Prescription Management System

Submitted By

Afshar Miah

ID: 2013-3-96-015

Submitted To

Dr. Taskeed Jabid

Assistant Professor

Department of Computer Science & Engineering

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of

M. Sc in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

EAST WEST UNIVERSITY, DHAKA

May 10, 2015

Abstract

The importance of a computerized prescription has been recognized by doctors and patients in the last few years. Doctors usually write prescription by hands. This way doctors have been habituated to write prescription for last few decades. Now technology is being using everywhere and it makes the nature of our life easier and flexible too. Therefore, doctor requires a prescription which will be provided by computer for the patient and it will be delivered with a unique identity number. doctor stores all data belongs patient in the local database before printing the prescription. As a result, doctor can see previous prescription of a patient if requires by simply typing the identity number and can provide another prescription with the same identity number. If requires doctors can easily edit, delete and update the details of a patient. Another important function is also available in the system that patient can communicate with their doctor just like sending email. doctors will receive the message and reply any query. To do this patient just need to open an account with the valid information. Patient can ask for any query related with the prescription. Besides, patient can take appointment by directly contacting with the doctor.

Declarations

This is to certify that this project is an original work and was done by me and it has not been submitted elsewhere for the requirement of any degree or diploma or for any other purposes except for publication.

Signature of the Student

Afshar Miah

ID: 2013-3-96-015

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Letter of Acceptance

This is project entitled Prescription management System submitted by Afshar Miah, ID 2013-3-96-015 to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted as satisfactory for partial fulfilled of the requirements for the degree of M. Sc in Computer Science and Engineering on April25,2015.

Board of Examiners

Dr. Taskeed Jabid

Assistant Professor, Department of Computer Science and Engineering

East West University

Chairperson

Dr. Shamim Hasnat Ripon

Associate Professor and chairperson

Department of Computer Science and Engineering

East West University

Acknowledgement

This project would not be possible without the help of my project supervisor, Dr. Taskeed Jabid, Assistant professor, Department of Computer Science and Engineering, East West University. He makes all the issues clear to me. I would like to express my sincere and deep regards to him.

TABLE OF CONTENTS

List of Figures

Chapter1

Introduction

1.1 A Problem Scenario

Dr. Abdul H Chowdhury is a heart-disease specialist. He is a senior surgeon of heart disease at Dhaka medical college hospital. He has been working as a heart surgeon for five years in that hospital. as he is working in a public hospital therefore he has to meet with hundreds of patient every day. he is doing that task very carefully. patient comes and Dr. Abdul H Chowdhury prescribe them. the traditional way to prescribe a patient is to write the details of a patient in a pad then write tests name the patient requires and finally list of medicine with the quantity of a medicine and specific indication how long have to continue with it. Besides doctor has to write few advices which help the patient to overcome the disease. Doctor do these task by hands. To complete these task one by one doctor needs around ten minutes to write the prescription. Ten minutes is not a short time for a patient really. before prescribe doctor has to spend another five to ten minutes to understand the problem of a patient. As a result, important minutes goes vain. besides, there has been added a new problem. Dr. abdul H chowdhury noticed that majority of the patient forget to bring the previous prescription. if ask the patient they replies, I came here three months ago how can I preserve the prescription for a long time even its a single page. someone replies I forgot to bring the prescription with me. As a consequences, doctor needs to ask the patient to tell the medicine names which he prescribed last time. sometimes doctor prescribe without asking anything to the patient because in most cases patient can't say the name of medicines.

In this scenario what can be a solution? a prescription management system can be provided for Dr. Abdul H Chowdhury. So that he can easily take the information of patient, selecting few easy combo box for tests and medicine can make his prescription nice. Thus, doctor can save his valuable time. Doctor need not remember anything. everything such as tests name and medicines list will be displayed in the screen. doctor just need to select the required tests and medicines. Once doctor receive the patient details and select tests and medicine he can store the prescription information in the local database. Thus, doctor see previous prescription of a patient if the patient forget to bring the prescription with them in the next meet or if the patient lost the prescription.

1.2 Traditional Recommended System

Traditional recommended system of prescription management system is to write the prescription by hand. doctor has been doing that since the early age of prescribing a patient. when patient visit a doctor, he bring the paper in front of him and pen. then listen the problem of a patient. based on the patient condition doctor write the names of test and medicine in the paper. he writes patients details on the paper.

1.3 prescription management System

The goal is to construct an management system to assist a doctor in prescribing for the patient. thus doctor can provide computer printed prescription. doctor can save the information of patients. therefore, doctor can see information of any patient from the database. doctor can also prescribe second or many times by maintaining unique id of a patient. every time same patient will not get new unique id. only the first time visiting patient will get unique id. then doctor will use the same id for prescribing a patient for second time or many times as usual.

if the patient forget to bring the prescription in their second visit they just need to say the unique id to the doctor. doctor can see patients prescription by searching by id in the database. after that doctor need not generate new unique id for the existing patient instead doctor need to add new patient. it will simply bring the details of a patient by his id. patient details will be displayed on the page. doctor just need to select new test if requires and medicines with their quantity and date.

In many cases, patient feels that he needs to contact with the doctor. contacting with doctor means take the appointment again and visit doctors chamber. its time consuming really. This system aim is to provide a system such a way that patient can easily communicate with their doctor. they can ask any question related with their disease or medicine. it will works like email. they just need to type the problem or any query and send to the doctor. After that doctor will receive a notification for new mail. thus, doctor can communicate with their patient and if necessary doctor give important information to the patient. In this way patient can also request for appointment. if the appointment is available doctor can provide a serial.

There is another system has been developed for the patient. In this system patient can view his prescription online. this is very easy system for the patient. suppose, farida akther is a patient. she visited doctors chamber few days ago. Accidently she lost her prescription. that's why can't continue having medicine regularly. even she can't buy new medicine as she forgot the names of medicines. thus she can't continue her medicine. to solve this problem system has been developed. in this System patient has access to the list of all prescription. as we developed a system for the doctor such kind of system will be available for the patient too. by typing patient id a patient can easily view his/her prescription anytime. if he/she requires can print the prescription. there is one dissimilarity with the system of doctor's. we have seen that doctor can edit, delete any prescription he thinks. but in case of patient, patient will not be able to edit any previous prescription neither can delete any prescription. patient will see only his/her prescription. a patient will not be able to see any ones prescription. only the prescription he searched will be visible to him/her.

1.4 Motivation

Doctor can provide computer printed prescription. doctor can save the information of patients. therefore, doctor can see information of any patient from the database. doctor can also prescribe second or many times by maintaining unique id of a patient. every time same patient will not get new unique id. only the first time visiting patient will get unique id. then doctor will use the same id for prescribing a patient for second time or many times as usual.

if the patient forget to bring the prescription in their second visit they just need to say the unique id to the doctor. doctor can see patients prescription by searching by id in the database. after that doctor need not generate new unique id for the existing patient instead doctor need to add new patient. it will simply bring the details of a patient by his id. patient details will be displayed on the page. doctor just need to select new test if requires and medicines with their quantity and date.

patient feels that he needs to contact with the doctor. contacting with doctor means take the appointment again and visit doctors chamber. its time consuming really. This system aim is to provide a system such a way that patient can easily communicate with their doctor. they can ask any question related with their disease or medicine. it will works like email. they just need to type the problem or any query and send to the doctor. After that doctor will receive a notification for new mail. thus, doctor can communicate with their patient and if necessary doctor give important information to the patient. In this way patient can also request for appointment. if the appointment is available doctor can provide a serial. Dr. Abdul H Chowdhury or hundreds of doctor will be benefited by this system. it will be easy to operate and provide nice computer printed prescription and doctor can save the prescription as well as.

There is another system has been developed for the patient. In this system patient can view his prescription online. this is very easy system for the patient. suppose, farida akther is a patient. she visited doctors chamber few days ago. Accidently she lost her prescription. that's why can't continue having medicine regularly. even she can't buy new medicine as she forgot the names of medicines. thus she can't continue her medicine. to solve this problem system has been developed. in this System patient has access to the list of all prescription. as we developed a system for the doctor such kind of system will be available for the patient too. by typing patient id a patient can easily view his/her prescription anytime. if he/she requires can print the prescription. there is one dissimilarity with the system of doctor's. we have seen that doctor can edit, delete any prescription he thinks. but in case of patient, patient will not be able to edit any previous prescription neither can delete any prescription. patient will see only his/her prescription. a patient will not be able to see any ones prescription. only the prescription he searched will be visible to him/her.

1.5Objectives

1. There are some objectives of prescription management system. these are following
2. Doctor can provide Computer printed prescription to the patient.
3. Doctor can save the details of a patient included with tests, medicine etc.
4. If required doctor can view any prescription from the database.
5. Every patient will be given a unique patient id.
6. In case of prescription lost or forget to bring in the next visit doctor can handle the situation.

7. Doctor can maintain database of patients. if required doctor can search any patients prescription by his patient id.
8. Doctor need not provide different patient id for a patient in the next meet. he can easily update the existing prescription.
9. Patient can communicate with the doctor. For any query patient can get information from the doctor.
10. Patient can also receive appointment by simply sending email to the doctor.

1.6Contribution

My system is not like the other prescription system, it is also different from the regular prescription system. Regular prescription system does not have the way to store the information of a patient. regular prescription system can't provide unique id for a patient. if patient lost his prescription system can't help the doctor. by the regular prescription system doctor can communicate with the patient and patient also can't contacts with the doctor for any query related with the prescription or can't ask for an appointment.

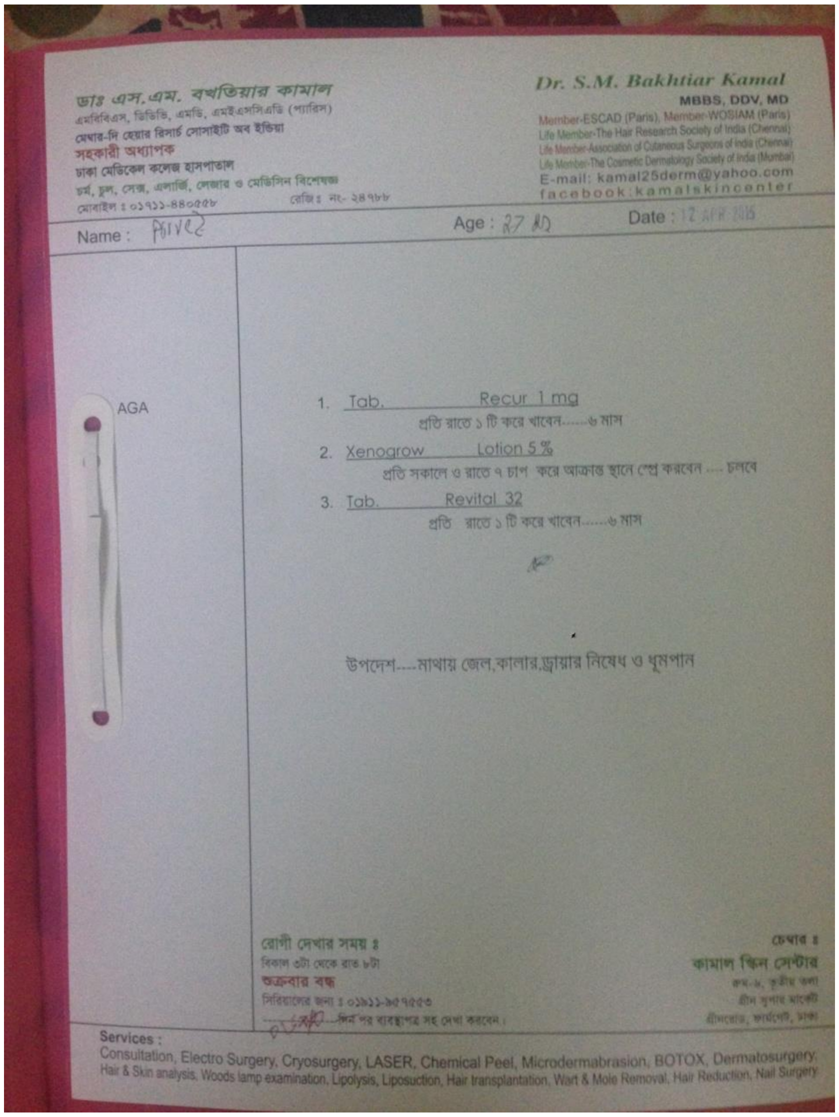
1.7 Report Outline

Chapter 2: In this chapter we will see summary and review of some papers which is related to this project. Finally we will present our work in this project and discuss the difference between our sytem and the other.
Chapter 3: Our proposed model will describe in this chapter. It will explain our work in this project step by step. There will be the source codes of this project end of this chapter.
Chapter 4: This chapter will talk about the evaluation of the project. It will show usability study.
We will taste the system by some patient. We will take opinion from the patient. We will consider their feedback and base on the feedback we will discuss how we can make this system better.
Chapter 5: It will be the conclusion part of this paper. In this section we will also discuss the achievement from this project and the future work with this project to develop it more.

