

Project Report

Online Prescription Systems for Doctor.

By

Md. Mineuddin Ahmed Dipu

Professional Master’s in computer Science

Department of Computer Science and Engineering

October 2022

Declaration:

Acknowledgment:

I would like to give special thanks to following people who have made the completion of this project possible:

My parents, for their continued support and generous funding.

Dr. Md. Ezharul Islam (My supervisor), for giving me the opportunity to undertake this project with her as well as for his support and guidance.

Md. Rahul (My friend), for his unyielding knowledge on anything computer or math related and his willingness to always help.

Arob Hossain (Younger Brother), for his unrested help to make the error vanished and also to the coding.

Abstract:

This document is the final report for the project online prescription system for doctors. The key objective of this project is to provide a high tech online prescription with full of medicine list, investigation list and patient/doctor satisfaction. We will release the newer version day by day and always try to make it smooth, easy and fast loader.

Besides we will create more opportunities for new doctors so they can connect, share and learn more from our prescription.

Table of Contents

1. Introduction
   1. Purpose ………………………………………………………..8
   2. Scope of project ……………………………………………….8
   3. Existing System ……………………………………………….9
   4. Proposed System ………………………………………………9
2. Software Requirement Specification (SRS)
   1. System Environment……………………………………………9
   2. Functional Requirement Specification ………………………9
   3. Non-Functional Requirements…………………………………10
   4. Detailed Requirement Specification……………………………10
      1. Functional Requirement………………………………….10
      2. Non-functional Requirement……………………………..12
   5. Logical Structure of the Data……………………………………12
   6. Security………………………………………………………….14
3. Design
   1. User Interface Design……………………………………………15
      1. Introduction………………………………………………..15
      2. Interface Analysis………………………………………….15
      3. User Analysis………………………………………………15
   2. System Design Environment…………………………………….17
      1. Frontend Interface Design …………………………………….17
      2. Backend Design……………………………………………...17
      3. Frontend and Backend Programming…………………………..17
      4. Backend Database……………………………………………17
   3. System Running Environment…………………………………17
   4. Interface Design Steps…………………………………………18-23
4. Testing
   1. Overview………………………………………………………23
   2. Scope…………………………………………………………..23
   3. Appendix………………………………………………………24
5. Conclusion…………………………………………………………28

List of figures

1. Figure 2.1 System Environment…………………………..…………10
2. Figure 2.2 Logical Structure of Prescription data…………...…..…...13
3. Dashboard design …………………………………………..………. 19
4. sidebar menu page design ……………………………………………20
5. : Doctor add page design ……………………………………..……. 21
6. : Doctor List page design ………………………………….….…….21
7. patient add page design ………………………………..……………22
8. Medicine Add page design …………………………………………..23
9. Investigation insert page design …………………………………….. 23
10. 1Prescription page design …………………………………………….24
11. Introduction
    1. Purpose:

The purpose of this document is to present a detailed description about online prescription project. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

* 1. Scope of Project:

By this project doctors and patient both can be benefited. Doctors can easily prescribe patient. They can keep record of their patient and patient also get email of the prescription. So they also can track their health history. They don’t need to preserve any piece of paper at their home. Moreover a piece of paper may destroy at any time. On the other hand if the patient will come after a long time then the doctor can easily track this patient previous record. So it will easy to find the previous health history of any patient. A doctor also can research on the data he has. He also can share his experience with other practitioners and medical student for their study purpose. Overall it can play an incredible role in our health care sector.

* 1. Existing System:

There are a few existing systems in our marketplace now. There are more then 7-8 prescription systems in Bangladesh. We have some added features what is going to amaze doctors to take our systems and also our system is not only design for doctors but also design for patient and organization like hospital and diagnostic center.

* 1. Proposed System:

The proposed system is designed for an online prescription system with doctor appointment and payment. Where patient can take serial of any doctor, confirm appointment through payment and doctor prescribe the patient through our prescription system. The patient will get the prescription through the email along with printed copy.

1. Software Requirement Specification (SRS)

2.1 System Environment

Admin

Doctor

Appointment

Prescription

Print

Payment

Figure 2.1: Online Prescription system Environment

The Admin accesses the Appointment and payment option. Doctor will create the prescription and print the prescription.

2.2 Functional Requirements Specification

This section outlines the use cases for each of the active doctors separately while the admin is main actor in this system.

2.3 Non-Functional Requirements

The Online Prescription will be on a cloud server. We will use either google cloud or Amazon cloud. There will not be any physical server. So this will be very efficient and there won’t be any risk of server crash or something like this.

Easy Load, Superfast, Communication, Good font, Cloud server, Concurrency, Real time feedback.

2.4 **Detailed** **Requirement Specification**

2.4.1 Functional Requirements

The Logical Structure of the Data is contained in Section.

