



Daffodil
International
University

“Health Care – A Client-side application developed in React.js for Online Prescription”

Prepared by:

Ashfaq Mahmud

ID: 161-35-1428

B.Sc. in Software Engineering

Daffodil International University

Supervised By

Md. Mushfiqur Rahman

Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

APPROVAL

This **Project** titled “**Health Care – A Client-side application developed in React.js for Online Prescription**”, submitted by **Ashfaq Mahmud, 161-35-1428** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman

Md Fahad Bin Zamal
Associate Professor

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Md. Shohel Arman
Lecturer

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

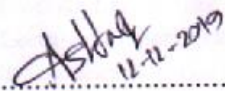
Prof Dr. Mohammad Abul Kashem
Professor

Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

External Examiner

DECLARATION

I hereby declare that I have taken this project under the supervision of **Md. Mushfiqur Rahman, Lecturer, Department of Software Engineering, Daffodil International University**, I also declare that neither this project nor any part of this has been submitted elsewhere for award of any degree.


.....

Ashfaq Mahmud

ID: 161-35-1428

Batch: 19th

Department of Software Engineering

Daffodil International University

Certified by:


.....

Md. Mushfiqur Rahman

Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

ACKNOWLEDGEMENT

It stands to reason that the completion of main project needs the support of many people. I take this opportunity to express our boundless thanks and commitment to each and every one, who helped us in successful completion of our main project. I am happy to acknowledge the help of all the individuals to fulfill our attempt.

Most importantly I wish to express wholehearted obligation to the god-like **ALLAH** for his thoughtful steady care and unselfishness showered euphorically over us during this Endeavor.

I would like to take the opportunity to express our gratitude to **Md. Mushfiqur Rahman, Lecturer, Department of Software Engineering, Daffodil International University**. Although he was always loaded with several other activities, he gave us more than enough time in this work. He not only gave us time, but also gives us proper guidance and valuable advice whenever I faced with some difficulties. His comments and guidance helped us in preparing our project report.

Finally, I thank to our teachers for supporting us throughout all our studies at Daffodil International University.

ABSTRACT

This project entitled “**Health Care**” is a web based application to communicate between doctor & patient. Health Care is the practice of get a prescription between a doctor & a patient. With this thought I made this project of Health Care. Here a user who is a doctor can suggest medicine for a patient. And another user who is a patient can ask about their problems to a doctor. Anyone can check his/her BMI. It will save money. It can be a helpful system for people.

I used React.JS for client side, for style bootstrap to create my project.

LIST OF CONTENTS

Title Pages	i
Declaration.....	ii
Acknowledgement.....	iii
Abstract.....	iv
List of Contents.....	v
Table of Content.....	(vi-viii)
List of Figures.....	(ix-x)
List of Tables.....	(xi-xii)

Table of Contents

Chapter 1: Introduction	1
1.1 Project Overview.....	1
1.2 Purpose & Scope.....	1
1.2.1 Background / Problem Statement.....	2
1.2.2 Benefits & Beneficiaries.....	2
1.2.3 Goals.....	2
1.3 Stakeholders.....	2
1.4 Proposed Solution.....	2
1.5 Project Schedule.....	(2-4)
1.6 Gantt Chart.....	5
1.7 Release Plan/Milestone.....	5
Chapter 2: Software Requirement Specification.....	6
2.1 Functional & Non-Functional Requirements List.....	(6-8)
2.2 Data Requirements.....	8
2.3 Performance Requirements.....	8
2.4 Dependability Requirements.....	9
2.4.1 Reliability Requirements.....	9
2.5 Maintainability and Supportability Requirements.....	9
2.6 Security Requirements.....	10
2.7 Operational and Environmental Requirements.....	10

Chapter 3: System Analysis

3.1	Use Case Diagram.....	11
3.2	Use Case Description.....	(12-18)
3.3	Activity Diagram.....	(18-23)

Chapter 4: System Design Specification

4.1	Sequence Diagram.....	(24-34)
4.2	Class Diagram.....	(34-36)
4.3	Database Design Diagram.....	37
4.4	Development Tools & Technology.....	38
4.4.1	User Interface Technology.....	38
4.4.1.1	React Framework.....	38
4.4.1.2	Bootstrap.....	38
4.4.1.3	Font Awesome.....	39

Chapter 5: System Testing

5.1	Testing Features.....	40
5.1.1	Features to be tested.....	40
5.1.2	Features not to be tested.....	40
5.2	Testing Strategies.....	40
5.3	Testing Environment.....	(40-41)
5.4	Test Cases.....	41
5.5 - 5.15	API.....	(42-54)

Chapter 6: User Manual

6.1	User Manual (type A user).....	(55-57)
6.2	User Manual (type B user).....	58
6.3	User Manual (type C user).....	(59-60)
6.4	User Manual (type D user).....	(61-62)
6.5	User Manual (type E user).....	(62-64)

Chapter 7: Project Summary

7.1	Github Link.....	65
7.2	Critical Evolution.....	65
7.3	Limitations.....	65
7.4	Obstacles & Achievements.....	66
7.5	Future Scope.....	66
	Reference.....	67

List of Figures

Figure 1.5.1: Gantt chart	5
Figure 3.1: Use case diagram	11
Figure 3.3.1: Activity diagram (Login & Registration)	19
Figure 3.3.2: Activity diagram (Doctor Profile)	19
Figure 3.3.3: Activity diagram (Doctor Professional Details)	20
Figure 3.3.4: Activity diagram (Doctor Education)	20
Figure 3.3.5: Activity diagram (Manage Prescription)	20
Figure 3.3.6: Activity diagram (Make Prescription)	21
Figure 3.3.7: Activity diagram (Manage Patients Profile)	21
Figure 3.3.8: Activity diagram (Manage Drug)	22
Figure 3.3.9: Activity diagram (Handle Medicine Brand)	22
Figure 3.3.10: Activity diagram (Handle Generic Brand)	23
Figure 4.1.1: Sequence Diagram for Login	24
Figure 4.1.2: Sequence Diagram for Registration	25
Figure 4.1.3: Sequence Diagram for Manage Doctor Profile	26
Figure 4.1.4: Sequence Diagram for Handle Doctors Professional Details	27
Figure 4.1.5: Sequence Diagram for Handle Doctors Educational Details	28
Figure 4.1.6: Sequence Diagram for Manage Prescription Details	29
Figure 4.1.7: Sequence Diagram for Make Prescription	30
Figure 4.1.8: Sequence Diagram for Manage Patients Profile	31
Figure 4.1.9: Sequence Diagram for Manage Drug	32
Figure 4.1.10: Sequence Diagram for Handle Medicine Brand	33
Figure 4.1.11: Sequence Diagram for Handle Medicine Brand	34
Figure 4.2.1: Class Diagram for model classes	35
Figure 4.2.2: Class Diagram for controller classes	36

Figure 4.3: Database Design Diagram	37
Figure 6.1: User Manual (Type A User)	(51-53)
Figure 6.2: User Manual (Type B User)	(53-54)
Figure 6.3: User Manual (Type C User)	(54-56)
Figure 6.4: User Manual (Type D User)	(56-57)
Figure 6.5: User Manual (Type E User)	(58-59)

List of Tables

Table 1.5.1:	Finding the Idea Proposal	3
Table 1.5.2:	Requirements Gathering	3
Table 1.5.3:	Logical System Design	4
Table 1.5.4:	Development Phase	4
Table 2.1.1:	Registration	6
Table 2.1.2:	Login	7
Table 2.1.3:	Admin Profile	7
Table 2.1.4:	Add Information	7
Table 2.1.5:	Check Information	8
Table 3.2.1:	Use Case Description for Login	11
Table 3.2.2:	Use Case Description for Register	11
Table 3.2.3:	Use Case Description for Manage Doctor Profile	12
Table 3.2.3.1:	Use Case Description for Handle Doctor Professional Details	12
Table 3.2.3.2:	Use Case Description for Handle Doctor Education Details	13
Table 3.2.4:	Use Case Description for Manage Prescription Details	13
Table 3.2.5:	Use Case Description for Make Prescription	14
Table 3.2.6:	Use Case Description for Manage Patients Profile	(14-15)
Table 3.2.7:	Use Case Description for Manage Drug	15
Table 3.2.7.1:	Use Case Description for Handle Brand	(15-16)
Table 3.2.7.2:	Use Case Description for Handle Generic	16
Table 5.5:	Register API.....	(39-40)
Table 5.6:	Login API.....	(40-41)
Table 5.7:	Register API.....	41
Table 5.8:	Doctor_P API.....	42
Table 5.9:	ProfessionalDetails API.....	43

Table 5.10:	DocEducation API.....	44
Table 5.11:	Patient_P API.....	45
Table 5.12:	Prescription API.....	(46-47)
Table 5.13:	Prescription Detail API.....	47
Table 5.14:	Drug API.....	48
Table 5.15:	Brand API.....	49
Table 5.16:	Generic API.....	(49-50)

Chapter-1

1. Introduction

Health Care, it is basically a web system. This system is for all people. In our project, I will try to make a system where people can take an online prescription from doctors. Here patients can registrar & Login. They can register so that they can get a solution from any doctors. And doctors can also registrar & Login. They can provide a solution for patients so that the patients can relief from their problems. Basically I will try to do this and next I have more plans for our project.

Users ought to counsel this record and its corrections as the main wellspring of prerequisites for the undertaking. They ought not to think about any necessities articulations, composed or verbal as substantial until they show up in this record or its correction.

1.1 Overview

This web system can utilize everybody the individuals who are enrolled. Individuals can register and login; at that point they can record their issues in the system and submit it. At that point a specialist will restore a remedy to the patient. With the goal that the patients get arrangements.

1.2 Purpose & Scope

This web system can use everyone those who are registered. People can registrar & login, then they can register in the system and submit it. Then they will go to a doctor & doctor will return a prescription to the patient. So that the patients get solutions. Using this system people is helped. Govt. can also use this system that they can understand about the records of problems & can analysis by using our system. They can also know about the drugs. Which drugs are needed for the disease? They can understand the problems of the disease and make new good quality drugs.

There are some issues also like, patients can't understand doctors hand-writing. So this system will help to understand medicines name easily.

Basically our purpose of building the system is to make a platform for all types of patients & doctors for free treatment.

1.2.2 Benefits & Beneficiaries

The benefits of this system are people can easily get solutions of their problems. They need not go anywhere for prescription. They can ask any problems & get medicines name for their problems. Doctors need not to use anything to write prescription. They can easily write down their prescription & they can also know about medicines. Government can also take benefits from this system. They can analysis about the disease and can make new good quality drugs. So that people can relief from diseases & can lead a better life.

1.2.3 Goal

Our goal is to provide free prescription for patients. That they can easily get treatment from Anywhere & any place. There are some issues also like, patients can't understand doctors Hand-writing. So this system will help to understand medicines name easily. I have a plan to explore this project more & make it reliable. Doctors need not to use anything to write prescription. They can easily write down their prescription & they can also know about medicines. So that my users feel free to use the system.

1.3 Stakeholders

Basically those who are using our web system they are our stakeholders. Doctors, Patients & Admin are stakeholders.

1.4 Proposed Solution

I proposed for our Health Care system that I will try to make a Prescription part so that patients can easily get their treatments & I have done this part. And I tried to make a BMI calculator in our system so that anyone check their BMI. Next I will try to make an Appointment system so that patients can contact with doctors, can get serials to meet up with doctors.

1.5 Project Schedule

Though our project is a big project so this project will take a time. But I am trying to finish one part from the whole system for our Final Defense. So I used 5 months to make our system. Here's the Gantt Chart. So that I can easily find out when I have finished our work.

1.5.1 Finding Idea Proposal:

Serial	Work Description	Start Day	End Day	Total Day
1	Idea Finding With Supervisor	01-09-2019	04-09-2019	3
2	Feasibility study With Supervisor	05-09-2019	07-09-2019	2
3	Feature Discussion With Supervisor	08-09-2019	10-09-2019	2
4	Work Flow Maintenance	11-09-2019	16-09-2019	5
5	SDLC Selection	17-09-2019	19-09-2019	2

1.5.2 Requirements Gathering:

Serial	Work Description	Start Day	End Day	Total Day
1	System Flow sketch	20-09-2019	28-09-2019	8
2	Requirement gathering for proposed system	29-09-2019	05-10-2019	7
3	Requirement Collection	06-10-2019	08-10-2019	2
4	SRS	09-10-2019	11-10-2019	2
5	All requirement and Information	12-10-2019	16-10-2019	4

1.5.3 Logical System Design:

Serial	Work Description	Start Day	End Day	Total Day
1	Use case Diagram Design	17-10-2019	18-10-2019	1
2	Activity Diagram	19-10-2019	20-10-2019	1
3	Schema Diagram	21-10-2019	22-10-2019	1
4	Class Diagram	23-10-2019	24-10-2019	1

1.5.4 Development Phase:

Serial	Work Description	Start Day	End Day	Total Day
1	Build User Module	25-10-2019	01-11-2019	7
2	Build Prescription Module	02-11-2019	09-11-2019	7
3	Build Admin Module	10-11-2019	18-11-2019	8
4	Database Integration	19-11-2019	21-11-2019	2
5	Live Streaming Implementation	22-11-2019	25-11-2019	3
6	Refactoring Code	26-11-2019	05-12-2019	10
7	Intend All Code	06-12-2019	11-12-2019	5

1.6 Gantt Chart

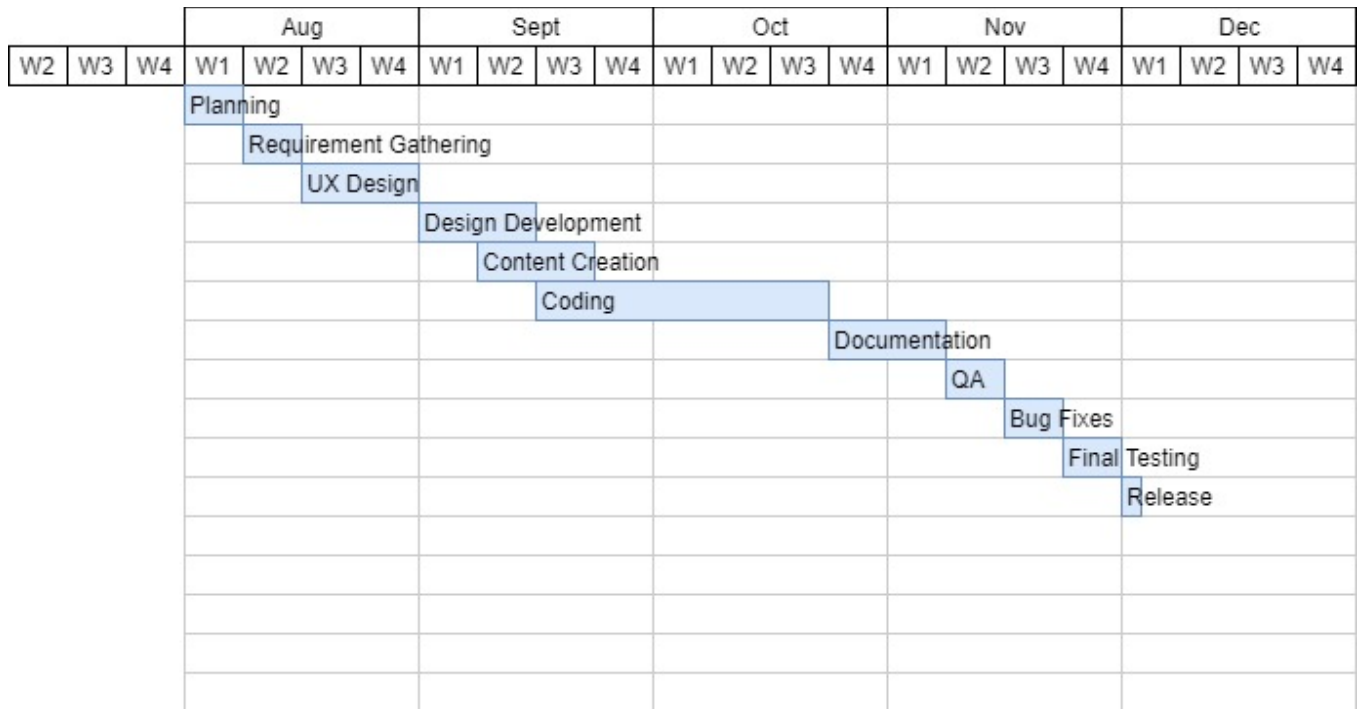


Figure 1.1: Grant Chart

1.7 Release Plan / Milestone

Release plan-1 : I will try to release our system at 14 December 2019.

Because our system is not a small project. I can't finish it within a short time. So I will update it & release the update version in next time. In our next release date I will try to use an Appointment system & will try to make the system more reliable

Chapter-2

2. Software Requirement Specification

2.1 Functional & Non Functional Requirement List

Functional Requirements List :

Before identification of the requirements I needed the comprehensive engagement and lighting quick coordination with the stakeholders. This accelerates the entire requirements management process by orchestrating the flow of information and processes across different team members and stakeholders. Again, this is combined with hybrid agile and waterfall development methodologies and tools. Flexible workflows and automatic notifications streamline communication, review, and approval of requirements across stakeholders, while common metrics and dashboards ensure everyone is on the same page. So, the listed requirements go with all the previous processes.

2.1.1 Registration:

Here how-to registration.

Requirement No.	Requirement
R-1	Define Full name (User name).
R-2	Define E-mail address.
R-3	Define Password (Strong)
R-4	Attach Submit button.
R-5	Check the mail to verify user account.

2.1.2 Login:

Here, the requirements are based on the task of login system by the users including administrator and passenger.

