Remote Doctor

By

Raihan Mahmud
(161-35-1542)

A project submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Fall – 2019

DECLARATION

We hereby declare that we have taken this project under the supervisor of **Md. Anwar Hossen**, Senior Lecturer, **Department of Software Engineering**, **Daffodil International University**. We also declare that neither this thesis/project nor any part of thesis has been submitted elsewhere for award of any degree.

Raihan Mahmud

Raihan Mahmud
ID: 161-35-1542
Batch: 19th
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Certified by:

Md. Anwar Hossen
Senior Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Acknowledgement

First of all, I am thankful to the Almighty Allah for making me eligible to complete this project. Then I would like to thank my supervisor **Md. Anwar Hossen**, Senior Lecturer, Department of Software Engineering. I am grateful and responsible to him for his expert, sincere and valuable guidance and encouragement extended to me.

I would like to express my special thanks them who were helped in my project by their very important suggestions without their passionate participation and input; the project could not be successfully guided. I take this opportunity to record my sincere thanks to all the faculty members of the Department of Software Engineering for their help and encouragement.

Table of Contents

APPROVAL i

DECLARATION...ii

Acknowledgement iii

Chapter 1 1

1.1 Project Overview 2

1.2 Project Purpose 2

1.2.1 Background 2

1.2.2 Benefits and Beneficiaries..2

1.2.3 Goals 3

1.3 Stakeholders 3

1.4 Proposed System Model 4

1.5 Project Schedule 4

1.5.1 Gantt Chart...5

1.5.2 Release Plan or Milestone...6

Chapter 2 7

2.1 Functional Requirements 8

2.1.1 Priority Chart 8

2.1.2.1 Login for Admin 9

2.1.2.2 Add Doctor...9

2.1.2.3 View Doctors Information 9

2.1.2.4 Update Doctor Information..9

2.1.2.5 Delete Doctor Profile 9

2.1.2.6 Add Coordinator...9

2.1.2.7 View Coordinators Information...10

2.1.2.8 Update Coordinator Information..10

2.1.2.9 Delete Coordinator Profile 10

2.1.2.10 Add Patient 10

2.1.2.11 View Patients Information..10

2.1.2.12 Update Patient Information...10

2.1.2.13 Delete Patient Profile 11

2.1.2.14 View Transactions History 11

2.1.2.15 Login for Coordinator 11

2.1.2.16 Login for Patient..11

2.1.2.17 Login for Doctor 11

2.1.2.18 Search Doctor 11
2.1.2.19 Take an Appointment 12
2.1.2.20 Appointment Request 12
2.1.2.21 Upload Report or Previous Prescription 12
2.1.2.22 Live conference 12
2.1.2.23 Prescription or Refer 12
2.2 Data Requirements 12
2.3 Performance Requirements 13
2.3.1 Speed and Latency Requirements 13
2.3.2 Precision or Accuracy Requirements 13
2.3.3 Capacity Requirements 13
2.4 Dependability Requirements 14
2.4.1 Reliability Requirements 14
2.4.2 Availability Requirements 14
2.4.3 Robustness or Fault-Tolerance Requirements 14
2.5 Maintainability & Supportability Requirements 14
2.5.1 Maintenance Requirements 14
2.5.2 Supportability Requirements 15
2.6 Security Requirements 15
2.6.1 Access Requirements 15
2.6.2 Integrity Requirements 15
2.6.3 Privacy Requirements 15
2.7 Usability and Human-Interaction Requirements 15
2.7.1 Ease of Use Requirements 15
2.7.2 Personalization and Internationalization Requirements 16
2.7.3 Understandability and Politeness Requirements 16
2.7.4 Accessibility Requirements 16
2.7.5 User Documentation Requirements 16
2.7.6 Training Requirements 16
Chapter 3 17
3.1 Use Case Diagram 18
Use Case Description 19
3.1.1 User Login 19
3.1.2 Manage Doctor 19
3.1.3 Manage Coordinator 20

3.1.4 Manage Patient 20
3.1.5 Manage Department 21
3.1.6 Manage Transaction 21
3.1.7 Manage Appointment 21
3.1.8 Manage Prescription 22
3.1.9 Manage Report 22
3.1.10 Live Conference 23
3.2 Activity Diagram 24
3.2.1 User Login 24
3.2.2 Manage Doctor 25
3.2.3 Manage Coordinator 26
3.2.4 Manage Patient 27
3.2.5 Manage Department 28
3.2.6 Manage Transaction 29
3.2.7 Manage Appointment 30
3.2.8 Manage Prescription 31
3.2.9 Manage Report 32
3.2.10 Live Conference 33
3.3 Sequence Diagram 34
3.3.1 User Login 34
3.3.2 Manage Doctor 35
3.3.3 Manage Coordinator 36
3.3.4 Manage Patient 37
3.3.5 Manage Department 38
3.3.6 Manage Appointment 39
3.3.7 Manage Prescription 40
3.3.8 Manage Report 41
3.3.9 Live Conference 42
Chapter 4 43
4.1 Development Tools and Technology 44
4.1.1 User Interface Technology 44
4.1.1.1 Front-end 44
4.1.1.2 Programming Language 44
4.1.2 Implemented Tools and Platform 44
4.1.2.1 IDE 44

4.1.2.2 Web Server 44
4.1.2.3 Database Server 44
4.2 Class Diagram 45
4.3 Entity Relationship Diagram (ERD) 46
Chapter 5 47
5.1 Testing Features 48
5.1.1 Feature to be tested 48
5.2 Testing Strategy 49
5.2.1 Test approach 49
5.2.1.1 Black Box Testing 49
5.2.1.2 White Box Testing 50
5.2.2 Pass or Fail Criteria 50
5.3 Testing Schedule.. 50
5.4 Trace Ability Matrix 50
5.5 Testing Environment 53
5.6 Test Cases 53
5.6.1 Login for Admin 53
5.6.2 Add Department.. 54
5.6.3 View Departments 54
5.6.4 Delete Department 55
5.6.5 Doctors Registration 55
5.6.6 View Doctors List 56
5.6.7 Update Doctor Info 56
5.6.8 Delete Doctor Profile 56
5.6.9 Patients Registration 57
5.6.10 View Patients List 57
5.6.11 Update Patient Info 58
5.6.12 Delete Patient Profile 58
5.6.13 Coordinators Registration 59
5.6.14 View Coordinators List 59
5.6.15 Update Coordinator Info 60
5.6.16 Delete Coordinator Profile 60
5.6.17 Registration for Doctor 61
5.6.18 Login for Doctor 61
5.6.19 View Profile for Doctor 61

5.6.20 Update Profile for Doctor 62
5.6.21 Patient History 62
5.6.22 Accept Appointment Request 63
5.6.23 Prescribe 63
5.6.24 Live Conference for Doctor 64
5.6.25 Registration for Patient 64
5.6.26 Login for Patient 65
5.6.27 View Profile for Patient 65
5.6.28 Update Profile for Patient 66
5.6.29 Take Appointment for Patient 66
5.6.30 Submit Report 66
5.6.31 View Prescription 67
5.6.32 Live Conference for Patient 67
5.6.33 Registration for Coordinator 68
5.6.34 Login for Coordinator 68
5.6.35 View Profile for Coordinator 69
5.6.36 Update Profile for Coordinator 69
5.6.37 Take Appointment for Coordinator 70
Chapter 6 71
6.1 Home Page 72
6.1.1 Doctors Page 72
6.1.2 Contact Page 73
6.2 Admin module 73
6.2.1 Login for Admin 73
6.2.2 Admin Panel 74
6.2.3 Doctors 74
6.3 Doctor module 75
6.3.1 Login for Doctor 75
6.3.2 Doctor Dashboard 75
6.3.3 Doctor Profile 76
6.3.4 Treatment History 76
6.3.5 Appointment 77
6.3.6 Appointment Requests 77
6.3.7 Treatment 78
6.3.8 Live Conference 78

6.4 Patient module 79

6.4.1 Login for Patient 79

6.4.2 Patient Dashboard 79

6.4.3 Patient Profile 80

6.4.4 Appointment 80

6.4.5 Reports 81

6.4.6 Prescription 81

6.4.7 Live Conference 82

6.5 Coordinator module 82

6.5.1 Login for Coordinator 82

6.5.2 Coordinator Dashboard 83

6.5.3 Coordinator Profile 83

6.5.4 Appointment 84

Chapter 7 85

7.1 GitHub Link 86

7.2 Project Summary 86

7.3 Limitations 86

7.4 Obstacles and Achievements 86

7.5 Future Scope 87

Chapter 1

Introduction

1.1 Project Overview

Healthcare should be simple, fast and uncomplicated. But in Bangladesh, the medical sector is not that fast and affordable for the general people of our country. So we have decided to solve this existing problem by a web application named Remote Doctor.

This system will make it easy to visit a doctor in minutes through our web and get access to quality healthcare without ever leaving your home, your job or wherever you are. You can't plan when you'll get sick, but you can take your doctor with you and get the care you need anywhere in Bangladesh through this system.

The main purpose of this project is to utilize modern technology to take medical expertise to remote communities in Bangladesh, which in other words, means reshaping and improving primary health care for all.

The existing system is based on manual work and all the processes are done manually, so they maintain registers and files for recording all the details of the system. But this system is a lengthy process and like any time lose all data. So when they are using this website then they can easily solve their problem.

1.2 Project Purpose

Remote doctor system development has begun to change the healthcare industry landscape. Remote doctor is a subset of telehealth, which uses a broad range of modern technologies and specializes in providing medical services from a distance with the help of software and communication tools.

1.2.1 Background

Due to the huge number of population, we face many problems in the medical sector. It becomes more difficult to find a specialist doctor in rural areas because most of the time a specialist is not available. Using this system people of rural areas can easily get a specialist doctors' appointment and treatment.

1.2.2 Benefits and Beneficiaries

The main objective is to utilize modern technology to take medical expertise to remote communities in Bangladesh, which in other words, means reshaping and improving primary health care for all. Benefits and beneficiaries are given below:

Doctor create her schedule

Patient can see doctor details to choose doctor

Patient can take appoint via Coordinator

Doctor can check appointments list for accept appointment

Doctor can see patient live pathology report

Doctor can prescribe or refer to another doctor

Appointment via video or chatting

Patient can upload reports and previous prescription

Doctor can check patients previous medical history

Admin can see all record history

1.2.3 Goals

Using Remote Doctor system easily takes medical treatment rural area people. And another side we want to extend this service in world wide. Like anyone can want to take the appointment in USA doctor then he or she can be using this system than can easily be taking an appointment.

1.3 Stakeholders

Four stakeholder in this system. They are given below:

Admin

Patient

Coordinator

Doctor

Admin: Admin can monitor and control all user information of this system. Also can add Department, Doctor, Patient, Coordinators. And see all transaction information.

Patient: Patients can see doctor details to choose a doctor. After registration patient can take an appointment. Get treatment using video or text conversation, also can submit pathology reports, upload reports, and previous prescriptions.

Coordinator: Patients can take appointments via registered Coordinators. Submit patient pathology reports, upload reports and previous prescriptions which appointment was taken by the coordinator.

Doctor: A registered doctor can accept appointment request, after accepting request can see patient pathology reports. Using video or text conversation doctor can give treatment or refer to another doctor.

1.4 Proposed System Model

This model describe how to works full system.

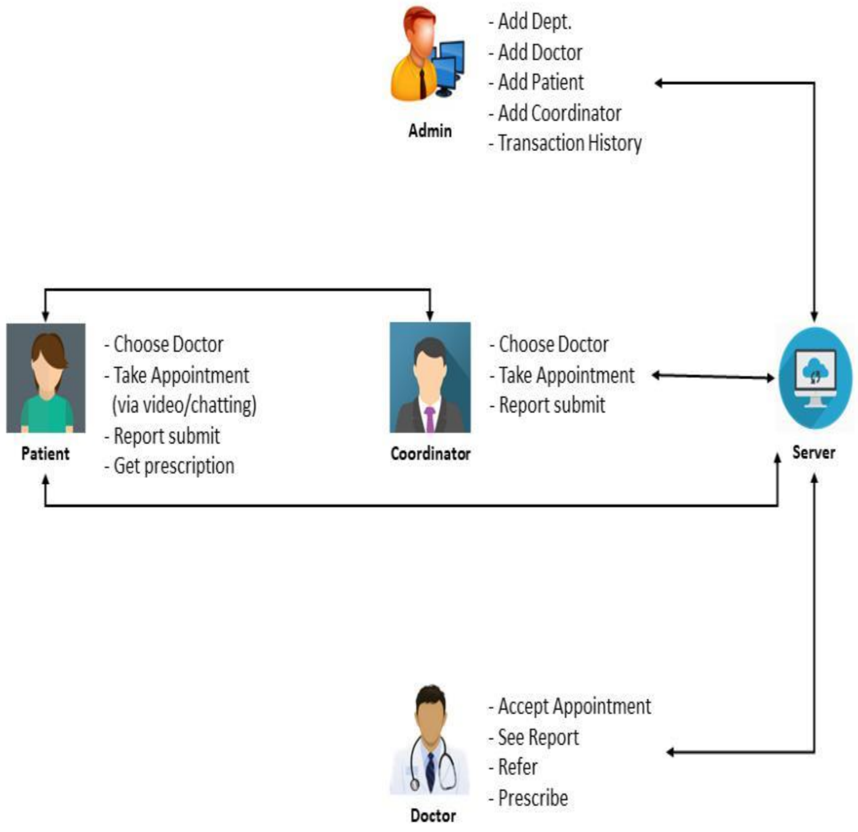


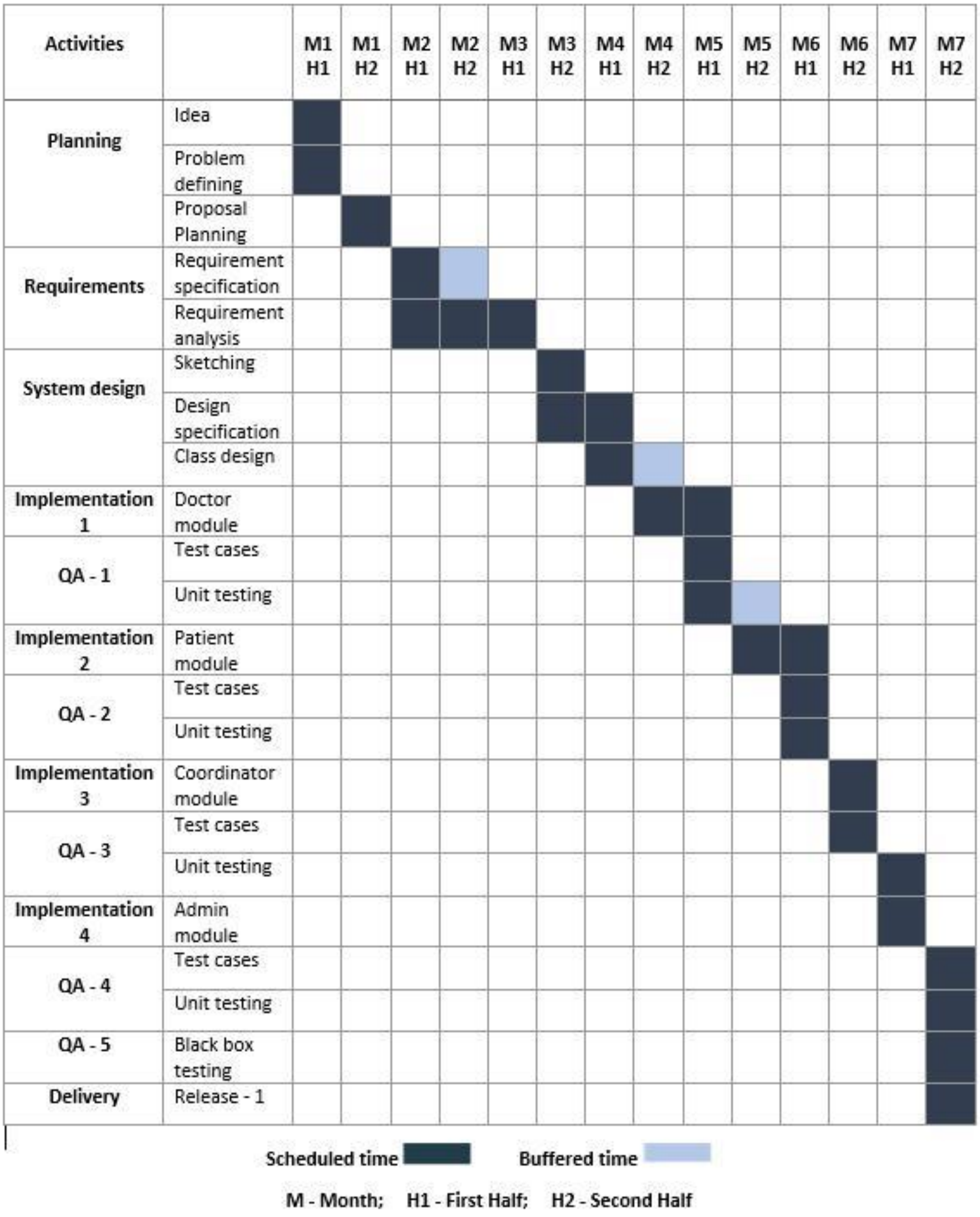
Figure 1.1: Proposed system model.

1.5 Project Schedule

In software development project scheduling is very important, for completing the project at perfect time. In our project, we make a Gantt chat and set release plan for as per version release. Here we show first version of our project Gantt chart and release plan.

1.5.1 Gantt Chart

This project I follow Gantt chart for minting time and release process properly. Our Gantt chart given below:



Activities		M1 H1	M1 H2	M2 H1	M2 H2	M3 H1	M3 H2	M4 H1	M4 H2	M5 H1	M5 H2	M6 H1	M6 H2	M7 H1	M7 H2
Planning	Idea	■													
	Problem defining	■													
	Proposal Planning		■												
Requirements	Requirement specification			■	░										
	Requirement analysis			■	■	■									
System design	Sketching						■								
	Design specification						■	■							
	Class design							■	░						
Implementation 1	Doctor module								■	■					
QA - 1	Test cases									■					
	Unit testing									■	░				
Implementation 2	Patient module										■	■			
QA - 2	Test cases											■			
	Unit testing											■			
Implementation 3	Coordinator module												■		
QA - 3	Test cases												■		
	Unit testing													■	
Implementation 4	Admin module													■	
QA - 4	Test cases														■
	Unit testing														■
QA - 5	Black box testing														■
Delivery	Release - 1														■

Scheduled time ■ Buffered time ░

M - Month; H1 - First Half; H2 - Second Half

Figure 1.2: Gantt chart.

1.5.2 Release Plan or Milestone

First version of our project release plan or milestone given below:

Activities	Duration in Month	Total Month
Planning	First month	1
Requirements	Second month – Third month(First Half)	1.5
System design	Third month(Second Half) – Fourth month	1.5
Implementation	Fourth month(Second Half) – Seventh month(First Half)	3
Quality Assurance	Fifth month – Seventh month	3 (Include in implementation)
Release - 1	Seventh month	7

Chapter 2

Software **R**equirements **S**pecification (**SRS**)

2.1 Functional Requirements

Functional requirement means basic behavior of the system. In our project have functional requirements. Now, we are going to discuss functional requirements of our project.

2.1.1 Priority Chart

Using priority matrix we can easily find out which function should get High Priority, which one is Medium and which one is Low.

In Priority Matrix there is two part one is “Important” another one is “Urgent”.
If any function is:

Important also Urgent	High Priority
Important but Not Urgent	Medium Priority
Not Important but Urgent	Low Priority
Not Important and Not Urgent	Ignore

Priority of our system features based on Priority Chart:

Functional Requirement ID	Important	Urgent	Priority
FR01	Yes	Yes	High
FR02	Yes	Yes	High
FR03	No	Yes	Medium
FR04	No	Yes	Medium
FR05	Yes	Yes	High
FR06	Yes	Yes	High
FR07	No	Yes	Medium
FR08	No	Yes	Medium
FR09	Yes	Yes	High
FR10	Yes	Yes	High
FR11	No	Yes	Medium
FR12	No	Yes	Medium
FR13	Yes	Yes	High
FR14	Yes	Yes	High
FR15	Yes	Yes	High
FR16	Yes	Yes	High
FR17	Yes	Yes	High
FR18	No	Yes	Medium
FR19	Yes	Yes	High
FR20	Yes	Yes	High
FR21	Yes	Yes	High
FR22	Yes	Yes	High
FR23	Yes	Yes	High

2.1.2.1 Login for Admin

FR01	**Login for Admin**		
Description	System admin created admin account and provide username, password to admin. Then admin can login to admin panel.		
Stakeholders	Admin	**Priority**	High

2.1.2.2 Add Doctor

FR02	**Add Doctor**		
Description	Admin or doctor can create a doctor account and provide email, password to the doctor. Then doctor can verify his account and login to the doctor panel.		
Stakeholders	Admin, Doctor	**Priority**	High

2.1.2.3 View Doctors Information

FR03	**View Doctors Information**		
Description	Admin can view all registered doctors information. Like name, reg no etc.		
Stakeholders	Admin	**Priority**	Medium

2.1.2.4 Update Doctor Information

FR04	**Update Doctor Information**		
Description	Admin or doctor can edit or add all registered doctor information. Like name, reg no etc.		
Stakeholders	Admin, Doctor	**Priority**	Medium

2.1.2.5 Delete Doctor Profile

FR05	**Delete Doctor Profile**		
Description	Only admin can delete doctor profile.		
Stakeholders	Admin	**Priority**	High

2.1.2.6 Add Coordinator

FR06	**Add Coordinator**		
Description	Admin or coordinator can create a coordinator account and provide email, password to the coordinator. Then coordinator can verify his account and login to the coordinator panel.		
Stakeholders	Admin, Coordinator	**Priority**	High

2.1.2.7 View Coordinators Information

FR07	**View Coordinators Information**		
Description	Admin can view all registered coordinators information. Like name, address etc.		
Stakeholders	Admin	**Priority**	Medium

2.1.2.8 Update Coordinator Information

FR08	**Update Coordinator Information**		
Description	Admin or coordinator can edit or add all registered coordinator information. Like name, address etc.		
Stakeholders	Admin, Coordinator	**Priority**	Medium

2.1.2.9 Delete Coordinator Profile

FR09	**Delete Coordinator Profile**		
Description	Only admin can delete coordinator profile.		
Stakeholders	Admin	**Priority**	High

2.1.2.10 Add Patient

FR10	**Add Patient**		
Description	Admin or patient can create a patient account and provide email, password to the patient. Then patient can verify his account and login to the patient panel.		
Stakeholders	Admin, Patient	**Priority**	High

2.1.2.11 View Patients Information

FR11	**View Patients Information**		
Description	Admin can view all registered patients information. Like name, address etc.		
Stakeholders	Admin	**Priority**	Medium

2.1.2.12 Update Patient Information

FR12	**Update Patient Information**		
Description	Admin or patient can edit or add all registered patient information. Like name, address etc.		
Stakeholders	Admin, Patient	**Priority**	Medium

2.1.2.13 Delete Patient Profile

FR13	**Delete Patient Profile**		
Description	Only admin can delete patient profile.		
Stakeholders	Admin	**Priority**	High

2.1.2.14 View Transactions History

FR14	**View Transactions History**		
Description	Only admin can view all transactions history.		
Stakeholders	Admin	**Priority**	High

2.1.2.15 Login for Coordinator

FR15	**Login for Coordinator**		
Description	After creating account coordinator can login to her panel using email and password.		
Stakeholders	Coordinator	**Priority**	High

2.1.2.16 Login for Patient

FR16	**Login for Patient**		
Description	After creating account patient can login to her panel using email and password.		
Stakeholders	Patient	**Priority**	High

2.1.2.17 Login for Doctor

FR17	**Login for Doctor**		
Description	After creating account doctor can login to her panel using email and password.		
Stakeholders	Doctor	**Priority**	High

2.1.2.18 Search Doctor

FR18	**Search Doctor**		
Description	Coordinator or patient can search doctor by doctor name.		
Stakeholders	Coordinator, Patient	**Priority**	Medium

2.1.2.19 Take an Appointment

FR19	**Take an Appointment**		
Description	For taking an appointment search and choose doctor then sent request for the appointment.		
Stakeholders	Coordinator, Patient	**Priority**	High

2.1.2.20 Appointment Request

FR20	**Appointment Request**		
Description	Doctor can accept appointment request and set appointment schedule for treatment.		
Stakeholders	Doctor	**Priority**	High

2.1.2.21 Upload Report or Previous Prescription

FR21	**Upload Report or Previous Prescription**		
Description	Coordinator or patient can upload report or previous prescription.		
Stakeholders	Coordinator, Patient	**Priority**	High

2.1.2.22 Live conference

FR22	**Live conference**		
Description	Patient can share her problem using video/live chat with doctor.		
Stakeholders	Doctor, Patient	**Priority**	High

2.1.2.23 Prescription or Refer

FR23	**Prescription or Refer**		
Description	Doctor can prescribe to a patient or refer to another doctor.		
Stakeholders	Doctor	**Priority**	High

2.2 Data Requirements

For achieve the model objectives data requirements prescribed scope and level of detail required. What data is required for building the model can be known by gathering system data. Instead of gathering general system data it’s better to gathering specific system data. An overall process flow can provide more detailed information.

