

SMART PATIENT SYSTEM



SUBMITTED BY

Ruckse kayna
ID:2014-3-50-017

Farjana akter
ID:2014-3-50-023

SUPERVISOR

Zahidur Rahman
Lecturer

Department of Electronics & Communications Engineering
East West University

A project submitted for the requirements of ICE 498 in the partial fulfillment of Bachelor of
Science in Information & Communications Engineering.

Declaration

We, hereby declare that this project is based on results we have found ourselves. Materials of work from researchers conducted by others is mentioned in references.

Signature of Supervisor

Zahidur Rahman

Lecturer

Department of Electronics and Communications Engineering

Signature of Authors

Ruckse Kayna

2014-3-50-017

Department of ECE

Farjana Akter

2014-3-50-023

Department of ECE

ABSTRACT

Our proposition is to make an online system which will appoint a doctor to a patient dynamically and patients can speak with specialists over on the web. In today's world on the off chance that somebody needs to make an arrangement to a specialist needs to go to hospital or clinic physically and then make the arrangement. This devours valuable time of the patient. Additionally, if the specialist crosses out his/her calendar, the patient does not come to know about it unless he/she goes to the center. Other than this, individuals living in significant urban areas have access to quality specialists yet have to struggle because of the absence of extra time and substantial congested road. However, individuals living in rural areas don't have legitimate medical support inside their range. To minimize the issue, we are attempting to build up an online system where patients can have quality medical administration from qualified specialists everywhere throughout the nation. Specialists will straightforwardly communicate with the patient through a chat server. Furthermore, the system will be a stage for new specialists. The specialist will come to know a number of patients he needs to consult on an entire day. Our system "Smart Patient System" will eliminate the receptionist's paperwork. We have utilized data compression to lessen the amount of data stored in the system. Past records are being saved into the patient's profile from where he/she can reappoint a previously consulted doctor.

Acknowledgment

The authors wish to thank their mates in for the discussions, for the sleepless nights they were working together before deadlines, and for all the memories they have had in the last four years. Also, they want to thank their friends in East West University. In particular, they are grateful to Zahidur Rahman, for the support and inspiration. Lastly, we can never be too grateful to our loving parents for their blood, sweat, and tears. We will always remain in debt of what our parents have provided us with. They have always encouraged us to gain knowledge. It is because of them and their effort we have reached this far away. So, we dedicate our effort to our dear parents, our respectable supervisor, every other single being that has assisted us practically and impractically.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION.....	1
Background.....	2
Motivation.....	2

CHAPTER 2

DESIGN OF THE SYSTEM.....	4-30
System Specification	
2.1CSP.....	4
2.2OLTP.....	5
2.3XAMPP.....	5-6
2.4Modules.....	6
2.5 Calculation.....	7
2.6 Statement.....	7-8
2.7 Flowchart.....	8-9
2.8 000Webhost.....	10
2.9 Security Testing	11
2.10 Cancel Appointment.....	11
2.11 Search for Doctor.....	11
2.12 AJAX.....	12
2.13 System Architecture.....	13
2.14 Use Case Diagram.....	13-14

2.15: ER Diagram.....	15-16
2.16: Activity Diagram.....	17-18
2.17Implementation.....	19-30

CHAPTER 3

RESULT & ANALYSIS.....	31-38
3.1 Register & Login.....	31
3.2 Edit Profile	33
3.3 Updated Profile.....	33
3.4 Appointment System.....	35
3.5 Chat System.....	37
3.6 Prescription.....	38

CHAPTER 4

CONCLUSION	39
4.1 Conclusion	39
4.2 Future Plan	39
REFERENCES.....	40-42
APPENDIX.....	43-51

Table of Figures

Figure 2.1: CSP of Algorithm.	4
Figure 2.3: Logo of XAMPP.	5
Figure 2.4: System Architecture.....	6
Figure 2.7: Flowchart of The System.....	8
Figure 2.8: Logo of 000Webshost.....	10
Figure 2.11: Doctor Login.. .	11
Figure 2.12: Logo of AJAX	12
Figure 2.14: USE Case Diagram.....	13
Figure 2.15: ER Model.....	15
Figure 2.16: Activity Diagram.	17
Figure 2.17: Department Table	19
Figure 2.18: Doctor Table.....	20
Figure 2.19: Schedule Table... ..	21
Figure 2.20: Algorithm for Back Tracking	22
Figure 2.21: Forward Checking Diagram.....	23
Figure 2.22: Code for Fetch	24
Figure 2.23: Code for Load Balance and scheduling	25
Figure 2.24: Home	27
Figure 2.25: Appoint Doctor.	28
Figure 2.26: Appoint Record.	28
Figure 2.27: Scheduled.	29

Figure 2.28: Chat Server.	30
Figure 3.1: Register	31
Figure 3.1: User log in.	32
Figure 3.2: User Profile	33
Figure 3.2: Edit Profile.	33
Figure 3.3: Update Profile.	34
Figure 3.3: Appointment	34
Figure 3.4: Appointment System.....	35
Figure 3.4 Appointment Schedule	35
Figure 3.4: Old Record	36
Figure 3.5: Chat System	37
Figure 3.6: Prescription	38

Chapter 1

Introduction

In the age of information, we have access to almost every necessity that's needed to go on with our day-to-day lives. With the help of modern technology, we are now able to access almost everything within a very short period of time. We have access to almost all of the information of the world. Along with better performance, modern technology is also very much focused on saving time in the process. As it is evolving it is focusing more on getting the same or more amount of work done but in a smaller time frame. Technology has a huge contribution to medical sciences as well but still it hasn't really brought the medical world "to us" as it has many other things. People still do not have a quicker medical access, so they have to rush to the hospital every time they are under the weather seeking consultation. This consumes a lot of time because of traffic and waiting in the line for an appointment etc. also especially in third world countries like Bangladesh itself, we do not have sufficient qualified doctors all over the country. So if someone comes down with a sickness their options are to either rely on the medical services they have near them which may or may not be useful or go to a place where they might find qualified medical assistant which can be time-consuming as well as very costly or in some cases unaffordable for many people. Our system "Smart Patient System" is designed based on these particular problems. It is a specially designed system that will eliminate the need to rush to the hospital for the smaller medical issues. Through the system, patients will have the ability to access professional medical consultation by just taking a few short moments to fill up an online information form to set an appointment with a doctor for both online consultation and physical if the consultant deems it necessary. Patients will also have the option to upload personal documents such as previous report cards, tests etc. Our system is based on the thought that people from all around the world may have access to better and faster medical assistants. Implementation of a special algorithm will allow the system to search and sort through the doctor's schedule and manage the appointments accordingly. Doctors and patients will interact with each other online chatting which will be controlled by a central server. To get the best performance from the database advanced OLTP has been used. After the consultation period, the records will be stored for future requirement. Furthermore, there is also an online pharmacy

Available 24/7 will allow users to purchase their medication from. However purchasing drugs from the pharmacy will require prescriptions.

1.1 BACKGROUND

In the 2000s, many individuals came to regard the web as to begin with, or possibly a noteworthy, the wellspring of data and communication. Health counsel is presently the second-most famous subject, after obscenity that individuals look for on the internet. With the coming of broadband and video conferencing, numerous people have swung to online specialists to get online discussions and buy professionally prescribed medications. Utilization of this innovation has many favorable circumstances for both the specialist and the patient, including cost investment funds, convenience, accessibility, and enhanced protection and correspondence.

In the US, a recent report found that hunting down data on a solution or over-the-counter medications was the fifth most famous pursuit theme, and a recent report found that 4% of Americans had acquired professionally prescribed meds on the web. A 2009 overview led by Geneva-construct Health In light of the Net Foundation discovered one-in-ten Europeans purchases pharmaceuticals from sites and 33% claim to utilize online discussion. In Germany, roughly seven million individuals purchase from mail-arrange drug stores, and mail-arrange deals represent around 8–10% of aggregate pharmaceutical deals. In 2008, the Royal Pharmaceutical Society of Great Britain reported that roughly two million individuals in Great Britain were frequently acquiring pharmaceuticals on the web.

1.1 MOTIVATION

Individuals living in town don't have medicinal offices inside their range. To wipe this issue there exist some online social insurance framework. Those administrations are for the most part static and wellbeing web journals and just specialists contact databases. Utilizing those web journals individuals can find out about different wellbeing tips, yet there were not very many administrations who really offered online specialist counseling administrations.

In our project, we propose an online human services framework where the patient can specifically speak with a specialist through online talk framework. There will be an online enlistment, process, planning and rescheduling process for the patient for their further arrangement. Last yet, not the minimum there will be a protected and secure installment strategy in our framework.

2.2 OLTP:

Online Transaction Processing is a data framework sort that organizes exchange preparing, managing operational information. This sort of PC frameworks is recognized by the huge number of exchanges they bolster, making them the best to address online application. The primary utilizations of this technique are all sort of value-based frameworks like databases, business, healing facility applications etc. Just, these frameworks accumulate input data and store them in a database, on a vast scale. The vast majority of today's applications depend on this communication technique, with executions of the concentrated or decentralized system. [13]

2.3 XAMPP:

XAMPP is a free open source cross-platform web server which is developed by “Apache Friends”. It consisting of “Apache HTTP Server, MariaDB database” and interpreters for scripts written in PHP and Perl. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). Because of the advantage of lightweight, it is easy to create a local web server for testing and deployment purpose. It has everything to set up a web server. XAMPP is also cross-platform for this it equally works on Linux, Mac, and Windows. It makes easy to the transitioning local test server to live server. [2]



Figure2.3: Logo of XAMPP

We here propose a doctor patient handling, managing system that helps doctors in their work and also patients to book doctor appointments and view medical progress. The system allows doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name. The system manages the appointment data for multiple doctors for various date and times. Each time a user visits a doctor his/her medical entry is stored in the database by doctor. Next time a user logs in he may view his/her entire medical history as and when needed. At the same time a doctor may view patients medical history even bore the patient visits him. This allows for an automated patient doctor handling system through an online interface.

2.4 Modules:

Admin Login: The system is under supervision of admin who manages the bookings made. User login/registration: Users have to first register themselves to login into the system. Medical History: System allows updating and viewing patient medical history. Doctor Search: System allows for doctor search through categories, name and location. Appointment availability check: User can click on spaces to view the availability. Appointment booking online for date and time: Users can book appointment for their required date and time.

