Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Final Report

Group 23 has created a website and database for the use of flight reservations. The website is designed to be used by both customers and employees. The four user types (customer, pilot, flight attendant, and administrator) are given unique user IDs, which will redirect them to a dashboard based on their user type. Customers and employees have separate logins, as the homepage is tailored to customer use. There are dashboards custom to the user types, which reflects what the user will be able to see. Our goal was to make a website that could be used seamlessly by either an employee or customer and be able to store information and pull data based on their needs.

http://cs3380.rnet.missouri.edu/group/CS3380GRP23/www/

Pilot credentials: ID: 123492 Password: 12345678

ID: 382901 Password: 12345678

Flight Attendant: ID: 111980 Password: 12345678

Administrator credentials:ID: 0 Password: Admin

Delegations

Milestone 1 was completed by everyone in the group collaboratively. At this point, we all had the same knowledge and practice with ERDs so it was the easiest to to work together on. The project planning was done by Kaitlin. MIlestone 2 was the same as milestone 1, that we could all easily work together on it. These two milestones had the most group involvement on one task.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Sam was our group's DBA. He worked on the database creation and management. He worked closely with the ERD and creating SQL statements to query the table. Sam also created the flight attendants page. He worked closely with Nick and LeAndre configuring the database to their needs. Sam also created the flight attendant dashboard, showing this user their assigned flights both previous and upcoming.

Nick developed the administrator dashboard as well as the employee login page. The administrator page is able to add, change and delete any records in the system. They are the only user that can add flights for pilots and attendants. The employee login is a separate page from the home page for employees to select. Nick's administrator page also has pages that redirect from the dashboard that include user management, flight viewer, logs, equipment entry, and flight update. He did all of the HTML, styling, and PHP for this user.

Tom created the customer dashboard which shows the customer their upcoming flights, previous flights, personal information.

