



Daffodil
International
University

TITLE OF THE PROJECT

E-psychologist.com

Submitted By

Maheen Absar

(ID: 171-35-1961)

Supervised By

Mr. Khalid Been Badruzzaman Biplob

Lecturer (Senior Scale)

Faculty of Science and Information Technology

Department of Software Engineering

Daffodil International University

Fall-2020

A project (SWE 431) submitted in fulfillment of the

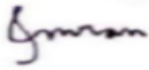
Requirements for the degree of

BSc in Software Engineering

APPROVAL

This project titled on “E-Psychologist”, submitted by Maheen Absar, ID: 171-35-1961 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



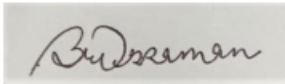
Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

Chairman



Syeda Sumbul Hossain
Senior Lecturer
Department of Software Engineering
Daffodil International University

Internal Examiner 1



Khalid Been Badruzzaman Biplob
Senior Lecturer
Department of Software Engineering
Daffodil International University

Internal Examiner 2



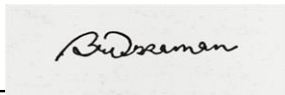
Professor Dr. Mohammed Nasir Uddin
Department of Computer Science and Engineering
Jagannath University, Dhaka

External Examiner

DECLARATION

I, the author, hereby declare that the project becomes the property of Daffodil International University (DIU). I give permission to the DIU that they can change in whole or in part for the purpose of research or academic exchange only.

Supervised By



Mr. Khalid Been Badruzzaman Biplob

Lecturer (Senior Scale)

Department of Software Engineering Daffodil International University

Submitted By



Maheen Absar

ID: 171-35-1961

Department of Software Engineering

Daffodil International University

ACKNOWLEDGEMENT

At first, I am very grateful to the almighty Allah who gave me the opportunity to complete this project by keeping me mentally strong and physically fit. I am also very grateful to my parents who always inspire me to do good things, study attentively and do something for my country. They inspired me a lot to complete this project during this hard situation of covid-19. My teachers helped me a lot to achieve the qualifications to complete this project. They always taught me how to improve my skills, become a good person and become a successful man in my carrier. They all are my inspirations. I get mental strength from their motivational speeches, which encouraged me a lot, and I trusted believing in my skills, do hard work. I want thank all my well-wishers, my friends and many others who helped me a lot to complete this project in spite of being in a tough situation like Covid19 pandemic. I will always remain very grateful to my supervisor sir who kept giving me guidelines continuously, motivating me through online meetings. He inspired me a lot to continue my studies in spite of physical classes being not available in this time as well as to complete this project by giving strength.

Abstract

The project that I have completed is "e-Psychologist" which is a web-based project. The people of the third world countries are not getting any psychiatric services or psychological consultancy even though they have access to basic medical services nowadays. As a result, they are living in the midst of various mental complications including, suicide and depression. So I have developed a system which is E-Psychologist. By using this system, any people from any place who has computer or phone in his hand with internet connection can take psychological treatment at any time. In this system, there are many sections such as online therapy; tutorials and it contains varieties of psychological information. There are four types of users in this system such as Doctor, Head of Doctor, Admin and patient. By using this system, a very helpful connection will be made between the doctors and the patients. By using this system patients can take appointments and the doctors can see the appointment requests of the patients. Doctors also can post information, therapy and tutorials in this system. Head of doctors and admin will look after the whole system.

Table of Contents

1. CHAPTER-1 INTRODUCTION	1
1.1 Project Overview	1
1.2 Project Purpose.....	1
1.2.1 Background.....	1
1.2.2 Advantages of users	1
1.2.3 Goals	2
1.3 Stakeholders	2
1.4 Proposed System Model (Block Diagram)	2
1.5 Project Schedule	3
1.5.1 Gantt Chart	3
2. CHAPTER-2 SOFTWARE REQUIREMENT SPECIFICATION	4
2.1 Functional Requirements.....	4
2.2 Non-Functional Requirements	6
2.3 Performance requirements.....	7
2.3.1 Speed and Latency requirements	7
2.3.2 Accuracy Requirements	7
2.3.3 Capacity requirements	7
2.4 Dependability Requirements	7
2.5 Maintainability and supportability requirements.....	7
2.5.1 Maintenance Requirements	7
2.5.2 Supportability Requirements	7
2.5.3 Adaptability Requirements	7
2.6 Security Requirements	8
2.6.1 Access Requirement	8
2.6.2 Integrity Requirement.....	8
2.6.3 Privacy Requirements	8
2.6.4 Authentication Requirements	8
2.6.5 Accessibility Requirements	8
2.7 User Friendliness Requirements.....	8
2.8 Operational and Environmental Requirements	8
2.8.1 Release Requirements.....	9
2.9 Legal Requirements	9
2.10 Standard Requirements	9

3. CHAPTER-03 SYSTEM ANALYSIS	10
3.1 Use Case	10
3.2 Use Case Description	11
3.2.1 Login.....	11
3.2.2 Registration.....	12
3.2.3 Manage Info.....	13
3.2.4 Patient history	14
3.2.5 Make Appointments	15
3.2.6 Approve Doctor	16
3.2.7 See Appointments.....	17
3.2.8 Approve Info, Tutorials, Therapy.....	18
3.2.9 Set counselling hour	19
3.2.10 Share info, tutorials and therapy.....	20
3.2.11 See Info, tutorial and therapy	20
3.2.12 Evaluate problem.....	21
3.2.13 Manage Profile	22
3.2.14 live communications.....	23
3.3 Activity Diagram.....	24
3.3.1 Login.....	24
3.3.2 Registration.....	25
3.3.3 Add Information	26
3.3.4 Add Therapy/Tutorials	27
3.3.5 Appointment List.....	28
.....	28
3.3.6 Approve Request and info	28
3.3.7 Make Appointment	29
3.3.8 Doctor List.....	30
.....	30
3.3.9 See Info/Tutorials	30
3.3.10 Patient History	31
3.3.11 Set Counselling Hour.....	32
3.3.12 Live Communications	33
3.3.13 Evaluate Problem.....	34
3.4 Sequence Diagram.....	35
3.4.1 Login.....	35

3.4.2 Registration.....	36
3.4.3 Approve Doctor	37
3.4.4 Appointment Lists	37
3.4.5 Make Appointment	38
3.4.6 Approve Info/Tutorials	38
3.4.7 Doctor List.....	39
3.4.8 See Post Tutorials/Therapy.....	39
3.4.9 Get Info.....	40
3.4.10 Manage Profile	41
3.4.11 Communication	42
3.4.12 See previous history.....	43
4. CHAPTER-04 SYSTEM DESIGN SPECIFICATION	44
4.1 Entity Relationship Diagram	44
4.2 ERD.....	45
4.3 Development Tolls and Technology	46
4.3.1 User Interface Technology	46
4.3.2 Implementation tools and platform	46
5. CHAPTER-05 SYSTEM TESTING.....	47
5.1 Feature to be Tasted	47
5.2 Testing Strategies	47
5.2.1 Test Approach.....	47
5.2.2 Pass or Fail Criteria	47
5.3 Testing Schedule	47
5.4 Taste Case	48
5.4.1 Taste Case-01	48
5.4.2 Taste Case-02	48
5.4.3 Taste Case-03	49
5.4.4 Taste Case-04	49
5.4.5 Taste Case-05	50
5.4.6 Taste Case-06	50
5.4.7 Taste Case-07	51
5.4.8 Taste Case-08	51
5.4.9 Taste Case-09	52
5.4.10 Taste Case-10	52
5.4.11 Taste Case-11	53

5.5 Test Environment (Browser)	54
6. CHAPTER-06 USER INTERFACE.....	55
6.1 User/Patient Panel	55
6.1.1 User Landing Page.....	55
6.1.2 Make an Appointment	56
6.1.3 Browse Blogs and info	56
6.1.4 Browse Blogs and info	57
6.1.5 Appointment confirmation mail	57
6.1.6 Prescription copy mail	58
6.1.7 Check History	58
6.1.8 Survey.....	59
6.2 Doctor Panel.....	59
6.2.1 Doctor Dashboard.....	59
6.2.2 Doctor Profile	60
6.2.3 Doctor Edit Profile.....	60
6.2.4 Add Education	61
6.2.5 Add Schedule.....	61
6.2.6 Add Experience	62
6.2.7 Appointment List.....	62
6.2.8 Doctor Experience	63
6.2.9 View Education	63
6.2.10 View Schedule.....	64
6.2.11 Write Prescription.....	64
6.2.12 Add post.....	65
6.2.13 Login.....	66
6.2.14 Registration.....	66
6.3 Admin panel	67
6.3.1 Admin Dashboard.....	67
6.3.2 Admin Profile	67
6.3.3 Admin Edit profile.....	68
6.3.4 Blog Request.....	68
6.3.5 Doctor Request	69
6.3.6 Current Doctor	69
6.3.7 Current Blog	70
7. CHAPTER-07 PEOJECT SUMMARY.....	71

7.1 Limitations	71
7.2 GitHub Link:	71
7.3 Future Work Plan	71
APENDIX A	72
APPENDIX B	89
References.....	95

1. CHAPTER-1 INTRODUCTION

1.1 Project Overview

“E-Psychologist” is an E-Doctor site, where people can meet up with a Psychiatrist. Here patients can get appointments of the doctors through the internet and meet up with them. Besides, in this system there is information that is much more psychological. Mainly it is a live E-communication system between the doctors and the patients. This will made treatments too much easier. In addition, there are many more activities. In this system, the patients will take appointments from the doctors. There will be an appointment request sent to the doctor. The doctor will see the appointment request and confirm or decline it. Does not matter whether it is accepted or rejected there will be an email sent to the patients email which will be a must give information to send the request. If the doctor approves the request of the patients then there will be made a very helpful connection by the system. The patients will get opportunities to meet will the doctor. If the patient is far away from the doctor or he cannot meet with the doctor live then the system will arrange a video conference between the patient and doctor. The doctors can post medical information, therapy and tutorials, which will be verified by the admins or head of doctor. Then the posts, will be uploaded in the system. If there is any past medical reports of the patients then patients will be able to see them using their email address.

1.2 Project Purpose

1.2.1 Background

This project is designed to provide psychological medical services to the patients whom are looking for quality medical treatments. The purpose of this project is to make psychological medical services reachable for all the patients. The patients whom are living in rural areas can get quality psychological supports by sitting in their respective places. Necessary Psychological details are not always gettable all together. This project will help patients to get all their necessary psychological information together at a time from on platform.

1.2.2 Advantages of users

- Patients can get appointments of quality doctors easily
- Patient can get medical services from anywhere
- A very helpful connection will be made between the doctors and the patient
- Posts from the doctors will be verified by the head of doctor
- Patients can get psychological treatment from doctors
- Patient can see his past medical histories
- Doctors can share their research information in the system

1.2.3 Goals

- Assure Psychological medical services for everybody
- Assure user-friendly interface
- Create a very close connection between the doctors and the patients
- Ensure quality psychological information

1.3 Stakeholders

- ❖ Admin
- ❖ Head of Doctor
- ❖ Doctor
- ❖ Patient

1.4 Proposed System Model (Block Diagram)

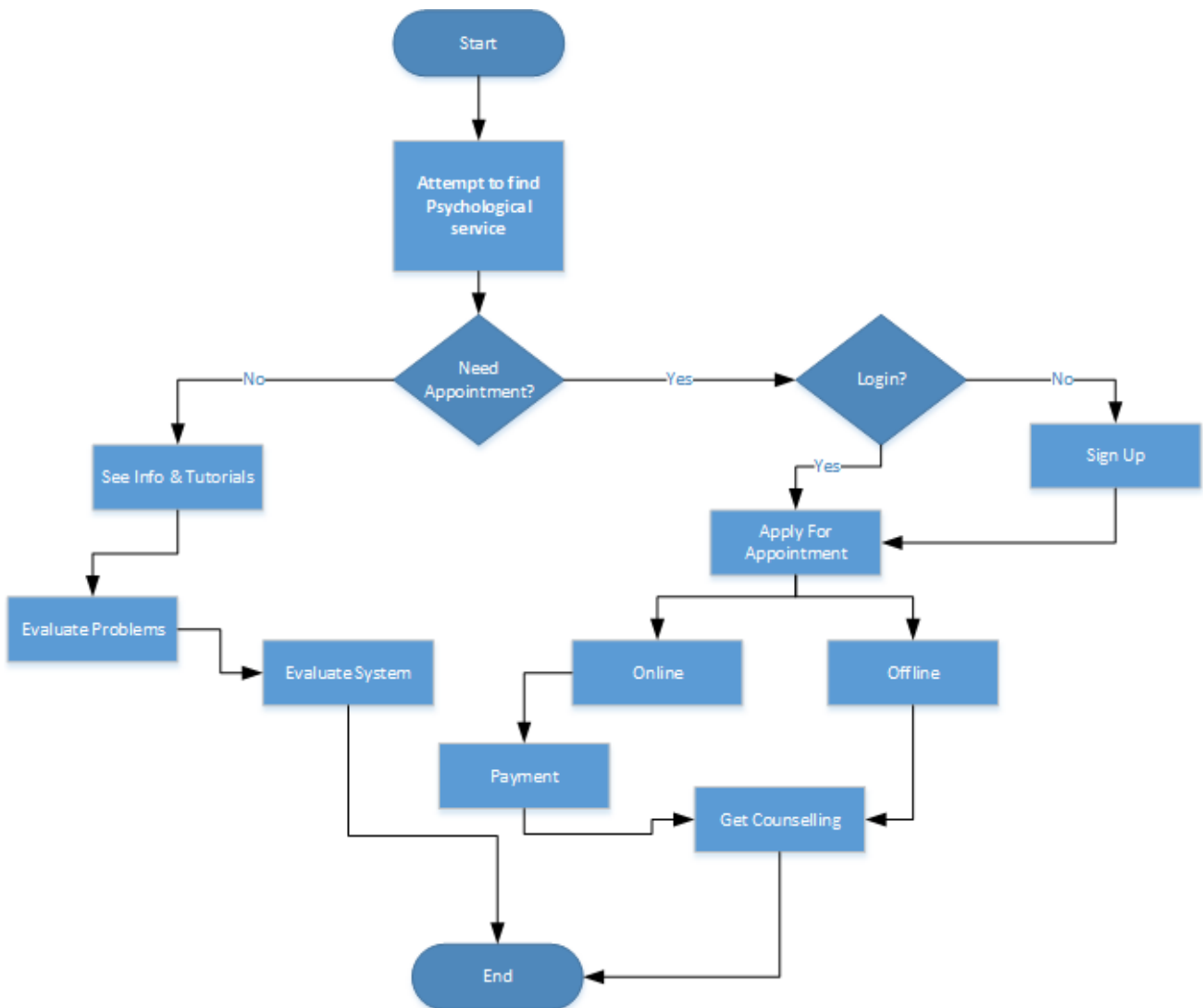


Figure 1.1 Block Diagram

1.5 Project Schedule

1.5.1 Gantt Chart



Figure 1.2 Gantt chart

Project Proposal	Maheen Absar	5 Days
Software Requirement Specification	Maheen Absar	15 Days
Software Design	Maheen Absar	15 Days
Coding	Maheen Absar	70 Days
Software Testing	Maheen Absar	35 Days
Project Finalization	Maheen Absar	35 Days

2. CHAPTER-2 SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requirements

Table 2.1 Functional Requirements

ID	Name	Description	Stakeholder
FRQ-1	Can Create Doctor Profile	This form build for Doctor. Where doctor gives key information like personal biological information, educational, experience etc. Which can be used to manipulate the profile later if necessary	Doctor, Admin(Head of Doctor)
FRQ-2	Can Approve doctor enroll request	In this section, the admin or head of doctor will see the list of doctors registered in the main system and will allow the doctors to use the system as required	Admin or Head of Doctor
FRQ-3	Can manage doctor	This form is only boiled for head of doctor (Admin). He will be able to allow doctors to use the system as needed	Admin or Head of Doctor
FRQ-4	Can add appointment schedule	Doctors will be able to schedule appointments according to their schedule. Which will show up in the appointment list of patients in the main system	Doctor
FRQ-5	Can take up patients	All the appointments will be seen on the dashboard of the doctor who will make the appointment. Doctors will be able to take them up sequentially	Doctor
FRQ-6	Can prescribed patients	After taking up the patient's appointment, the doctor will be able to give the prescription to the patient through online	Doctor
FRQ-7	Prescription will send through mail	After the doctor gives the prescription to the patient, the copy of the prescription will be automatically emailed to the patient from the system. This will help the patient to get the prescription easily	System
FRQ-8	Can see all patient history	Doctor can easily see all the information of the patients taken up by the doctor in the doctor dashboard	Doctor, Admin

FRQ-9	Can post Information	Doctor can write an informative article and post it to the system	Doctor
FRQ-10	Can post Therapy	Doctor can upload various kind of mental therapy and post it to the system	Doctor
FRQ-11	Can post Tutorial	Doctor can upload various kind of mental therapy and tutorial link to the system	Doctor
FRQ-12	Can connect with video call	Doctor can send a video connectivity link to the patient and patient and doctor can meet up through online	Doctor, Patient
FRQ-13	Can approve doctor	The list of doctors who will register to enter the system will be displayed in the admin or head of the doctor's dashboard. They will check the qualifications from here and allow the doctors to enter the system	Admin or Head of Doctor
FRQ-14	Can approve posted information	The admin or the head of the doctor will give permission to select the information that the doctors will post and upload it to the system	Admin or Head of Doctor
FRQ-15	Can approve posted tutorial	The admin or the head of the doctor will give permission to select the tutorial that the doctors will post and upload it to the system	Admin or Head of Doctor
FRQ-16	Can approve posted therapy	The admin or the head of the doctor will give permission to select the therapy that the doctors will post and upload it to the system	Admin or Head of Doctor
FRQ-17	Can manage posts	The admin or the head of the doctor will manage all posts	Admin or Head of Doctor
FRQ-18	Can take appointments	Patients will be able to make an appointment with the doctor of their choice via email only without the hassle of any login registration	Patient
FRQ-19	Get mail confirmation	After making an appointment, patient will receive automatic mail confirmation from the system. Appointment will be confirmed through confirmation	Patient
FRQ-20	Can browse all information	Users will be able to browse all the information in the system without any hassle	Patient or User

FRQ-21	Can browse all tutorial	Users will be able to browse the entire tutorial in the system without any hassle	Patient or User
FRQ-22	Can browse all therapy	Users will be able to browse all the therapy in the system without any hassle	Patient or User
FRQ-23	Can browse all doctor profile	Users will be able to browse all the doctor profile in the system without any hassle	Patient or User
FRQ-24	Can see his appointment history	Patients will be able to view their previous history by providing their email address, using that email address to see the details of the treatment they received. User verification will be done through OTP	Patient
FRQ-25	Can join survey	Anyone who joins its survey can get an idea of his or her primary disease. The system will suggest a doctor according to the prognosis	Patient
FRQ-26	Login	Doctor, Admin and Head of doctor can login	
FRQ-27	Registration	Doctor, Admin and Head of doctor can login	
FRQ-28	Tress history	Patient or User can tress his/her treatment history	

2.2 Non-Functional Requirements

Table 2.2 Non-Functional Requirements

ID	Name	Description	Priority
NFR-1	Security	There developed multi authentication system and also using 2 factor verification through email OTP which gives strong security	High
NFR-2	Availability	The system is available for 24/7	Medium
NFR-3	Accuracy	Trying to maintain highest accuracy to developed this system	High

NFR-4	Maintenance	The participant would benefit from this system's user-friendliness. They can quickly navigate to the resources they need.	Medium
-------	-------------	---	--------

2.3 Performance requirements

2.3.1 Speed and Latency requirements

- Required data set has been maintained through the MYSQL using Laravel artisan command like php artisan migrate
- When upgrading the last migration log, I've done the rollback –step=1 process a few times, and it gives me a lot of speed in maintaining the databases.

2.3.2 Accuracy Requirements

- The system is developed to ensure maximum accuracy. All of the patients and doctors of this system can operate and get the result according to their needs
- If they want to update anything in their information, they can do it in no time.

2.3.3 Capacity requirements

- This system can operate with the patients and the doctors at the same time
- The doctor can get connected with patient by using this system

2.4 Dependability Requirements

- If user wants to manage profile, he has to log in
- Admin filters the post which was posted by doctor
- Admin filters the doctor joining request

2.5 Maintainability and supportability requirements

2.5.1 Maintenance Requirements

- This system is maintained by the authority.
- Without the permission of the authority/admin panel no body will be able to get access of this system
- The whole system is inquired by the authority
- Permission can be modified anytime

2.5.2 Supportability Requirements

- If users or doctors face any problem with the system, they will report accordingly
- The admin panel will receive alert notification from the system if anyone tries to access the system

2.5.3 Adaptability Requirements

The patients and doctors will be able to receive quality services from this system at any time. They will face no problem in doing this. Because this system has been developed with the highest concern of the user's perspective.

