HOSPITAL APPOINTMENT SYSTEM

Assignment 3 - Student Report

CMPE 321 Spring 2016

Batuhan Demir - 2013400159

May 17, 2016

1 Introduction

In this project i tried to implement a hospital appointment system using PHP. The system includes some main objectives that can be summed as;

- Make / Edit / Cancel Appointments
- Add / Edit / Remove Doctors
- Add / Edit / Remove Branches

In this system we have 2 main user roles that *Admin* and *Patient.* Admins are authorized to see appointments, add or edit or remove doctors and branches. On the other side patients only can make or edit or cancel appointments for themselves.

The system redirects users according to their roles at homepage. Also a new user can register to the system. But administrators are defined beforehand in the system. As can be understood doctors and branches are defined by admins.

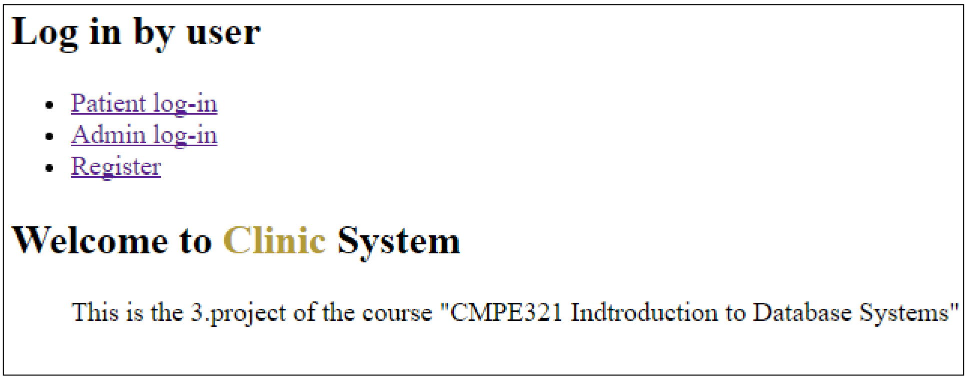
In the system each doctor has a branch and a doctor can not be added without branch that he/she serves. Per contra a branch can be defined without any doctors.

Appointments in the system are 5 minutes long. A user first selects branch and then doctor and time. In the appointments real world constraints are conserved. User can change his appointment or delete it.

2 Interface

The interface of the system is quite simple because of lack of style sheets. Users can easily find and use functions of the system though. We can say system has beauty of simplicity.

Main Page consists of welcome statement and user logins and a register link...



Log in by user

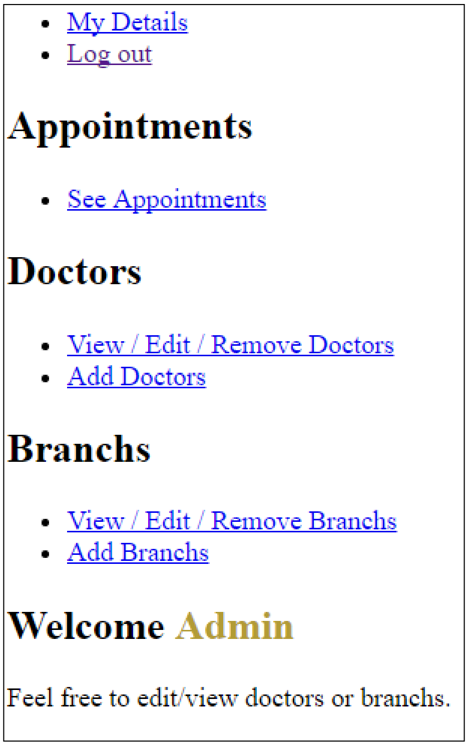
- Patient log-in
- Admin log-in
- Register

Welcome to Clinic System

This is the 3.project of the course "CMPE321 Indtroduction to Database Systems"

Figure 1: Main Page of Site

Admins have 3 main functionalities...