2.1 Pre-configuration for medicine and Prescription like medicine type, generic entry and some others.

|  |  |
| --- | --- |
| **Trigger** | The admin selects to add new entities to the database. |
| **Precondition** | The admin has accessed the configuration screen. |
| **Basic Path** | 1. The system presents a blank grid to enter the required information. 2. The admin enters the information and submits the form. 3. The system checks that the required fields are not blank and updates the database. |
| **Alternative Paths** | 1. If in step 2, either field is blank, the Editor is instructed to add an entry. No validation for correctness is made. |
| **Post Condition** | The entity has been added to the database. |
| **Exception Paths** | The admin may abandon the operation at any time. |

2.2 Add Patient and Doctor

|  |  |
| --- | --- |
| **Trigger** | The admin selects to add new patient and doctor to the database. |
| **Precondition** | The admin has accessed the configuration screen. |
| **Basic Path** | 1 The system presents a blank grid to enter the required information.  2 The admin enters the information and submits the form.  3 The system checks that the required fields are not blank and updates the database. |
| **Alternative Paths** | 1. If in step 2, either field is blank, the Editor is instructed to add an entry. No validation for correctness is made. |
| **Post Condition** | The patient and doctor has been added to the database. |
| **Exception Paths** | The admin may abandon the operation at any time. |
| **Trigger** | The admin selects to add new entities to the database. |

2.3 Create Prescription

|  |  |
| --- | --- |
| **Trigger** | The doctor selects to create new prescription to the database. |
| **Precondition** | The doctor has accessed the configuration screen. |
| **Basic Path** | 1. The system presents a blank grid to enter the required information. 2. Doctor enters the information and submits the form 3. The system checks that the required fields are not blank and updates the database. |

2.4 Edit Prescription

|  |  |
| --- | --- |
| **Trigger** | The doctor selects to update existing prescription to the database. |
| **Precondition** | The doctor has accessed the configuration screen. |
| **Basic Path** | 1. The system presents a editable grid to update the required information. 2. Doctor edit the information and submits the form 3. The system checks that the required fields are not blank and updates the database. |

* + 1. Non-Functional Requirements

2.5 Logical Structure of the Data

The logical structure of the data to be stored in the internal database is given below.

Prescription

Doctor

writes

sent to

Patient

Figure .2 - Logical Structure of the Prescription Data

The data descriptions of each of these data entities is as follows:

|  |  |  |
| --- | --- | --- |
| **SL** | **Table\_name** | **Column\_name** |
| 1 | roles | id,name,status,time |
| 2 | users | id,role\_id,name,email,password,status,time |
| 3 | Chamber Address | id,name,address,phone,email,city,state,country,website,logo\_img,description,status,time ,user\_id |
| 4 | doctor\_department | id,name,short\_name,department\_code,description,user\_id,time |
| 5 | doctor\_designation | id,name,short\_name,designation\_code,description,user\_id,time |
| 6 | doctors | id,name,email,phone,department\_id,designation\_id,birth\_date,blood\_group,about\_me,image,experience,speciality,degrees,user\_id,status,time |
| 7 | patient | id,patient\_ID,first\_name,last\_name,full\_name,email,phone,address,city,state,post\_code,district,birth\_date,nid,photo,gender,blood\_group,status,time |
| 8 | generics | id,name,status,user\_id,time |
| 9 | medicine\_type | id,company\_name,type\_code,name,short\_name,status,user\_id,time |
| 10 | strength | id,name,status,user\_id,time |
| 11 | company | id,supplier\_code,name,email,note,phone,address,status,user\_id,time |
| 12 | medicine | id,medicine\_code,name,supplier\_id,generic\_id,strength\_id,medicine\_type\_id,details,side\_effect,favourite,user\_id,status,time |
| 13 | dose\_duration\_advice | id,name,remark,user\_id,time |
| 14 | investigation | id,name,remark, status,user\_id,time |
| 15 | general\_advise | id,advise,remark,user\_id,time |
| 16 | visiting fee | id, doctor id, visit type, visit fees,user\_id,time |
| 17 | schedule | id,place\_id,doctor\_id,days,start\_time,end\_time,per\_patient\_time,user\_id,status,time |
| 18 | appointment | id,date,patient\_id,place\_id,serial\_number,general complain,doctor\_id,user\_id,status,time |
| 19 | patient payment | id, patient id, doctor id, visiting fees id, discount,net payable,paid amount, due, appointment id, date,user\_id,time |
| 20 | template | id,doctor\_id,group\_name,item\_type,item\_id,value1,valu2,value3,value4,remark,user\_id,status,time |
| 21 | prescription | id,appointment\_id,patient\_id,doctor\_id,date,bp,pulse,temp,weight,spo2,sugar,complain,diagnosis,past\_history,drug\_history,lpd,edd,followup,template\_id,user\_id,time |
| 22 | prescription\_medicine | prescription\_id,record\_date,medicine\_id,dose\_id,duration\_id,advice,remark,status,user,time |
| 23 | prescription\_advice | prescription\_id,record\_date,advice\_id,advice,remark,status,user,time |
| 24 | prescription\_investigation | prescription\_id,record\_date,investigation\_id,advice,remark,status,user,time |
| 25 | pres\_patient\_details | prescription\_id,record\_date,status,user,time |

* 1. Security

The server on which the online prescription resides will have its own security to prevent unauthorizedwrite/delete access. There is no restriction on read access. The use of email by a doctor is on the client systems and thus is external to the system.

The PC on which the doctor prescription resides will have its own security. Only the Admin will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the admin with write access to the prescription system to create the prescription.

1. Design

3.1 User Interface Design:

3.1.1 Introduction

User interface design creates an effective communication medium between a human and a computer. The interface has to be right because it models a user’s perception of the software. As we know that a key tenet of all software engineering process models is “understand the problem before you attempt to design a solution”, we analysis the interface before starting the design steps.