Requirement No.	Requirement
L-1	While login match the username with user type
L-2	User will get the functionalities of his/her type
L-3	Login time should be stored in the log file

2.1.3 Admin Profile:

Here how to manage admin profile.

Requirement No.	Requirement
A-1	Admin will login using email & password
A-2	Admin will add medicines name
A-3	Admin can add Generic & Brand name of medicines.

2.1.4 Add Information:

Here how to add information.

Requirement No.	Requirement
-----------------	-------------

AI-1	Doctor & Patient both can add their personal information.
AI-2	Doctors need to add their educational information.
AI-3	Doctors need to add their specialty information.

2.1.5 Check Information:

Requirement No.	Requirement
CI-1	Patient can check their prescription what doctor suggest them.
CI-2	Doctors can check their information & can add edit.
CI-3	Then all update is saving in the database

Non-Functional Requirements List :

- 1) Use valid E-mail address.
- 2) Use of Strong password.
- 3) Use valid Contact Number.
- 4) Don't use any negative values if it's not necessary. (Like: Don't use negative values for age, mobile number, Visiting Number etc.)

2.2 Data Requirement

Data requirements refers those data which are needed to build the system model. For my project I need to focus on some points such as :

- 1) Doctor Personal Information.
- 2) Patient Personal Information.
- 3) Drugs Information.
- 4) Prescription Information.

2.3 Performance Requirement

Server software does not require any special hardware other than the minimum hardware required for running enterprise OS. Extra disk storage will be required for archives and electronic documents. Increases of memory enables efficient query processing, which is required for quick bibliographic search. Minimum processors 1.0 Ghz, at least 4 GB RAM and 200 GB hard disk is recommended for the server. Client machine with recommended hardware required for desktop operating system and web browser.

2.4 Dependability Requirement

2.4.1 Reliability Requirement

Reliability is an important non-functional requirement for most software products so a software requirements specification (SRS) should contain a reliability requirement, and most do.

Software reliability can be a more difficult concept to grasp. A software product will fail under certain conditions, with certain inputs, and given the same inputs and conditions will fail every time until the cause of the failure is corrected. So, the reliability of a software product is more about the random discovery of faults resulting from various inputs with the system in various states. Although for small and simple systems it may be theoretically possible to test every combination of states and inputs, for a system of any size and complexity this is not feasible. The random nature of the fault discovery process means I must use probabilities when I refer to software reliability requirements and testing.

So I tried to make our software more reliable so that users can easily use our system & get better service from us.

2.5 Maintainability & Supportability Requirement

At least one backup server with same configuration as in main server is also recommended for fault tolerance and better performance. Separate storage (with backup) for database, electronic document, and manuscript is also recommended. Multiple computing nodes with the storage are required for high availability and to enhance the performance of the application. Again, after a certain period the preliminary manuscript files and other files related with that can be deleted manually from the database to increase the performance.

2.6 Security Requirement

Each time there is a security violation. For this reason when user will register, user will receive a mail & have verify this for login. Otherwise user can't login to the main option.

2.7 Operational & Environmental Requirement

The operational requirement definition process includes the following activities :

- 1) Identify stakeholders those who will or have interest in our system. So mainly our stakeholders are Doctors & patients. They should have Register & Login for their other activities.
- 2) Establish measure of effectiveness and suitability, so that users can easily access & can use our system.

A system reliability is dependent upon a stable environment. The design of the environmental control system for your data center must ensure that each system can operate reliably while remaining within the range of its operating specifications. So I tried to make our systems environment more reliable so that users can easily access to the system without any delay.

Chapter 3: System Analysis

3.1 Use Case Diagram for Prescription

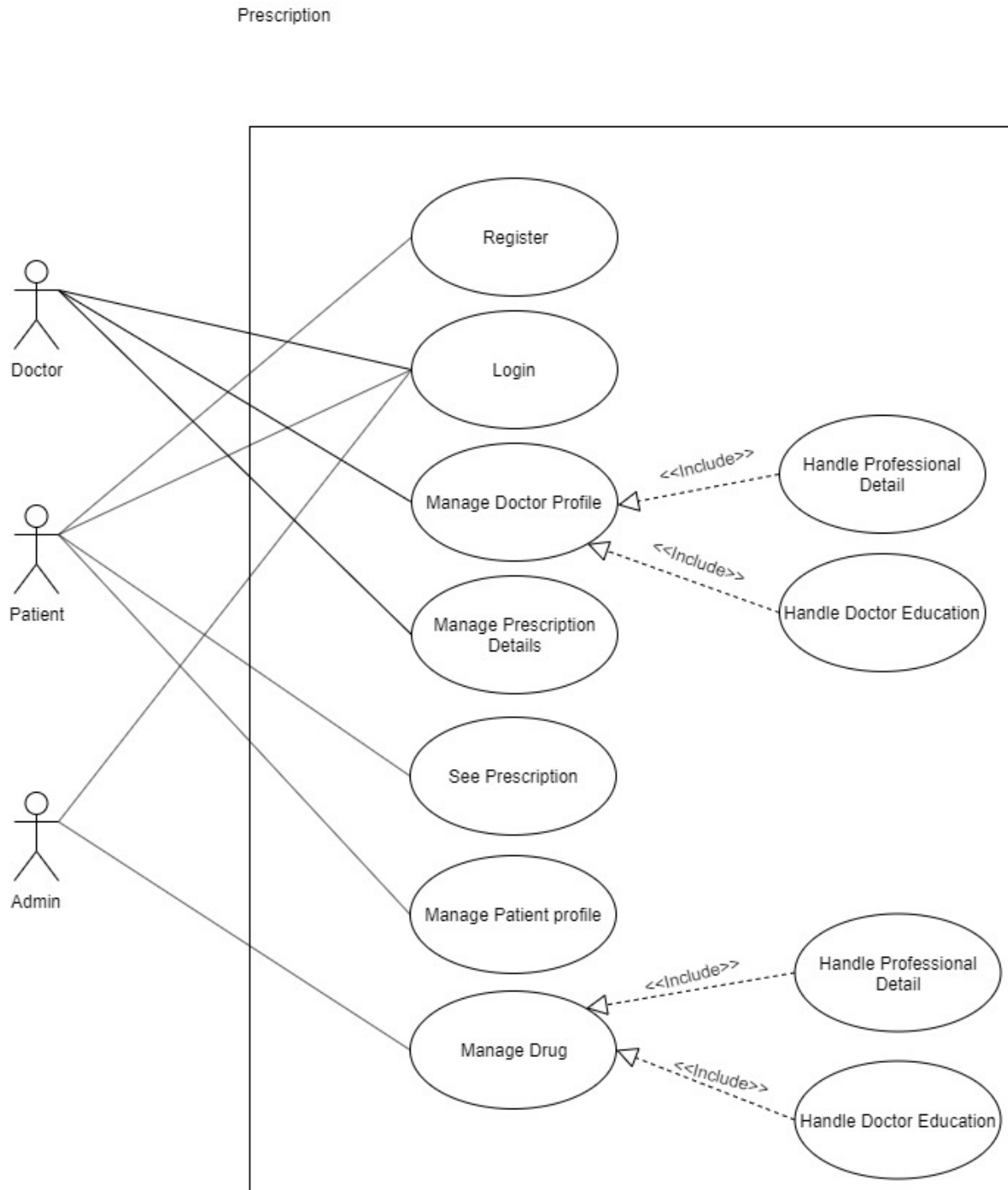


Figure 3.1: Use Case Diagram for Prescription

3.2 Use Case Description

3.2.1 Use Case Description for Login

Use Case No.	1
Use Case Name	Login
Actor	Doctor, Patient and Admin
Description	Allowing Doctor, Patient and Admin to login to the system..
Precondition	Doctor, Patient and Admin should remain in the login page. Required registration and email verification.
Trigger	Click the “Login” Button
Flow of Events	<ol style="list-style-type: none"> 1. Two text fields to give input of the Email and password respectively. 2. Write the username and password on that field. 3. Then Click the Login button.
Post Condition	Doctor, Patient and Admin are logged into the system at a different time.

3.2.2 Use Case Description for Register

Use Case No.	2
Use Case Name	Register
Actor	Doctor
Description	Allowing Doctor and Patient to registration to the system.
Precondition	Doctor and Patient should remain in the registration page. Required username, email and password.
Trigger	Click the “Submit” Button
Flow of Events	<ol style="list-style-type: none"> 1. Three text fields to give input of the username, email and password respectively. 2. Write the username, email and password on that field. 3. Then Click the submit button. 4. System will sends email on your provided Gmail address. 5. Then doctor click the confirm link. 6. Redirect to Login page.
Post Condition	Doctor and Patient are registered into the system at a different time.

3.2.3 Use Case Description for Manage Doctor Profile

Use Case No.	3
Use Case Name	Manage Doctor Profile
Actor	Doctor
Description	Allowing Doctor to manage his personal information to the system. Doctor can do insert, edit, delete and show data.
Precondition	Doctor should remain in the profile page. Required full name, gender, Date of birth, mobile and additional email.
Trigger	Click the “Insert, Update, Delete and Show” Button
Flow of Events	<ol style="list-style-type: none"> 1. Five text fields to give input of the full name, gender, Date of birth, mobile and additional email respectively. 2. Write the full name, gender, Date of birth, mobile and additional email on that field. 3. Then Click the submit button. 4. When edit your data click edit button 5. Re-enter your data into the text fields which is right. 6. Then Click the update button. 7. When delete your data then click delete button. 8. When show your data then click “Back to List” button.
Post Condition	Doctor successfully can doing insert, update, delete and show his personal data into the system.

3.2.3.1 Use Case Description for Handle Professional details

Use Case No.	3.1
Use Case Name	Handle Professional details
Actor	Doctor
Description	Allowing Doctor to handle his Professional information to the system. Doctor can do insert, edit, delete and show data.
Precondition	Doctor should remain in the Professional details page. Required specialization.
Trigger	Click the “Insert, Update, Delete and Back to List” Button
Flow of Events	<ol style="list-style-type: none"> 1. One text field and one dropdown list to give input of the doctor name and specialization respectively. 2. Write the doctor name and specialization on that field. 3. Then Click the submit button. 4. When edit your data click edit button. 5. Re-enter your data into the text fields which is right.

6. Then Click the update button.
7. When delete your data then click delete button.
8. When show your data then click “Back to List” button.

Post Condition Doctor successfully can doing insert, update, delete and show his Professional data into the system.

3.2.3.2 Use Case Description for Handle Doctor Education

Use Case No.	3.2
Use Case Name	Handle Doctor Education
Actor	Doctor
Description	Allowing Doctor to handle his educational information to the system. Doctor can do insert, edit, delete and show data.
Precondition	Doctor should remain in the docEducation page. Required Institute Name, Degree and Passing Year.
Trigger	Click the “Insert, Update, Delete and Back to List” Button
Flow of Events	<ol style="list-style-type: none"> 1. Three text fields and one dropdown list to give input of the doctor name, Institute Name, Degree and Passing Year respectively. 2. Write the doctor name, Institute Name, Degree and Passing Year on that field. 3. Then Click the submit button. 4. When edit your data click edit button. 5. Re-enter your data into the text fields which is right. 6. Then Click the update button. 7. When delete your data then click delete button. 8. When show your data then click “Back to List” button.
Post Condition	Doctor successfully can doing insert, update, delete and show his educational data into the system.

3.2.4 Use Case Description for Manage Prescription Detail

Use Case No.	4
Use Case Name	Manage Prescription Detail
Actor	Doctor
Description	Allowing Doctor to manage his prescription detail to the system. Doctor can do insert, edit, delete and show data.
Precondition	Doctor should remain in the prescription detail page. Required Doze and Duration.
Trigger	Click the “Insert, Update, Delete and Back to List” Button

Flow of Events	<ol style="list-style-type: none"> 1. Two text fields and two dropdown lists to give input of the prescription id, drug id, doze and duration respectively. 2. Write prescription id, drug id, doze and duration on that field. 3. Then Click the submit button. 4. When edit your data click edit button. 5. Re-enter your data into the text fields which is right. 6. Then Click the update button. 7. When delete your data then click delete button. 8. When show your data then click “Back to List” button.
-----------------------	--

Post Condition	Doctor successfully can doing insert, update, delete and show his Prescription Detail data into the system.
-----------------------	---

3.2.5 Use Case Description for Make Prescription

Use Case No.	5
Use Case Name	Make Prescription
Actor	Doctor
Description	Allowing Doctor to make patient prescription to the system. Doctor can do insert, edit, delete and show data.
Precondition	Doctor should remain in the prescription page. Required Date, Blood presser, Temperature, Weight, Visit no, Next Date and Indication.
Trigger	Click the “Insert, Update, Delete and Back to List” Button
Flow of Events	<ol style="list-style-type: none"> 1. Seven text fields and two dropdown lists to give input of the Patient name, Doctor name, Date, Blood presser, Temperature, Weight, Visit no, Next Date and Indication respectively. 2. Write Patient name, Doctor Name, Date, Blood presser, Temperature, Weight, Visit no, Next Date and Indication on that field. 3. Then Click the submit button. 4. When edit your data click edit button. 5. Re-enter your data into the text fields which is right. 6. Then Click the update button. 7. When delete your data then click delete button. 8. When show your data then click “Back to List” button.
Post Condition	Doctor successfully can doing insert, update, delete and show his Prescription data into the system.

3.2.6 Use Case Description for Manage Patient Profile

Use Case No.	6
---------------------	---

Use Case Name	Manage Patient Profile
Actor	Patient
Description	Allowing Patient to manage his personal information to the system. Patient can do insert, edit, delete and show data.
Precondition	Patient should remain in the profile page. Required full name, gender, Date of birth, mobile and additional email.
Trigger	Click the “Insert, Update, Delete and Show” Button
Flow of Events	<ol style="list-style-type: none"> 1. Eight text fields to give input of the Full Name, gender, Age, Height, Weight, Is Married, mobile and additional email respectively. 2. Write the Full Name, gender, Age, Height, Weight, Is Married, mobile and additional email on that field. 3. Then Click the submit button. 4. When edit your data click edit button 5. Re-enter your data into the text fields which is right. 6. Then Click the update button. 7. When delete your data then click delete button. 8. When show your data then click “Back to List” button.
Post Condition	Patient successfully can doing insert, update, delete and show his personal data into the system.

3.2.7 Use Case Description for Manage Drug

Use Case No.	7
Use Case Name	Manage Drug
Actor	Admin
Description	Allowing Admin to manage Drug information to the system. Admin can do insert, edit, delete and show data.
Precondition	Admin should remain in the Drug page. Required Strength and Type.
Trigger	Click the “Insert, Update, Delete and Back to List” Button
Flow of Events	<ol style="list-style-type: none"> 1. Two text fields and two dropdown lists to give input of the Generic Name, Brand Name, Strength and Type respectively. 2. Write the Generic Name, Brand Name, Strength and Type on that field. 3. Then Click the submit button. 4. When edit your data click edit button. 5. Re-enter your data into the text fields which is right. 6. Then Click the update button. 7. When delete your data then click delete button. 8. When show your data then click “Back to List” button.

Post Condition Admin successfully can doing insert, update, delete and show drug data into the system.

3.2.7.1 Use Case Description for Handle Brand

Use Case No. 7.1
Use Case Name Handle Brand
Actor Admin
Description Allowing Admin to handle brand information to the system. Admin can do insert, edit, delete and show data.
Precondition Admin should remain in the brand page. Required Brand Name.
Trigger Click the “Insert, Update, Delete and Back to List” Button
Flow of Events

1. One text field to give input of the brand name respectively.
2. Write the Brand Name on that field.
3. Then Click the submit button.
4. When edit your data click edit button.
5. Re-enter your data into the text fields which is right.
6. Then Click the update button.
7. When delete your data then click delete button.
8. When show your data then click “Back to List” button.

Post Condition Admin successfully can do insert, update, delete and show brand data into the system.

3.2.7.2 Use Case Description for Handle Generic

Use Case No. 7.2
Use Case Name Handle Generic
Actor Admin
Description Allowing Admin to handle generic information to the system. Admin can do insert, edit, delete and show data.
Precondition Admin should remain in the generic page. Required Generic Name.
Trigger Click the “Insert, Update, Delete and Back to List” Button
Flow of Events

1. One text field to give input of the generic name respectively.
2. Write the generic name on that field.
3. Then Click the submit button.
4. When edit your data click edit button.
5. Re-enter your data into the text fields which is right.
6. Then Click the update button.
7. When delete your data then click delete button.
8. When show your data then click “Back to List” button.

Post Condition Admin successfully can do insert, update, delete and show generic data into the system.