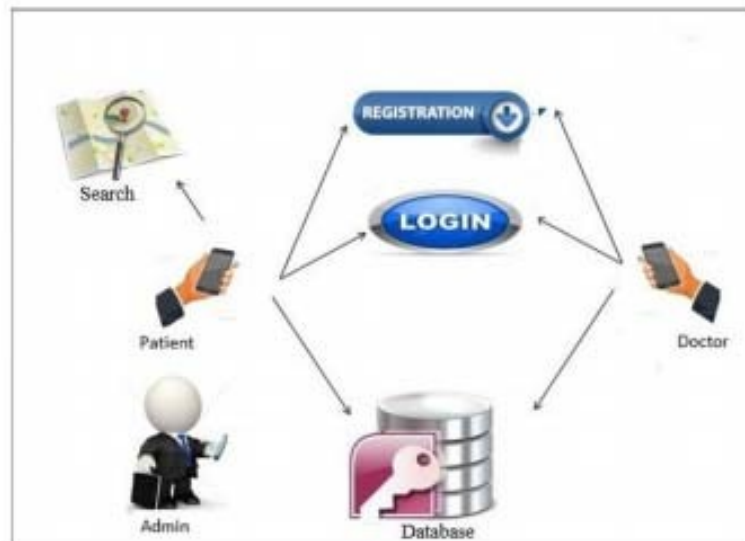


Figure2.4: System Architecture (researchgate.net)

2.5 Automatic cost calculation:

The system calculates the total cost incurred for parking based on the time that user has asked for booking. Booking cancellation: User may even cancel their bookings by login into the system anytime. Email on appointment booking:.

"Smart Patient system" is being developed to assist a patient and doctor can register; view the medical record anywhere, anytime. Also reduce the overhead of handling files every time. Medical record is available all time. This is an innovative application for patient and doctor acceptance that will provide comfort, convenience and efficiency in everyday life.

i) Goals: To eliminate the burden of carrying medical files including reports and prescription. To save time of managing all the tasks for searching reports which are in paper format. To provide user friendly efficient service to all users. To provide reminders for medicine consumptions.

ii) Objectives:

To develop an application which allows patient to view the medical history /records online. To reduce the time consumption required for visiting different doctor at different places. To provide user friendly and time saving experience to the patient and doctor.

2.6 Statement of scope:

This application is used in several Hospitals and Organizations. In the future, patients can view their medical reports/history. All in one app. The scope of the proposed system is justifiable because a large amount of the population face the problem of managing the medical records in form of files and papers. The system architecture will simplify the whole system in such a way that every user of the system gets benefits. As shown in the figure, there are 3 main users: Admin, Doctor, and Patient. It's a tedious job to handle all the medical reports and the generated prescriptions through that. One finds it hard to maintain the prescription and reports while travelling or shifting to a new place. A new doctor needs to examine the patient overall again to get the exact detail medical condition of the patient.

Therefore if the data is digitized and available to Doctor as well as Patient on the tap of their finger then it becomes easy to handle such digitized data. Here the main function is to provide digitized report and prescription to Doctor and Patient and to provide free and online access to the data. Patient registers on the system with detailed information.

2.7 Flowchart of the system:

User ID and password is also provided to the Patient for login. Doctor is verified by Admin. Doctor also registers and gets provided with Unique user ID and password, which he/she can use for login

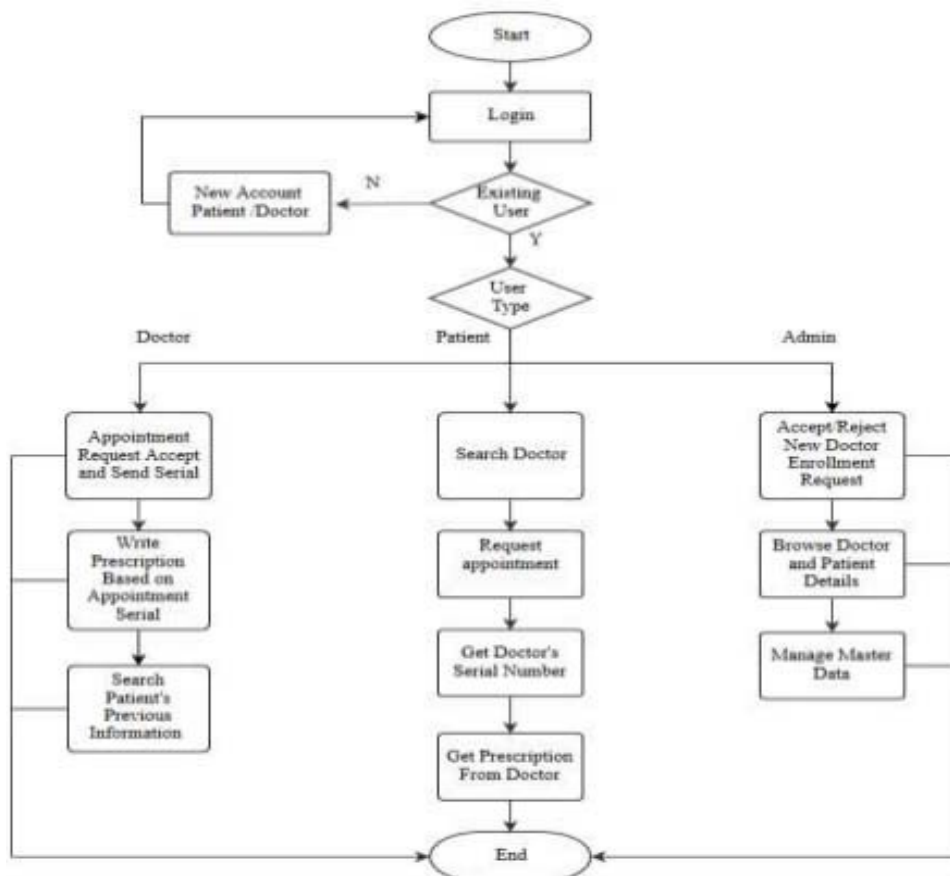


Figure2.7: Flowchart of the system

This application doesn't provide user registration on their website. Therefore, every time a

user wants to make an appointment, he/she must input all the contact information such as email, name, date of birth and phone. If it's a frequent patient, it might be a waste of time to introduce this information every time he wants to make an appointment. After making an appointment. This is a measure to prevent false appointments scheduling. It also means that all the appointments made online must be looked by one employee who will have to contact the patients. After an appointment was made, the only things that can be done is to cancel it or reschedule the appointment to another day. Moving to Search for Doctors feature, doctors can be searched by different criterions such as working day of the week and time of the day, gender, name or specialty. It's a very nice system since it's very specific and anyone can find a perfect doctor match. If a user has any kind of inquiry, he/she can easily contact the clinic using a contact form. Once again, the user will have to fill their contact information every time they will want to contact a clinic. If it's an inquiry for another person, the user will have to put his/her contact information too.

2.8 000WebHost:

00Webhost is a free web facilitating space that is being utilized all through the world. A free site facilitating is a non-paid facilitating administration that gives constrained advertisements on to the supporter's site. For the most part gives a sub-space or a catalog. It is protected, free and simple to dispatch site. Along these lines, we have utilized 000WebHost for starting testing of our venture.



Figure2.8: Logo of 000WebHost.

2.9 Security testing:

Security testing checks whenever a software/website is secure or not by checking is it is vulnerable to attacks, of anyone can access a database or if anyone can login without authorization. Several test cases exist to check some of the security of the website that we performed before deployment phase:

- Access bookmarked page without logging into the system
- Check password strengths
- Restricted pages can't be accessed by restricted users

2.10 Cancel appointment:

In order to cancel the appointment, the patient should be logged into his account and, first of all, list all his appointments and, near the appointment that a patient wants to cancel, click on a button Cancel appointment. There will be a confirmation before doing the cancellation in case a patient accidentally clicked on the button.

2.11 Search for doctors

Website users are able to search for doctors that work in the clinic based on different criterions. All the criterions are not required to fill in. A user has to access “Search for doctors”page and fill the fields that he finds necessary (therefore the inclusive gateway in a diagram above). If none fields are filled, all the doctors will be shown. After submitting the search form, the resulting list of doctors will be shown.

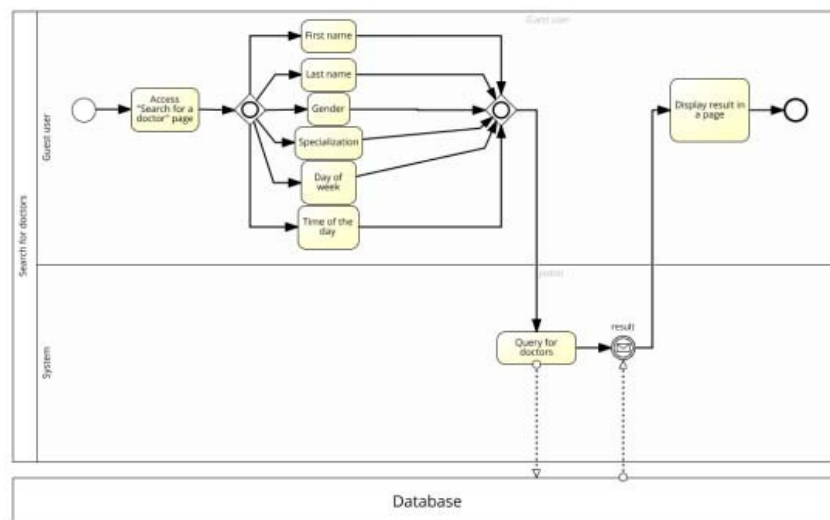


Figure2.11: Doctor login

2.12 Ajax:

Ajax (Asynchronous JavaScript and XML) is a technique for building intuitive applications for the Web that procedure client asks for promptly. Ajax consolidates a few programming devices including JavaScript, dynamic HTML (DHTML), Extensible Markup Language (XML), falling templates (CSS), the Document Object Model (DOM), and the Microsoft protest, XMLHttpRequest. Ajax permits content on Web s to upgrade instantly when a client plays out an activity, not at all like HTTP ask for, amid which clients must sit tight for a radical new to stack. For instance, a climate estimating site could show nearby conditions on one side of the immediately after a client sorts in a postal division [16].



Figure2.12: Logo of AJAX

2.13 System Architecture

In this section, we discuss system architecture which will describe our system flow, actors, entities, coordination. ER diagram and Activity diagram will describe the full design of the system and flow of the system.

2.14 Use case diagram:

A use case diagram is a graphic depiction of the interactions among the elements of a system. A [use case](#) is a methodology used in system analysis to identify, clarify, and organize system requirements .

