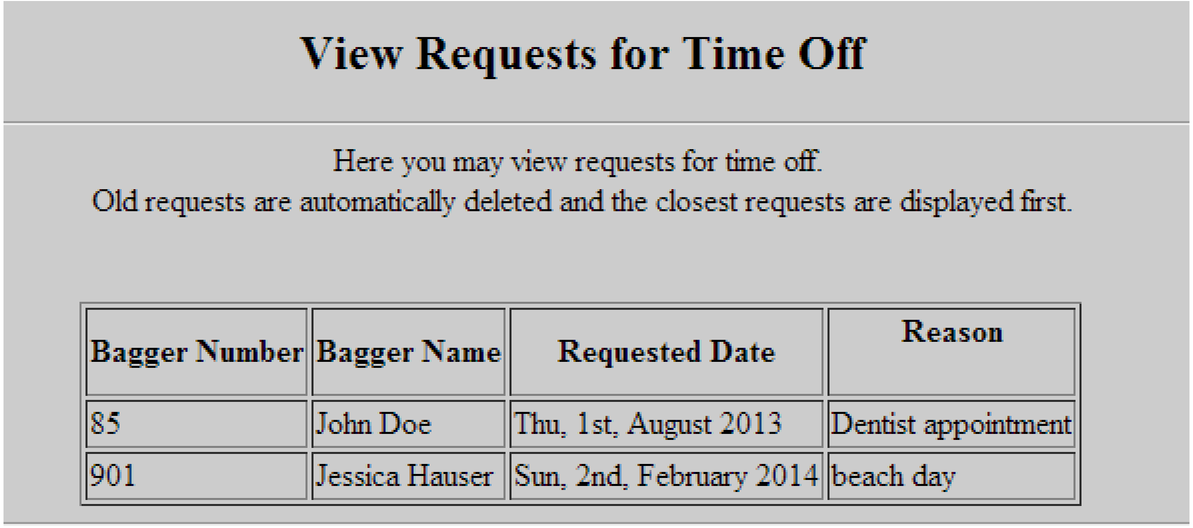
Daily Schedule

Below you can see the schedule for today as well as bagger fees.

Bagger Name	Bagger Number	Shift Start	Shift End	Bagger Fee
John Doe	98	9:00	17:00	$6.00

This page is accessed exclusively by rank "a" users. The PHP code within the page checks for the current day of the week through use of the "date()" and "time()" functions. Next, the code scans through the list of rank "u" users and checks to see which are working that day and then pulls their hours from the MYSQL database. Finally, the page evaluates the time difference between the starting and ending times and outputs a bagger fee equal to seventy-five cents per hour. The reason only rank "u" users are pulled into this table is because managers do not pay a bagger fee, they collect it.