2.6 Security Requirements

2.6.1 Access Requirement

- Only patients whose appointment request is accepted can get prescribed
- Only approved doctor can use dashboard
- Only approved doctor can take up patients an post article
- Guest user allowed to browse site and gather information

2.6.2 Integrity Requirement

- This system has been navigated by the employer, candidate, agent and super admin

2.6.3 Privacy Requirements

- All the personal information of the doctors and the patients are always secured in this system.

2.6.4 Authentication Requirements

- All the passwords and sensitive information's are being hashed

2.6.5 Accessibility Requirements

- Since this is an web based application it is device independent.
- The application can be accessed 24/7 from anywhere with any device

2.7 User Friendliness Requirements

- The font type of this system is standard
- Suitable color combination
- Effective and smooth content
- Easy navigation
- Tooltip text for user guidance
- Responsive

2.8 Operational and Environmental Requirements

This requirement focuses on how the doctor, patient, super admin is going to operate and the navigate the system like online service, including interfaces and interoperability with other systems as well. The requirements establish how well and efficient way has been followed.

2.8.1 Release Requirements

There are no clear release conditions, but it is often dependent on the expectations of the applicant. The project will be published with as little change as possible to the version.

2.9 Legal Requirements

These provisions take into account any violations of laws and regulations, as well as the rules should be enforced in order to keep the system running.

In this system, there is no violation. All the users can use it easily and perform their activity.

2.10 Standard Requirements

There are also several standard requirements

Table 2.3 Standard Requirements

01	Doctors Quality
Description	Considered as high qualification
Stakeholder	Doctors

3. CHAPTER-03 SYSTEM ANALYSIS

3.1 Use Case

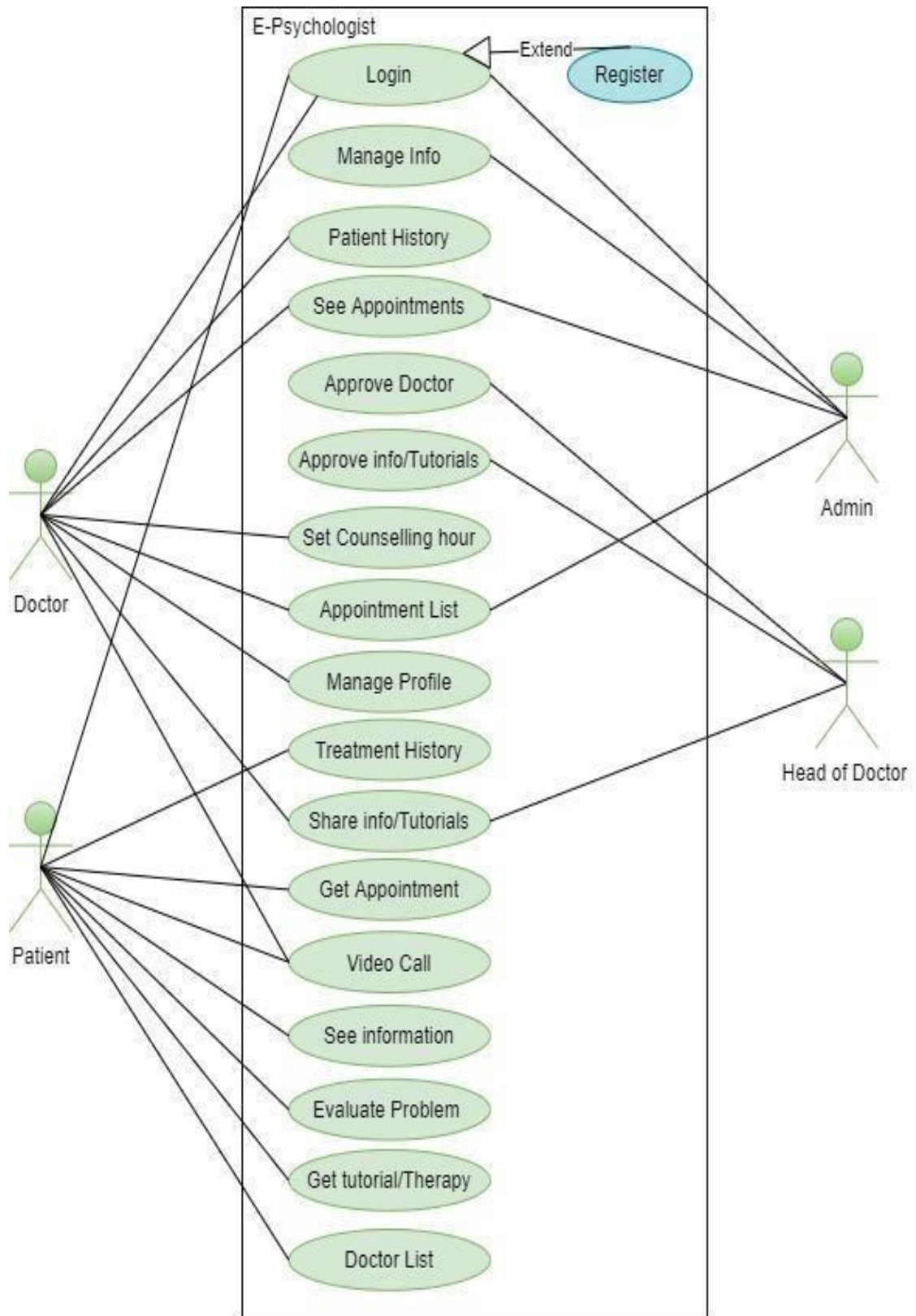


Figure 3.1 Use Case

3.2 Use Case Description

3.2.1 Login

Table 3.1 Use Case Description Login

Use Case	Login	
Goal	Doctor, Head of Doctor and Admin can get access to the system through this login system and get some credential resources	
Preconditions	N/A	
Success End Condition	Doctor, Head of Doctor and Admin can get into the system successfully. After that they perform their action as expected successfully.	
Failed End Condition	Doctor, Head of Doctor and Admin cannot get into the system. After that they could not perform their action as expected successfully	
Primary Actors:	Doctor, Head of Doctor and Admin	
Secondary Actors:	Head of Doctor and Admin	
Trigger	Login request comes in	
Description/main Success Scenario	Step	Action
	1	User gives the required credentials and hit the login button to get the access into the system
	2	Credentials include an email address and a password that must suit. If the matches are found, the login will be successful; if not, they will be redirected to their login tab.
Alternative Flows	Step	Branching Action
	1a	The user inputs their login information, which includes their email address and password. Obviously, the password must be saved in an encrypted format.
Quality Requirements	Step	Requirement
	1	Requirements are including email and password for completion of the whole process.

3.2.2 Registration

Table 3.2 Use Case Description Registration

Use Case	Registration	
Goal	Doctors are going to give their required information such as skill, interest, bio etc. for completion of the registration process.	
Preconditions	N/A	
Success End Condition	The user enters the required information and presses enter, and the registration is completed successfully and without errors. They will then proceed to their login tab.	
Failed End Condition	User gives the required information, hit enter, and then the registration is not completed. The view page appears with an error.	
Primary Actors:	Doctor	
Secondary Actors:	Head of Doctor and Admin	
Trigger	Registration request comes in	
Description/main Success Scenario	Step	Action
	1	User gives the required information and hit the registration button to get the access into the login system
	2	Credentials can include ability, bio, and contact information, which varies from user to user. If all is in order, their registration will be successful; if not, they will be redirected to their sign up page with an error message.
Alternative Flows	Step	Branching Action
	1a	The user enters their sensitive and accurate information including skill, bio, interests, about etc. then again hit enter to complete the registration process successfully.
Quality Requirements	Step	Requirement
	1	Requirements are including skill, bio, about, contact etc. for completion of the whole process.

3.2.3 Manage Info

Table 3.3 Use Case Description Manage Info

Use Case	Manage Info	
Goal	Admin can manage entire information, tutorials and therapy by creating, updating and deleting their information.	
Preconditions	N/A	
Success End Condition	Admin select the information and hit the edit/add/delete function. After that, he can edit/add/delete information/tutorials, therapy successfully.	
Failed End Condition	After edit/delete/add If admin does not click Save Change then it will not work.	
Primary Actors:	Admin	
Secondary Actors:	N/A	
Trigger	Edit/delete/add request comes in	
Description/main Success Scenario	Step	Action
	1	User gives the required information and hit the save change button
	2	Credentials can edit/add/delete and click the save change button to perform action successfully. Otherwise the action will not take place.
Alternative Flows	Step	Branching Action
	1a	Sometimes he can be fallen into the trouble if he could not click the save change button properly. Therefore, he needs to provide accurate and required information to complete this process successfully.
Quality Requirements	Step	Requirement
	1	Requirements are including edit, delete, and add.

3.2.4 Patient history

Table 3.4 Use Case Description Patient History

Use Case	Patient history	
Goal	Doctors can see all the patients' histories and patients can see only their histories.	
Preconditions	Patients have to have old medical histories.	
Success End Condition	The doctors and patients can watch the histories.	
Failed End Condition	Doctors and patients are trying to inquire the histories but it is not there. In this case, the system will show nothing.	
Primary Actors:	Doctor, Patient	
Secondary Actors:	N/A	
Trigger	History request comes in	
Description/main Success Scenario	Step	Action
	1	Doctors click the history button to inquire histories.
	2	If patient's email is wrong then there will be nothing to show.
Alternative Flows	Step	Branching Action
	1a	The doctors have to click the history button properly. Patients have use accurate email address.
Quality Requirements	Step	Requirement
	1	Email must be correct.

3.2.5 Make Appointments

Table 3.5 Use Case Description Make Appointment

Use Case	Make Appointments	
Goal	Patients can choose a doctor according to their needs and take an appointment	
Preconditions	Patient must have an correct email address	
Success End Condition	After giving all the information's patients click the make appointment button and confirmation message sent to the email.	
Failed End Condition	If email address is wrong no appointment will be made.	
Primary Actors:	Patient	
Secondary Actors:	N/A	
Trigger	Appointment request comes in	
Description/main Success Scenario	Step	Action
	1	User gives the required information and hits the make appointment button.
	2	Credentials can include patient name, email, address, cell no, selected doctor and schedule. Otherwise appointment will not be made.
Alternative Flows	Step	Branching Action
	1a	The user enters their sensitive and accurate information including email, name, doctor etc. to successfully make an appointment
Quality Requirements	Step	Requirement
	1	Requirements are including email, name, doctor, schedule etc. for completion of the whole process.

3.2.6 Approve Doctor

Table 3.6 Use Case Description Approve Doctor

Use Case	Approve Doctor	
Goal	Doctors will register by giving their information's. The admin or head of doctor will verify the information's given by the doctor and approve.	
Preconditions	Doctor have to register	
Success End Condition	After approving the doctor can enter to the dashboard.	
Failed End Condition	If the request is denied by the admin then the doctor cannot enter the dashboard	
Primary Actors:	Admin/Head of Doctor	
Secondary Actors:	Doctor	
Trigger	Approve request comes in	
Description/main Success Scenario	Step	Action
	1	Doctor gives the required information and hit the Register button to get the access into the approve system
	2	Credentials can include ability, bio, and contact information, which varies from user to user. If all is in order, their registration will be successful; if not, they will be redirected to their sign up page with an error message.
Alternative Flows	Step	Branching Action
	1a	The user enters their sensitive and accurate information including skill, bio, interests, about etc. then again hit enter to complete the registration process successfully.
Quality Requirements	Step	Requirement
	1	Requirements are including skill, bio, about, contact etc. for completion of the whole process.

3.2.7 See Appointments

Table 3.7 Use Case Description See Appointments

Use Case	See Appointments	
Goal	The doctor and the admin can see the appointment list of the patients	
Preconditions	There must be appointments	
Success End Condition	Doctors and admins can see the appointments by clicking the appointment list button.	
Failed End Condition	If there is no appointment there will be shown nothing.	
Primary Actors:	Doctor, Head of Doctor and Admin	
Secondary Actors:	Patient	
Trigger	Appointment list request comes in	
Description/main Success Scenario	Step	Action
	1	User must log in
	2	After clicking the appointment list button they will see the list
Alternative Flows	Step	Branching Action
	1a	User must click the appointment list button
Quality Requirements	Step	Requirement
	1	N/A

3.2.8 Approve Info, Tutorials, Therapy

Table 3.8 Use Case Description Approve Info, Tutorials, Therapy

Use Case	Approve Info, Tutorials, Therapy	
Goal	Doctor will post info, tutorials and therapy. Then the admin/head of doctor will sort out and decide whether to approve or deny	
Preconditions	Must post info, Tutorials, Therapy	
Success End Condition	After approving the info, tutorials, therapy will be shown in the system	
Failed End Condition	If it is denied then info, tutorials, therapy will not be shown in the system	
Primary Actors:	Doctor, Head of Doctor and Admin	
Secondary Actors:	Doctor	
Trigger	info, tutorials, therapy approve request comes in	
Description/main Success Scenario	Step	Action
	1	Admin log in to the system dashboard and see that requests came in
	2	After verifying he can approve or deny
Alternative Flows	Step	Branching Action
	1a	The doctor must represent the info, tutorials, therapy properly
Quality Requirements	Step	Requirement
	1	Quality must be ensured

3.2.9 Set counselling hour

Table 3.9 Use Case Description Set Counselling hour

Use Case	Set counselling hour	
Goal	Doctor can set counselling hour according to his conveniently	
Preconditions	Doctor must be approved	
Success End Condition	After setting the counselling hour the patients can see the counselling hour and make appointment	
Failed End Condition	If the counselling hour is no set properly then the patients cannot make an appointment.	
Primary Actors:	Doctor	
Secondary Actors:	Patient	
Trigger	Counselling request comes in	
Description/main Success Scenario	Step	Action
	1	Doctor must be logged in
	2	By choosing proper time the doctor will set the counselling hour
Alternative Flows	Step	Branching Action
	1a	Time format
Quality Requirements	Step	Requirement
	1	Choosing proper time

3.2.10 Share info, tutorials and therapy

3.2.11 See Info, tutorial and therapy

Table 3.11 Use Case Description see Info, Tutorials, and Therapy

Use Case	See Info, tutorial and therapy	
Goal	User browse info, tutorials, and therapy from the system which is posted by doctors	
Preconditions	Must have internet connection	
Success End Condition	User browse info, tutorials, and therapy	
Failed End Condition	If internet connection getting lost then the user cannot brose those	
Primary Actors:	Patient or User	
Secondary Actors:	System	
Trigger	Browsing request comes in	
Description/main Success Scenario	Step	Action
	1	User enter to the site
	2	If everything going well, user can easily brose info, tutorials and therapy
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Requirements are including good internet connectivity

3.2.12 Evaluate problem

Table 3.12 Use Case Description Evaluate problem

Use Case	Evaluate problem	
Goal	User or Patient enter the site and joining the system survey. By answering some MCQ user can identify his problem as primary stage. System recommended doctor for user	
Preconditions	Must answer all MCQ	
Success End Condition	After answering all of them user can identify problem	
Failed End Condition	If user mistakenly missing some MCQ the system can't work	
Primary Actors:	User or Patient	
Secondary Actors:	System	
Trigger	Survey request comes in	
Description/main Success Scenario	Step	Action
	1	User enter the site and joining survey
	2	After providing all MCQ answer, the system identify the primary stage of problem and suggest doctor
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

3.2.13 Manage Profile

Table 3.13 Use Case Description Manage Profile

Use Case	Manage Profile	
Goal	Users can manage their profile by creating, updating and deleting their information.	
Preconditions	Must be logged in	
Success End Condition	User or Doctor gives all of the required information to create the profile and hit enter. After that he creates his profile successfully and get into their own panel successfully.	
Failed End Condition	User or Doctor gives all of the required information to create the profile and hit enter. After that he could not create his profile successfully and could not get into his own panel successfully.	
Primary Actors:	Doctor or User	
Secondary Actors:	System	
Trigger	Manage request comes in	
Description/main Success Scenario	Step	Action
	1	User gives the required information to create his profile. After he hits enter then the profile has been created successfully, after that, he gets into his panel. Later he or she can update and delete his/her profile and can do the others stuffs.
	2	After that user click to save change button to perform the action
Alternative Flows	Step	Branching Action
	1a	User should provide accurate information to success the action
Quality Requirements	Step	Requirement
	1	Requirements are including skill, bio, degree etc. for completion of the whole process.

3.2.14 live communications

Table 3.14 Use Case Description Live communication

Use Case	Live communications	
Goal	Build a communication network for doctor to patient	
Preconditions	Doctor must be logged in	
Success End Condition	After sending a link for making communication, the doctor and the patient will be connected	
Failed End Condition	If email is not correct, the communication will be not successful	
Primary Actors:	Doctor, Patient	
Secondary Actors:	System	
Trigger	Communication request comes in	
Description/main Success Scenario	Step	Action
	1	Doctor give the communication link to patient through email
	2	After that the patient open the link and start communicating
Alternative Flows	Step	Branching Action
	1a	The user use fake email
Quality Requirements	Step	Requirement
	1	Email must be correct

3.3 Activity Diagram

3.3.1 Login

Doctor/Admin Login

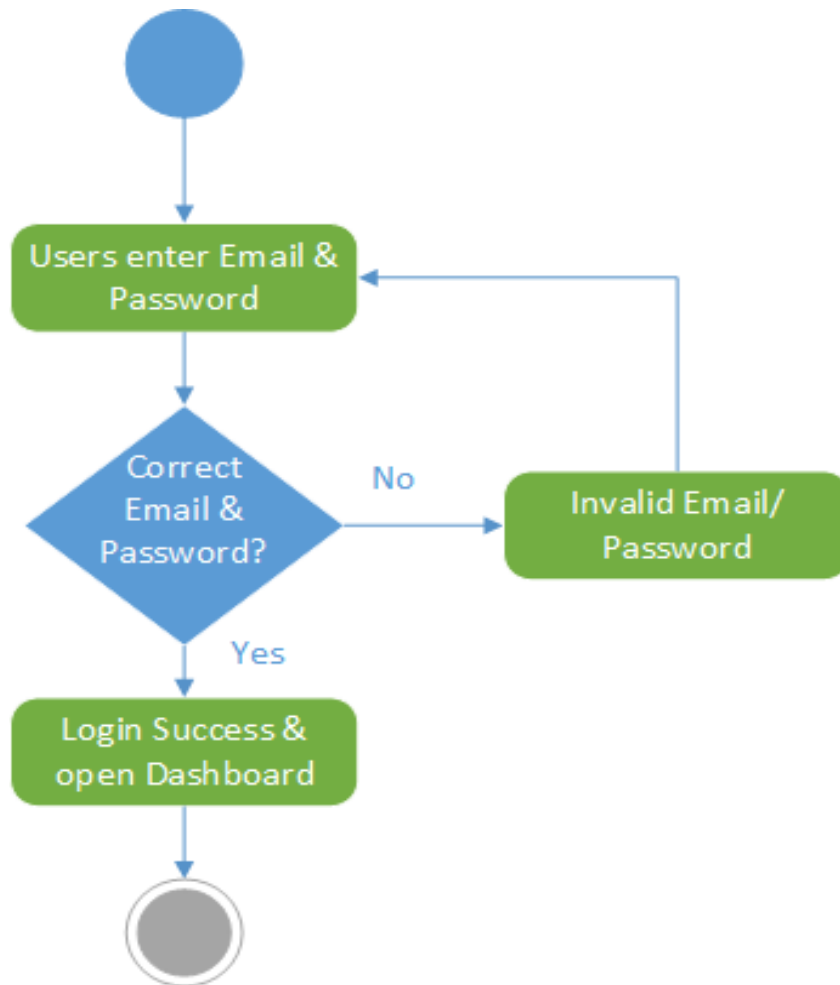


Figure 3.2 Activity (Login)