3.1.2 Interface Analysis

We divide interface analysis into following parts:

i. User Analysis

ii. Task Analysis

3.1.3 User Analysis

In this part we follow two steps:

a. Identify user

b. Know user

**Identify user**

From the requirements specification we have identified following three user categories.

1. Admin

2. Doctor

3. Assistant

**Know user**

We collect following information about the users.

**Admin**

Age: Any

Skills: Average

Domain expert: Yes

Office hour: Any time

Frequency of use: Very frequently

Consequence of a mistake: High

General computer experience: Yes

**Doctor**

Age: Any

Skills: Average

Frequency of use: very frequently

Consequence of a mistake: Medium

General computer experience: Yes

**Assistant**:

1. Create configuration for medicine and prescription.

Goal: doctor can use the prescription smoothly

Frequency of use: Normal

Consequence of a mistake: Medium

General computer experience: Yes

3.3 System Design Environment

3.3.1 Admin Interface Design:

* HTML5.0, CSS3, React JS.

3.3.2 Frontend and Backend Programming:

Laravel 8.0 with Restful API.

3.3.3 Backend Database

* Mysql database.
  1. System Running Environment

This system is tested and running in windows operating system. This is completely ok for Mozilla firefox, google chrome, opera, safari and also definitely internet explorer Browser.

3.5 Interface Design Steps

We follow the following steps to design the prescription systems user interface.

i. Define interface objects and actions

ii. Define events that will cause the state of the user interface to change

3.5.1 Define interface objects and actions

We identified following objects and actions for the user interface. This is the prototype for the system.