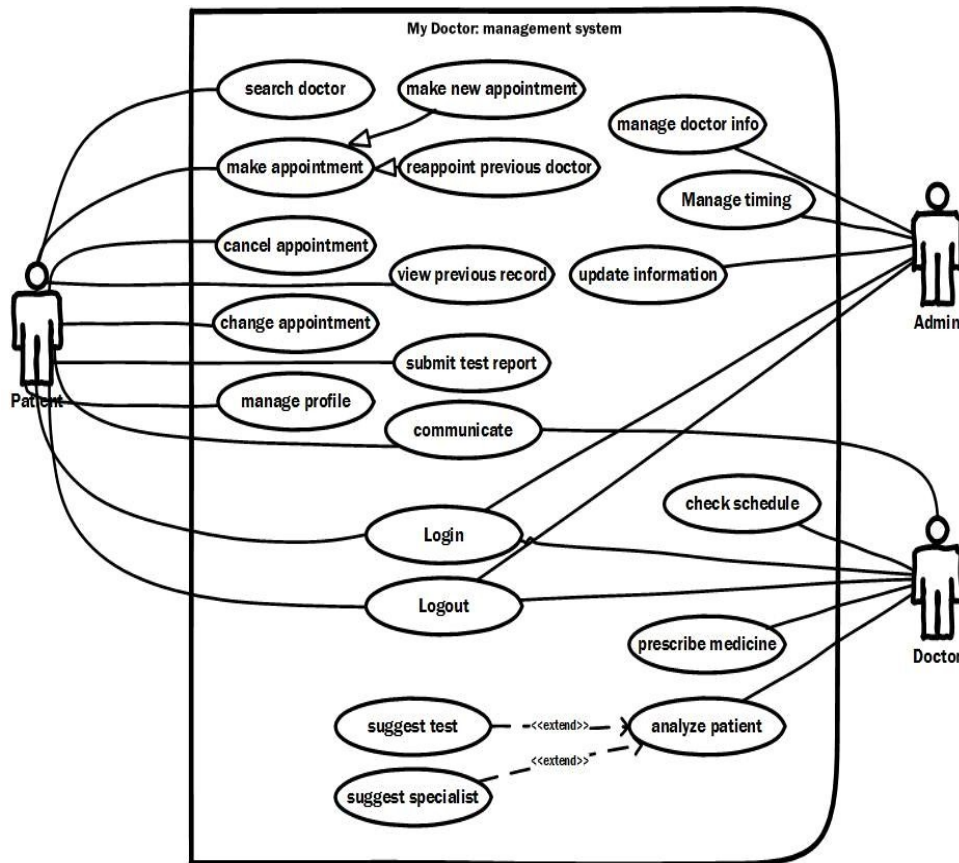


Figure 2.14: Use Case Diagram

Here in our system, there are three actors who will be able to access the system. They are an admin, doctor, and patient. All of them will have to log in with appropriate information. Admin will be able to see all the information and activity of the system. He can also update any information including adding or discarding a doctor and timing slot modification. The doctor will be able to check the timing and the information about the patient he is being assigned to. Doctor and patient will communicate through an internal chat server. If any test or specialist in other sector is needed the doctor can also suggest that to the patient otherwise he will prescribe medicine accordingly. Doctor and patient will be able to update their personal information. After logging in the patients will be able to select any specific medical sector and request for an appointment. He can either a new specialist or choose the previous one with whom they already consulted.

2.15 ER Diagram:

ER-modelling is a data modeling technique used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modelling technique are called Entity-Relationship Diagrams, or ER diagrams or ERDs. So we can say that Entity Relationship Diagrams illustrate the logical structure of databases [20].

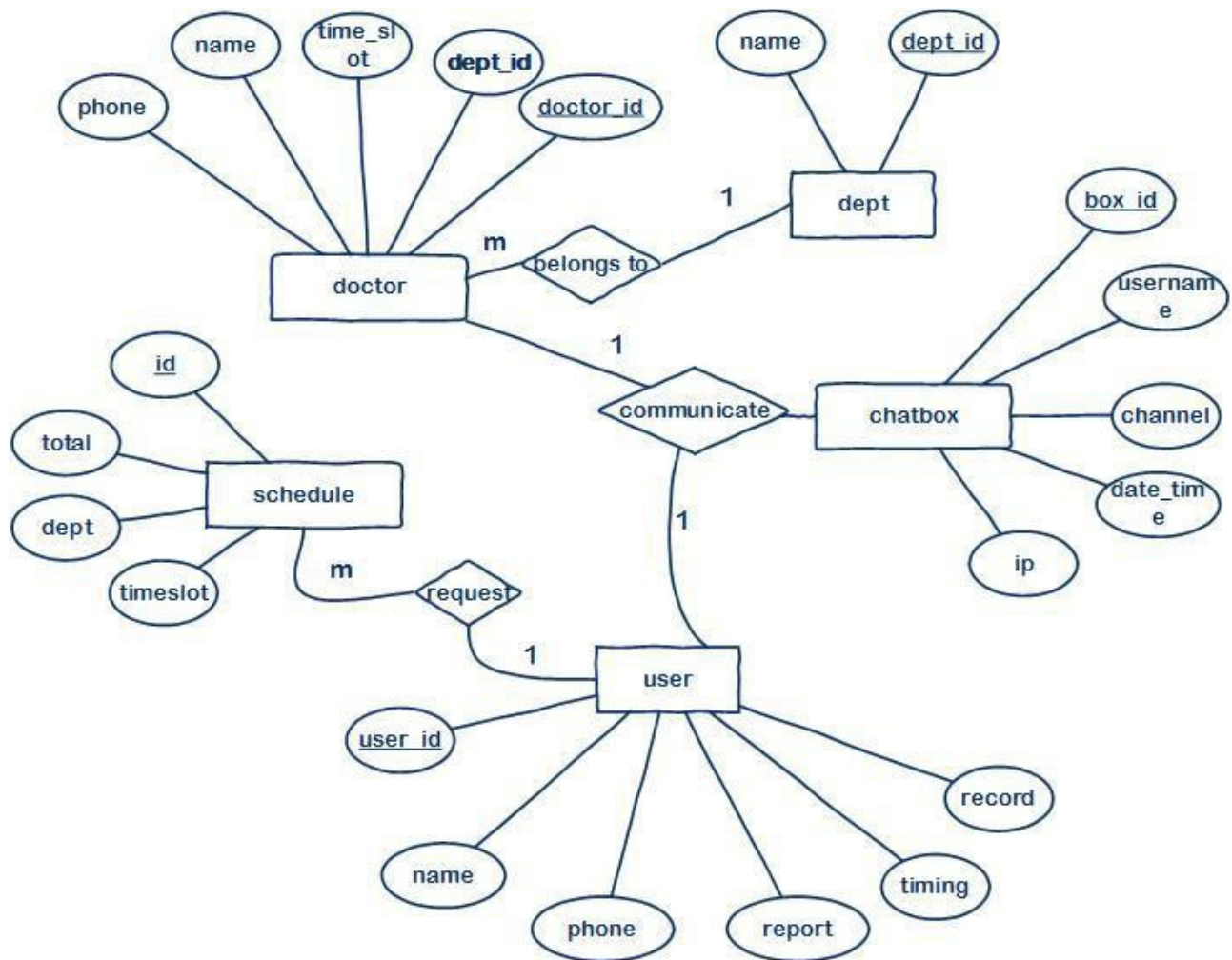


Figure2.15: ER Diagram

In the system we developed, we used this database model shown above. Here, there are five main tables in our database. Those are a doctor, user, schedule, dept (department) and chatbox. User

and doctor table contains all the information about the user and doctor accordingly. Both of the tables have a specific id for doctor and user which is automatically generated when inserted a new record into the database. Dept table will contain the information about which doctor is of what specialty. Every doctor belongs to some department and one department can have more than one doctor. Chatbox table is used for maintaining the communication among the doctor and the patient. Schedule table keeps all the data about timing. This table has an array of time slots allocated for any appointment which keeps track of all the available and booked slot. It can also provide the information about which doctor have more appointment and which doctor have less in order to make balance the load more efficiently.

2.16 Activity Diagram:

An activity diagram is used to model a large activity's sequential workflow by focusing on action sequences and respective action initiating conditions. The state of an activity relates to the performance of each workflow step.