3.3.2 Registration

Doctor Registration

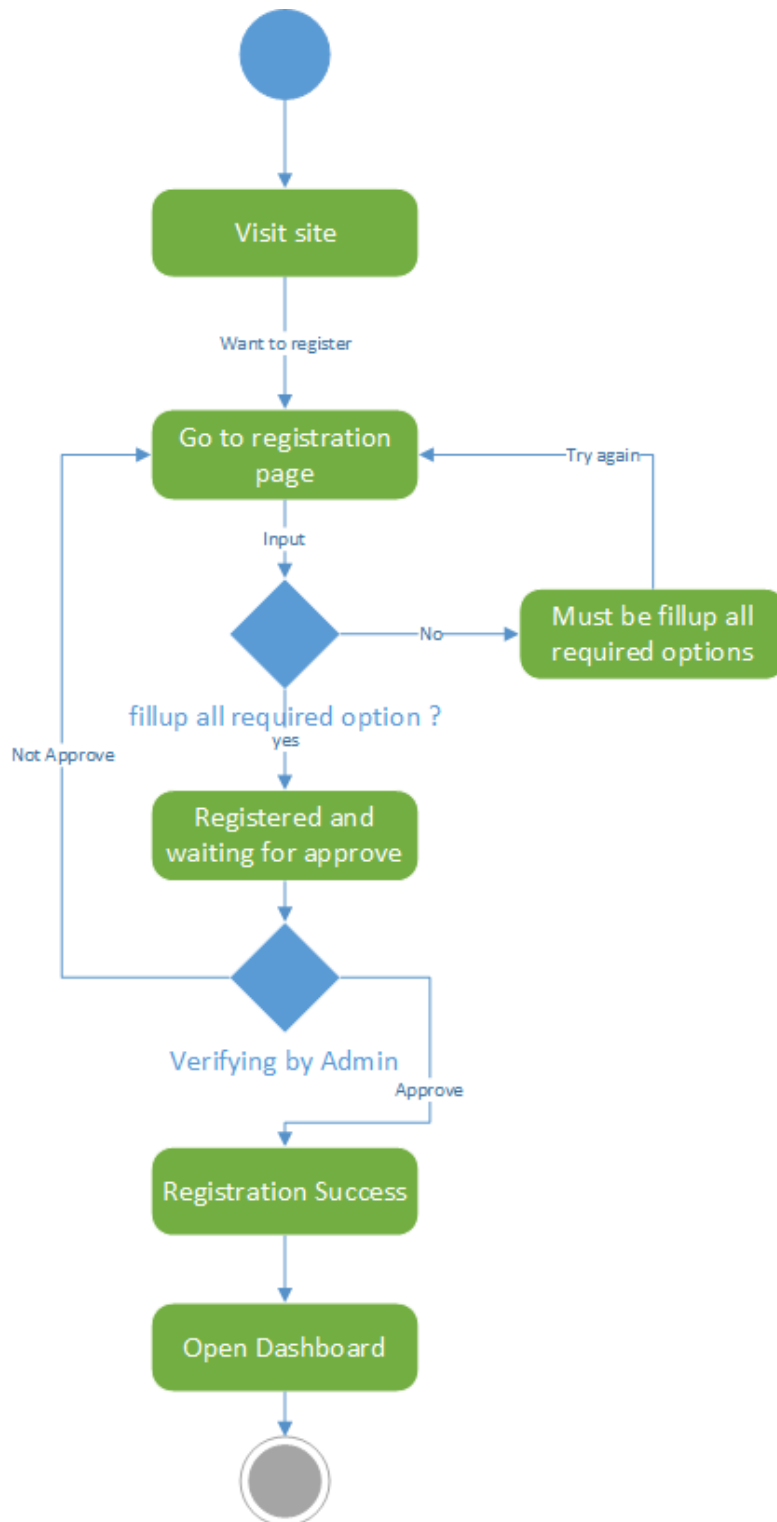


Figure 3.3 Activity (Registration)

3.3.3 Add Information

Doctor add info portion

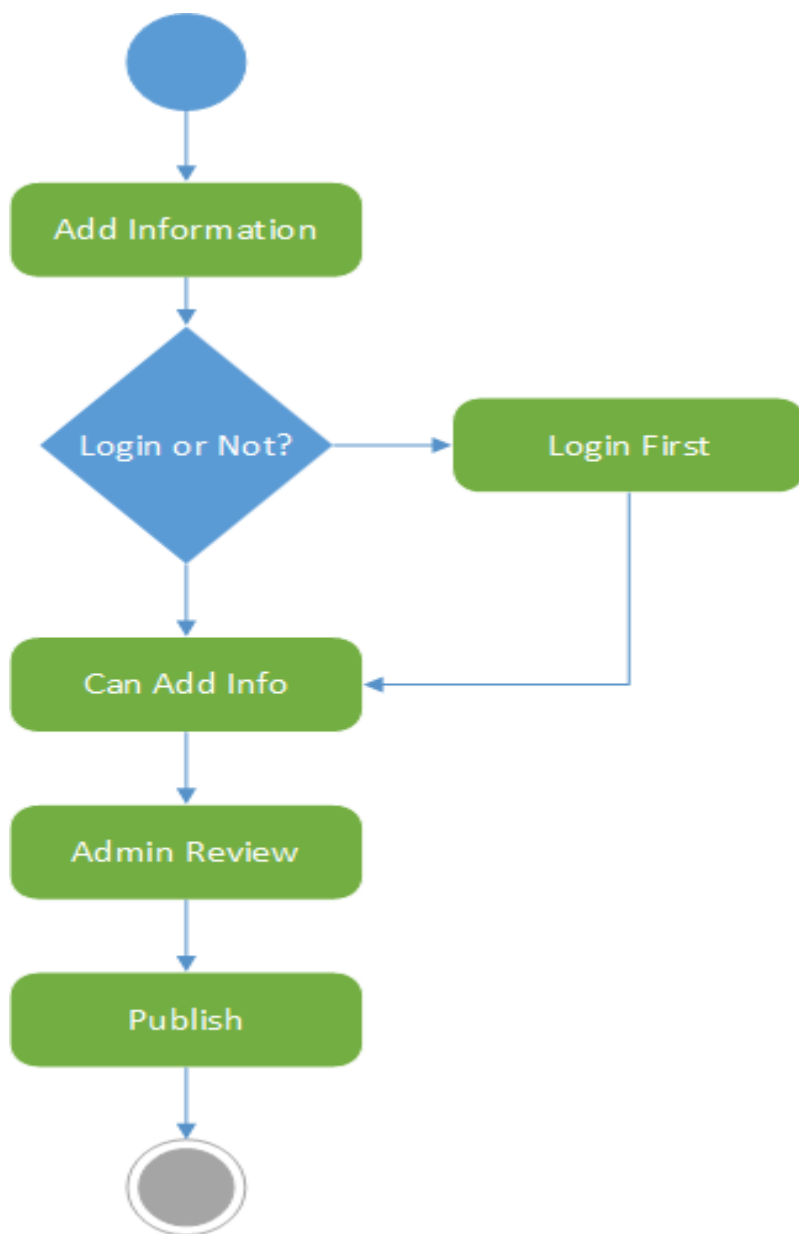


Figure 3.4 Activity (Add Info)

3.3.4 Add Therapy/Tutorials

Doctor add therapy/post portion

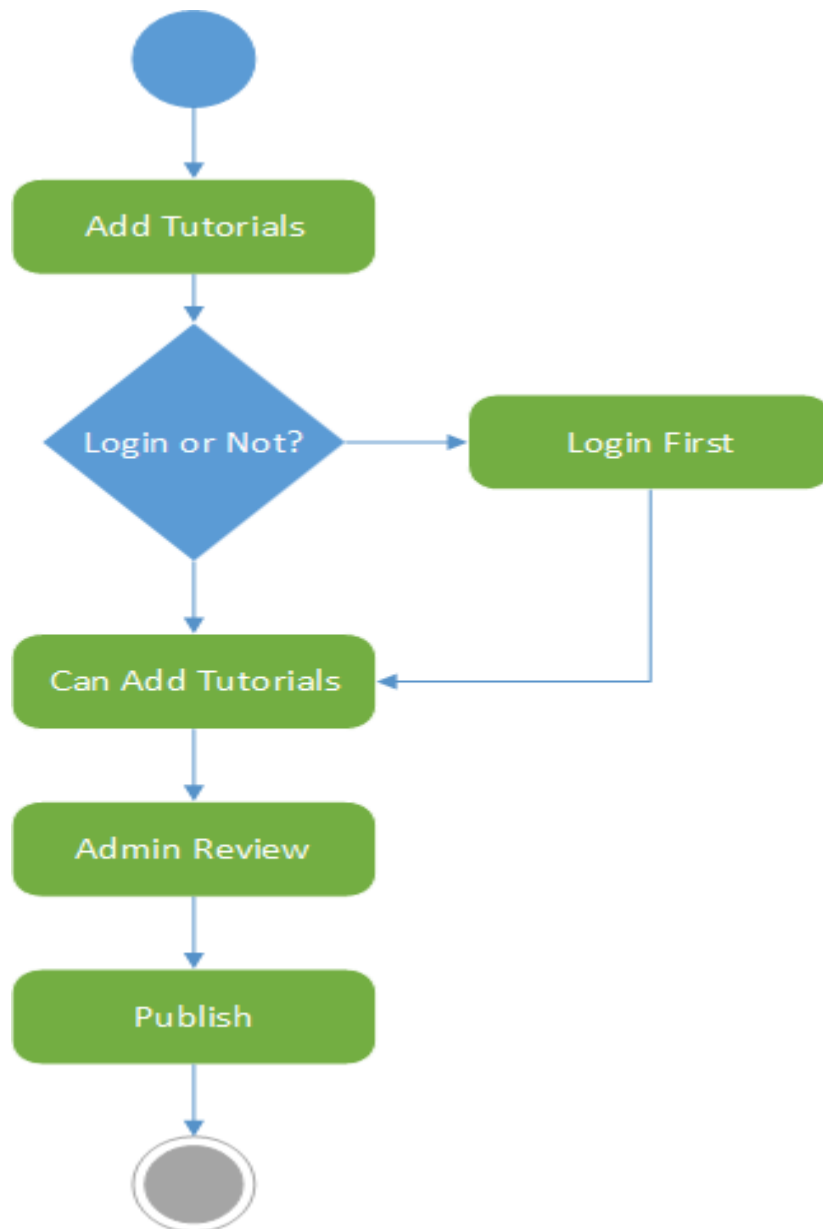


Figure 3.5 Activity (Therapy/Tutorials)

3.3.5 Appointment List

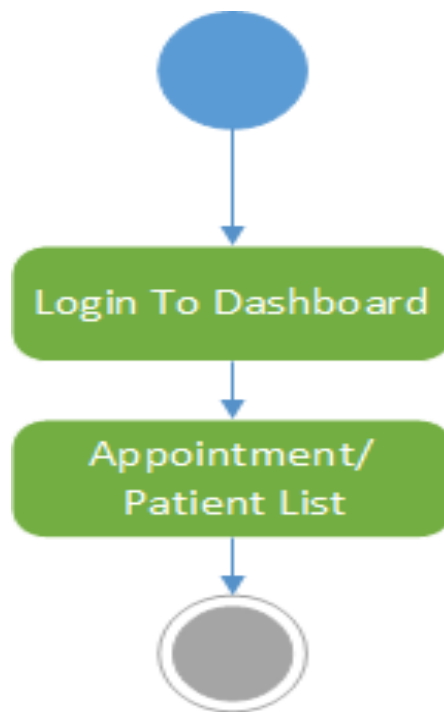


Figure 3.6 Activity (Appointment List)

3.3.6 Approve Request and info

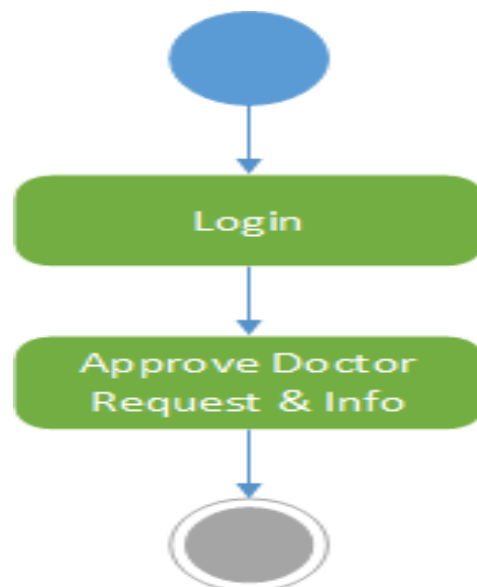


Figure 3.7 Activity (Approve Post Request)

3.3.7 Make Appointment

Patient make appointment section

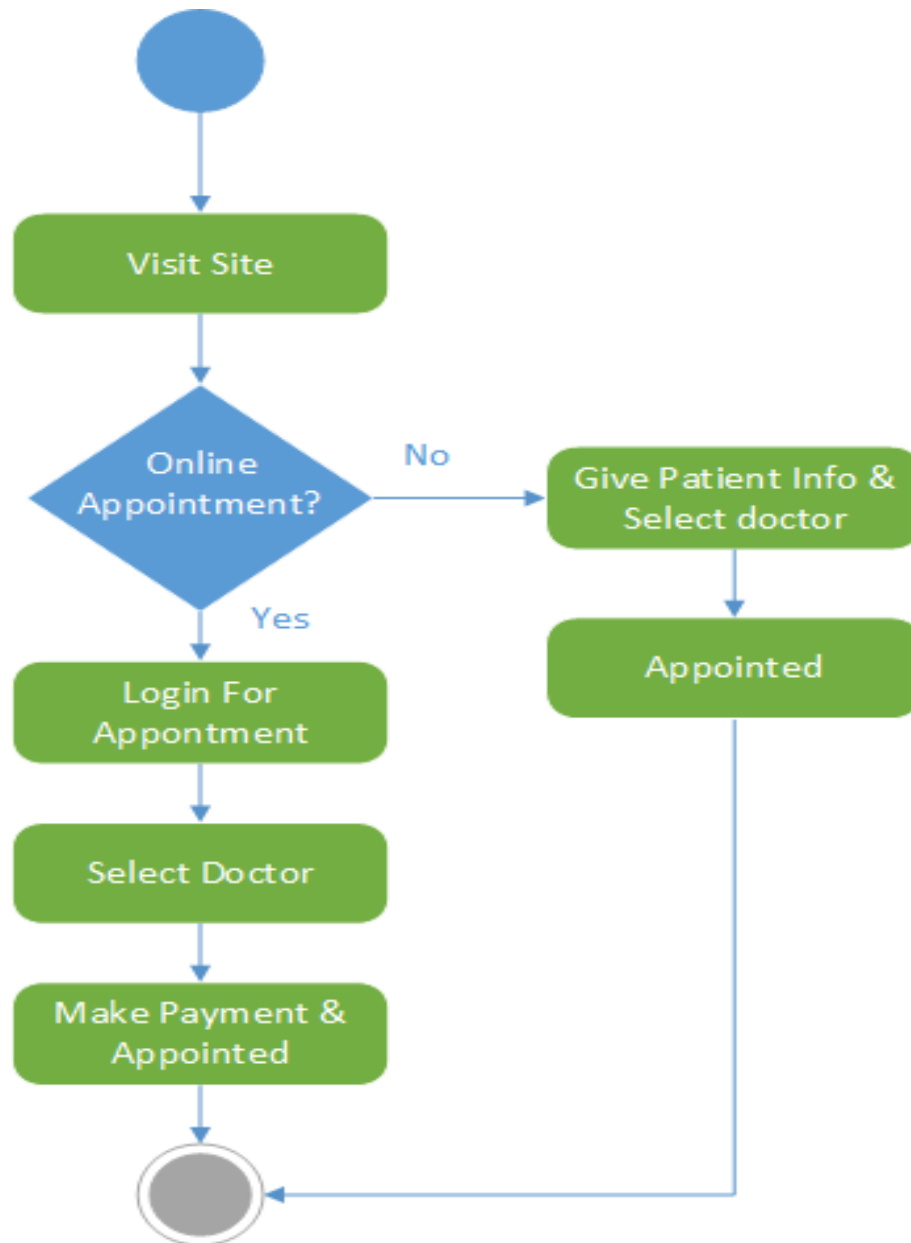


Figure 3.8 Activity (Make Appointment)

3.3.8 Doctor List

Patient see the doctor list

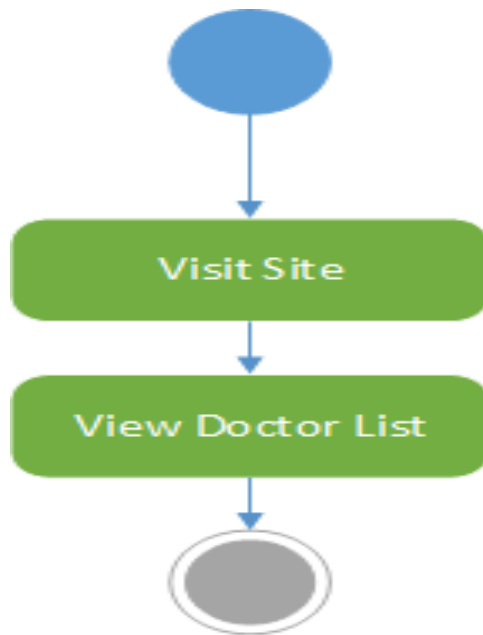


Figure 3.9 Activity (Doctor List)

3.3.9 See Info/Tutorials

Patient see the info



Figure 3.10 Activity (See Info)

3.3.10 Patient History

Doctor can see the patient previous history

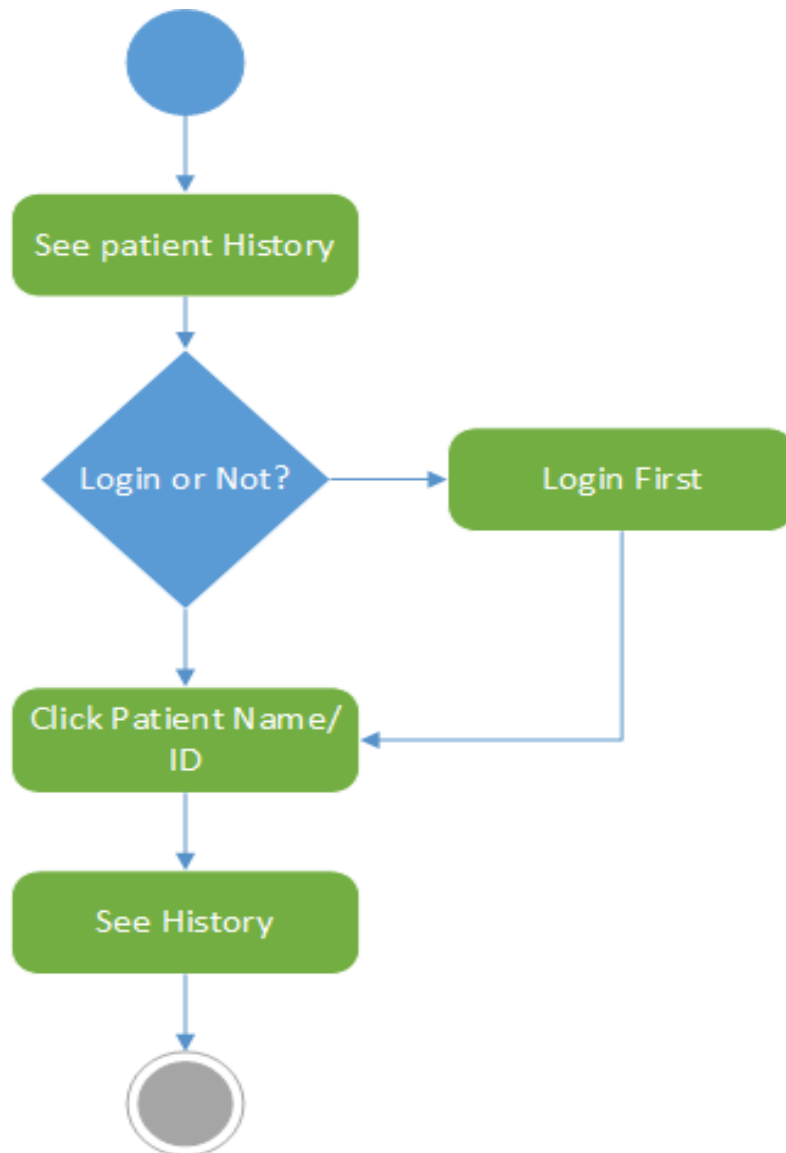


Figure 3.11 Activity (Patient History)