* + - 1. Prescription Dashboard

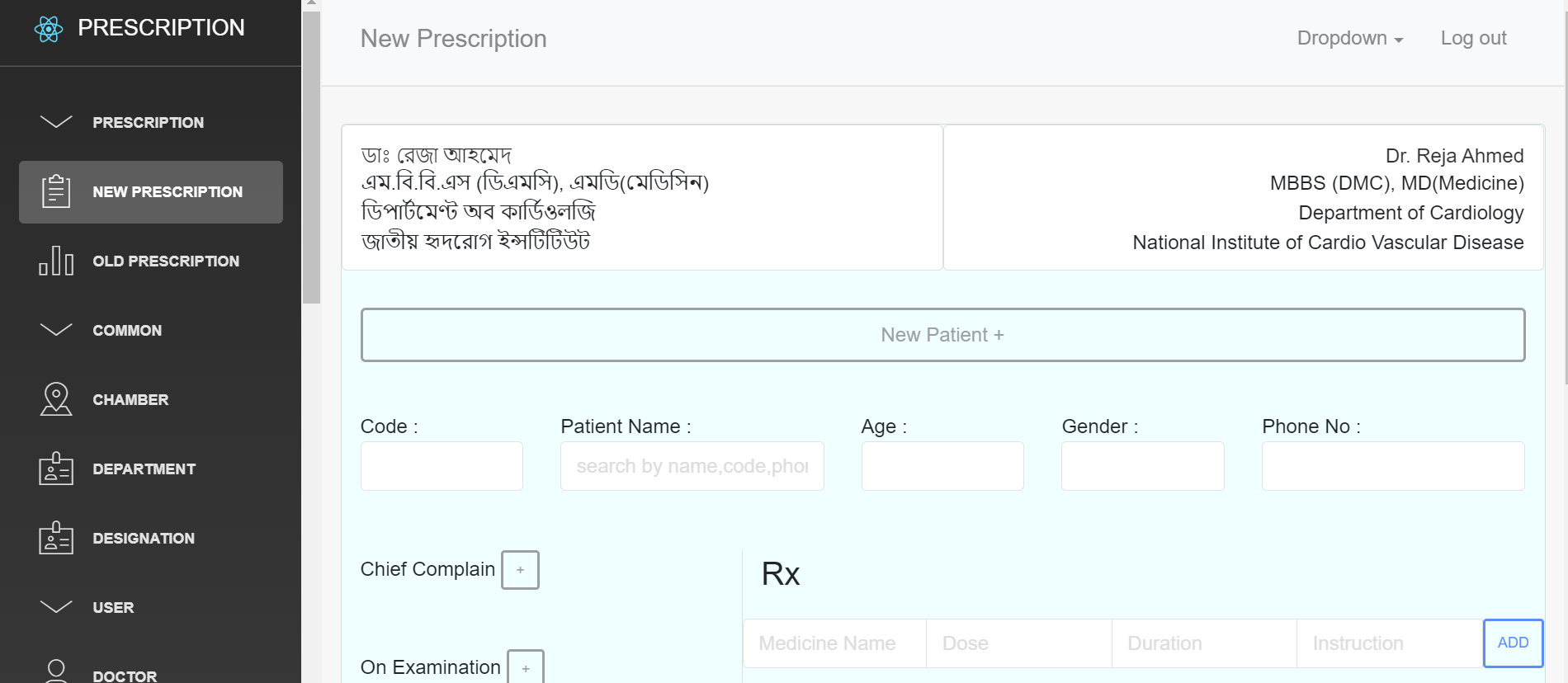
****

Figure 3.1: Dashboard design

* + - 1. Sidebar Menu

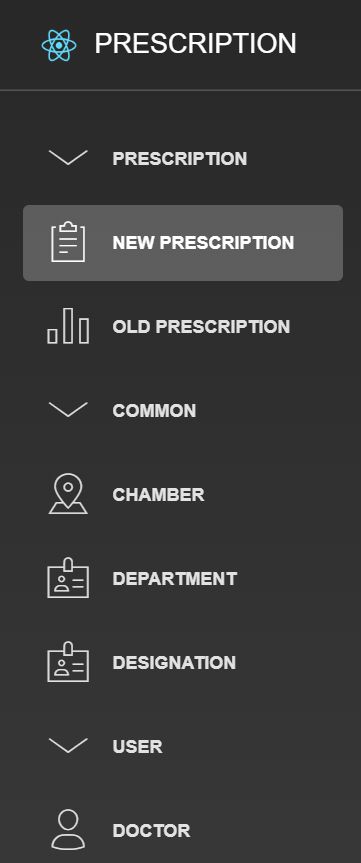