We focus some of main points. Such as:

Route locations and their
capacity Routing decision
The type of entity and their attributes
Types of resources
Quantity of data
What activity
Entities movement

2.3 Performance Requirements

Following performance requirement to maintain performance of a software. In our project most of the features must be maintain performance requirements.

2.3.1 Speed and Latency Requirements

For user satisfaction an application response time is very important.

SLR01	**Live conference connected less than 5 seconds**		
Description	When patient and doctor got to the live conference, the call will be connected in less than 5 seconds.		
Stakeholders	Doctor, Patient	**Priority**	High

SLR02	**Text conversation sent and receive in less than 2 seconds**		
Description	When patient and doctor are in the text conversation, the message will be sent and receive less than 2 seconds.		
Stakeholders	Doctor, Patient	**Priority**	High

2.3.2 Precision or Accuracy Requirements

PAR01	**Appointment information must be accurate**		
Description	Doctor received appointment information must be accurate.		
Stakeholders	Doctor	**Priority**	High

PAR02	**Prescription must be accurate**		
Description	Patient received prescription data must be accurate.		
Stakeholders	Patient	**Priority**	High

2.3.3 Capacity Requirements

CR01	**Certain amount of appointment take**		
Description	Coordinator can take up to 10 appointments and patient can take up to 3 appointments daily.		
Stakeholders	Coordinator, Patient	**Priority**	High

2.4 Dependability Requirements

The dependability is measured based on four dimensions. Such as:

- **Reliability**
- **Availability**
- **Durability**
- **Security**

Safety

2.4.1 Reliability Requirements

RR01	**System must be failure-free**		
Description	The system must be failure-free operation system and technical error free over a specified time within this type of environment.		
Stakeholders	Admin, Doctor, Coordinator and Patient	**Priority**	High

2.4.2 Availability Requirements

AR01	**The ability of the system to deliver service when requested**		
Description	The system must be available on 24 X 7. The ability of the system to deliver service when requested.		
Stakeholders	Admin, Doctor, Coordinator and Patient	**Priority**	High

2.4.3 Robustness or Fault-Tolerance Requirements

RFTR01	**Ensure that system can handle all user access or any external tolerance.**		
Description	The system can cope with error during execution. Also ensure that the system continue properly even of the failure of some of its components.		
Stakeholders	Admin, Doctor, Coordinator and Patient	**Priority**	High

2.5 Maintainability & Supportability Requirements

Maintenance is the activity of modifying a software product after initial delivery.

2.5.1 Maintenance Requirements

Its main purpose is to modify and update software application after delivery to correct faults and to improve performance. Ensure the all type of maintenance requirements

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

2.5.2 Supportability Requirements

Software Supportability is the capability of supporting a software system over its whole product life. It covers the following key aspects associated to the system

Operation (install, loading or unloading, configuration, error recovery and execution) Logistics Management (once a new baseline has been produced)
Modification (fixing bugs, or adding/changing functionality due to changing user needs)

2.6 Security Requirements

System security is most important requirements. For lack of security the data would be stealing, manipulating data and causing denial of service.
Some security related categories

Permission to access
data Verification
Securing information
Security policies

2.6.1 Access Requirements

Reduce unauthorized access there have some barrier. There remain some authentication and authorization techniques.

Strong Access Control Measures
Maintain an Information Security Policy
Regularly Monitor and Test Networks
Build and Maintain a Secure Network

2.6.2 Integrity Requirements

Ensure that data and communications are not intentionally corrupted via unauthorized creation, deletion, modification.
We use Hash value for sensitive data.

2.6.3 Privacy Requirements

Privacy requirements are mostly needed for a system. In our system we associate the admin, doctor, patient and coordinator area and everyone can see only the area that is associated with them.

2.7 Usability and Human-Interaction Requirements

The system can be failed for Usability. User Experience is one of the most important factors to any system. The system must be easy to use, easy to understand and easy to learn.

2.7.1 Ease of Use Requirements

Our system is very easy to use and easy to understandable. There is no long process to complete a task. User can complete their action with a few steps.

2.7.2 Personalization and Internationalization Requirements

In our Project there are no personalization and internationalization requirements.

2.7.3 Understandability and Politeness Requirements

We made this system based on a targeted area and all of those targeted area people have known about our system because this system is make their life easier. User can understand this system by own.

2.7.4 Accessibility Requirements

The requirements for how easy it should be for people and our system is very much easy to use and understand.

2.7.5 User Documentation Requirements

Gather the user expects and must do it in system properly is mandatory because this requirement given by user directly. And after the complete the system user can give new requirements then it’s also be added in system.

2.7.6 Training Requirements

If it’s need to training the user for use the system properly then it’s must to training them properly.

In our project we don’t need this type of training for user but a simple guide for them.

Chapter 3

Requirement Analysis

3.1 Use Case Diagram

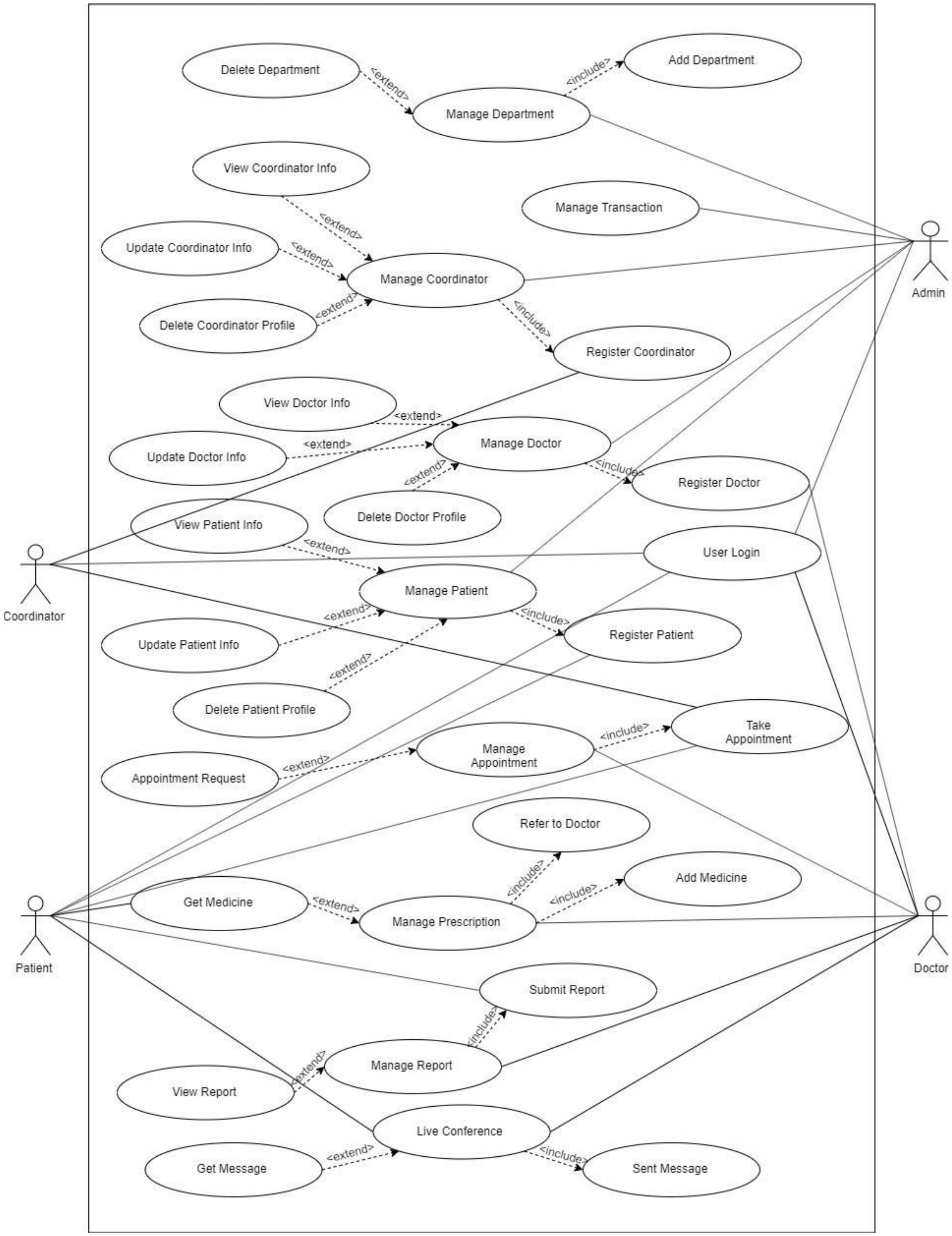


Figure 3.1: Use case diagram.

Use Case Description

3.1.1 User Login

Use Case Title	User Login	
Goal	User successfully login to the user panel	
Preconditions	User must have an verified account	
Success End Condition	User access the user panel	
Failed End Condition	Email and password invalid	
Primary Actors: **Secondary Actors:**	Coordinator, Patient Admin, Doctor	
Trigger	Login	
Description / Main Success Scenario	**Step**	**Action**
	1	System show login page
	2	Enter email, password and click login button
	3	System show user panel
Alternative Flows	**Step**	**Branching Action**
	3a	System show email and password invalid
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.2 Manage Doctor

Use Case Title	Manage Doctor	
Goal	Control doctors profile	
Preconditions	Admin must have logged in	
Success End Condition	View doctors information	
Failed End Condition	Not show registered doctors information	
Primary Actors: **Secondary Actors:**	Doctor Admin	
Trigger	Doctors	
Description / Main Success Scenario	**Step**	**Action**
	1	Register doctor account
	2	View doctors information
	3	Update doctor information
	4	Delete doctor profile
Alternative Flows	**Step**	**Branching Action**
	1a	Email already exist
	3a	Fill up required fields
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.3 Manage Coordinator

Use Case Title	Manage Coordinator	
Goal	Control coordinator profile	
Preconditions	Admin must have logged in	
Success End Condition	View coordinators information	
Failed End Condition	Not show registered coordinators information	
Primary Actors: **Secondary Actors:**	Coordinator Admin	
Trigger	Coordinators	
Description / Main Success Scenario	**Step**	**Action**
	1	Register coordinator account
	2	View coordinators information
	3	Update coordinator information
	4	Delete coordinator profile
Alternative Flows	**Step**	**Branching Action**
	1a	Email already exist
	3a	Fill up required fields
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.4 Manage Patient

Use Case Title	Manage Patient	
Goal	Control patient profile	
Preconditions	Admin must have logged in	
Success End Condition	View patients information	
Failed End Condition	Not show registered patients information	
Primary Actors: **Secondary Actors:**	Patient Admin	
Trigger	Patients	
Description / Main Success Scenario	**Step**	**Action**
	1	Register patient account
	2	View patients information
	3	Update patient information
	4	Delete patient profile
Alternative Flows	**Step**	**Branching Action**
	1a	Email already exist
	3a	Fill up required fields
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.5 Manage Department

Use Case Title	Manage Department	
Goal	Control all department	
Preconditions	Admin must have logged in	
Success End Condition	View departments information	
Failed End Condition	Not show registered departments information	
Primary Actors: **Secondary Actors:**	N/A Admin	
Trigger	Departments	
Description / Main Success Scenario	**Step**	**Action**
	1	Added department and department details
	2	View departments information
	3	Delete department
Alternative Flows	**Step**	**Branching Action**
	1a	Department already exist
	1a	Fill up required fields
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.6 Manage Transaction

Use Case Title	Manage Transaction	
Goal	Control all transaction	
Preconditions	Admin must have logged in	
Success End Condition	View transactions history	
Failed End Condition	Not show previous transaction history	
Primary Actors: **Secondary Actors:**	N/A Admin	
Trigger	Transactions	
Description / Main Success Scenario	**Step**	**Action**
	1	Show total transactions list
Alternative Flows	**Step**	**Branching Action**
	1a	Expired session
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.7 Manage Appointment

Use Case Title	Manage Appointment
Goal	Successfully take an appointment
Preconditions	Patient/Coordinator and doctor must have logged in
Success End Condition	Patient or Coordinator take an appointment

Failed End Condition	Submit invalid information	
Primary Actors: **Secondary Actors:**	Patient, Coordinator Doctor	
Trigger	Appointments	
Description / Main Success Scenario	**Step**	**Action**
	1	Patient or Coordinator search and sent appointment request to a doctor.
	2	Doctor accept appointment request
	3	Patient or Coordinator submit required information of appointment.
Alternative Flows	**Step**	**Branching Action**
	3a	Patient or Coordinator submit invalid required information of appointment.
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.8 Manage Prescription

Use Case Title	Manage Prescription	
Goal	Providing treatment to the patient	
Preconditions	Patient and doctor must have logged in	
Success End Condition	Providing prescribe to the patient	
Failed End Condition	Not providing prescribe to the patient	
Primary Actors: **Secondary Actors:**	Patient Doctor	
Trigger	Prescription	
Description / Main Success Scenario	**Step**	**Action**
	1	Click solution and provide medicine to the patient.
Alternative Flows	**Step**	**Branching Action**
	1a	Click refer and referred to another doctor
	1a	Fill up required fields
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.9 Manage Report

Use Case Title	Manage Report
Goal	Providing treatment to the patient
Preconditions	Patient and doctor must have logged in
Success End Condition	Providing report to the doctor
Failed End Condition	Not providing report to the doctor

Primary Actors: **Secondary Actors:**	Patient Doctor	
Trigger	Reports	
Description / Main Success Scenario	**Step**	**Action**
	1	Doctor give medical test to the patient
	2	Patient submit the test report
	3	Doctor provide medicine to the patient
Alternative Flows	**Step**	**Branching Action**
	3a	Click refer and referred to another doctor
Quality Requirements	**Step**	**Requirements**
	1	N/A

3.1.10 Live Conference

Use Case Title	Live Conference	
Goal	Conference with expected user	
Preconditions	Patient and doctor must have logged in	
Success End Condition	Successfully connected with expected user	
Failed End Condition	Connection time out	
Primary Actors: **Secondary Actors:**	Patient Doctor	
Trigger	Chat	
Description / Main Success Scenario	**Step**	**Action**
	1	Take appointment get connected expected user
	2	Select user and sent message or video call
	3	Get received message or video call
Alternative Flows	**Step**	**Branching Action**
	2a	Time out of video calling
Quality Requirements	**Step**	**Requirements**
	1	Wait 2 second for connected

3.2 Activity Diagram

We are preparing activity diagram as per use case. Those activity diagrams represent workflow of behavior our system.

3.2.1 User Login

Users login to the user panel through email and password that are prepared after verifying the account.

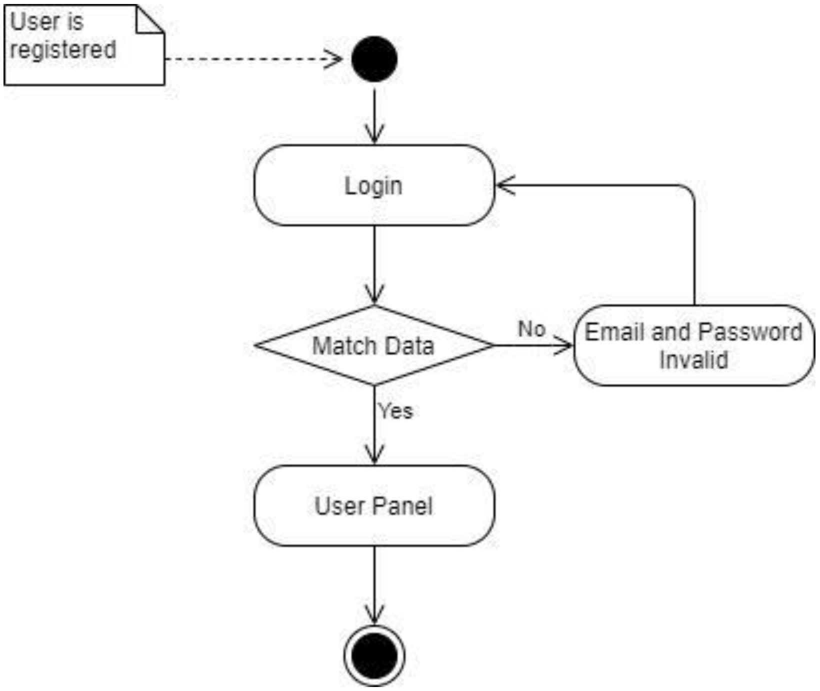


Figure 3.2: Activity diagram for User Login.

3.2.2 Manage Doctor

Admin fully control and monitor doctors profile for verification purpose.

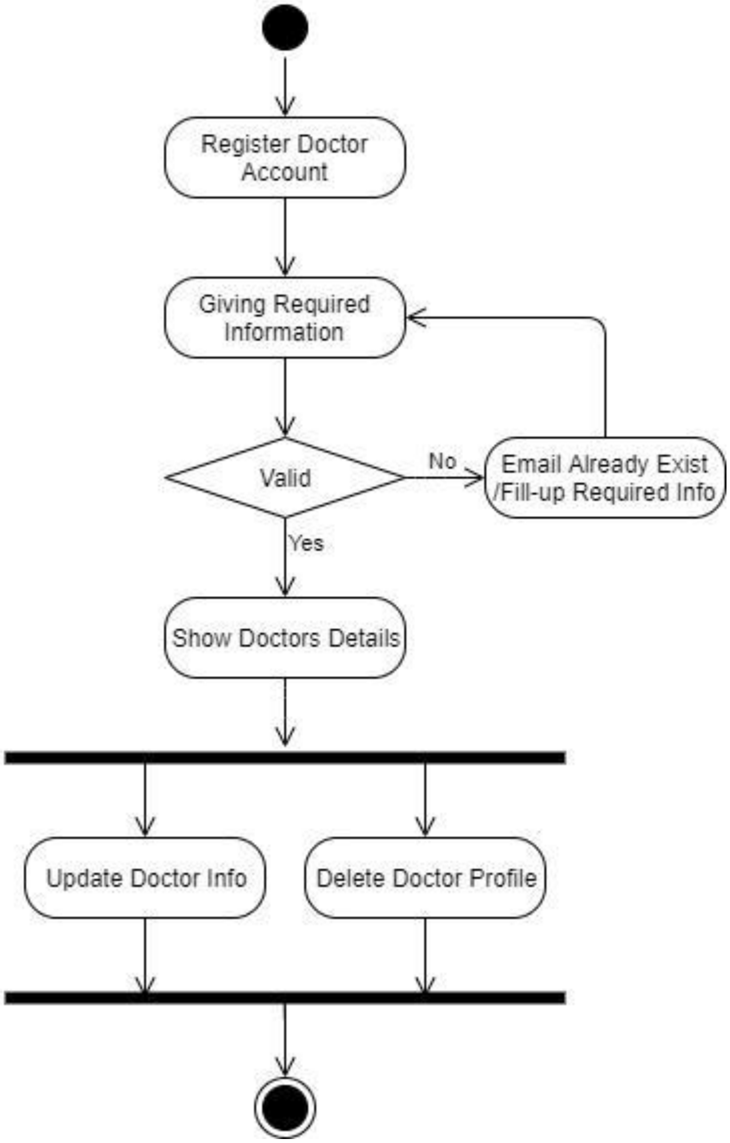


Figure 3.3: Activity diagram for Manage Doctor.

3.2.3 Manage Coordinator

Admin fully control and monitor coordinator profile for verification purpose.

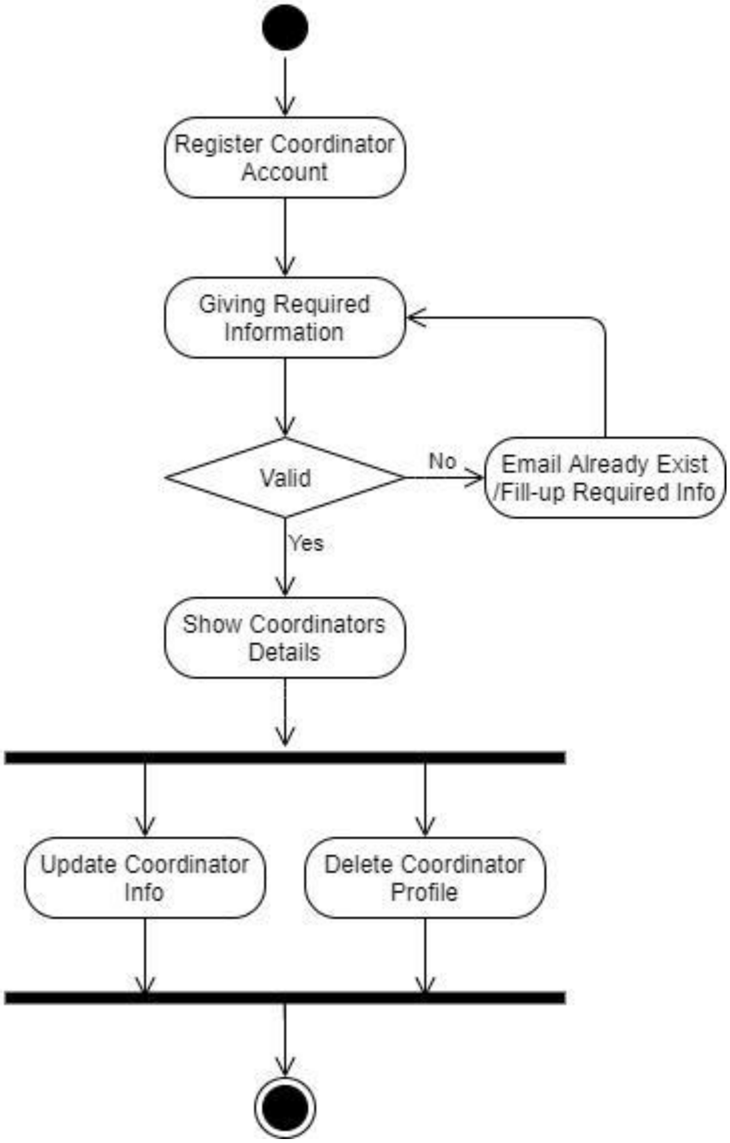

Register Coordinator Account
Giving Required Information
Valid
No
Email Already Exist /Fill-up Required Info
Yes
Show Coordinators Details
Update Coordinator Info
Delete Coordinator Profile

Figure 3.4: Activity diagram for Manage Coordinator.