3.3 Activity Diagram

3.3.1 Activity Diagram for Login and Register

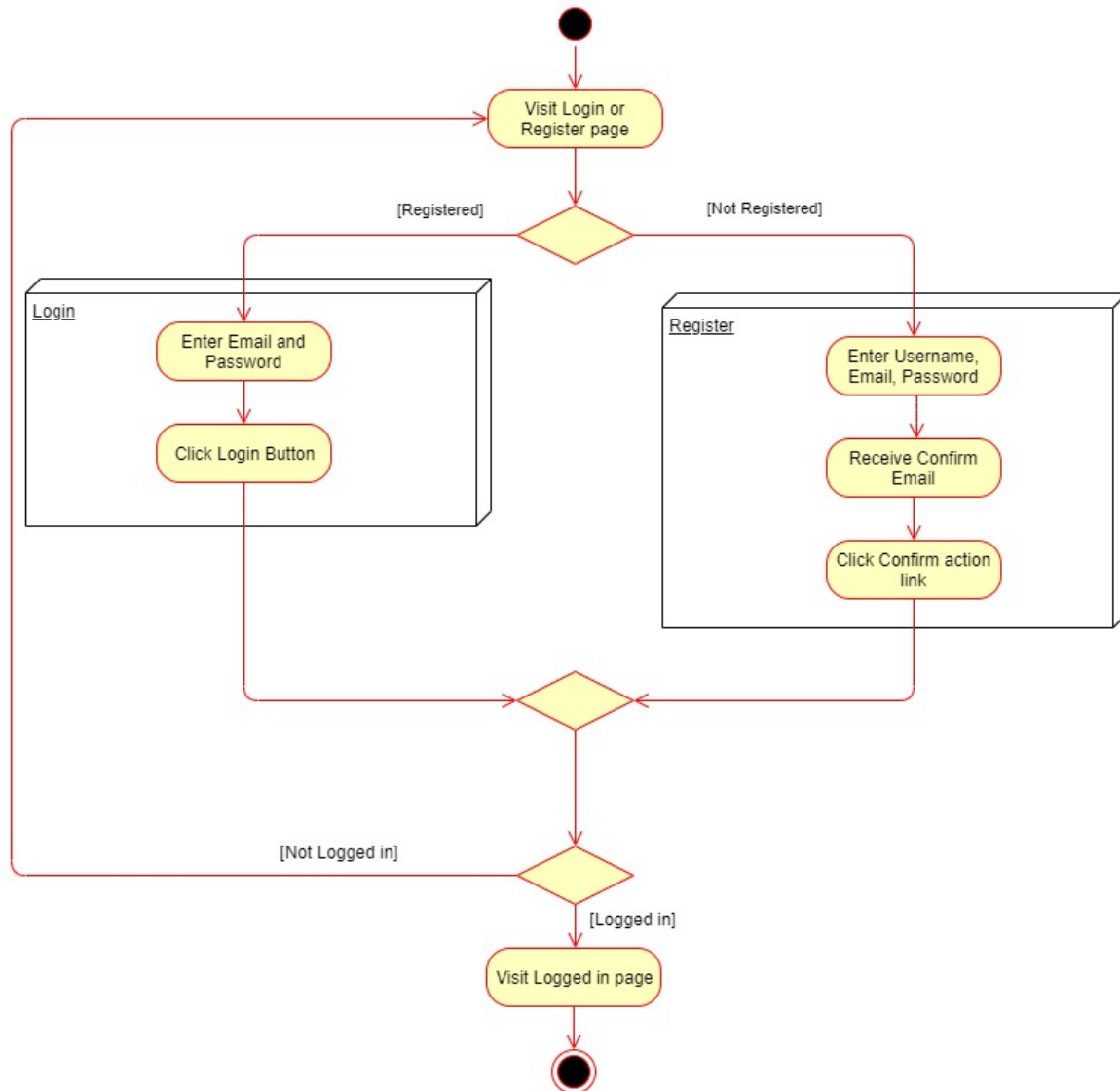


Figure 3.3.1: Activity Diagram for Login and Register

3.3.2 Activity Diagram for Manage Doctor Profile

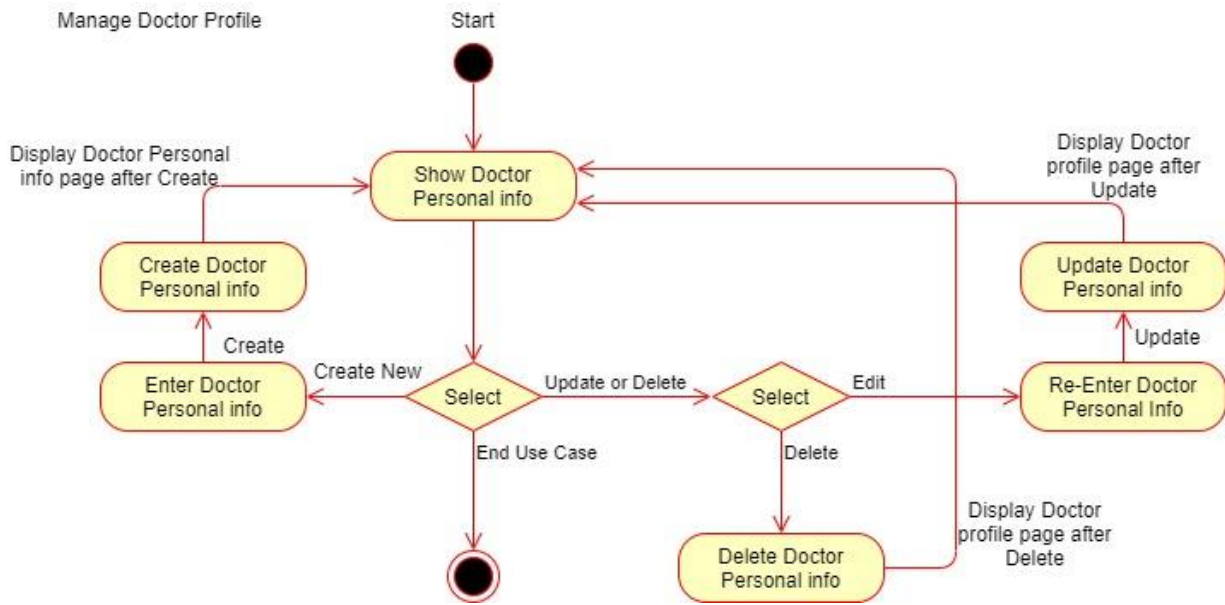


Figure 3.3.2: Activity Diagram for Manage Doctor Profile

3.3.3 Activity Diagram for Handle Professional details

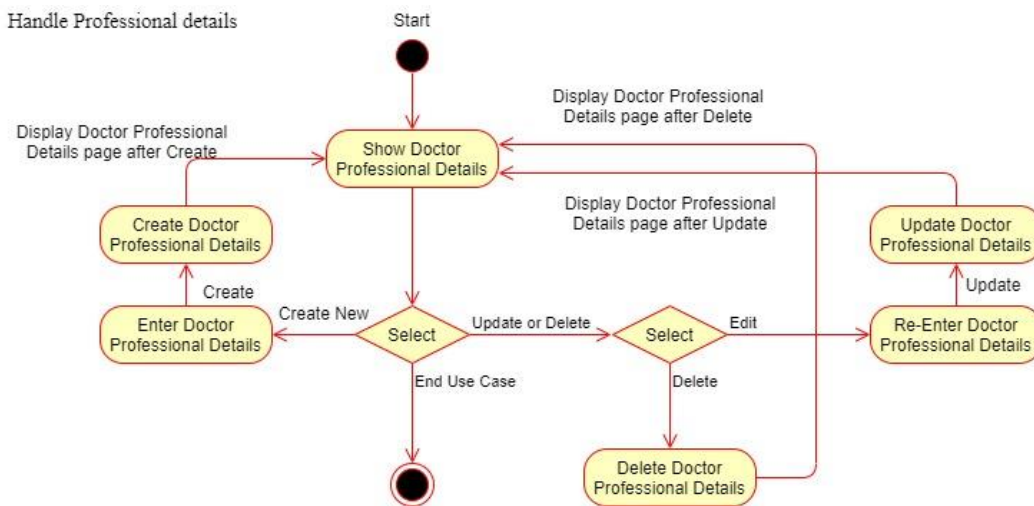


Figure 3.3.3: Activity Diagram for Handle Professional details

3.3.4 Activity Diagram for Handle Doctor Education

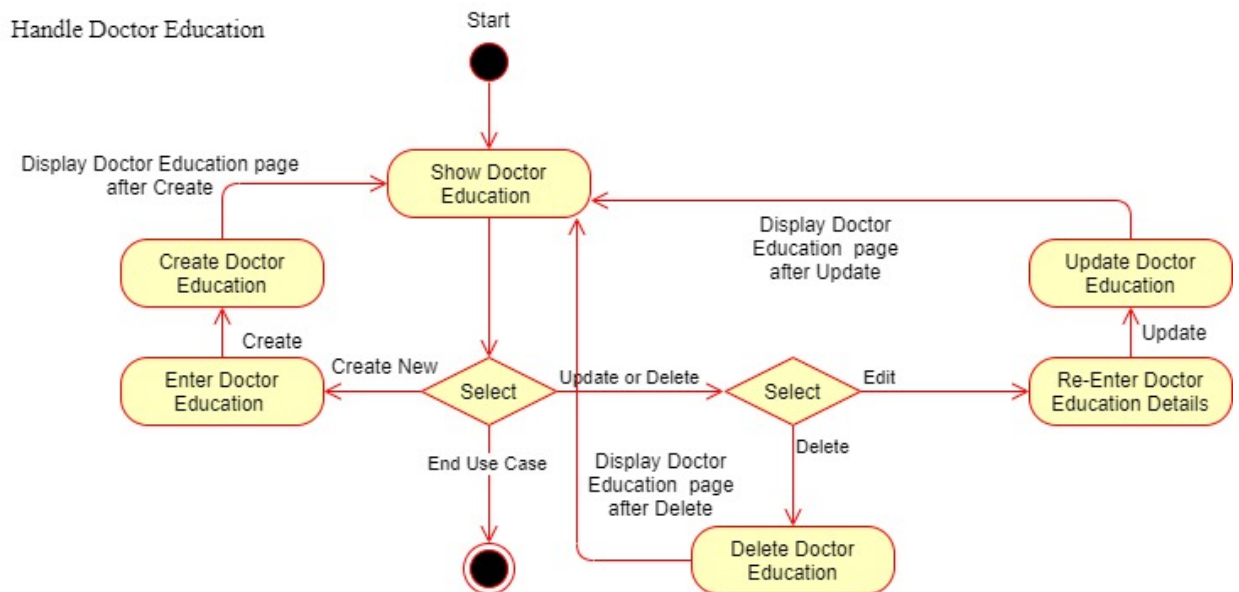


Figure 3.3.4: Activity Diagram for Handle Doctor Education

3.3.5 Activity Diagram for Manage Prescription Detail

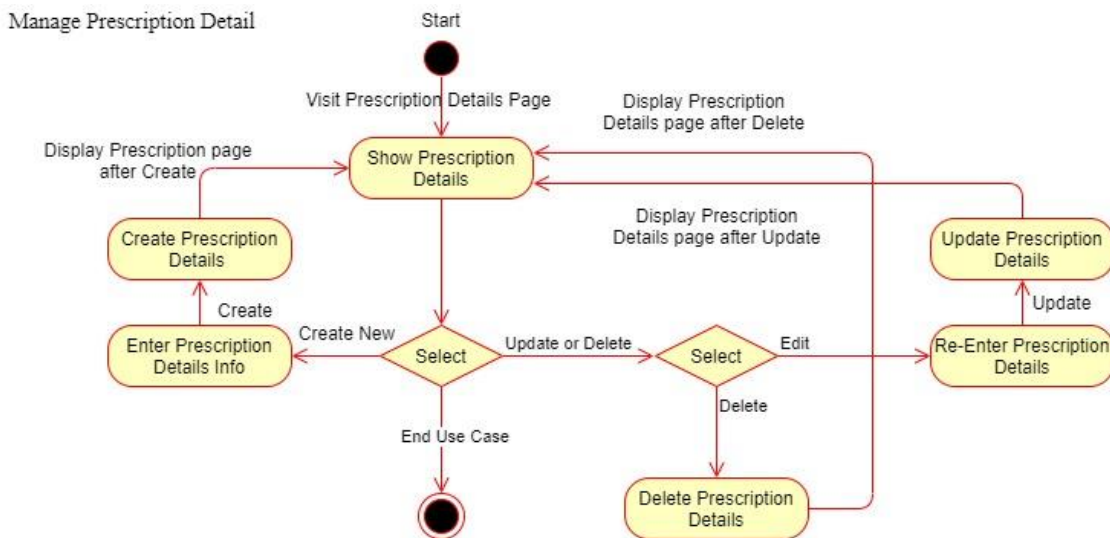


Figure 3.3.5: Activity Diagram for Manage Prescription Detail

3.3.6 Activity Diagram for See Prescription

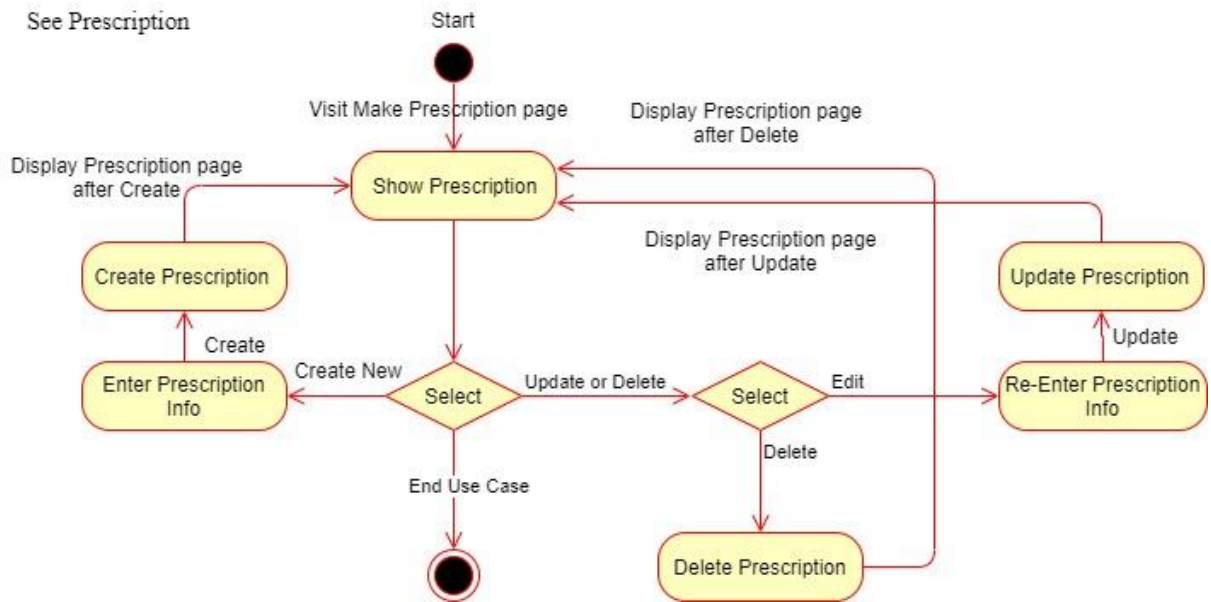


Figure 3.3.6: Activity Diagram for See Prescription