- My Details
- Log out

Appointments

- See Appointments

Doctors

- View / Edit / Remove Doctors
- Add Doctors

Branchs

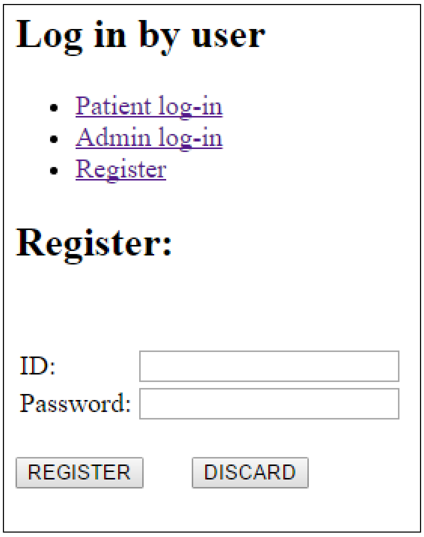
- View / Edit / Remove Branchs
- Add Branchs

Welcome Admin

Feel free to edit/view doctors or branchs.

Figure 2: Admin Page of Site

Users are only need an id and a password to register...



Log in by user

- Patient log-in
- Admin log-in
- Register

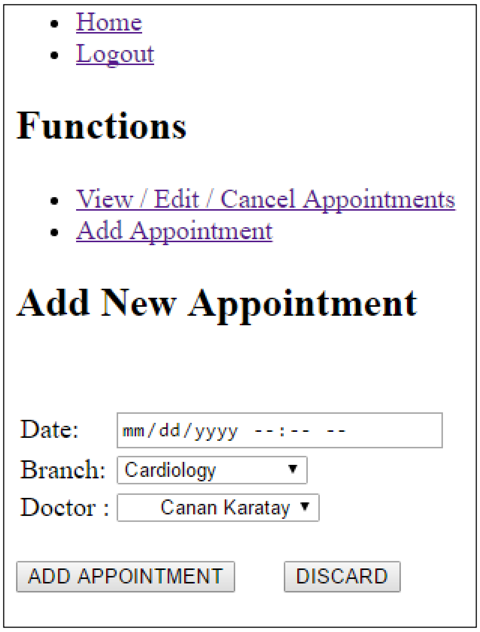
Register:

ID:

Password:

Figure 3: Register Page

Patients add new appointments...



- Home
- Logout

Functions

- View / Edit / Cancel Appointments
- Add Appointment

Add New Appointment

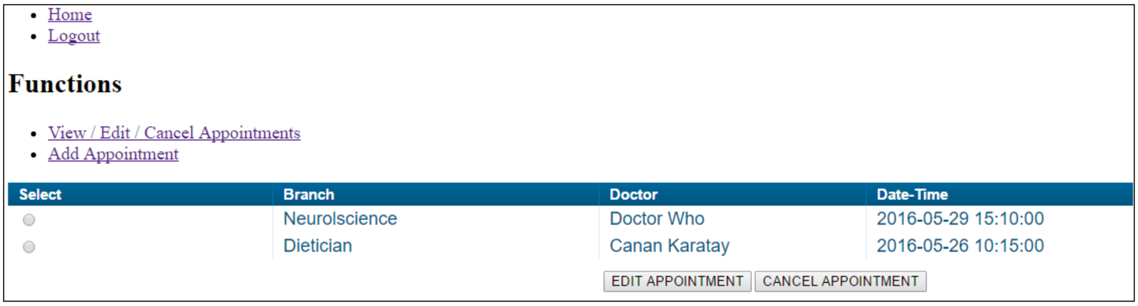
Date: mm/dd/yyyy --:-- --

Branch: Cardiology ▼

Doctor : Canan Karatay ▼

Figure 4: Add Appointment Page

Patients can see their appointments...