3.2.4 Manage Patient

Admin fully control and monitor patient profile for verification purpose.

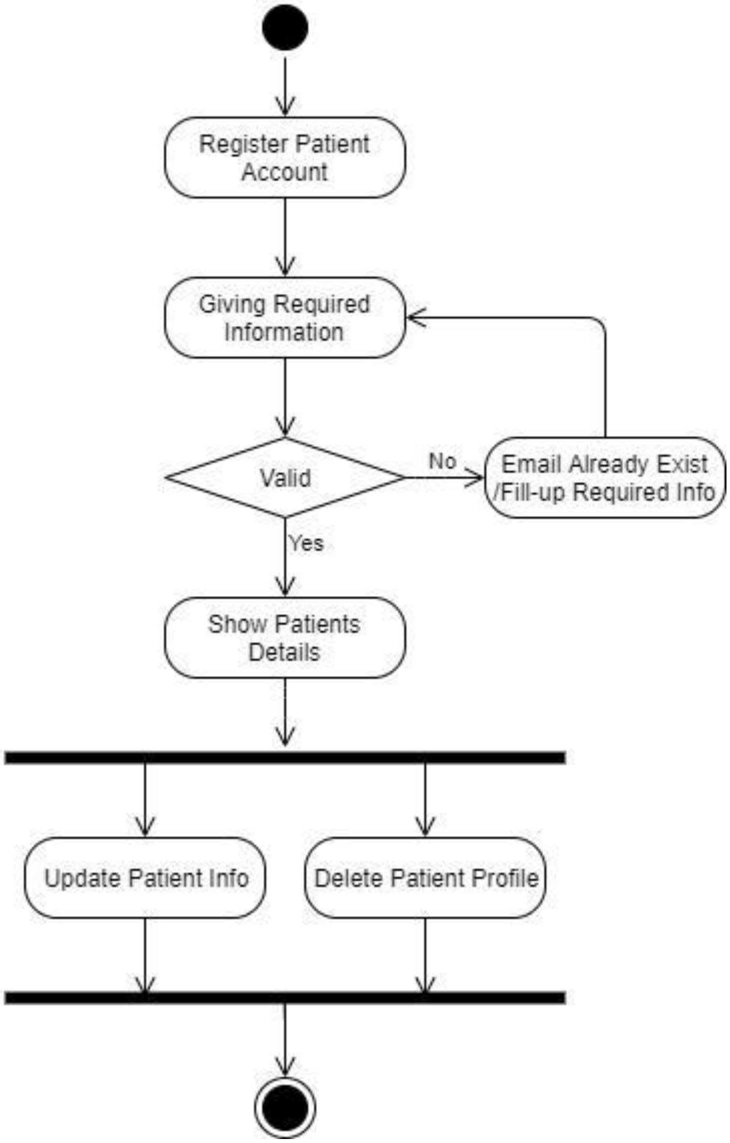


Figure 3.5: Activity diagram for Manage Patient.

3.2.5 Manage Department

Admin fully control department list, added new department or delete department.

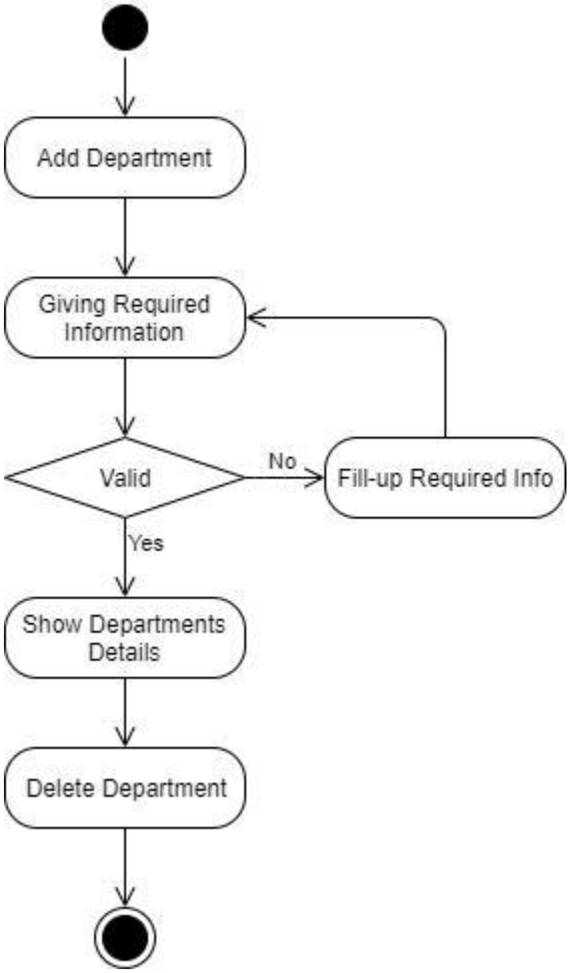


Figure 3.6: Activity diagram for Manage Department.

3.2.6 Manage Transaction

Admin fully control and monitor transaction and transaction history.

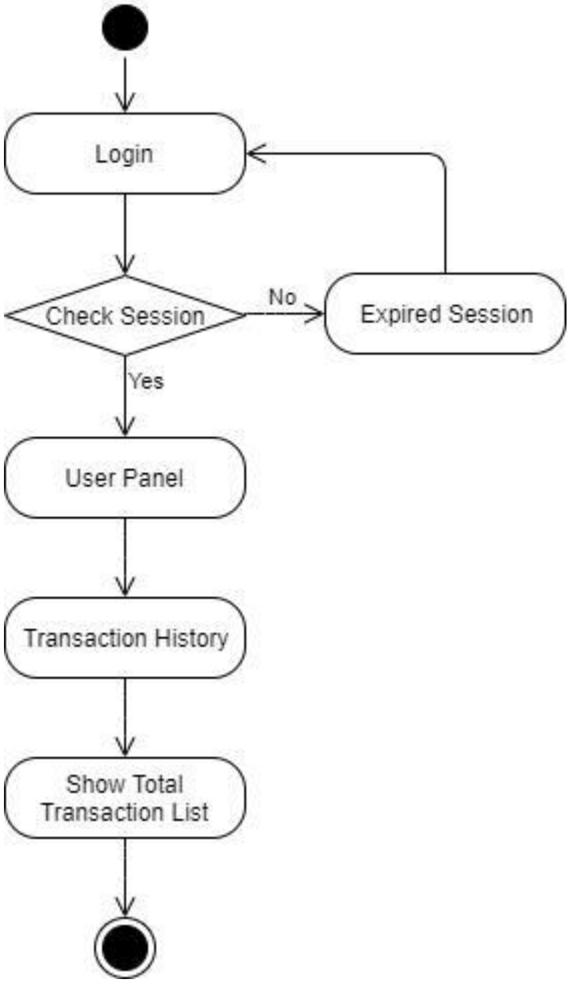


Figure 3.7: Activity diagram for Manage Transaction.

3.2.7 Manage Appointment

Patient or coordinator sent an appointment request to a doctor then doctor accepts the appointment request. After accepting the appointment request patient or coordinator submit required information.

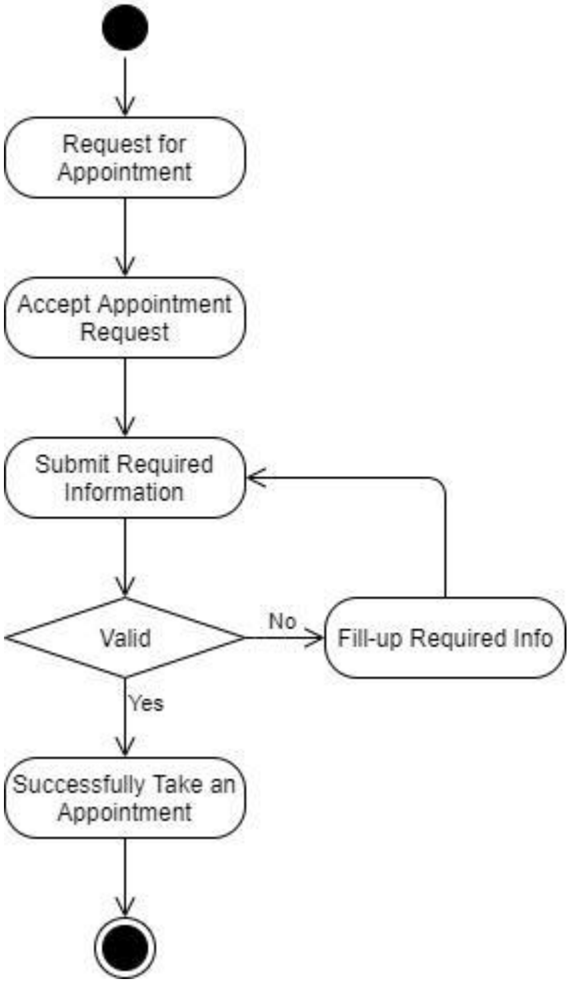

Request for Appointment
Accept Appointment Request
Submit Required Information
Valid
No
Fill-up Required Info
Yes
Successfully Take an Appointment

Figure 3.8: Activity diagram for Manage Appointment.

3.2.8 Manage Prescription

Doctor can give treatment for specific disease or refer to another doctor.

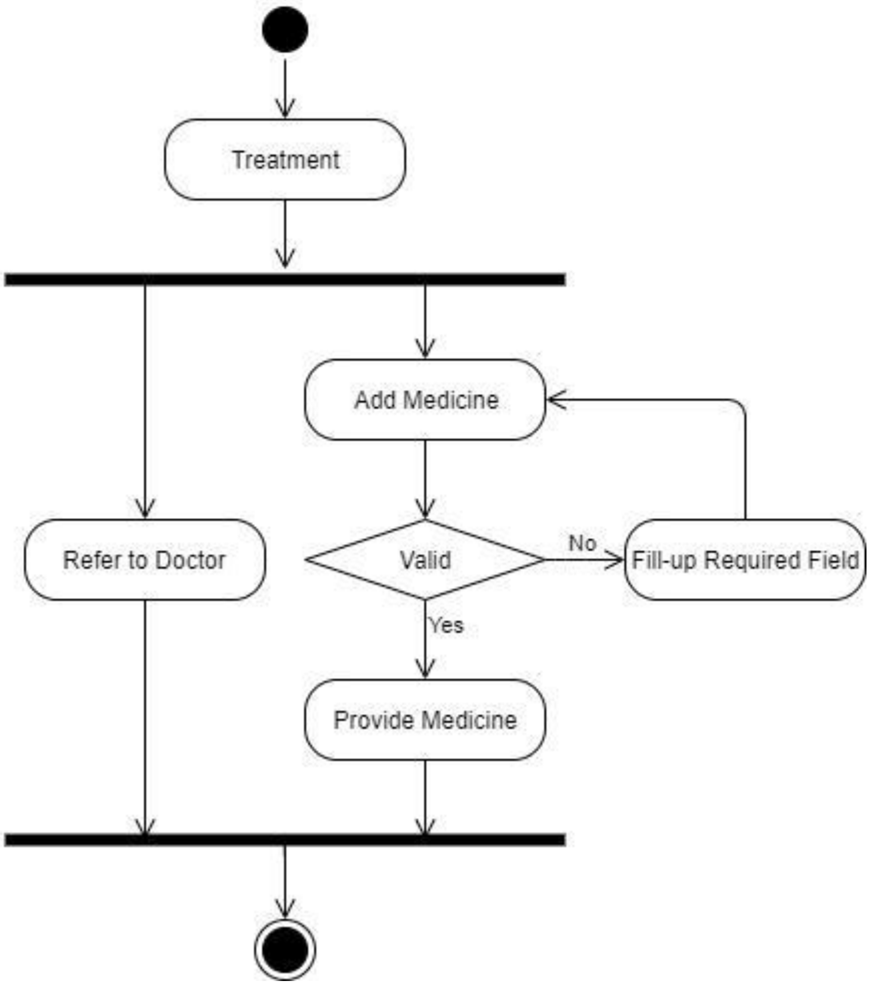


Figure 3.9: Activity diagram for Manage Prescription.

3.2.9 Manage Report

Doctor give medical test to patient for identifying disease, patient submit this test report as picture to the doctor.

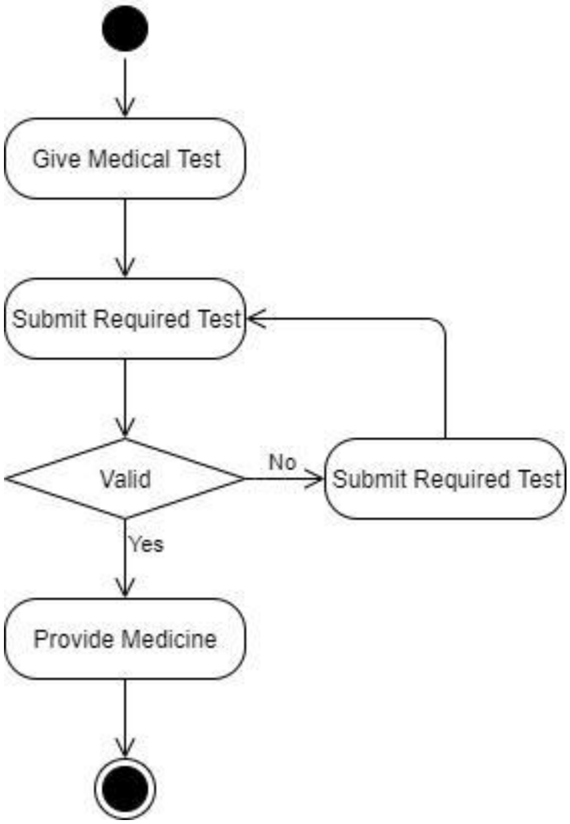


Figure 3.10: Activity diagram for Manage Report.

3.2.10 Live Conference

After taking an appointment doctor and patient communicate via text chat or video conferencing.

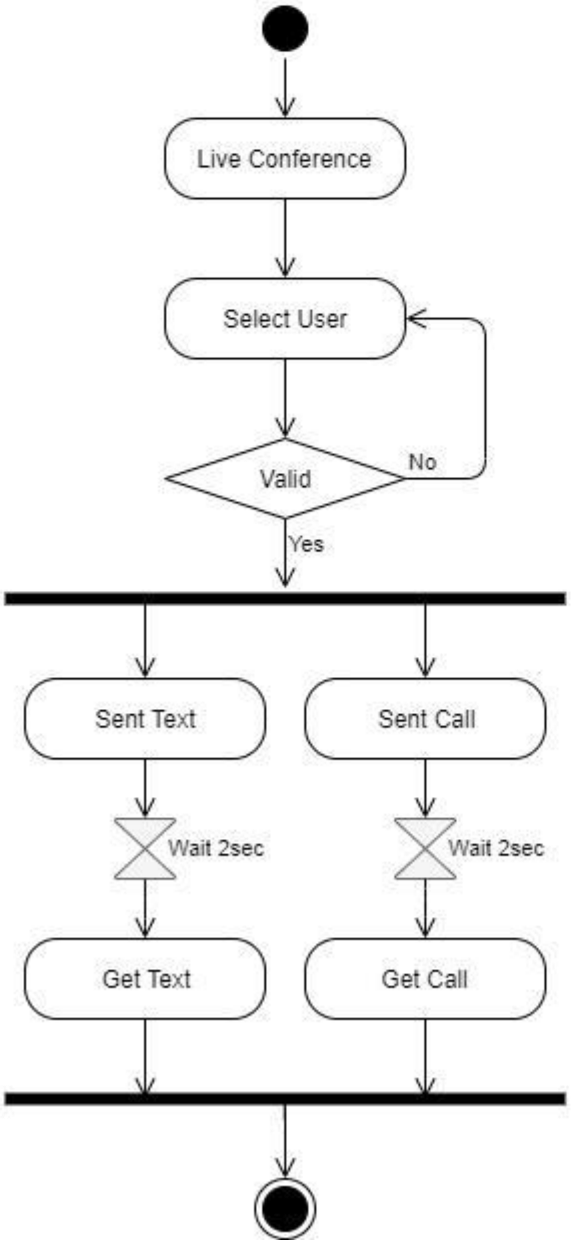


Figure 3.11: Activity diagram for Live Conference.

3.3 Sequence Diagram

We are preparing sequence diagram according to activity diagram. Those sequence diagrams represent object interactions arranged in time sequence.

3.3.1 User Login

Users login to the user panel through email and password that are prepared after verifying the account.

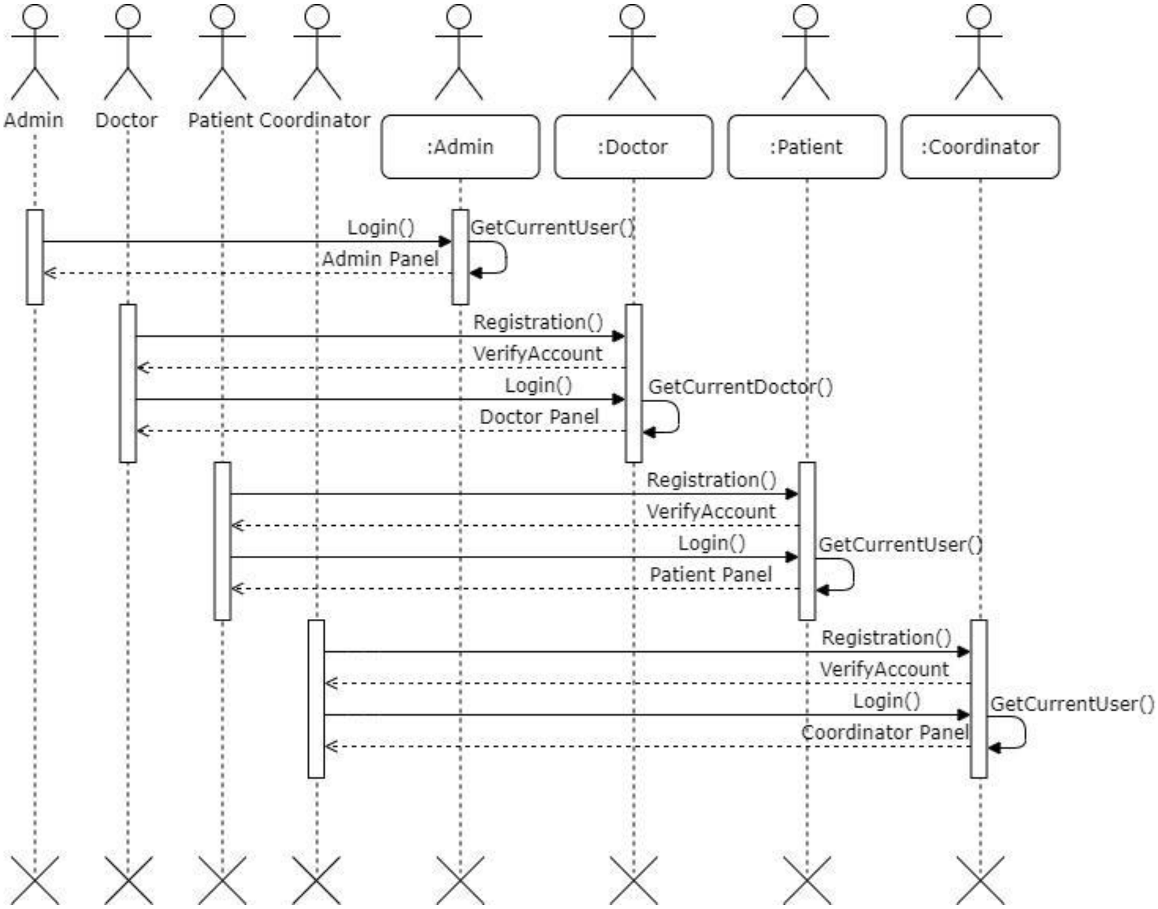


Figure 3.12: Sequence diagram for User Login.

3.3.2 Manage Doctor

Admin fully control and monitor doctors profile for verification purpose.

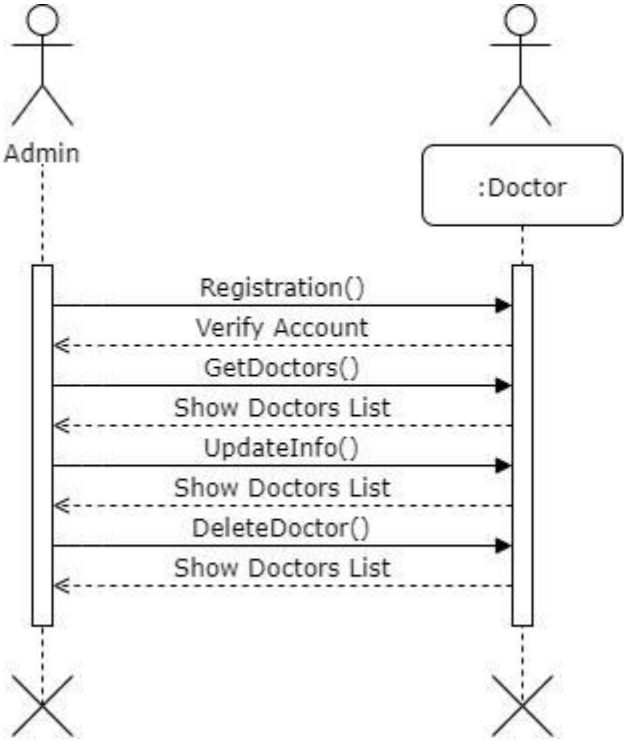


Figure 3.13: Sequence diagram for Manage Doctor.

3.3.3 Manage Coordinator

Admin fully control and monitor coordinator profile for verification purpose.

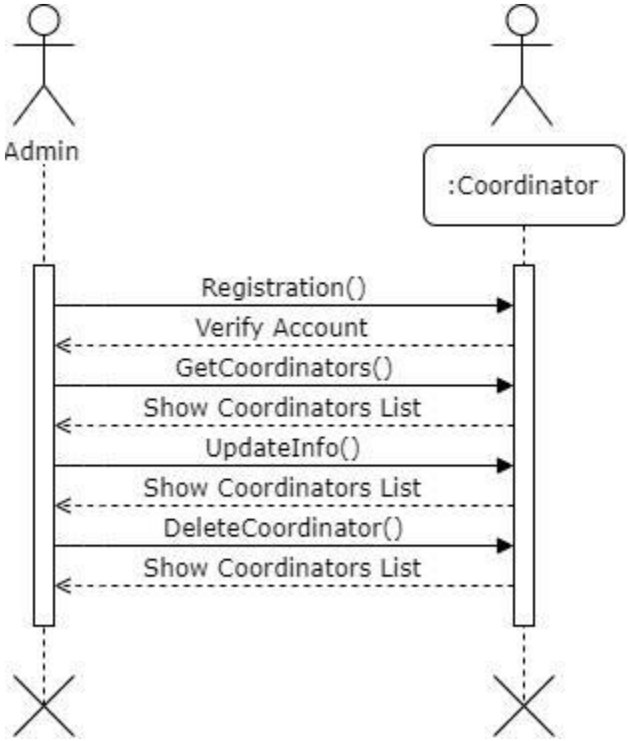


Figure 3.14: Sequence diagram for Manage Coordinator.