3.3.11 Set Counselling Hour

Doctor set counselling hour for take appointment

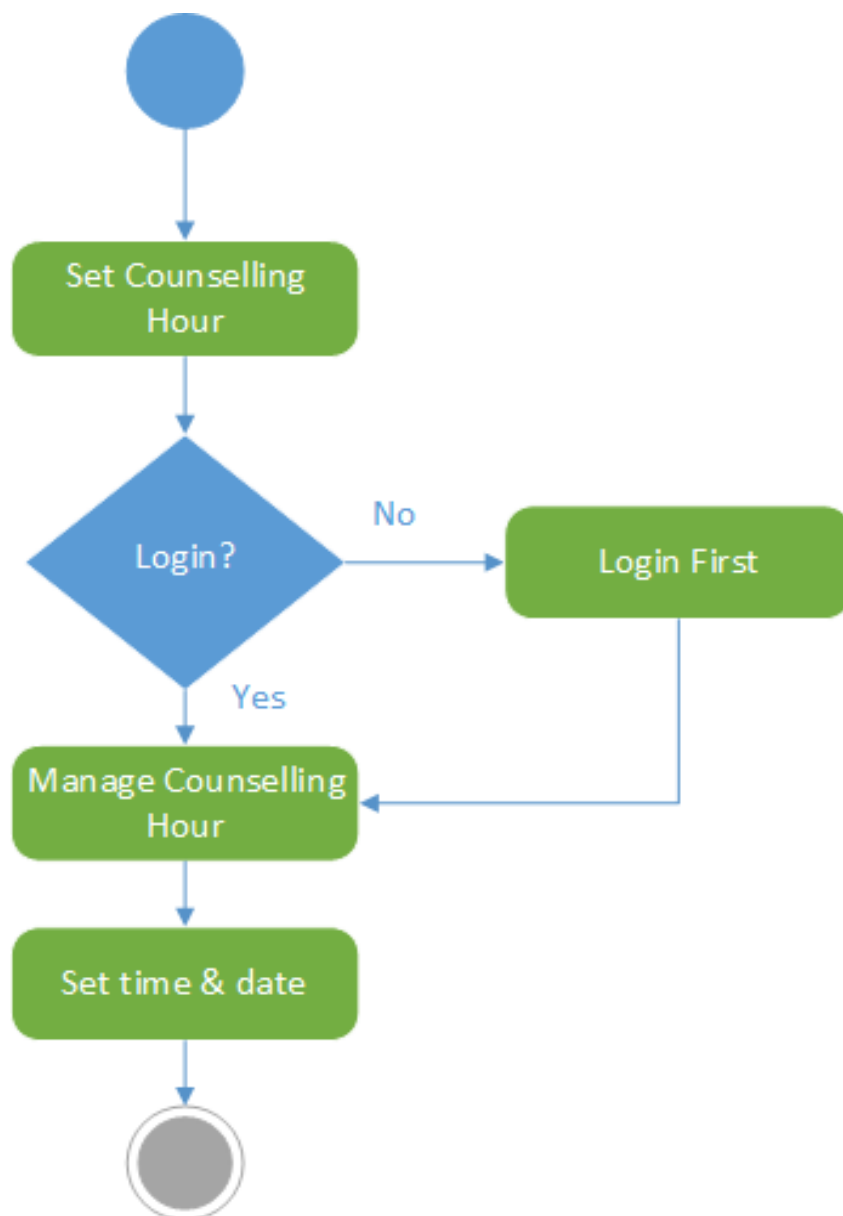


Figure 3.12 Activity (Set Counselling

3.3.12 Live Communications

This portion is for Live Communications for doctor and patient

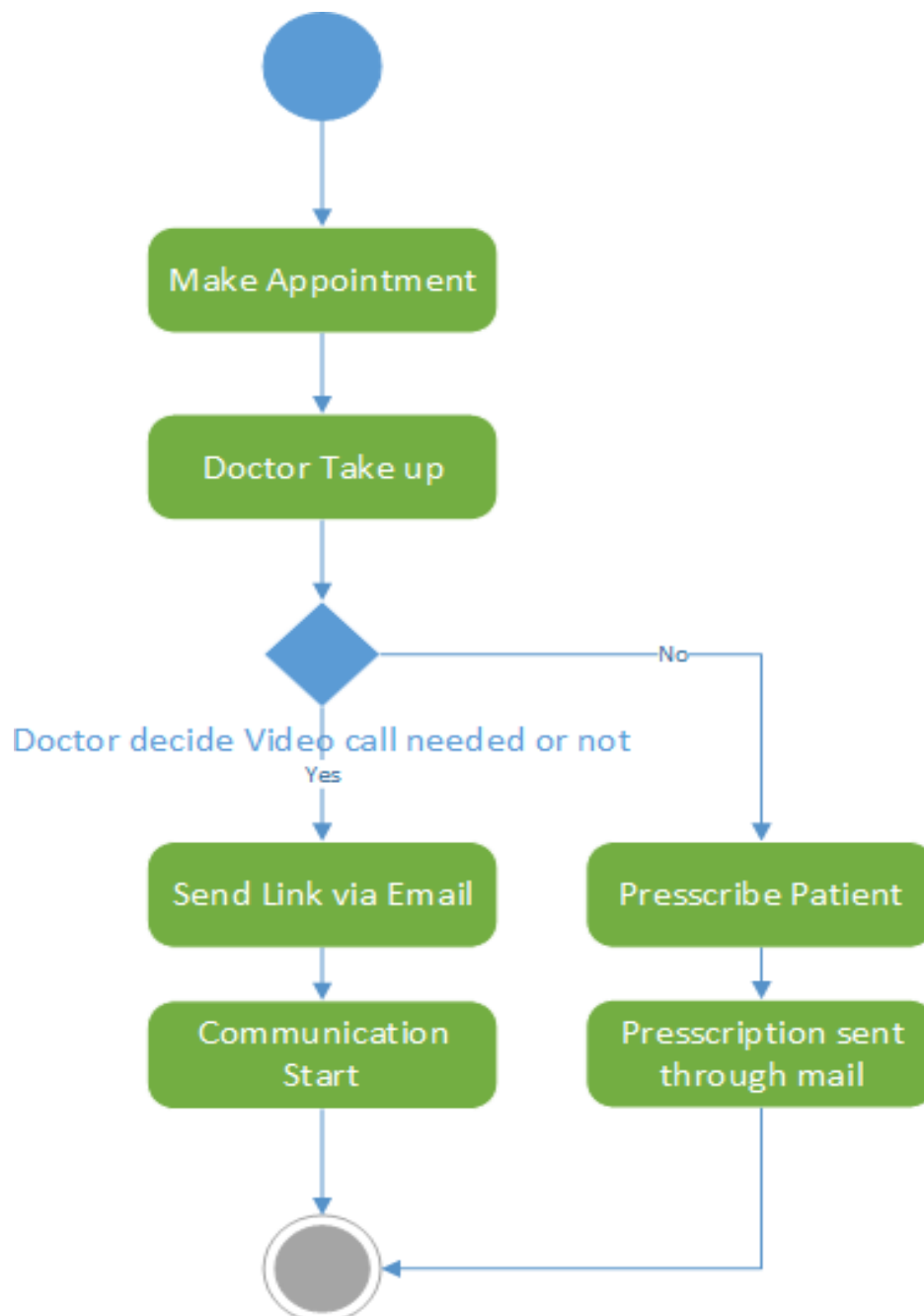


Figure 3.13 Activity (Live Communication)

3.3.13 Evaluate Problem

This portion is for online survey. Users get response by answering some MCQ

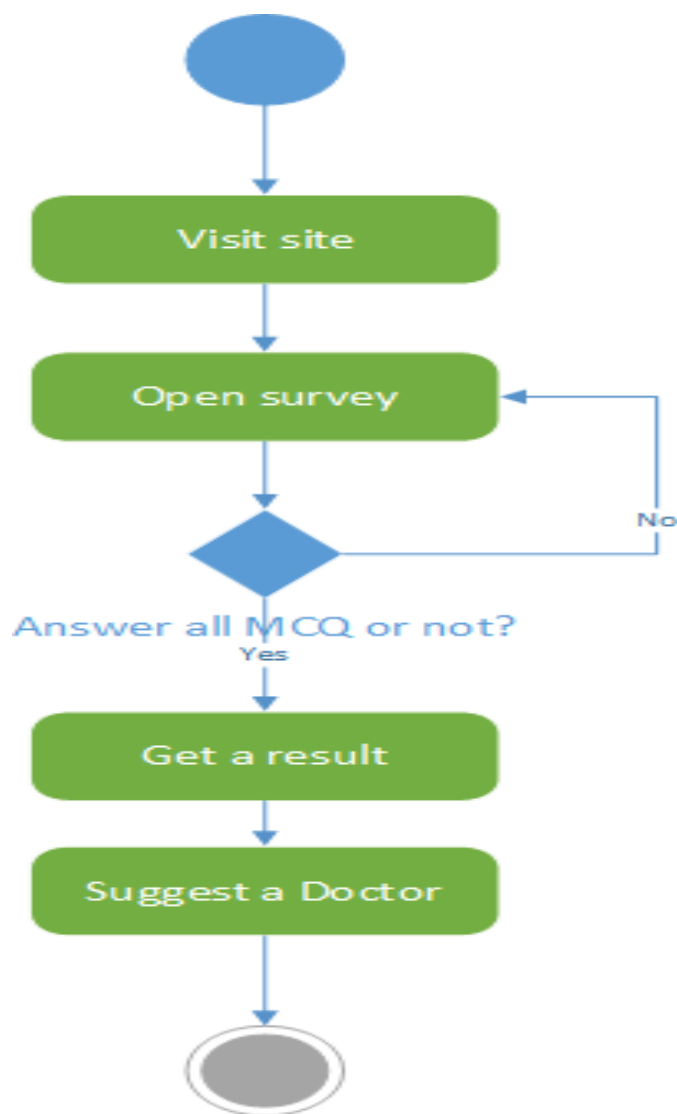


Figure 3.14 Activity (Evaluate Problem)

3.4 Sequence Diagram

3.4.1 Login

Doctor/Admin login section

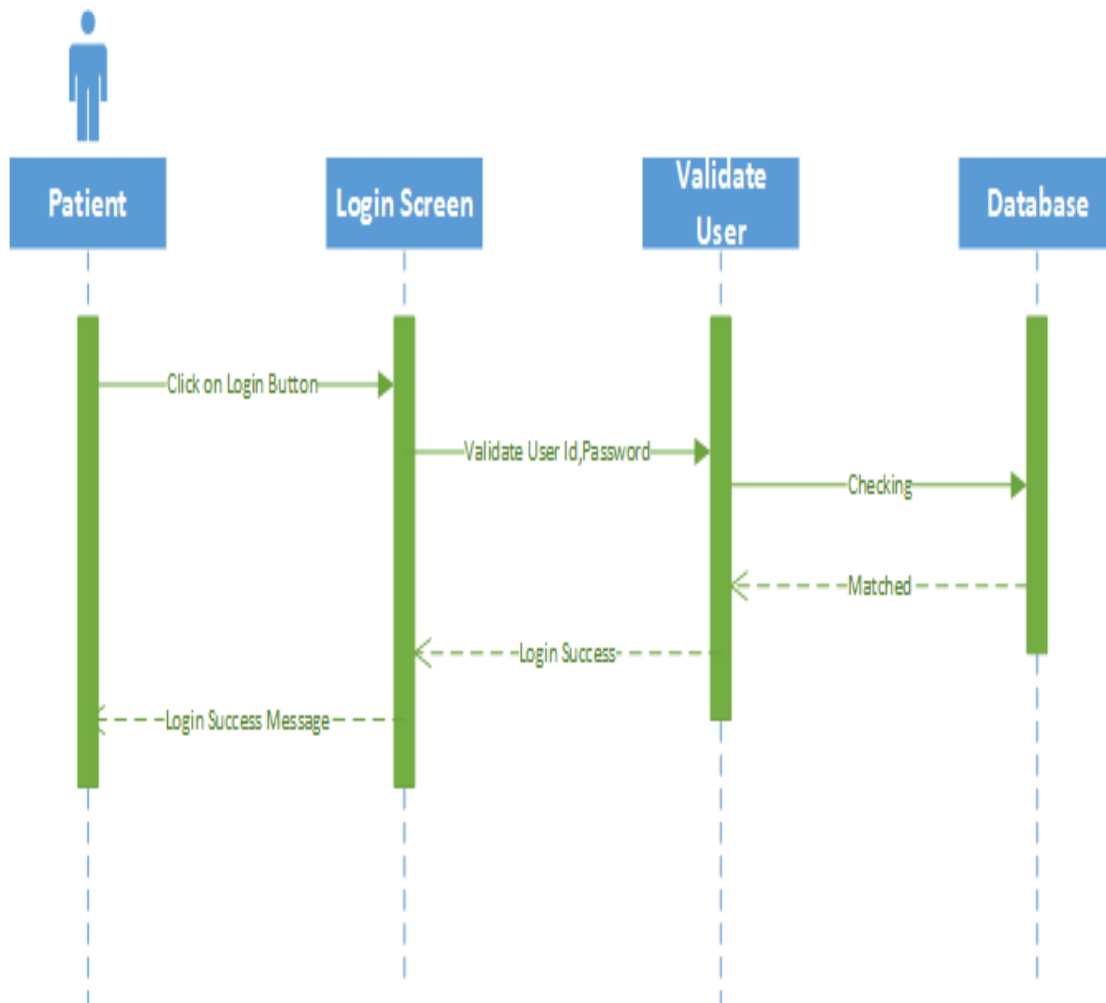


Figure 3.15 Sequence (Login)

3.4.2 Registration

Doctor Registration

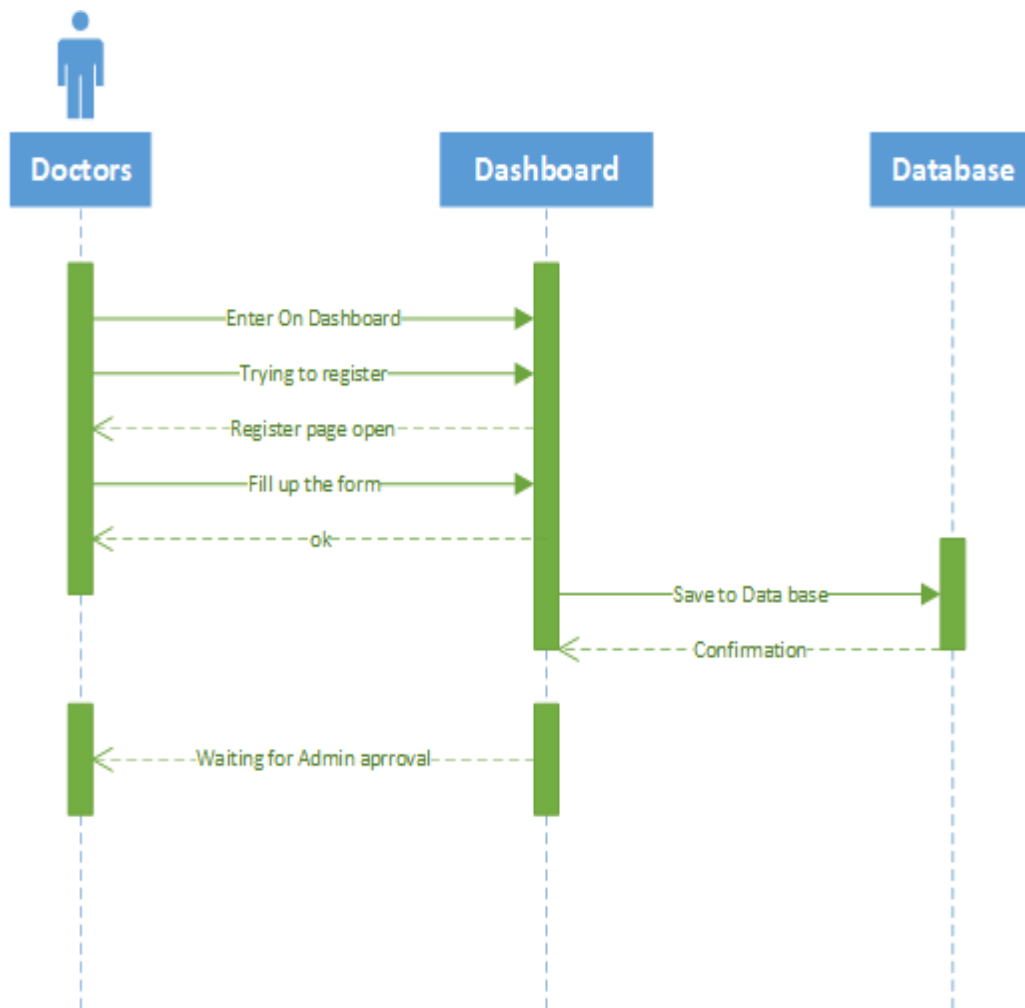


Figure 3.3 Activity (Registration)

3.4.3 Approve Doctor

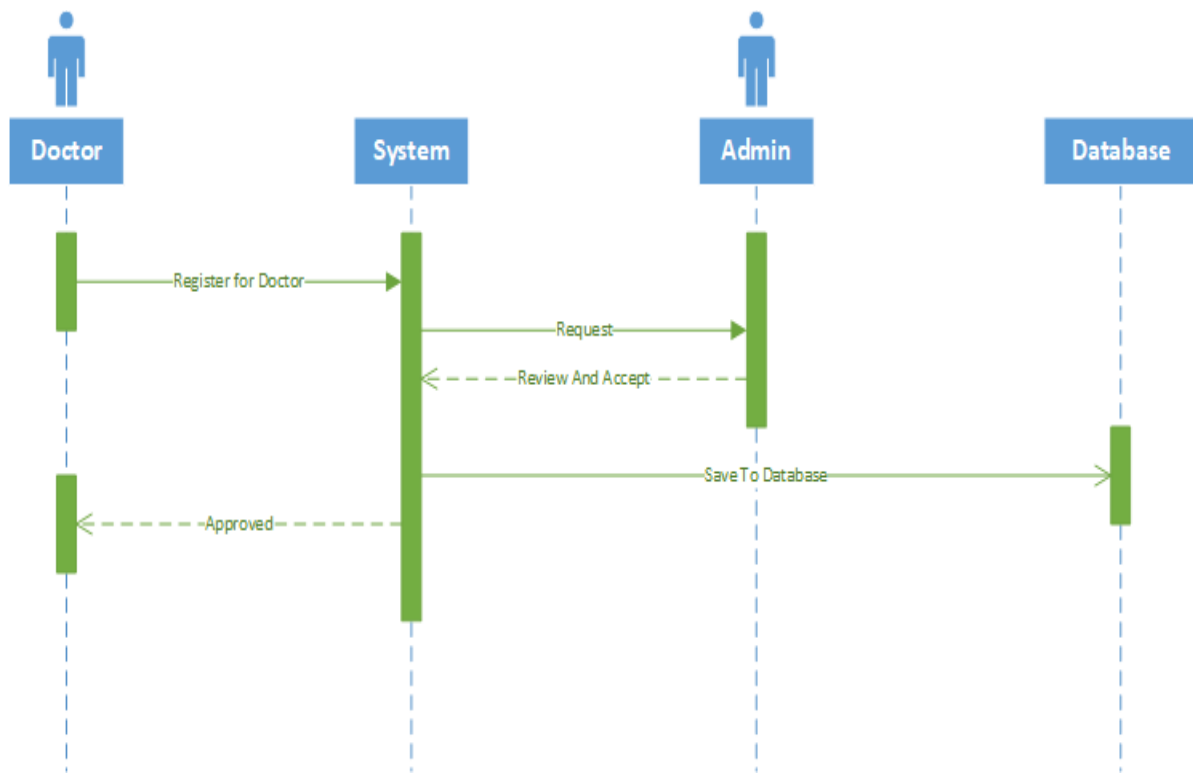


Figure 3.17 Activity (Approve Doctor)

