**Ranking Document - Team 8**

Member 1 : Mohammed Al-hazza

My teammates and I agrees I did 25% of the overall project. My specific tasks include

Task 1: I created the ERD after discussing the system with teammate Seth.

Task 2: I created relational model and made mods after peer review

Task 3: I coded the following

1. Login for 3 types of users with different pages forms and functionalities (Admin,Doctor,Patient)

2. Image can be inserted and updated to database.

3. Admin can do the following:

* Add-List-Edit-Delete for (Employees, Patients, Rooms,Job Titles)
* Admin can change job title for doctor to be Admin.
* Admin can see the occupied rooms and the status of rooms.
* In editing, admin can assign doctor name and (room building and number) by retrieving information from database. Doctor name from “Doctor View” and (building number and room number) from room table.

4. Doctor can do the following:

* See all patients related to the login doctor.
* Edit some information about patient related to doctor.
* Doctor can (Add-List-Edit-Delete) report of a patient.
* Doctor can (Add-List-Edit-Delete) medicine of selected report for a patient.

5. Patient can do the following:

* Patient can see his/her profile and edit information.
* Patient can see his/her report and medicine of that report.

6. Create Views:

* Doctor view: Creating this view to only have doctor information (suppose that we have many job titles ) when admin wants to assign a patient to a doctor.
* Occupiedrooms view: Creating this view to list all rooms that is not empty and each rooms is occupied will be shown with doctor name and patient name by using three tables: employee, patient, room.
* Roomstatus view: Creating this view to list all rooms in DB and to show name of doctor with a room that is occupied or empty under doctor name with rooms that are not occupied by using “Left Join” of two tables employee and room.

7. Cascading:

* Use delete cascade, update cascade and set null on the following tables ( Patient, Report, Medicine, Room).
* Report or Medicine of any patient can be deleted without any error of foreign key.
* When patient will be deleted, all his/her reports and medicines will be deleted but his/her room will not be deleted and in room table under PatientID, it will be “NULL”

8. Create new table called “jobtitle” to retrieve data of job title to PHP page upon add new employee or edit doctor job title to make easier for admin to new job title instead of just put select form in HTML to show job title.

9. Fill database with information for more than 20 records of Patient, Employee, Room, Report, Medicine tables.

10. Upload HTML and CSS design for (index, Admin, Admin List Employees) pages in git to let other members finish final design of our project.

11. Admin can not be deleted because the button for “Delete” will be hidden unless they are changed to have another job title.

12. Before deleting, a notification will be shown to confirm deleting in order to avoid any delete mistake.

13. If Email or Password is wrong, a message will appear and reload to index page again.

14. A new patient can register from the index page.

Member 2: Seth Fields

My teammates and I agrees I did 25 % of the overall project. My specific tasks include

* Task 1 : I created the ERD after discussing the system with the team.
* Task 2: I created relational model and made mods after peer review
* Task 3: I coded the following:

1. Made the Doctor's view for when Doctor logged in.
2. Made search function for the Doctor view so that they can search through the patients they have.
   1. Utilized AJAX for this function
   2. doctor.php, doctorTable.php
   3. Used php to query the database to grab information for the function(Table)
3. Made settings pages for the doctor page:
   1. Two page was for changing their passwords, settings.php and changePassword.php
      1. Used php to update info in the database
   2. One page for changing user information: EditDoctor.php (Took Mohammed AdminEditEmployee and made it work for Doctor )
      1. Used php to update info in the database
4. Made session tracking for our website so you can’t access certain pages if you're not signed in as the correct user.
   1. There are three different session files: patientSession.php, doctorSession.php, adminSesision.php
5. I made a admin search function so that they can filter through the patients and employees.
   1. Utilized AJAX for this function
   2. AdminSearch.php, AdminSearchTable.php
   3. Used php to query the database to grab information for the function (Table)
6. Got the CSS template to work on all doctor pages, AdminSearch,and AdminSearchTable. (Then Yuchen edited the information on the template)

Member 3: Eleftherios Lazaridis

(on github under lelakos90 and Mohammed1011 names since I was using two different computers as I had issues many times pushing on github from my personal pc, so I had to use a classmates pc to push sometimes)

My teammates and I agrees I did 25 % of the overall project. My specific tasks include

* Task 1 : I created the proposal.
* Task 2:  I created all the tables for our database on XAMPP and I set their primary keys, foreign keys according to the relationships.
* Task 3: I added sampled data to the tables I created that used to test our project.
* Task 4: I coded the part that when a patient logs in to his/her profile can make an appointment with a specific doctor on a specific date/time.
* Task 5: I tested two of my teammates forms offering suggestions on improvement
* Task 6: I coded:
  + Put common database code in common.php to simplify writing queries
  + Added the make appointment part to the patient.php
  + Page and forms to view, create, and cancel appointments with flexible times (e.g. thursday at 3:00 pm, tomorrow at 8:00 am).
  + Added create-appointment.php
  + Added cancel-appointment.php
  + Updated code to work with PHP 7
  + Got CSS/HTML template to work on front page
  + The "forgot password" process including extra columns in the database.
  + Added forgot-password1.php
  + Added forgot-password2.php
  + Added forgot-password3.php

Member 4: Yuchen Xun

My teammates and I agrees I did 25 % of the overall project. My specific tasks include

* Task 1 : I edited the proposal.
* Task 2: I wrote the final paper with all screenshots and all captures. The content of the final paper is approved by all team members.
* Task 3: I created the powerpoint presentation for demo after discussed and approved by the team.
* Task 4: I created and outlined the Ranking document for the team.
* Task 5 : I coded and edited all UI interface design. I maintain the github repository for the team.
* Task 6: Combine all codes from team members and set up database for the demo laptop.
* Task 7: Me and Seth present the presentation.
* I coded:
  1. UI design for newindex.html
  2. Admin.php and all other pages content design.
  3. Edited the direct.php page redirect back to the newindex.html if no credential has been put in.
  4. Doctor.php design alternation. Edited the page to let doctors see their name on the top along with “doctor portal” to show type of login. Then assembled other content from Seth to make sure the functionality of doctor’s page.
  5. UI design for patient.php. I edited patient.php to add separated navigation bar for patients’ to have access to profile, report and other related information.
  6. Organize github repository. Update wiki and readme
  7. I assembled all the code together with other team members to ensure connection to database is correct.
  8. I tested all the functionalities for the final code with all team members.

Team 8 members read and agreed with all the points made on the ranking documents.

Signature

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mohammed Al-hazza 12/6/2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Seth Fields 12/6/2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eleftherios Lazaridis 12/6/2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Yuchen Xun 12/6/2016