3.3.4 Manage Patient

Admin fully control and monitor patient profile for verification purpose.

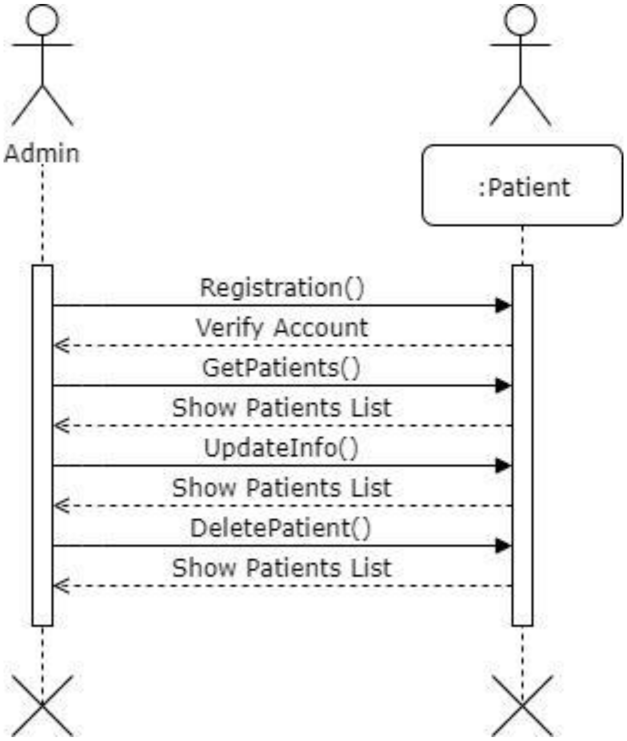


Figure 3.15: Sequence diagram for Manage Patient.

3.3.5 Manage Department

Admin fully control department list, added new department or delete department.

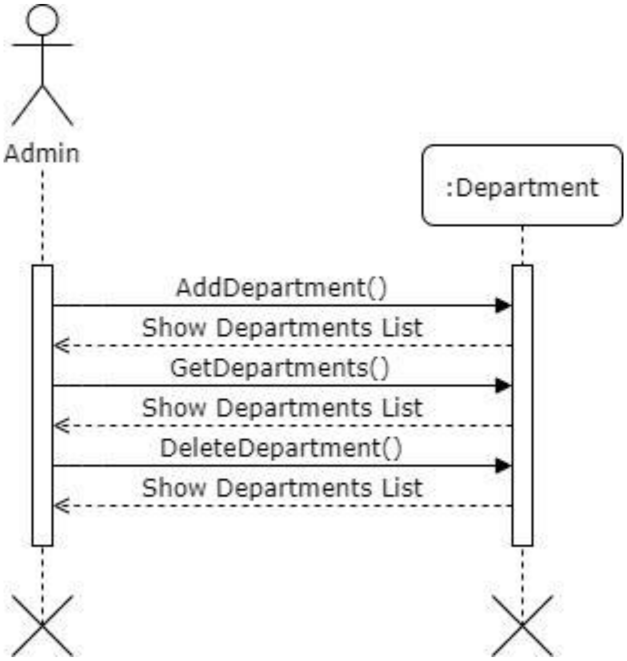


Figure 3.16: Sequence diagram for Manage Department.

3.3.6 Manage Appointment

Patient or coordinator sent an appointment request to a doctor then doctor accepts the appointment request. After accepting the appointment request patient or coordinator submit required information.

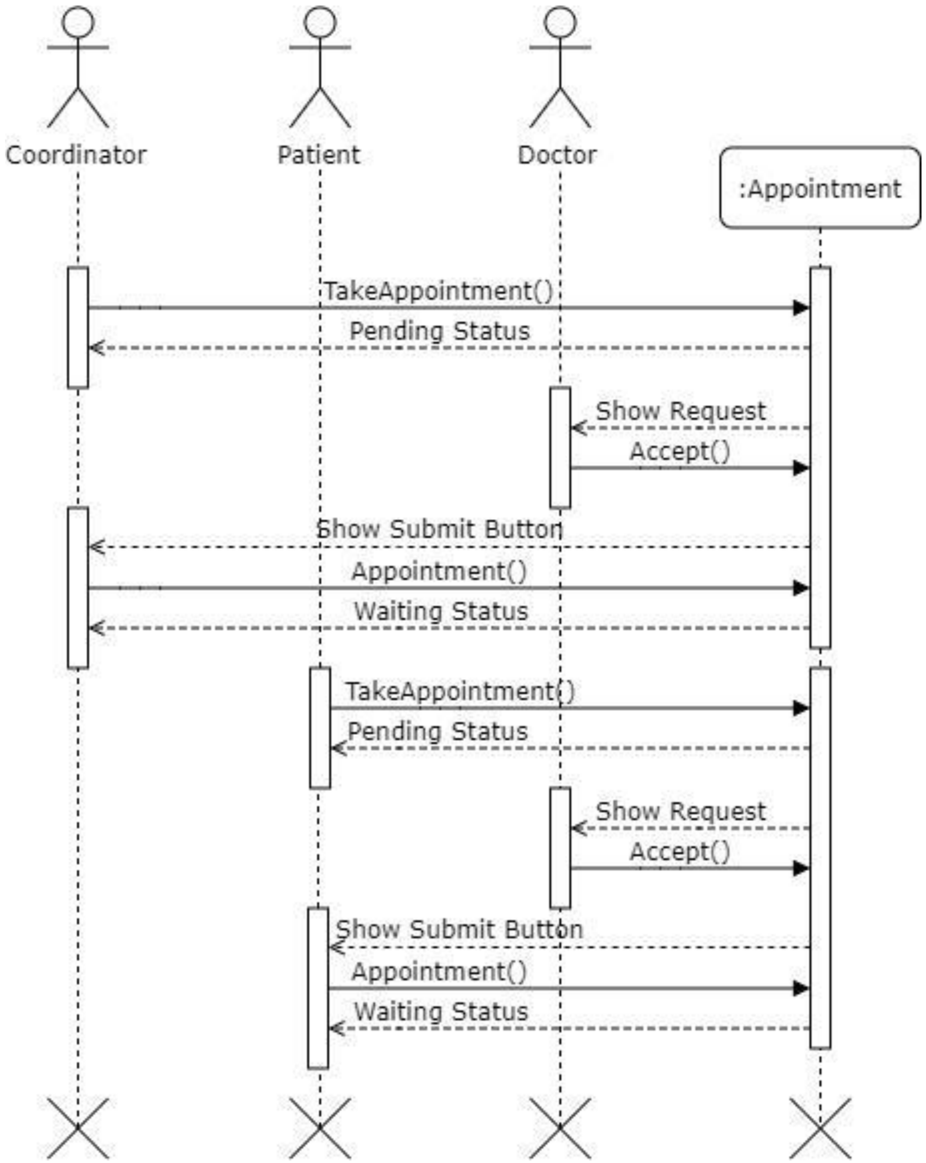


Figure 3.17: Sequence diagram for Manage Appointment.

3.3.7 Manage Prescription

Doctor can give treatment for specific disease or refer to another doctor.

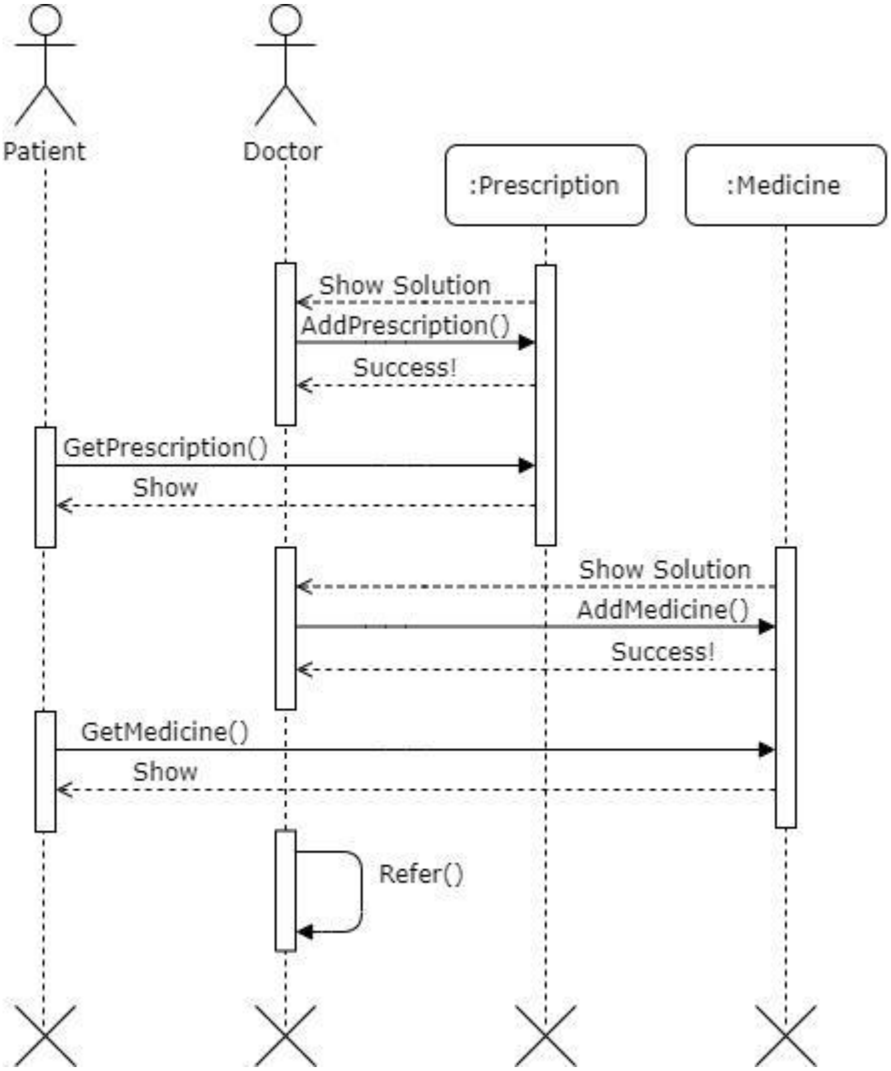


Figure 3.18: Sequence diagram for Manage Prescription.

3.3.8 Manage Report

Doctor give medical test to patient for identifying disease, patient submit this test report as picture to the doctor.

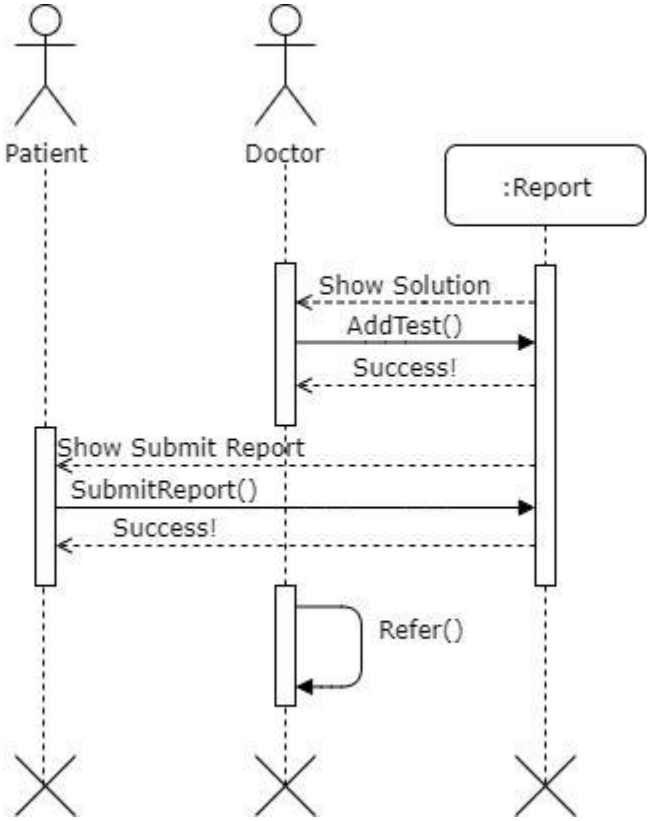


Figure 3.19: Sequence diagram for Manage Report.

3.3.9 Live Conference

After taking an appointment doctor and patient communicate via text chat or video conferencing.

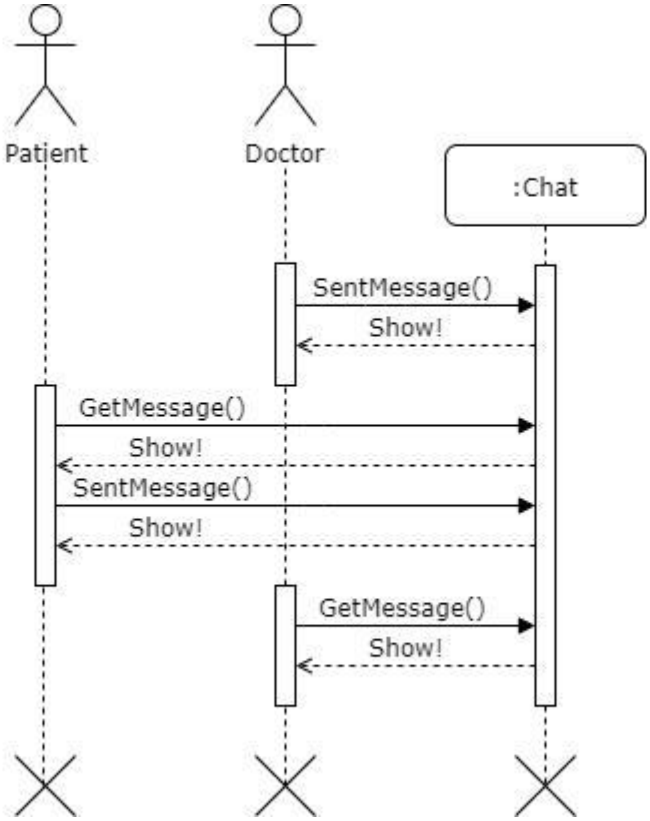


Figure 3.20: Sequence diagram for Live Conference.

Chapter 4

System Design Specification

4.1 Development Tools and Technology

Tools are essential for software development. Day by day many kinds of technology release for developed software easy and smoothly. Using this technology developer can developed software efficiently.

We have used some of those tools and technologies to implement our project.

4.1.1 User Interface Technology

User interface represents functionality of a system. The main point of user interface in a system is how much easy is to use and how simple it is.

4.1.1.1 Front-end

We have used HTML Helpers, CSS, Bootstrap, JavaScript for front-end development.

4.1.1.2 Programming Language

Programming language is the most important part to make a software functional.

We have used C# programming language for back-end site.

4.1.2 Implemented Tools and Platform

We have used .Net MVC Framework and Node.js for back-end and Entity Framework (Code First) for manage and design database. For testing purpose we have used Unit Testing.

4.1.2.1 IDE

Integrated Development Environment (IDE) is essential for software development. That amalgamates basic tools required to write and test software.

We have used Visual Studio 2019 Community and Visual Studio Code for developing our project. It is a free version for students powered by Microsoft.

4.1.2.2 Web Server

A web server is a server software that delivers content or service to end users over the internet.

We have used IIS Server and js Server. We have used Visual Studio for developing our project so IIS Server and js Server is free for local host.

4.1.2.3 Database Server

In our project we have used SQL Server 2017 database. It is free version for students powered by Microsoft.

4.2 Class Diagram

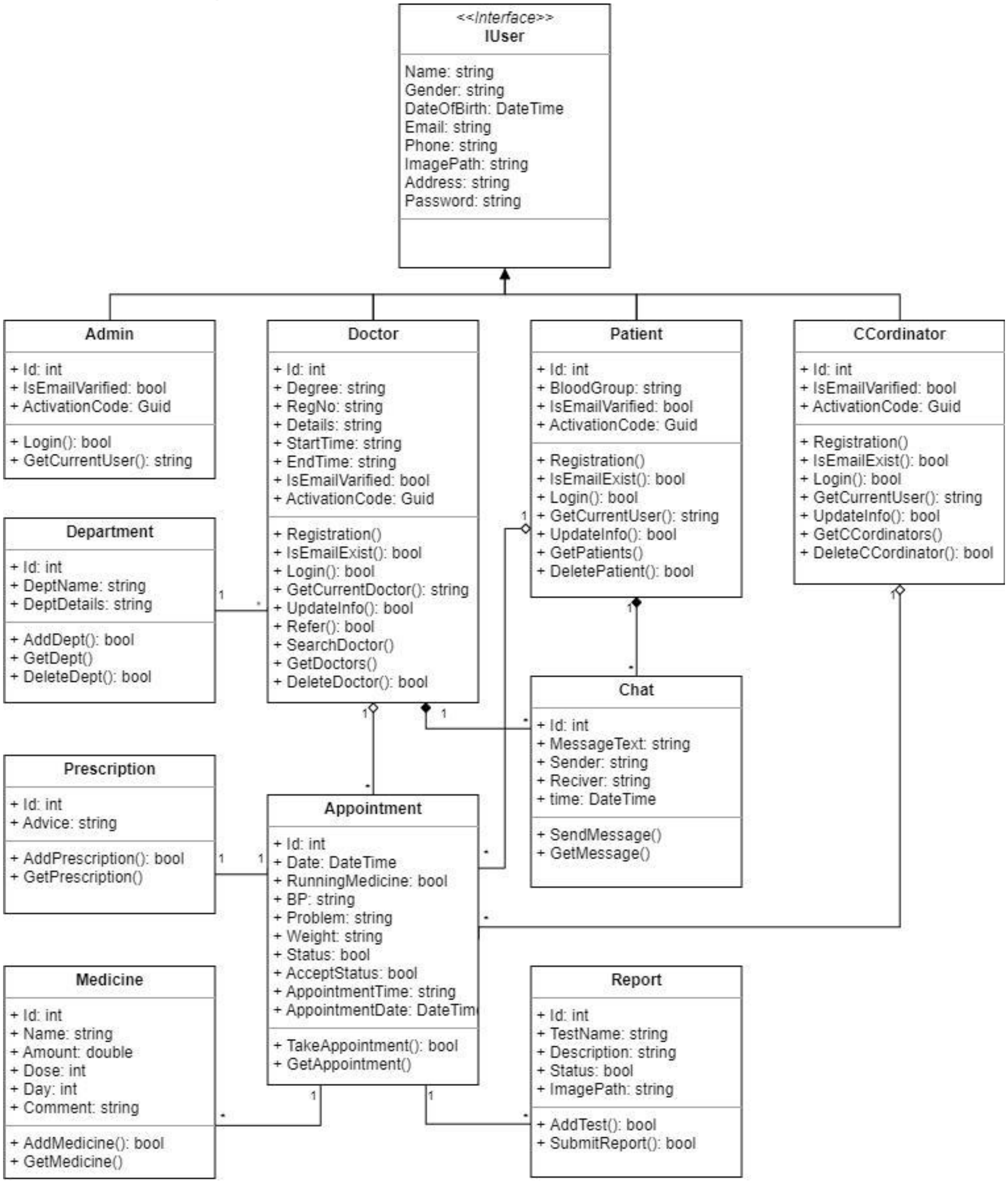


Figure 4.1: Class diagram.

4.3 Entity Relationship Diagram (ERD)

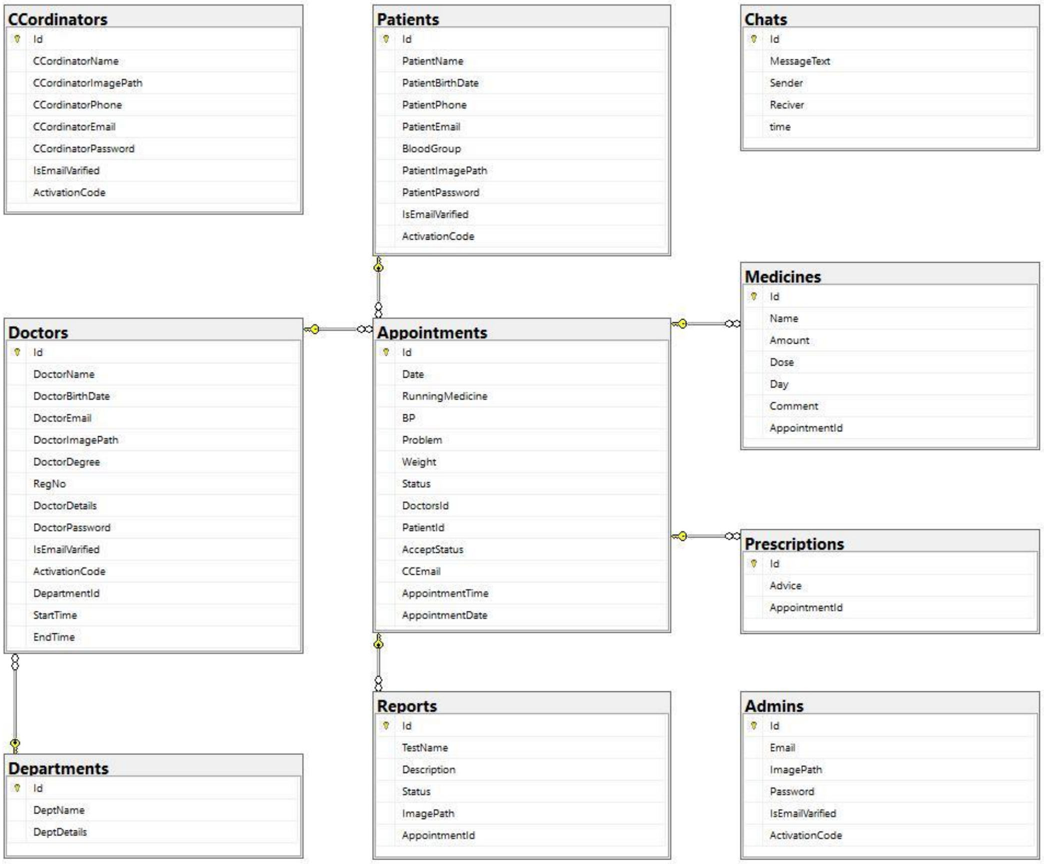


Figure 4.2: ER diagram.

Chapter 5

System Test

5.1 Testing Features

Feature Testing means the procedure of making changes in application system to add many new features or to make interchange in the already existing features. Almost all feature and function different type and every functions have an objective. Using feature tests find out how much useful, reliable, intuitive, secured, effective and scalable an application.