3.4.4 Appointment Lists

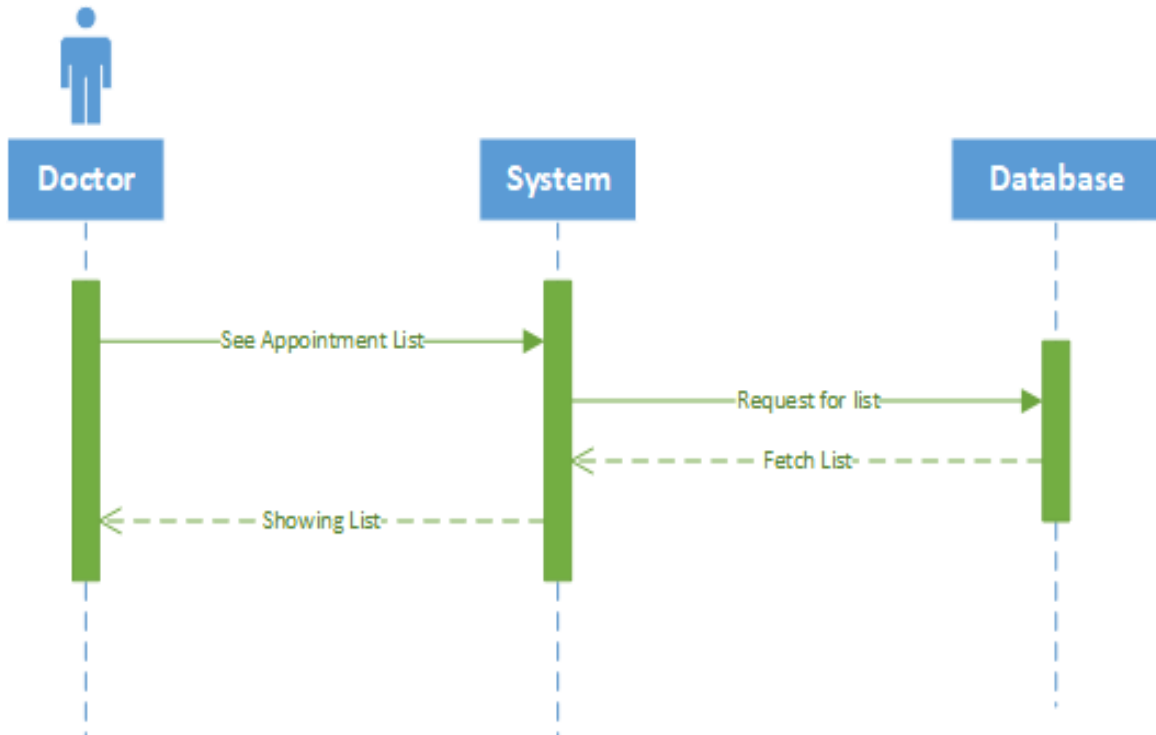


Figure 3.18 Activity (Appointment list)

3.4.5 Make Appointment

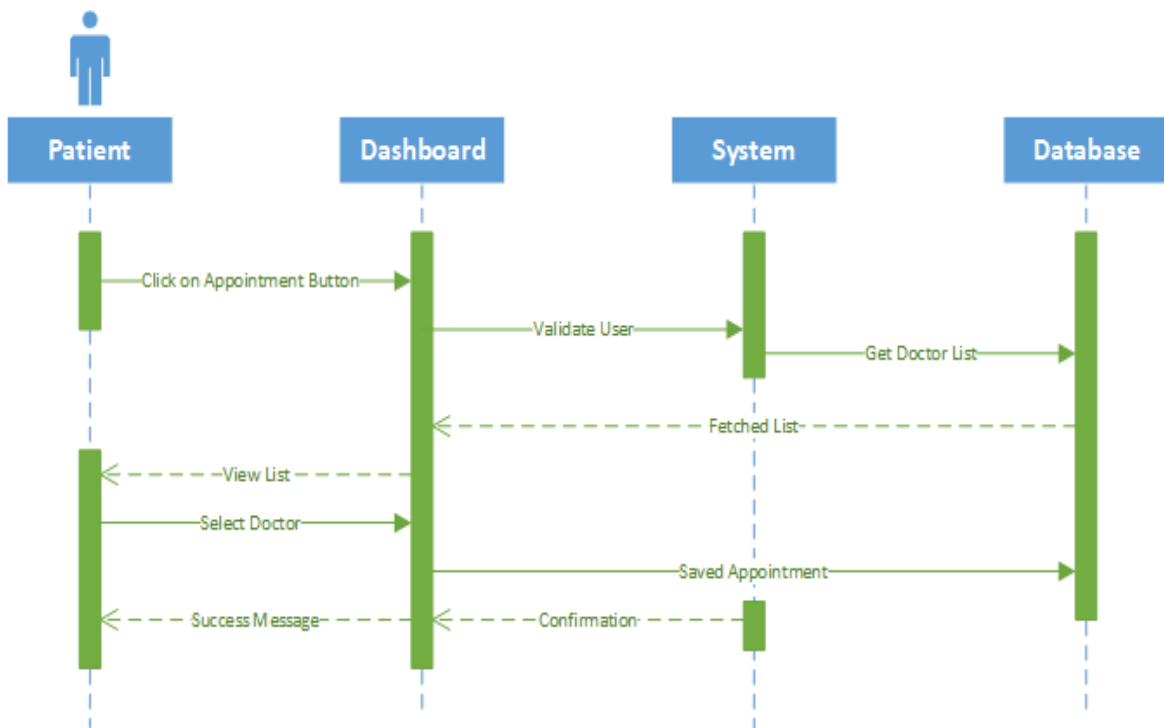


Figure 3.119 Activity (Make Appointment)

3.4.6 Approve Info/Tutorials

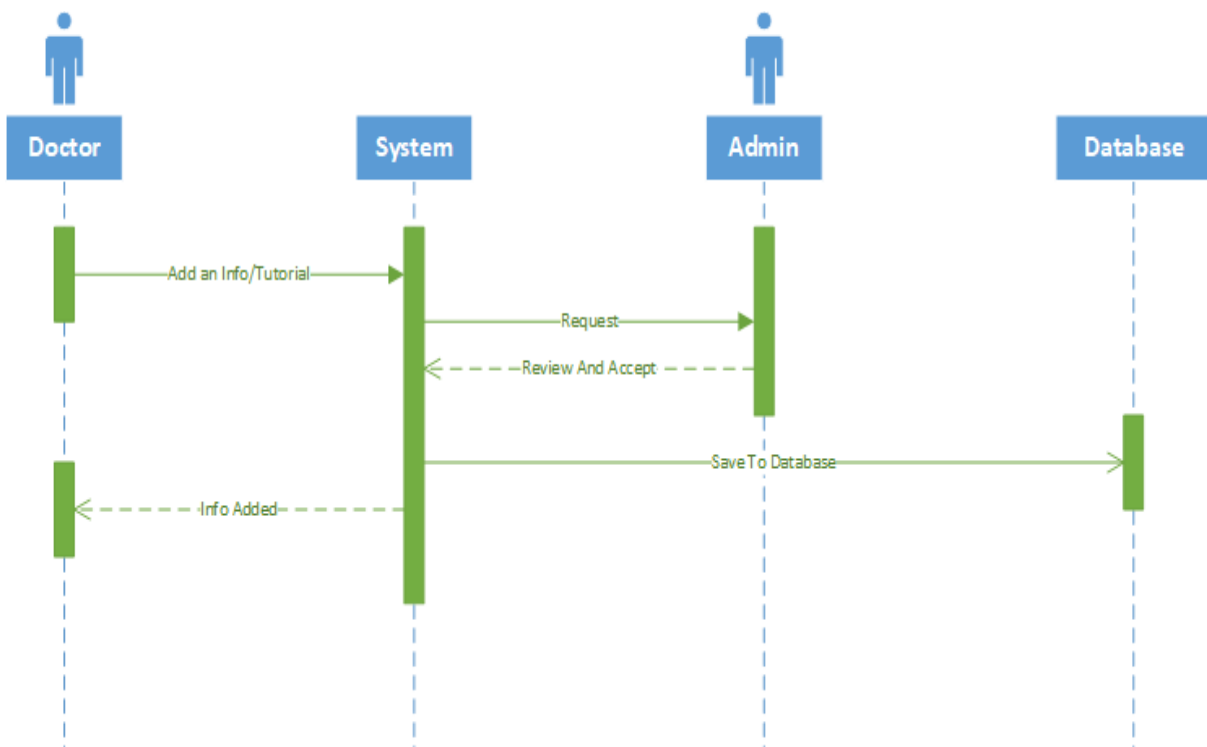


Figure 3.20 Activity (Approve Info)

3.4.7 Doctor List

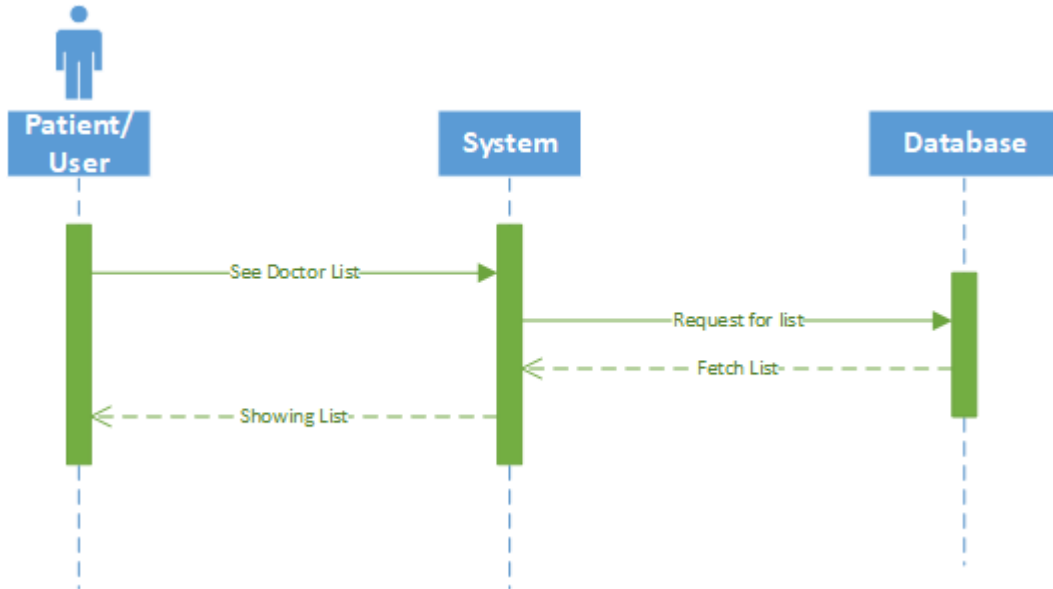


Figure 3.21 Activity (Doctor List)

3.4.8 See Post Tutorials/Therapy

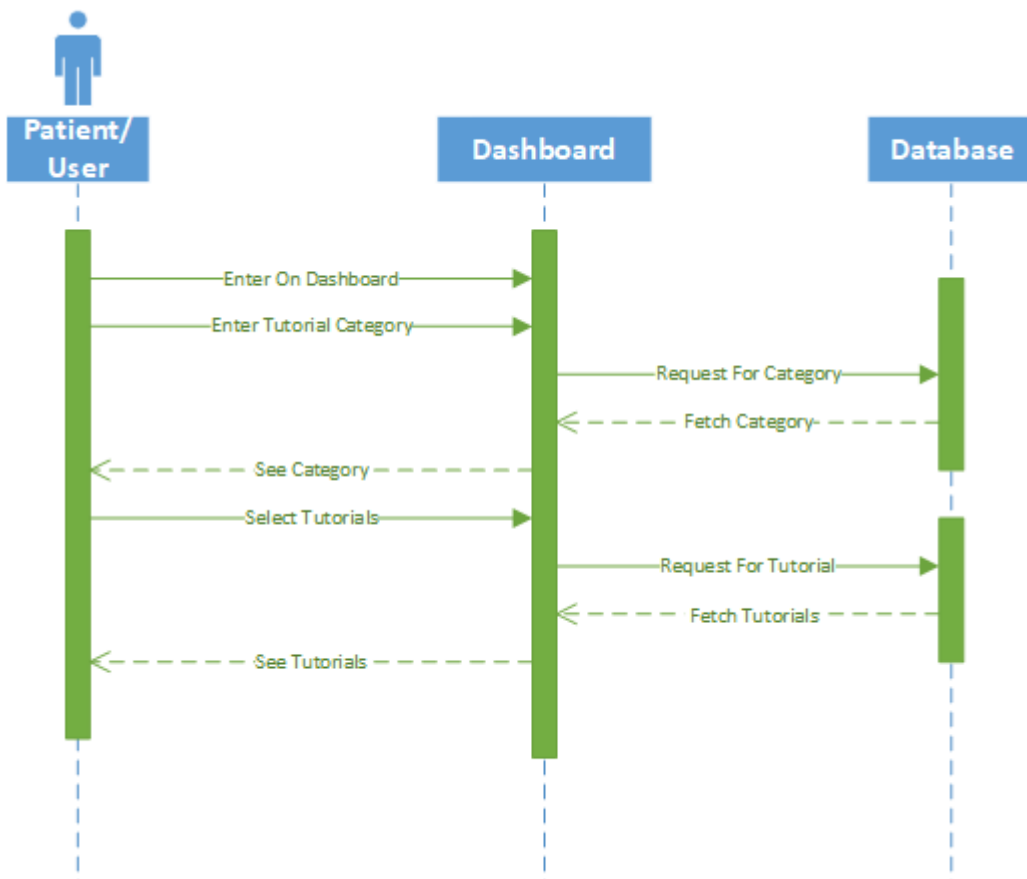


Figure 3.22 Activity (See post/tutorials)

3.4.9 Get Info

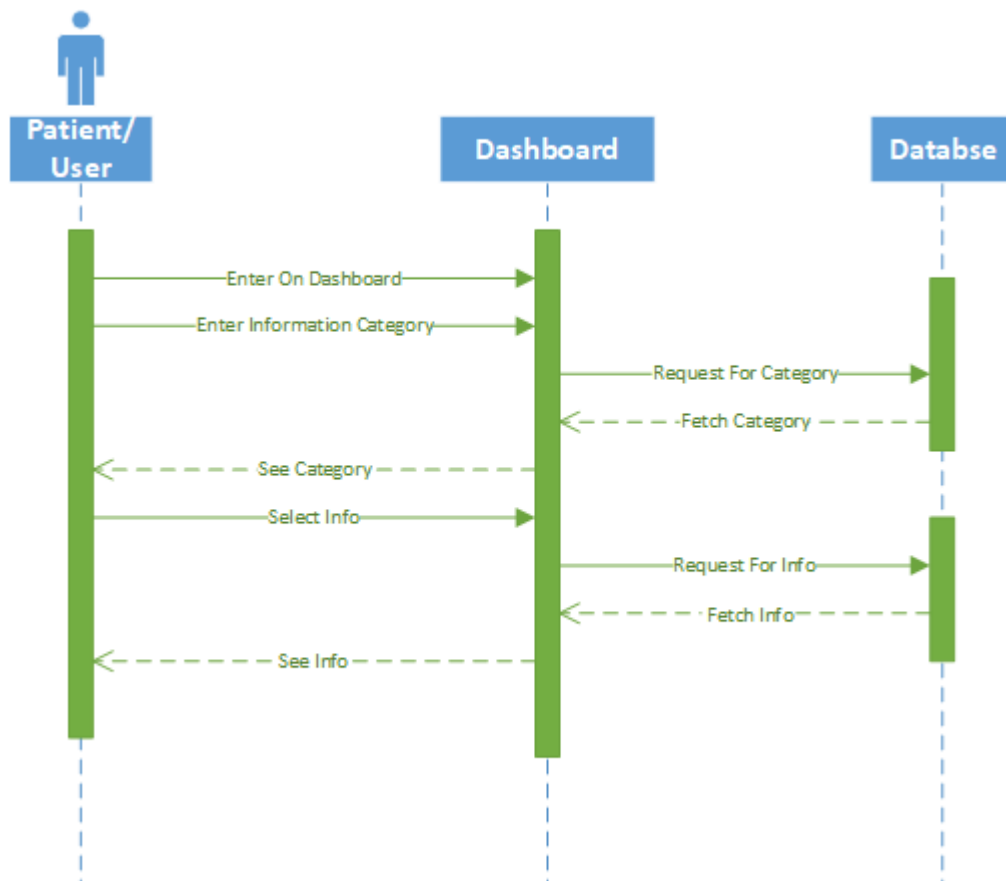


Figure 3.23 Activity (Get Info)

3.4.10 Manage Profile

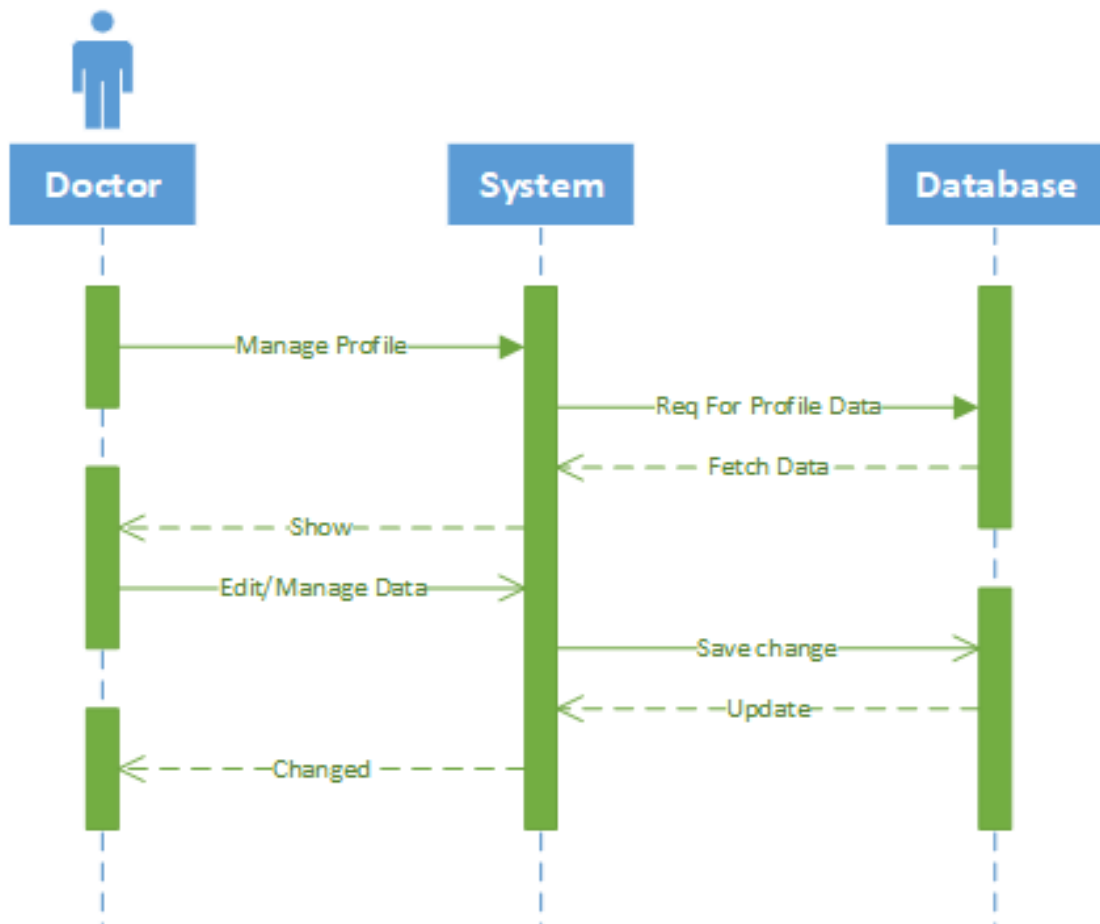


Figure 3.24 Activity (Manage Profile)

3.4.11 Communication

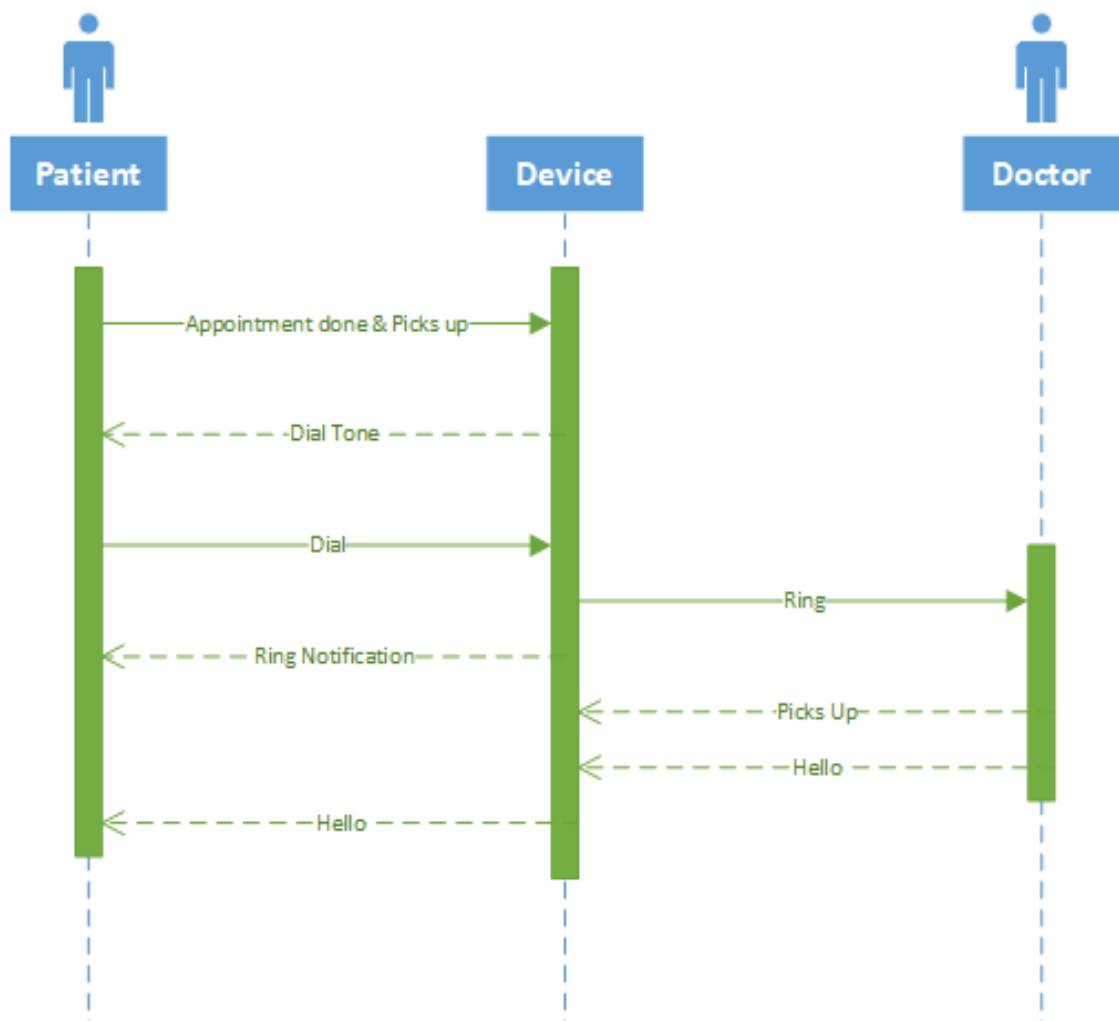


Figure 3.25 Activity (Live Communication)