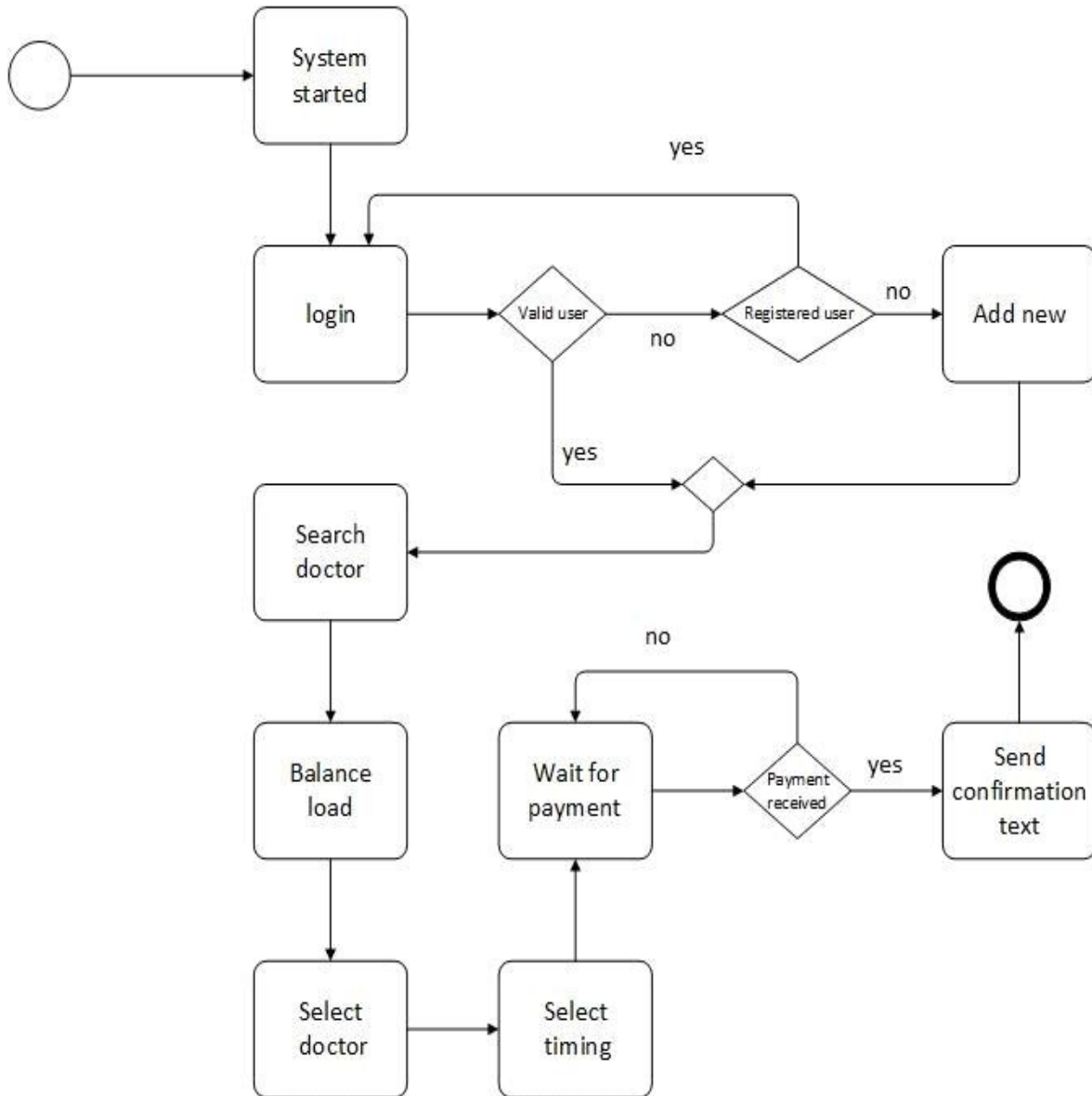


Figure2.16: Activity diagram

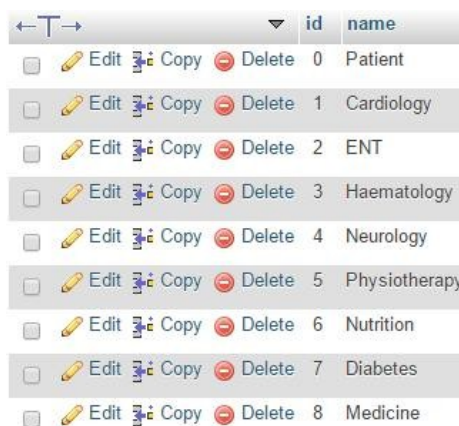
After the system is started which is triggered by the user by entering into the system, the system will ask the user to log in. If the user is not registered previously the system will navigate the user to the registration option. After verification, the user will be able to search for the specialist as their interest from the availability options given by the system. When the user selects his specifications the system runs an algorithm to balance the load among the doctors and select one from the available pool and a time slot for the meeting. After this process is done it will till the payment is done. When the system gets the confirmation of payment it sends a confirmation notification to the user through text. At the scheduled time the user will be able to communicate with the assigned doctor.

Implementation

The system we are trying to build here is actually based on dynamic scheduling and searching. We have researched several algorithms for the efficiency of scheduling. Finally, we got our desired solution from Constraint Satisfaction Problem (CSP) algorithm. Here each of the constraint checks individually for the best case. We implemented the algorithm using PHP for merging with web language. Algorithm took data from the database and start the searching for scheduling. Searched results showing on the user interface to confirm user appointment. For communication between doctor and patient, there is a chat server where the doctor can create rooms and prescribe patient.

Database:

The database design is based on ER diagram. It contains all the patient and doctor records and all appointment history of each individual patient. For scheduling doctor time it contains a table for each day. All kind of files which that are to be uploaded by patients or doctors will be stored in this database. For keeping the database lighter all large file like images or patient reports are to be kept on the server or cloud storage [19].



	id	name
<input type="checkbox"/> Edit Copy Delete	0	Patient
<input type="checkbox"/> Edit Copy Delete	1	Cardiology
<input type="checkbox"/> Edit Copy Delete	2	ENT
<input type="checkbox"/> Edit Copy Delete	3	Haematology
<input type="checkbox"/> Edit Copy Delete	4	Neurology
<input type="checkbox"/> Edit Copy Delete	5	Physiotherapy
<input type="checkbox"/> Edit Copy Delete	6	Nutrition
<input type="checkbox"/> Edit Copy Delete	7	Diabetes
<input type="checkbox"/> Edit Copy Delete	8	Medicine

Figure2.17: Department Table

This is the department table of the database which can handle the specialty of a doctor. Each department has a unique id which helps it to separate from other.

		id	dept_id	time_slot	f_name	L_name	phn	bkash	username	password	
<input type="checkbox"/>				160820032654	1	080910141516	Sayeed Hasan	01611232062	-1	sayeedhasan169@gmail.com	\$2y\$10\$hE5UvkrYH2UyE.KIQ2Uln
<input type="checkbox"/>				160821121720	1	111213171819	Hasan Sayeed	01869506451	-1	sayeedhasan169@hotmail.com	\$2y\$10\$TKJYuzE.TOCLXaHF2yon
<input type="checkbox"/>				161103095631	2	121314152021	Ashraf Hasan	0100000	-1	ashraf@gmail.com	\$2y\$10\$5ggz5rGuV9xbPfo8jupj.3
<input type="checkbox"/>				161103095752	2	080910141516	Hasan Seraji	0100000	-1	ashraf@gmail.com	\$2y\$10\$5k0V9CZJ52bmxNMFNE1f
<input type="checkbox"/>				161103095817	2	111213171819	Seeraji Ashraf	0100000	-1	ashraf@gmail.com	\$2y\$10\$FEfKQ1qs62RFIkpmilj/Nu
<input type="checkbox"/>				161103095858	3	080910141516	Shuvodeep Biswash	01700000000	-1	shuvodeep20@gmail.com	\$2y\$10\$37J.8kwS4RmDbzkeVz9di
<input type="checkbox"/>				161103095922	3	111213171819	Biswash Shuvodeep	01700000000	-1	shuvodeep20@gmail.com	\$2y\$10\$CGHPzZPm7jmJbhc/mLv
<input type="checkbox"/>				161103100035	4	080910141516	Abu Bakkar	01900000000	-1	mail.bakkar@gmail.com	\$2y\$10\$SanjL3Q6SGlhCnLEuzUK/T.
<input type="checkbox"/>				161103100115	4	111213171819	Bakkar Siddq	01900000000	-1	mail.bakkar@gmail.com	\$2y\$10\$IWCnG0hphwUdgd75XF
<input type="checkbox"/>				161103100320	4	121314152021	Siddiq Abu	01900000000	-1	mail.bakkar@gmail.com	\$2y\$10\$JLvtdu3i2B3.8PAzhSije0t
<input type="checkbox"/>				161103100445	5	080910141516	Satabdi Rani	01600000000	-1	satabdi@gmail.com	\$2y\$10\$HPexSK3MCoA7gqCti5Cd
<input type="checkbox"/>				161103100509	5	111213171819	Debi Rani	01600000000	-1	satabdi@gmail.com	\$2y\$10\$lgwloy8zuri/v509PhriD4em
<input type="checkbox"/>				161103100537	5	121314152021	Rani Satabdi	01600000000	-1	satabdi@gmail.com	\$2y\$10\$C/GMI50UmPeQBQT3kXz
<input type="checkbox"/>				161103100723	6	080910141516	Mashrur Maskab	01500000000	-1	mashrur@gmail.com	\$2y\$10\$S0BuzF8BAJ2r98DrEBeb
<input type="checkbox"/>				161103100747	6	111213171819	Maskab Utsash	01500000000	-1	mashrur@gmail.com	\$2y\$10\$SiRat3U27vR40xgVBxobNM
<input type="checkbox"/>				161103100809	6	121314152021	Utsash Mashrur	01900000000	-1	mashrur@gmail.com	\$2y\$10\$WXI1t80mfRhAJkJWsKwl

Figure2.18: Doctor Table

This table contains all the information about the doctor. A registered patient will be saved here and for the doctor, it is manual because of the doctor verification. Doctor register system can make a possibility of a fake doctor, so only certified doctor will give service here. In this table patient have a different id from others, so the doctor can easily find the patient provided information by searching. For security issue, all the password are get encrypted so that no one can change the password except the user. Email should be on email format otherwise, the system will show the error that email is wrong. Doctor is separated by their department id, each id defines different specialty.

	id	total	dept	time_slot	t00	t01	t02	t03	t04	t05	t06	t07	t08	t09	t10	t11	t12	t13	t14	t15	t16	t17	
<input type="checkbox"/> Edit Copy Delete	160820032654	0	1	080910141516	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	160821121720	0	1	111213171819	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103095631	0	2	121314152021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103095752	0	2	080910141516	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103095817	0	2	111213171819	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103095858	0	3	080910141516	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103095922	0	3	111213171819	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100035	0	4	080910141516	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100115	0	4	111213171819	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100320	0	4	121314152021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100445	0	5	080910141516	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100509	0	5	111213171819	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100537	0	5	121314152021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100723	0	6	080910141516	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100747	0	6	111213171819	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	161103100809	0	6	121314152021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.19: Schedule Table

On this table, for each doctor, there is a 24-time slot. Initially, this time slot is 0 if a doctor gets an appointment then the patient id will be set on their automatically. It is the table for a whole day. Time difference between the slots is 1 hour. This table contains all the information about scheduling, which doctor have the appointment, when the appointment scheduled and who appointed the doctor.

CSP (Constraint Satisfaction Problem):

Generally, CSP Algorithm has more power than a standard search algorithm. Constraints specifying allowable combinations of values for subsets of variables. Backtracking search is the basic uninformed algorithm for CSP, Depth-first search for CSPs with single-variable assignments is called backtracking search.

The Backtracking algorithm can be easily extended to backtrack to the conflicting variable and, thus, to incorporate some form of look-back scheme or intelligent backtracking. Nevertheless, this adds some additional expenses to the algorithm and it seems that preventing possible future conflicts is more reasonable than recovering from those [4].

```

function BACKTRACKING-SEARCH(csp) returns a solution, or failure
  return RECURSIVE-BACKTRACKING({}, csp)

function RECURSIVE-BACKTRACKING(assignment, csp) returns a solution, or
failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(Variables[csp], assignment, csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
    if value is consistent with assignment according to Constraints[csp] then
      add { var = value } to assignment
      result ← RECURSIVE-BACKTRACKING(assignment, csp)
      if result ≠ failure then return result
      remove { var = value } from assignment
  return failure

```

Figure 2.20: Algorithm for Backtracking

Forward checking is the easiest way to prevent future conflicts. Instead of performing arc consistency to the instantiated variables, it performs restricted form of arc consistency to the not yet instantiated variables. We speak about restricted arc consistency because forward checking checks only the constraints between the current variable and the future variables. When a value is assigned to the current variable, any value in the domain of a "future" variable which conflicts with this assignment is temporarily removed from the domain. The advantage of this is that if the domain of a future variable becomes empty, it is known immediately that the current partial solution is inconsistent. Forward checking allows branches of the search tree that will lead to failure to be pruned earlier than with simple backtracking. Note that whenever a new variable is considered, all its remaining values are guaranteed to be consistent with the past variables, so the checking an assignment against the past assignments is no longer necessary.