5.1.1 Feature to be tested

Admin Part

Features	Priority	Description
Login	3	Admin must be authenticated by login
Logout	2	Cookies must be destroyed after logout
Add Department	3	Department data inserted properly
View Departments	2	Show departments list and information
Delete Department	3	Department remove properly
Doctors Registration	3	Doctor information inserted properly
View Doctors List	2	Show doctors list and information
Update Doctor Info	1	Doctor information edited properly
Delete Doctor Profile	2	Doctor profile remove properly
Patients Registration	3	Patient information inserted properly
View Patients List	2	Show patient list and information
Update Patient Info	1	Patient information edited properly
Delete Patient Profile	2	Patient profile remove properly
Coordinators Registration	3	Coordinator information inserted properly
View Coordinators List	2	Show coordinator list and information
Update Coordinator Info	1	Coordinator information edited properly
Delete Coordinators Profile	2	Coordinator profile remove properly

Doctor Part

Features	Priority	Description
Registration	3	Doctor information inserted properly
Login	3	Doctor must be authenticated by login
Logout	2	Cookies must be destroyed after logout
View Profile	1	Show doctor information
Update Profile	1	Doctor information edited properly
Patient History	2	Show previous patients treatment history
Accept Appointment	3	Show accept list and data inserted properly
Prescribe	3	Prescription data inserted properly
Live Conference	3	Message data inserted properly and show

Patient Part

Features	Priority	Description
Registration	3	Patient information inserted properly
Login	3	Patient must be authenticated by login
Logout	2	Cookies must be destroyed after logout
View Profile	1	Show patient information
Update Profile	1	Patient information edited properly
Take Appointment	3	Take appointment data inserted properly
Submit Report	3	Report data inserted properly
View Prescription	3	Show prescription data
Live Conference	3	Message data inserted properly and show

Coordinator Part

Features	Priority	Description
Registration	3	Coordinator information inserted properly
Login	3	Coordinator must be authenticated by login
Logout	2	Cookies must be destroyed after logout
View Profile	1	Show coordinator information
Update Profile	1	Coordinator information edited properly
Take Appointment	3	Take appointment data inserted properly

Here,

3	High Priority
2	Medium Priority
1	Low Priority

5.2 Testing Strategy

Test Strategy is an outline that explains the testing approach and determines how testing needs to be done.

5.2.1 Test approach

Test approach is an implementation process of test strategy. Determines how to be tested an application. There are two technique of testing approach:

Preventive: In the case of preventive method, tests are designed prior to software development.

Reactive: In the case of reactive method, tests are designed after design and coding are completed.

5.2.1.1 Black Box Testing

Black box testing strategy is the behavior testing of a software. In black box testing, tester test the software without internal knowledge and implementation knowledge.

In our project we have test black box testing after fully completed the project. We have tested our project following the requirement specification.

5.2.1.2 White Box Testing

White box testing strategy is the logical testing of a software. In white box testing, tester has knowledge about internal knowledge and implementations of the software. White box testing can be categorized at some levels:

Unit Testing

Integration Testing

System Testing

In our project we used unit testing during development time. Using unit testing we checked out every method in our project work properly or not.

5.2.2 Pass or Fail Criteria

The software tester will determine the pass and fail criteria. They will prepare pass or fail criteria based on which feature perfectly work and which feature not work perfectly. Now I will give the pass or fail criteria below:

If system crash, will not be considered as pass case.

If any feature pass 100% times, then it will be considered as pass criteria only.

If data can’t be displayed to the application properly, then it is also to be considered as fail criteria.

5.3 Testing Schedule

Test Phase	**Time**
Testing plan create	1 week
Test specification	2 week
Unit testing	During development time
Component test	1 week
Validating use cases	1 week
Testing user interfaces	1 week
Load testing	1 week
Performance testing	1 week
Release to production	1 week

5.4 Trace Ability Matrix

Admin Part

Project Manager			**Business Analyst Lead**		
QA Lead			**Target Implementation Date**		
BR#	**Category/ Functionality/Activity**	**Requirement Description**	**Use Case Reference**	**Test Case Reference**	**Comments**
BR-1	Functional	Login	Use-case 3.1.1	Test-case 5.6.1	
BR-2	Functional	Add Department	Use-case	Test-case	

			3.1.5	5.6.2	
BR-3	Functional	View Departments	Use-case 3.1.5	Test-case 5.6.3	
BR-4	Functional	Delete Department	Use-case 3.1.5	Test-case 5.6.4	
BR-5	Functional	Doctors Registration	Use-case 3.1.2	Test-case 5.6.5	
BR-6	Functional	View Doctors List	Use-case 3.1.2	Test-case 5.6.6	
BR-7	Functional	Update Doctor Info	Use-case 3.1.2	Test-case 5.6.7	
BR-8	Functional	Delete Doctor Profile	Use-case 3.1.2	Test-case 5.6.8	
BR-9	Functional	Patients Registration	Use-case 3.1.4	Test-case 5.6.9	
BR-10	Functional	View Patients List	Use-case 3.1.4	Test-case 5.6.10	
BR-11	Functional	Update Patient Info	Use-case 3.1.4	Test-case 5.6.11	
BR-12	Functional	Delete Patient Profile	Use-case 3.1.4	Test-case 5.6.12	
BR-13	Functional	Coordinators Registration	Use-case 3.1.3	Test-case 5.6.13	
BR-14	Functional	View Coordinators List	Use-case 3.1.3	Test-case 5.6.14	
BR-15	Functional	Update Coordinator Info	Use-case 3.1.3	Test-case 5.6.15	
BR-16	Functional	Delete Coordinators Profile	Use-case 3.1.3	Test-case 5.6.16	

Doctor Part

Project Manager			**Business Analyst Lead**		
QA Lead			**Target Implementation Date**		
BR#	**Category/ Functionality/Activity**	**Requirement Description**	**Use Case Reference**	**Test Case Reference**	**Comments**
BR-1	Functional	Registration	Use-case 3.1.2	Test-case 5.6.17	
BR-2	Functional	Login	Use-case 3.1.1	Test-case 5.6.18	
BR-3	Functional	View Profile	Use-case 3.1.2	Test-case 5.6.19	
BR-4	Functional	Update Profile	Use-case	Test-case	

			3.1.2	5.6.20	
BR-5	Functional	Patient History	Use-case 3.1.8	Test-case 5.6.21	
BR-6	Functional	Accept Appointment	Use-case 3.1.7	Test-case 5.6.22	
BR-7	Functional	Prescribe	Use-case 3.1.8	Test-case 5.6.23	
BR-8	Functional	Live Conference	Use-case 3.1.10	Test-case 5.6.24	

Patient Part

Project Manager			**Business Analyst Lead**		
QA Lead			**Target Implementation Date**		
BR#	**Category/ Functionality/Activity**	**Requirement Description**	**Use Case Reference**	**Test Case Reference**	**Comments**
BR-1	Functional	Registration	Use-case 3.1.4	Test-case 5.6.25	
BR-2	Functional	Login	Use-case 3.1.1	Test-case 5.6.26	
BR-3	Functional	View Profile	Use-case 3.1.4	Test-case 5.6.27	
BR-4	Functional	Update Profile	Use-case 3.1.4	Test-case 5.6.28	
BR-5	Functional	Take Appointment	Use-case 3.1.7	Test-case 5.6.29	
BR-6	Functional	Submit Report	Use-case 3.1.9	Test-case 5.6.30	
BR-7	Functional	View Prescription	Use-case 3.1.8	Test-case 5.6.31	
BR-8	Functional	Live Conference	Use-case 3.1.10	Test-case 5.6.32	

Coordinator Part

Project Manager			**Business Analyst Lead**		
QA Lead			**Target Implementation Date**		
BR#	**Category/ Functionality/Activity**	**Requirement Description**	**Use Case Reference**	**Test Case Reference**	**Comments**
BR-1	Functional	Registration	Use-case 3.1.3	Test-case 5.6.33	

BR-2	Functional	Login	Use-case 3.1.1	Test-case 5.6.34	
BR-3	Functional	View Profile	Use-case 3.1.3	Test-case 5.6.35	
BR-4	Functional	Update Profile	Use-case 3.1.3	Test-case 5.6.36	
BR-5	Functional	Take Appointment	Use-case 3.1.7	Test-case 5.6.37	

5.5 Testing Environment

Testing environment is the environment that support to execution the test with software, hardware and network configures.

For set up the environment of testing, some key area need. Those are:

Test Data

Database server

Client’s operating system

Front end running

environment Brower

Network

System and application

Documentation is also required. Like: user manual, installation guideline etc.

People are involve with test environment setup. Those are:

System Admins

Developers

Testers

5.6 Test Cases

Test case means set some rules and regulations or conditions by which it can be determined whether a system can be able to meet the works or requirements under test cases correctly.

5.6.1 Login for Admin

<table>
<tr><td colspan="3"><b>Test case</b> #1</td><td colspan="3"><b>Test case name</b>: Login</td></tr>
<tr><td colspan="3"><b>System:</b> Remote Doctor</td><td colspan="3"><b>Subsystem:</b></td></tr>
<tr><td colspan="3"><b>Designed by:</b> Raihan Mahmud</td><td colspan="3"><b>Designed date:</b> 15-Nov-2019</td></tr>
<tr><td colspan="3"><b>Executed by:</b> Raihan Mahmud</td><td colspan="3"><b>Executed date:</b> 15-Nov-2019</td></tr>
<tr><td colspan="6"><b>Short description:</b> Admin needs to login for access to the admin panel and before login system check authentication.</td></tr>
<tr><td colspan="6"><b>Pre-conditions:</b>
User type must be admin
Assume that, the email is raihan35@gmail.com and password “1234567”</td></tr>
<tr><td><b>Step</b></td><td><b>Email</b></td><td><b>Password</b></td><td><b>Response</b></td><td><b>Pass/Fail</b></td><td><b>Comment</b></td></tr>
</table>

1	raihan@gmail.com	1765432	Wrong email or password	Fail	
2	raihan@gmail.com		Wrong email or password	Fail	
3		1234567	Email required	Fail	
4	raihan35@gmail.com	1234567	Show admin panel	Pass	

Post-conditions: Admin will successfully login to the admin panel

5.6.2 Add Department

Test case #2	**Test case name**: Add Department
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 15-Nov-2019
Executed by: Raihan Mahmud	**Executed date:** 15-Nov-2019

Short description: Admin can added new department

Pre-conditions:

User type must be admin
Admin must be authorized
Admin must be login
Admin must have information about the new department

Step	Action	Response	Pass/Fail	Comment
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Show dept. list	Pass	

Post-conditions: Admin added new department information successfully

5.6.3 View Departments

Test case #3	**Test case name**: View Departments
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 15-Nov-2019
Executed by: Raihan Mahmud	**Executed date:** 15-Nov-2019

Short description: Admin can view departments list

Pre-conditions:

User type must be admin
Admin must be authorized
Admin must be login
There must be some departments

Step	Action	Response	Pass/Fail	Comment
1	Click “Departments” option	Empty list show	Fail	This is an initial stage. So here, no

				department was recorded.
2	Click “Departments” option	Show dept. list	Pass	
Post-conditions: Admin able to see list of department				

5.6.4 Delete Department

Test case #4		**Test case name**: Delete Department		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 15-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 15-Nov-2019		
Short description: Admin can delete department				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login There must be some departments				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Go to departments and hit the “Delete” button.	Show “Are you wants to delete?”	Pass	Click Yes
Post-conditions: Admin has successfully deleted the department				

5.6.5 Doctors Registration

Test case #5		**Test case name**: Doctors Registration		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 16-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 16-Nov-2019		
Short description: Admin can register new doctor				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login Admin must have information about the new doctor				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Check email	Pass	
Post-conditions: Admin has successfully registered a new doctor				

5.6.6 View Doctors List

Test case #6		**Test case name**: View Doctors List		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 16-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 16-Nov-2019		
Short description: Admin can view doctors list				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login There must be some registered doctor				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click "Doctors" option	Empty list show	Fail	This is an initial stage. So here, no doctor was registered.
2	Click "Doctors" option	Show list	Pass	
Post-conditions: Admin able to see list of doctors				

5.6.7 Update Doctor Info

Test case #7		**Test case name**: Update Doctor Info		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 16-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 16-Nov-2019		
Short description: Admin can updated doctor information				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login Admin must have updated information about the doctor				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Show doctors list	Pass	
Post-conditions: Admin successfully updated doctor information				

5.6.8 Delete Doctor Profile

Test case #8	**Test case name**: Delete Doctor Profile
System: Remote Doctor	**Subsystem:**

Designed by: Raihan Mahmud		**Designed date:** 16-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 16-Nov-2019		
Short description: Admin can delete doctor profile				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login There must be some registered doctor				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Go to doctors and hit the "Delete" button.	Show "Are you wants to delete?"	Pass	Click Yes
Post-conditions: Admin has successfully deleted the doctor profile				

5.6.9 Patients Registration

Test case #9		**Test case name**: Patients Registration		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 17-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 17-Nov-2019		
Short description: Admin can register new patient				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login Admin must have information about the new patient				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Check email	Pass	
Post-conditions: Admin has successfully registered a new patient				

5.6.10 View Patients List

Test case #10	**Test case name**: View Patients List
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 17-Nov-2019
Executed by: Raihan Mahmud	**Executed date:** 17-Nov-2019
Short description: Admin can view patients list	
Pre-conditions: User type must be admin Admin must be authorized	

Admin must be login There must be some registered patient				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Patients” option	Empty list show	Fail	This is an initial stage. So here, no patient was registered.
2	Click “Patients” option	Show list	Pass	
Post-conditions: Admin able to see list of patients				

5.6.11 Update Patient Info

Test case #11		**Test case name**: Update Patient Info		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 17-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 17-Nov-2019		
Short description: Admin can updated patient information				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login Admin must have updated information about the patient				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Show patients list	Pass	
Post-conditions: Admin successfully updated patient information				

5.6.12 Delete Patient Profile

Test case #12		**Test case name**: Delete Patient Profile		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 17-Nov-2019		
Executed by: Raihan Mahmud		**Executed date:** 17-Nov-2019		
Short description: Admin can delete patient profile				
Pre-conditions: User type must be admin Admin must be authorized Admin must be login There must be some registered patient				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**

1	Go to patients and hit the “Delete” button.	Show “Are you wants to delete?”	Pass	Click Yes

Post-conditions: Admin has successfully deleted the patient profile

5.6.13 Coordinators Registration

Test case #13	**Test case name**: Coordinators Registration
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 18-Nov-2019
Executed by: Raihan Mahmud	**Executed date:** 18-Nov-2019

Short description: Admin can register new coordinator

Pre-conditions:

User type must be admin

Admin must be authorized

Admin must be login

Admin must have information about the new coordinator

Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Check email	Pass	

Post-conditions: Admin has successfully registered a new coordinator

5.6.14 View Coordinators List

Test case #14	**Test case name**: View Coordinators List
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 18-Nov-2019
Executed by: Raihan Mahmud	**Executed date:** 18-Nov-2019

Short description: Admin can view coordinator list

Pre-conditions:

User type must be admin

Admin must be authorized

Admin must be login

There must be some registered coordinator

Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Coordinators” option	Empty list show	Fail	This is an initial stage. So here, no coordinator was registered.
2	Click “Coordinators” option	Show list	Pass	

<table>
<tr><td><b>Post-conditions:</b> Admin able to see list of coordinators</td></tr>
</table>

5.6.15 Update Coordinator Info

<table>
<tr><td colspan="2"><b>Test case</b> #15</td><td colspan="3"><b>Test case name</b>: Update Coordinator Info</td></tr>
<tr><td colspan="2"><b>System:</b> Remote Doctor</td><td colspan="3"><b>Subsystem:</b></td></tr>
<tr><td colspan="2"><b>Designed by:</b> Raihan Mahmud</td><td colspan="3"><b>Designed date:</b> 18-Nov-2019</td></tr>
<tr><td colspan="2"><b>Executed by:</b> Raihan Mahmud</td><td colspan="3"><b>Executed date:</b> 18-Nov-2019</td></tr>
<tr><td colspan="5"><b>Short description:</b> Admin can updated coordinator information</td></tr>
<tr><td colspan="5"><b>Pre-conditions:</b>
User type must be admin
Admin must be authorized
Admin must be login
Admin must have updated information about the coordinator</td></tr>
<tr><td><b>Step</b></td><td><b>Action</b></td><td><b>Response</b></td><td><b>Pass/Fail</b></td><td><b>Comment</b></td></tr>
<tr><td>1</td><td>All required field are not filled yet</td><td>Show required message</td><td>Fail</td><td></td></tr>
<tr><td>2</td><td>All input field is filled up</td><td>Show coordinators list</td><td>Pass</td><td></td></tr>
<tr><td colspan="5"><b>Post-conditions:</b> Admin successfully updated coordinator information</td></tr>
</table>

5.6.16 Delete Coordinator Profile

<table>
<tr><td colspan="2"><b>Test case</b> #16</td><td colspan="3"><b>Test case name</b>: Delete Coordinator Profile</td></tr>
<tr><td colspan="2"><b>System:</b> Remote Doctor</td><td colspan="3"><b>Subsystem:</b></td></tr>
<tr><td colspan="2"><b>Designed by:</b> Raihan Mahmud</td><td colspan="3"><b>Designed date:</b> 18-Nov-2019</td></tr>
<tr><td colspan="2"><b>Executed by:</b> Raihan Mahmud</td><td colspan="3"><b>Executed date:</b> 18-Nov-2019</td></tr>
<tr><td colspan="5"><b>Short description:</b> Admin can delete coordinator profile</td></tr>
<tr><td colspan="5"><b>Pre-conditions:</b>
User type must be admin
Admin must be authorized
Admin must be login
There must be some registered coordinator</td></tr>
<tr><td><b>Step</b></td><td><b>Action</b></td><td><b>Response</b></td><td><b>Pass/Fail</b></td><td><b>Comment</b></td></tr>
<tr><td>1</td><td>Go to coordinators and hit the “Delete” button.</td><td>Show “Are you wants to delete?”</td><td>Pass</td><td>Click Yes</td></tr>
<tr><td colspan="5"><b>Post-conditions:</b> Admin has successfully deleted the coordinator profile</td></tr>
</table>

5.6.17 Registration for Doctor

Test case #17		**Test case name**: Registration		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 01-Sep-2019		
Executed by: Raihan Mahmud		**Executed date:** 19-Nov-2019		
Short description: Doctor can create her account				
Pre-conditions: User type must be doctor				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Check email	Pass	
Post-conditions: Doctor has successfully created her account				

5.6.18 Login for Doctor

Test case #18			**Test case name**: Login		
System: Remote Doctor			**Subsystem:**		
Designed by: Raihan Mahmud			**Designed date:** 02-Sep-2019		
Executed by: Raihan Mahmud			**Executed date:** 19-Nov-2019		
Short description: Doctor needs to login for access to the doctor panel and before login system check authentication.					
Pre-conditions: User type must be doctor Assume that, the email is raihan35@gmail.com and password “1234567”					
Step	**Email**	**Password**	**Response**	**Pass/Fail**	**Comment**
1	raihan@gmail.com	1765432	Wrong email or password	Fail	
2	raihan@gmail.com		Wrong email or password	Fail	
3		1234567	Email required	Fail	
4	raihan35@gmail.com	1234567	Show admin panel	Pass	
Post-conditions: Doctor will successfully login to the doctor panel					

5.6.19 View Profile for Doctor

Test case #19	**Test case name**: View Profile
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 03-Sep-2019
Executed by: Raihan Mahmud	**Executed date:** 19-Nov-2019
Short description: Doctor can view her profile	

Pre-conditions: User type must be doctor Doctor must be authorized Doctor must be login				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Profile” option	Show profile	Pass	
Post-conditions: Doctor able to see her profile				

5.6.20 Update Profile for Doctor

Test case #20		**Test case name**: Update Profile		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 04-Sep-2019		
Executed by: Raihan Mahmud		**Executed date:** 19-Nov-2019		
Short description: Doctor can updated her information				
Pre-conditions: User type must be doctor Doctor must be authorized Doctor must be login				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Show profile	Pass	
Post-conditions: Doctor successfully updated her information				

5.6.21 Patient History

Test case #21		**Test case name**: Patient History		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 05-Sep-2019		
Executed by: Raihan Mahmud		**Executed date:** 19-Nov-2019		
Short description: Doctor can view previous patient history				
Pre-conditions: User type must be doctor Doctor must be authorized Doctor must be login There must be previous prescribed patient				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Patients” option	Empty list show	Fail	This is an initial stage. So here, no

				patient was recorded.
2	Click “Patients” option	Show list	Pass	
Post-conditions: Doctor able to see patients history				

5.6.22 Accept Appointment Request

Test case #22	**Test case name**: Accept Appointment
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 06-Sep-2019
Executed by: Raihan Mahmud	**Executed date:** 19-Nov-2019
Short description: Doctor can accept appointment request and set appointment schedule.	
Pre-conditions: User type must be doctor Doctor must be authorized Doctor must be login Must be patient sent appointment request	

Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Appointment” option	There are no appointment request show	Fail	This is an initial stage. So here, no appointment was recorded.
2	Click “Appointment” option, accept appointment request and set schedule.	Show list	Pass	
Post-conditions: Doctor successfully accepted appointment request				

5.6.23 Prescribe

Test case #23	**Test case name**: Prescribe
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 07-Sep-2019
Executed by: Raihan Mahmud	**Executed date:** 19-Nov-2019
Short description: Doctor can prescribe to patient	
Pre-conditions: User type must be doctor Doctor must be authorized Doctor must be login Must be patient submit required information	

Step	**Action**	**Response**	**Pass/Fail**	**Comment**

1	Click “Appointment” option	There are no treatment request show	Fail	This is an initial stage. So here, no appointment was recorded.
2	Click “Appointment” option, click “Solution” button and prescribe or give test to the patient. Or refer to another doctor.	Show list	Pass	

Post-conditions: Doctor has successfully prescribed to patient

5.6.24 Live Conference for Doctor

Test case #24	**Test case name**: Live Conference
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 08-Sep-2019
Executed by: Raihan Mahmud	**Executed date:** 19-Nov-2019
Short description: Doctor can live conference with patient	
Pre-conditions: User type must be doctor Doctor must be authorized Doctor must be login Must be patient appointment with doctor	

Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Chat” option	There are no patient show	Fail	This is an initial stage. So here, no patient was recorded.
2	Click “Chat” option, choose patient and connected with patient.	Show list	Pass	
Post-conditions: Doctor successfully conference with patient				

5.6.25 Registration for Patient

Test case #25	**Test case name**: Registration
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 01-Oct-2019
Executed by: Raihan Mahmud	**Executed date:** 20-Nov-2019
Short description: Patient can create her account	
Pre-conditions: User type must be patient	

Step	**Action**	**Response**	**Pass/Fail**	**Comment**

1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Check email	Pass	

Post-conditions: Patient has successfully created her account

5.6.26 Login for Patient

Test case #26			**Test case name**: Login		
System: Remote Doctor			**Subsystem:**		
Designed by: Raihan Mahmud			**Designed date:** 02-Oct-2019		
Executed by: Raihan Mahmud			**Executed date:** 20-Nov-2019		
Short description: Patient needs to login for access to the patient panel and before login system check authentication.					
Pre-conditions: User type must be patient Assume that, the email is raihan35@gmail.com and password "1234567"					
Step	**Email**	**Password**	**Response**	**Pass/Fail**	**Comment**
1	raihan@gmail.com	1765432	Wrong email or password	Fail	
2	raihan@gmail.com		Wrong email or password	Fail	
3		1234567	Email required	Fail	
4	raihan35@gmail.com	1234567	Show admin panel	Pass	
Post-conditions: Patient will successfully login to the patient panel					