3.4.12 See previous history

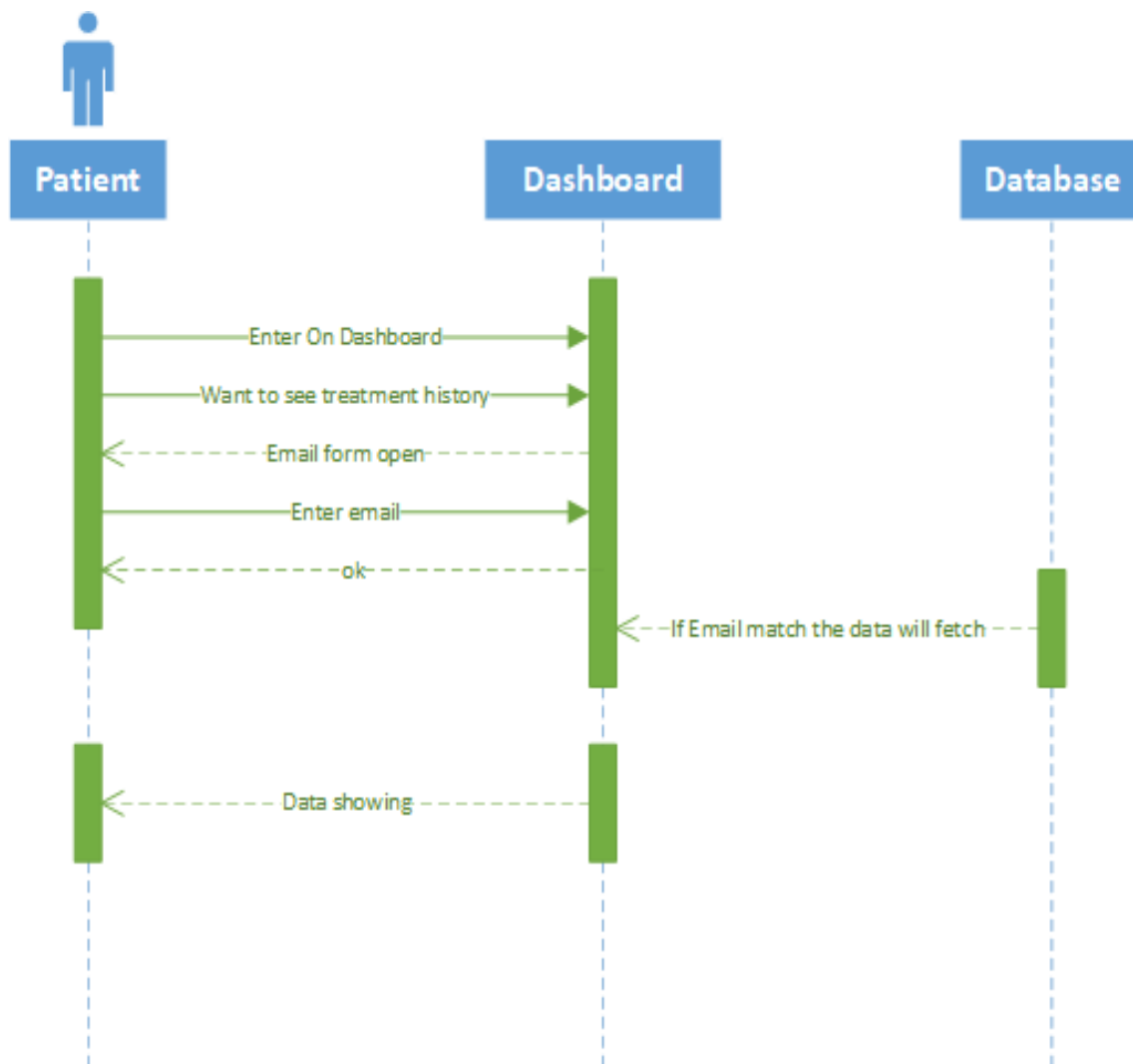


Figure 3.26 Activity (History)

4. CHAPTER-04 SYSTEM DESIGN SPECIFICATION

4.1 Entity Relationship Diagram

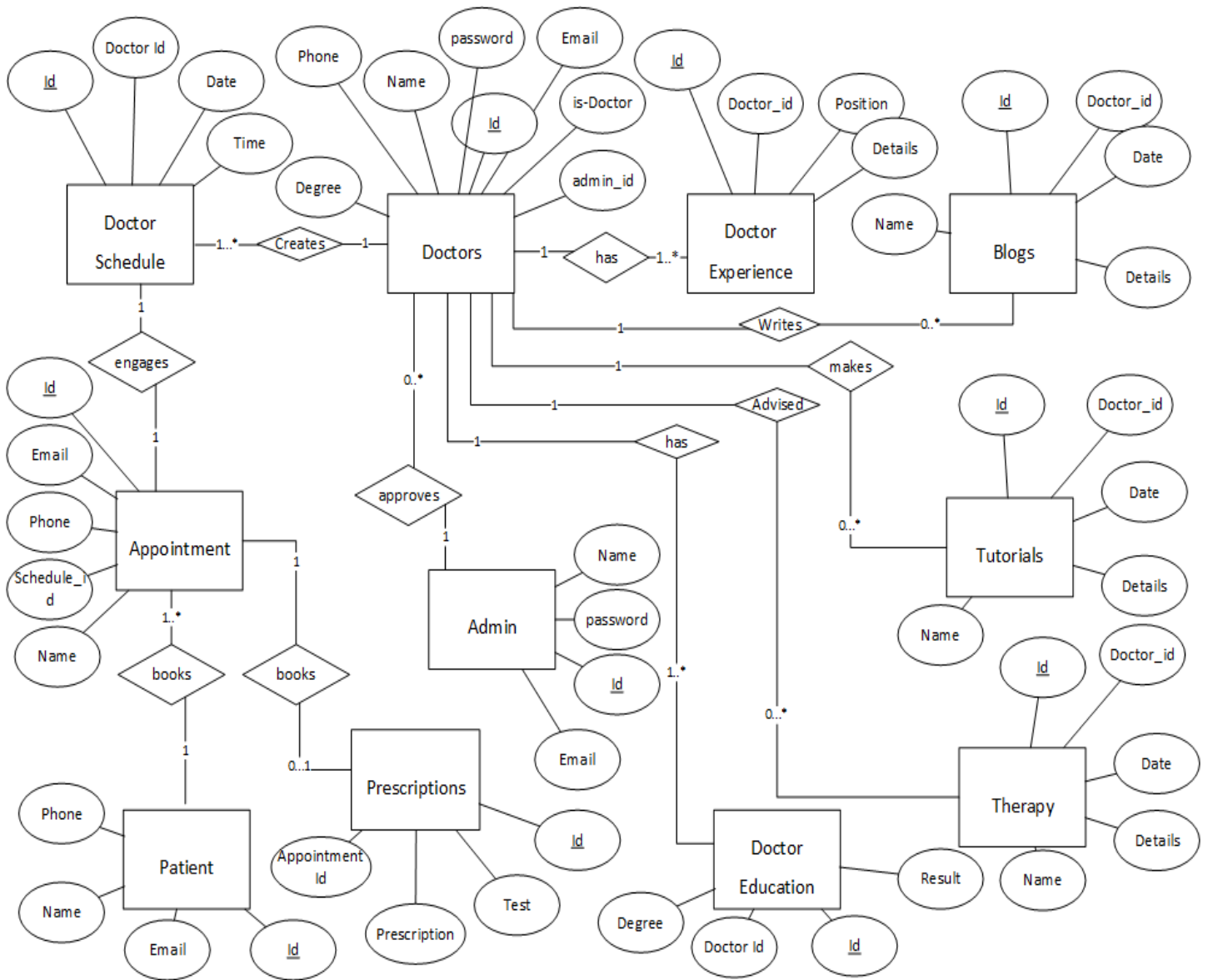


Figure 4.1 Entity Relation Diagram

4.2 Schema Diagram

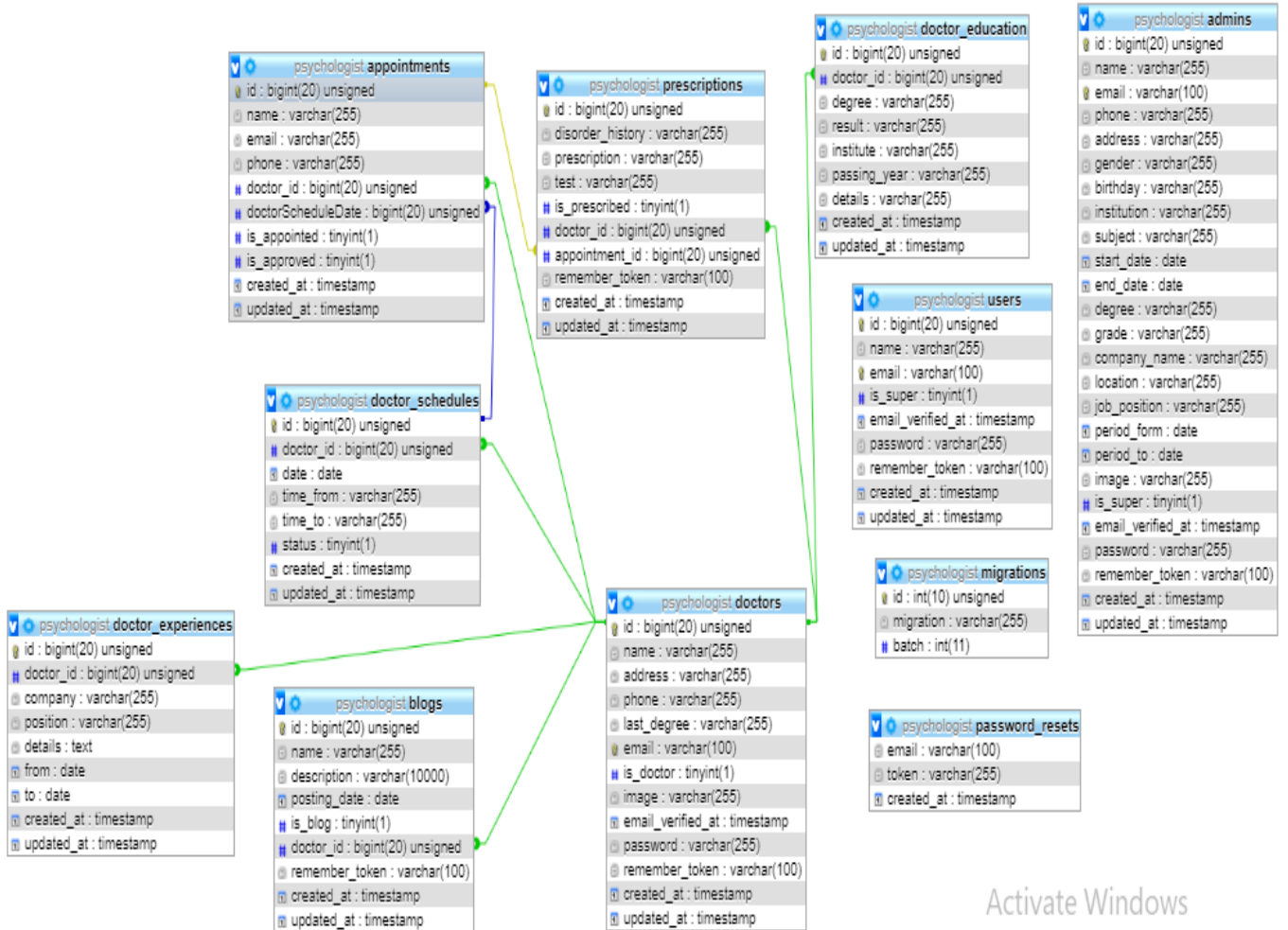


Figure 4.2 Schema Diagram

4.3 Development Tolls and Technology

4.3.1 User Interface Technology

- HTML
- CSS
- JAVASCRIPT

4.3.2 Implementation tools and platform

- PHP
- Laravel 5.8
- XAMMP
- PHP Storm

5. CHAPTER-05 SYSTEM TESTING

5.1 Feature to be Tasted

- Login
- Registration
- Manage Profile
- Make Appointment
- Manage Profile
- Take Up
- Approve post
- Manage Post
- Get History
- Online Survey
- Make a Post

5.2 Testing Strategies

5.2.1 Test Approach

- Testing approaches are done by the side of author or admin panel of this system.
- Testing approaches are going to apply in the above-mentioned features.
- To complete the test approach, we required to meet some criteria specifically.

5.2.2 Pass or Fail Criteria

- If the expected criteria is meet then mark as pass
- If the expected criteria is not meet then mark as fail

5.3 Testing Schedule

Here I presented the whole project part by part and add the schedule for each task as below

Table 5.1 Tasting Schedule

Taste Phase	Time
To create test plan	7 Days
To create the test Specification	14 Days
Test Specification Team	8 Days
Component Test	15 Days
Integration Test	7 Days

System Testing	24 Days
----------------	---------

5.4 Taste Case

5.4.1 Taste Case-01

Table 5.2 Taste Case 1 (Login)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Login	Try to login with registered data	1. Browse site 2. Input Email 3. Input Password 4. Press the login button	Email: <u>maheen@gmail.com</u> Password: 12345678	Logged in successfully	Pass
TU 02	Login	Try to login with unregistered data	1. Browse site 2. Input Email 3. Input Password 4. Press the login button	Email: <u>maheen12@gmail.com</u> Password: 12345445	Wrong email/Password and don't login	Pass

5.4.2 Taste Case-02

Table 5.3 Taste Case 2 (Registration)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
--------------	----------------	---------------	-----------	-----------	-----------------	-----------

TU 01	Registration	Check registration with valid data	1. Browse site 2. Input Email 3. Input Password 4. Fill requirement field 5. Click the register button	Email: <u>maheen@gmail.com</u> Password: 12345678 Req: ok	Registration successfully	Pass
TU 02	Registration	Check Registration with invalid data	1. Browse site 2. Input Email 3. Input Password 4. Fill requirement field 5. Click the register button	Email: <u>maheen12@gmail.com</u> Password: Req: nil	Every filled must required	Pass

5.4.3 Taste Case-03

Table 5.4 Taste Case 3 (Manage profile)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Manage profile	Check edit/update with valid data	1. Login 2. My Profile 3. Edit/Update with suitable data	Degree: MBBS Home: Dhaka Experience: 5 years Etc.	Registration successfully	Pass
TU 02	Manage profile	Check edit/update with invalid data	1. Login 2. My Profile 3. Edit/Update with suitable data	Degree: MBBS Home: Dhaka Experience: %/*& Etc.	Must be an number form	Pass

5.4.4 Taste Case-04

Table 5.5 Taste Case 4 (Take Up)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Take Up	Check take up with valid data	1. Login 2. Appointments 3. Take up 4. Write Prescriptions	Prescriptions: Sleep 8 hours Paracetamol 3 pc etc.	Registration successfully	Pass
TU 02	Take Up	Check Take up with invalid data	1. Login 2. Appointments 3. Take up 4. Write Prescriptions	Prescriptions:	Empty field	Pass

5.4.5 Taste Case-05

Table 5.6 Taste Case 5 (Approve doctor)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Approve doctor	Check Doctor request which contain valid data	1. Login 2. Go to Doctor Request 3. Check info	Name: Absar Degree: MBBS Experience: ok Etc.	Registration successfully	Pass
TU 02	Approve doctor	Check Doctor request which contain invalid data	1. Login 2. Go to Doctor Request 3. Check info	Name: john Degree: FCPS Experience: nil	Empty field	Pass

5.4.6 Taste Case-06

Table 5.7 Taste Case 6 (Approve posts)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Approve posts	Check Post request which contain valid data	1. Login 2. Go to post Request 3. Check info	Name: Depression Body: ok Etc.	Registration successfully	Pass
TU 02	Approve posts	Check Post request which contain invalid data	1. Login 2. Go to Post Request 3. Check info	Name: Depression Body: null Etc.	Empty field Denied post	Pass

5.4.7 Taste Case-07

Table 5.8 Taste Case 7 (Make Appointment)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Make Appointment	Check appointment request which contain valid data	1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit	Name: Maheen Email: maheen@gmail.com Doctor: Selected Time: Selected	Make appointment successfully	Pass
TU 02	Make Appointment	Check appointment request which contain invalid data	1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit	Name: Maheen Email: maheen.gmail.com Doctor: Selected Time: Selected	Appointment denied due to false email	Pass

5.4.8 Taste Case-08

Table 5.9 Taste Case 8 (Manage post)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Manage post	Check edit/update with valid data	1. Login 2. Posts 3. Edit/Update with suitable data	Name: ok Details: ok Etc.	Edit/delete success	Pass
TU 02	Manage post	Check edit/update with invalid data	1. Login 2. Posts 3. Edit/Update with suitable data	Name: nil Details: nil Etc.	Empty Fields redirect to post	Pass

5.4.9 Taste Case-09

Table 5.10 Taste Case 9 (Get history)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Get history	Check History with valid email	1. Enter site 2. Go to History email text box 3. Enter email 4. Submit	Email: maheen@gmail.com	Open history	Pass
TU 02	Get history	Check History with invalid email	1. Enter site 2. Go to History email text box 3. Enter email 4. Submit	Email: maheen.gmail.com	Wrong email	Pass

5.4.10 Taste Case-10

Table 5.11 Taste Case 10 (Survey)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Survey	Join a survey and identify illness	1. Enter site 2. Go to Survey 3. Answer all the MCQ 4. Submit	MCQ answers	Suggest doctor and identify problem	Pass
TU 02	Survey	Join a survey and identify illness	1. Enter site 2. Go to Survey 3. Do not answer all the MCQ 4. Submit	MCQ answers	Answer all the MCQ first	Pass

5.4.11 Taste Case-11

Table 5.12 Taste Case 11 (Make a post)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Make a post	Check appointment request which contain valid data	1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit	Name: Maheen Email: maheen@gmail.com Doctor: Selected Time: Selected	Make appointment successfully	Pass
TU 02	Make a post	Check appointment request which contain invalid data	1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit	Name: Maheen Email: maheen.gmail.com Doctor: Selected Time: Selected	Appointment denied due to false email	Pass

