**Clinic Booking System**

**Abstract**

Our project helps patients to book an appointment with a clinic, find an appropriate doctor and visit time. It also helps doctors and their assistants manage which patients are to be seen. In the form of a website using a database backend that presents different views depending on who logs into it. It allows patients to select potential times to see a certain doctor for their condition, and for clinic personnel (doctors and assistants) to manage patients as well as view their associated medical history. The idea of this was to simplify the process of seeing a doctor for a patient.

This prototype of a booking system uses snippets of *bootstrap* for the appeal, however heavily focuses on the *sql*, *javascript* and *php* aspect. The focus was the functionality of which was shown as this implements the deliverables suggested from the initial stages of the project. *phpMyAdmin* was used to setup a database and *XAMPP* was used to host a server to link the databaseand the front-end of the website for this prototype. A database was created using *MySQL* using *phpMyAdmin* to manage the tables located in it.

The remainder of this report will follow the recommended format, which we will describe in the sections following below.

**Introduction**

**Problem:**

A majority of walk-in and specialty clinics have been and are currently still booking patients in by phone call only. Surprisingly, only very recently have practices begun adopting online booking for patients. This problem is interesting since many different industries have had this level of convenience for over a decade. There are some extra complexities in the healthcare industry that may have slowed this adoption, but we believe we can create a solution (given the constraints of our model) that can effectively handle patients booking themselves to see their doctor of choice.

This problem occurs whenever someone has a minor health issue such as an infection or a more serious condition that requires continual treatment from a medical professional. This problem is already solved in some specialty situations such as physiotherapy (LifeMark) or optometrist clinics (FYidoctors). Generally, they ask you for basic information like your full name, date of birth, etc. as well as some medical information to create a record of you. We look to add these kinds of features to general clinics. This is because there are general clinics that don’t have things that one would assume are basic functions. Certain clinics I’ve seen allow one to choose not only a date that had already passed, but hours outside of their available hours of operation (Rockyview Medical). As a result of the previous example, We believe our solution is needed because we feel that many medical clinics are behind in offering patients an easy way to see what times a doctor is available to be seen. We think our project will be a model system that can be widely adopted and offers something unique in that it will provide medical personnel a convenient way to schedule patients.

**Solution:**

Our solution attempted to create a streamlined and convenient way for patients, doctors and receptionists to facilitate medical visits. This was achieved through a website that gives each user type has a tailored experience that interacts with other user types to seamlessly transition between the end of one person’s role and the beginning of another. That is, between patients, general physicians and receptionists. This website was created using PHP, javascript + javascript technologies. The specifics of the solution will be discussed in the section following.

**Project Design**

In our project, we have 4 types of users who can access the system. Those would be the patients, general physicians, receptionists and a database administrator. Physicians and receptionists are grouped together as “Clinic Staff”. Clinic Staff, the admin and patients each have their own login page and from their access different actions. All pages have a logout button

For patients, they have an easy way to select a general physician of their choice after they have chosen an appropriate clinic with a location. They can then select one date and time for a visit with any available doctor. If a patient tries to select a time that’s already booked, the appointment will fail and they will be asked to book another appointment. When a patient creates an appointment request successfully, the default value set at the confirmation is a *maybe* denoted as an "m" as it has yet to be approved by a receptionist. Patients can only see appointments associated with their accounts.The appointments made by the user would be located in the appointments table that is seen by the patient, receptionist and doctors but in different manners.

Doctors can view appointments and update patient medical logs.

Patients have attributes such as their ID represented as “HealthNumber”. Each patient would also have their own medical record attached by their ID as well. This medical record would include things such as their height, weight, current medications currently taking. Doctors can view these records and update each field accordingly. It should be noted that for the doctor to include a medication that the patient is taking in their record it must exist in the medication database of the system. Also if that patient has been referred to a specialist at some point the specialist ID must be in the specialist database of the system.

The appointments made by the patient are viewable by a certain doctor if the appointment is with the logged in doctor and has been approved by a receptionist. The doctor will see the time, reason and date of the appointment because they already know the location they work at and that they are the doctor seeing the patient.

Receptionists confirm appointments and view logs.

If a patient has requested an appointment with an “m” then this is for the receptionist to confirm (denotes as a “y”) or deny (denotes as a “n”) the appointment if any other emergency or event were to occur. Thus by checking their appointments if the patient sees a yes it means the appointment has been confirmed, if they see a no they can then cancel the request to reduce clutter on their inbox and make that time slot available for other patients to request again. Receptionists can also view the medical record of any patient at anytime but unlike the doctor, a receptionist cannot update them. This is mostly for administrative purposes(people need to see their information while in the clinic, doctor is away from computer etc etc).