Figure 3.2: sidebar menu page design

3.5.1.3 Add Doctor Page

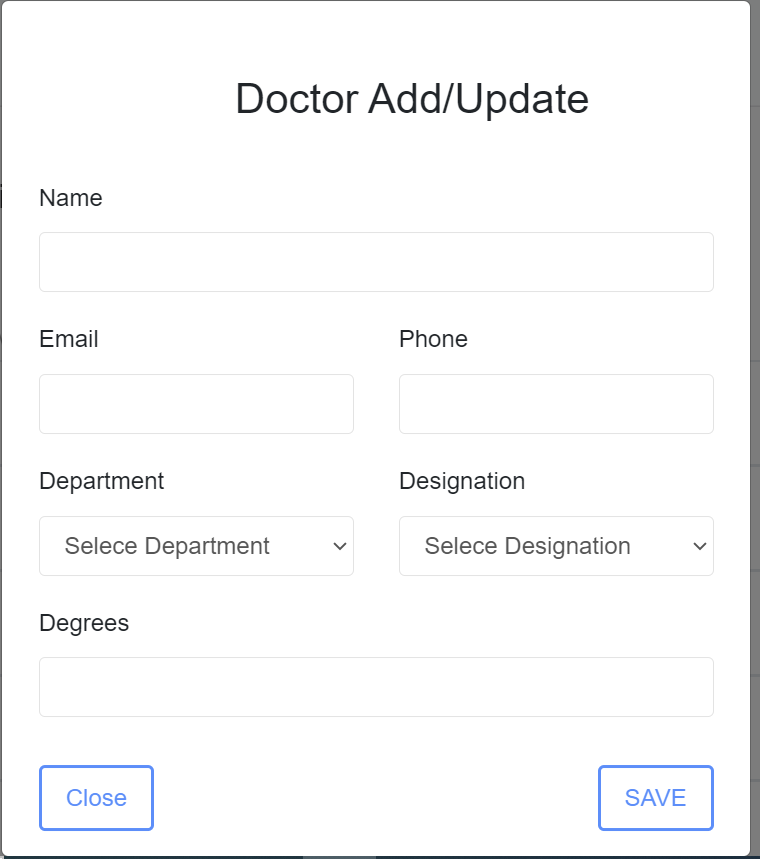
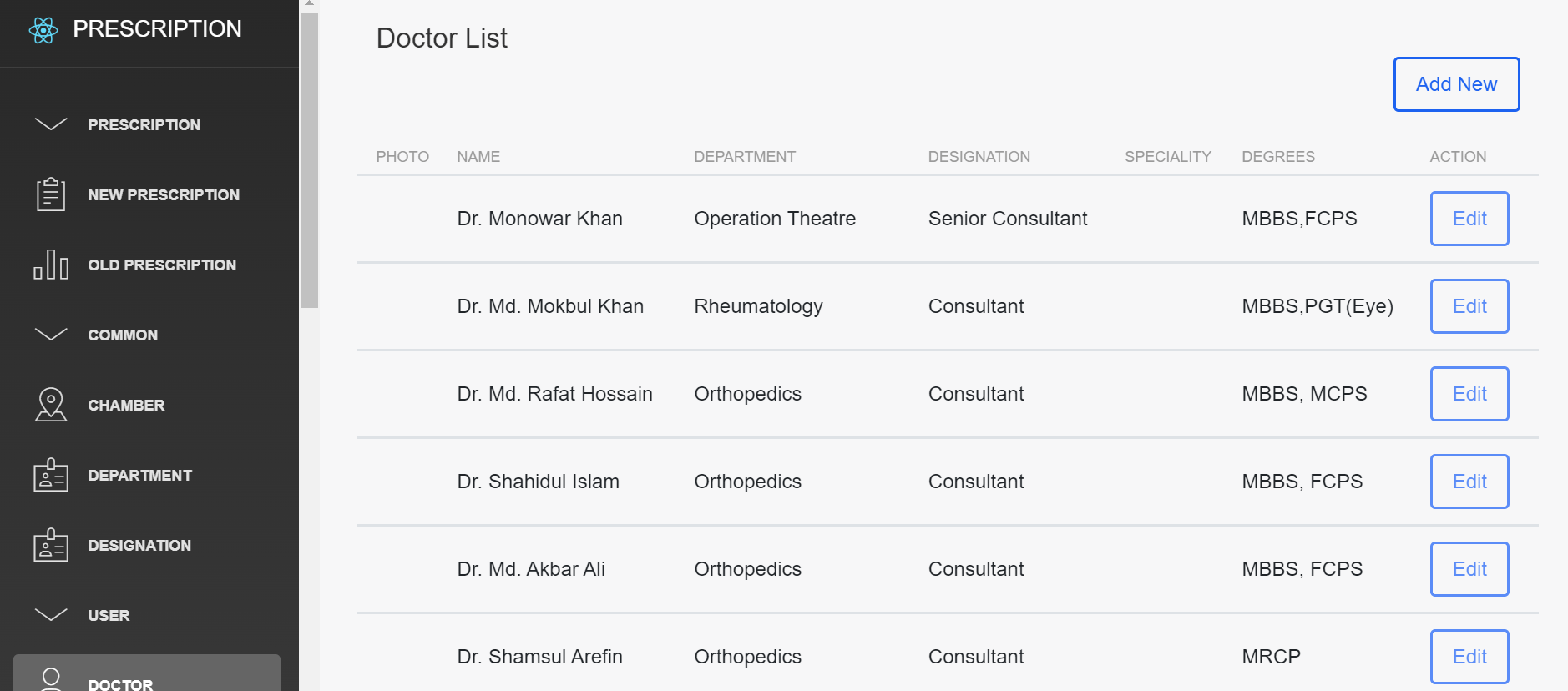