5.6.27 View Profile for Patient

Test case #27		**Test case name**: View Profile		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 03-Oct-2019		
Executed by: Raihan Mahmud		**Executed date:** 20-Nov-2019		
Short description: Patient can view her profile				
Pre-conditions: User type must be patient Patient must be authorized Patient must be login				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click "Profile" option	Show profile	Pass	
Post-conditions: Patient able to see her profile				

5.6.28 Update Profile for Patient

Test case #28		**Test case name**: Update Profile		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 04-Oct-2019		
Executed by: Raihan Mahmud		**Executed date:** 20-Nov-2019		
Short description: Patient can updated her information				
Pre-conditions: User type must be patient Patient must be authorized Patient must be login				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Show profile	Pass	
Post-conditions: Patient successfully updated her information				

5.6.29 Take Appointment for Patient

Test case #29		**Test case name**: Take Appointment		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 05-Oct-2019		
Executed by: Raihan Mahmud		**Executed date:** 20-Nov-2019		
Short description: Patient can take appointment from the doctor				
Pre-conditions: User type must be patient Patient must be authorized Patient must be login Doctor must be available				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Appointment” option, choose doctor and confirm appointment.	Show status “Pending”	Pass	
Post-conditions: Patient successfully took an appointment				

5.6.30 Submit Report

Test case #30	**Test case name**: Submit Report
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 06-Oct-2019
Executed by: Raihan Mahmud	**Executed date:** 20-Nov-2019
Short description: Patient can submit the report by taking picture	

Pre-conditions:
User type must be patient
Patient must be authorized
Patient must be login

Step	Action	Response	Pass/Fail	Comment
1	Click “Reports” option	Empty report list show	Fail	This is an initial stage. So here, no test was recorded.
2	Click “Reports” option, click “Submit Report” button and submit.	Show list	Pass	

Post-conditions: Patient submitted the report successfully

5.6.31 View Prescription

Test case #31	**Test case name**: View Prescription
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 07-Oct-2019
Executed by: Raihan Mahmud	**Executed date:** 20-Nov-2019

Short description: Patient can view her prescription

Pre-conditions:
User type must be patient
Patient must be authorized
Patient must be login

Step	Action	Response	Pass/Fail	Comment
1	Click “Prescription” option	Empty prescription show	Fail	This is an initial stage. So here, no prescription was recorded.
2	Click “Prescription” option	Show prescription	Pass	

Post-conditions: Patient able to see her prescription

5.6.32 Live Conference for Patient

Test case #32	**Test case name**: Live Conference
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 08-Oct-2019
Executed by: Raihan Mahmud	**Executed date:** 20-Nov-2019

Short description: Patient can live conference with doctor

Pre-conditions:
User type must be patient

Patient must be authorized Patient must be login Must be patient appointment with doctor				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Chat” option	There are no doctor show	Fail	This is an initial stage. So here, no patient was recorded.
2	Click “Chat” option, choose doctor and connected with doctor.	Show list	Pass	
Post-conditions: Patient successfully conference with doctor				

5.6.33 Registration for Coordinator

Test case #33		**Test case name**: Registration		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 09-Oct-2019		
Executed by: Raihan Mahmud		**Executed date:** 21-Nov-2019		
Short description: Coordinator can create her account				
Pre-conditions: User type must be coordinator				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Check email	Pass	
Post-conditions: Coordinator has successfully created her account				

5.6.34 Login for Coordinator

Test case #34			**Test case name**: Login		
System: Remote Doctor			**Subsystem:**		
Designed by: Raihan Mahmud			**Designed date:** 10-Oct-2019		
Executed by: Raihan Mahmud			**Executed date:** 21-Nov-2019		
Short description: Coordinator needs to login for access to the coordinator panel and before login system check authentication.					
Pre-conditions: User type must be coordinator Assume that, the email is raihan35@gmail.com and password “1234567”					
Step	**Email**	**Password**	**Response**	**Pass/Fail**	**Comment**
1	raihan@gmail.com	1765432	Wrong email or password	Fail	

2	raihan@gmail.com		Wrong email or password	Fail	
3		1234567	Email required	Fail	
4	raihan35@gmail.com	1234567	Show admin panel	Pass	

Post-conditions: Coordinator will successfully login to the coordinator panel

5.6.35 View Profile for Coordinator

Test case #35		**Test case name**: View Profile		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 11-Oct-2019		
Executed by: Raihan Mahmud		**Executed date:** 21-Nov-2019		
Short description: Coordinator can view her profile				
Pre-conditions: User type must be coordinator Coordinator must be authorized Coordinator must be login				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click “Profile” option	Show profile	Pass	
Post-conditions: Coordinator able to see her profile				

5.6.36 Update Profile for Coordinator

Test case #36		**Test case name**: Update Profile		
System: Remote Doctor		**Subsystem:**		
Designed by: Raihan Mahmud		**Designed date:** 12-Oct-2019		
Executed by: Raihan Mahmud		**Executed date:** 21-Nov-2019		
Short description: Coordinator can updated her information				
Pre-conditions: User type must be coordinator Coordinator must be authorized Coordinator must be login				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	All required field are not filled yet	Show required message	Fail	
2	All input field is filled up	Show profile	Pass	
Post-conditions: Coordinator successfully updated her information				

5.6.37 Take Appointment for Coordinator

Test case #37	**Test case name**: Take Appointment
System: Remote Doctor	**Subsystem:**
Designed by: Raihan Mahmud	**Designed date:** 13-Oct-2019
Executed by: Raihan Mahmud	**Executed date:** 21-Nov-2019
Short description: Coordinator can take appointment from the doctor	
Pre-conditions: User type must be coordinator Coordinator must be authorized Coordinator must be login Doctor must be available	

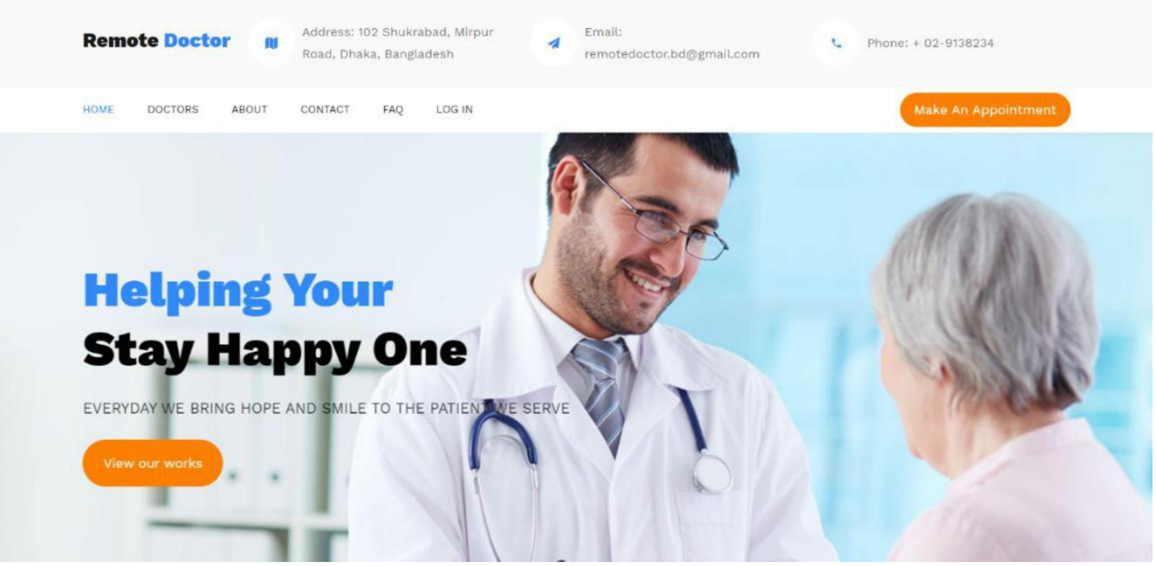
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click "Appointment" option, choose doctor and patients email field are not filled yet.	Show required message	Fail	
2	Click "Appointment" option, choose doctor and confirm appointment.	Show status "Pending"	Pass	
Post-conditions: Coordinator successfully took an appointment				

Chapter 6

User Manual

6.1 Home Page

Home page describe all kind of information about Remote Doctor application.



Address: 102 Shukrabad, Mirpur Road, Dhaka, Bangladesh

Email: remotedoctor.bd@gmail.com

Phone: + 02-9138234

EVERYDAY WE BRING HOPE AND SMILE TO THE PATIENT WE SERVE

Figure 6.1: Home page.

6.1.1 Doctors Page

In doctors page display registered doctor profile. Patient or coordinator can find doctor for appointment.

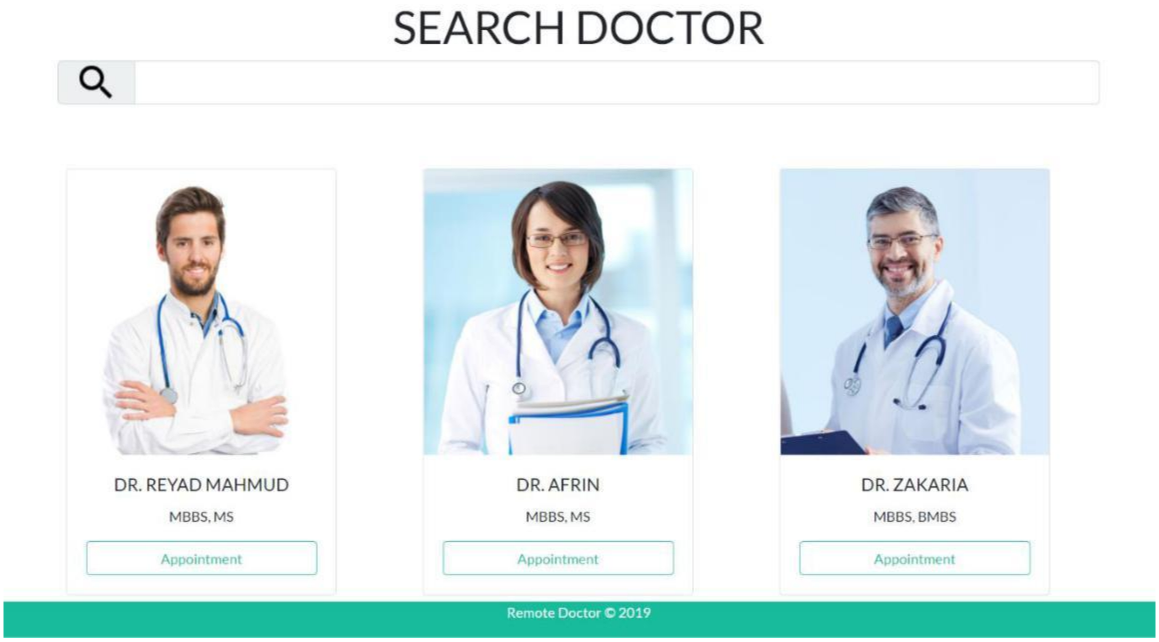
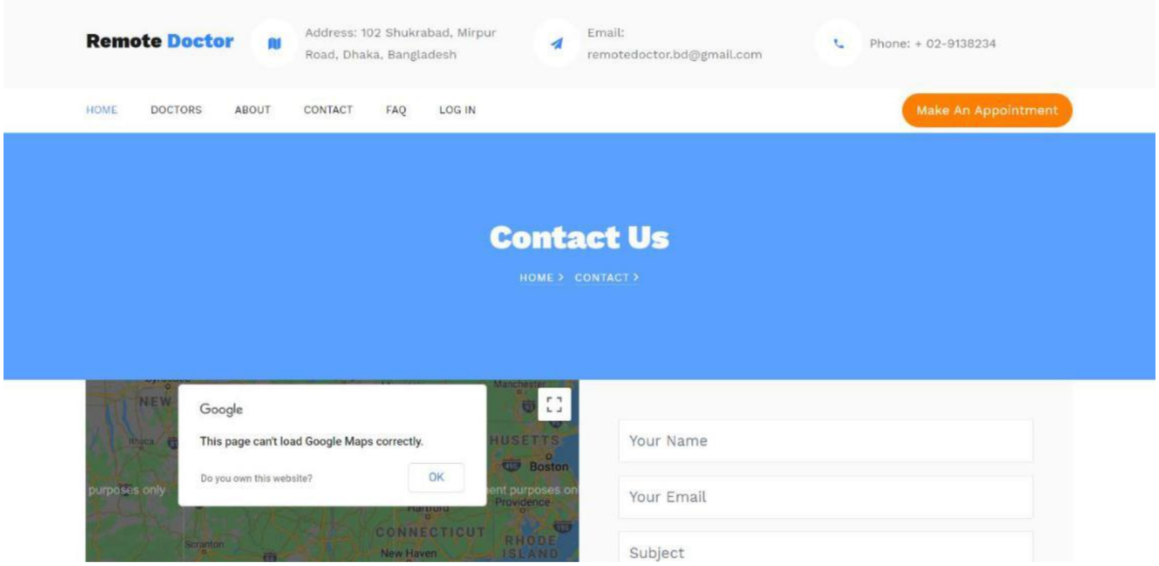


Figure 6.2: Doctors page.

6.1.2 Contact Page

Can any kind of query or feedback to admin using Contact page.



Remote Doctor

Address: 102 Shukrabad, Mirpur Road, Dhaka, Bangladesh

Email: remotedoctor.bd@gmail.com

Phone: + 02-9138234

Contact Us

Google

This page can't load Google Maps correctly.

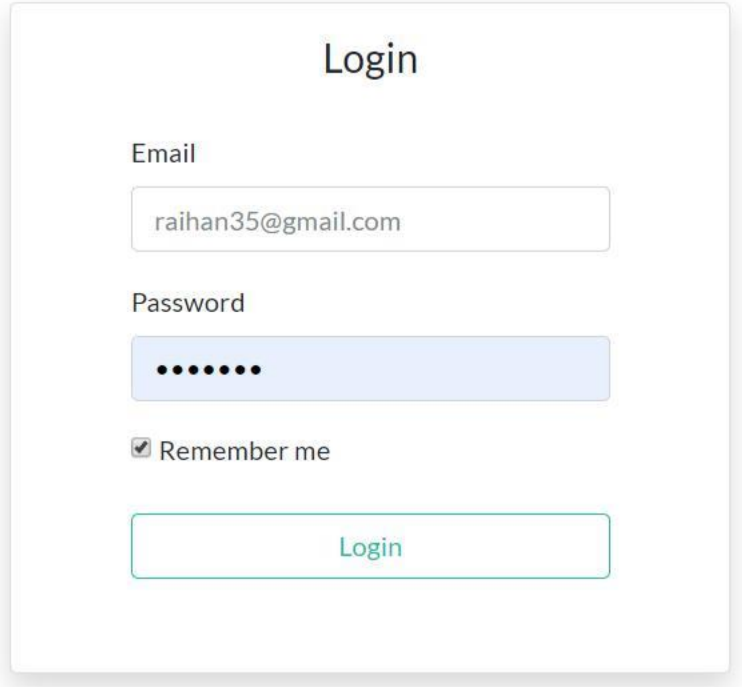
Do you own this website?

Figure 6.3: Contact page.

6.2 Admin module

6.2.1 Login for Admin

Enter valid email and password admin can enter admin panel**.**



Login

Email: raihan35@gmail.com

Password: ●●●●●●●

☑ Remember me

Figure 6.4: Login page for admin.

6.2.2 Admin Panel

After login display admin dashboard. Dashboard show real time weather, date and total number of registered user.

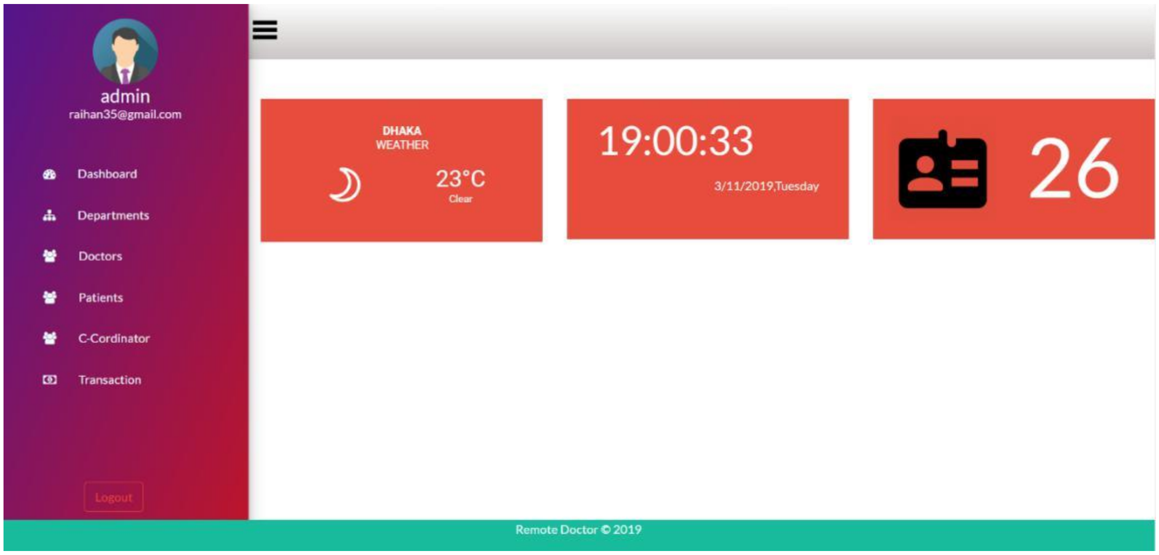


Figure 6.5: Admin dashboard page.

6.2.3 Doctors

Click "Doctors", display list of doctors and Add Doctor, Edit, Delete option. Admin can register new doctor and edit doctor info, delete doctor. As same as departments, patients, coordinator features.

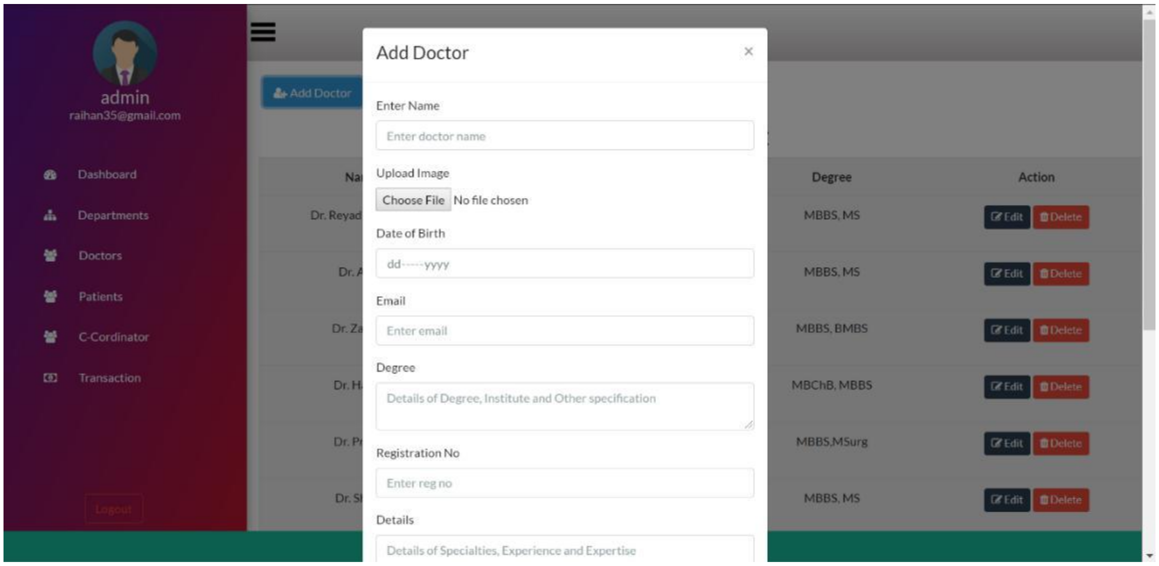
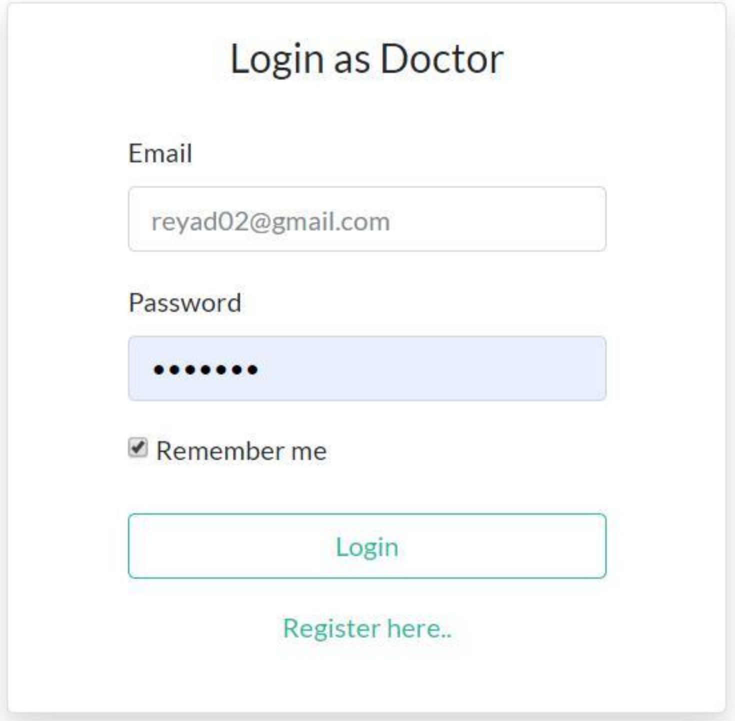


Figure 6.6: Doctors page.

6.3 Doctor module

6.3.1 Login for Doctor

Go to **Home** >> **LOG IN**, enter valid email and password register doctor enter doctor panel. If not register create account click “Register here” and enter valid info.



Login as Doctor

Email

reyad02@gmail.com

Password

•••••••

☑ Remember me

Figure 6.7: Login page for doctor.

6.3.2 Doctor Dashboard

Dashboard show real time weather, date and total number of appointment request.