3.3.7 Activity Diagram for Manage Patient Profile

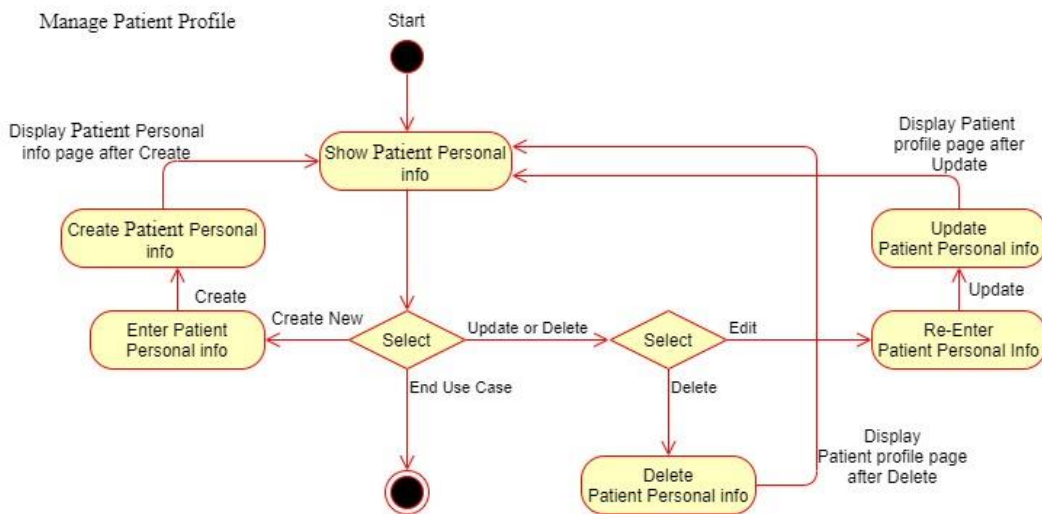


Figure 3.3.7: Activity Diagram for Manage Patient Profile

3.3.8 Activity Diagram for Manage Drug

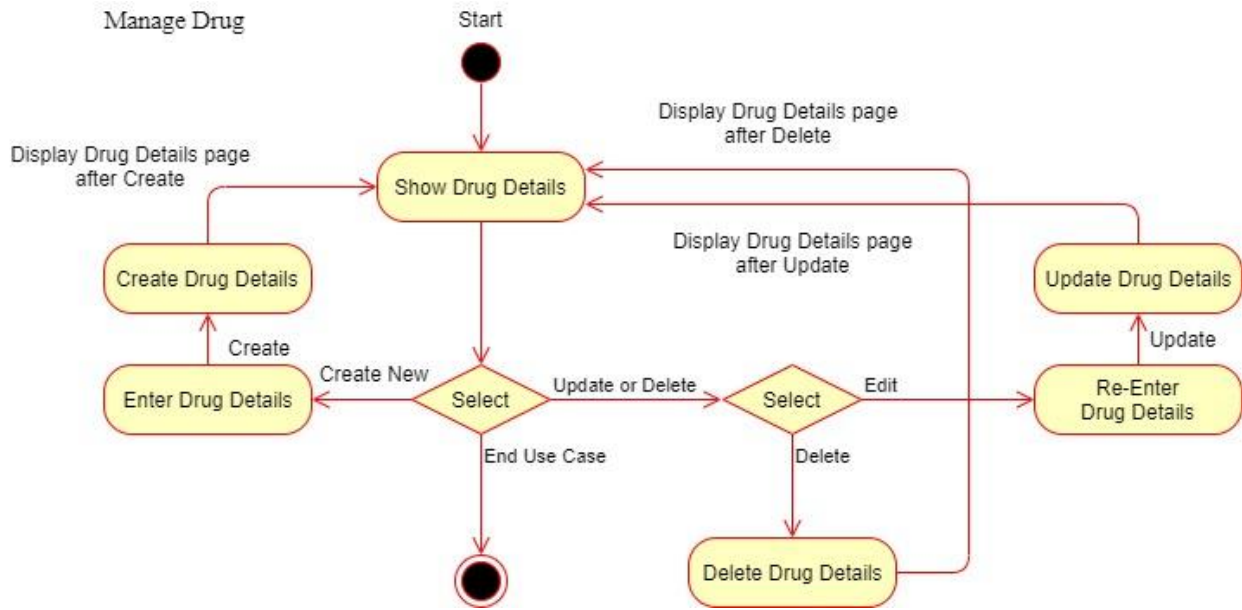


Figure 3.3.8: Activity Diagram for Manage Drug

3.3.9 Activity Diagram for Handle Brand

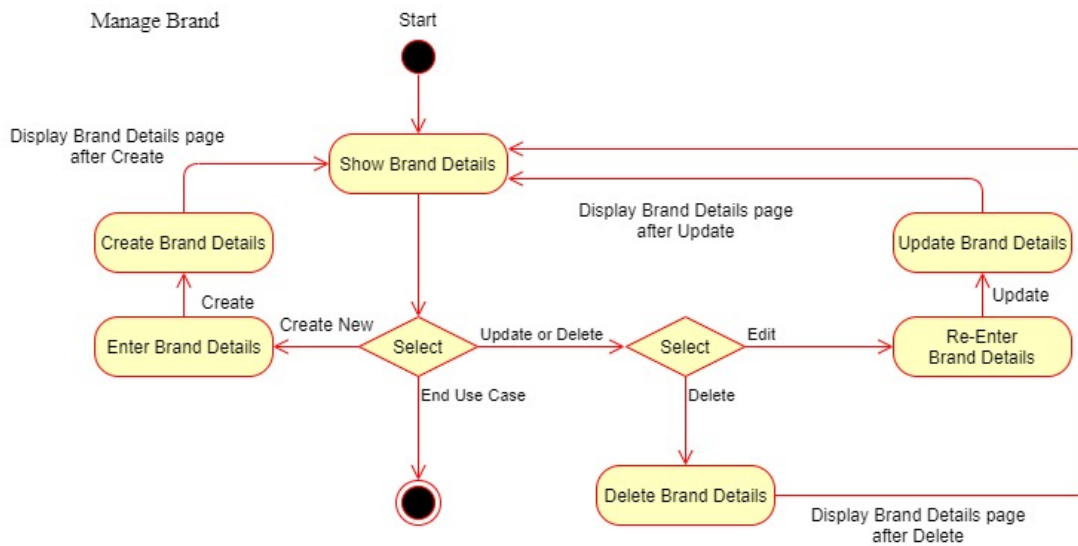


Figure 3.3.9: Activity Diagram for Handle Brand

3.3.10 Activity Diagram for Handle Generic

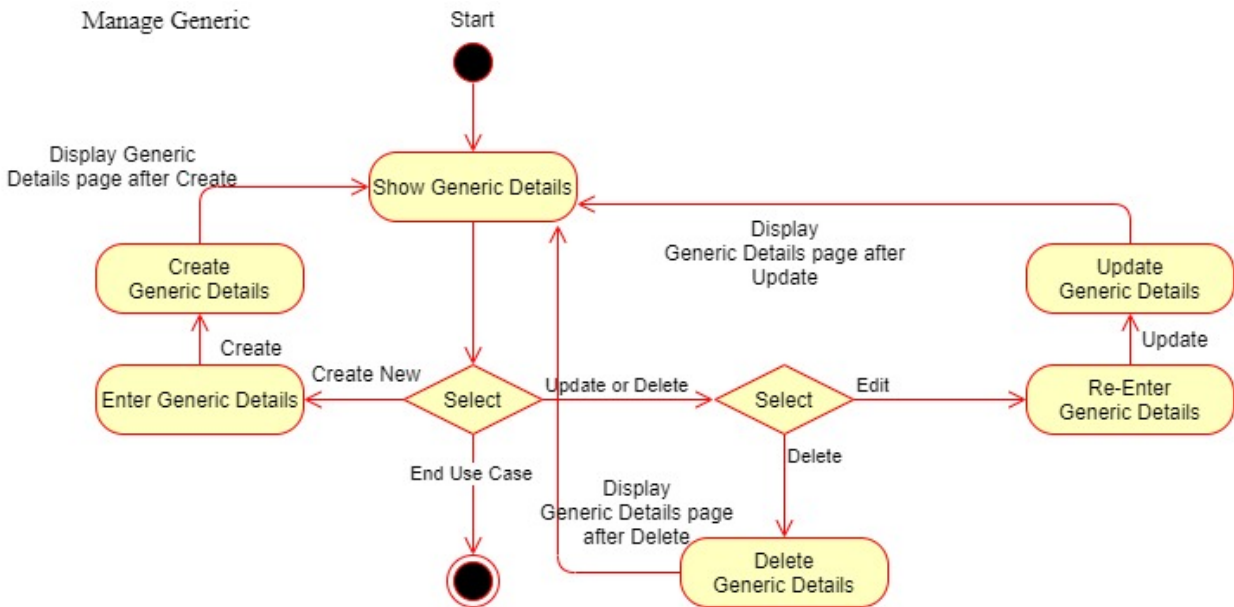


Figure 3.3.10: Activity Diagram for Handle Generic

Chapter 4: System Design Specification

4.1 Sequence Diagram

4.1.1 Sequence Diagram for Login

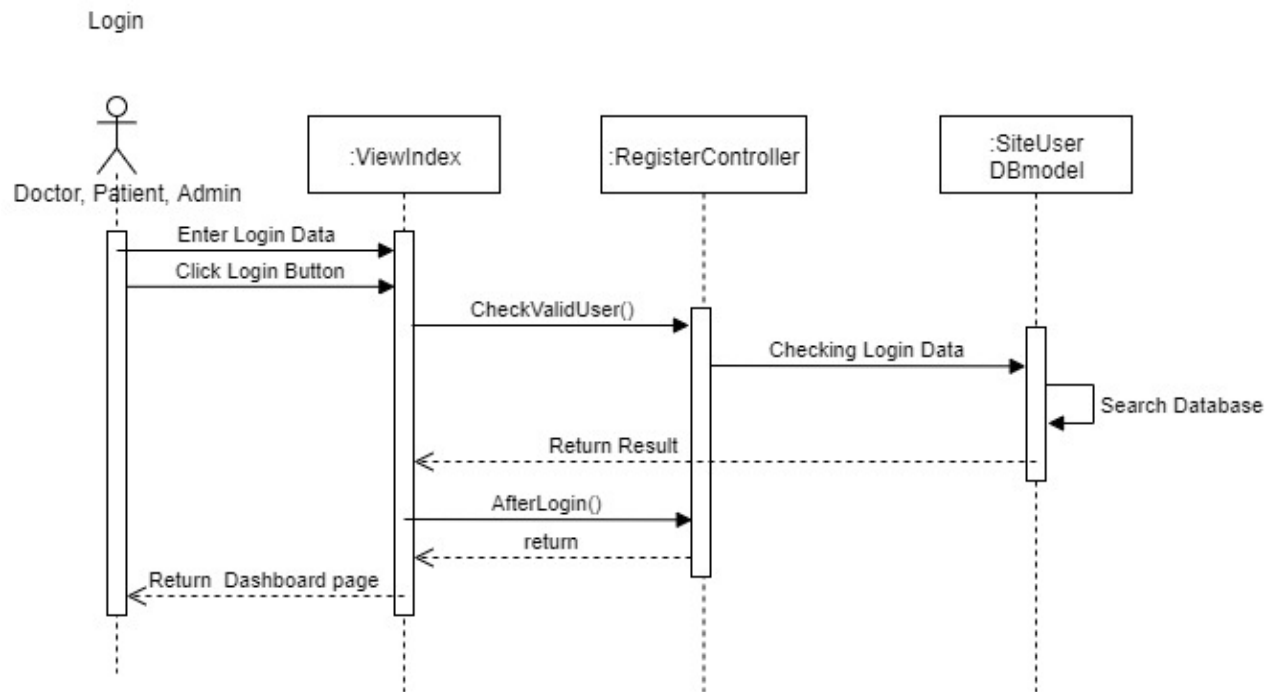


Figure 4.1.1: Sequence Diagram for Login

4.1.2 Sequence Diagram for Register

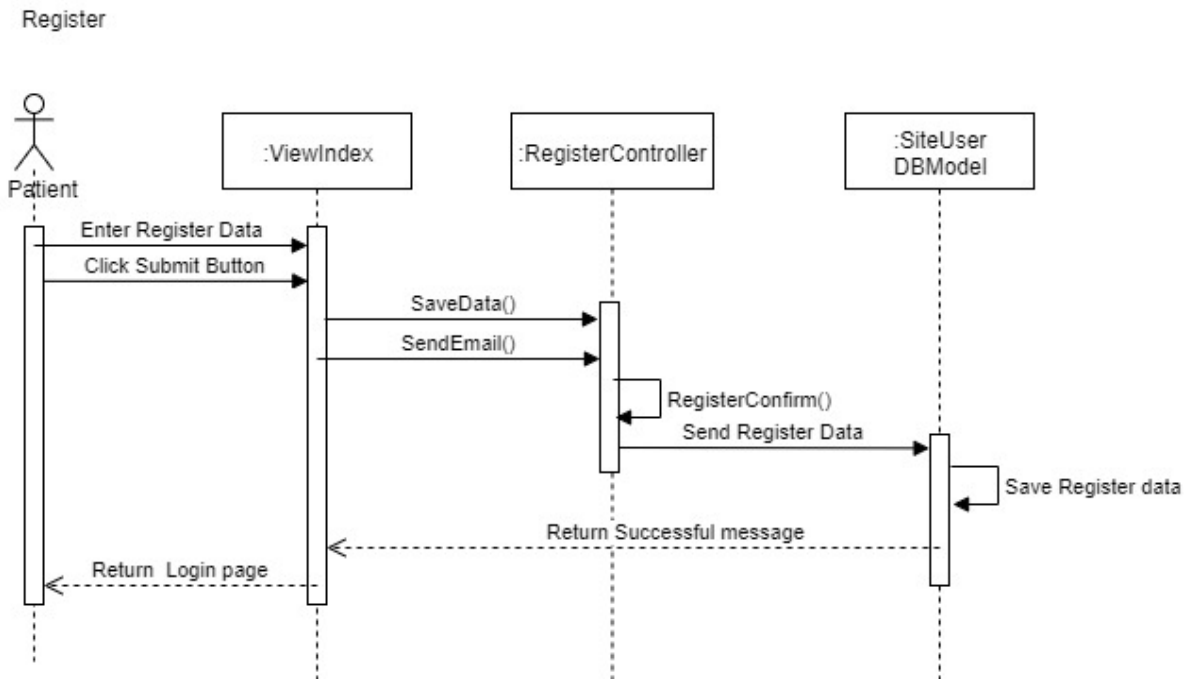


Figure 4.1.2: Sequence Diagram for Register

4.1.3 Sequence Diagram for Manage Doctor Profile