Chapter 2

Survey of Existing Model

It has been observed that existing systems were useful but in cases it was not so much effective for the doctor. Some things can be mention. doctor could not update any information from the existing material. doctor also could not manage all the patient at a time. doctor also can't view patients information. besides, doctor was not able to view previous prescription of a patient. the whole system was static. doctor could not update, delete, edit anything. To solve these problem an advanced system was demanded to the doctor since few years.



ডাঃ এস.এম. বখতিয়ার কামাল
এমবিবিএস, ডিডিভি, এমডি, এমইএসসিএডি (প্যারিস)
মেম্বার-দি হেয়ার রিসার্চ সোসাইটি অব ইন্ডিয়া
সহকারী অধ্যাপক
ঢাকা মেডিকেল কলেজ হাসপাতাল
চর্ম, চুল, সেক্স, এলার্জি, লেজার ও মেডিসিন বিশেষজ্ঞ
মোবাইল : ০১৭১১-৪৪০৫৫৮ রেজিঃ নং- ২৪৭৮৮

Dr. S.M. Bakhtiar Kamal
MBBS, DDV, MD
Member-ESCAD (Paris), Member-WOSIAM (Paris)
Life Member-The Hair Research Society of India (Chennai)
Life Member-Association of Cutaneous Surgeons of India (Chennai)
Life Member-The Cosmetic Dermatology Society of India (Mumbai)
E-mail: kamal25derm@yahoo.com
facebook:kamalskincenter

Name : Parvez Age : 27 yrs Date : 12 APR 2015

AGA

1. Tab. Recur 1 mg
প্রতি রাতে ১ টি করে খাবেন......৬ মাস
2. Xenogrow Lotion 5 %
প্রতি সকালে ও রাতে ৭ চাপ করে আক্রান্ত স্থানে স্প্রে করবেন .... চলবে
3. Tab. Revital 32
প্রতি রাতে ১ টি করে খাবেন......৬ মাস

উপদেশ----মাথায় জেল,কালার,ড্রায়ার নিষেধ ও ধূমপান

রোগী দেখার সময় :
বিকাল ৩টা থেকে রাত ৮টা
শুক্রবার বন্ধ
সিরিয়ালের জন্য : ০১৯১১-৯৫৭৫৫৩
৩ মাস দিন পর ব্যবস্থাপত্র সহ দেখা করবেন।

চেম্বার :
কামাল স্কিন সেন্টার
রুম-৮, তৃতীয় তলা
গ্রীন সুপার মার্কেট
গ্রীনরোড, ফার্মগেট, ঢাকা

Services :
Consultation, Electro Surgery, Cryosurgery, LASER, Chemical Peel, Microdermabrasion, BOTOX, Dermatosurgery, Hair & Skin analysis, Woods lamp examination, Lipolysis, Liposuction, Hair transplantation, Wart & Mole Removal, Hair Reduction, Nail Surgery

Figure: Existing software management system

Name	Do you think existing systems design was nice?	Do you think existing systems was informative?	Was it useful?	Are you satisfied?
Ruman Hossain	Yes	No	Yes	No
Selina Sharmin	Yes	No	No	No
Ruksan begum	Yes	No	No	Yes
Imam Hossain	Yes	No	No	No
Kasha Micah	Yes	No	No	Yes

I have asked few persons to observed the existing system. and make some comments how can it be improve. maximum person said existing system was little useful for the doctor patient. although there was not any implementation for the patient. once patient got prescription his job is totally done. patient could not have right access to the system. patient could not view his patient in case of prescription lost or forgotten. Therefore, majority of the people suggests that existing system need to be developed more to help the doctor as well as patient.

Chapter 3

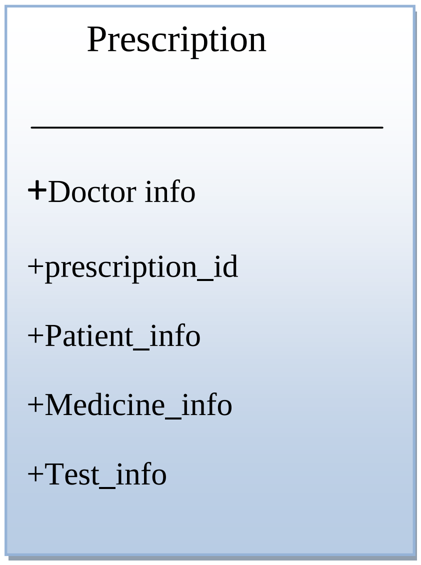
Proposed Model

Most of the doctor do not want to provide prescription by handwritten. they prefer computer printed prescription. thus they can easily prescribe to the patient. in this way doctor can save their time. proposed system will be very easy to handle for the doctor. doctor need not to be expert on computer. if they know how to use a computer or have basic of a computer then they can handle the system. this system requires only two or three minutes to provide a computer printed prescription for the patient. doctor can save the prescription for further requirement. if required doctor can view from the database.

3.1 Model Structure

3.1.1 Class Diagram

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed



In the diagram, classes are represented with boxes which contain three parts:

- The top part contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle part contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom part contains the methods the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

In order to further describe the behavior of systems, these class diagrams can be complemented by a state diagram or UML state machine.

3.1.2 Purpose of Class Diagram

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.

So the purpose of the class diagram can be summarized as:

1. Analysis and design of the static view of an application.
2. Describe responsibilities of a system.
3. Base for component and deployment diagrams.
4. Forward and reverse engineering.

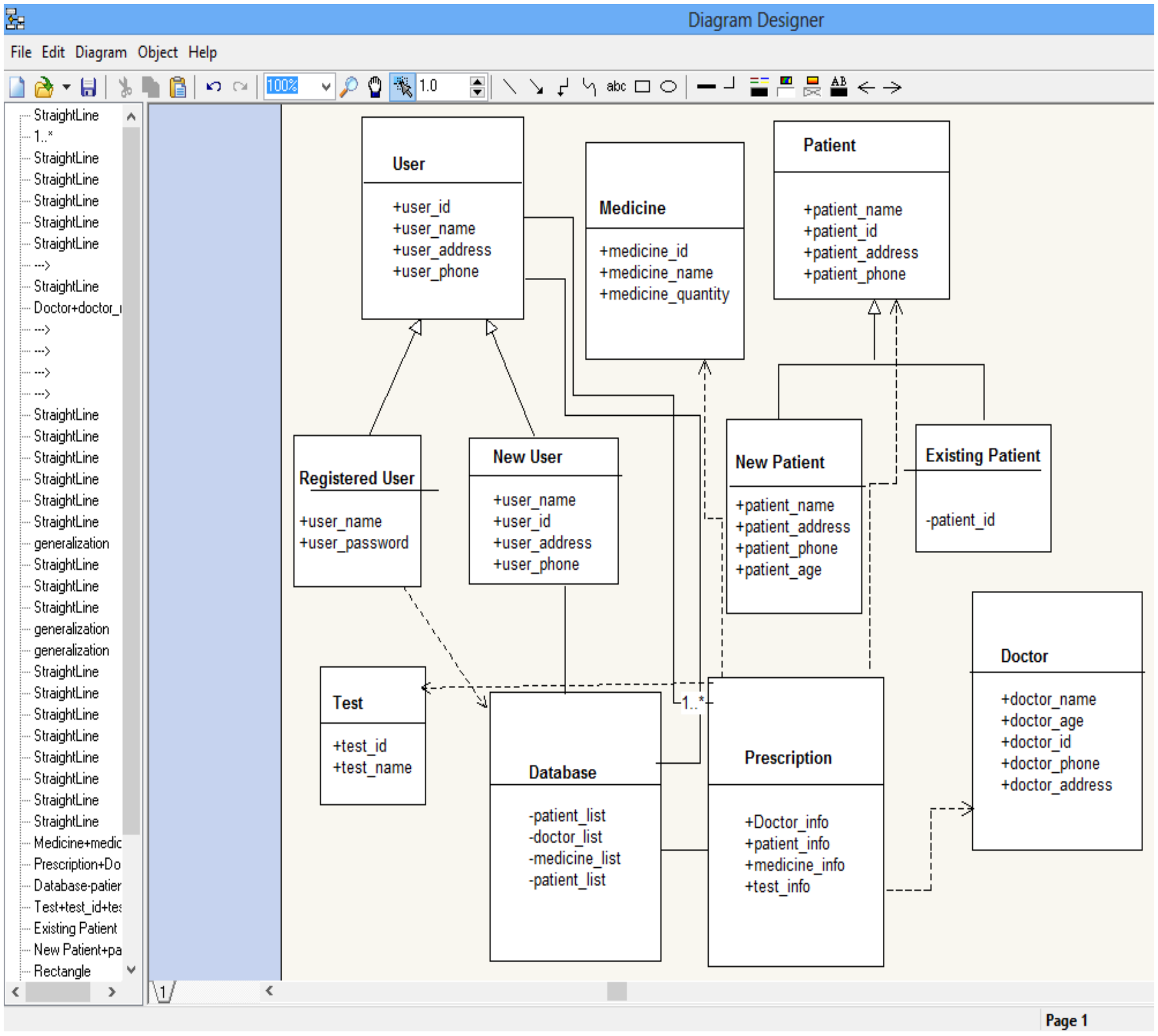


Figure 3.1: Class diagram of my Proposed Model.

3.1.3 Use Case Diagram

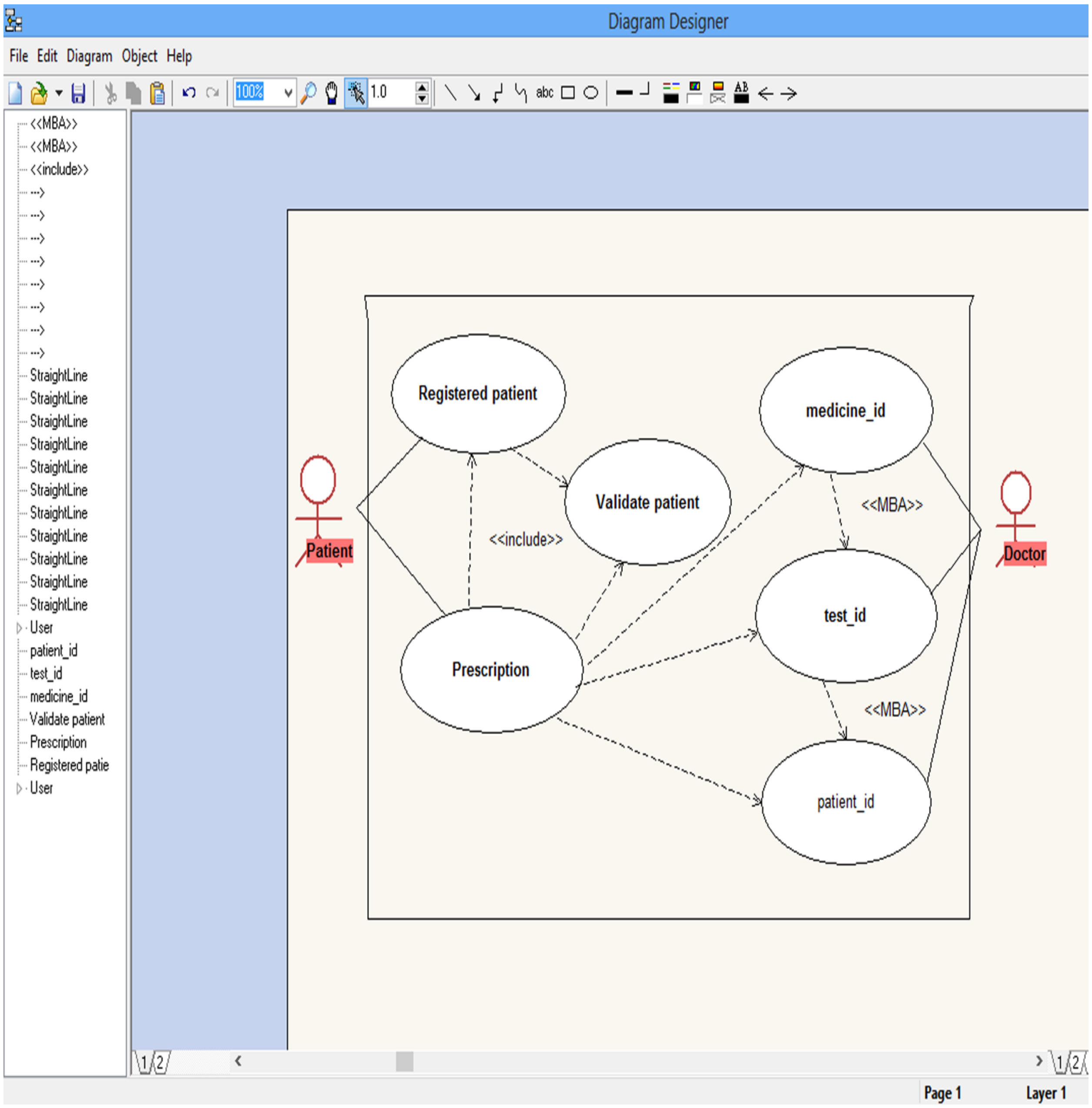
A use case diagram is a graphic depiction of the interactions among the elements of a system. A Use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components. The boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which are the specific roles played by the actors within and around the system. The relationships between and among the actors and the use cases.