Figure 3.3: Doctor add page design

3.5.1.4 Add Patient Page:

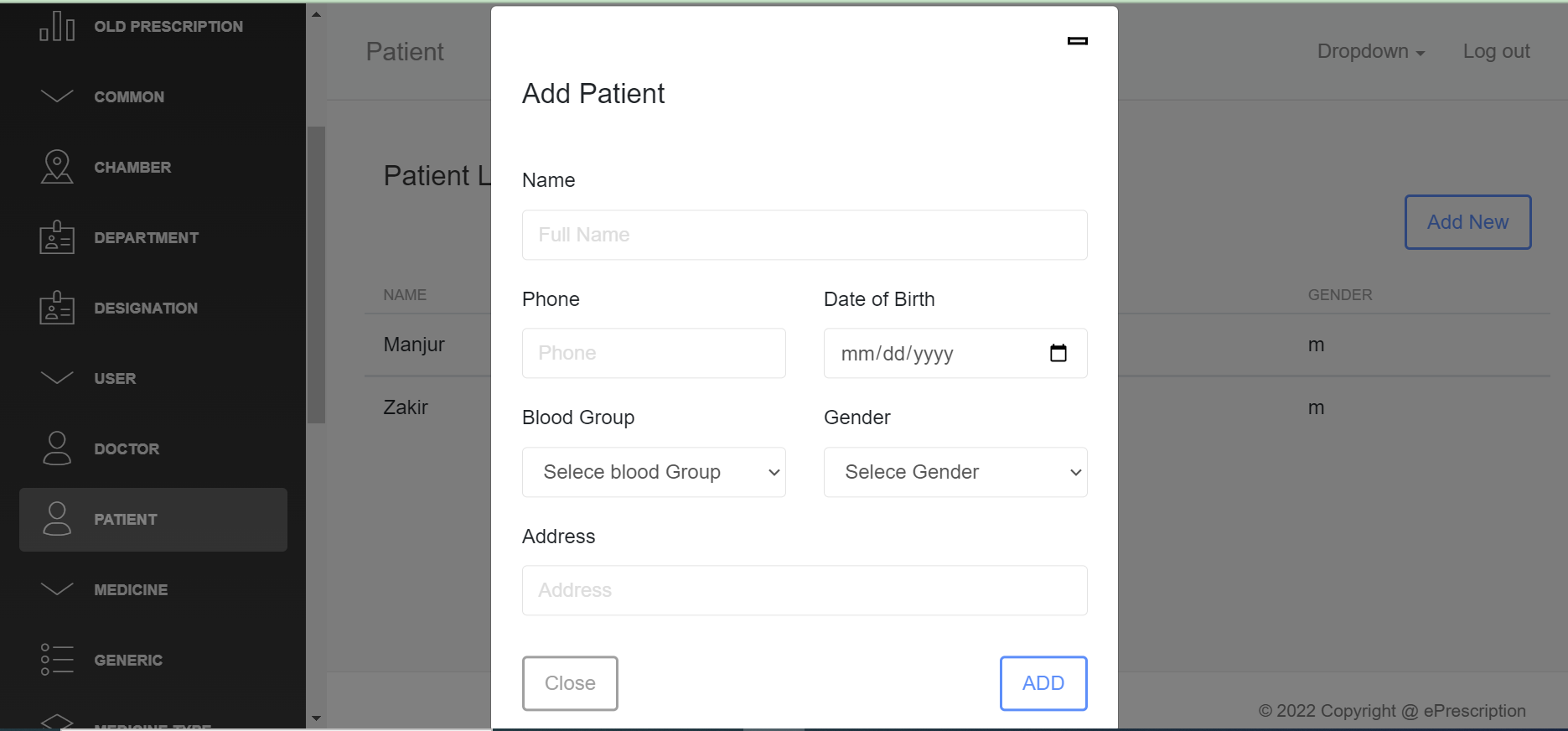


Figure 3.6: patient add page design

3.5.1.5 Add Medicine:

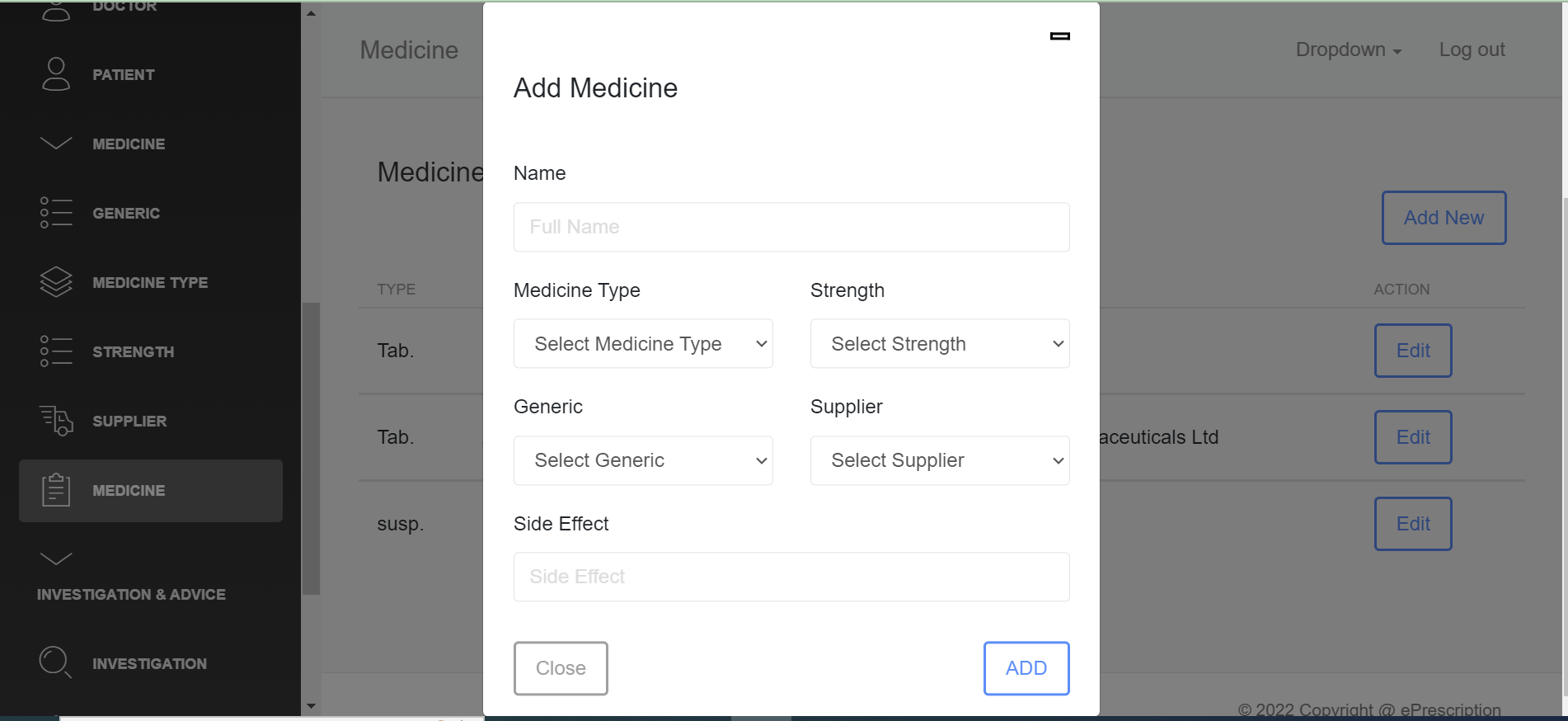


Figure 3.7: Medicine Add page design

3.5.2.3 Add Investigation Page:

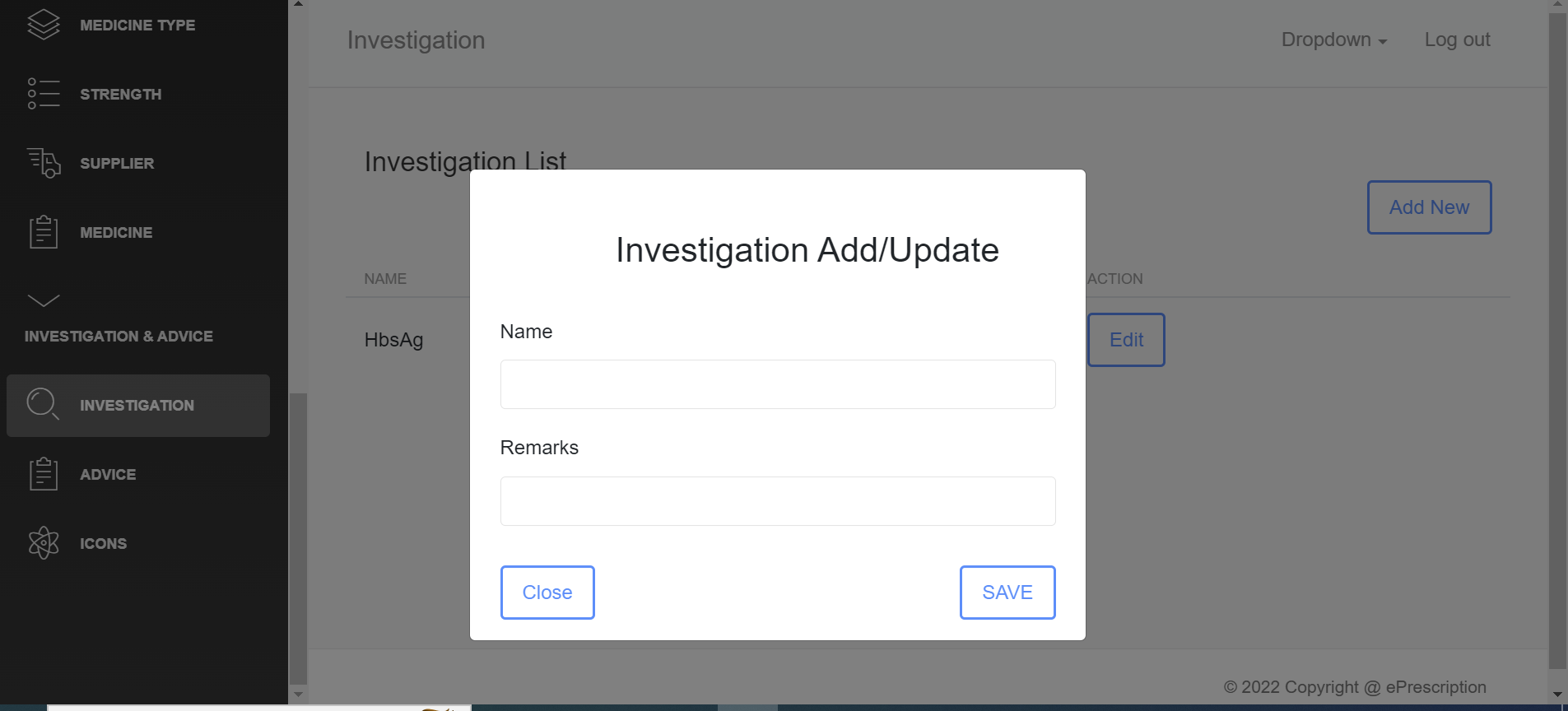


Figure 3.9: Investigation insert page design

3.5.2.4 Prescription Page:

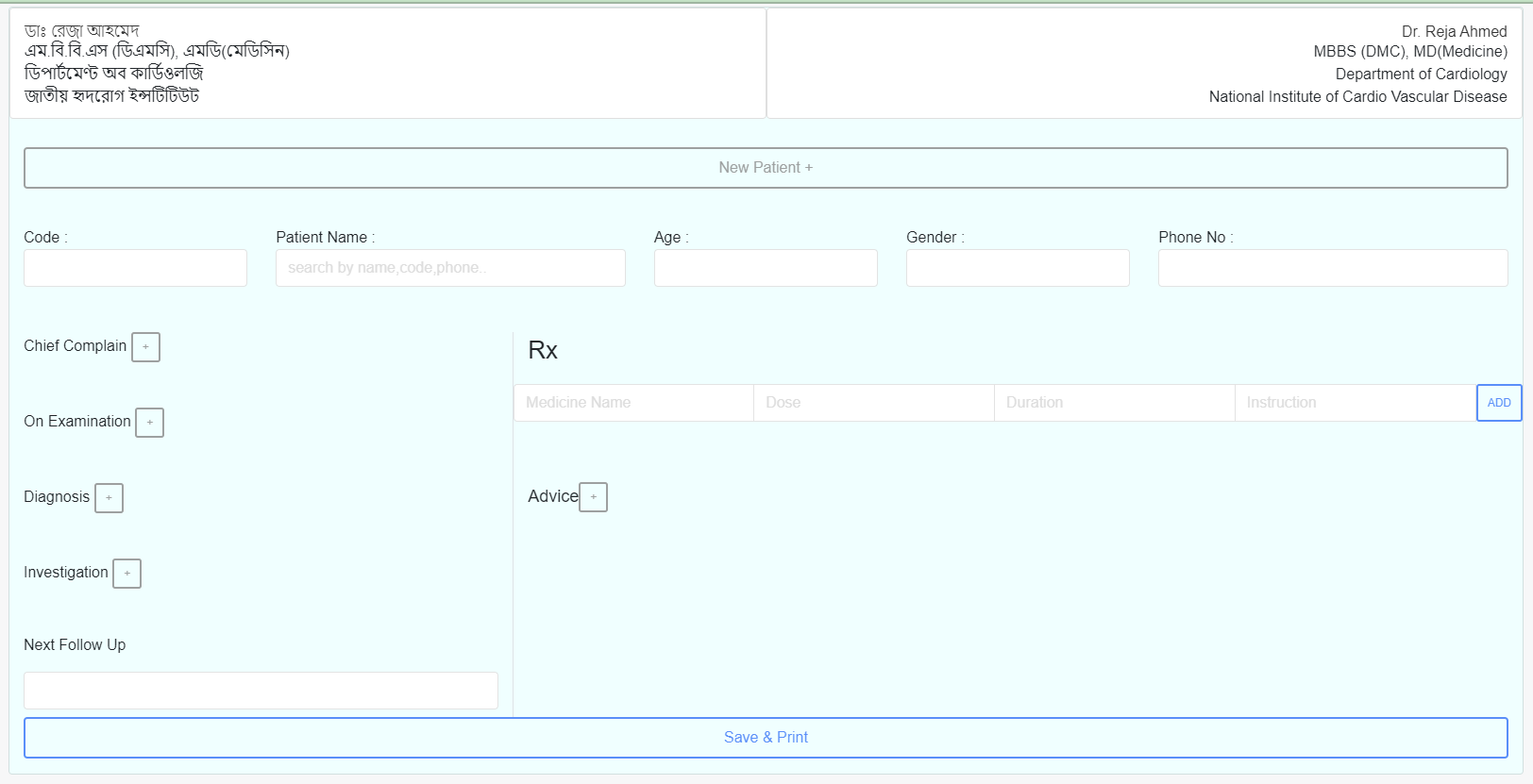


Figure 3.1Prescription page design

1. Testing

**4.1 Overview**

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

**4.2 Scope**

Testing depends on the source code but reviewing requirements and developing test cases is independent from the developed code. Finding bugs in the Software is the task of testers. Developers are only responsible for the specific component or area that is assigned to them but testers understand the overall workings of the software, what the dependencies are and what the impacts of one module on another module are. This section of the document describes the overall testing activities with test case.

**4.3 Appendix:**

**White box testing:**

4.3.1 **user login system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, Null | Login | Message display **Error** | Pass |
| 02. | Null, 123456789 | Login | Message display **Error** | Pass |
| 03. | 123456789, Null | Login | Message display **Error** | Pass |
| 04. | 123456789, admin | Login | Message display **Error** | Pass |
| 05. | admin,Null | Login | Message display **Error** | Pass |
| 06. | Null, admin | Login | Message display **Error** | Pass |
| 07. | 123456789, 123456789 | Login | Message display **Error** | Pass |
| 08. | @$%#\*&!~(){}[], admin | Login | Message display **Error** | Pass |