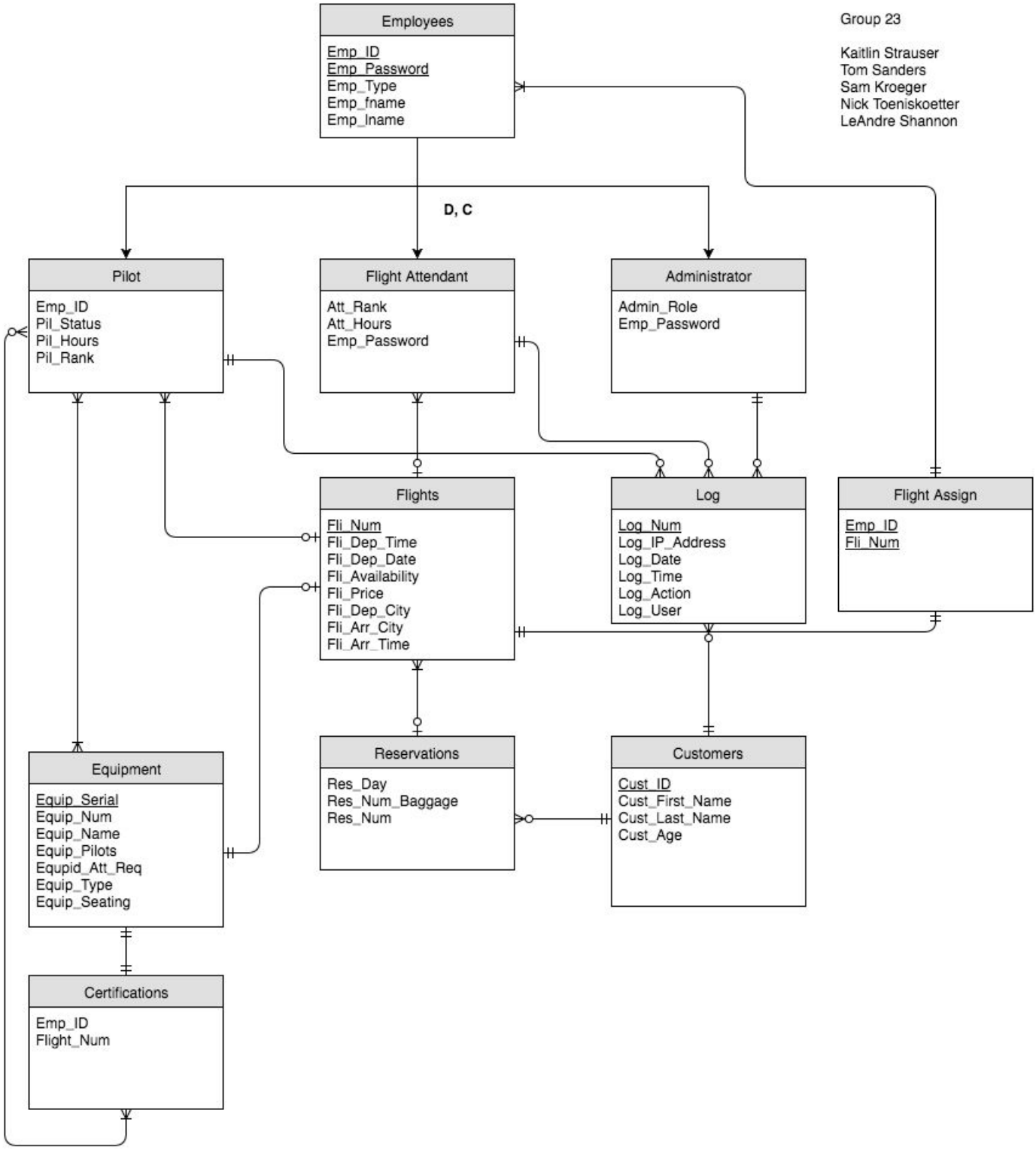
LeAndre made the main page that redirects to a page for the customers to view flights via a search bar or drop down. The flights page views all flights and has the ability to search by departure city, destination city, day/time, and price. He also made the corresponding flight reservation page that has the customer input, price calculation, checks if they are an existing customer and lets them confirm their reservation, and the customer login to allow returning customers to view their reserved flights. He also worked with the pilot and flight attendant dashboards, allowing pilots and attendants to view/edit personal info, view flight logs, and add hours to the log.

Kaitlin created the employee dashboards for pilot dashboard before the PHP was implemented by LeAndre. There was also an employee dashboard that she created at one point

but was later decided that it was not necessary. The early stages of all of the documentation was done by her and group organization. The final report is primarily being compiled by her except for the the specific areas that the group members worked on, down below.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

ERD



Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Customer User

The customer user is able to search the directory of flights based on departure city, destination city, day/time, and price. By filtering the options, the flights that meet those criteria will appear. After creating their first reservation, they are able to view all of their reservations made and edit their personal info, such as their names and age.

To accomplish this, the flights page was made to show only the search fields relevant to the type of search the user requests. After selecting a flight, the user asked how many bags they'd like to bring in a number restricted input field to prevent sql injections and ensure the entry is valid. The customer is then shown to a confirmation screen where they can confirm or cancel the reservation.If confirmed, the page will generate a random alphanumeric reservation code. Again, here the customer is met with a number restricted input field that will allow them to tie the reservation to their account if they are already a customer, or , if left blank, a new customer will be created with a randomly generated numeric id. If the user decided to make another reservation after logging in first, the customer id field will be populated using sessions and will be uneditable for the user. The customer profile page, made very similar to the employee profile pages, includes a read-only customer id field to prevent tampering, two text fields for the customer to change their name, and a number restricted input field for them to add their age.

From making these pages I learned about input restricted form fields to help prevent sql injections and ensure only valid data is entered. I also learned about bootstrap and jquery to only show search fields based on what the user wants to search by. In making everything dependant on the user, I learned how to use sessions with logging and passing variables along using the superglobal.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Pilot and Flight Attendants

The pilots and flight attendants have very similar pages and are what they can see on their dashboards doesn't vary as dramatically compared to the administrator user or the customer. They have the ability to see their previous and upcoming flights, be able to view/edit personal information, and see their status/rank. Pilots can also add equipment that they are certified to fly.