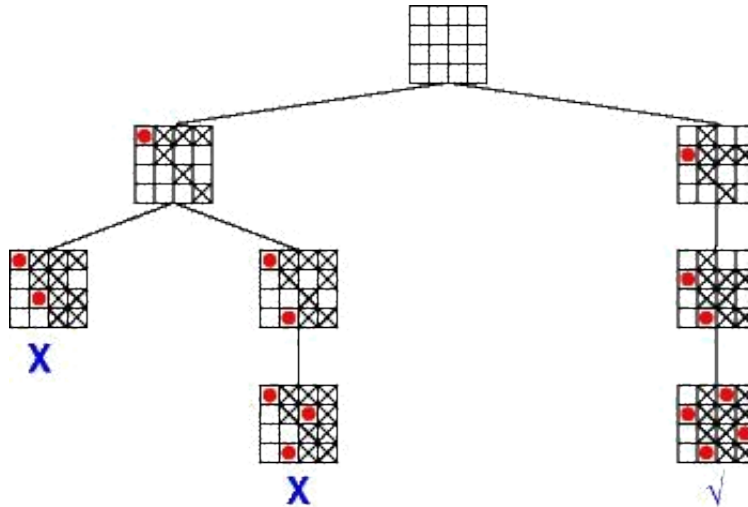


Figure 2.21: Forward Checking Diagram

Forward checking detects the inconsistency earlier than simple backtracking and thus it allows branches of the search tree that will lead to failure to be pruned earlier than with simple backtracking. This reduces the search tree and the overall amount of work done. But it should be noted that forward checking does more work when each assignment is added to the current partial solution.

To apply CSP:

- States with unsatisfied constraints
- Randomly select any conflicted variable
- Pick value that violates the fewest constraints
- Total number of violated constraints

In CSP states defined by values of a fixed set of variables and goal test defined by constraints on variable values. Variable ordering and value selection heuristics help significantly. Forward checking prevents assignments that guarantee probable future failure. Constraint propagation does additional work to constrain values and detect inconsistencies.

Reason for using CSP:

In our project, we tried to implement a system which will appoint a doctor for a patient based on the doctor's specialty, selected date of the patient, the doctor's free schedule and best possible time for the patient. It contains multiple constraints to search simultaneously and need backtracking for the best possible way.

Code Implementation:

Firstly it fetches the list of doctors from the database then sorts them by the total number of patients.

```
<?php
/**
 *Fetches list of doctors from database
 *sort them by total number of patients each have
 *@returns array
 **/
function appoint(){
    $db = new database();
    $conn = $db->db_connect();
    $table = sprintf("t%02d%02d",$_POST['day'],$_POST['month'],$_POST['year']);

    $sql = $conn->prepare("select * from ".$table." where dept=:dept order by total");
    $sql->bindParam(":dept",$_POST['dept']);

    $sql->execute();
    $result = $sql->fetchAll();
    $conn = null;

    $str[0] = $table;
    $str[1] = $result;
    return $str;
}
```

Figure 2.22: Code for Fetch

Then it balances the doctor's busy schedule from individual specialty, the number of patients each doctor has and free slots. It chooses the best possible slot for the patient and appoints it.


```

/**
 *CSP Algo for load balance
 *chose time for patient
 *returns array (converted time,time_slot,doctor id)
 */
function setTime($table, $array){
    $db = new database();
    $conn = $db->db_connect();

    foreach ($array as $key => $value) {
        $column = 100;
        $colName = "";
        $d = "";

        for($i=0; $i<strlen($value['time_slot']); $i+=2){
            $time[$i] = $value['time_slot'][$i].$value['time_slot'][$i+1];
        }
        foreach ($time as $val) {
            $col = "t".$val;
            if(strlen($value[$col])<$column){
                $column = strlen($value[$col]);
                $colName = $col;
                $d = $value['id'];
            }
        }

        if($column<35){
            $sql = $conn->prepare("update $table set total=total+1,
                $colName=concat($colName,:p_id) where id=:id");
            $sql->bindParam(':p_id',$_SESSION['user_id']);
            $sql->bindParam(':id',$d);
            if($sql->execute()){
                $conn = null;
                $info[0] = convert_time($val);
                $info[1] = $val;
                $info[2] = $value['id'];
                return $info;
            }
        }else{
            $info = null;
        }
    }

    return $info;
}

```

Figure 2.23: Code for Load Balance and Scheduling

Implementation of User Interface:

To develop a standard user interface we compute HTML, CSS and JavaScript code and merged them.

HTML:

Hypertext Markup Language (HTML) is a simple markup language used to create hypertext documents that are platform independent. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of domains [7]. HTML markup can represent hypertext news, mail, documentation, and hypermedia, menus of options, database query results and simple structured documents with in-line graphics and hypertext views of existing bodies of information [6].

CSS:

Cascading Style Sheets (CSS) is the language for describing the presentation of Web s, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across s, and tailor s to different environments [8].

JavaScript:

JavaScript is Netscape's cross-platform, object-oriented scripting language. Core JavaScript contains a core set of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects [11]. Client-side JavaScript extends the core language by supplying objects to control a browser and its Document

Object Model and Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server.

Implemented:

Register is for a new user who wants to use the service. All data from registration will be saved in the database and for each patient, there will be specific account. User login is for registered users who can log in their account and use its features.

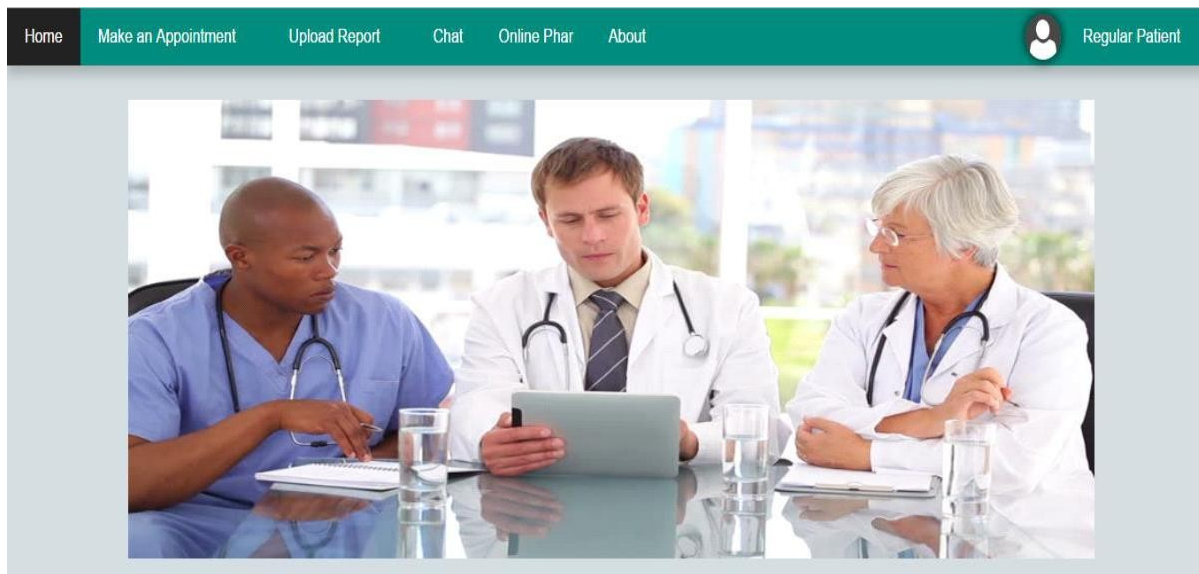


Figure 2.24: Home

This is the home of the website. It contains a navigation bar which will help the user to use the system perfectly. The “Home” bar will redirect the user home wherever he is, “Make an Appointment” bar will help him to appoint a doctor. By “Upload report” user can upload his treatment report for the doctor. By “chat” he can communicate with the doctor on scheduled time. “Online Phar” will redirect him to the pharmacy store where he can buy doctor prescribed

Make An Appointment

Today
Thu-aug-2019
01-08-2019

07:15.04

Select Department Select On ▾

Appoint Date 01-Dec-20

01/07/2019

02/07/2019

03/07/2019

04/07/2019

05/07/2019

06/07/2019

07/07/2019

08/07/2019

09/07/2019

10/07/2019

11/07/2019

Make an Appointment

Figure 2.25: Appoint Doctor

This is for making a new appointment. By this , the user can select specialty of doctor and expected appointment date. After selecting the option the user can press “Make an Appointment” button which will create an appointment for the user.

New Appointment		
From Old Record	Department	Time
<input checked="" type="radio"/> none		
<input type="radio"/>	Patient	//
<input type="radio"/>	ENT	01/08/2019
<input type="radio"/>	ENT	01/08/2019
<input type="radio"/>	Nutrition	01/08/2019
<input type="radio"/>	Nutrition	01/08/2019
<input type="radio"/>	Nutrition	01/08/2019
<input type="radio"/>	Cardiology	01/08/2019
<input type="radio"/>	Cardiology	01/08/2019
<input type="radio"/>	ENT	01/08/2019

Figure 2.26: Appointed Record

The user can also make an appointment from his/her previous appointment history. It will give privileges to appoint the previously appointed doctor. By this , the user can see his/her total appointment history, appointment date, and doctor specialty. To appoint a previous doctor the user has to check the checkbox and click the “Make Appointment” button.

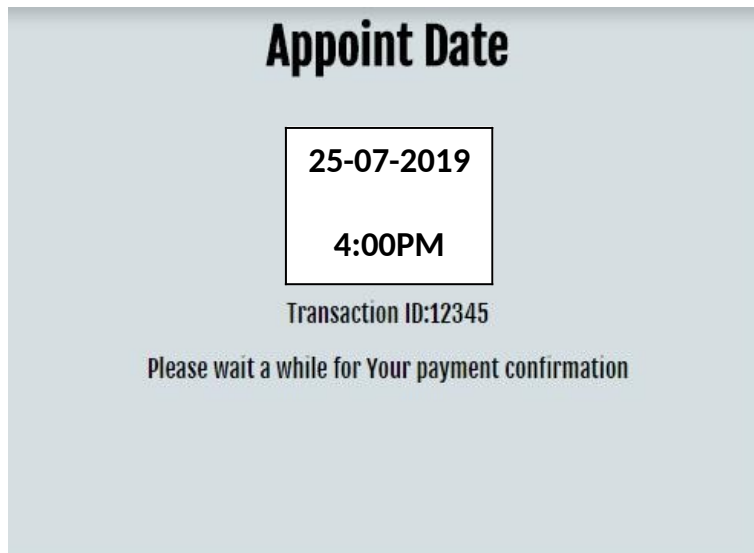


Figure 2.27: Scheduled Appointment

By clicking “Make Appointment” button user will successfully make an appointment. This will show the user his/her appointment confirmation, date and scheduled time. This will directly show to the user when the user confirms his payment. Otherwise, an appointment will not be fixed.