viewrequests.php



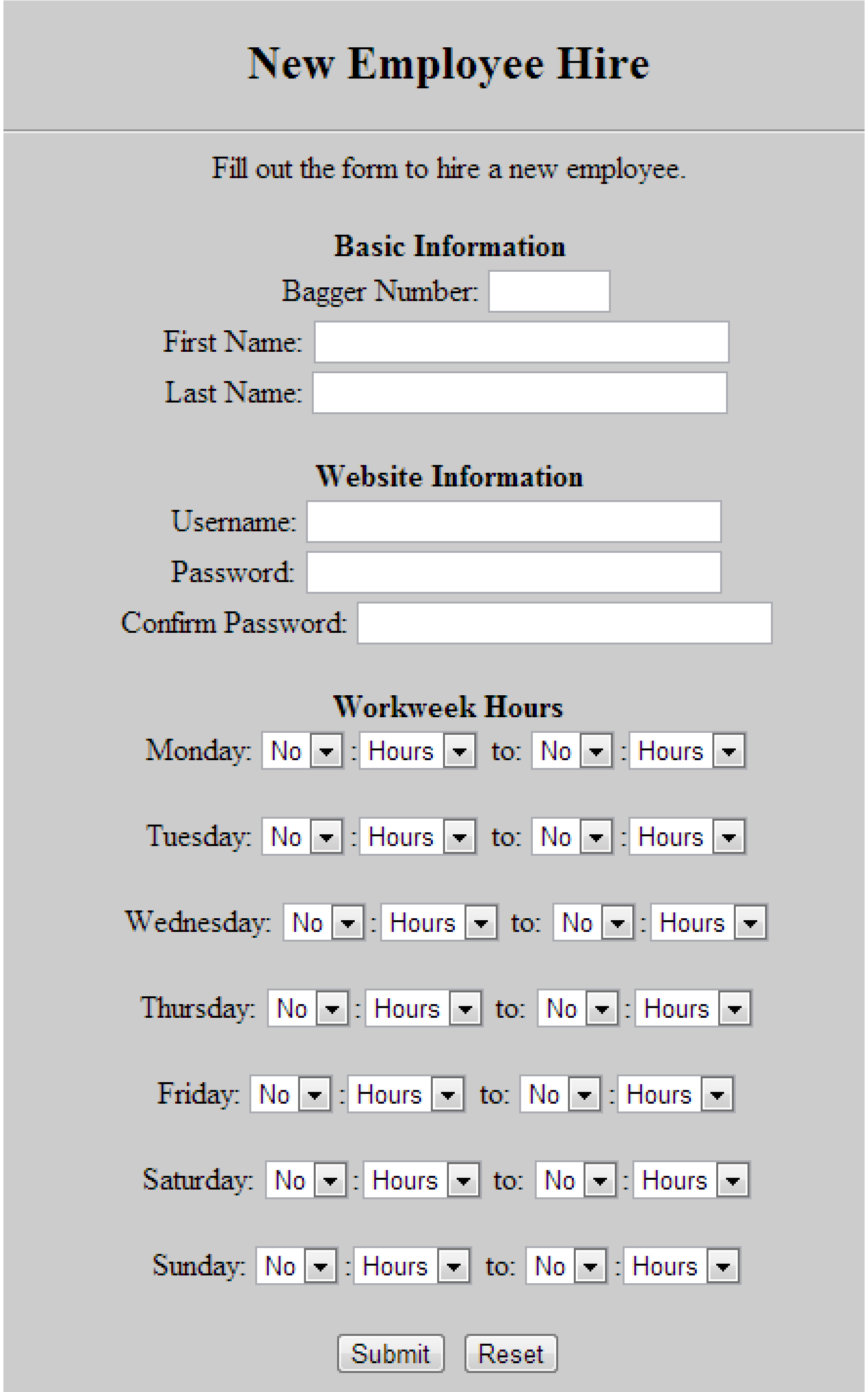
View Requests for Time Off

Here you may view requests for time off.
Old requests are automatically deleted and the closest requests are displayed first.

Bagger Number	Bagger Name	Requested Date	Reason
85	John Doe	Thu, 1st, August 2013	Dentist appointment
901	Jessica Hauser	Sun, 2nd, February 2014	beach day

Another exclusive page for rank "a" users, "viewrequests.php" allows admins to view the time off requests that were submitted in "requestconfirm.php." The entries are retrieved from the MYSQL Table "Requests" and are sorted by the requested date, with the closest date being placed at the top. In addition, this page automatically clears outstanding entries from the database when it is accessed.

hire.php



New Employee Hire

Fill out the form to hire a new employee.

Basic Information

Bagger Number:

First Name:

Last Name:

Website Information

Username:

Password:

Confirm Password:

Workweek Hours

Monday: No : Hours to: No : Hours

Tuesday: No : Hours to: No : Hours

Wednesday: No : Hours to: No : Hours

Thursday: No : Hours to: No : Hours

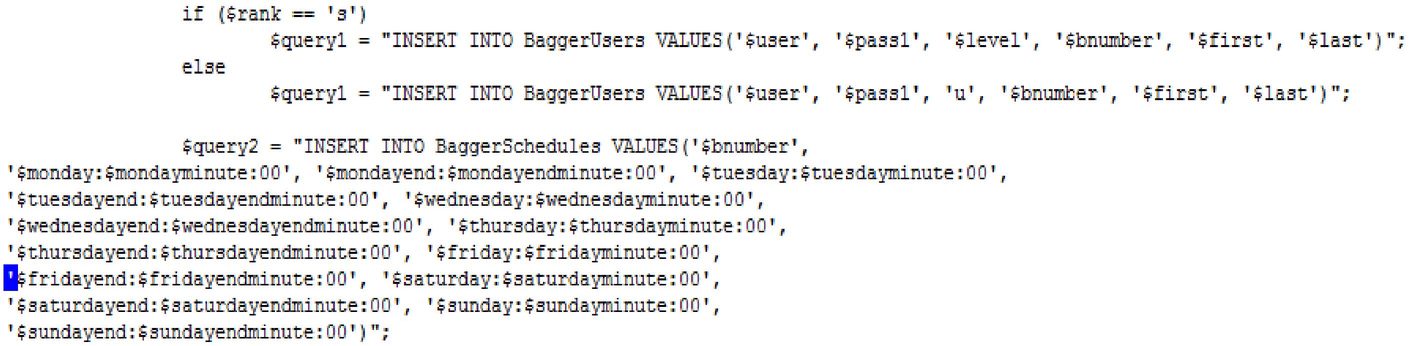
Friday: No : Hours to: No : Hours

Saturday: No : Hours to: No : Hours

Sunday: No : Hours to: No : Hours

This page is accessible by rank “s” and “a” users. It is an HTML form that allows for the creation of new employees. The only difference between superadmins and regular admins on this page is that superadmins can set the rank of employees directly through this page. Admins cannot, they must instead make use of the modify function in order to modify employees. This is a safety feature since admins will never “hire on” managers due to the fact that employees must have worked at the establishment for a certain number of years before becoming candidates for election to manager status.

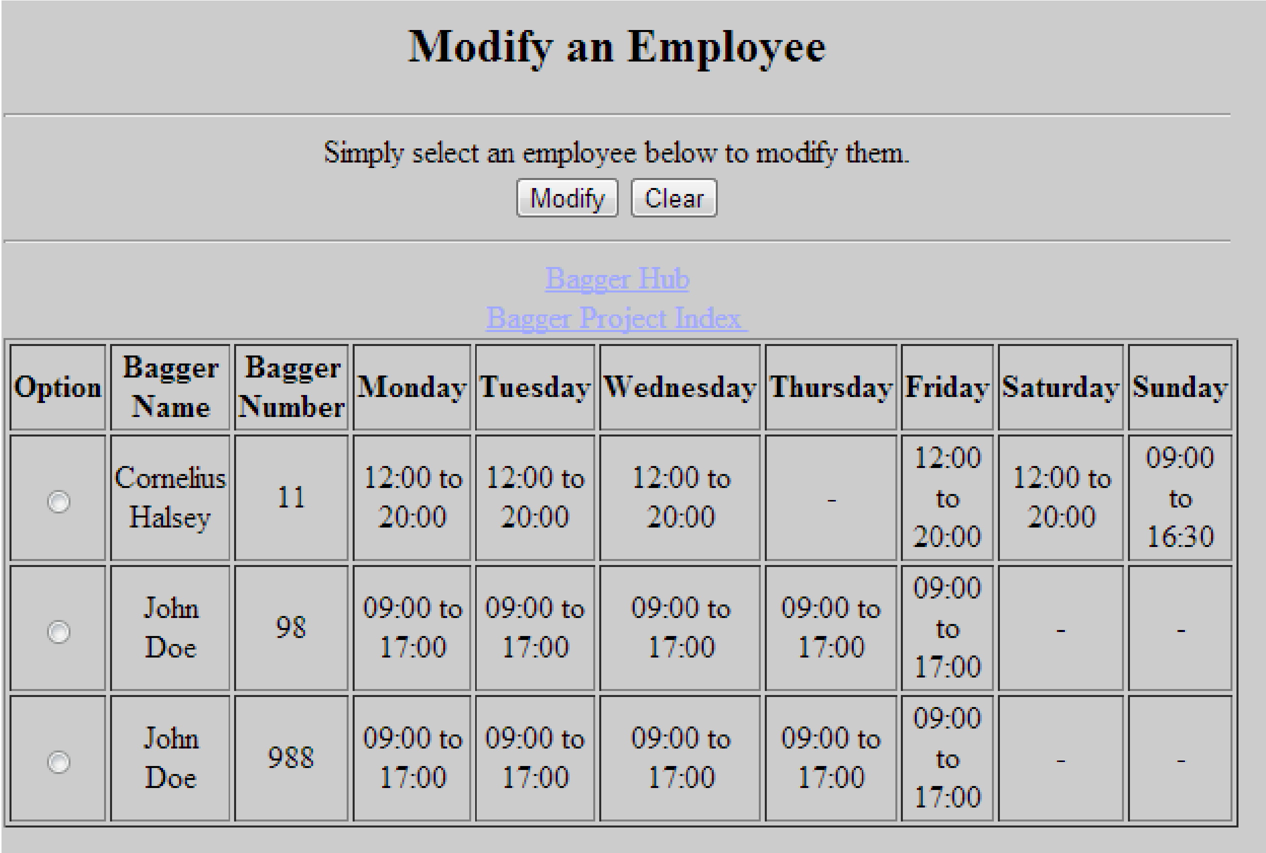
hireconfirm.php



```
                if ($rank == 's')
                        $query1 = "INSERT INTO BaggerUsers VALUES('$user', '$pass1', '$level', '$bnumber', '$first', '$last')";
                else
                        $query1 = "INSERT INTO BaggerUsers VALUES('$user', '$pass1', 'u', '$bnumber', '$first', '$last')";

                $query2 = "INSERT INTO BaggerSchedules VALUES('$bnumber',
'$monday:$mondayminute:00', '$mondayend:$mondayendminute:00', '$tuesday:$tuesdayminute:00',
'$tuesdayend:$tuesdayendminute:00', '$wednesday:$wednesdayminute:00',
'$wednesdayend:$wednesdayendminute:00', '$thursday:$thursdayminute:00',
'$thursdayend:$thursdayendminute:00', '$friday:$fridayminute:00',
'$fridayend:$fridayendminute:00', '$saturday:$saturdayminute:00',
'$saturdayend:$saturdayendminute:00', '$sunday:$sundayminute:00',
'$sundayend:$sundayendminute:00')";
```