3.1.4 Purpose of Use Case Diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. Now when the initial task is complete use case diagrams are modeled to present the outside view.
So in brief, the purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system. `
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.



\

Figure: Use case diagram of prescription management system

3.1.5 State diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events, that could occur in one or more possible states. Hereby "each diagram usually represents objects of a single class and track the different states of its objects through the system".

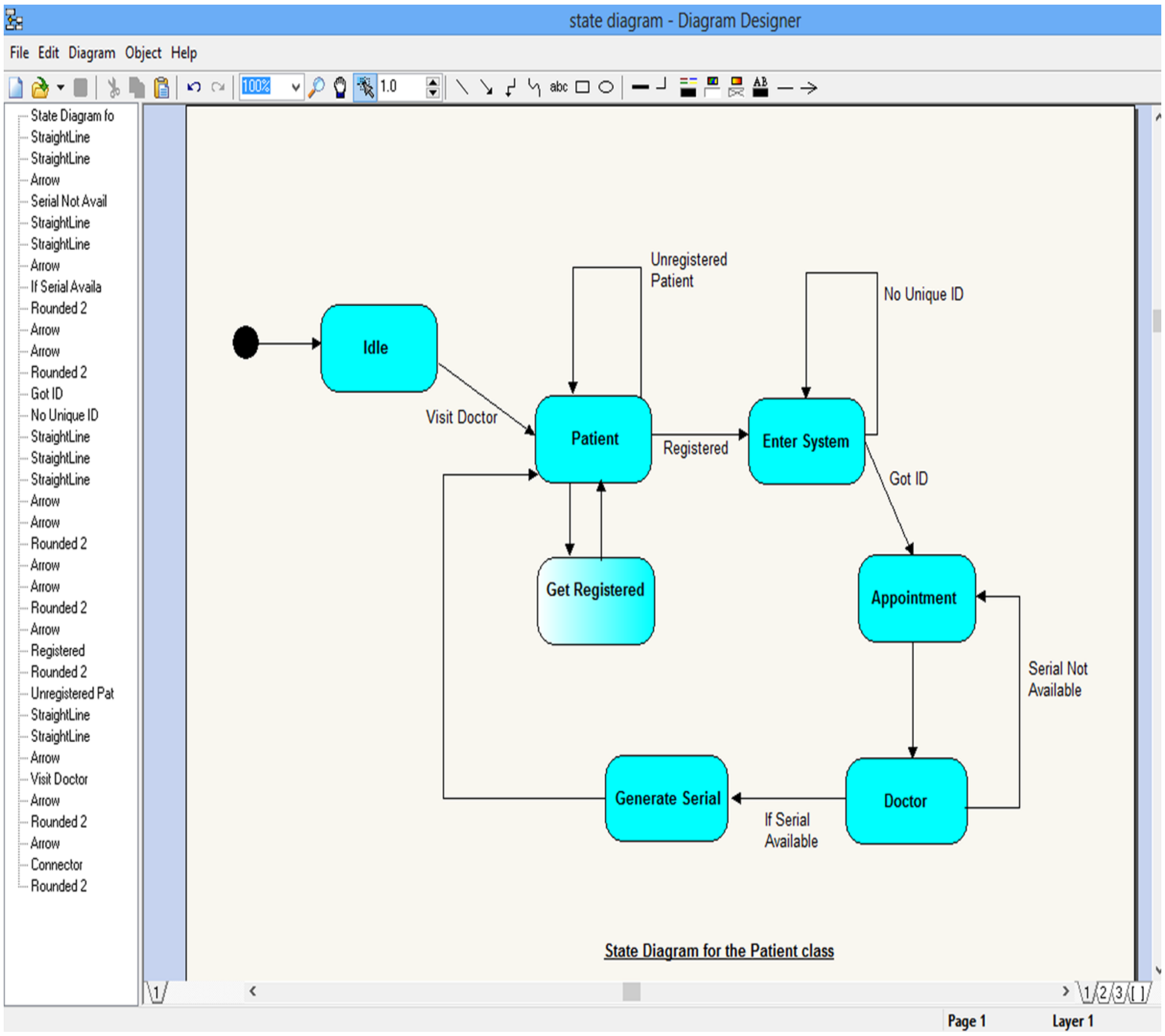
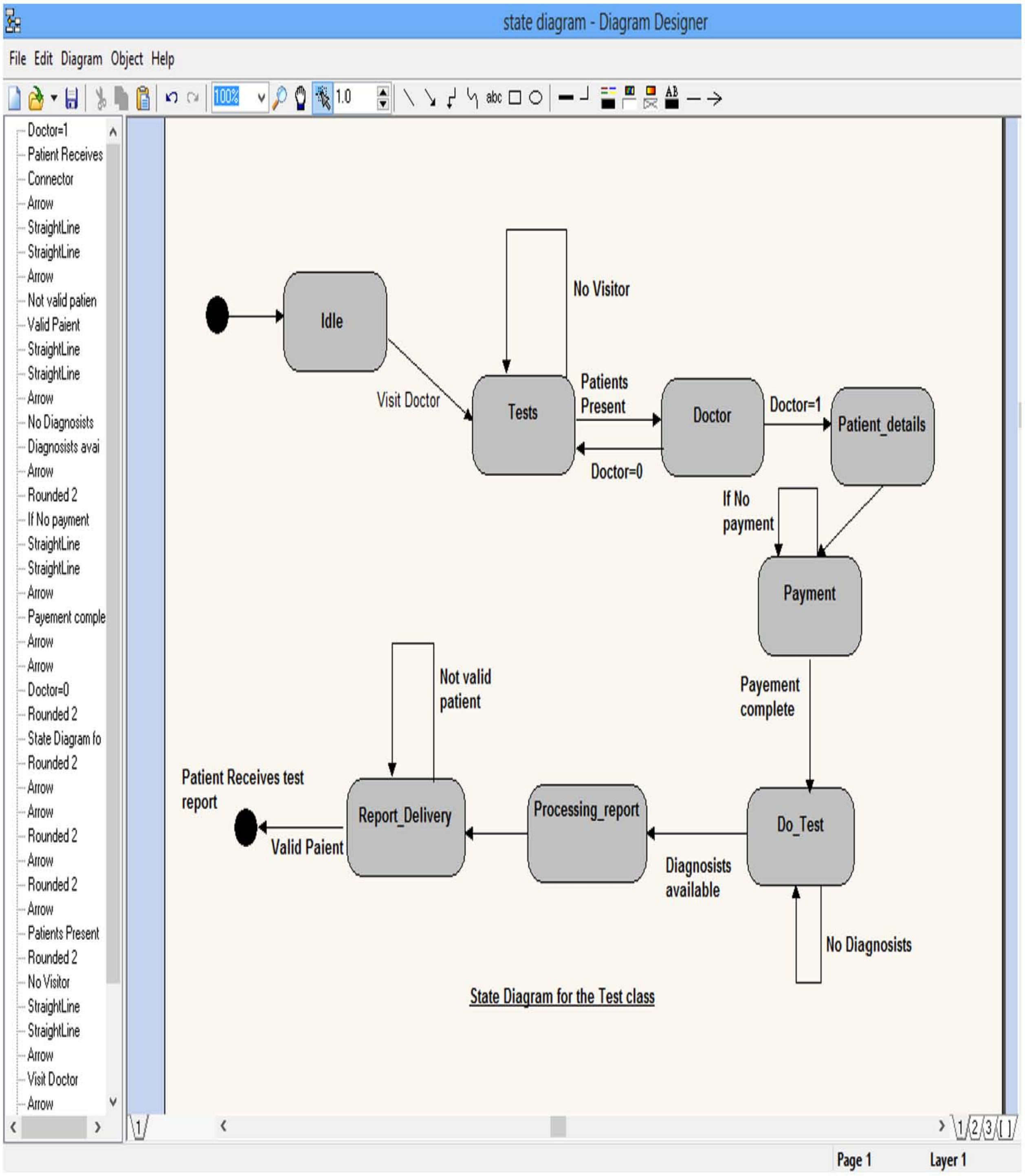


Figure: State diagram for the class of patient


state diagram - Diagram Designer
File Edit Diagram Object Help
100%
1.0
Doctor=1
Patient Receives
Connector
Arrow
StraightLine
StraightLine
Arrow
Not valid patien
Valid Paient
StraightLine
StraightLine
Arrow
No Diagnosists
Diagnosists avai
Arrow
Rounded 2
If No payment
StraightLine
StraightLine
Arrow
Payement comple
Arrow
Arrow
Doctor=0
Rounded 2
State Diagram fo
Rounded 2
Arrow
Arrow
Rounded 2
Arrow
Rounded 2
Arrow
Patients Present
Rounded 2
No Visitor
StraightLine
StraightLine
Arrow
Visit Doctor
Arrow
Idle
No Visitor
Visit Doctor
Tests
Patients Present
Doctor
Doctor=1
Patient_details
Doctor=0
If No payment
Payment
Not valid patient
Payement complete
Patient Receives test report
Report_Delivery
Processing_report
Do_Test
Valid Paient
Diagnosists available
No Diagnosists
State Diagram for the Test class
Page 1
Layer 1

3.2 Few fields of the System

In this project we use four required fields for this system. They are-

- Registered patient
- Doctor's Registration
- Prescriptions form
- Doctor's database

3.2.1 Registered patient

There are two ways in the system. patient should be either registered before or patient will be registered later by doctor while doctor prescribe the patient. when patient visit a doctors chamber doctor collect all the information of the patient. give input of a patients information in the input field of the system. after that doctor select tests, medicine for the patient and finally print the prescription for the patient. before printing the prescription doctor save the prescription into the database. every patient is given a unique identity number. that has no chance to match with other patient. in his next meet if the patient can't remember his patient id or forget to bring the prescription then doctor can looks into the database based on either mobile number or name. Thus all patient become registered as a patient into the system.

3.2.2 Doctor's Registration

There is a way to be registered in the system for the doctor. To get access in the system doctor's first need to be registered. Once doctor is registered he can easily get access the system by providing real username and password. Doctor then will see few pages that is allocated for the doctors only. Patient will not be able to see doctors pages like patients info, patients tests, medicines, advices etc .besides, patients can't see the database of doctors. doctor can only see the database and handle the database. if requires doctor can modify, edit, delete the patients info.

3.2.3 Prescription form

Basically doctor write the prescription on the paper. doctor write patients name, age, address, phone number on the paper. then ask the problem of a patient. based on the problem doctor suggests few tests and doctor write all the tests in the paper by hands, then write medicines that patients have daily after or before the meal. eventually, doctor write few advices for the patients which patients should be follow with having medicines properly. Thus, doctor usually write prescription. In my system there is a nice prescription form. doctor first ask the details of a patient like name, age, address, phone number of a patient. doctor will give input in the data field. after that doctor will click on next button. there will be list of tests. once doctor select the required tests the doctor will be directed to the medicine page. there are numerous medicine for the patients.