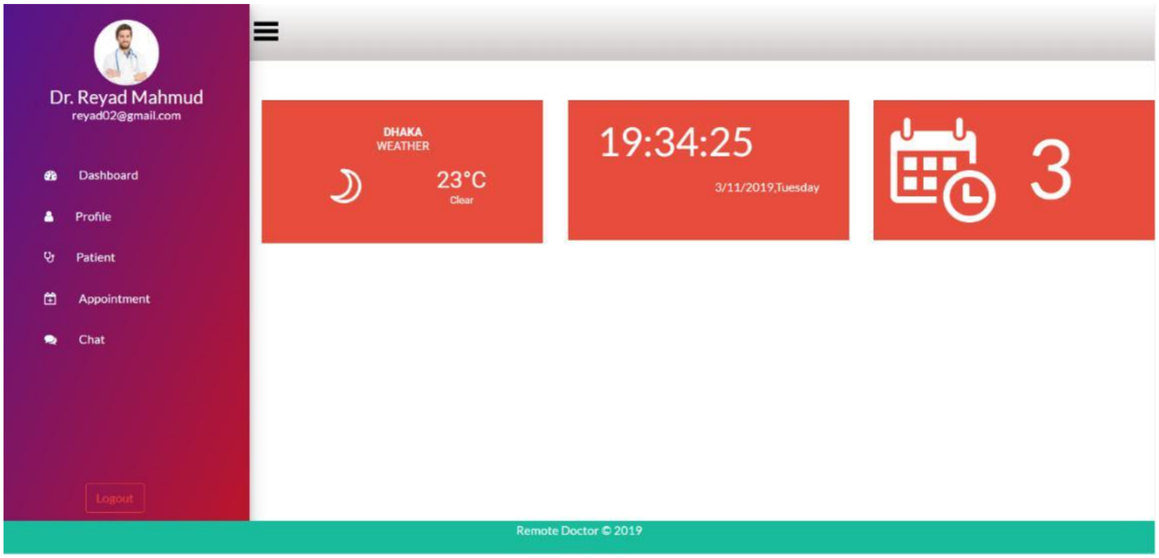
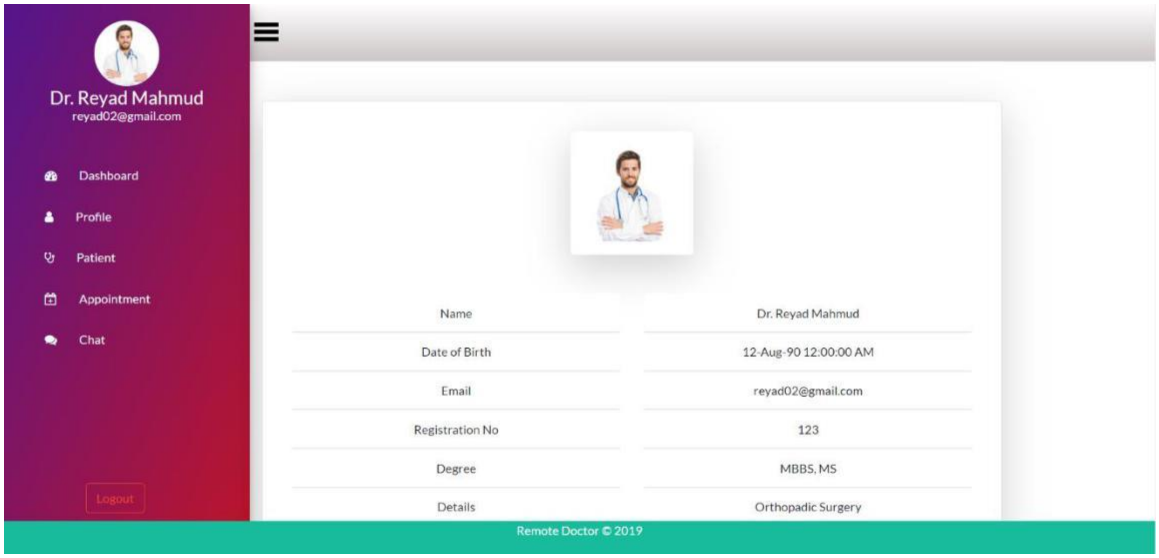


Figure 6.8: Doctor dashboard page.

6.3.3 Doctor Profile

Click “Profile”, display information of authorize doctor. Doctor can update her info click “Update Information”.

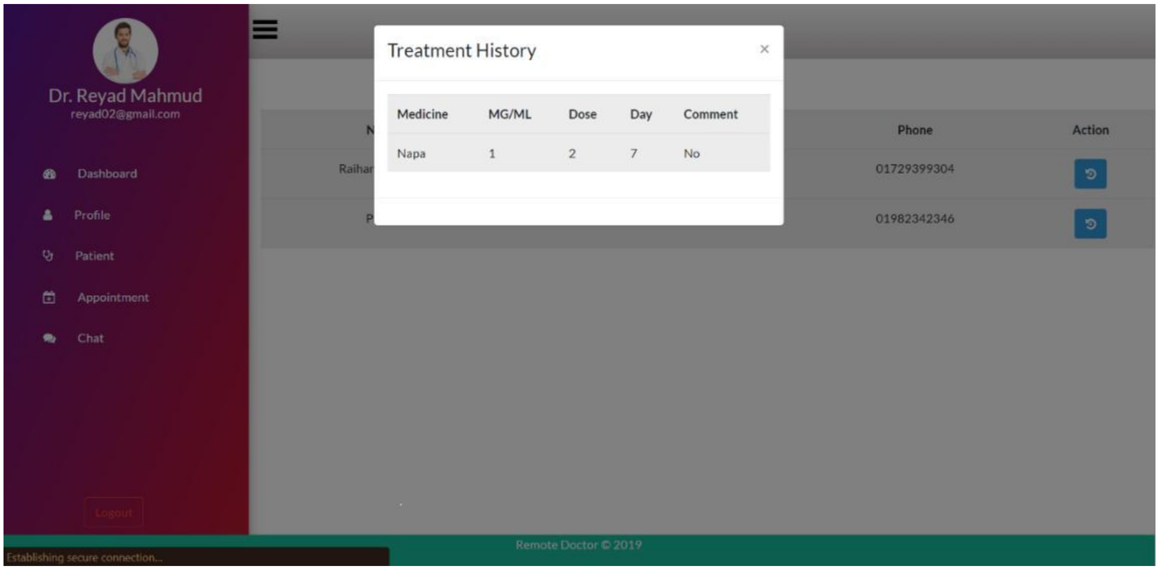


Name	Dr. Reyad Mahmud
Date of Birth	12-Aug-90 12:00:00 AM
Email	reyad02@gmail.com
Registration No	123
Degree	MBBS, MS
Details	Orthopadic Surgery

Figure 6.9: Doctor profile page.

6.3.4 Treatment History

Click “Patient”, display previous patients list and action show treatment history.



Treatment History ×

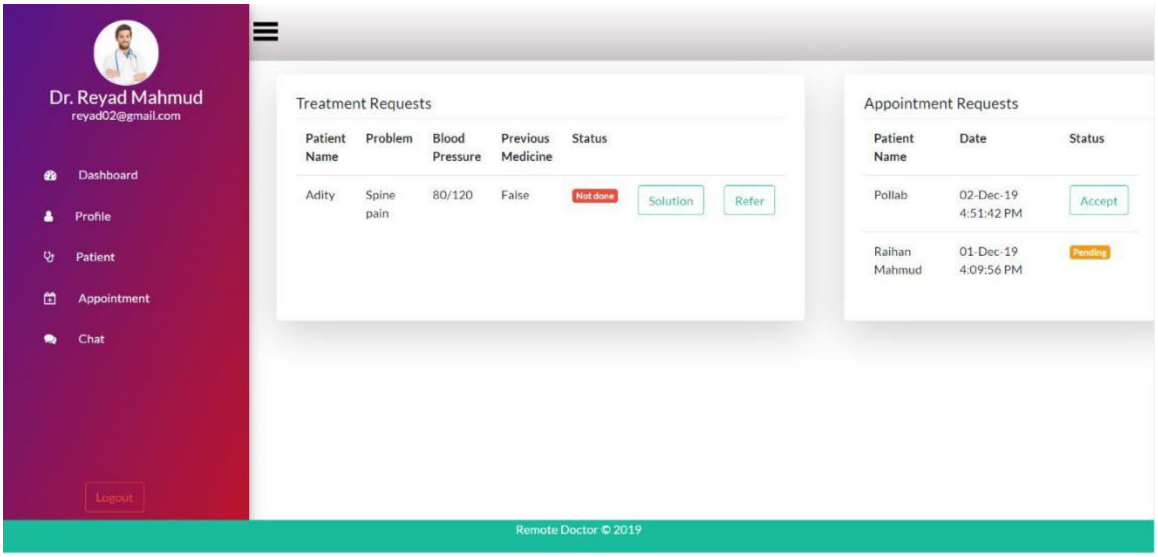
Medicine	MG/ML	Dose	Day	Comment
Napa	1	2	7	No

N	Phone	Action
Raihar	01729399304	⟲
P	01982342346	⟲

Figure 6.10: Treatment history page.

6.3.5 Appointment

Click “Appointment”, display list of treatment request in left site and appointment request in right site. After treatment status show “Success” otherwise “Not done” and accepting request status show “Pending”.



Treatment Requests

Patient Name	Problem	Blood Pressure	Previous Medicine	Status		
Adity	Spine pain	80/120	False	Not done	Solution	Refer

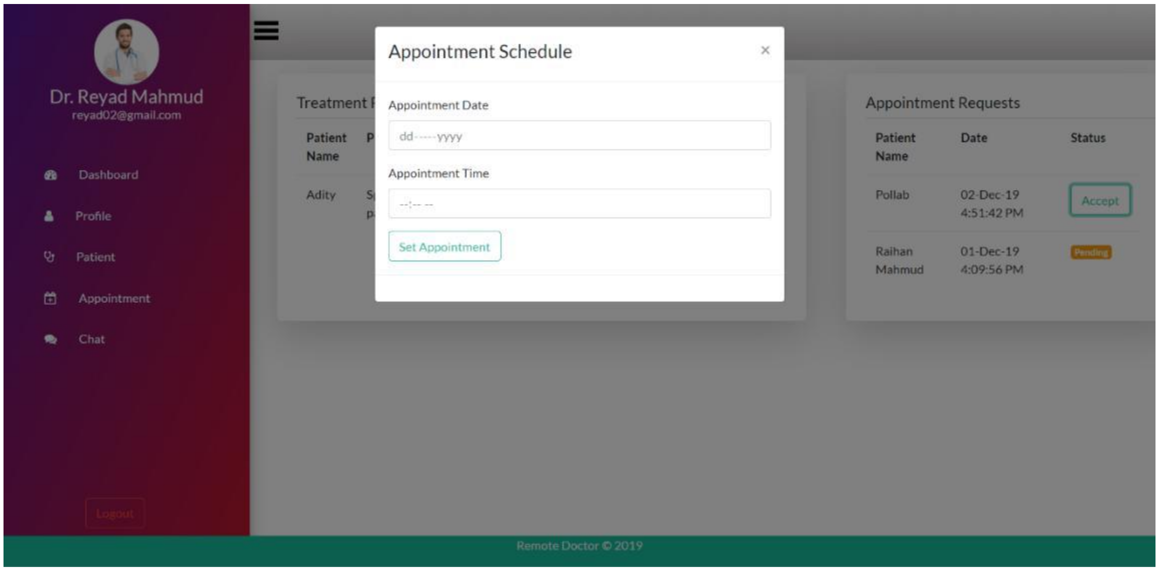
Appointment Requests

Patient Name	Date	Status
Pollab	02-Dec-19 4:51:42 PM	Accept
Raihan Mahmud	01-Dec-19 4:09:56 PM	Pending

Figure 6.11: Appointment page for doctor.

6.3.6 Appointment Requests

Click “Accept” button, doctor can set appointment schedule.



Appointment Schedule ×

Appointment Date

dd-----yyyy

Appointment Time

--:-- --

Set Appointment

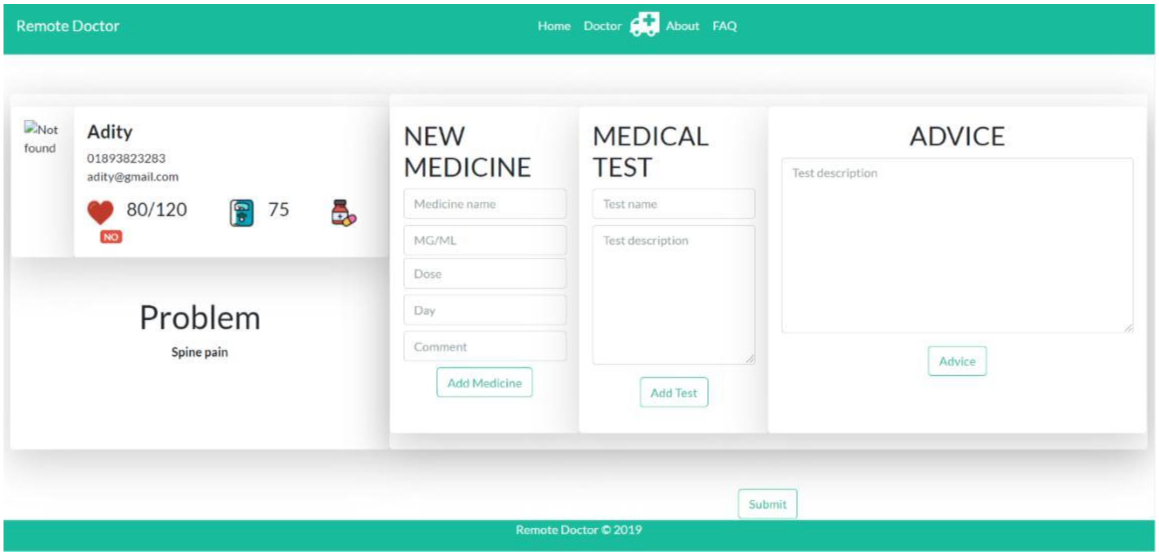
Appointment Requests

Patient Name	Date	Status
Pollab	02-Dec-19 4:51:42 PM	Accept
Raihan Mahmud	01-Dec-19 4:09:56 PM	Pending

Figure 6.12: Accept appointment page.

6.3.7 Treatment

Click “Solution” button, show this page. Doctor can prescribe, test and advice to patient. After click “Submit” button all medicine, test and advice passed to the patient. Otherwise click “Refer” button, doctor can refer patient to another doctor.



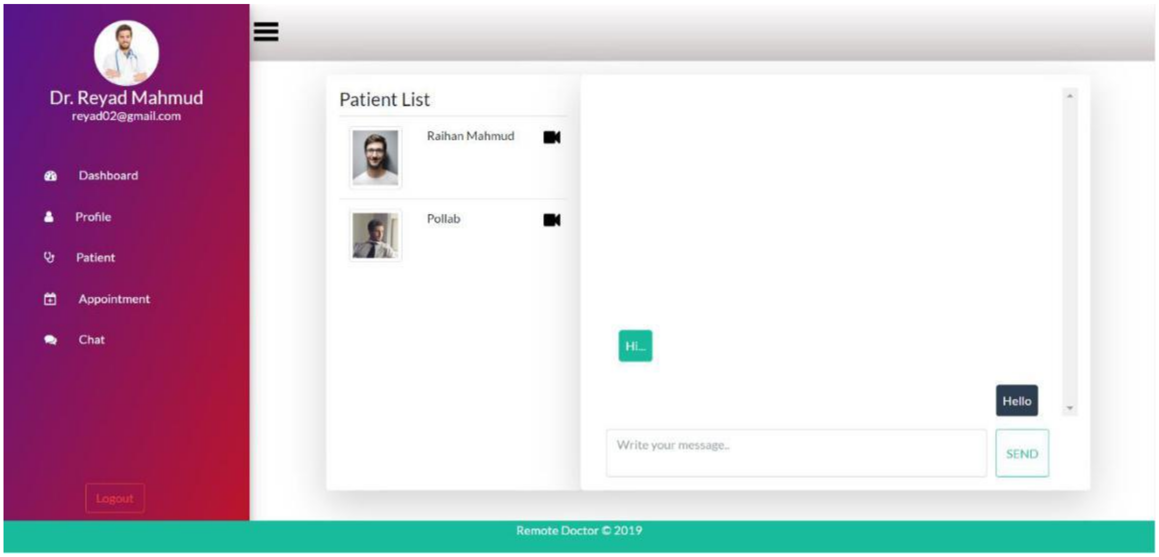
Adity
01893823283
adity@gmail.com
80/120
NO
75

Problem
Spine pain

Figure 6.13: Treatment page.

6.3.8 Live Conference

Click “Chat”, display list of appointments patient. Doctor can chat with specific patient or connected with patient via video call after click on video icon.



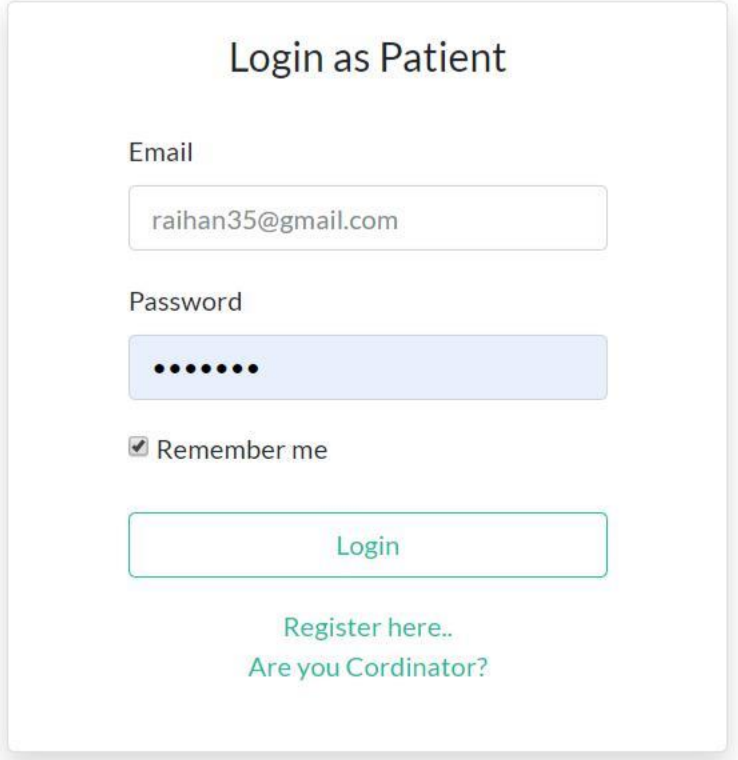
Hi...
Hello

Figure 6.14: Live chat page for doctor.

6.4 Patient module

6.4.1 Login for Patient

Go to **Home** >> **Make An Appointment**, enter valid email and password register patient enter patient panel. If not register create account click “Register here” and enter valid info.



Login as Patient

Email

raihan35@gmail.com

Password

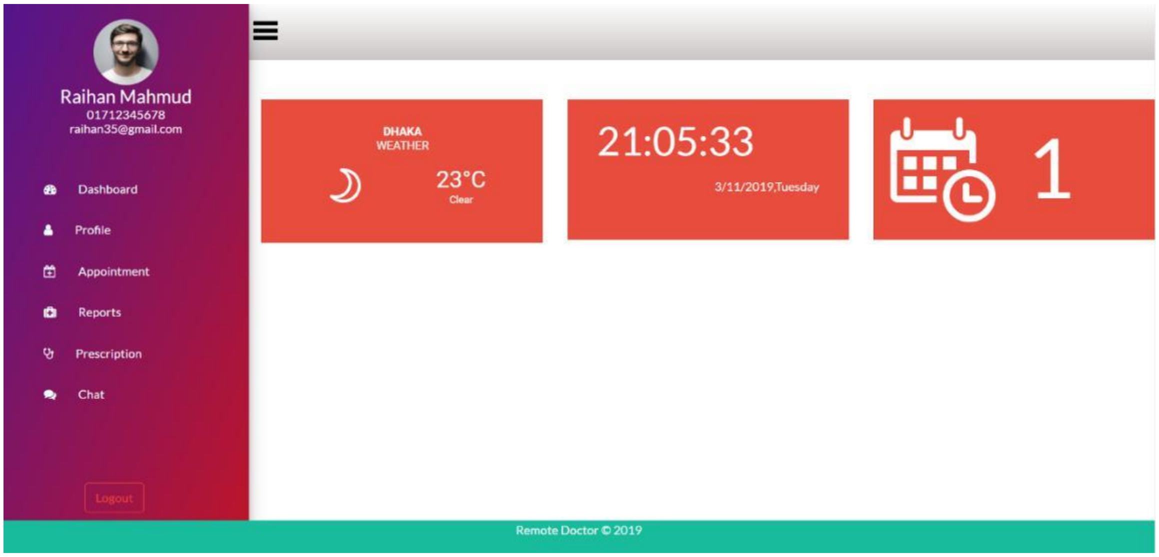
•••••••

☑ Remember me

Figure 6.15: Login page for patient.

6.4.2 Patient Dashboard

Dashboard show real time weather, date and total number of appointment.



Raihan Mahmud
01712345678
raihan35@gmail.com

DHAKA
WEATHER
23°C
Clear

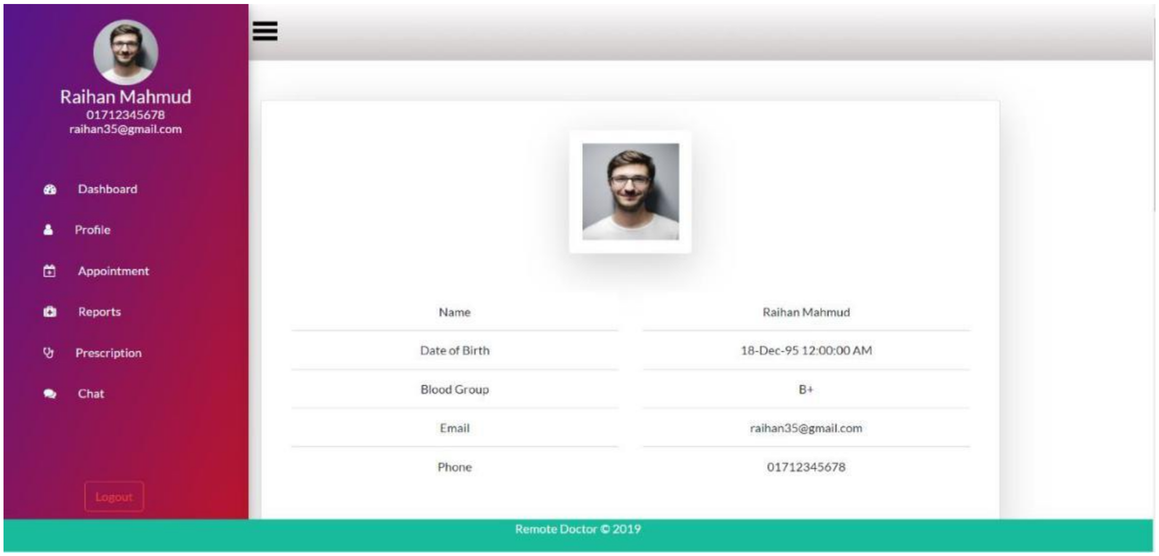
21:05:33
3/11/2019,Tuesday

1

Figure 6.16: Patient dashboard page.

6.4.3 Patient Profile

Click “Profile”, display information of authorize patient. Patient can update her info click “Update Information”.