This page creates the bagger employee in question by adding them to both the “BaggerUsers” and “BaggerSchedules” table. In the rank “a” variant, the new employee is automatically assigned a rank of “u”, while the superadmin version uses whatever rank was selected in “hire.php.” The times, are formatted to fit with the MYSQL time datatype, and the user is then returned to the hub page with a message confirming the hiring process was successful.

modify.php



Modify an Employee

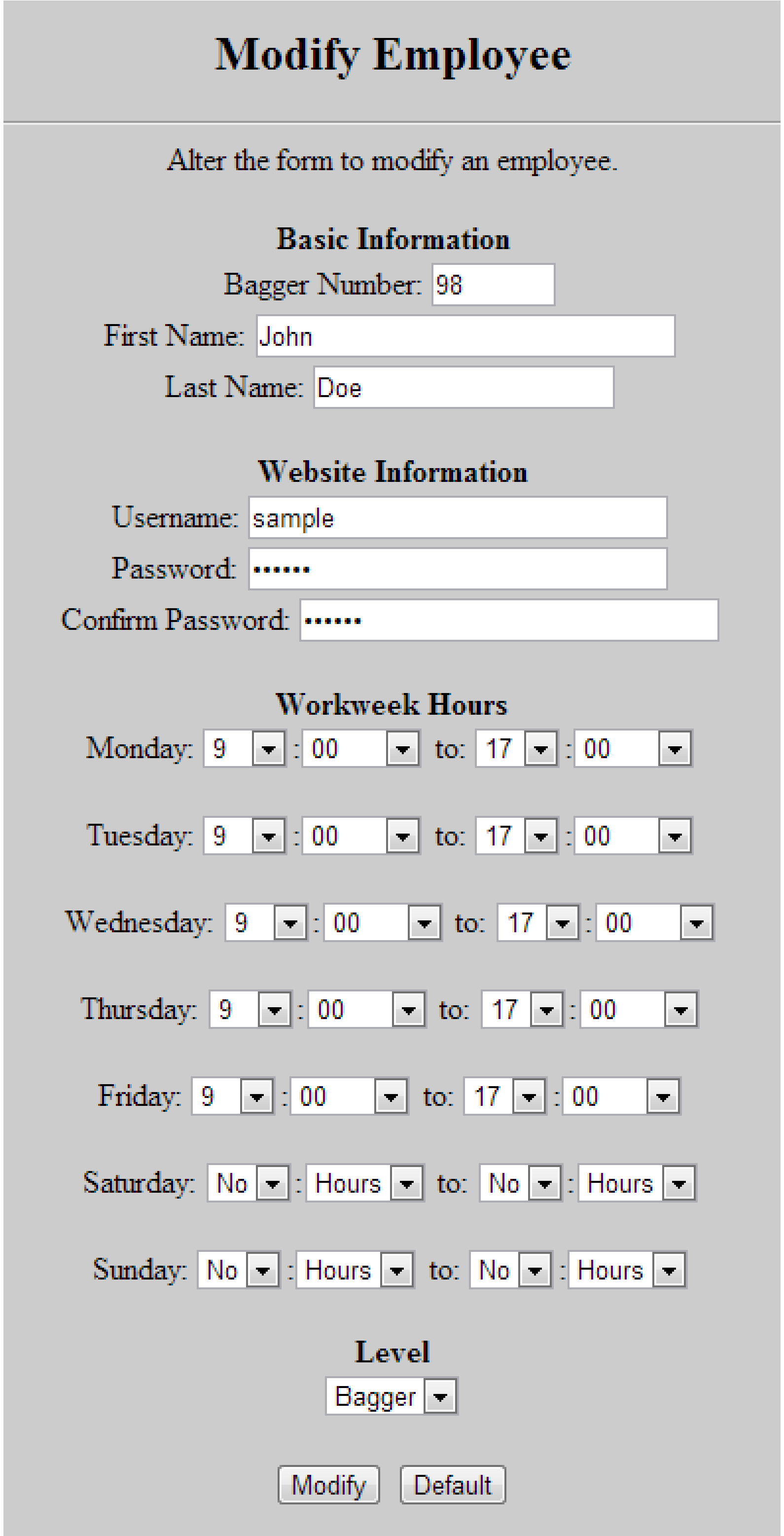
Simply select an employee below to modify them.