To accomplish all of this, the employee id was saved to the sessions superglobal, allowing every page that gathered information on the user from the database to do so depending on the id it gathered from the login page. The initial page of each dashboard was coded to display the user's relevant information such as their certifications, and flights they are assigned to. In order for the employee to edit the information relevant to them, links were provided within the navigation bar of the page for them to both get a better look at their personal info. In the certifications page, a pilot has the ability to view and edit all of their current certifications obtained. To ensure the pilot only adds certifications that are relevant to TomAir, they must select equipment that TomAir already owns. On the profile page, both pilots and attendants are able to view and edit their personal information based on their employee id. To ensure the user can only enter valid data, only three of the input fields, such as the password and name fields, allow for text entry. Everything else is locked to specific input using radio buttons, drop downs, and number restricted fields. The employee id is locked in a readonly field to prevent the user from losing the data relevant to them.

From these pages, I learned about html form inputs that can restrict user input to help prevent sql injections and ensure valid entries and how to effectively use sessions to gather and

keep relevant user data. In creating and styling a few of the pages I learned more about bootstrap and jquery to make the styling and displaying certain page information easier.

Administrator

In the administrator role, they are the more complex user. This user has the ability to edit all of the information from pilots, flight attendants, customer information to flight reservations. They have the permissions to edit it all. This is also the user that creates flights for customers to reserve and flight attendants and pilots to be assigned to. To organize it for the user, there are different tabs that redirect them to a specific task. The main dashboard shows flights that recently departed, there is also a user management tab, flight viewer, log viewer, update equipment, and update flights tabs.

With the large amount of task administrators can do, it was easiest to split each ability into its on page/tab. This allowed for a clean layout in the navigation bar as well as a clearly set plan of attack for each page for development. It started off with a basic layout that was based off of a previous project Nick made for his web development class in a previous semester. The page takes advantage of Bootstrap Version 3 and a little bit of jquery, css, and javascript for a repeatable template to be made for each page. He mocked up an idea during the first few meetings and with some input from the group we had a pretty good idea what the page would look like when finished.

Each page needs to check if the current logged in user was an admin before accessing any of these high level controls. To conquer that goal Nick implemented sessions on from the Login Page. From that a few lines of code was written that could then check the type of the user logged in and direct them to where they needed to be if in the wrong location. In the instance of the Admin Pages the user is type 0. If the session is not set or they do not meet the criteria for

the admin employee type(0) they are routed back to the login page via a php header. On the login page they should be redirected to their proper page if logged in as another employee type or forced to login if the session is not set.

E.g.)

```
1. <?php
2. session_start();
3.
4. if (isset($_SESSION['Emp_Type'])){
5.   if (($_SESSION['Emp_Type']) != 0){
        a. header("Location:
           http://cs3380.rnet.missouri.edu/group/CS3380GRP23/www/Login/Login.php");
6.   }
7. }
8. else {
9.   header("Location:
   http://cs3380.rnet.missouri.edu/group/CS3380GRP23/www/Login/Login.php");
10. }
```

The main page of the Admin's dash had some big ideas on it at first, but were sadly dropped. We had hoped to implement graphs and real time flight information, but after some development we realized time constraints had gotten the best of us and therefore cut those ideas from the final product. Though we believe we had the knowledge to overcome those challenges, we came victim to time.

The user management page was another large challenge during this project. It truly became a dance between members working on the database and the creation of the page as we tossed ideas back and forth and made changes as we overcome new needs. A good example of this was when we found out that when an Employee was added they were not getting properly put into their specific Employee type tables. This meant we would have to write code to add them to the Employee table and then send that information and some defaults to