3.2.4 Doctor's database

There is a database in the system. when doctor prescribe a patient doctor receive all the information of the patients. doctor put those in the system and select tests and medicine and advices. all of these are saved in the database. only doctor can see the database. he can update a patients information or medicine or tests. if requires he can edit and can delete too. patients can't get access in the system. only doctor can see the database.

3.3 Strategies of this Model

When a patient is new in the system he needs to registered first. once he get the username and password then he can get access in the system. there are few pages for the patients only. after successful sign up patients can log in the system easily. then patients will see few pages that is allocated for him. patients can communicate with the doctor by simply sending email. patients can attached picture of his previous prescription if needed.

Chapter 4

User Manual

4.1 Framework

In Computer programming a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software. A software framework is a universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate development of software applications, products and solutions. Software frameworks may include support programs, compilers, code libraries, tool sets, and application programming interfaces(APIs) that bring together all the different component to enable development of a project or solution. Frameworks contain key distinguishing features that separate them from normal libraries: Inversion of control: In a framework, unlike in libraries or normal user applications, the overall program's flow of control is not dictated by the caller, but by the framework. default behavior: A framework has a default behavior. This default behavior must be some useful behavior and not a series of no-ops.

Extensibility: A framework can be extended by the user usually by selective overriding or specialized by user code to provide specific functionality.

non-modifiable framework code: The framework code, in general, is not supposed to be modified, while accepting user-implemented extensions. In other words, users can extend the framework, but should not modify its code.

The designers of software frameworks aim to facilitate software development by allowing designers and programmers to devote their time to meeting software requirements rather than dealing with the more standard low-level details of providing a working system, thereby reducing overall development time. For example, a team using a web application framework to develop a banking web-site can focus on writing code particular to banking rather than the mechanics of request handling and state management.

Frameworks often add to the size of programs, a phenomenon termed "code bloat". Due to customer-demand driven applications needs, both competing and complementary frameworks sometimes end up in a product. Further, due to the complexity of their APIs, the intended reduction in overall development time may not be achieved due to the need to spend additional time learning to use the framework; this criticism is clearly valid when a special or new framework is first encountered by development staff. If such a framework is not used in subsequent job tasking, the time invested in learning the framework can cost more than purpose-written code familiar to the project's staff; many programmers keep copies of useful boilerplate for common needs. However, once a framework is learned, future projects can be faster and easier to complete; the concept of a framework is to make a one-size-fits-all solution set, and with familiarity, code production should logically rise. There are no such claims made about the size of the code eventually bundled with the output product, nor its relative efficiency and conciseness. Using any library solution necessarily pulls in extras and unused extraneous assets unless the software is a compiler-object linker making a tight (small, wholly controlled, and specified) executable module. The issue continues, but a decade-plus of industry experience has shown that the most effective frameworks turn out to be those that evolve from re-factoring the common code of the enterprise, instead of using a generic "one-size-fits-all" framework developed by third parties for general purposes. An example of that would be how the user interface in such an application package as an office suite grows to have common look, feel, and data-sharing attributes and methods, as the once disparate bundled applications grow unified into a suite which is tighter and smaller; the newer/evolved suite can be a product that shares integral utility libraries and user interfaces.

This trend in the controversy brings up an important issue about frameworks. Creating a framework that is elegant, versus one that merely solves a problem, is still an art rather than a science. "Software elegance" implies clarity, conciseness, and little waste (extra or extraneous functionality, much of which is user defined). For those frameworks that generate code, for example, "elegance" would imply the creation of code that is clean and comprehensible to a reasonably knowledgeable programmer (and which is therefore readily modifiable), versus one that merely generates correct code. The elegance issue is why relatively few software frameworks have stood the test of time: the best frameworks have been able to evolve gracefully as the underlying technology on which they were built advanced. Even there, having evolved, many such packages will retain legacy capabilities bloating the final software as otherwise replaced methods have been retained in parallel with the newer methods.

4.2 Framework Architecture

According to Preen, software frameworks consist of frozen spots and hot spots. Frozen spots define the overall architecture of a software system, that is to say its basic components and the relationships between them. These remain unchanged (frozen) in any instantiation of the application framework. Hot spots represent those parts where the programmers using the framework add their own code to add the functionality specific to their own project. In an object-oriented environment, a framework consists of abstract and concrete classes. instantiation of such a framework consists of composing and sub classing the existing classes. When developing a concrete software system with a software framework, developers utilize the hot spots according to the specific needs and requirements of the system. Software frameworks rely on the Hollywood principle: "Don't call us, we'll call you." This means that the user-defined classes (for example, new subclasses), receive messages from the predefined framework classes. Developers usually handle this by implementing super class abstract methods.

4.3 Code Igniter

Code Igniter is loosely based on the popular Model-View-Controller development pattern. While controller classes are a necessary part of development under Code Igniter, models and views are optional. Code Igniter is most often noted for its speed when compared to other PHP frameworks. In a critical take on PHP frameworks in general, PHP creator Resumes Leadoff spoke at fresco in August 2008, noting that he liked Code Igniter " because it is faster, lighter and the least like a framework.
Using Code Igniter requiring knowledge of using the object-oriented programming technique in order to be able to use Code Igniter effectively, and to understand what happens when you are using certain features in Code Igniter.

4.4 Code Igniter Classes and methods

What are classes and methods? These are the first concepts you'll be introduced to if you are learning object-oriented programming from most books or online resources. Say you're creating a framework. You'll have different classes for different parts of your framework. One being a "Database Class", one being an "E-mail Class", and so forth. Of course, in this case, the Database Class is like the Code Igniter Database Class, providing a set of ready-made methods for you to use so you don't have to create them yourself in order to execute certain application logic, such as inserting, updating and removing database records quickly, without having to "reinvent the wheel". The methods are what contain the application logic, and the class merely holds many related methods together. And this is exactly how your applications would work with the use of the object-oriented programming techniques.

4.5 How does Code Igniter works

Code Igniter has a very extensive user guide, which is much better than documentation on some other frameworks, such as Cake PHP (which is another PHP framework.

Code Igniter has classes and helpers.

Classes: Classes contain a collection of methods and properties (properties are essentially variables in an object-oriented context).

For example, here is an example using the Database library within Code Igniter:

$this->db->get('users', data);

In any application you make using Code Igniter, your own classes inherit (or extends) the Code Igniter class, and so this is why the $this variable is used, which refers to the current class/object. So to call another method within your class, you would use $this->method name().

Helpers: Helpers contain ordinary PHP functions. Such as the form helper..

4.6 Code Igniter History

The first public version of Code Igniter was released by Elli slab on February 28, 2006, and the latest stable version 3.0.0 was released March 31, 2015. On July 9, 2013,EllisLab announced that it was seeking a new owner for Code Igniter, citing a lack of resources to give the framework the attention they felt it deserved. On October 6, 2014, Elli slab announced that Code Igniter would continue development under the stewardship of the British c Columbia Institute of Technology.

4.7 Code Igniter Functions

Code Igniter uses a few functions for its operation that are globally defined, and are available to you at any point. These do not require loading any libraries or helpers.

Is_ pup ('version _number')

is_ pup () determines of the PHP version being used is greater than the supplied version_ number

```
if(is _ pup ('5.3.0'))
{
$ star=quoted_ printable_ encode($ star)
}
```

Returns Boolean TRUE if the installed version of PHP is equal to or greater than the supplied version number. Returns FALSE if the installed version of PHP is lower than the supplied version number.

is_ really_ writable('path / tm /file')

is_ writable()returns true on windows servers when you really can't write to the file as the OS reports to PHP as false only if the read-only attribute is marked. This function determines if a file is a writable by attempting to write to it first. Generally only recommended on platforms where this information may be unreliable.

```
if(is_ really_ writable('file.txt'))
{
echo "I could write to this if I wanted to";
}
else
{
echo "file is not writable";
}
```

(3) config item ('item _key')

The config library is the preferred way of accessing configuration information, however Config item()can be used to retrieve single keys. see config library documentation for more information

(4) show_error('message'),show_404('page0'),log_message('level'),'message')

(5) set _status _header(code,' test');

(6) remove_ invisible _characters($ star)

(7)html_ escape ($mixed).

Screen short of the system

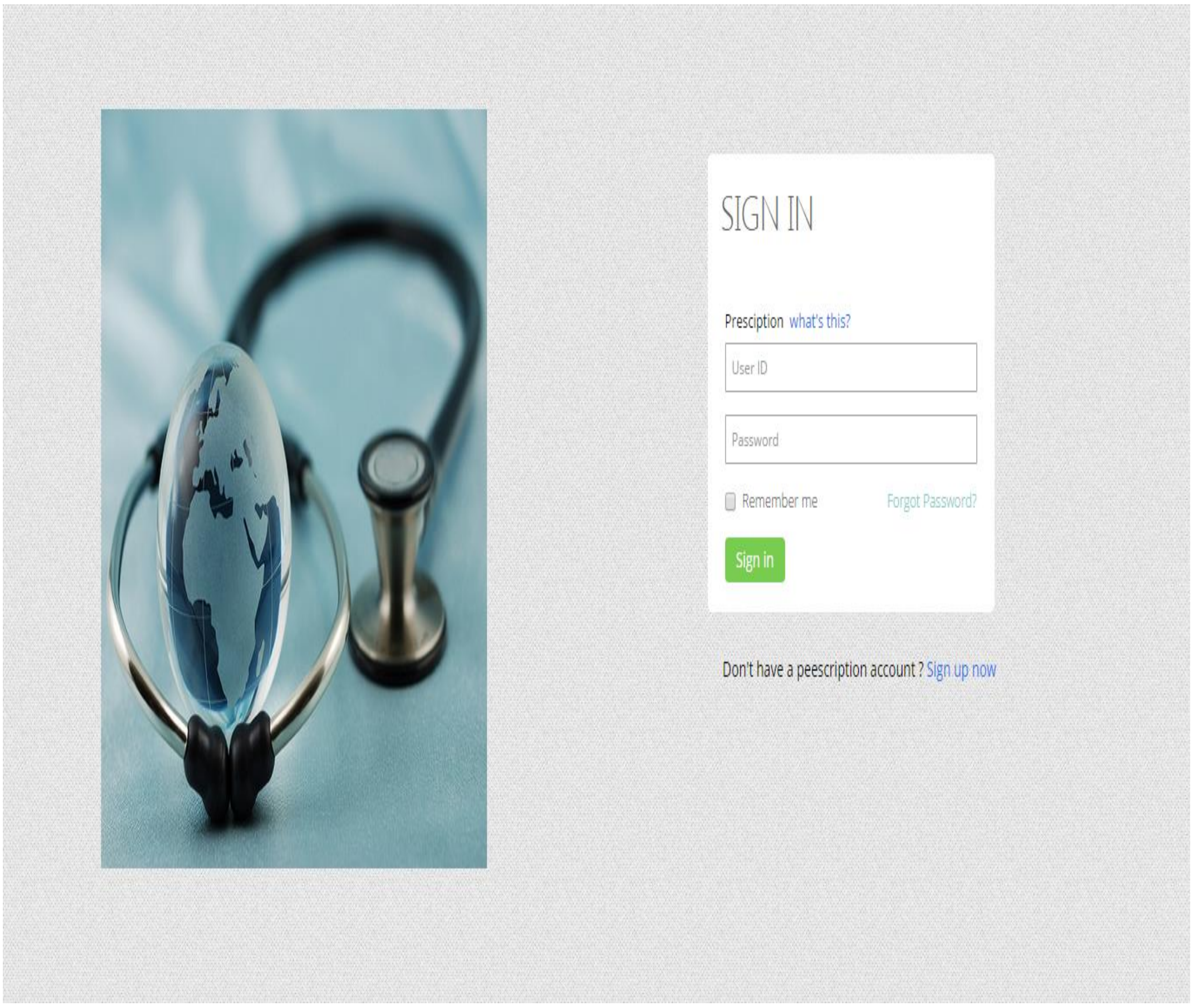
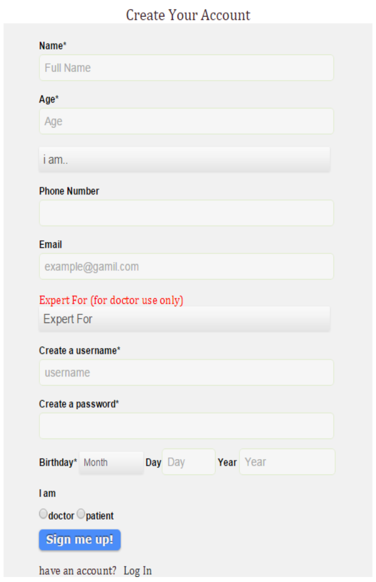


Figure: Sign up form for the patient and doctor



Create Your Account

Name*

Full Name

Age*

Age

i am..