Bagger Hub
Bagger Project Index

Option	Bagger Name	Bagger Number	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
○	Cornelius Halsey	11	12:00 to 20:00	12:00 to 20:00	12:00 to 20:00	-	12:00 to 20:00	12:00 to 20:00	09:00 to 16:30
○	John Doe	98	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-
○	John Doe	988	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-

This page, accessible by both rank “s” and “a” users allows managers to select an employee in order to start the modification process. From this page, the selected employee’s bagger number is transferred into “modifyform.php” through use of HTML Form.

modifyform.php



Modify Employee

Alter the form to modify an employee.

Basic Information

Bagger Number: 98

First Name: John

Last Name: Doe

Website Information

Username: sample

Password: ••••••

Confirm Password: ••••••

Workweek Hours

Monday: 9 : 00 to: 17 : 00

Tuesday: 9 : 00 to: 17 : 00

Wednesday: 9 : 00 to: 17 : 00

Thursday: 9 : 00 to: 17 : 00

Friday: 9 : 00 to: 17 : 00

Saturday: No : Hours to: No : Hours

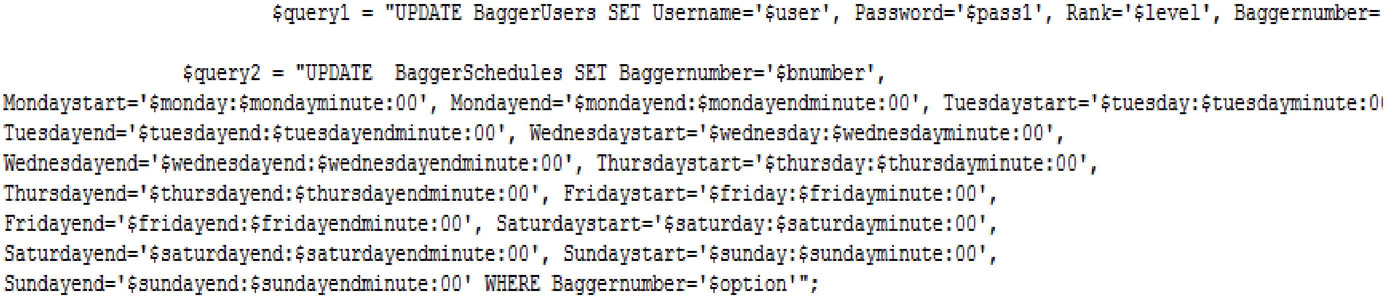
Sunday: No : Hours to: No : Hours

Level

Bagger

This page pulls the bagger number of the selected employee from “modify.php” and sets the default value of each of the fields to the employee’s information. Also of note is the ability to change the ranking of an employee through this page. It may be used to upgrade employees to managers and demote managers to employees. The only variant on this page is that a rank “s” user can modify an employee into a fellow rank “s” user.