5.5 Test Environment (Browser)

- Firefox
- MS Edge
- Chrome

6. CHAPTER-06 USER INTERFACE

6.1 User/Patient Panel

6.1.1 User Landing Page

1. User Landing Page

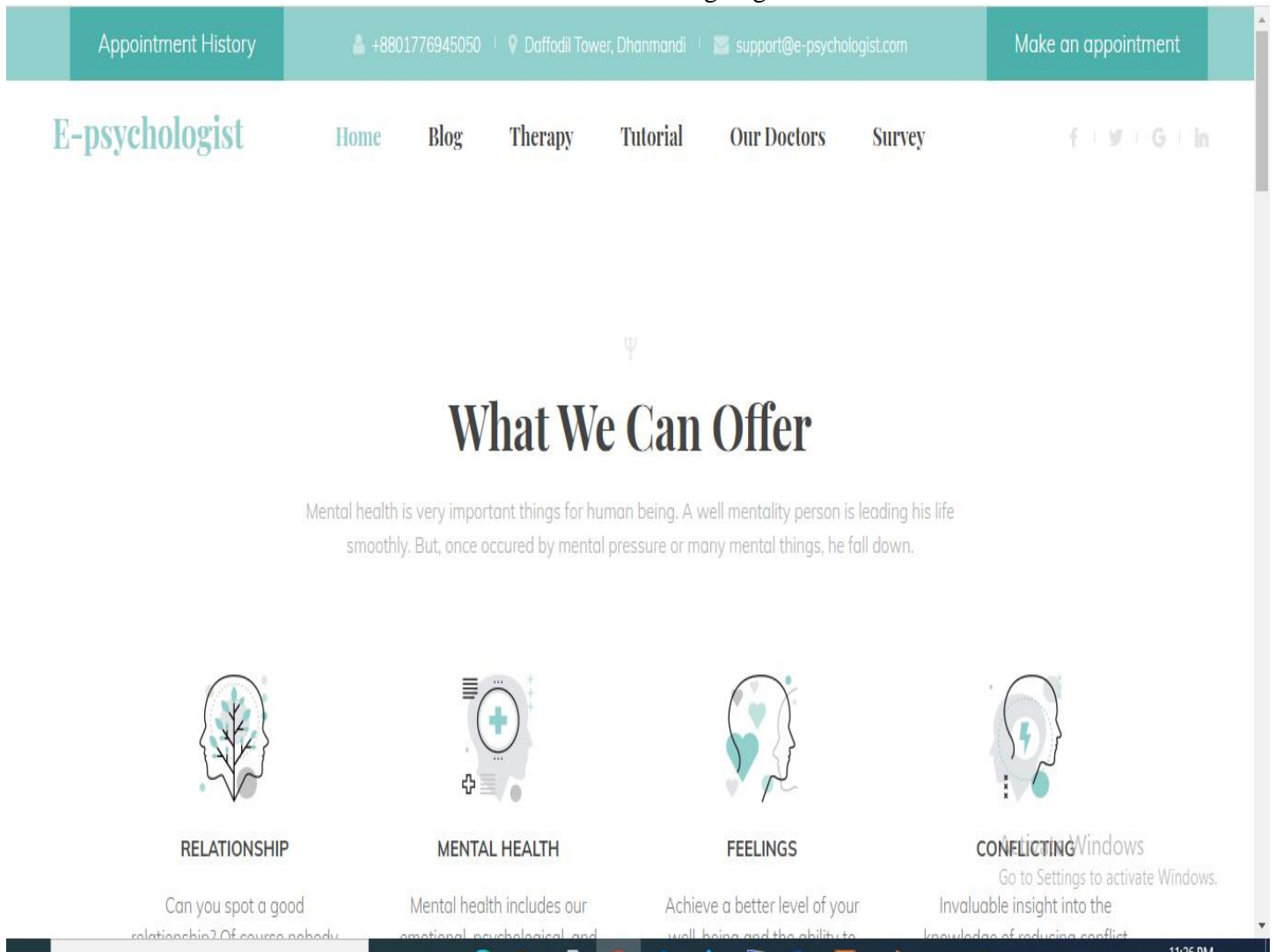


Figure 6.1 User Landing Page

6.1.2 Make an Appointment

2. Appointment form

Make an Appointment

Exercitation cupim ex, short ribs cow in ullamco corned beef veniam kevin.
Eu frankfurter ham hock ball tip reprehenderit adipiscing ipsum

Your Name Email Address

Select Doctor Phone

mm/dd/yyyy ---:--:--

Make Appointment

Figure 6.2 Make Appointment Page

6.1.3 Browse Blogs and info

3. Blogs page

Appointment History +8801776945050 Daffodil Tower, Dhanmandi support@e-psychologist.com Make an appointment

E-psychologist Home Blog Therapy Tutorial Our Doctors Survey f | | G | |

0000-00-00 Test Test

2021-06-06 test tgjhbjrgkmrkgnfgmfkm...

2021-06-12 regregre gregregregre

2021-06-12 test ক্রোনার কারণে আর্জেন্ট...

Activate Windows
Go to Settings to activate Windows.

6.1.4 Browse Blogs and info

4. Therapy/Tutorials page

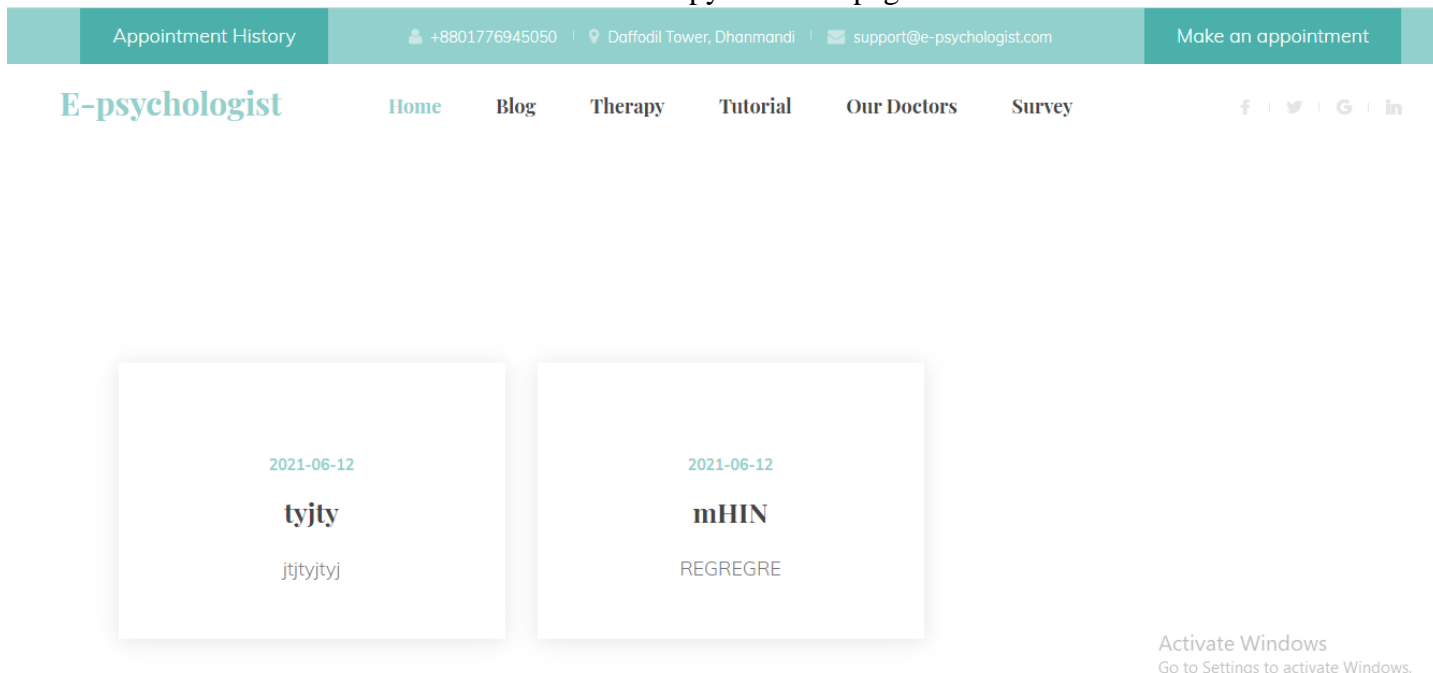


Figure 6.4 Therapy Page

6.1.5 Appointment

5. Confirmation mail

confirmation mail

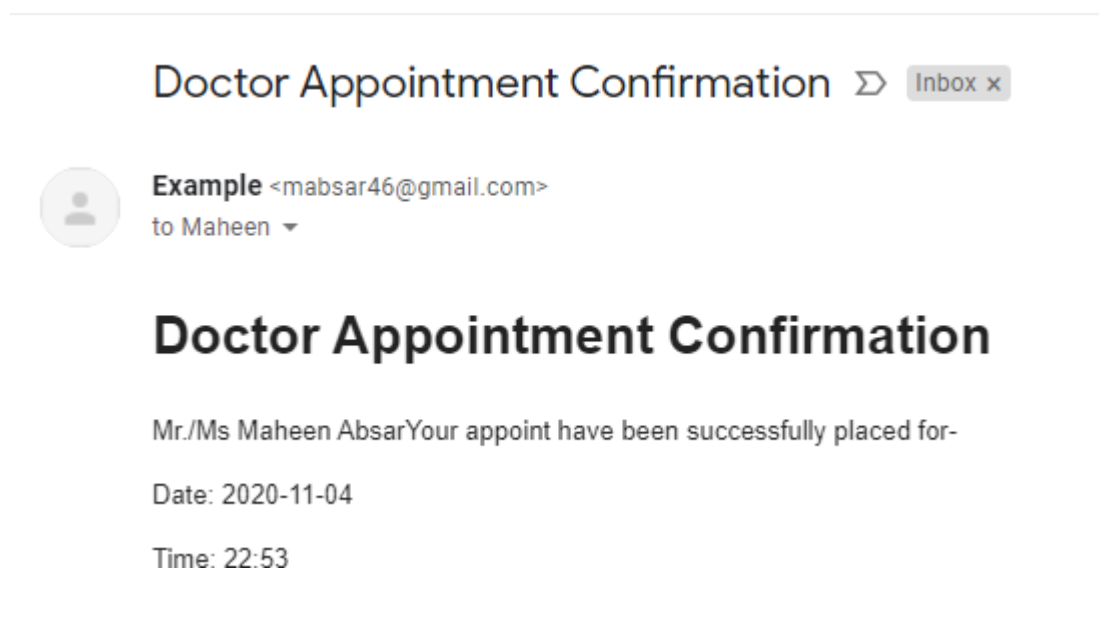


Figure 6.5 Appointment Confirmation

6.1.6 Prescription copy mail

6. Prescription mail

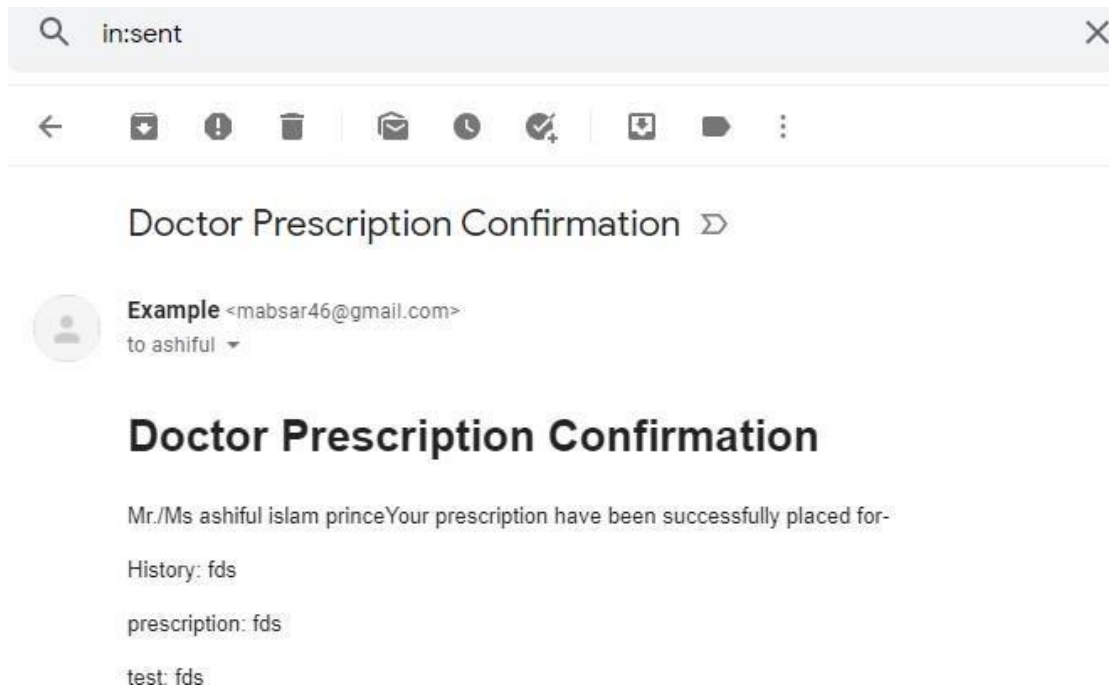


Figure 6.6 Prescription Confirmation

6.1.7 Check History

7. Check history mail text box



Figure 6.7 Check History

6.1.8 Survey

8. Survey page

• 19. Specifically for Machine Translation post-editing, how useful would language-specific keyboard shortcuts be:

	Not useful	Useful	Don't know	Not applicable
For changing the gender of a word (e.g. masculine to feminine)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For changing the number of a word (e.g. singular to plural)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For changing verb form (e.g. future to conditional)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For adding or deleting a conjunction?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For adding or deleting a preposition?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For adding or deleting a particle/postposition?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
For adjusting word order within a sentence?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 6.8 Survey