- Home
- Logout

Functions

- View / Edit / Cancel Appointments
- Add Appointment

Select	Branch	Doctor	Date-Time
○	Neurolscience	Doctor Who	2016-05-29 15:10:00
○	Dietician	Canan Karatay	2016-05-26 10:15:00

Figure 5: View Appointment Page

3 Database

The backend of the system consists of nearly 30 PHP pages and 5 SQL tables. The tables are such(underlines means primary key):

Patients (UserID, Password)

Doctors (ID, Doctor_Name, Branch_Name)

Admin (AdminID, Password)

Branch (Name)

Appointments (Patient, Doctor_Name, Branch, Date&Time)

There are some integrity constraints between tables. That is;

- Branch name in *Doctors* table is a foreign key references to "Name" field of *Branch* table.
- In *Appointments* table basically "patient" refer Patients(UserID) , "Doctor_Name" refers Doctors(Doctor_Name) and "branch" refers Branch(Name).
- When a branch is deleted from system all doctors that serves in the branch are also deleted.

Entity-Relationship diagram shows relations clearly.

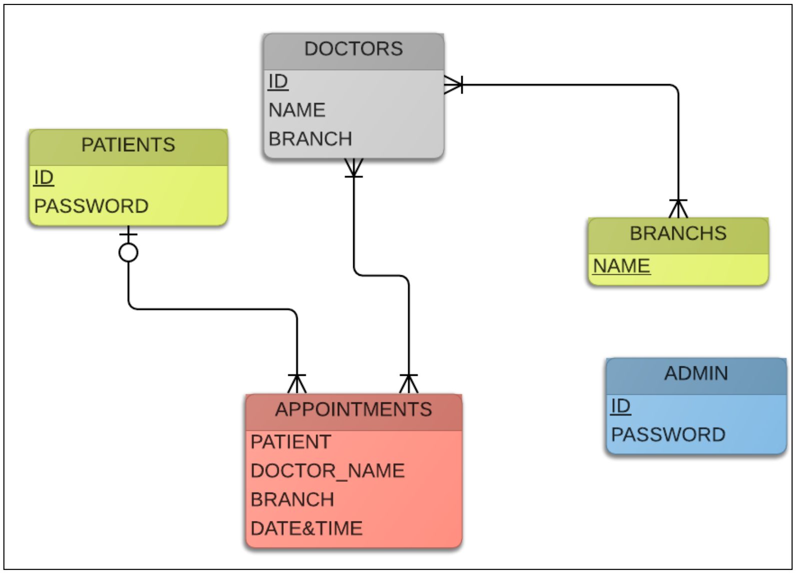


Figure 6: ER Diagram of the System Database

Stored Procedures

The system has 2 stored procedures in the database that lets admins to see all appointments according to time. Future, before appointments can be seen with the help of these procedures.

-
  ```
  CREATE PROCEDURE ‘get_app_bef‘(IN ‘br‘ VARCHAR(20))
      NO SQL
  SELECT pat,docname,datetime
  FROM appons
  WHERE br = branch AND datetime < NOW()
  ```

-
  ```
  CREATE PROCEDURE ‘get_app_bef‘(IN ‘br‘ VARCHAR(20))
      NO SQL
  SELECT pat,docname,datetime
  FROM appons
  WHERE br = branch AND datetime < NOW()
  ```

Trigger

The system has a trigger that is used when a doctor deleted all his future appointments are also deleted.

```
CREATE TRIGGER ‘doc_delete‘ AFTER DELETE ON ‘docs‘
 FOR EACH ROW DELETE FROM appons
    WHERE appons.docname = old.name
```

4 Conclusions & Assessment

The system is very simple and has limited functionality but with some effort this can be improved to work in more professional ways. Though simplicity, system has protection against SQL injections and has ’session’ control. These make it a bit advantageous.

After many hours spent, the project is done. At the end, I’ve implemented ”The Clinic System” that satisfy requirements given in the project description.It was my first web application and i think i got quite familiar with PHP and HTML. It helped to me to learn many things and I think i will make use of them at work also.