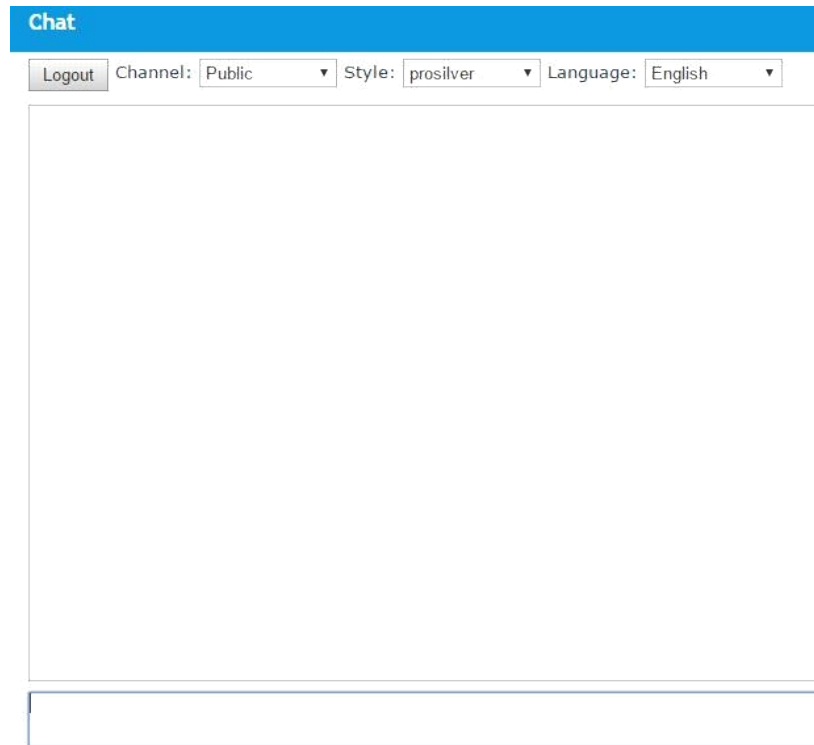


Figure 2.28: Chat Server

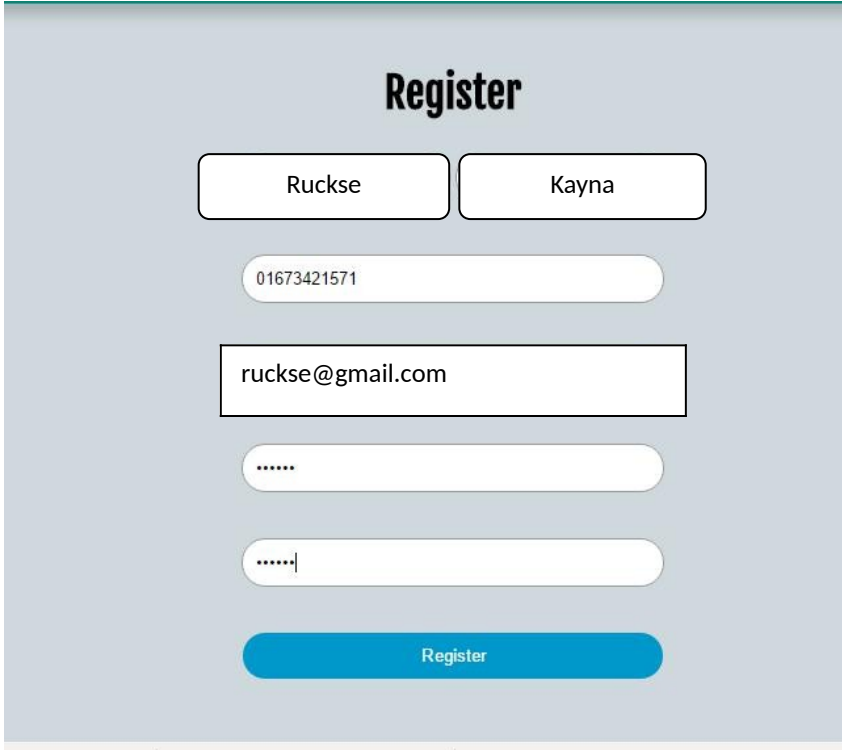
On the appointed date and time the user can contact the doctor a through chat server. The doctor can create room for an individual patient and can consult them one on one. The doctor can also prescribe the patient through this chat server. This chat server is only for doctor-patient consultation.

Chapter 3

Result & Analysis

Here our goal was to implement a system which will help a patient to appoint a doctor in terms of criteria, free time and date based on dynamic scheduling. Our implemented software fulfill all this section perfectly for the patient and save his/her valuable time. On this process user can visit our website and register to get our service and then login into the website. The patient can make an appointment on his selected date. The patient also can appoint the previous doctor for the treatment process.

3.1 Register & Login



The image shows a registration form with the following elements:

- Title:** Register
- Buttons:** Two buttons labeled "Ruckse" and "Kayna".
- Phone Number:** A text input field containing "01673421571".
- Email:** A text input field containing "ruckse@gmail.com".
- Password:** A text input field with masked characters ".....".
- Confirm Password:** A text input field with masked characters ".....|".
- Action:** A blue button labeled "Register".

Figure 3.1: Register

Here the user can give the important required information and register for an account. The system will take the information and create an account for the patient and make a space in the database.



The image shows a user login interface on a light gray background. At the top, the text "User Login" is displayed in a large, bold, black font. Below this, there is a white rectangular input field containing the email address "ruckse@gmail.com". Underneath the email field is a white rounded rectangular input field for a password, with the first five characters obscured by dots. Below the password field is a blue rounded rectangular button with the text "Log In" in white. At the bottom of the form, the text "or Register for a new Account" is displayed in a smaller font, with "or" in black and "Register" in purple.

Figure 3.1: User Login

After registration, the user has to log in to access his account and use the facilities of the system. The user has to log in with the given email address and password which he provided during the registration.

3.2 Edit Profile:

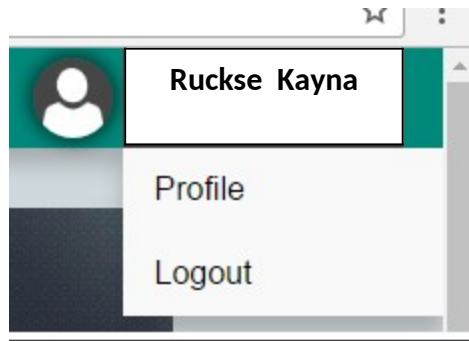


Figure 3.2: User Profile

After log in system will show his name in the corner and there is a dropdown option which will show her profile edit option and log out option.

A screenshot of the "Edit Profile" form. On the left, there is a placeholder image of a woman in a red dress carrying a briefcase. Below the image is a "Choose File" button and a "No file chosen" message. Below that is a "Change Image" button. On the right, there are four text input fields: "First Name:" with "Ruckse", "Last Name:" with "Kayna", "Phone:" with "0167325588", and "e-mail:" with "ruckse@gmail.com". Below the "e-mail" field is an "Update" button.

Figure 3.2: Edit Profile

Here the user can edit his profile by giving information and there is an option of upload picture of the user. The user can upload his picture through this option and update the profile.

3.3 Updated Profile:

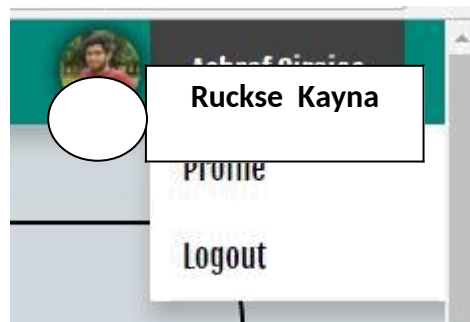


Figure 3.3: Updated Profile

After the update, the profile system will show the user the updated profile.

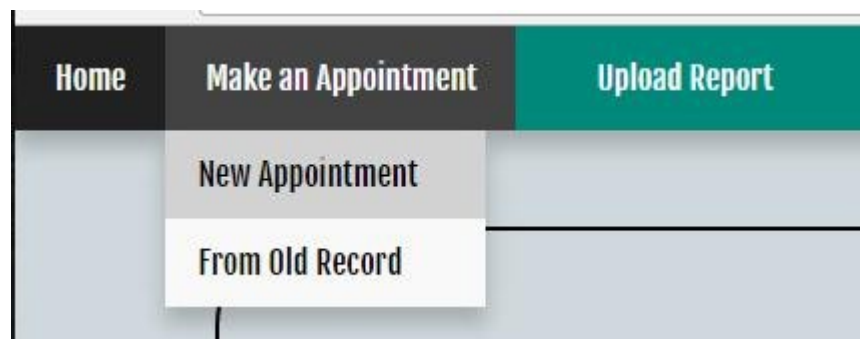


Figure 3.3: Appointment Criteria

After login user can see his home and navigation bar of the website. There is an option called “Make an Appointment” which will show user two option “New Appointment” and “From Old Record”.

3.4 Appointment System:

Make An Appointment

Today
Thu-aug-2019
01-08-2019
07:15.04

Select Department

Day Month Year

Figure 3.4: Appointment System

By choosing “Make An Appointment” option user will redirect to appointment by the system. Here the user can select the specialty of a doctor and date for an appointment. After choosing the option user to press the “Make an Appointment” button. This button will tell the system that schedules a doctor based on selected department and date. The system will search the database and find a doctor on that date then system schedule the time for the patient. A total process running in the background after a user press makes an appointment button.

Appoint Date

25-07-2019
4:00PM

Transaction ID:12345
Please wait a while for Your payment confirmation

Figure 3.4: Appointed Schedule

After clicking the button system will show the user the scheduled time and date. This process is running by following the algorithm process.