Phone Number

Email

example@gamil.com

Expert For (for doctor use only)

Expert For

Create a username*

username

Create a password*

Birthday* Month Day Day Year Year

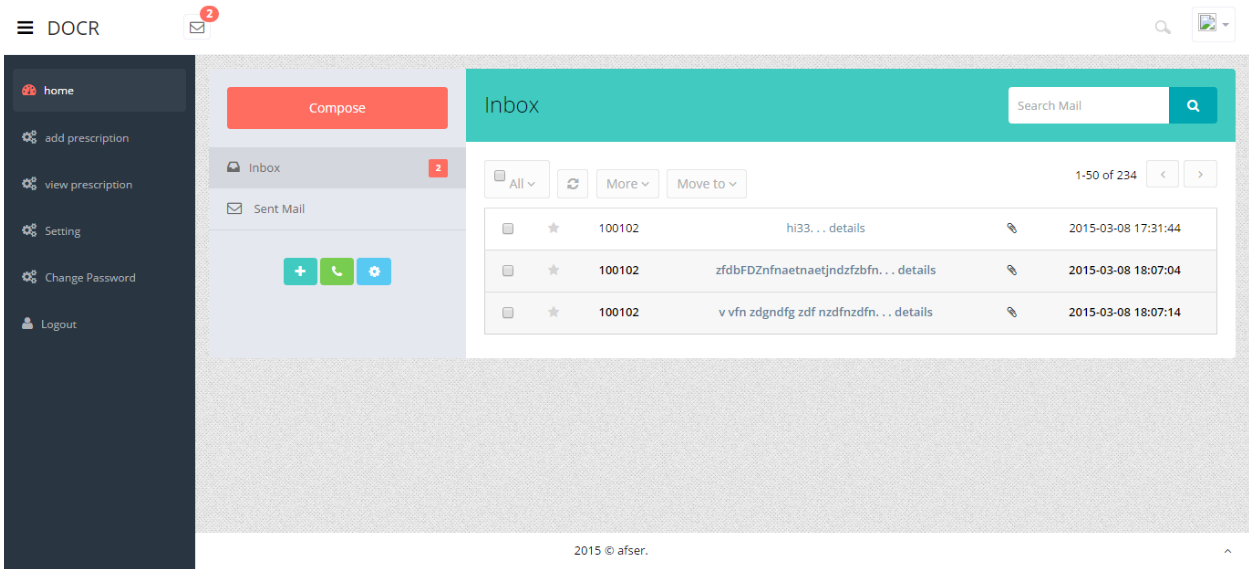
I am

doctor patient

Sign me up!

have an account? Log In

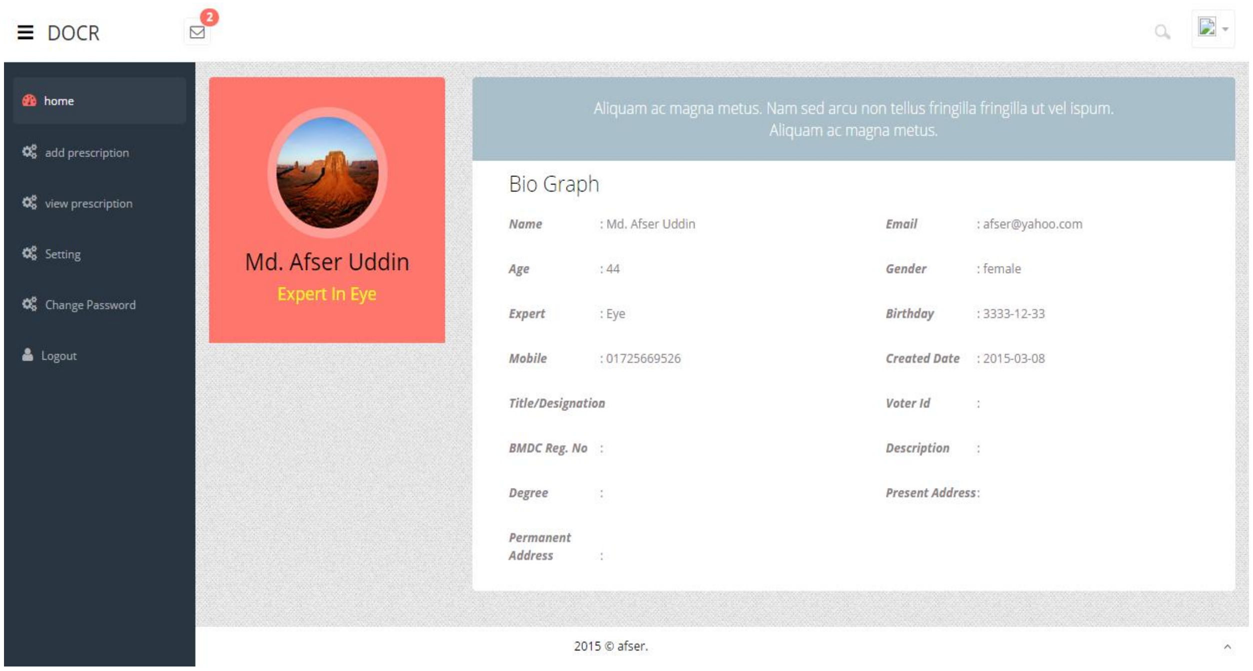
Figure: sign up form for the patient and doctor



		100102	hi33. . . details		2015-03-08 17:31:44
		100102	zfdbFDZnfnaetnaetjndzfzbfn. . . details		2015-03-08 18:07:04
		100102	v vfn zdgndfg zdf nzdfnzdfn. . . details		2015-03-08 18:07:14

2015 © afser.

Figure: Sending Email and Receiving Email between doctor and patient



DOCR

home
add prescription
view prescription
Setting
Change Password
Logout

Md. Afser Uddin
Expert In Eye

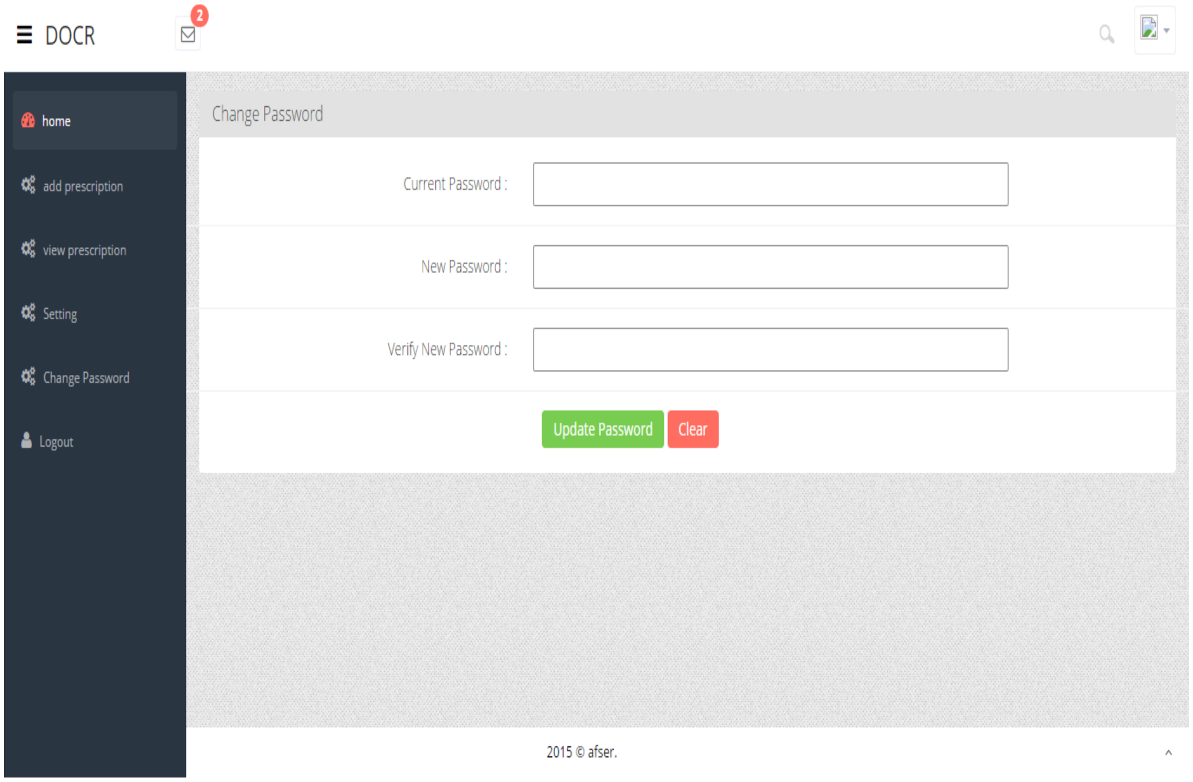
Aliquam ac magna metus. Nam sed arcu non tellus fringilla fringilla ut vel ispum. Aliquam ac magna metus.

Bio Graph

Name	: Md. Afser Uddin	*Email*	: afser@yahoo.com
Age	: 44	*Gender*	: female
Expert	: Eye	*Birthday*	: 3333-12-33
Mobile	: 01725669526	*Created Date*	: 2015-03-08
Title/Designation		*Voter Id*	:
BMDC Reg. No	:	*Description*	:
Degree	:	*Present Address*	:
Permanent Address	:		

2015 © afser.

Figure: when doctor log in



DOCR

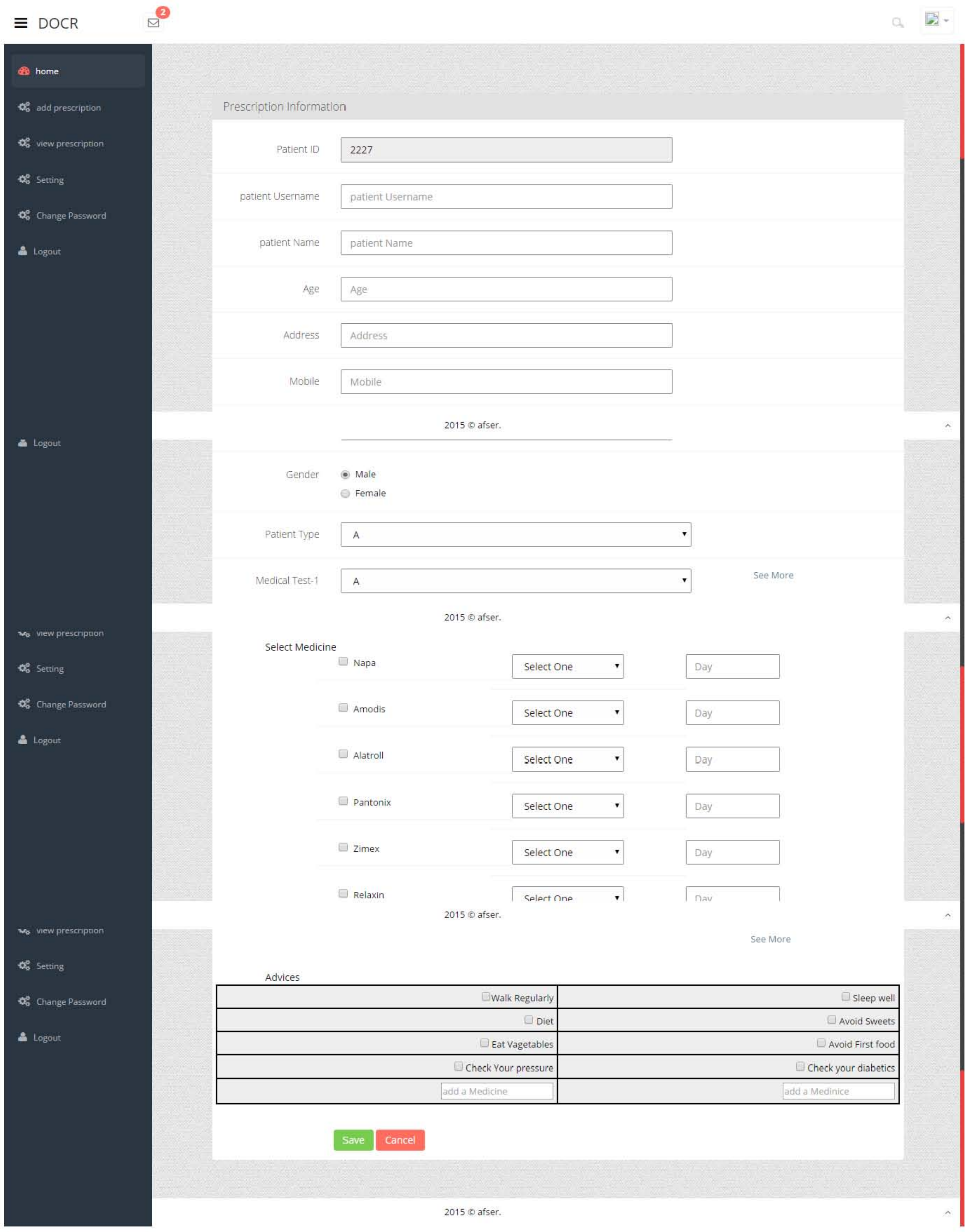
home
add prescription
view prescription
Setting
Change Password
Logout

Change Password

Current Password :
New Password :
Verify New Password :

2015 © afser.