Another user to mention would be the admin. There exists a single admin who is able to add more clinic staff and patients. When adding in patients and staff one cannot create patients and staff using emails already in either database. The Admin is also able to delete clinic staff and patients from the database. The admin selects the corresponding radio button and then the user type specified is deleted by their ID. When deleting a patient from the database the corresponding patient log is also deleted.

**SEE ERD in FOLDER**

**Implementation**

Xampp was our DBMS. We chose it because it supported MySQL, PHP and APACHE thus giving us everything we needed to create and test our website in a convenient way. It also allowed us to both access the database and website at the same time.

**SEE RM in FOLDER**

**SQL Queries:**

1. Extra error checking statement to make sure the staff works at the location selected by the user appointment. Later found a better way to check this.

SELECT Name

FROM ClinicStaff

WHERE Name = '$doctor' AND ClinicLocation = '$location'

1. Creates appointment from patient request in table appointment

INSERT INTO Appointment (ID,DateTime,Reason,Location,Doctor) VALUES ('$name','$date', '$reason','$location','$doctor')

1. Prevents admin from creating an account where an email is already used for patient

SELECT Email FROM Patient WHERE email = '$email'

1. Creates new entry in patient table

INSERT INTO Patient (HealthNumber,Email,Name,Password,Sex,Address,PhoneNumber,DOB) VALUES ('". $healthnumber ."','". $email ."','". $name."','". $password."','". $sex ."','". $address ."','". $phonenumber ."','". $dob ."')

1. Same as 3 but for ClinicStaff

SELECT Email FROM ClinicStaff WHERE email = '$email'

1. Creates new entry in ClinicStaff table

INSERT INTO ClinicStaff (EmID,Email,Name,Password,JobType,ClinicLocation) VALUES ('". $emid ."','". $email ."','". $name."','". $password."','". $jobtype ."','". $cliniclocation ."')

1. Grabs every clinic location for appointment booking

SELECT location

FROM Clinic

1. Admin deletes a Patient

DELETE

FROM Patient

WHERE healthnumber = '$id'

1. Admin deletes Staff

DELETE

FROM ClinicStaff

WHERE emID = '$id'

1. Gets doctors that work at the patients desired location for appointment booking

SELECT Name

FROM ClinicStaff

WHERE JobType = 'p' AND ClinicLocation = ?

1. Confirms that the entered credentials on login match a patient in the corresponding database

SELECT \*

FROM Patient

WHERE healthnumber = '$ID' AND Password = '$password'

1. Same thing as above but this time for staff

SELECT \*

FROM ClinicStaff

WHERE EmID = '$ID'

1. Same thing as above but for Admin

SELECT \*

FROM Admin

WHERE master = '$ID'

1. Gets all attributes of patient log corresponding to given patient ID to display to doctor and same query is used for receptionist

SELECT \* FROM PatientLog where ID=".$ID

1. Gets the all medication in database in case the doctor wants to prescribe patient more medicine

SELECT Name

FROM Medication

1. Shows all patients to the doctor

SELECT \* FROM Patient

1. Allows patient to cancel appointments (DateTime + Doctor is primary key for Appointments)

DELETE FROM Appointment WHERE datetime = '$DateTime' AND doctor = '$Doctor'

1. Allows patients to see all appointment requests and appointments they have made

SELECT DateTime,Confirmation,Reason,Doctor,Location FROM Appointment WHERE ID = '$name'

1. Allows doctors to see their appointments with patients

SELECT \*

FROM Appointment

INNER JOIN ClinicStaff

ON Appointment.Doctor = ClinicStaff.Name

WHERE confirmation = 'y' AND emID = '$ID'

1. Used to save changes doctors make to the patient log

UPDATE PatientLog

SET Medication = '$medication',

Height = '$height',

Weight = '$weight',

RefersTo = '$refers',

Pastappointments = '$pappt'

WHERE ID = '$ID'

1. Allows receptionist to confirm patient appointment by updating patient log with “y”

UPDATE Appointment

SET Confirmation = 'y' WHERE Datetime = '$DateTime' AND Doctor = '$Doctor'

1. Allows receptionist to deny patient appointment by updating patient log with “n”

UPDATE Appointment

SET Confirmation = 'n' WHERE Datetime = '$DateTime' AND Doctor = '$Doctor'

**References:**

The textbook

Lecture Slides

Code references are in comments inside of code files.