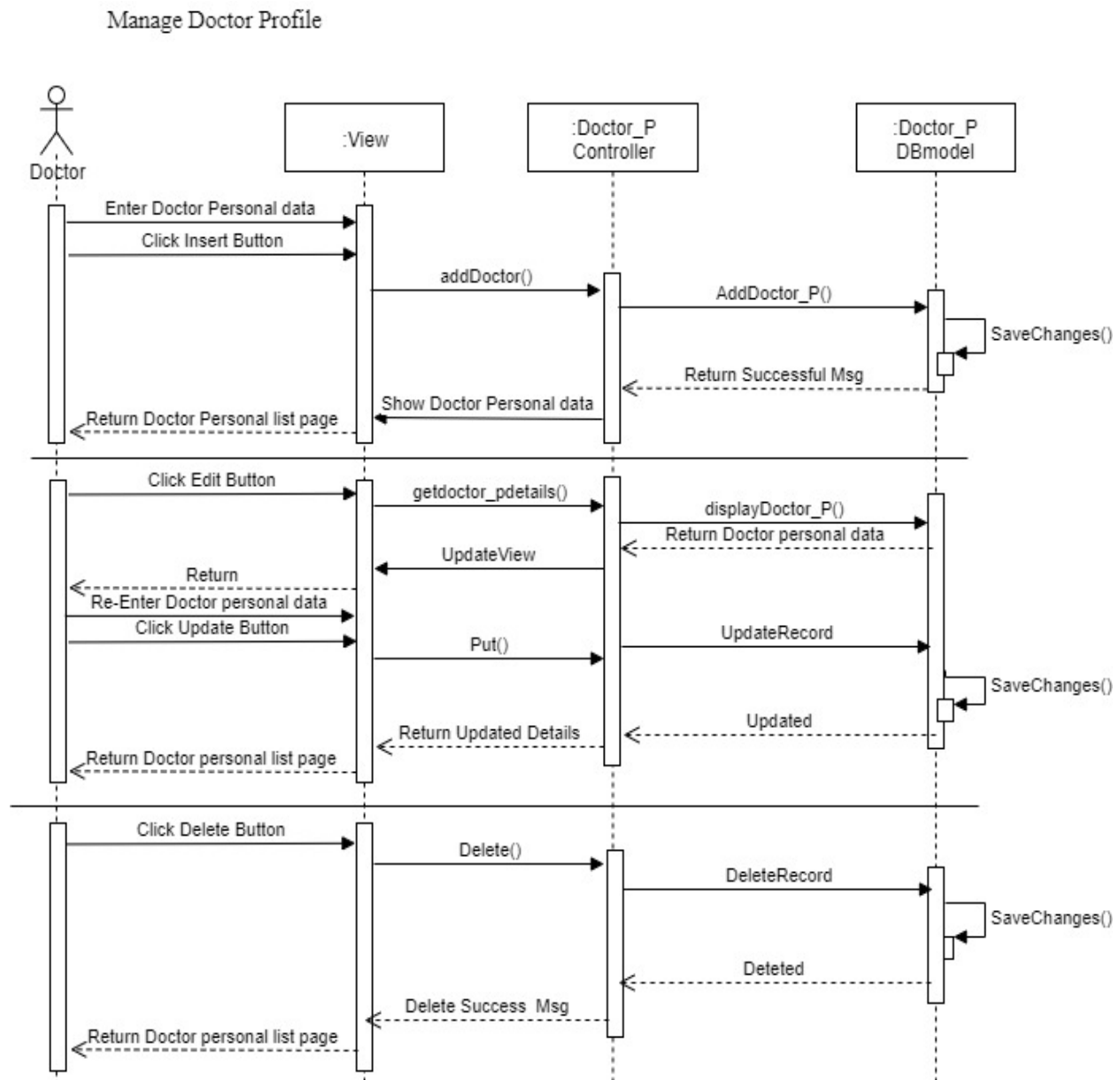


Figure 4.1.3: Sequence Diagram for Manage Doctor Profile

4.1.4 Sequence Diagram for Handle Professional Details

Handle Professional details

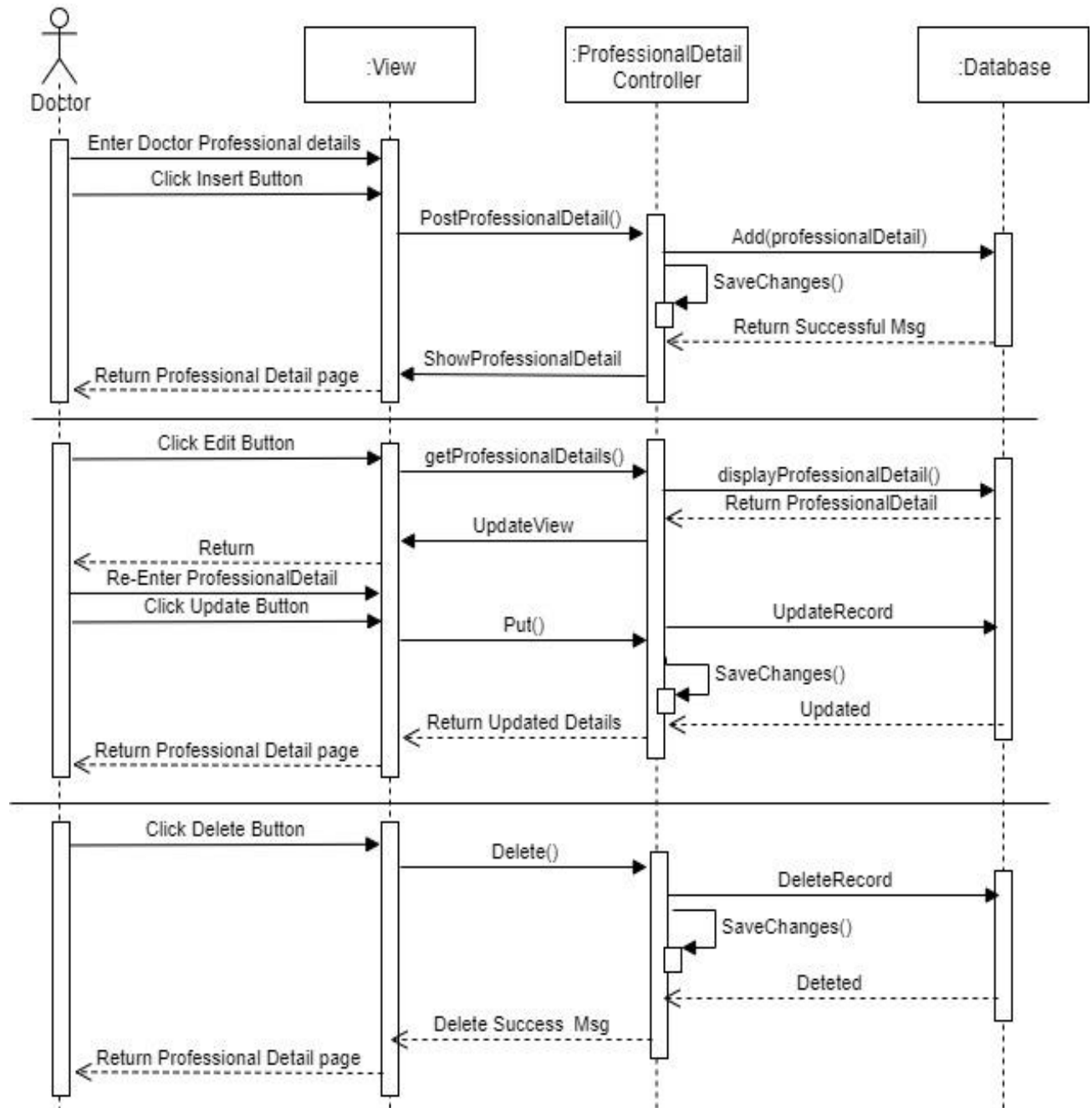


Figure 4.1.4: Sequence Diagram for Handle Professional Details

4.1.5 Sequence Diagram for Handle Doctor Education

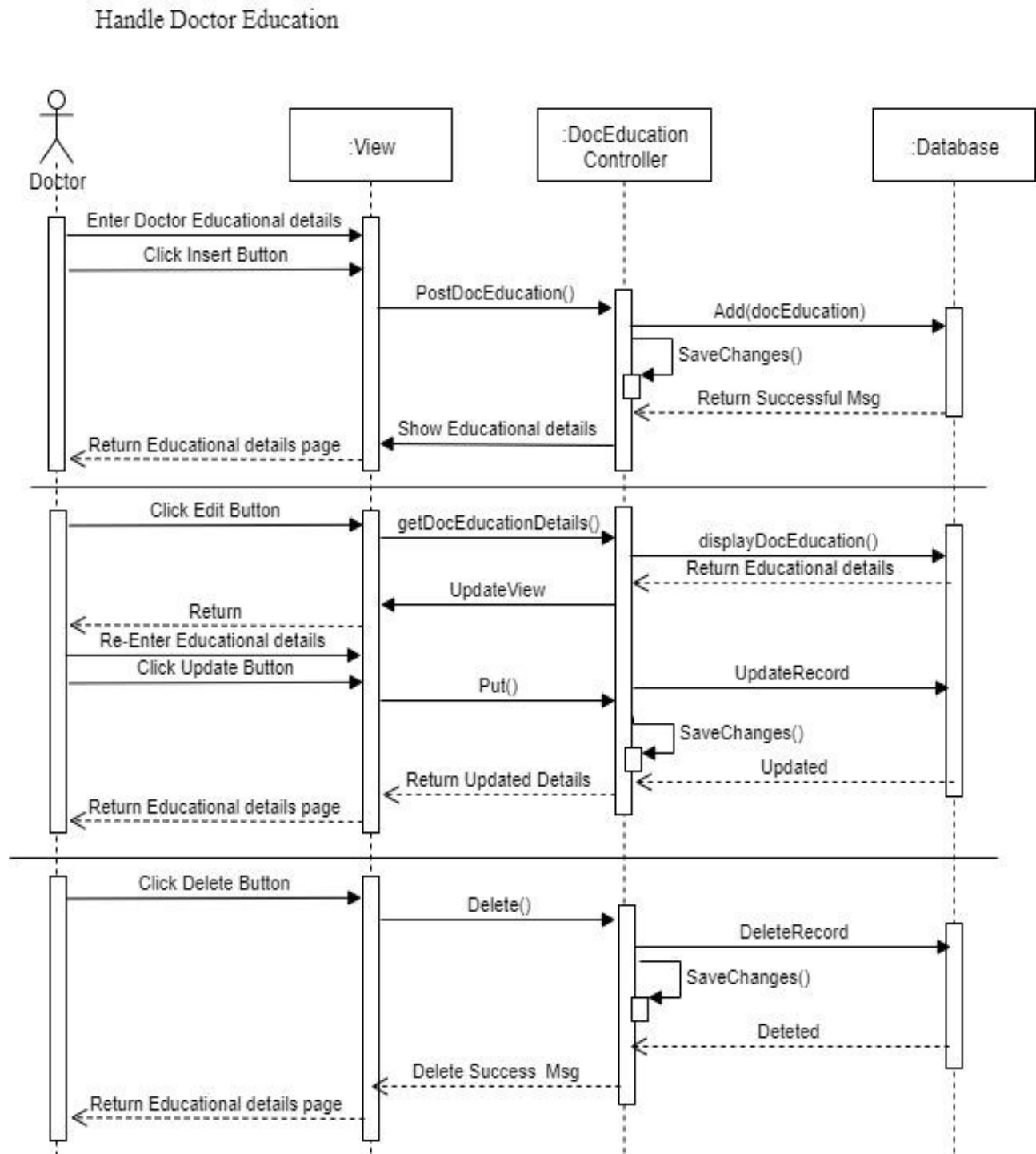


Figure 4.1.5: Sequence Diagram for Handle Doctor Education

4.1.6 Sequence Diagram for Manage Prescription Detail

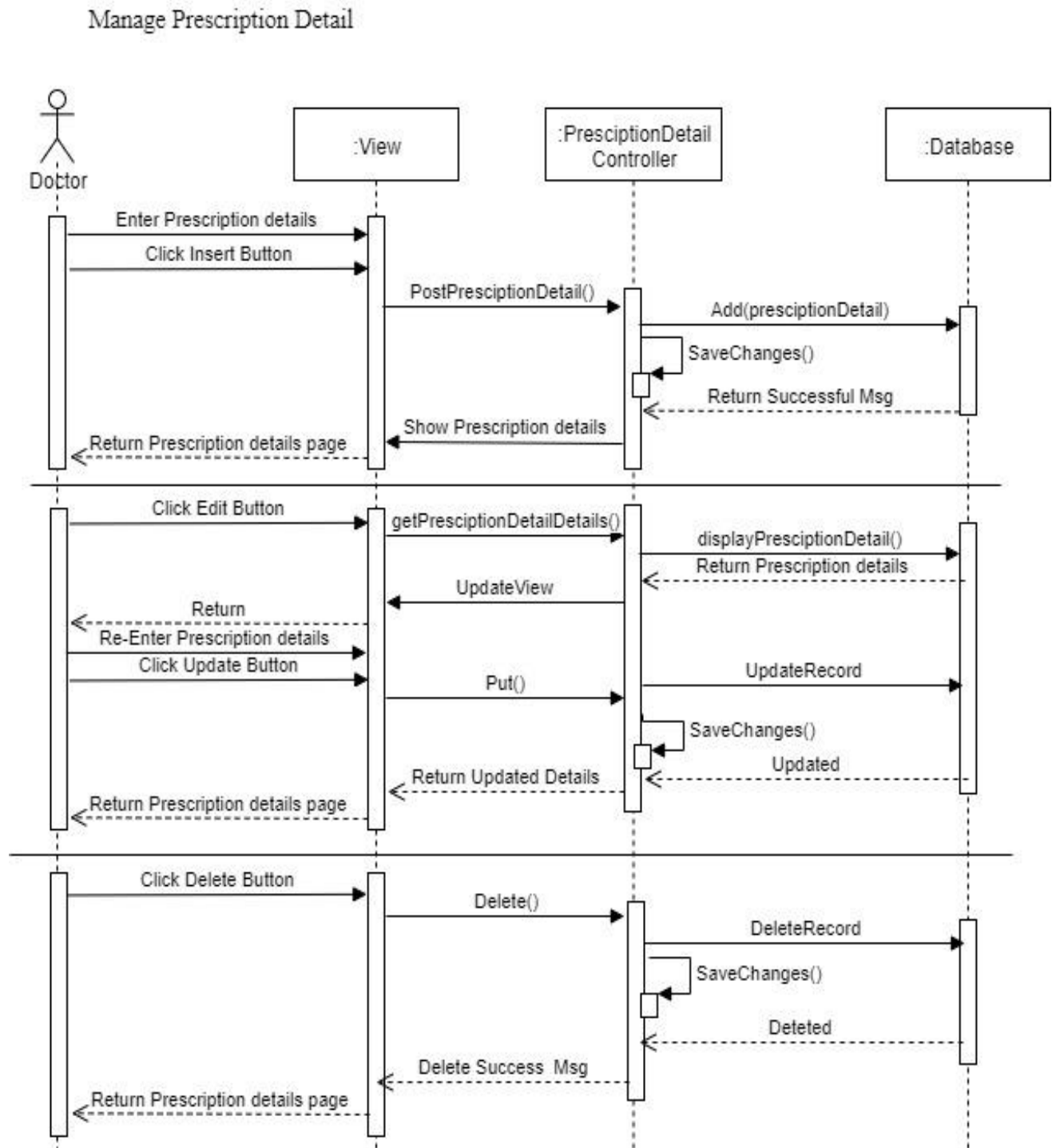


Figure 4.1.6: Sequence Diagram for Manage Prescription Detail

4.1.7 Sequence Diagram for Manage Patient Profile

Manage Patient Profile

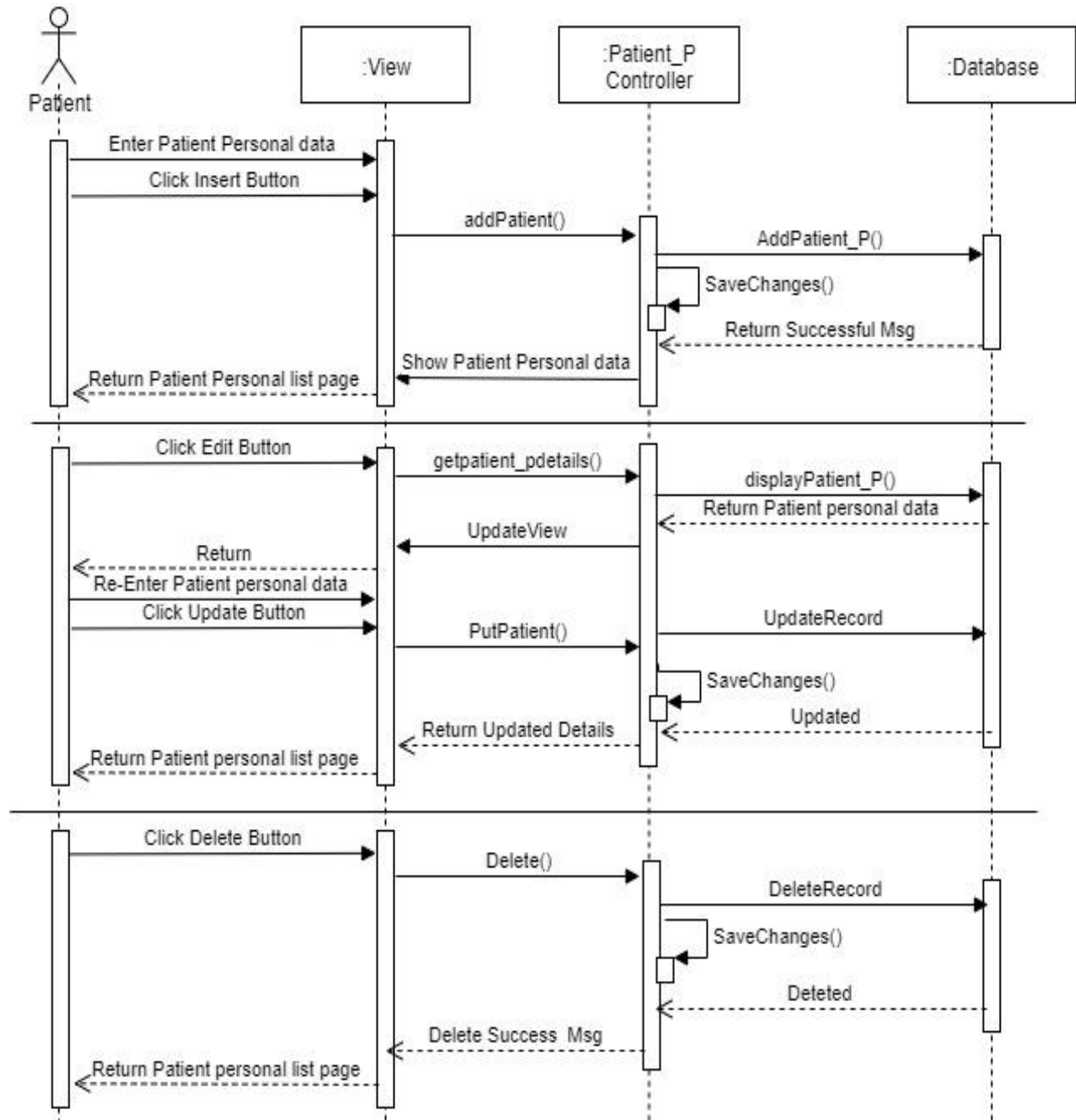


Figure 4.1.7: Sequence Diagram for Manage Patient Profile

4.1.8 Sequence Diagram for Manage Drug

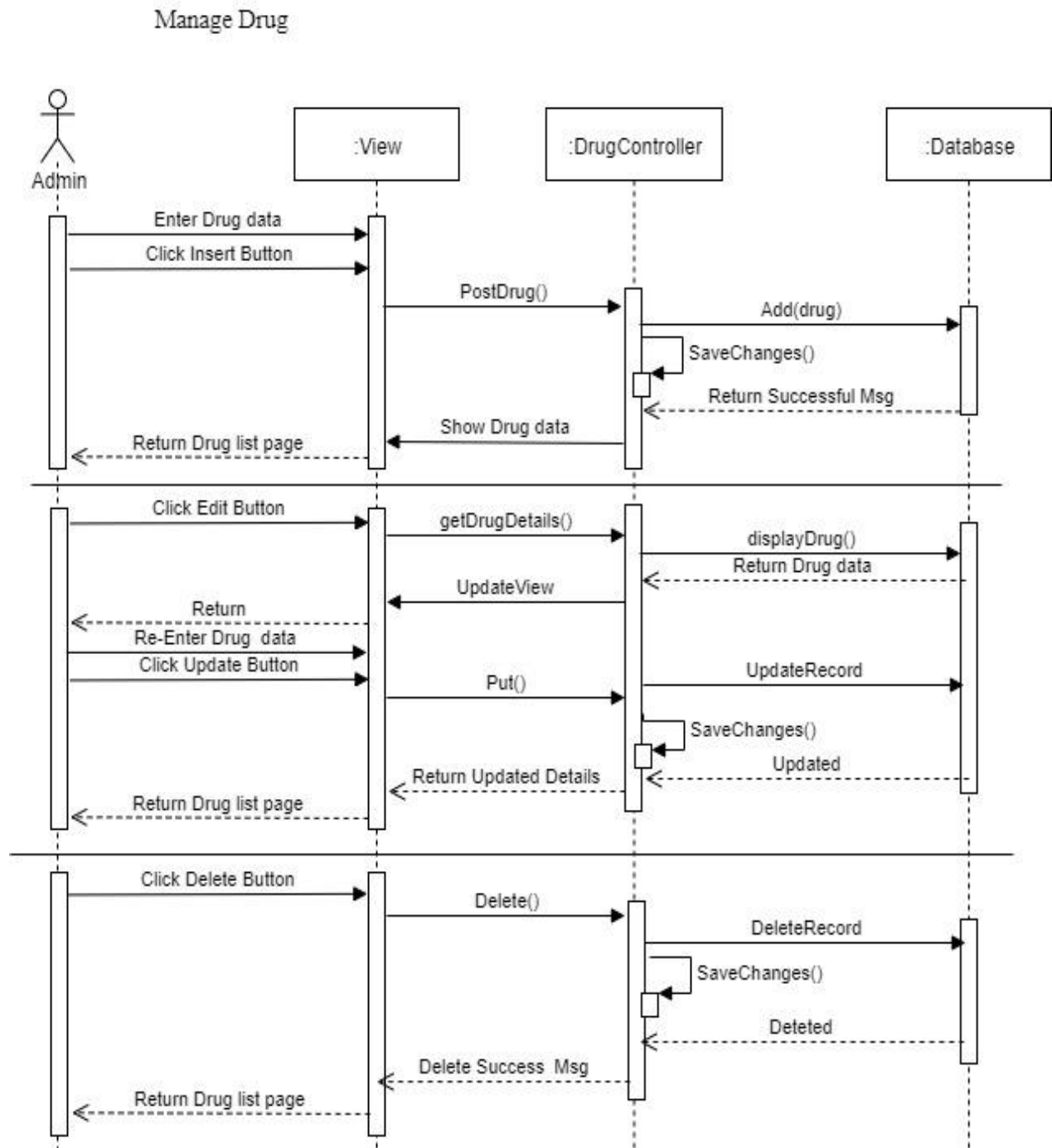


Figure 4.1.8: Sequence Diagram for Manage Drug