Figure: Doctor change his system's password



DOCR

home
add prescription
view prescription
Setting
Change Password
Logout

Prescription Information

Patient ID	2227
patient Username	patient Username
patient Name	patient Name
Age	Age
Address	Address
Mobile	Mobile
Gender	[✓ Male] Female
Patient Type	A
Medical Test-1	A

Select Medicine

Napa	Select One	Day
Amodis	Select One	Day
Alatroll	Select One	Day
Pantonix	Select One	Day
Zimex	Select One	Day
Relaxin	Select One	Day

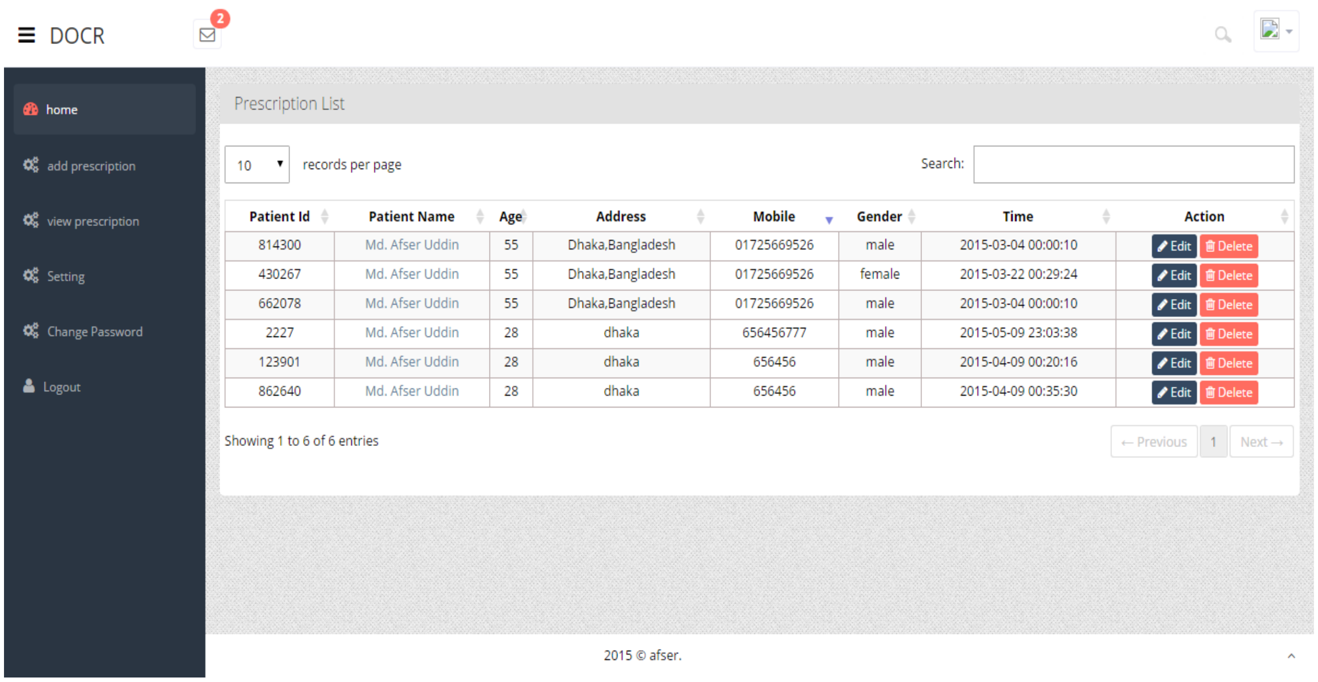
Advices

Walk Regularly	Sleep well
Diet	Avoid Sweets
Eat Vagetables	Avoid First food
Check Your pressure	Check your diabetics
add a Medicine	add a Medinice

Save Cancel

2015 © afser.

Figure: prescription form



Prescription List

10 records per page

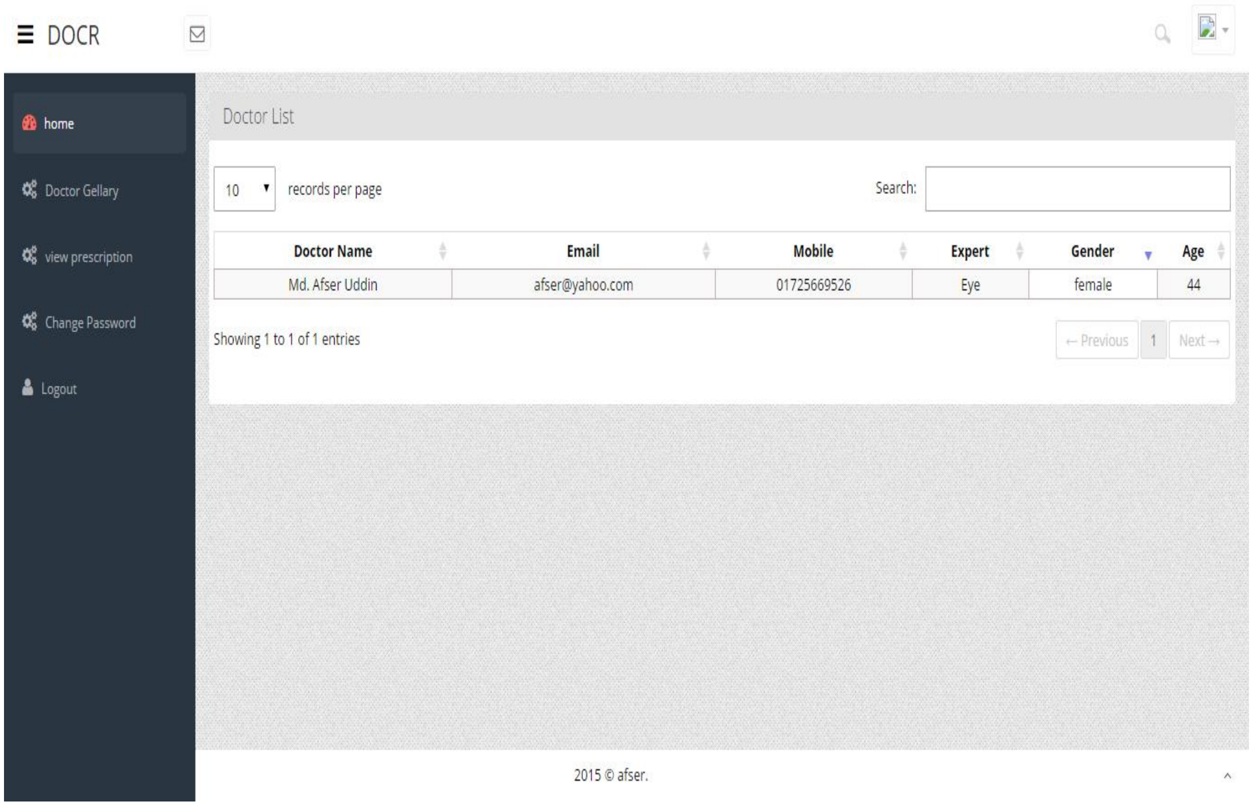
Search:

Patient Id	Patient Name	Age	Address	Mobile	Gender	Time	Action
814300	Md. Afser Uddin	55	Dhaka,Bangladesh	01725669526	male	2015-03-04 00:00:10	Edit Delete
430267	Md. Afser Uddin	55	Dhaka,Bangladesh	01725669526	female	2015-03-22 00:29:24	Edit Delete
662078	Md. Afser Uddin	55	Dhaka,Bangladesh	01725669526	male	2015-03-04 00:00:10	Edit Delete
2227	Md. Afser Uddin	28	dhaka	656456777	male	2015-05-09 23:03:38	Edit Delete
123901	Md. Afser Uddin	28	dhaka	656456	male	2015-04-09 00:20:16	Edit Delete
862640	Md. Afser Uddin	28	dhaka	656456	male	2015-04-09 00:35:30	Edit Delete

Showing 1 to 6 of 6 entries

2015 © afser.

Figure: list of prescription



Doctor List

10 records per page

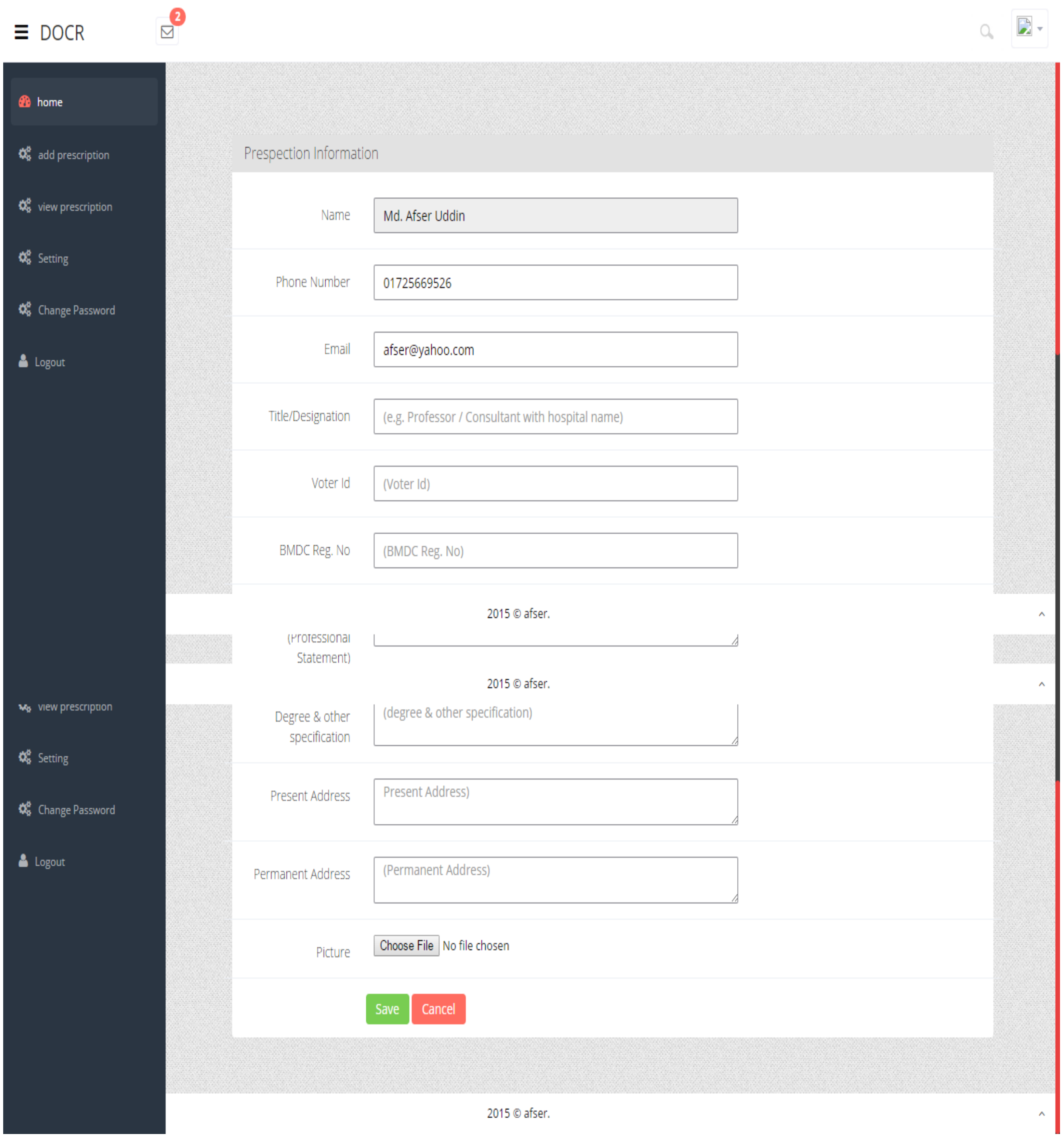
Search:

Doctor Name	Email	Mobile	Expert	Gender	Age
Md. Afser Uddin	afser@yahoo.com	01725669526	Eye	female	44

Showing 1 to 1 of 1 entries

2015 © afser.