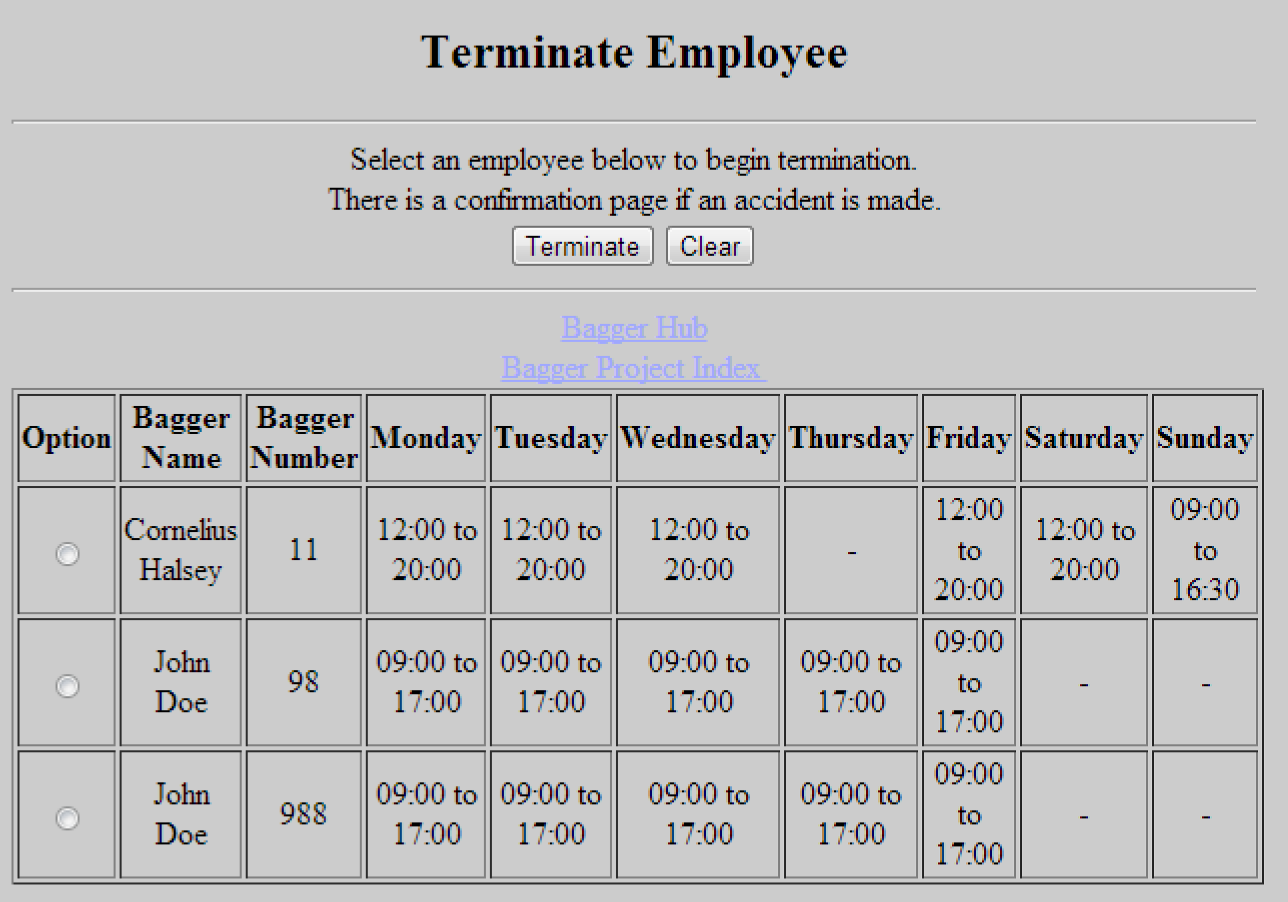
modifyconfirm.php



```
                            $query1 = "UPDATE BaggerUsers SET Username='$user', Password='$pass1', Rank='$level', Baggernumber=

                 $query2 = "UPDATE  BaggerSchedules SET Baggernumber='$bnumber',
Mondaystart='$monday:$mondayminute:00', Mondayend='$mondayend:$mondayendminute:00', Tuesdaystart='$tuesday:$tuesdayminute:0
Tuesdayend='$tuesdayend:$tuesdayendminute:00', Wednesdaystart='$wednesday:$wednesdayminute:00',
Wednesdayend='$wednesdayend:$wednesdayendminute:00', Thursdaystart='$thursday:$thursdayminute:00',
Thursdayend='$thursdayend:$thursdayendminute:00', Fridaystart='$friday:$fridayminute:00',
Fridayend='$fridayend:$fridayendminute:00', Saturdaystart='$saturday:$saturdayminute:00',
Saturdayend='$saturdayend:$saturdayendminute:00', Sundaystart='$sunday:$sundayminute:00',
Sundayend='$sundayend:$sundayendminute:00' WHERE Baggernumber='$option'";
```

This code snippet shows some of the PHP code behind “modifyconfirm.php” The page provides a very similar function to “hireconfirm.php” except the notable difference is that it updates “BaggerUsers” and “BaggerSchedules” rather than adding a new entry. As with the other “confirm” pages, it will automatically redirect the user to “main.php” after finishing the modification.