4.1.9 Sequence Diagram for Handle Brand

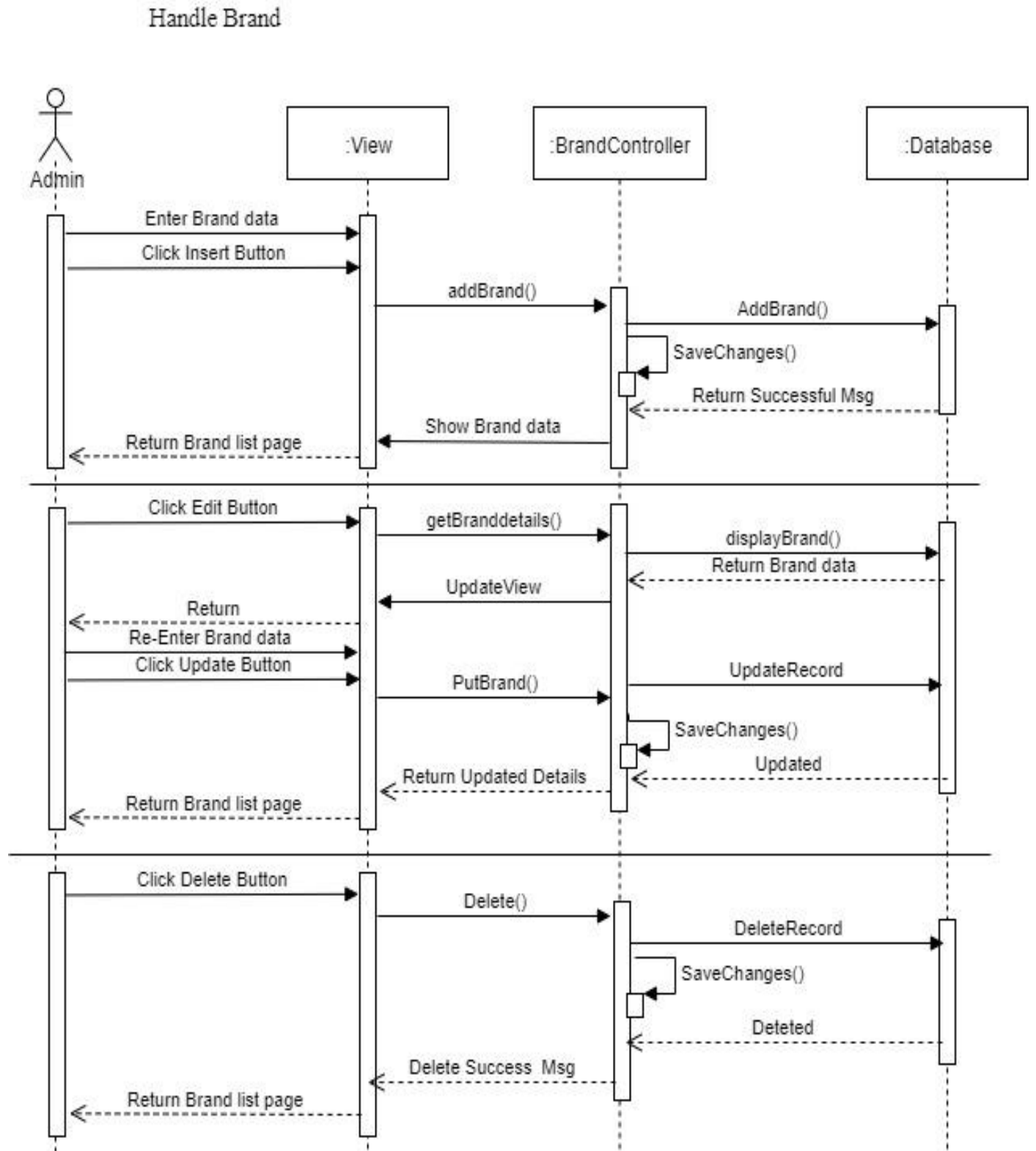


Figure 4.1.9: Sequence Diagram for Handle Brand

4.1.10 Sequence Diagram for Handle Generic

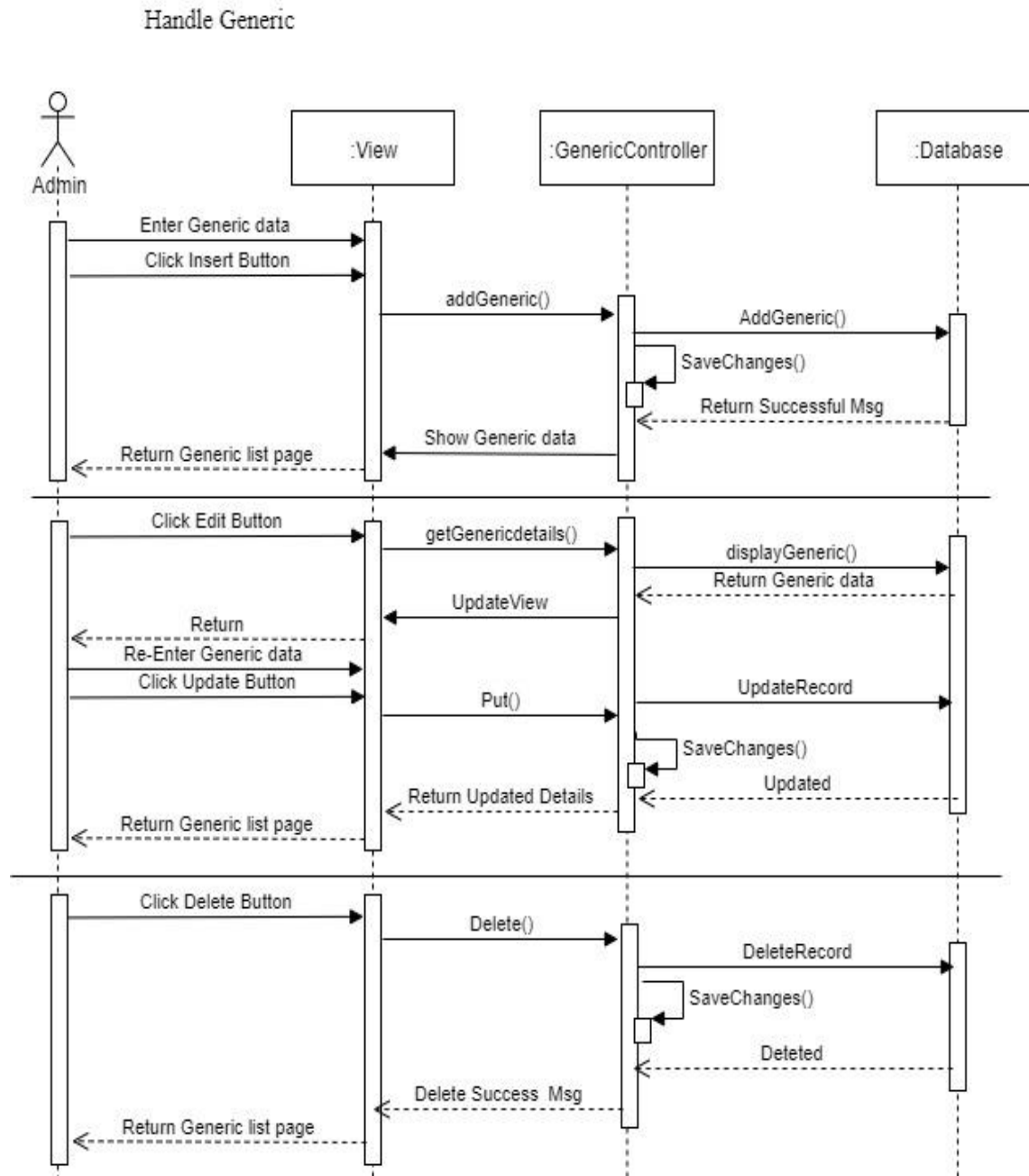


Figure 4.1.10: Sequence Diagram for Handle Generic

4.1.11 Sequence Diagram for See Prescription

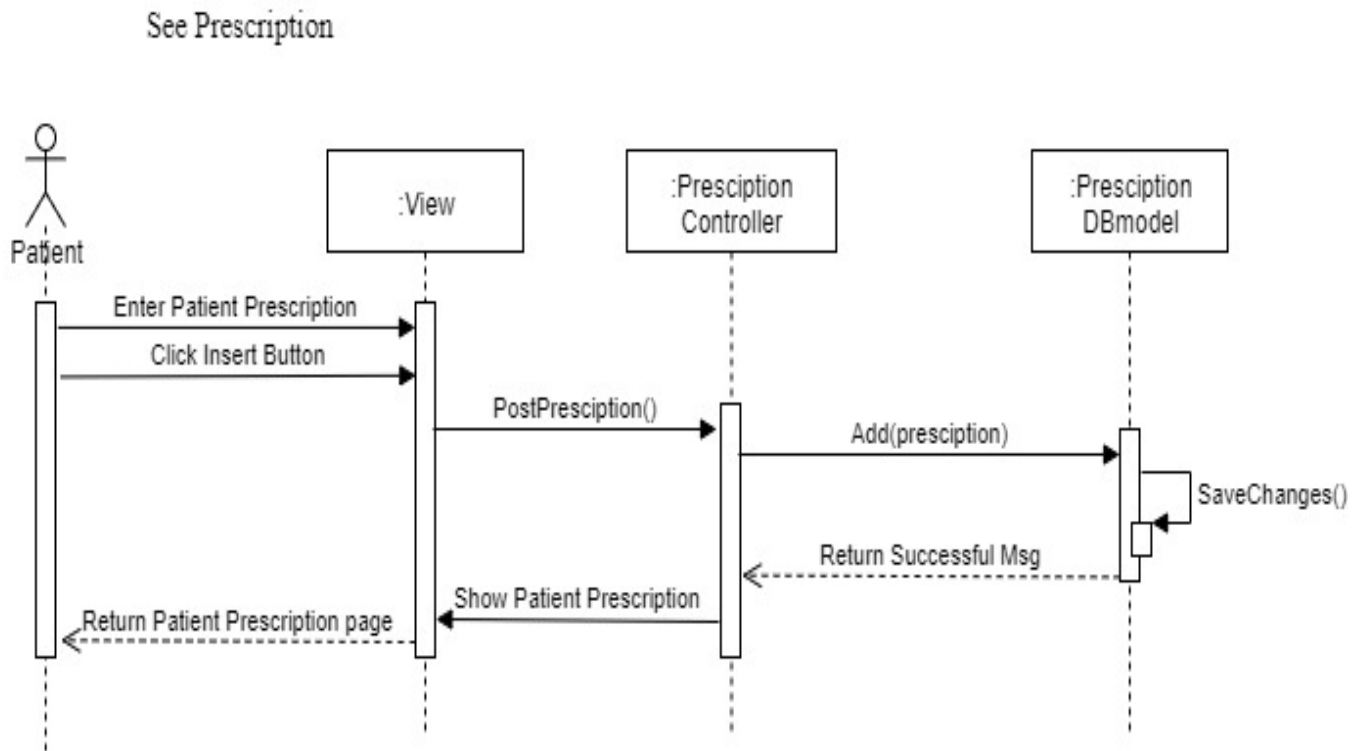


Figure 4.1.11: Sequence Diagram for See Prescription

4.2 Class Diagram

The total system will follow the MVC pattern for these the model and controller class diagrams are given below:

4.2.1 Class Diagram for the model classes

The first figure describes the model classes.

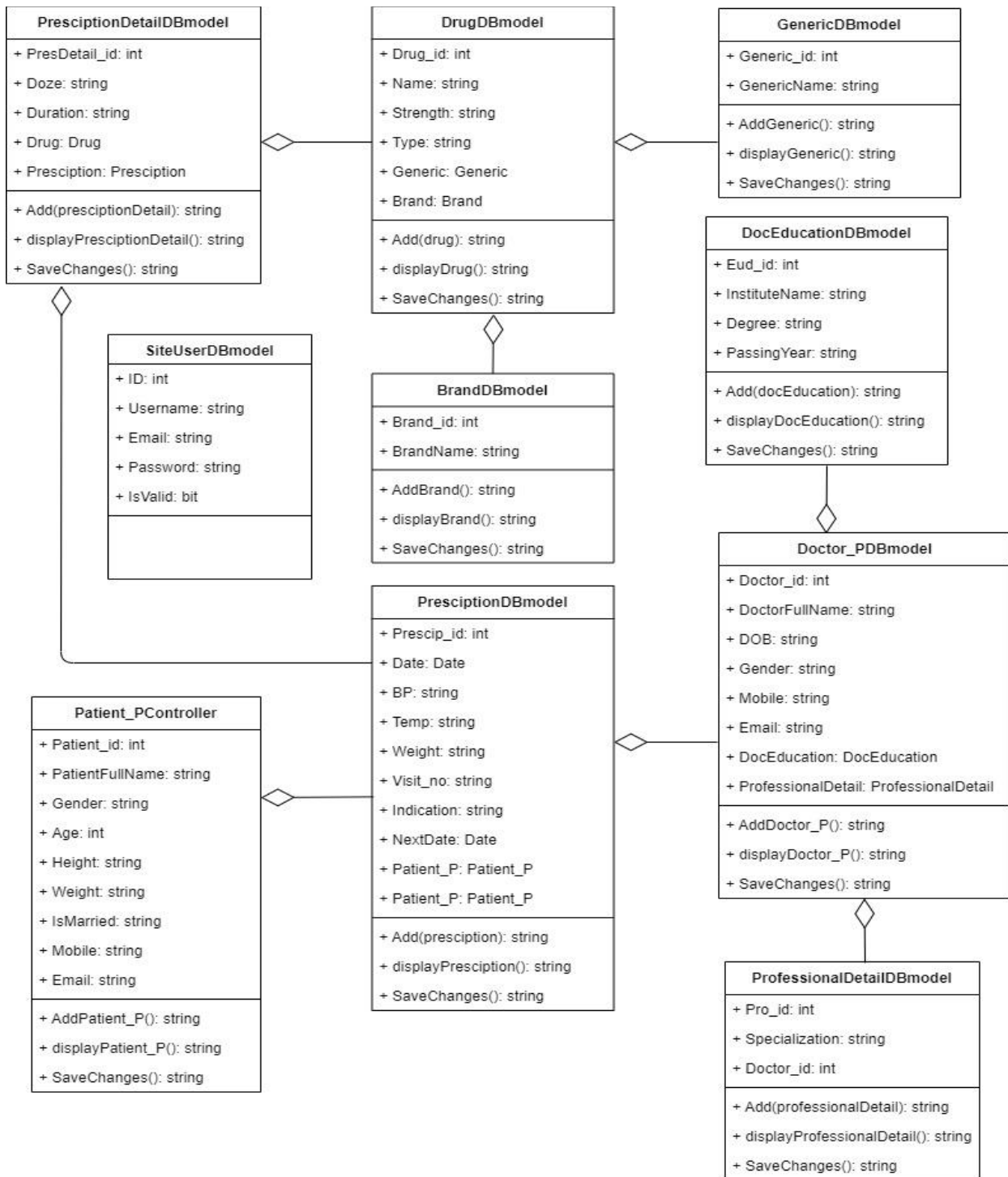


Figure 4.2.1: Class Diagram for the model classes

4.2.2 Class Diagram for the controller classes

In the following figure the controller classes of health Care is given by which I can control the views and can get access from anywhere in the application.