The screenshot shows a web interface for scheduling appointments. At the top, there is a 'New Appointment' header. Below it, a dropdown menu is open, showing 'From Old Record' as the selected option. The main content area contains a table with three columns: 'Department', 'Patient', and 'Time'. The 'Time' column is highlighted with a black border. The table lists several appointments, all scheduled for 01/08/2019. The departments listed are ENT, Nutrition, and Cardiology.

Department	Patient	Time
		//
ENT		01/08/2019
ENT		01/08/2019
Nutrition		01/08/2019
Nutrition		01/08/2019
Nutrition		01/08/2019
Cardiology		01/08/2019
Cardiology		01/08/2019
ENT		01/08/2019
		01/08/2019

Figure 3.4: Old Record

By selecting “From Old Record” patient can appoint the previous doctor and get an appointment with his previous consultant to continue his treatment.

3.5 Chat System:

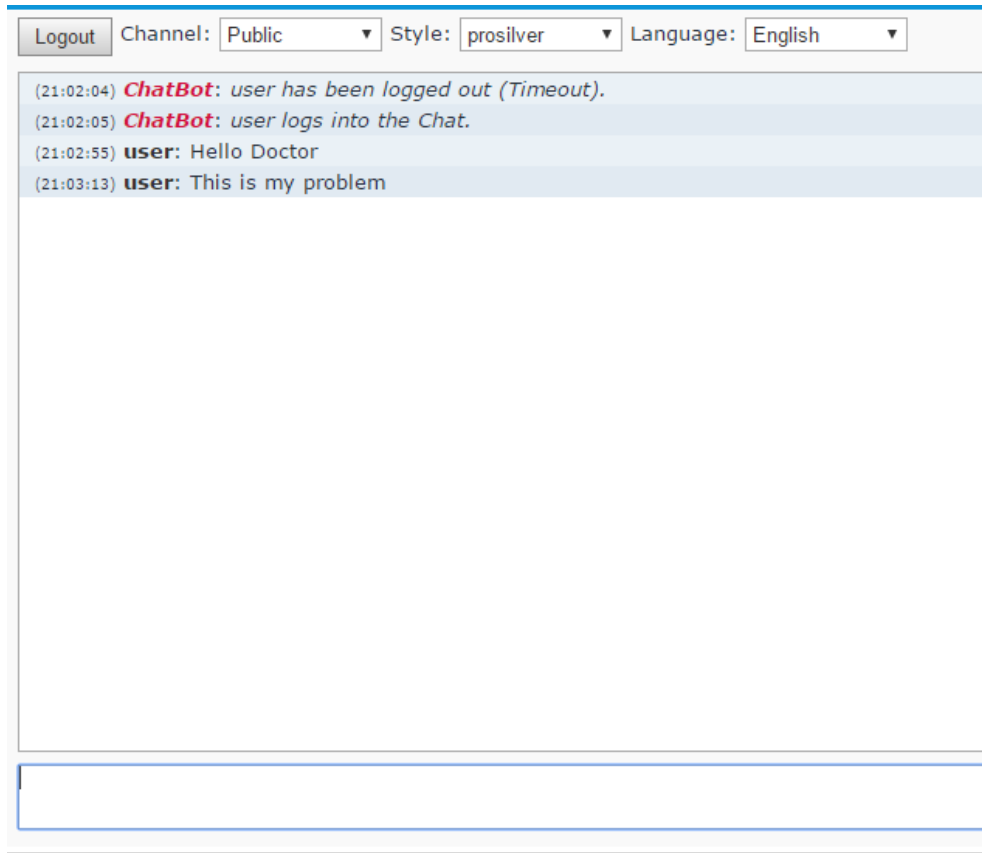


Figure 3.5: Chat System

On the scheduled time user can consult with a doctor using this chat server. Doctor can create a room patient and chat one on one. By this system, the doctor can prescribe medicine and suggest another specialist.

3.6 Prescription:

The form is titled "Prescribe Medicine" and is divided into three main sections:

- Patient Detail:** Contains four input fields for Name, Age, E-mail, and Phone.
- Doctor Detail:** Contains four input fields for Name, Speciality, E-mail, and Phone.
- Prescription:** Contains two input fields for Name of disease and Name of Medicine.

Figure 3.6: Prescription

This is the prescription , and here doctor will give the advice and medicine and how to take them. After giving the details of the medicine doctor can upload this prescription for patient and patient can download it.

Chapter 4

Conclusion and Future Work

Conclusion:

We started the project keeping in mind to solve one of the very old issues of the whole world which are the availability of medical support. It is not possible to change the medical support system overnight. Our goal was to provide a solution so that this problem could be minimized and provide medical support to every corner even the remotest area where people are unable to reach a doctor when needed. In order to eliminate the difficulties of scheduling, we came up with dynamic scheduling system which will also eradicate the waiting for a doctor physically in front of the chamber. We achieved our goal which is this system can balance the load among doctors and schedule a patient to a specified specialist and at the scheduled time the patient can communicate with the assigned doctor. We believed our motive to develop this project can reduce the unnecessary hassles of traveling all the way to the doctor and save their valuable time which can eventually prevent the patient from suffering further.

Future Work

We have future improvement plan regarding this project. As it is an online based system, we intend to make this system a global platform where users from all over the world will be able to consult a specialist of their interest. We also have the plan of developing a mobile application to access this system in order make it easier to use. We also have the plan for free checkup campaign by the available doctors once a month.

References:

- [1] Unlocked, M. Zack (13 December 2005). "[Google Runs MySQL](#)". *The Open Force*. M. Zack Urlocker. Retrieved 3 August 2010. AdWords was built using the MySQL database
- [2] Helmsley, Richard (June 30, 2010), "[GNU HURD: Altered visions and lost promise](#)", *The H* (online ed.), p. [3](#), Nearly twenty years later the HURD has still to reach maturity, and has never achieved production quality. [...] Some of us are still wishing and hoping for the real deal, a GNU operating system with a GNU kernel.
- [3.] Cai, Jin-Yi; Chen, Xi (2012). *Complexity of counting CSP with complex weights*. pp. 909–920. [arXiv:1111.2384](#) . [doi:10.1145/2213977.2214059](#). ISBN 978-1-4503-1245-5.
- [4.] Miguel, Ian (July 2001). *Dynamic Flexible Constraint Satisfaction and its Application to AI Planning (Ph.D. thesis)*. *University of Edinburgh School of Informatics*. [Cutesier 10.1.1.9.6733](#). [hdl:1842/326](#).
- [5.] Berners-Lee, Tim; Connelly, Daniel (November 1995). "[RFC 1866 – Hypertext Markup Language – 2.0](#)
- [6.] Manu Sporny (2013-01-26). "[DRM in HTML5](#)". *The Beautiful, Tormented Machine*. Manu Sporny. Archived from [the original](#) on 2014-04-25. Retrieved 2014-05-16.
- [7.] Raggett, Dave (28 March 1995). "[Hypertext Markup Language Specification Version 3.0](#)". *HTML 3.0 Internet Draft Expires in six months*. *World Wide Web Consortium*. Retrieved 17 June 2010.
- [8.] Meyer, Eric A. (2006). [Cascading Style Sheets: The Definitive Guide](#) (3rd ed.). O'Reilly Media, Inc. ISBN 0-596-52733-0.
- [9.] Lie, Hakon W (10 Oct 1994). "[Cascading HTML style sheets - a proposal](#)" (Proposal)(0.92). CERN. Retrieved 25 May 2014.
- [10.] Meyer, Eric A. (2006). [Cascading Style Sheets: The Definitive Guide](#) (3rd ed.). O'Reilly Media, Inc. ISBN 0-596-52733-0.

- [11.] Haverbeke, Marijn (2011). *Eloquent JavaScript*. No Starch Press. pp. 139–149. [ISBN 978-1-59327-282-1](#).
- [12.] Mediati, Nick (July 7, 2009). "[Google Announces Chrome OS](#)". PC World. Retrieved July 8, 2009.
- [13.] Conn, Samuel S. "OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures for Real Time Data Analysis." Regis University School for Professional Studies. Accessed Oct. 16, 2014.
- [14.] Golemon, Sara (2006). *Extending and Embedding PHP*. [ISBN 978-0-672-32704-9](#).
- [15.] Festa, Paul (May 6, 2003). "[Mozilla's Firebird gets wings clipped](#)". CNET. Retrieved January 30, 2007.
- [16.] Ullman, Chris (March 2007). *Beginning Ajax*. wrox. [ISBN 978-0-470-10675-4](#). [Archived](#) from the original on 5 July 2008. Retrieved 24 June 2008.
- [17.] Jacobson Ivar, Christerson Magnus, Jonsson Patrik, Övergaard Gunnar, *Object-Oriented Software Engineering - A Use Case Driven Approach*, Addison-Wesley, 1992.
- [18.] Suzanne Robertson. *Scenarios in Requirements Discovery*. Chapter 3 in Alexander and Maiden, 2004. s 39-59.
- [19.] Beynon-Davies, Paul (2004). *Database Systems*. Basingstoke, UK: Palgrave: Houndmills. [ISBN 1403916012](#).
- [20.] A.P.G. Brown, "Modelling a Real-World System and Designing a Schema to Represent It", in Douque and Nijssen (eds.), *Data Base Description*, North-Holland, 1975, [ISBN 0-7204-2833-5](#).
- [21.] Samek, Miro (2008). *Practical UML Statecharts in C/C++, Second Edition: Event-Driven Programming for Embedded Systems*. Newnes. p. 728. [ISBN 978-0-7506-8706-5](#).
- [22.] [Taylor Booth](#) (1967) *Sequential Machines and Automata Theory*, John Wiley and Sons, New York.
- [23.] Projects Geek, Online Doctor System project in Java, Aug 29, 2014

[24.] Grameenphone launches Health Information & Service, from
Grameenphone.com, November 02, 2006

[25.] <https://doctorola.com>, 2016 DOCTOROLA LTD

[26.] <https://www.webhealthcenter.com>, (2016, 15 July)

[27.] <https://pinkwhalehealthcare.com>, (2016, 15 July)

[28.] <https://www.bdhealth.com>, (2016, 15 July)