| 09. | @$%#\*&!~(){}[], @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 10. | admin, @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 11. | @$%#\*&!~(){}[], 123456789 | Login | Message display **Error** | Pass |
| 12. | 123456789, @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 13. | admin, @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 14. | admin, 123456789 | Login | Message display **Success** | Pass |
| 15. | admin, admin123456789 | Login | Message display **Success** | Pass |
| 16. | admin, 123456789admin | Login | Message display **Success** | Pass |
| 17. | ADMIN, 123456789 | Login | Message display **Success** | Pass |
| 18. | ADMIN, admin123456789 | Login | Message display **Success** | Pass |
| 19. | ADMIN, 123456789admin | Login | Message display **Success** | Pass |
| 20. | ADMIN, ADMIN | Login | Message display **Success** | Pass |
| 21. | admin, ADMIN 123456789 | Login | Message display **Success** | Pass |
| 22. | admin, 123456789ADMIN | Login | Message display **Success** | Pass |
| 23. | ADmin, AdminN | Login | Message display **Success** | Pass |

Table 4: white box testing of "**login system**"

4.3.2 **Configiration insert system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, Config name | add | Message display **Error** | Pass |
| 02. | Null, Null, Null, captcha | add | Message display **Error** | Pass |
| 03. | Null, Null, Config name cd, | add | Message display **Error at least 3 character** | Pass |
|  |  |  |  |  |

4.3.3 **Doctor insert system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, Doctor name | add | Message display **Error** | Pass |
| 02. | Null Doctor | add | Message display **Error** | Pass |
| 03. | Doctor name 3 character | add | Message display **Error at least 3 character** | Pass |

4.3.4 **patient insert system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, patient | add | Message display **Error** | Pass |