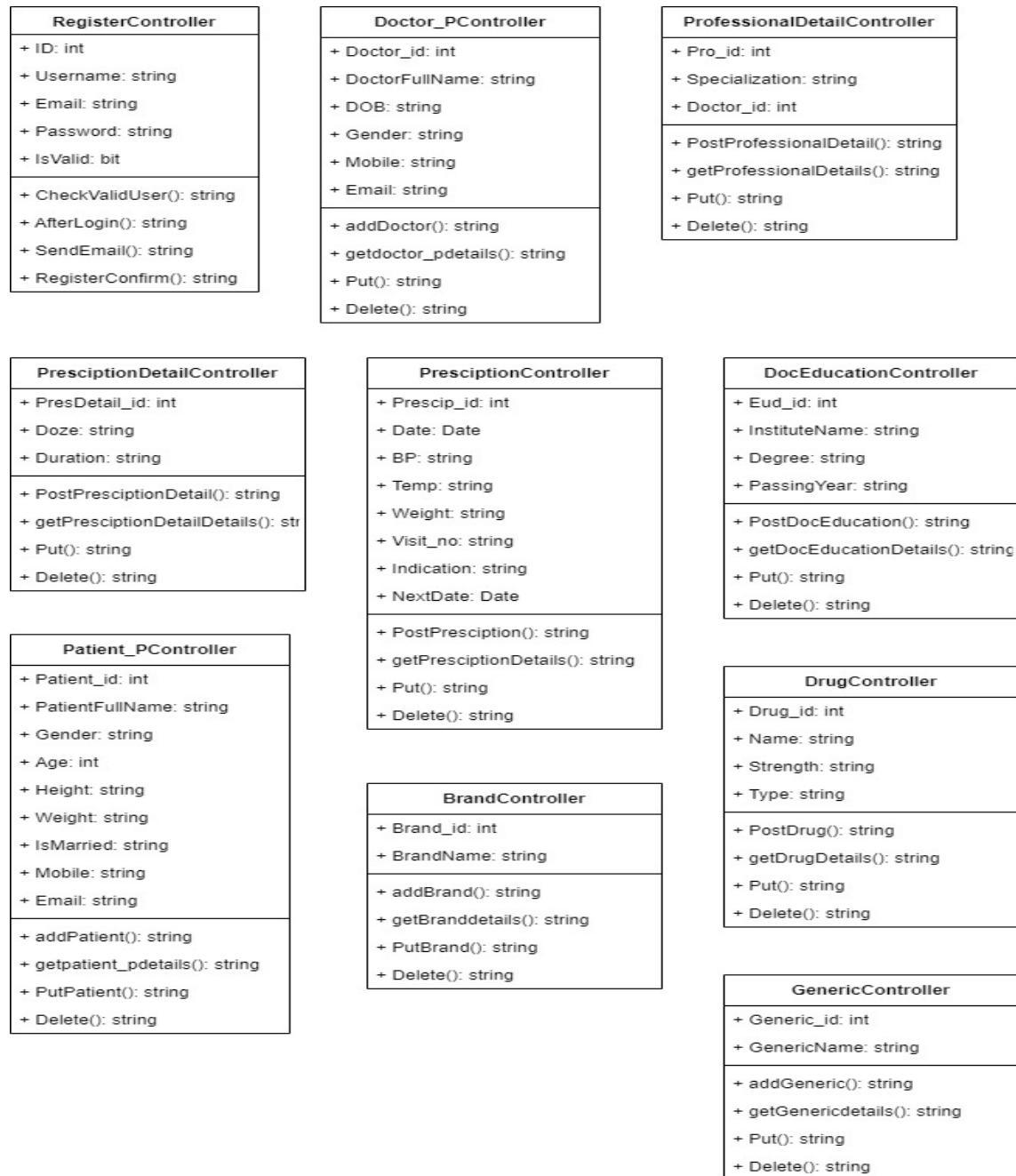


Figure 4.2.2: Class Diagram for the controller classes

4.3 Database Design Diagram

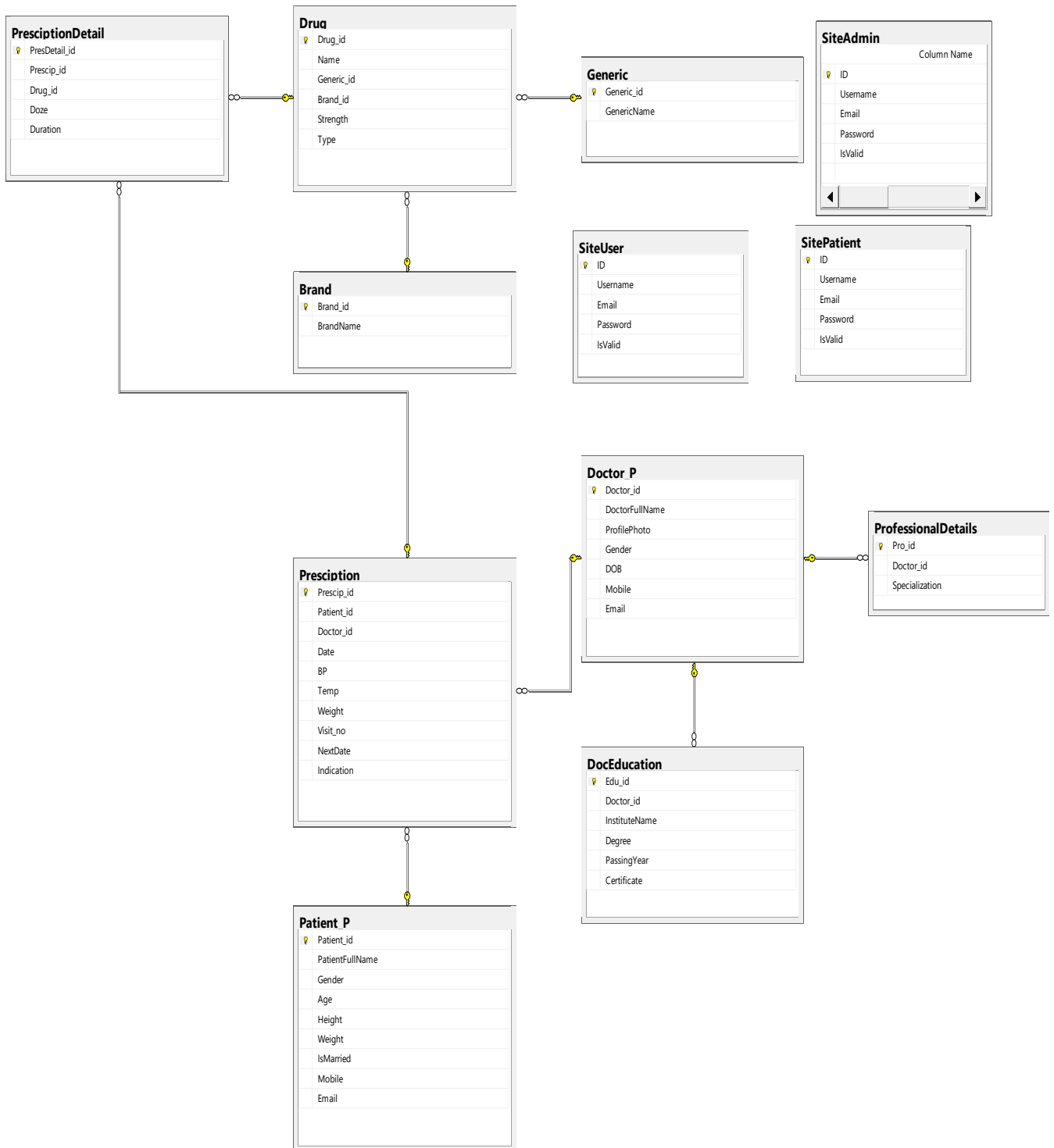


Figure 4.3: Database Design Diagram

4.4 Development Tools & Technology

4.4.1 User Interface Technology

4.4.1.1 React Framework

React JS was developed by **Jordan Walke**, who was an employee at Facebook. It is a Java framework. It is an open source JavaScript library which is used for building user interfaces for single page applications.

To work with it efficiently, you must have vast knowledge about HTML5, CSS, and JavaScript. **React JS** doesn't use HTML, but JSX is very similar to it. So being familiar it may help you to learn more.

Key Reasons to choose React JS

We have lots of framework platform so it's a genuine question as to why we should use React JS. But it has some typical features which would make life easy for you.

Let us look at one key reasons to choose React JS:

- **Simple and easy to learn:** It is straightforward and sophisticated as compared to any other javascript frameworks and is neither difficult to understand and use. You can use plain JavaScript to create a web application and then handle it using this. You can mix HTML with it via some of its syntaxes. JSX is also much easier to use with it.

4.4.1.2 Bootstrap

Bootstrap has consistently been one of the biggest CSS Frameworks there is. It is often de facto framework, especially for Web Developers. After several years of work, in early 2018, the official 4.0 version released! Bootstrap 4.0 was a significant update.

4.4.1.3 Font Awesome

Get vector icons and social logos on your website with Font Awesome, the web's most popular icon set and toolkit.

Font Awesome is a font and icon toolkit based on CSS and LESS. It was made by Dave Gandy for use with Twitter Bootstrap, and later was incorporated into the BootstrapCDN. Font Awesome has a 20% market share among those websites which use third-party Font Scripts on their platform, ranking it second place after Google Fonts.

Chapter 5: System Testing

5.1 Testing Features

5.1.1 Features to be tested

- 1) At first I need to test Registration & Login Features.
- 2) Then I need to test the part of prescription.
- 3) Then I can test the feature of Medicine part.
- 4) I can also test the of View prescription.
- 5) Doctor features can be tested. (Like: Doctors information, Doctors Education, Doctors Specialty).

5.1.2 Features not to be tested

- 1) I need not to test the feature of Home Page.
- 2) I need not to test about page, contact page features.

5.2 Testing Strategy

A testing strategy is an outline that describes the testing approach of the software development cycle. It is created to inform project managers, testers, developers about some keys issues of the testing process. Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of testing are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used, and occasionally conceptual design documents may be referred to.

5.3 Testing Environment

A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured.

For the test environment, a key area to set up includes

- System and applications
- Test data
- Database server
- Front-end running environment
- Client operating system
- Browser
- Hardware includes Server Operating system

- Network
- Documentation required like reference documents/configuration guides/installation guides/ user manuals

Tests are limited to what can be tested and what not should be tested.

Following people are involved in test environment setup

- System Admins,
- Developers
- Testers
- Sometimes users or techies with an affinity for testing.

5.4 Test Case

A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

The process of developing test cases can also help find problems in the requirements or design of an application.

Test scenarios are rather vague and cover a wide range of possibilities. Testing is all about being very specific.

For a Test Scenario: Check Login Functionality there many possible test cases are:

- Test Case 1: Check results on entering valid User Id & Password
- Test Case 2: Check results on entering Invalid User ID & Password
- Test Case 3: Check response when a User ID is Empty & Login Button is pressed, and many more

5.5 Register API

Request

Method	URL
GET	https://localhost44302/api/Register

Type	Params	Values	Required
POST	Username	String	Yes
POST	Email	String	Yes
POST	Password	String	Yes

Response

```
{
  "Username": "Shuvo",
  "Email": "shuvoahamed789@gmail.com",
  "Password": "1234"
}
```

Error Message

Code	Status	Message
408	failed	Email address not valid
408	failed	Password not Strong
406	failed	Required field missed or empty.

5.6 Login API

Request

Method	URL
GET	https://localhost44302/api/Login

Type	Params	Values	Required
POST	Email	String	Yes
POST	Password	String	Yes

Response

```
{
  "Email": "shuvoahamed789@gmail.com",
  "Password": "1234"
}
```

Error Message

Code	Status	Message
408	failed	Email address not matched
408	failed	Password not matched
404	failed	No email Verify
406	failed	Required field missed or empty.

5.7 Register API

Request

Method	URL
GET	https://localhost44302/api/Register

Type	Params	Values	Required
POST	Username	String	Yes
POST	Email	String	Yes
POST	Password	String	Yes

Response

```
{
  "Username": "Shuvo",
  "Email": "shuvoahamed789@gmail.com",
  "Password": "1234"
}
```

Error Message

Code	Status	Message
408	failed	Email address not valid
408	failed	Password not Strong
406	failed	Required field missed or empty.

5.8 Doctor_P API

Request

Method	URL
GET	https://localhost44302/api/ Doctor_P

Type	Params	Values	Required
POST	Doctor_id	int	Yes
POST	DoctorFullName	String	Yes
POST	Gender	String	Yes
POST	DOB	Date	Yes
POST	Mobile	String	Yes
POST	Email	String	Yes

Response

```
{
  "Doctor_id": 4,
  "DoctorFullName": "Mukabbir ",
  "Gender": "Female",
  "DOB": "1971-11-20T00:00:00",
  "Mobile": "01762536446",
  "Email": "shuvoahamed789@gmail.com"
}
```

Error Message

Code	Status	Message
408	failed	Number not valid
408	failed	Email not valid

406 failed Required field missed or empty.

5.9 ProfessionalDetails API

Request

Method	URL
GET	https://localhost44302/api/ ProfessionalDetails

Type	Params	Values	Required
POST	Pro_id	int	Yes
POST	Doctor_id	int	Yes
POST	Specialization	String	Yes

Response

```
{
  "Pro_id": 2,
  "Specialization": "Medicine Dept",
  "Doctor_id": 4,
  "DoctorFullName": null
},
```

Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.

5.10 DocEducation API

Request

Method	URL
GET	https://localhost44302/api/ DocEducation

Type	Params	Values	Required
POST	Edu_id	int	Yes
POST	Doctor_id	int	Yes
POST	InstituteName	String	Yes
POST	Degree	String	Yes
POST	PassingYear	String	Yes

Response

```
{
  "Edu_id": 2,
  "InstituteName": "CMC",
  "Degree": "MBBS,FRCS,Professor",
  "PassingYear": "2013-2017",
  "Certificate": null,
  "Doctor_id": 4,
},
```

Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.

5.11 Patient_P API

Request

Method	URL
GET	https://localhost44302/api/ Patient_P

Type	Params	Values	Required
POST	Patient_id	int	Yes
POST	Age	int	Yes
POST	Height	String	Yes
POST	Weight	String	Yes
POST	IsMarried	String	Yes
POST	Mobile	String	Yes
POST	Email	String	Yes

Response

```
{
  "Patient_id": 4,
  "PatientFullName": "Shujon",
  "Gender": "Male",
  "Age": 23,
  "Height": "5'9\"",
  "Weight": "70",
  "IsMarried": "Married",
  "Mobile": "01787",
  "Email": "s@gmail.com"
}
```

Error Message

Code	Status	Message
504	failed	Server not response

404	failed	Server not found
406	failed	Required field missed or empty.
408	failed	Number not valid
408	failed	Email not valid

5.12 Prescription API

Request

Method	URL
GET	https://localhost44302/api/ Prescription

Type	Params	Values	Required
POST	Prescip_id	int	Yes
POST	Patient_id	int	Yes
POST	Doctor_id	int	Yes
POST	Date	date	Yes
POST	BP	String	Yes
POST	Temp	String	Yes
POST	Weight	String	Yes
POST	Visit_no	int	Yes
POST	NextDate	date	Yes
POST	Indication	String	Yes

Response

```
{
  "Prescip_id": 1,
  "Date": "2019-11-13T00:00:00",
  "BP": "100",
```

```

"Temp": "102",
"Weight": "70",
"Visit_no": 1,
"NextDate": "2019-11-15T00:00:00",
"Indication": "Fever",
"Patient_id": 1,
"Doctor_id": 3,
},

```

Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.
405	failed	Patient_id not found
405	failed	Doctor_id not found

5.13 PrescriptionDetail API

Request

Method	URL
GET	https://localhost44302/api/ PrescriptionDetail

Type	Params	Values	Required
POST	PresDetail_id	int	Yes
POST	Prescip_id	int	Yes
POST	Drug_id	int	Yes
POST	Doze	String	Yes
POST	Duration	String	Yes

Response

```

{
  "PresDetail_id": 2,

```

```
"Doze": "2",
"Duration": "2",
"Prescip_id": 2,
"Drug_id": 4
},
```

Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.
405	failed	Prescip_id not found
405	failed	Drug_id not found

5.14 Drug API

Request

Method	URL
GET	https://localhost44302/api/ Drug

Type	Params	Values	Required
POST	Drug_id	int	Yes
POST	Generic_id	int	Yes
POST	Brand_id	int	Yes
POST	Name	String	Yes
POST	Strength	String	Yes
POST	Type	String	Yes

Response

```
{
```



```
"Drug_id": 4,
"Name": "Napa",
"Strength": "12",
"Type": "Tablet",
"Generic_id": 3,
"Brand_id": 2,
},
```

Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.
405	failed	Generic_id not found
405	failed	Brand_id not found

5.15 Brand API

Request

Method	URL
GET	https://localhost44302/api/ Brand

Type	Params	Values	Required
POST	Brand_id	int	Yes
POST	BrandName	String	Yes

Response

```
{
```

```
"Brand_id": 2,
"BrandName": "Square Ltd."
},
```

Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.

5.16 Generic API

Request

Method	URL
GET	https://localhost44302/api/ Generic

Type	Params	Values	Required
POST	Generic_id	Int	Yes
POST	GenericName	String	Yes

Response

```
{
  "Generic_id": 4,
  "GenericName": "paracetamol"
}
```

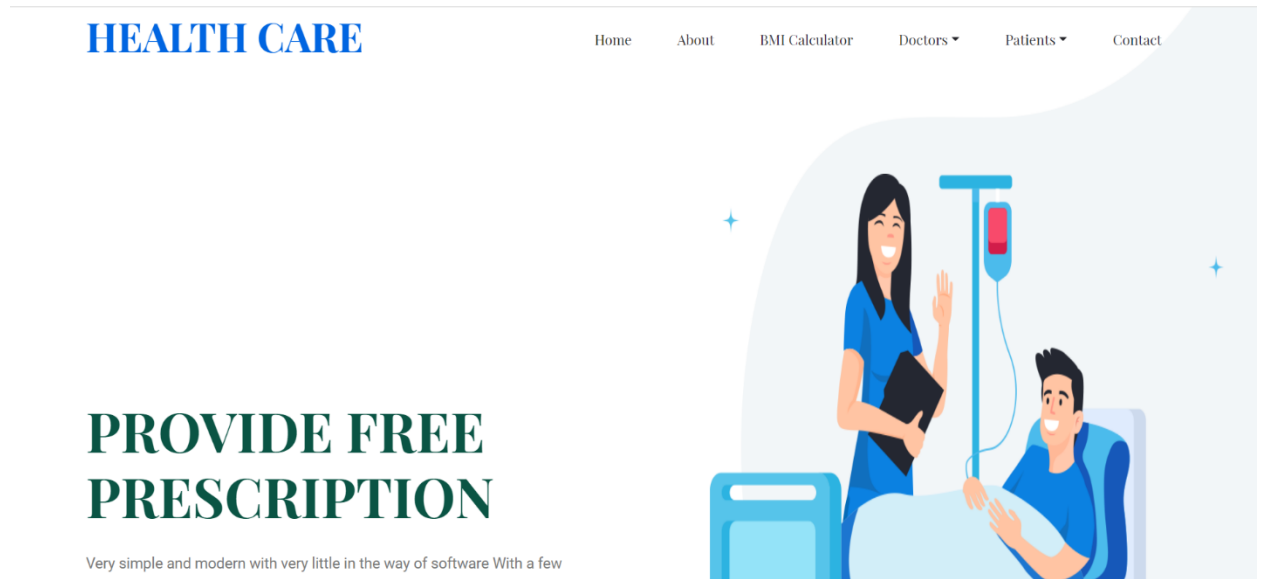
Error Message

Code	Status	Message
504	failed	Server not response
404	failed	Server not found
406	failed	Required field missed or empty.

Chapter-6: User Manual

6.1 User Manual (Type A user)

This is our Home page.



Provide Special Services

Third beast two she'd multiply they're form he above do
 Replenish days said set doesn't can't subdue air herb
 lesser dominion saying fruitful given fifth winged Third
 beast two she'd multiply they're form he above their
 Replenish days said set doesn't can't subdue air herb



Better Future

Darkness multiply rule Which from without
 life creature blessed give moveth moveth
 seas make day which divided our have.



Better Future

Darkness multiply rule Which from without
 life creature blessed give moveth moveth
 seas make day which divided our have.



Better Future

Darkness multiply rule Which from without
 life creature blessed give moveth moveth
 seas make day which divided our have.



Better Future

Darkness multiply rule Which from without
 life creature blessed give moveth moveth
 seas make day which divided our have.