the Emp_Type specific table. We found many issues during this task foreign key constraints, improper keys being set, and missing columns, but with hard work and good communication they were slowly sanded out. Another challenge was getting the page to update upon users being added as well as combining the add Employee and update and delete controls to the list below it. During work on the update it was discovered how in depth it would need to be to change a user's ID or Type. Realizing the Employees ID was a primary key and main identifying factor lead to the decision to lock it from change. The type however lead to some creative coding. Nick found out you would need to locate the user in the employee table to display and change, but also find their specific row in their Emp_Type specific table. You would also need to delete the Type table data for that Employee before updating their Employee information or you would get foreign key constraint errors. To do this we had switch statements as well as a sql statement dedicated to finding the Employees type. Once found the first switch was started and used prepare statements and delete specific sql queries to delete the row where the employee was located in their table. After their old row is deleted they are updated in the Employee table. Final the second switch is triggered and using specific update queries for each type the user's new type table is updated with their new information and defaults are set. It was a long process figuring he update out, but in the end was a beautiful and powerful solution.

Deletion of an employee was very similar. Prepare statements and switches were used once again and the user is located in both of their tables and removed. This task was easy once the update task was figured out.

Adding an Employee shows a strong resemblance to Nick's Homework 2 from the database class. It uses a lot of code from his registration page and has a few checks to help make sure users being created have secure and proper input. An example of this is the

password must be a minimum of 8 characters. It also checks to see if the user already exist as well as displays any other errors that the code picks up before the database gets ahold of it. In the case that it does get to the database errors are also displayed as well. A variable called $fmsg was used for this. We also didn't want the page cluttered with unneeded information for each user type so Nick implemented some javascript to fade in and remove inputs for each type if needed. This was achieved using .click and document.ready functions.

Update flights and update equipment were much easier to conquer in comparison to user management. This was because they didn't require searching multiple tables for foreign keys. They use most of the same code and complete their task using the same methodology.

The Admin's dashboard was one of the more in depth pages and required many pages and code to work together to work. This helped us find many bugs in not just the web design aspect of our code, but also pinpoint some logic problems we had in the creation of our database. A lot of php logic including header redirects, switch statements, and sqli friendly commands like mysqli_fetch_row were used in it's development and really taught those who developed it a lot in creative coding. In the end we had a lot of ambitious ideas that were cut, but functionality wise the page came together very well.

Logins

The login pages were a major task as we decided to use sessions to keep the integrity of the pages. You cannot click around the website as a customer or be able to edit flights as a pilot due to many checks integrated in each page. The choice to use sessions over cookies was at first a random choice. Mostly due to Nick feeling more comfortable coding with them. We later found out that sessions are stored server side in comparison to cookies. This allows for a safer

website in the end due to the popular likelihood of cookie jacking allowing a user to pose as another active user. With sessions though this is unlikely due to the user not visibly having any data about their session stored in their browser.

The page's workings resemble a close relation to Nick's Homework 2 from the database class. It once again takes advantage of many Bootstrap resources as well as some clearly placed ajax/jquery commands to give pop-up messages during errors and incorrect input. That being said a few edits were decided on during out meetings. We decided registration was no longer needed in the code since the admin(Boss of the company) would be the one adding and removing Employees. This creates a safer environment so random users can't change database information.

The page has an initial check to see if a session currently exist. If one does so exist it checks the Employee's type and redirects them to their proper page. No reason logging in if you are already logged in. The page also does a few other checks before logging in as well. It makes sure the information is valid and matches the database's values as well as displays errors if any occur before the data can reach the databases error checks.

Login uses the same bootstrap and btn-primary class attributes that many other other pages take advantage of. Login was an easier page thanks to homework 2 having already having most of the required code. Login set's sessions using sql statements based on the Emp_Type and the Emp_Name via stmt executions. These sessions are used later for name display as well as checks for employee to make sure they have privileges to access specific pages.

Logout was a side creation of the login page. It deletes the sessions and releases the user to re access the login page. It does this via session_destroy and a header redirect.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Normalizing the database

The easiest way to go about normalization logically was to create the tables ahead of time with what we thought they would require and then step back and take a look at the logic behind it. After some time of analyzing we would make changes here and there which made sense to make sure no data conflicts could occur.