| 02. | Null city | add | Message display **Error** | Pass |
| 03. | city and patient name less than 3 character | add | Message display **Error at least 3 character** | Pass |

4.3.5 **Insert prescription system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null doctor,patient | create | Message display **Error** | Pass |
| 02. | Null, Medicine | create | Message display **Error** | Pass |
| 03. | Null 1st page | create | Message display **Error** | Pass |
| 04. | Null O/E | create | Message display **Error** | Pass |

4.3.6 **user sign up:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null name | Submit | Message display **Error** | Pass |
| 02. | Null, Email | Submit | Message display **Error** | Pass |
| 03. | Null password or password at least 6 character | Submit | Message display **Error** | Pass |
| 04. | Null address | Submit | Message display **Error** | Pass |
| 05. | Null country select | Submit | Message display **Error** | Pass |
| 07. | Null Gender | Submit | Message display **Error** | Pass |
| 08. | Null contact | Submit | Message display **Error** | Pass |

**Conclusion**:

I am pleased to submit the final Software documentation report on Online prescription. From this, the patients will get a clear and easy view of medicine names. To improve prescriptions efficiency, ePrescription is very essential. An online prescription system is more effective than paper based manual system. This document can be used effectively to maintain software development cycle. It will be very easy to conduct the whole project using it. Hopefully, this document can also help our junior BSSE batch students. We tried our best to remove all dependencies and make effective and fully designed document. We believe that reader will find it in order.