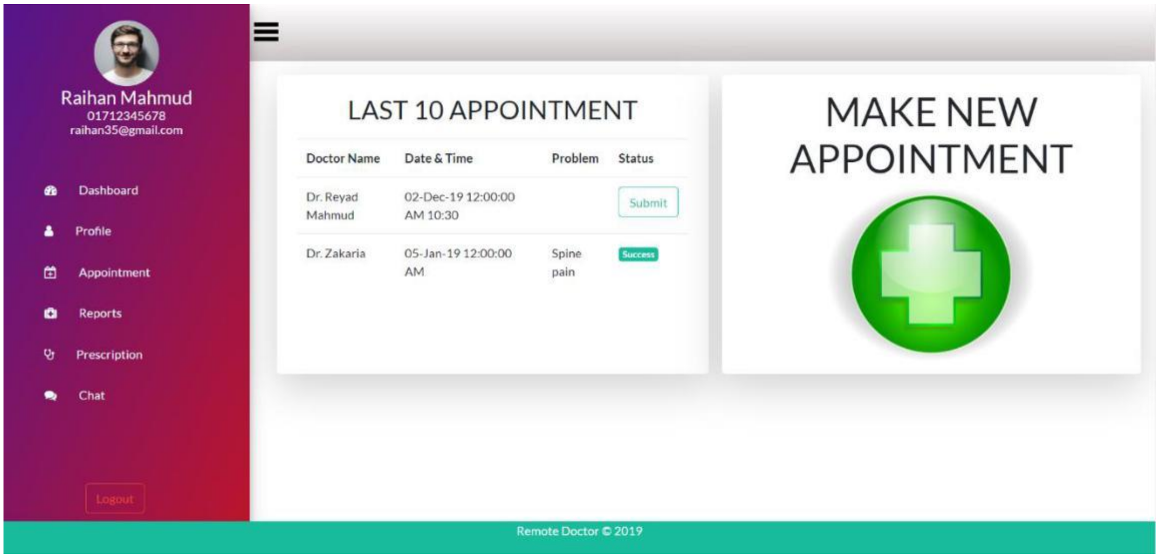


Name	Raihan Mahmud
Date of Birth	18-Dec-95 12:00:00 AM
Blood Group	B+
Email	raihan35@gmail.com
Phone	01712345678

Figure 6.17: Patient profile page.

6.4.4 Appointment

Click “Appointment”, display list of last 10 appointment. Patient can make appointment click plus button and sent appointment request to doctor, status show “Pending”. After accepting appointment request show “Submit” button and appointment time. Patient can click “Submit” button and basic required info submit to doctor for treatment. After submission status show “Waiting” and successfully treatment status show “Success”.



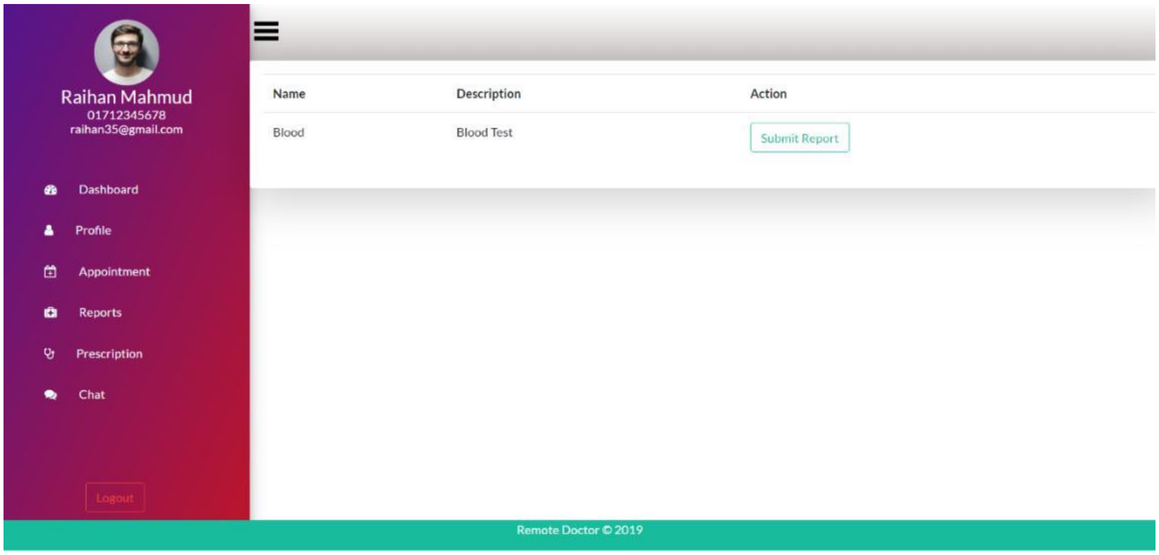
LAST 10 APPOINTMENT

Doctor Name	Date & Time	Problem	Status
Dr. Reyad Mahmud	02-Dec-19 12:00:00 AM 10:30		Submit
Dr. Zakaria	05-Jan-19 12:00:00 AM	Spine pain	Success

Figure 6.18: Appointment page for patient.

6.4.5 Reports

Click “Reports”, display list of reports and submission for proper treatment. After test, patient can submit test report like picture or pdf format.



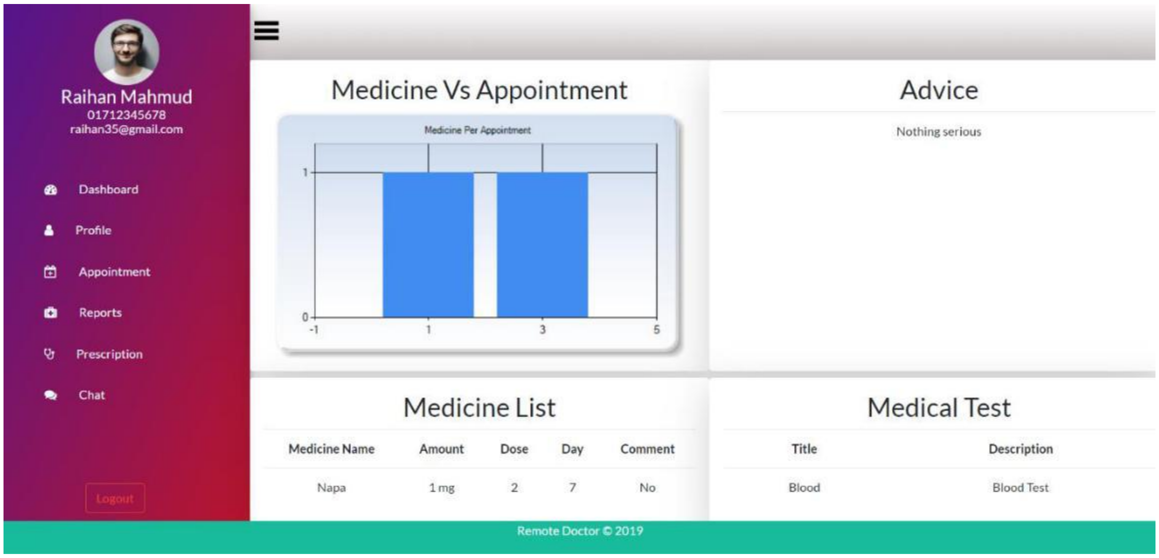
Raihan Mahmud
01712345678
raihan35@gmail.com

Name	Description	Action
Blood	Blood Test	Submit Report

Figure 6.19: Reports page.

6.4.6 Prescription

Click “Prescription”, display list of medicine in left bottom site, test list in right bottom site, advice in right top site and medicine vs appointment ratio in left top site.



Raihan Mahmud
01712345678
raihan35@gmail.com

Medicine Vs Appointment

Advice

Nothing serious

Medicine List

Medicine Name	Amount	Dose	Day	Comment
Napa	1 mg	2	7	No

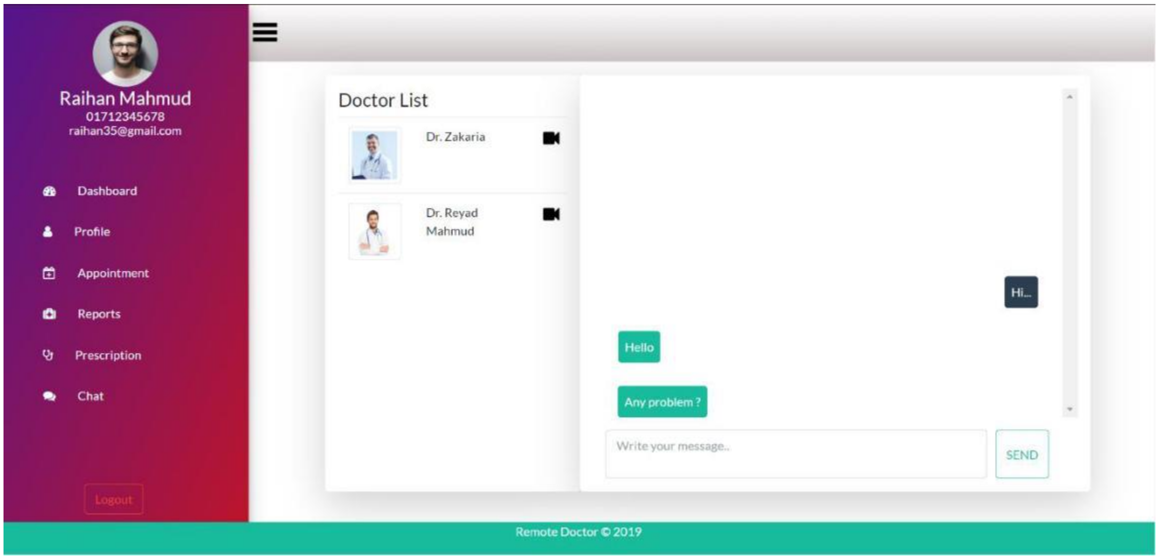
Medical Test

Title	Description
Blood	Blood Test

Figure 6.20: Prescription page.

6.4.7 Live Conference

Click “Chat”, display list of appointments doctor. Patient can chat with specific doctor or connected with doctor via video call after click on video icon.



Hi..

Hello

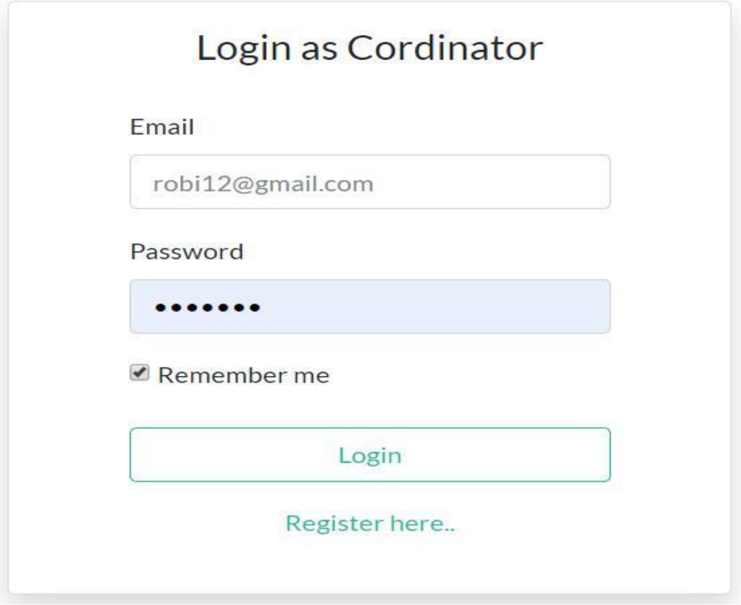
Any problem ?

Figure 6.21: Live chat page for patient.

6.5 Coordinator module

6.5.1 Login for Coordinator

Go to **Home** >> **Make An Appointment** >> **Are you Cordinator**, enter valid email and password register coordinator enter coordinator panel. If not register create account click “Register here” and enter valid info.



Login as Cordinator

Email: robi12@gmail.com

Password: •••••••

☑ Remember me

Figure 6.22: Login page for coordinator.

6.5.2 Coordinator Dashboard

Dashboard show real time weather, date and total number of appointment.

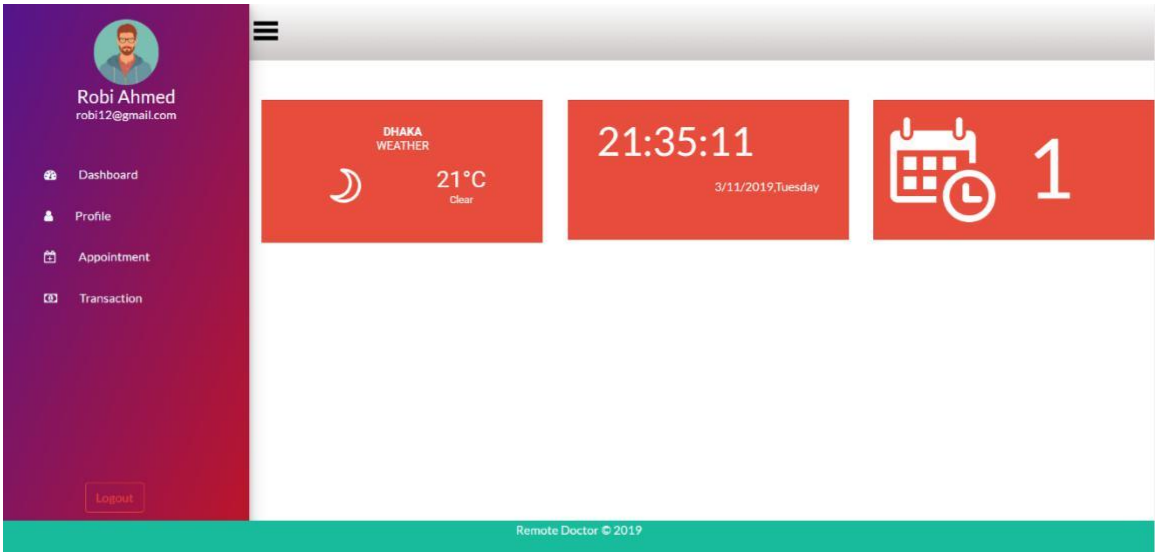
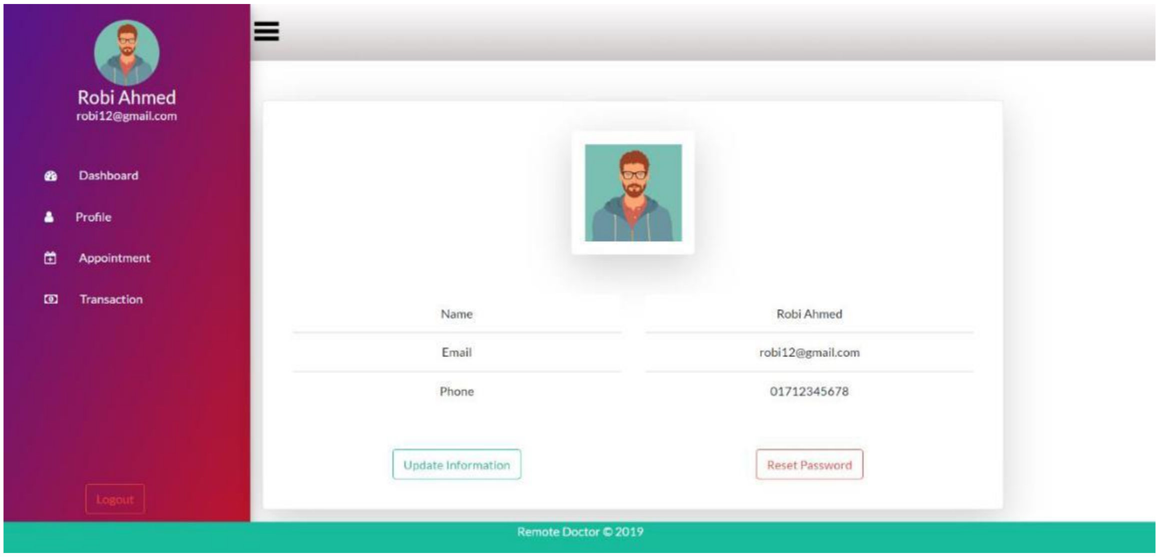


Figure 6.23: Coordinator dashboard page.

6.5.3 Coordinator Profile

Click “Profile”, display information of authorize coordinator. Coordinator can update her info click “Update Information”.

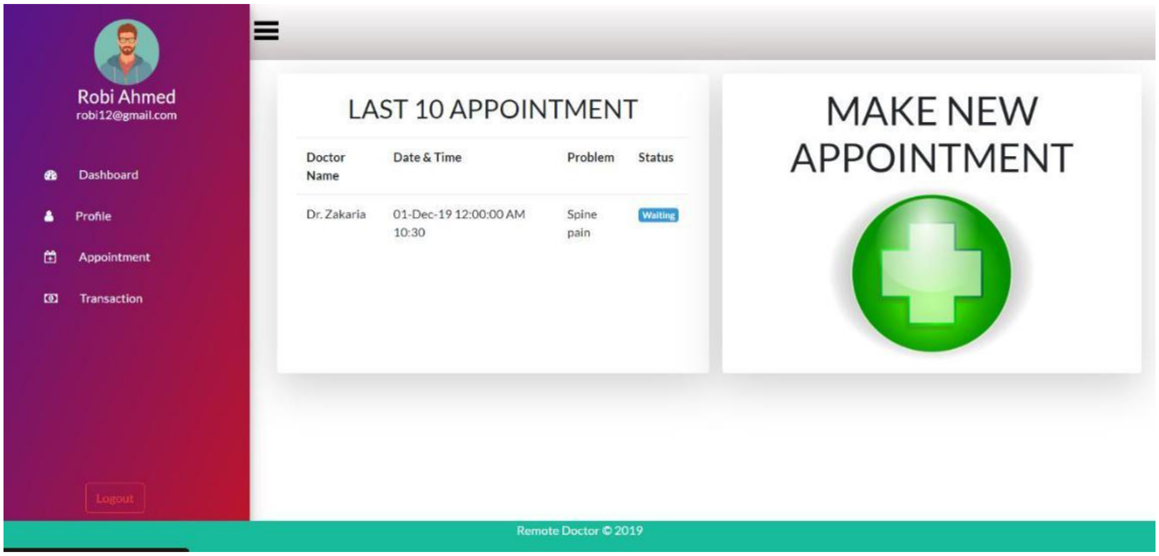


Name	Robi Ahmed
Email	robi12@gmail.com
Phone	01712345678

Figure 6.24: Coordinator profile page.

6.5.4 Appointment

Click “Appointment”, display list of last 10 appointment. Coordinator can make patient appointment click plus button and enter required patient email, sent appointment request to doctor, status show “Pending”. After accepting appointment request show “Submit” button and appointment time. Coordinator can click “Submit” button and basic required info submit to doctor for patient treatment. After submission status show “Waiting” and successfully treatment status show “Success”.

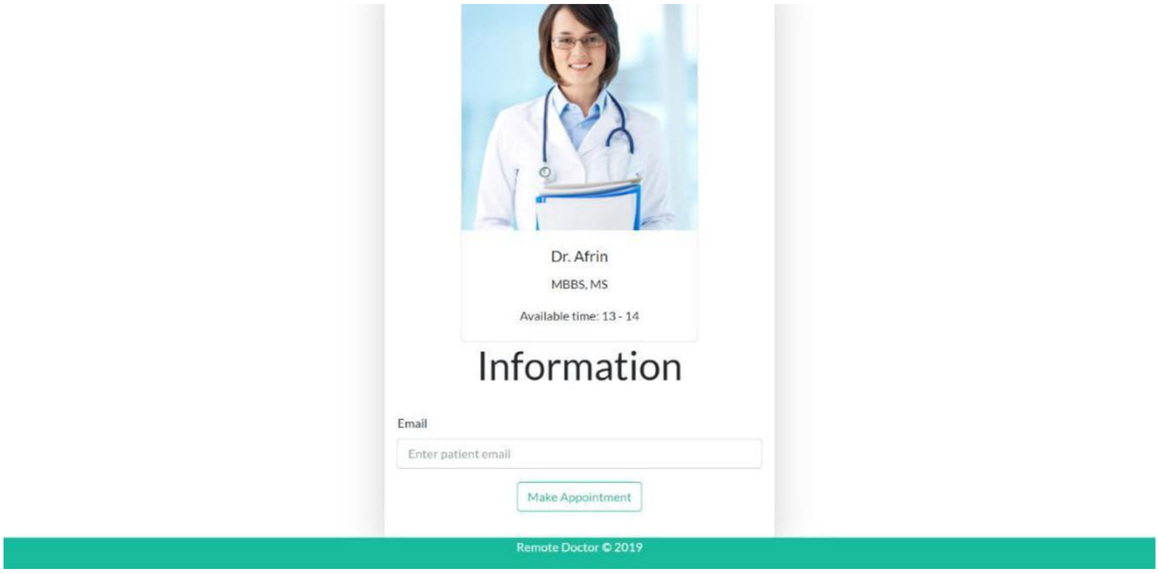


LAST 10 APPOINTMENT

Doctor Name	Date & Time	Problem	Status
Dr. Zakaria	01-Dec-19 12:00:00 AM 10:30	Spine pain	Waiting

Figure 6.25: Appointment page for coordinator.

Coordinator can enter specific patient email for make an appointment.



Dr. Afrin

MBBS, MS

Available time: 13 - 14

Information

Email

Enter patient email

Figure 6.26: Make appointment page.

Chapter 7

Conclusion

7.1 GitHub Link

- https://github.com/raihanM95/Remote-Doctor

7.2 Project Summary

I have started the development part on May, 2019. As like as any others project, you have to put yours very hard working, patience, dedication, and concentration to complete the project. There are many factors comes to when a project going to the development phase, such as fulfilling the stakeholder requirements properly, etc.

A proper requirements analysis then it helps the development phase a lot. We analysis the requirements of our project firstly then we are going to the next step design specification.

Database diagram is an important factor of any application system. That's why in the next step we are focusing to create the database design. We have designed class diagram with the proper relationship then design database using entity. In our project every module dependent on other module. So, we start development from doctor module and use dami data for uncomplete module. After that, we are checking the all thing again and move to the core functionality of the project.

It's not easy task to complete project but if it to be completed then the project is not the and actually. At the end of complete the project, you have to make sure that your project functionality works fine. For that, you have to come in the testing part, it's part of quality assurance. The responsibility of quality assurance is to find the vulnerability of the system. If any bug can be found before the system release then there is a change to fix that bug. So testing the project we have assured the quality of the project.

7.3 Limitations

For developing this project, we have faced some limitations. Now we will describe those in brief.

Payment Method: In this application, there are not having any payment method directly. So the user can't make any payment transaction through our system. User pays the payment manually.

Only Web Version: Our system in only web-based but there are many other platforms is popular such as Mobile version, iOS version, etc.

7.4 Obstacles and Achievements

When a project is developing then you have face many obstacles and those obstacle create challenges and you have to overcome this challenges. Without challenges you can't learn anything, challenge comes with opportunity and here your change to grave this opportunity.

By doing this project we have learned how collect the requirements and how to analysis them. Brainstorming is must for doing any project you have to do it for sure. The system analysis and database design that we learn properly by doing this project. My supervisor helps me in every step of this project

Also there are some obstacles and achievement in our project and that is below:

> **Requirements getting from stakeholders**: There are many types of stakeholders in this system and everyone have different functionality. It would be better, if we could gather requirements from them directly.
>
> **Scope Change:** When we check the requirements then some features has been added and a few of feature get cutting. When it was needed we did brainstorming for that.

7.5 Future Scope

I learned a lot while finishing this project. To build and improve this project, I have met some young entrepreneurs and enthusiasts. I thank all of them very much.

Besides, I am grateful to them because I have been able to complete my project by adopting their important opinions and discussions. It will help me in the future to work of this project.

References

[1] W3School, https://www.w3schools.com/

[2] ASP.NET Documentation, https://docs.microsoft.com/en-us/aspnet/#pivot=aspnet

[3] Bootstrap, https://getbootstrap.com/docs/4.4/getting-started/introduction/

[4] jQuery, https://jquery.com/

[5] Font Awesome, https://fontawesome.com/v4.7.0/icons/

[6] Dotnet Awesome, http://www.dotnetawesome.com/

[7] colorlib., https://colorlib.com/wp/