terminate.php



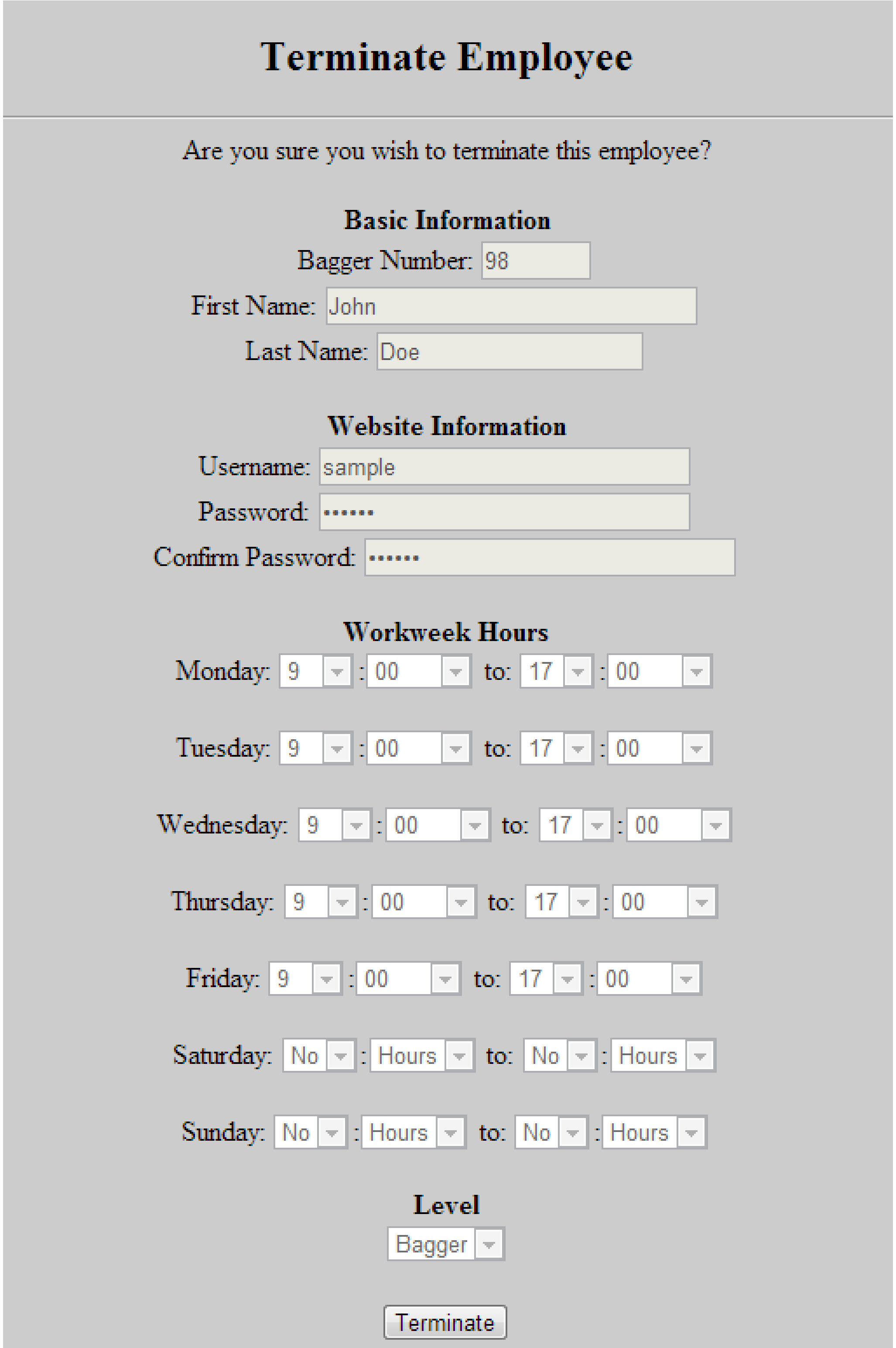
Terminate Employee

Select an employee below to begin termination.
There is a confirmation page if an accident is made.

Option	Bagger Name	Bagger Number	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
○	Cornelius Halsey	11	12:00 to 20:00	12:00 to 20:00	12:00 to 20:00	-	12:00 to 20:00	12:00 to 20:00	09:00 to 16:30
○	John Doe	98	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-
○	John Doe	988	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	09:00 to 17:00	-	-

This page works nearly identically to “modify.php” in that it shows the roster of workers and allows a selection to be made. Similarly it also logs the bagger number of the employee in question but instead transfers that information to “terminateform.php.”

terminateform.php



Terminate Employee

Are you sure you wish to terminate this employee?