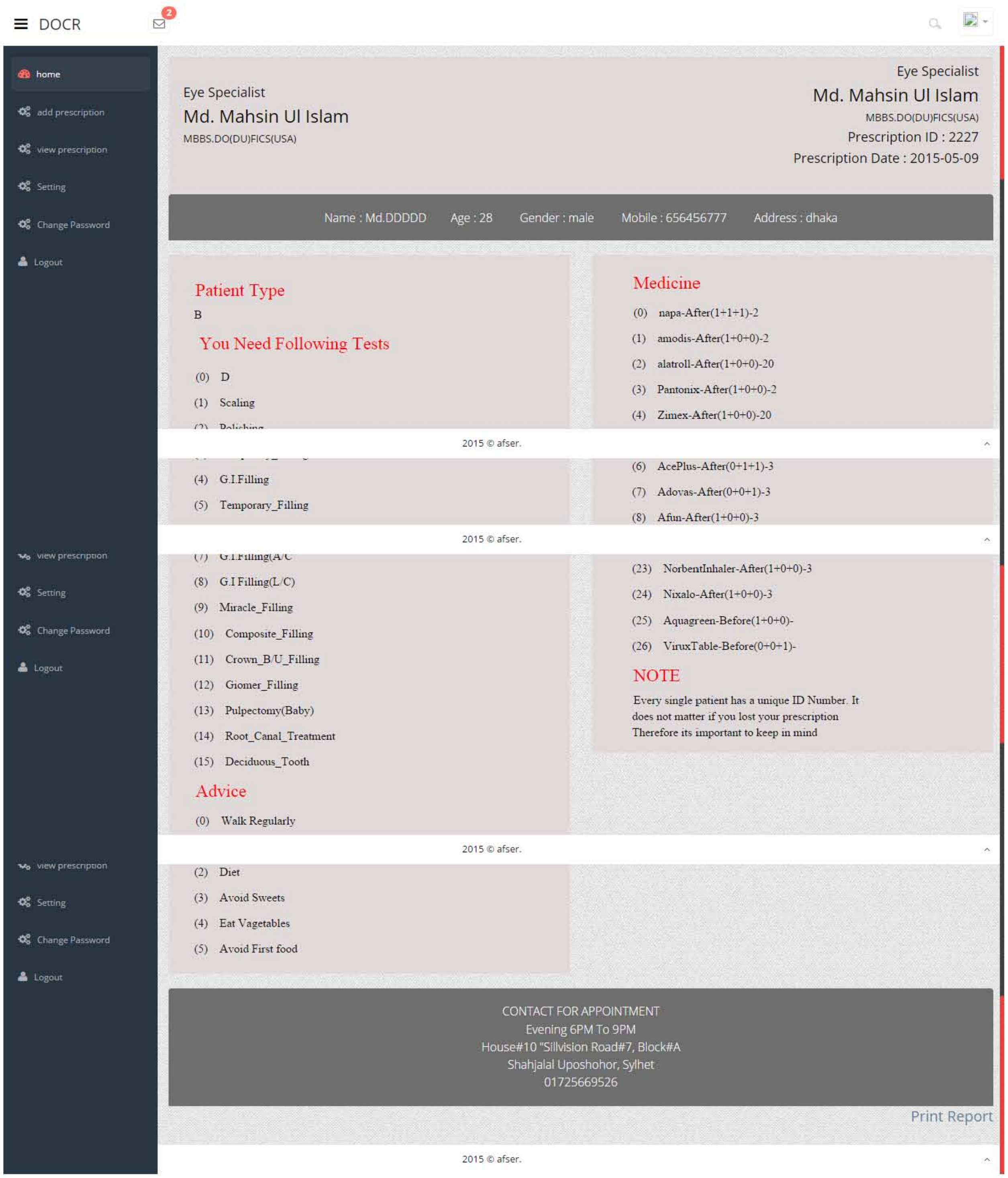
Figure: List of doctor's sign up for the system



Prespection Information

Name	Md. Afser Uddin
Phone Number	01725669526
Email	afser@yahoo.com
Title/Designation	(e.g. Professor / Consultant with hospital name)
Voter Id	(Voter Id)
BMDC Reg. No	(BMDC Reg. No)
(Professional Statement)	
Degree & other specification	(degree & other specification)
Present Address	Present Address)
Permanent Address	(Permanent Address)
Picture	No file chosen

Figure: writing new prescription on existing prescription



Eye Specialist
Md. Mahsin Ul Islam
MBBS.DO(DU)FICS(USA)

Eye Specialist
Md. Mahsin Ul Islam
MBBS.DO(DU)FICS(USA)
Prescription ID : 2227
Prescription Date : 2015-05-09

Name : Md.DDDDD Age : 28 Gender : male Mobile : 656456777 Address : dhaka

Patient Type
B

You Need Following Tests
(0) D
(1) Scaling
(4) G.I.Filling
(5) Temporary_Filling
(8) G.I Filling(L/C)
(9) Miracle_Filling
(10) Composite_Filling
(11) Crown_B/U_Filling
(12) Giomer_Filling
(13) Pulpectomy(Baby)
(14) Root_Canal_Treatment
(15) Deciduous_Tooth

Advice
(0) Walk Regularly
(2) Diet
(3) Avoid Sweets
(4) Eat Vagetables
(5) Avoid First food

Medicine
(0) napa-After(1+1+1)-2
(1) amodis-After(1+0+0)-2
(2) alatroll-After(1+0+0)-20
(3) Pantonix-After(1+0+0)-2
(4) Zimex-After(1+0+0)-20
(6) AcePlus-After(0+1+1)-3
(7) Adovas-After(0+0+1)-3
(8) Afun-After(1+0+0)-3
(23) NorbentInhaler-After(1+0+0)-3
(24) Nixalo-After(1+0+0)-3
(25) Aquagreen-Before(1+0+0)-
(26) ViruxTable-Before(0+0+1)-

NOTE
Every single patient has a unique ID Number. It does not matter if you lost your prescription Therefore its important to keep in mind

CONTACT FOR APPOINTMENT
Evening 6PM To 9PM
House#10 "Sillvision Road#7, Block#A
Shahjalal Uposhohor, Sylhet
01725669526

Print Report

2015 © afser.

Figure: Final Prescription that is hand over to the patient

4.8 Tools used for this Project

For this project I have used PHP as Primary language and also following code Igniter framework.

4.9 Hardware Configuration

For this project I have used the following hardware configuration-

- Xampp Software installed in any computer or laptop with minimum configuration.
- Code block or Notepad++ or Aptana

Chapter 5

Evaluation

5.1 Survey for this System

To taste this system we make a survey on this system. In this survey four people are participated. They are Faruque Micah, Supriya Paul, Faiz Micah, Shamim Ahmed and Roni Ahmed and Sadia Afrin and Raju Ahmed. I have explained the total system to them that how prescription management system works. They become registered to the system and take the service of this system. Faruque Micah and Faiz Micah and Supriya Paul and Sadia Afrin are overall happy with the system. they said it is overall a good system for the doctors who wants to write prescription in computer and like to hand over computer printed prescription to the patients. dynamic processes of the system like editing doctors details, patients information, patients prescription are attractive added by Faruque Micah. Besides this Roni ahmed thinks that design of the pages are not so nice to look. and the list of medicines and test should be decorated more nicely. He suggests me to give few more hours to develop the pages of prescription management system. it could make the huge difference. majority doctor will finds that it is useful and comfortable. On the other hand Shamim ahmed is satisfied about the system. he said it is undoubtedly a fantastic work to assist the doctor. It will help the doctor surely but doctor can't change the name of a medicine and test if required. That should be done by the doctor. He suggested to add this option to the system. Their comments are given below in a table.

Name	Do you think systems design is	Do you think doctor will be benefited by this system?	Do you think it is better than previous system?	Are you happy with this system?

	nice?			
Faiz Micah	Yes	Yes	Yes	Yes
Faruque Micah	Yes	No	Yes	No
Supriya Paul	No	Yes	Yes	No
Roni Ahmed	Yes	Yes	Yes	Yes
Sadia Afrin	Yes	Yes	Yes	Yes
Raju Ahmed	No	Yes	Yes	Yes

Table 4.1: Comments of different Users who used this system.

Chapter 6

Conclusion and Future Work

Doctor will be able to write a prescription by using this system. all of the necessary information like list of medicines, list of tests will be displayed on the screen. doctor just need to select the required medicines and test and finally print the prescription. If doctor required doctor can view patients prescription any time. patient with the id number can be track by doctor by searching from the database. then doctor can write another prescription by simply editing previous prescription. doctor can add as many as prescription in the database. Patient can also contact with the doctor. there is a system that patient can email to the respective doctor for any query or appointment . Patient can add previous prescription in the email and send to the doctor. doctor can also reply the email.

6.1 Strength of this System

- Well decorated pages of the system
- Strong databases to save the prescription and view if requires.
- Dynamic, user friendly. doctor can edit, delete and update the information from the system.

6.2Weakness of this System

- Doctor needs to know the basic of computer .
- Doctor should have idea to manage the system

6.3Future Work

In future I want to work with this project. I want to improve this system by-

- Adding auto reminder for the patient.