We had to think carefully about which tables to make first because other tables were dependant on the previous due to foreign key references. Also the database had to match up and be able to perform with the web page php seamlessly. The database is basically the foundation of the entire website.

We learned a lot of things about sql and databases during this assignment. We became more familiar with the Sql syntax and how to insert and alter tables without just wiping the whole table unless necessary. We also learned how to make tables correspond with each other in more efficient ways.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Instruction Manual

Customer User:

Welcome to Tom Air!

1. From our home page, you can click to see available flights or click customer login.
    a. Clicking to see all available flights will redirect you to a page for you to filter by flights
        i. You can filter by City, Day/Time, and Price
            1. To search by city, enter the city you would like to depart from in Departure City and the city you would like to visit in Arrival City. Click search when you have finished.
            2. To search by Day/Time, select the date and time you would like to take flight. Click search when you have finished.
            3. To search by Price, enter the price desired. Click search when you have finished.
        ii. Once you click a flight to reserve based on your filtering, you can click reserve in the flight field located at the end of the flight information.
        iii. Next, enter how many bags you would like to bringing. There is a $20 fee per bag. Your flight information should be located at the top of the page. Click submit when finished.
        iv. The next page is the Reservation Details page. You will be able to confirm your reservation or cancel to take you back to the home page. If you are a returning customer, you may enter your Customer ID. Are you a new

customer? No worries! Click confirm and you will be issued a Customer ID.

v. You are now on your Customer Dashboard. In the top left hand corner is your Customer ID. On this page you can see the flight you reserved. In the ribbon above, there are options to check your reservations, edit profile, and logout.

1. On reservations, it displays your flight reservations with reservation number, flight number, departure time, arrival time, departure date, departure city, arrival city, flight serial number, ticket price, number of bags, and the price paid including tax. This page will display all reservations made. You also may click Book Another Flight to be redirected to view other flights.
2. On the Edit Profile page, your customer ID is displayed and you will be able to add/edit your first name, last name, and age. Click Save once once finished.
3. Click Logout when you are done viewing/making changes.

b. Clicking Customer Login is for customers who already have a login

i. Enter in your Customer ID

ii. You are redirected to your Customer Dashboard. You can see the flight you reserved. In the ribbon above, there are options to check your reservations, edit profile, and logout.

1. On reservations, it displays your flight reservations with reservation number, flight number, departure time, arrival time,

departure date, departure city, arrival city, flight serial number, ticket price, number of bags, and the price paid including tax. This page will display all reservations made. You also may click Book Another Flight to be redirected to view other flights.

2. On the Edit Profile page, your customer ID is displayed and you will be able to add/edit your first name, last name, and age. Click Save once once finished.
3. Click Logout when you are done viewing/making changes.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Employee User:

1. From the main page, click “Employee Login” in the the top right corner.
2. On this page you will enter in your credentials.
3. You will be redirected to a page based on your employee type
    a. As a pilot, you will be brought to your dashboard where you can view your current certifications and flights. In the ribbon, you will see options that will redirect you to Certifications, Flight Viewer, Edit Profile, and Logout.
        i. Clicking on Certifications will allow you to add equipment certifications.
        ii. Clicking on Flight Viewer will allow you to see all of the flights you have been assigned to.
        iii. Clicking Edit Profile is where you change/view your password, name, status, hours, and rank. Remember to save all changes.
        iv. When you are finished viewing, click Logout to return to the login page.
    b. As a flight attendant, you are brought to your home dashboard where you can view your upcoming flights.
        i. Clicking Flight Viewer will allow you to see all of the flights you have been assigned to.
        ii. Clicking Edit Profile is how you can change/view your name, hours logged, and rank. Remember to save any changes.
        iii. When you are done viewing, click Logout to return to the main page.
    c. As an administrator, your home dashboard shows the recently departed flights. You can go through User Management, Flight Viewer, Log Viewer, Update Equipment, and Update Flights.