Basic Information

Bagger Number: 98

First Name: John

Last Name: Doe

Website Information

Username: sample

Password: ••••••

Confirm Password: ••••••

Workweek Hours

Monday: 9 : 00 to: 17 : 00

Tuesday: 9 : 00 to: 17 : 00

Wednesday: 9 : 00 to: 17 : 00

Thursday: 9 : 00 to: 17 : 00

Friday: 9 : 00 to: 17 : 00

Saturday: No : Hours to: No : Hours

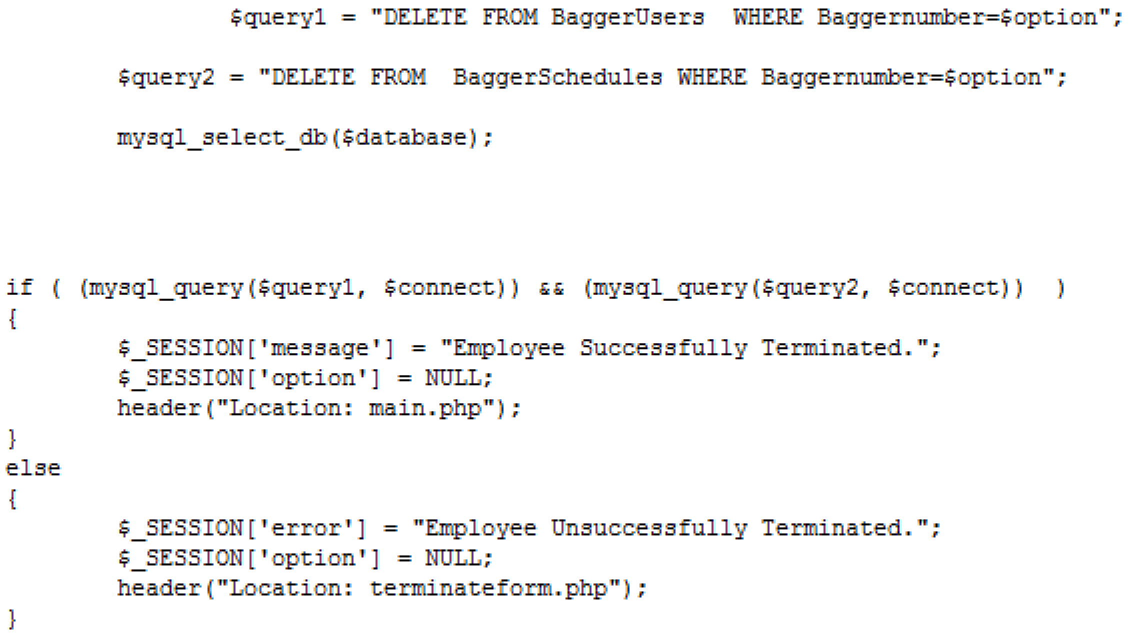
Sunday: No : Hours to: No : Hours

Level

Bagger

This page is very similar to “modifyform.php,” except that all of the fields are greyed out and unmodifiable. This is intended as a safety feature in case the incorrect employee is selected, the user is prompted to confirm their selection, preventing accidental firings.

terminateconfirm.php



```
                $query1 = "DELETE FROM BaggerUsers  WHERE Baggernumber=$option";

        $query2 = "DELETE FROM  BaggerSchedules WHERE Baggernumber=$option";

        mysql_select_db($database);



if ( (mysql_query($query1, $connect)) && (mysql_query($query2, $connect))  )
{
        $_SESSION['message'] = "Employee Successfully Terminated.";
        $_SESSION['option'] = NULL;
        header("Location: main.php");
}
else
{
        $_SESSION['error'] = "Employee Unsuccessfully Terminated.";
        $_SESSION['option'] = NULL;
        header("Location: terminateform.php");
}
```

This page works similarly to the previous confirm pages, but rather than pulling a large amount of information from “terminateform.php”, simply uses the bagger number from “terminate.php” that was passed through in order to clear the employee from both “BaggerUsers” and “BaggerSchedules.”

Conclusion

The Bagger Management Project is currently being implemented at Little Creek Commissary as a field test in order to evaluate the performance of this system in the scenario it was intended for. The feedback will be vital in continued efforts to both improve and expand the scope of this project, likely to become a free service offered to willing participant commissaries via private server. This project has indeed met all of the goals set forward at its initiation, and has been a phenomenal source of learning regarding the inner workings of HTML, PHP, and MYSQL.

References

Dev.mysql.com (2013) *MySQL :: MySQL 5.5 Reference Manual*. [online] Available at: http://dev.mysql.com/doc/refman/5.5/en/

Php.net (2013) *PHP: PHP Manual* - *Manual*. [online] Available at: http://php.net/manual/en/index.php