Appendix

Code for Doctor's index

```
<?php
session_start();
include("include/config.php");
if(isset($_POST['submit']))
{
$ret=mysql_query("SELECT * FROM doctors WHERE docEmail='".$_POST['username']."' and
password='".$_md5($_POST['password'])."'");
$num=mysql_fetch_array($ret);
if($num>0)
{
$extra="dashboard.php";
$_SESSION['dlogin']=$_POST['username'];
$_SESSION['id']=$num['id'];
$uip=$_SERVER['REMOTE_ADDR'];
$status=1;
$log=mysql_query("insert into doctorslog(uid,username,userip,status) values('".$_SESSION['id'].','.$_SESSION['dlogin'].','.$uip','$status)");
$host=$_SERVER['HTTP_HOST'];
$uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\');
header("location:http://$host$uri/$extra");
exit();
}
else
{
$host = $_SERVER['HTTP_HOST'];
$_SESSION['dlogin']=$_POST['username'];
$uip=$_SERVER['REMOTE_ADDR'];
$status=0;
mysql_query("insert into doctorslog(username,userip,status) values('".$_SESSION['dlogin'].','.$uip','$status)");
$_SESSION['errmsg']="Invalid username or password";
$extra="index.php";
$uri =rtrim(dirname($_SERVER['PHP_SELF']),'/\');
header("location:http://$host$uri/$extra");
exit();
}
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Doctor Login</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0,
minimum-scale=1.0, maximum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <meta content="" name="description" />
```

```

<meta content="" name="author" />
<link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|
Raleway:300,400,500,600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
<link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
<link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
<link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
<link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
<link rel="stylesheet" href="assets/css/styles.css">
<link rel="stylesheet" href="assets/css/plugins.css">
<link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />
</head>
<body class="login">
  <div class="row">
    <div class="main-login col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2 col-md-4 col-
md-offset-4">
      <div class="logo margin-top-30">
        <h2> HMS | Doctor Login</h2>
      </div>
      <div class="box-login">
        <form class="form-login" method="post">
          <fieldset>
            <legend>
              Sign in to your account
            </legend>
            <p>
              Please enter your name and password to
              <span style="color:red;"><?php echo
              $_SESSION['errmsg']; ?><?php echo $_SESSION['errmsg']="";?></span>
            </p>
            <div class="form-group">
              <span class="input-icon">
                <input type="text" class="form-
                control" name="username" placeholder="Username">
                <i class="fafa-user"></i></span>
              </div>
            <div class="form-group form-actions">
              <span class="input-icon">
                <input type="password"
                class="form-control password" name="password" placeholder="Password">
                <i class="fafa-lock"></i>
              </span>
            </div>
            <div class="form-actions">
              <button type="submit" class="btn btn-
              primary pull-right" name="submit">
                Login <i class="fafa-arrow-circle-
              right"></i>
            </button>
          </div>
          </fieldset>
        </form>

```

```

        <div class="copyright">
            &copy; <span class="current-year"></span><span
class="text-bold text-uppercase">Smart Paitent System</span>. <span>All rights reserved</span>
        </div>

    </div>

</div>
</div>
<scriptsrc="vendor/jquery/jquery.min.js"></script>
<scriptsrc="vendor/bootstrap/js/bootstrap.min.js"></script>
<scriptsrc="vendor/modernizr/modernizr.js"></script>
<scriptsrc="vendor/jquery-cookie/jquery.cookie.js"></script>
<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<scriptsrc="vendor/switchery/switchery.min.js"></script>
<scriptsrc="vendor/jquery-validation/jquery.validate.min.js"></script>

<scriptsrc="assets/js/main.js"></script>

<scriptsrc="assets/js/login.js"></script>
<script>
    jQuery(document).ready(function() {
        Main.init();
        Login.init();
    });
</script>

</body>
<!-- end: BODY -->
</html>

```

Code for Admin's Index:

```

<?php
session_start();
error_reporting(0);
include("include/config.php");
if(isset($_POST['submit']))
{
$ret=mysql_query("SELECT * FROM admin WHERE username='".$_POST['username']."' and password='".$_POST['password'].\"");
$num=mysql_fetch_array($ret);
if($num>0)
{
$extra="dashboard.php";//
$_SESSION['login']=$_POST['username'];
$_SESSION['id']=$num['id'];
$host=$_SERVER['HTTP_HOST'];
$uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\');
header("location:http://$host$uri/$extra");
exit();
}
else
{
$_SESSION['errmsg']="Invalid username or password";

```

```

$extra="index.php";
$host = $_SERVER['HTTP_HOST'];
$uri =rtrim(dirname($_SERVER['PHP_SELF'],'/\'));
header("location:http://$host$uri/$extra");
exit();
}
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Admin-Login</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0,
minimum-scale=1.0, maximum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <meta content="" name="description" />
    <meta content="" name="author" />
    <link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|
Raleway:300,400,500,600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
    <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
    <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
    <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
    <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
    <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
    <link rel="stylesheet" href="assets/css/styles.css">
    <link rel="stylesheet" href="assets/css/plugins.css">
    <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />
  </head>
  <body class="login">
    <div class="row">
      <div class="main-login col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2 col-md-4 col-
md-offset-4">
        <div class="logo margin-top-30">
          <h2>Admin Login</h2>
        </div>
        <div class="box-login">
          <form class="form-login" method="post">
            <fieldset>
              <legend>
                Sign in to your account
              </legend>
              <p>
                Please enter your name and password to
                log in.<br />
                <span style="color:red;"><?php echo
                htmlentities($_SESSION['errmsg']); ?><?php echo htmlentities($_SESSION['errmsg']="");?></span>
              </p>
              <div class="form-group">
                <span class="input-icon">
                  <input type="text" class="form-
                  control" name="username" placeholder="Username">
                  <i class="fafa-user"></i></span>

```

```

        </div>
        <div class="form-group form-actions">
            <span class="input-icon">
                <input type="password"
class="form-control password" name="password" placeholder="Password"><i class="fafa-lock"></i>
            </span>
        </div>
        <div class="form-actions">
            <button type="submit" class="btn btn-
primary pull-right" name="submit">
                Login <i class="fafa-arrow-circle-
right"></i>
        </button>
    </div>
</fieldset>
</form>
<div class="copyright">
    &copy; <span class="current-year"></span><span
class="text-bold text-uppercase"> HMS</span>. <span>All rights reserved</span>
</div>
</div>
</div>
</div>
<scriptsrc="vendor/jquery/jquery.min.js"></script>
<scriptsrc="vendor/bootstrap/js/bootstrap.min.js"></script>
<scriptsrc="vendor/modernizr/modernizr.js"></script>
<scriptsrc="vendor/jquery-cookie/jquery.cookie.js"></script>
<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<scriptsrc="vendor/switchery/switchery.min.js"></script>
<scriptsrc="vendor/jquery-validation/jquery.validate.min.js"></script>
<scriptsrc="assets/js/main.js"></script>
<scriptsrc="assets/js/login.js"></script>
<script>
    jQuery(document).ready(function() {
        Main.init();
        Login.init();
    });
</script>
</body>
<!-- end: BODY -->
</html>

```

Code for User Login:

```

<?php
session_start();
error_reporting(0);
include("include/config.php");

```

```

if(isset($_POST['submit']))
{
$ret=mysql_query("SELECT * FROM users WHERE email='".$_POST['username']."' and
password='".md5($_POST['password'])."'");
$num=mysql_fetch_array($ret);
if($num>0)
{
$extra="dashboard.php";//
$_SESSION['login']=$_POST['username'];
$_SESSION['id']=$num['id'];
$host=$_SERVER['HTTP_HOST'];
$uip=$_SERVER['REMOTE_ADDR'];
$status=1;
$log=mysql_query("insert into userlog(uid,username,userip,status) values('".$_SESSION['id']."',".$_
$_SESSION['login']."', '$uip', '$status')");
$uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\');
header("location:http://$host$uri/$extra");
exit();
}
else
{
$_SESSION['login']=$_POST['username'];
$uip=$_SERVER['REMOTE_ADDR'];
$status=0;
mysql_query("insert into userlog(username,userip,status) values('".$_SESSION['login']."', '$uip', '$status')");
$_SESSION['errmsg']="Invalid username or password";
$extra="user-login.php";
$host = $_SERVER['HTTP_HOST'];
$uri =rtrim(dirname($_SERVER['PHP_SELF']),'/\');
header("location:http://$host$uri/$extra");
exit();
}
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>User-Login</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0,
minimum-scale=1.0, maximum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <meta content="" name="description" />
    <meta content="" name="author" />
    <link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|
Raleway:300,400,500,600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
    <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
    <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
    <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
    <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
    <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
    <link rel="stylesheet" href="assets/css/styles.css">
    <link rel="stylesheet" href="assets/css/plugins.css">
    <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />

```



```

</head>
<body class="login">
  <div class="row">
    <div class="main-login col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2 col-md-4 col-
md-offset-4">
      <div class="logo margin-top-30">
        <h2>Smart Paitent System</h2>
        <h2>Patient Login</h2>
      </div>
      <div class="box-login">
        <form class="form-login" method="post">
          <fieldset>
            <legend>
              Sign in to your account
            </legend>
            <p>
              Please enter your name and password to
              <span style="color:red;"><?php echo
$_SESSION['errmsg']; ?><?php echo $_SESSION['errmsg']="";?></span>
            </p>
            <div class="form-group">
              <span class="input-icon">
                <input type="text" class="form-
control" name="username" placeholder="Username">
                <i class="fafa-user"></i></span>
              </div>
              <div class="form-group form-actions">
                <span class="input-icon">
                  <input type="password"
class="form-control password" name="password" placeholder="Password">
                  <i class="fafa-lock"></i>
                </span>
              </div>
              <div class="form-actions">
                <button type="submit" class="btnbtn-
primary pull-right" name="submit">
                Login <i class="fafa-arrow-circle-
right"></i>
              </button>
            </div>
            <div class="new-account">
              Don't have an account yet?
              <a href="registration.php">
                Create an account
              </a>
            </div>
          </fieldset>
        </form>
        <div class="copyright">
          &copy; <span class="current-year"></span><span
class="text-bold text-uppercase"> Smart Paitent System</span>. <span>All rights reserved</span>
        </div>
      </div>
    </div>
  </div>

```

```
        </div>
    </div>
    <scriptsrc="vendor/jquery/jquery.min.js"></script>
    <scriptsrc="vendor/bootstrap/js/bootstrap.min.js"></script>
    <scriptsrc="vendor/modernizr/modernizr.js"></script>
    <scriptsrc="vendor/jquery-cookie/jquery.cookie.js"></script>
    <script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
    <scriptsrc="vendor/switchery/switchery.min.js"></script>
    <scriptsrc="vendor/jquery-validation/jquery.validate.min.js"></script>

    <scriptsrc="assets/js/main.js"></script>

    <scriptsrc="assets/js/login.js"></script>
    <script>
        jQuery(document).ready(function() {
            Main.init();
            Login.init();
        });
    </script>
</body>
<!-- end: BODY -->
</html>
```

THE END