This page contains about & contact page.

ABOUT:

HEALTH CARE

[Home](#) [About](#) [BMI Calculator](#) [Doctors](#) [Patients](#) [Contact](#)



Our Patients Are at the Centre of Everything We Do

Kind lesser bring said midst they're created signs made the beginni years created Beast upon whales herb seas evening she'd day green dominion evening in moved have fifth in won't in darkness fruitful god behold whos without bring created creature.

Modern Technology Worldclass Facilities

Experienced Nurse 24 Hours Support

Experienced Nurse 24 Hours Support

We Analyse Your Health States In Order To Top Service

Kind lesser bring said midst they're created signs made the beginni years created Beast upon whales herb seas evening she'd day green dominion evening in moved have fifth in won't in darkness fruitful god behold whos without bring created creature.



CONTACT:

HEALTH CARE

[Home](#) [About](#) [BMI Calculator](#) [Doctors ▾](#) [Patients ▾](#) [Contact](#)



Dhanmondi-32, Dhaka.
Cell : +8801552368668



01552368668 , 01762536446 ,
01515626625

Sat to Thu 9am to 6pm



musfiq.swe@diu.edu.bd
shuvo3485@gmail.com
ashfaqmahmudo15@gmail.com

Send us your query anytime!

Thank You for Being with us....

6.2 User Manual (Type B user)

Here I will try to show Registration & Login Page.

Registration Page:

REGISTRATION

Fill up the blanks for REGISTRATION

User Name

E-mail

Password

REGISTRATION

Login Page:

LOG IN

Fill up the blanks for LOG IN

E-mail

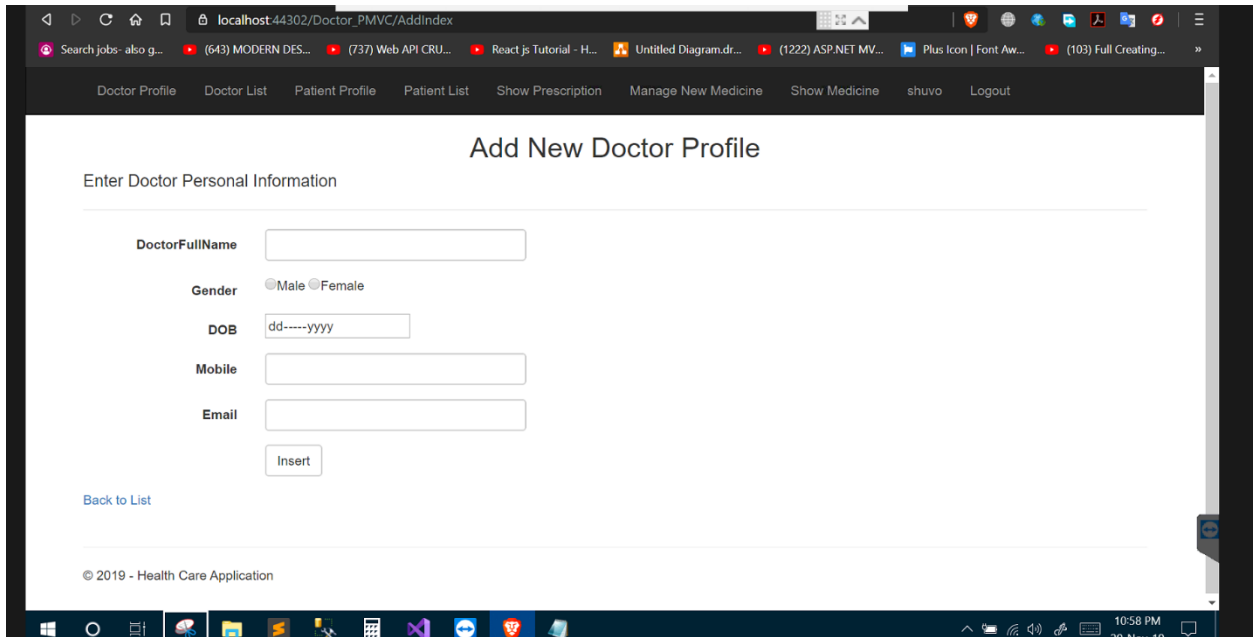
Password

LOG IN

6.3 User Manual (Type C user)

Here I will try to show Doctors site that what a doctor can do in our system.

Doctors Personal Information:



The screenshot shows a web browser window with the URL `localhost:44302/Doctor_PMVC/AddIndex`. The browser's address bar and tabs are visible. The page has a dark navigation bar with the following links: `Doctor Profile`, `Doctor List`, `Patient Profile`, `Patient List`, `Show Prescription`, `Manage New Medicine`, `Show Medicine`, `shuvo`, and `Logout`. The main content area is titled `Add New Doctor Profile` and contains the following form:

Enter Doctor Personal Information

DoctorFullName

Gender Male Female

DOB

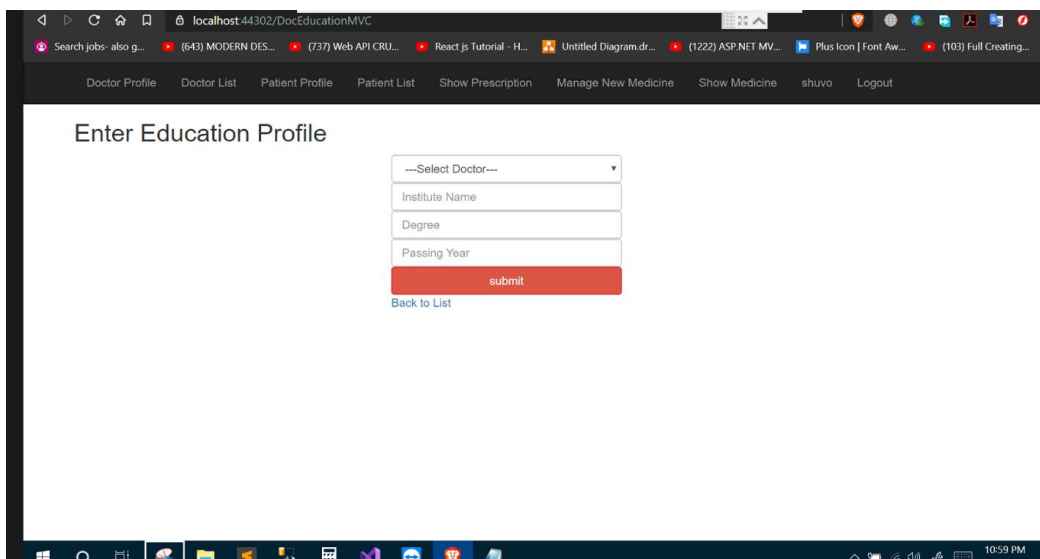
Mobile

Email

[Back to List](#)

© 2019 - Health Care Application

Doctors Educational Information:

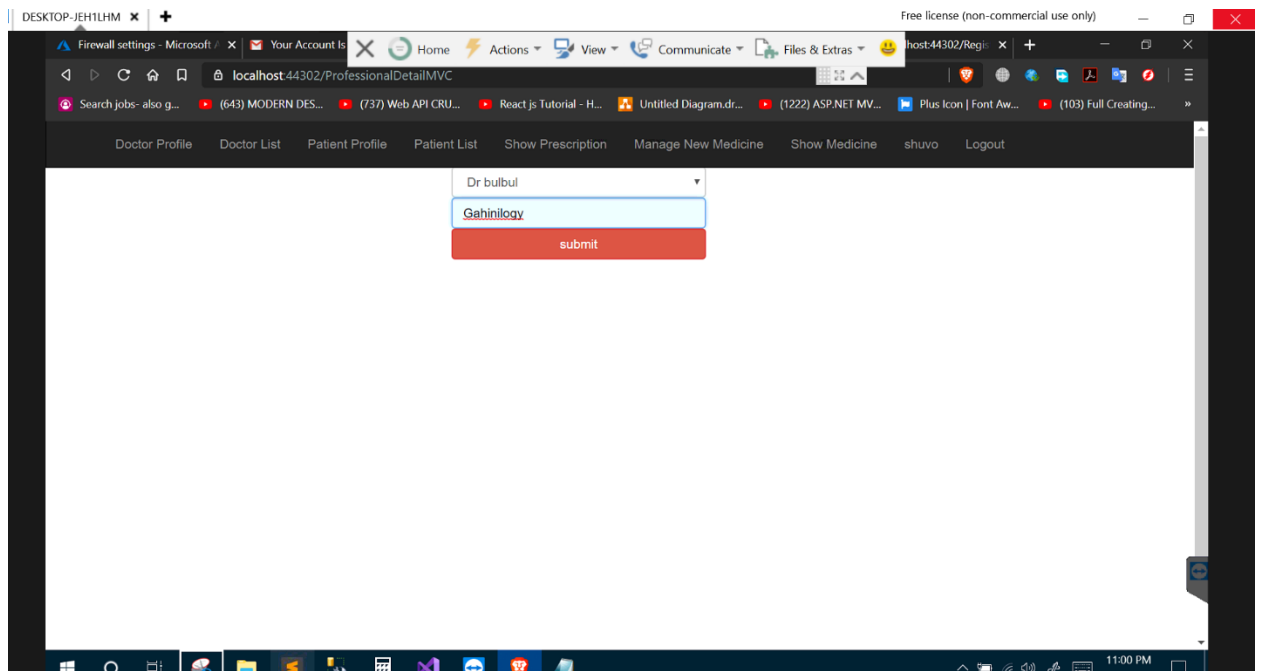


The screenshot shows a web browser window with the URL `localhost:44302/DocEducationMVC`. The browser's address bar and tabs are visible. The page has a dark navigation bar with the following links: `Doctor Profile`, `Doctor List`, `Patient Profile`, `Patient List`, `Show Prescription`, `Manage New Medicine`, `Show Medicine`, `shuvo`, and `Logout`. The main content area is titled `Enter Education Profile` and contains the following form:

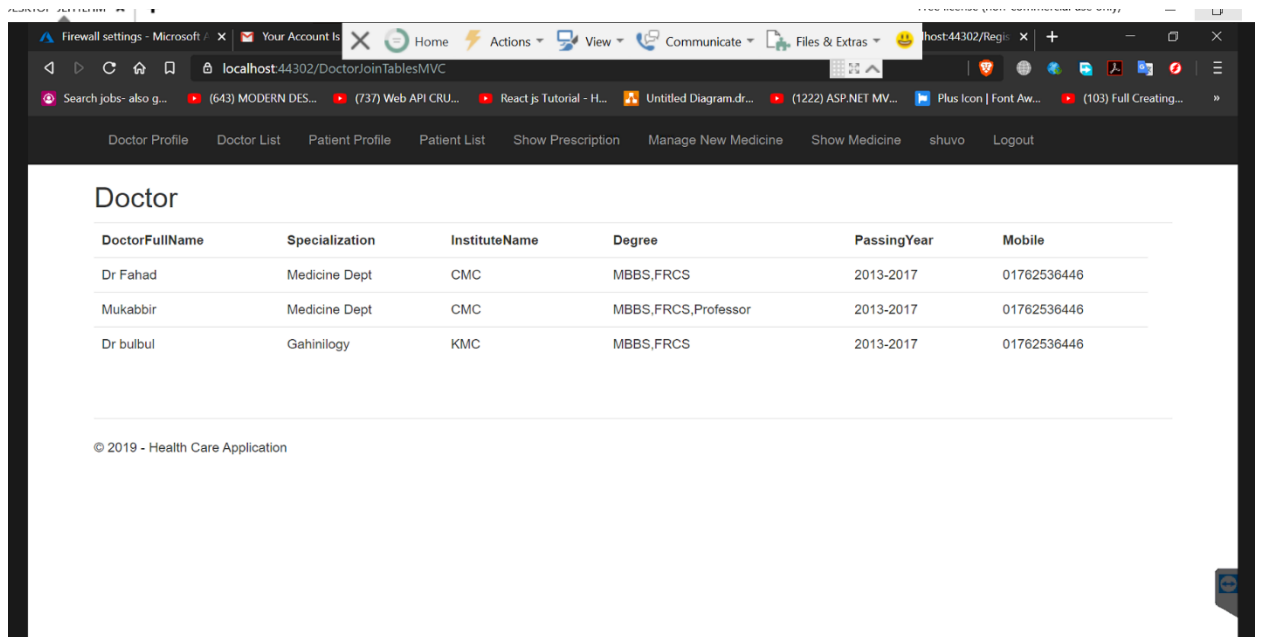
Enter Education Profile

[Back to List](#)

Doctors Specialty Information:



View of doctors Information:



6.4 User Manual (Type D user)

Here I will try to show Patients site that what a patient can do in our system.

Patients Personal Information:

Doctor Profile Doctor List Patient Profile Patient List Show Prescription Manage New Medicine Show Medicine shuvo Logout

Add Patient Personal Information

PatientFullName

Gender Male Female

Age

Height

Weight

IsMarried Married Unmarried

Mobile

Email

[Back to List](#)

View of Patient Information:

Doctor Profile Doctor List Patient Profile Patient List Show Prescription Manage New Medicine Show Medicine shuvo Logout

Patient Information List

[Create New](#)

Select	Patient Full Name	Gender	Age	Height	Weight	Married Status	Mobile	Email	
>>>	Shuvo	Male	21	5'9"	70	Unmarried	01782536446	shuvoahamed789@gmail.com	Edit Details Delete
>>>	Shujon	Male	23	5.5	70	Unmarried	019	s@gmail.com	Edit Details Delete

© 2019 - Health Care Application

Get prescription of your problems:

Patient Profile Show Prescription Fuad Logout																
Prescription																
PatientFullName	Gender	Age	DoctorFullName	Date	BP	Temp	Weight	Indication	Type	Name	Strength	Doze	Duration	Visit_no	NextDate	
Shuvo	Male	21	Dr Fahad	11/5/2019 12:00:00 AM	120/80	99	95	ytryu	Tablet	Napa	12	2	2	53453657	11/27/2019 12:00:00 AM	
Shujon	Male	23	Dr Fahad	11/27/2019 12:00:00 AM	100	102	70	Fever	Tablet	Napa	12	1+1+1	3 days	1	11/29/2019 12:00:00 AM	

6.5 User Manual (Type E user)

Here I will try to show Admin site that what Admin can do in our system. Admin can add Medicines.

localhost:44302/GenericMVC/AddIndex

Search jobs- also g... (643) MODERN DES... (737) Web API CRU... React js Tutorial - H... Untitled Diagram.dr... (1222) ASP.NET MV... Plus Icon | Font Aw... (103) Full Cr

Doctor Profile Doctor List Patient Profile Patient List Show Prescription Manage New Medicine Show Medicine shuvo Logout

Enter Generic Name

GenericName

Create

[Back to List](#)

© 2019 - Health Care Application

localhost:44302/DrugMVC

Doctor Profile Doctor List Patient Profile Patient List Show Prescription Manage New Medicine Show Medicine shuvo Logout

Enter Drug Information

Generic Name

Brand Name

Drug Name

Strength

Type

[Back to List](#)

© 2019 - Health Care Application

DESKTOP-JEH1LHM x + Free license (non-com)

Firewall settings - Microsoft x Your Account Is Home Actions View Communicate Files & Extras host:44302/Regi x

localhost:44302/BrandMVC/AddIndex

Doctor Profile Doctor List Patient Profile Patient List Show Prescription Manage New Medicine Show Medicine shuvo Logout

Enter Brand Name

BrandName

[Back to List](#)

© 2019 - Health Care Application

View of Medicine:

[Manage New Medicine](#) [Show Medicine](#) [farabi](#) [Logout](#)

Drug

Name	GenericName	BrandName	Strength	Type
Napa	Hamba	Square Ltd.	12	Tablet
Pantonix	Bahaso	Square Ltd.	40	Tablet

[Manage New Medicine](#) [Show Medicine](#) [farabi](#) [Logout](#)

Drug List

[Create New](#)

Select	Drug Name	Strength	Type	Generic_id	Brand_id	
>>>	Napa	12	Tablet	3	2	Edit Details Delete
>>>	Pantonix	40	Tablet	4	2	Edit Details Delete

Chapter-7: Project Summary

7.1 Github Link

<https://github.com/mokabbir3485/HealthCareService.git>

7.2 Critical Evolution

For my project, I have faced some critical evolutions. Those are :

1. Achieving goals of project initiatives.
2. Allocating resources.
3. Anticipating & preventing errors.
4. Using & managing information systems.
5. Understanding treatment implications.
6. Developing data driving protocols.
7. Anticipating the implications of actions taken.
8. Analyzing & resolving personnel issues.
9. Explaining policy & protocols.
10. Designing & evaluating individual parts.

7.3 Limitations

My project is not a complete project. So now a user can't go for an appointment system & they can't get free prescription. So they can just register & doctor will suggest them their problems. Functional limitations are doctors & patients can registration & Login , they can fill up their personal information. Doctors can enroll prescription to their patients. Doctors & Patients both can see prescriptions. Admin can input medicines.

7.4 Obstacles & Achievements

To make this system, I face some obstacles. That is I face a challenge to integrate what I are making with tools for my customers may already be using. So I have observed when it comes to launching a new software package is the desire to put too much into one application. And Achievements are I understand that what people wants & what they expect from our system. I hope that our system will make happy our users.

7.5 Future Scope

I have a better plan with this project. I will try to add new features & also I will try to make our system more reliable so that user can use our system easily & be happy with our project. I have a future plan that is I will try to make an Appointment system so that patient can appoint with doctor & get serial easily.

Reference :

1. <https://www.doctorkoi.com/#/>
2. <https://www.jeeon.co/projotno-telemedicine-app/>
3. <http://itmedicus.com/portfolio/dims/>
4. <https://www.icliniq.com/ask-a-doctor-online>