Appendix

Project Source Code

create_account.php

add_new.php

```
<section id="main-content">
<section class="wrapper">
<div class="row" style="margin: 5%">
<div class="col-lg-12">
<section class="panel">
<header class="panel-heading">
Prescription Information
</header>
<div class="panel-body">
<?pup
if ($this->session->user data('successful')):
echo '<div class="alert alert-success fade in"><button data-dismiss="alert" class="close close-sm" type="button"><I class="fa fa-times"></I></button><strong>Success Message !!! </strong> ' . $this->session->user data('successful') . '</div>';
$this->session->unset_userdata('successful');
end if;
if ($this->session->user data('failed')):
echo '<div class="alert alert-block alert-danger fade in"><button data-dismiss="alert" class="close close-sm" type="button"><I class="fa fa-times"></I></button><strong>Failed Message !!! </strong> ' . $this->session->user data('failed') . '</div>';
$this->session->unset_userdata('failed');
end if;
```

```
?>
<div class="form" style="margin-top: 20px">
<form class="cmxform form-horizontal tasi-form" id="signupForm" method="post" action="<?pup echo site_url('doctor/add_Patientinfo'); ?>">
<div class="form-group ">
<label for="patient_id" class="control-label col-lg-2">Patient ID</label>
<div class="col-lg-6">
<input class=" form-control" id="patient_id" name="patient_id" value="<?pup echo(rand(0, 999999)); ?>" readonly type="text" />
</div>
</div>
<div class="form-group ">
<label for="patient name" class="control-label col-lg-2">patient Username</label>
<div class="col-lg-6">
<input class=" form-control" id="username" name="username" type="text" placeholder="patient Username" />
</div>
</div>
<div class="form-group ">
<label for="patient name" class="control-label col-lg-2">patient Name</label>
<div class="col-lg-6">
<input class=" form-control" id="patient name" name="patient name" type="text" placeholder="patient Name" required />
</div>
</div>
<div class="form-group ">
<label for="age" class="control-label col-lg-2">Age</label>
<div class="col-lg-6">
<input class="form-control " id="age" name="age" type="text" placeholder="Age" required />
</div>
```

```
</div>
<div class="form-group ">
<label for="address" class="control-label col-lg-2">Address</label>
<div class="col-lg-6">
<input class="form-control " id="address" name="address" type="text" placeholder="Address" />
</div>
</div>
<div class="form-group ">
<label for="mobile" class="control-label col-lg-2">Mobile</label>
<div class="col-lg-6">
<input class="form-control " id="mobile" name="mobile" type="text" placeholder="Mobile" required />
</div>
</div>
<div class="form-group ">
<label for="subject name" class="control-label col-lg-2">Date</label>
<div class="col-lg-6">
<input class="form-control" type="text" id="dailysearchto" name="joining date" placeholder="Date" required />
</div>
</div>
<div class="form-group ">
<label for="subject name" class="control-label col-lg-2">Gender</label>
<div class="col-lg-8">
<div class="radio">
<label>
<input type="radio" name="sex" id="sex" value="male" checked>Male
</label>
</div>
<div class="radio">
```

```
<label>
<input type="radio" name="sex" id="sex" value="female">Female
</label>
</div>
</div>
</div>
<div class="form-group ">
<label for="patient type" class="control-label col-lg-2">Patient Type</label>
<div class="col-lg-6">
<select class="form-control" name="patient type" id="patient type" required>
<option value="A">A</option>
<option value="B">B</option>
<option value="C">C</option>
</select>
</div>
</div>
<div class="form-group ">
<label for="test" class="control-label col-lg-2">Medical Test-1</label>
<div class="col-lg-6">
<select class="form-control" name="medical test[]" id="medical test">
<option value="A">A</option>
<option value="B">B</option>
<option value="C">C</option>
<option value="D">D</option>
<option value="E">E</option>
<option value="F">F</option>
</select>
</div>
<a style="text-align: right; margin-left: 100px; cursor: pointer" id="seemorefortest">See More</a>
</div>
```

```
<span id="showmedicaltest" hidden>
<style>
table, th, td {
border: 2px solid black;
border-collapse: collapse;
}
th, td {
padding: 5px;
text-align: right;
}
</style>
<table style="width:100%">
<tr>
<div style="width: 15em; float: left;">
<td>Scaling <input id="" type="checkbox" name="medical test[]" value="Scaling" /> </td>
</div>
<div style="width: 12em; float: left;">
<td>Polishing <input id="" type="checkbox" name="medical test[]" value="Polishing" /> </td>
</div>
<div style="width: 12em; float: left;">
<td>Temporary Filling <input id="" type="checkbox" name="medical test[]" value="Temporary_Filling" /></td>
</div>
</tr>
<tr>
<div style="width: 15em; float: left;">
<td> G.I Filling (Baby)<input id="" type="checkbox" name="medical test[]" value="G.I.Filling" /></td>
```

```
<td>Temporary Filling <input id="" type="checkbox" name="medical test[]" value="Temporary_Filling" /></td>
<td>Amaglam Filling <input id="" type="checkbox" name="medical test[]" value="Amaglam_Filling" /></td>
</div>
</tr>
<tr>
<div style="width: 15em; float: left;">
<td>G.I Filling(A/C) <input id="" type="checkbox" name="medical test[]" value="G.I.Filling(A/C" /></td>
<td>G.I Filling(L/C) <input id="" type="checkbox" name="medical test[]" value="G.I Filling(L/C)" /></td>
<td>Miracle Filling <input id="" type="checkbox" name="medical test[]" value="Miracle_Filling" /></td>
</div>
</tr>
<tr>
<div style="width: 15em; float: left;">
<td>Composite Filling <input id="" type="checkbox" name="medical test[]" value="Composite_Filling" /></td>
<td>Crown B/U Filling <input id="" type="checkbox" name="medical test[]" value="Crown_B/U_Filling" /></td>
<td>Gilmer Filling <input id="" type="checkbox" name="medical test[]" value="Giomer_Filling" /></td>
</div>
</tr>
<tr>
<div style="width: 15em; float: left;">
<td>Pulpectomy (Baby) <input id="" type="checkbox" name="medical test[]" value="Pulpectomy(Baby)" /> </td>
```

```
<td>Root Canal Treatment <input id="" type="checkbox" name="medical test[]" value="Root_Canal_Treatment" /></td>
<td>Deciduous Tooth  <input id="" type="checkbox" name="medical test[]" value="Deciduous_Tooth" /> </td>
</div>
</tr>
<tr>
<div style="width: 15em; float: left;">
<td>Anterior Tooth <input id="" type="checkbox" name="medical test[]" value="Anterior_Tooth" /></td>
<td>Posterior Tooth <input id="" type="checkbox" name="medical test[]" value="Posterior Tooth" /></td>
<td>3rd Molar Tooth <input id="" type="checkbox" name="medical test[]" value="3rd Molar Tooth" /></td>
</div>
</tr>
<tr>
<div style="width: 15em; float: left;">
<td>Pus Drainage <input id="" type="checkbox" name="medical test[]" value="Pus Drainage" /></td>
<td>Operculectomy <input id="" type="checkbox" name="medical test[]" value="Operculectomy" /></td>
<td>Surgical Extraction(Simple) <input id="" type="checkbox" name="medical test[]" value="Surgical Extraction(Simple)" /></td>
</div>
</tr>
<tr>
<td><input type="text" name="medical test[]" placeholder="add a Test"> </td>
<td><input type="text" name="medical test[]" placeholder="add a Test"> </td>
<td><input type="text" name="medical test[]" placeholder="add a Test"> </td>
</tr>
```

```
</table>
</span><br><br>
<label style="margin-left: 70px; font-size: 14px; weight: 300;">Select Medicine</label>
<div class="row" style="margin-left: 160px">
<div class="form-group col-md-4">
<div class="col-md-8">
<input id="cheese" type="checkbox" name="napa" value="napa" />&nbsp;&nbsp;Napa
</div>
</div>
<div class="form-group col-md-4">
<div class="col-md-8">
<select class="form-control" name="medcine1" id="medcine1">
<option value="">Select One</option>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="AcePlus" type="checkbox" name="AcePlus" value="AcePlus" />
Ace Plus
<select class="form-control" name="medcine7" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
```

```
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day7" size="1" maxlength="2" placeholder="days" value=""> </td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Adovas" type="checkbox" name="Adovas" value="Adovas" />
Adovas
<select class="form-control" name="medcine8" id="medcine8">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day8" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td>
```

```
<div style="width: 22em; float: right;">
<input id="Afun" type="checkbox" name="Afun" value="Afun" />
Afun
<select class="form-control" name="medcine9" id="medcine9">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day9" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Agrizeb" type="checkbox" name="Agrizeb" value="Agrizeb" />
Agrizeb
<select class="form-control" name="medcine10" id="medcine10">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
```

```
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day10" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="Apsol" type="checkbox" name="Apsol" value="Apsol" />
Apsol
<select class="form-control" name="medcine11" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
```

```
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day11" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="GermisolHandRub" type="checkbox" name="GermisolHandRub" value="GermisolHandRub" />
Germisol Hand Rub
<select class="form-control" name="medcine12" id="medcine12">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day12" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td>
```

```
<div style="width: 22em; float: right;">
<input id="Goldazim" type="checkbox" name="Goldazim" value="Goldazim" />
Goldazim
<select class="form-control" name="medcine13" id="medcine13">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day13" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Neurolin" type="checkbox" name="Neurolin" value="Neurolin" />
Neurolin
<select class="form-control" name="medcine14" id="medcine14">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
```

```
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day14" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="Gensia" type="checkbox" name="Gensia" value="Gensia" />
Gensia
<select class="form-control" name="medcine15" id="medcine15">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
```

```
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day15" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Geston" type="checkbox" name="Geston" value="Geston" />
Geston
<select class="form-control" name="medcine16" id="medcine16">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<select class="form-control" name="medcine18" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day18" size="1" maxlength="2" placeholder="days" value="">
```

```
</td>
</div>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="Melixol" type="checkbox" name="Melixol" value="Melixol" />
Melixol
<select class="form-control" name="medcine19" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day19" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Mevin" type="checkbox" name="Mevin" value="Mevin" />
Mevin
<select class="form-control" name="medcine20" id="medcine7">
```

```
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day20" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="Rex" type="checkbox" name="Rex" value="Rex" />
Rex
<select class="form-control" name="medcine21" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
```

```
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day21" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Rezulin" type="checkbox" name="Rezulin" value="Rezulin" />
Rezulin
<select class="form-control" name="medcine22" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day22" size="1" maxlength="2" placeholder="days" value="">
</td>
```

```
</div>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="NorbentInhaler" type="checkbox" name="NorbentInhaler" value="NorbentInhaler" />
Norbent Inhaler
<select class="form-control" name="medcine23" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day23" size="1" maxlength="2" placeholder="days" value=""
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="Nixalo" type="checkbox" name="Nixalo" value="Nixalo" />
Nixalo
<select class="form-control" name="medcine24" id="medcine7">
```

```
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day24" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="Aquagreen" type="checkbox" name="Aquagreen" value="Aquagreen" />
Aquagreen® G
<select class="form-control" name="medcine25" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
```

```
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
</select> <input type="text" name="day25" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
<td>
<div style="width: 22em; float: right;">
<input id="ViruxTable" type="checkbox" name="ViruxTable" value="ViruxTable" />
Virux Tablet®
<select class="form-control" name="medcine26" id="medcine7">
<option value="">Select One</option>
<option value="After(1+1+1)">After(1+1+1)</option>
<option value="After(0+1+1)">After(0+1+1)</option>
<option value="After(1+0+1)">After(1+0+1)</option>
<option value="After(0+0+1)">After(0+0+1)</option>
<option value="After(1+0+0)">After(1+0+0)</option>
<option value="After(0+1+0)">After(0+1+0)</option>
<option value="Before(1+1+1)">Before(1+1+1)</option>
<option value="Before(0+1+1)">Before(0+1+1)</option>
<option value="Before(1+0+1)">Before(1+0+1)</option>
<option value="Before(0+0+1)">Before(0+0+1)</option>
<option value="Before(1+0+0)">Before(1+0+0)</option>
<option value="Before(0+1+0)">Before(0+1+0)</option>
```

```
</select> <input type="text" name="day26" size="1" maxlength="2" placeholder="days" value="">
</td>
</div>
</tr>
<tr>
<td><input type="text" name="medcine27" placeholder="add a Medicine"> </td>
<td><input type="text" name="medcine28" placeholder="add a Medinice"> </td>
</tr>
</table>
</span><br><br>
<label style="margin-left: 70px; font-size: 14px; weight: 300;">Advices</label>
<style>
table, th, td {
border: 2px solid black;
border-collapse: collapse;
background-color: #eee;
}
th, td {
padding: 5px;
text-align: right;
}
</style>
<table style="width:100%">
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Walk Regularly" />Walk Regularly
```

```
</div>
</td>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Sleep well" />
Sleep well
</div>
</td>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Diet" />
Diet
</div>
</td>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Avoid Sweets" />
Avoid Sweets
</div>
</td>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Eat Vagetables" />
Eat Vagetables
</div>
</td>
<td>
```

```
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Avoid First food" />
Avoid First food
</div>
</td>
</tr>
<tr>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Check Your pressure" />
Check Your pressure
</div>
</td>
<td>
<div style="width: 22em; float: right;">
<input id="" type="checkbox" name="adv[]" value="Check your diabetics" />
Check your diabetics
</div>
</td>
</tr>
<tr>
<td><input type="text" name="adv[]" placeholder="add a Medicine"> </td>
<td><input type="text" name="adv[]" placeholder="add a Medinice"> </td>
</tr>
</table>
<br><br>
<div class="form-group">
<div class="col-lg-offset-2 col-lg-10">
<button class="btn btn-success" type="submit">Save</button>
<button class="btn btn-danger" type="button">Cancel</button>
</div>
```

```
</div>
</form>
</div>
</div>
</section>
</div>
</div>
</section>
</section>
<script type="text/javascript">
$(document).ready(function() {
$("#seemorefortest").click(function() {
$("#showmedicaltest").toggle();
});
});
$(document).ready(function() {
$("#seemoreformedicine").click(function() {
$("#showmedicine").toggle();
});
});
</script>
```

Bibliography

[1] Advanced PHP for web development(The principle hall PTR advanced web development series) by Christopher Cosentino. prentice Hall PTR. Paperback-1 October, 2002

[2] Advanced PHP programming by Schossnagle. Sams. Paperback- October 2003

[3] A programmer's Introduction to PHP by W.J Gilmore. Apress. Paperback- 1 January, 2001

[4] Create Dynamic web pages using PHP and My SQL by David Tansley. Addison 1 November, 2001

[5] Beginning PHP, MySQL and Apache wrox press ltd. Paperback- 1 June, 2003

[6] Making use of PHP by Appu. John Wiley and sons Inc. Paperback- 24 July, 2002

[8] Professional PHP By Ed Lecky-Thompson, Steven D. Nowicki, and Thomas Myer**.** Paperback-14 February,2003

[9] Learning PHP, MySQL and JavaScript: A step by step guide to creating Dynamic websites by Robin Nixon. Paperback-30 August,2001

[10] PHP solutions: Dynamic web design made easy by David Powers. Paperback-25 July 2005.