i. Clicking User Management will allow you to edit user information.
    1. You are able to add an employee by entering in ID, First and Last name, Password, and employee type. Click add to add them to the database.
    2. You are able to Update users in the database as well.
        a. To update, click the update button. You will be redirected to a page to edit their Password, Employee Type, First, and Last name.
        b. Click Update after you have made your desired changes to make the changes in the database
    3. You are also able to Delete users.
        a. Click Delete, to remove a user from the database.
        b. You will directed to a page that will display a message if the user was deleted.
        c. PROCEED WITH CAUTION! ONCE A USER HAS BEEN DELETED, THEIR INFORMATION IS GONE
ii. Clicking Flight Viewer will display flights
    1. You are able to Update the flight information.
        a. See Update Flights down below
iii. Clicking Log Viewer will show all of the activity on the site.
    1. Profile edits
    2. Users logging in/out.
    3. Customer reservations

4. Pilot certification changes

iv. Clicking Update Equipment will show all of the planes that pilots are certified to fly

1. Clicking Update will allow you to make changes to the equipment.
    a. When are done making your revisions, click Add Equipment
2. Clicking Delete will delete the plane from the database.
    a. PROCEED WITH CAUTION! ONCE A PLANE HAS BEEN DELETED, THE INFORMATION IS GONE

v. Clicking Update Flights will allow you to Update or Delete scheduled flights

1. You can also access this page through Flight Viewer by clicking Update.
2. On this page, you can create or update a flight
    a. Enter in the desired inputs into the specified fields.Once you are done, click Add Flight.
3. You can also Delete a flight
    a. Clicking Delete in the flight field will delete the flight from the database.
    b. PROCEED WITH CAUTION! ONCE A FLIGHT HAS BEEN DELETED, THE INFORMATION IS GONE

vi. When you are done viewing or making changes, click Logout to return you to the login page.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

Issues and frustrations

Starting with the database, making changes in tables that other tables were dependant on was frustrating for Sam. Several times tables need to be deleted to make changes on primary and foreign keys.

The PHP was inevitably the hardest part of this assignment. Two group members have taken CS 2830, two members are in it currently, and one member has not. This project has challenged everyone in this area and has required a lot of Googling. Nick and LeAndre were our PHP coders. The foundation of the this website working is with the PHP, which the three other members didn't feel confident with. It would have been better for them if there was another person to take on the PHP. Also, towards the end of the deadline there is more stress on LeAndre and Nick to have their code working so the website can be finished.

As mentioned previously, we are all at different levels of experience so it was difficult to see the bigger picture and assign roles and also have an equal part in the work. As the deadline came closer, we were more aware of who was doing the most amount of work. Our deadlines for things got pushed back and we were not able to accomplish what we wanted. This was in part because there were new developments as we made more pages. There were pages that weren't needed and pages that weren't yet made. What was hard about this was knowing that we couldn't all carry our share in the project.

Time management is always the biggest challenge. We met weekly all throughout the assignment, but there wasn't progress until the week or two before Milestone 3 was due. In the later weeks, it was hard for all of us to juggle the labs and homeworks on top of working on the website. The labs and homeworks did help us out tremendously with the PHP implementation that we are using, but trying to meet all deadlines was not happening.

Group 23
Kaitlin Strauser
LeAndre Shannon
Sam Kroeger
Tom Sanders
Nick Toeniskoetter

If we could do this again, we would start earlier and try to set more roles earlier on in PHP. If more roles were designating to the PHP initially there would be less strain and more equal work being done. We had a very good group dynamic and our group meetings were full of discussion about the website. There was a lot of involvement but lacking in implementation.

Summary

Our website was designed for customer and employee users on an airline website. It is fully functioning to reserve flights for customers, managing flights and user for administrators, and look at scheduled flights for flight attendants and pilots. It was a challenging project that required a lot of group cooperation. We made a website that served a purpose and that has taught us about databases and their functionality in web development. This project has also taught us how challenging group work can be. In the real world, you have to work with people who have different strengths and group members may not be all have the same base knowledge. We had a good group dynamic that we are thankful for. Our website is done so we have completed them main goal of meeting all of the criteria and we have additionally learned real world skills.