6.2 Doctor Panel

6.2.1 Doctor Dashboard

9. Doctor Dashboard

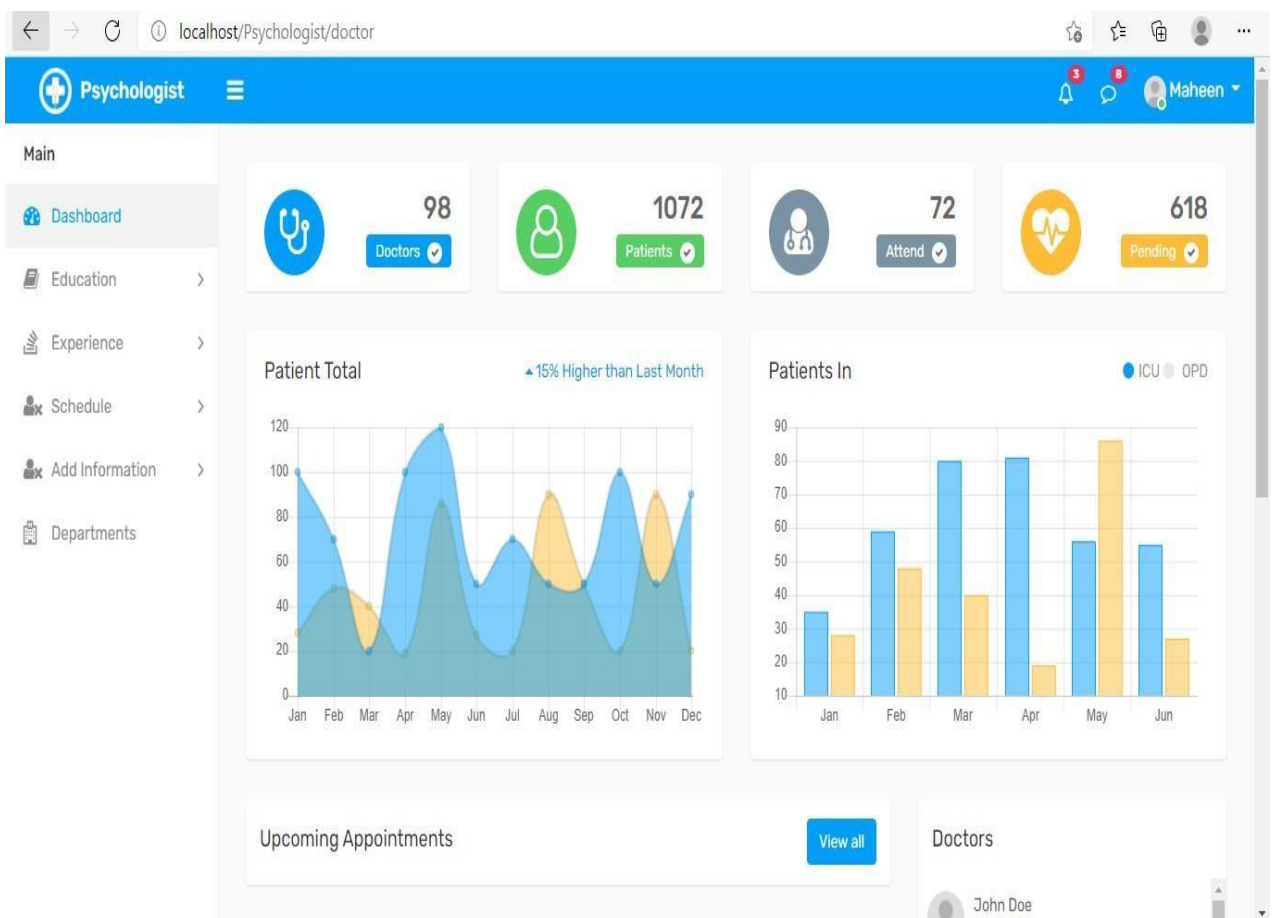


Figure 6.9 Doctor Dashboard

6.2.2 Doctor Profile

10. Profile Page

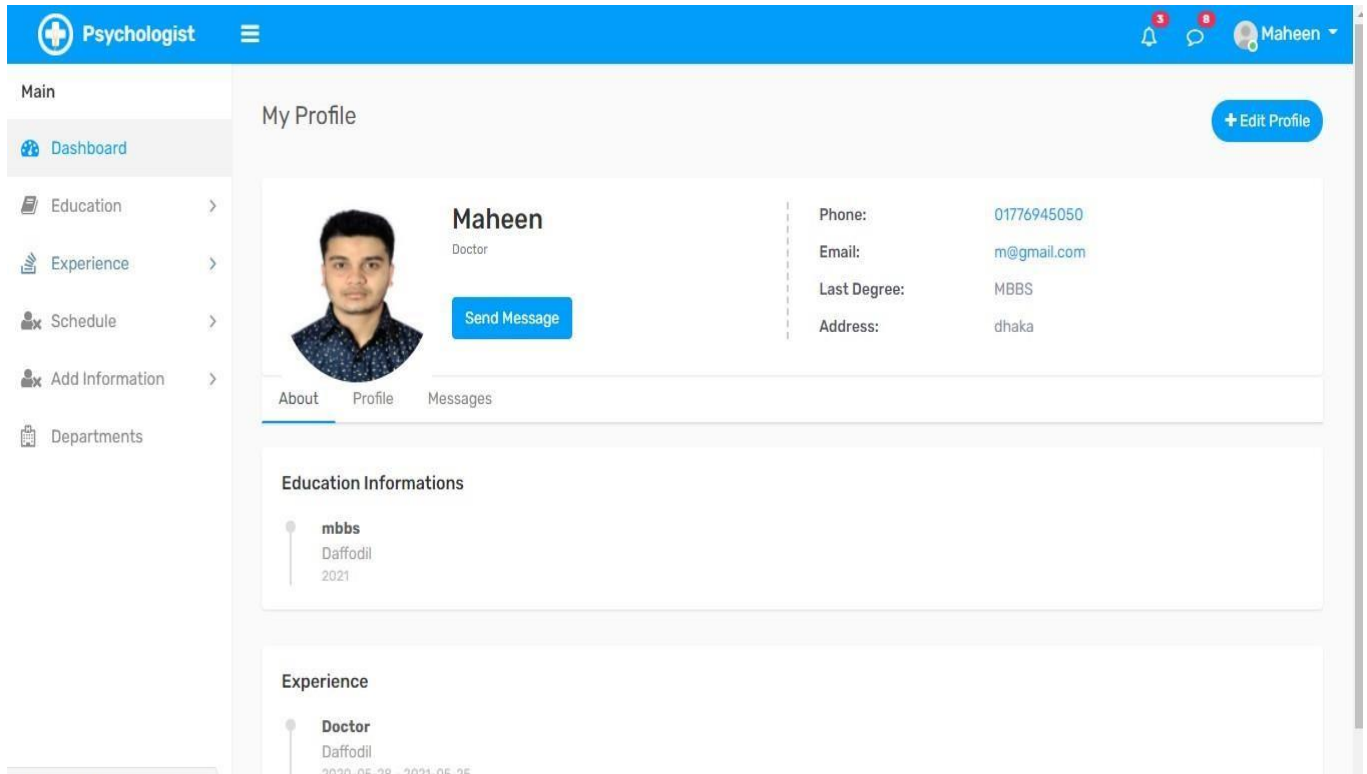


Figure 6.10 Doctor Profile Page

6.2.3 Doctor Edit Profile

11. Edit Profile Page

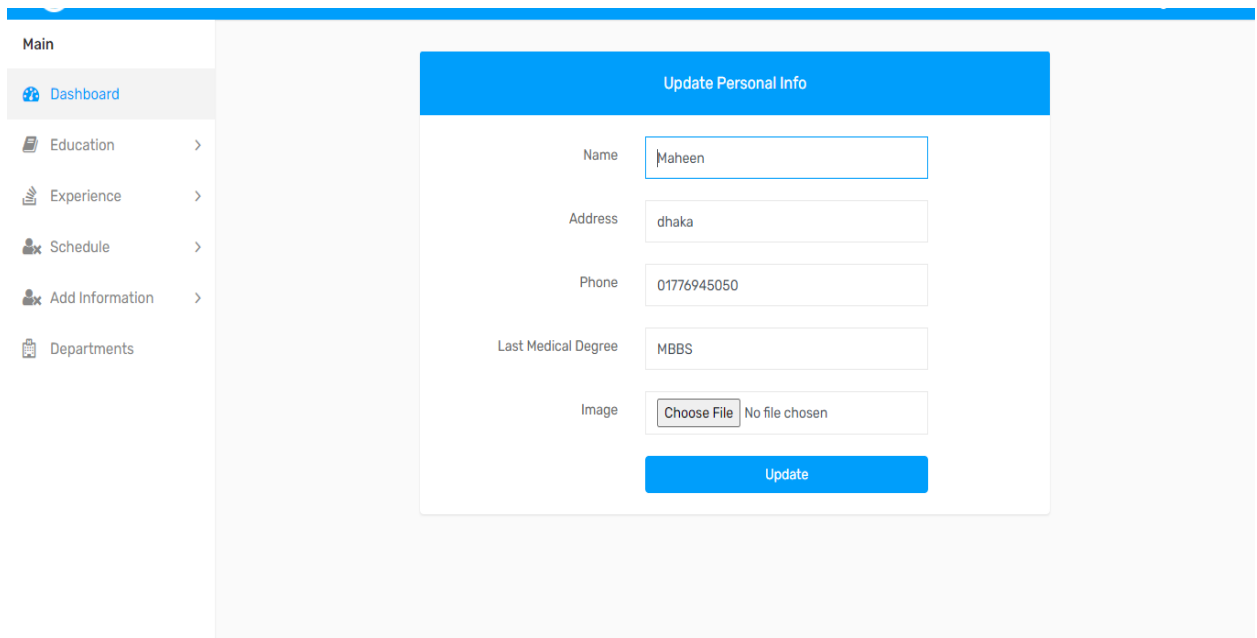
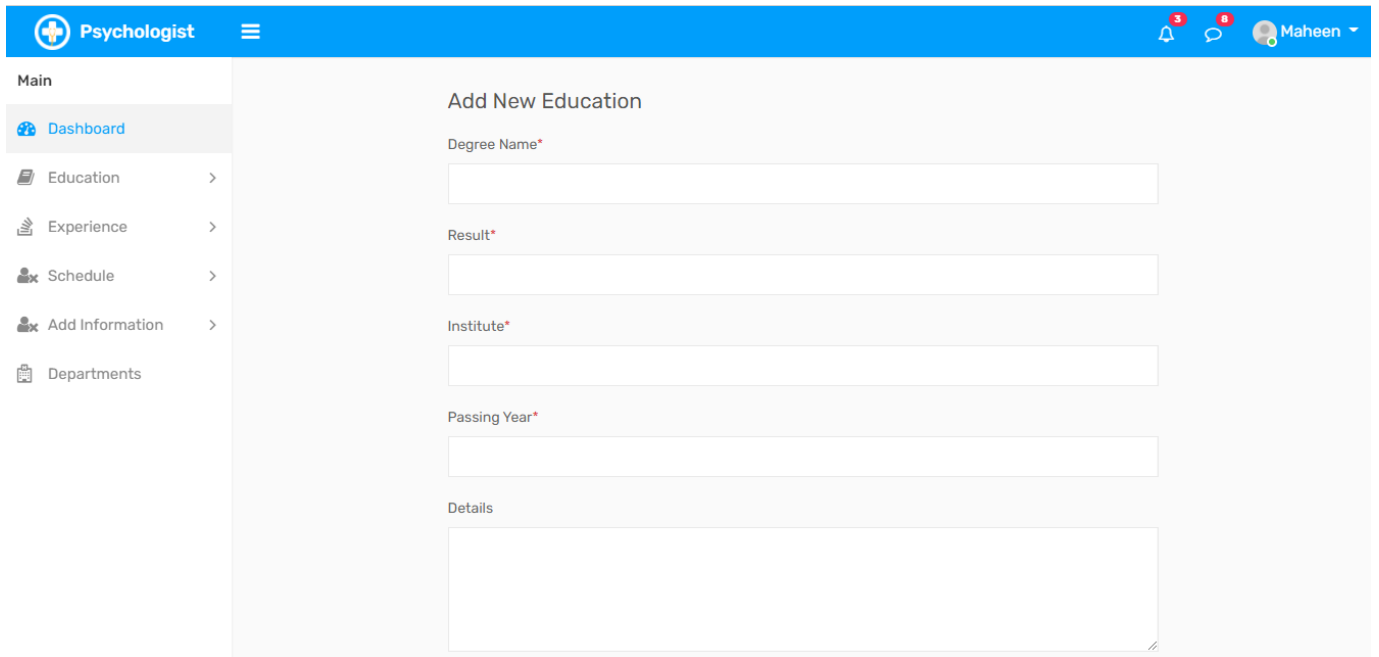


Figure 6.11 Doctor Edit Profile

6.2.4 Add Education

12. Add education page



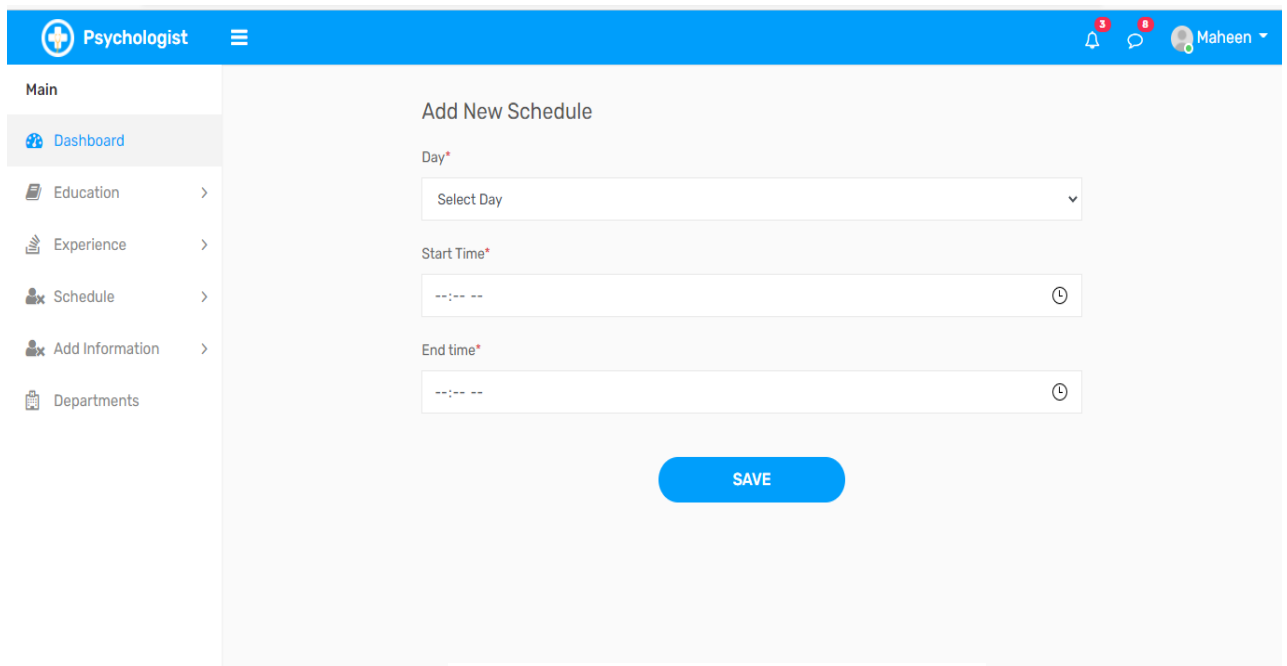
The screenshot shows a web application interface for a psychologist. The top navigation bar is blue with the text 'Psychologist' and a menu icon. On the right, there are notification icons and a user profile for 'Maheen'. A left sidebar menu is titled 'Main' and includes 'Dashboard', 'Education', 'Experience', 'Schedule', 'Add Information', and 'Departments'. The main content area is titled 'Add New Education' and contains the following form fields:

- Degree Name* (text input)
- Result* (text input)
- Institute* (text input)
- Passing Year* (text input)
- Details (text area)

Figure 6.12 Doctor Add Education

6.2.5 Add Schedule

13. Add schedule page



The screenshot shows the 'Add New Schedule' form in the same web application. The top navigation bar and sidebar are identical to the previous screenshot. The main content area is titled 'Add New Schedule' and contains the following form fields:

- Day* (dropdown menu with 'Select Day' selected)
- Start Time* (time picker)
- End time* (time picker)

Below the form fields is a blue 'SAVE' button.

Figure 6.13 Doctor Add Schedule

6.2.6 Add Experience

14. Add education page

The screenshot shows a web application interface for adding education. The header is blue with a white cross icon and the text 'Psychologist'. On the right of the header, there are notification icons and a user profile 'Maheen'. The left sidebar is white with a blue header 'Main' and lists navigation items: Dashboard, Education, Experience, Schedule, Add Information, and Departments. The main content area is titled 'Add New Education' and contains the following form fields:

- Position / Job Title* (text input)
- Organization* (text input)
- From* (date input with 'mm/dd/yyyy' placeholder and a calendar icon)
- To (date input with 'mm/dd/yyyy' placeholder and a calendar icon)
- Details (text area)

Figure 6.14 Add Experience

6.2.7 Appointment List

15. Appointment list page

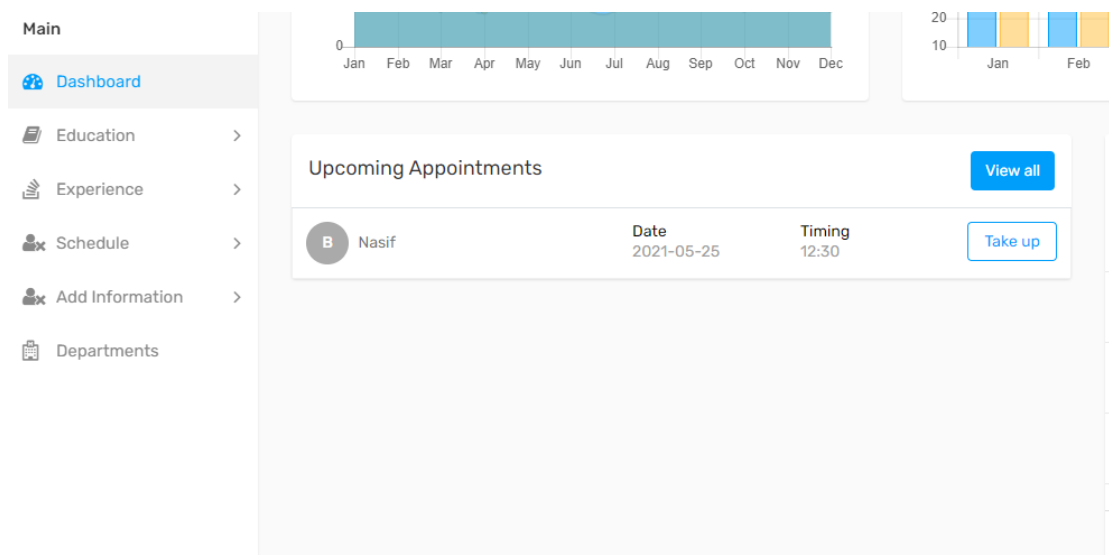


Figure 6.15 Appointment List

6.2.8 Doctor Experience

16. Doctor Experience page

Position	Organization	From	To	Details	Actions
R Doctor	Daffodil	2020-05-28	2021-05-25	ok	⋮

Figure 6.16 Doctor Add Experience

6.2.9 View Education

17. View education page

Degree	Institute	Result	Passing Year	Details	Actions
R mbbs	Daffodil	3.5	2021	ok	⋮

Figure 6.17 Doctor View Education

6.2.10 View Schedule

18. View schedule page

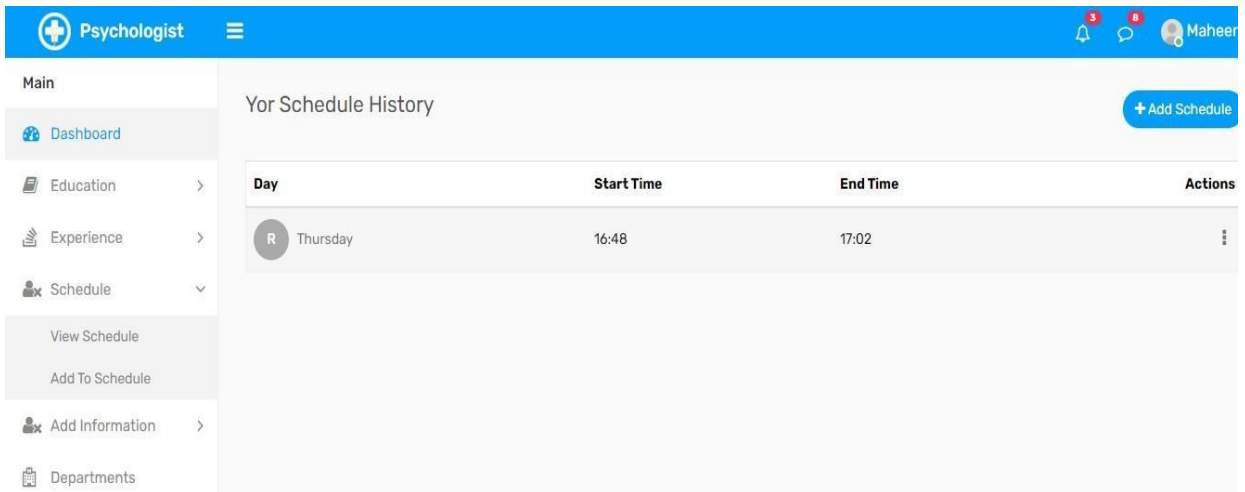


Figure 6.18 Doctor View Schedule

6.2.11 Write Prescription

19. Prescription page

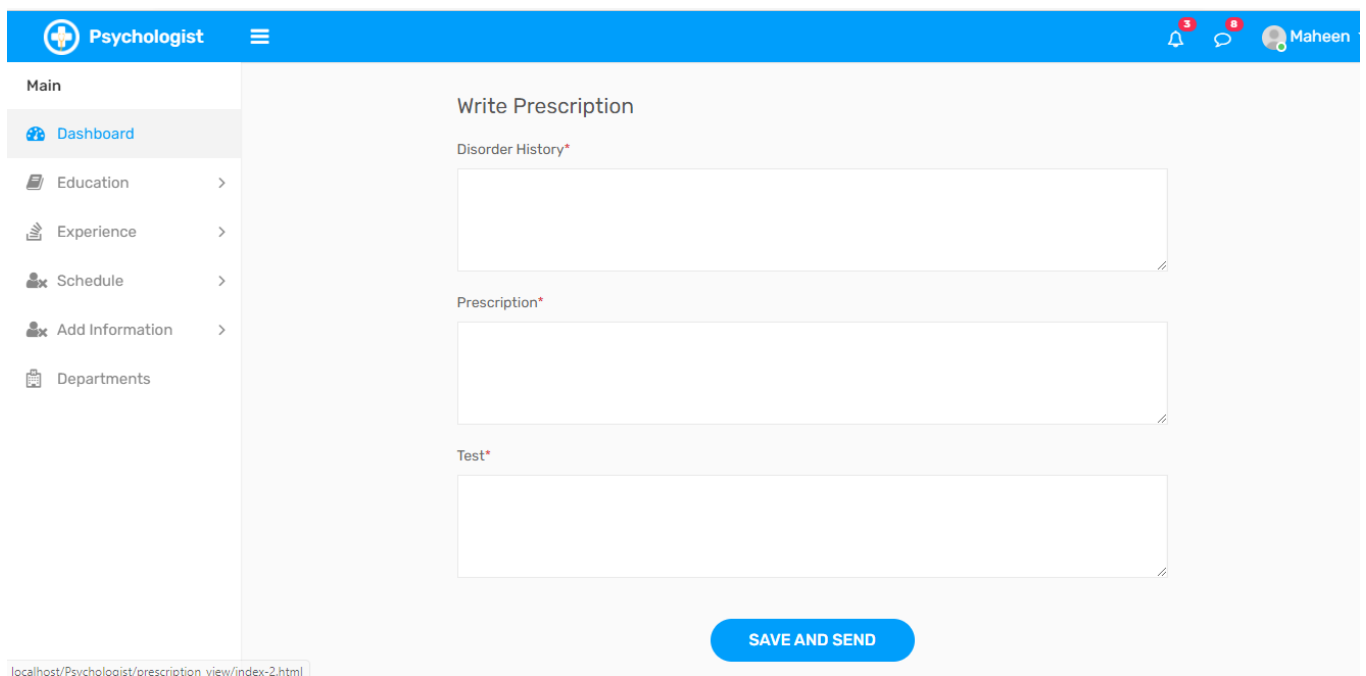


Figure 6.19 Prescription

6.2.12 Add post

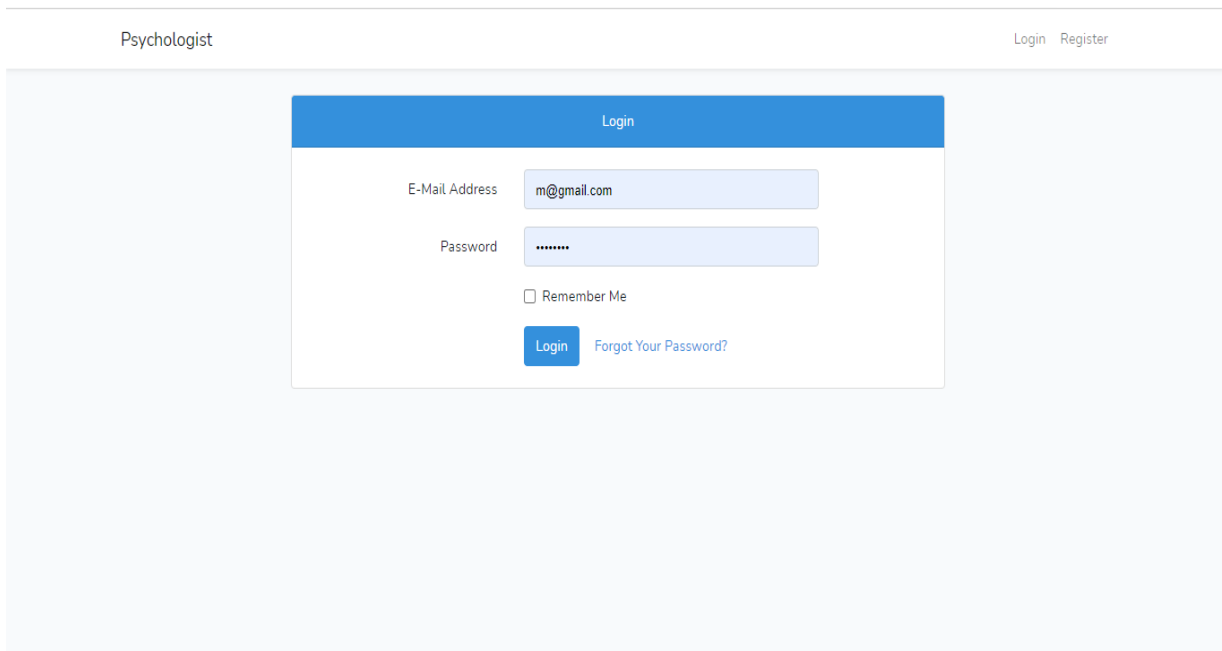
20. Add post page

The screenshot shows a web application interface for adding a new blog post. The header is blue and contains the 'Psychologist' logo, a menu icon, and the user's name 'Maheen' with a dropdown arrow. The left sidebar is white and lists navigation options: 'Main', 'Dashboard', 'Education', 'Experience', 'Schedule', 'Add Information', and 'Departments'. The main content area is light gray and titled 'Add New Blog Information'. It contains three input fields: 'Title*' with the value 'Diu', 'Description*' with the value 'Daffodil', and 'Posting Date*' with the value '05/25/2021'. A blue 'SAVE' button is centered at the bottom of the form.

Figure 6.20 Add Post

6.2.13 Login

21. Login form



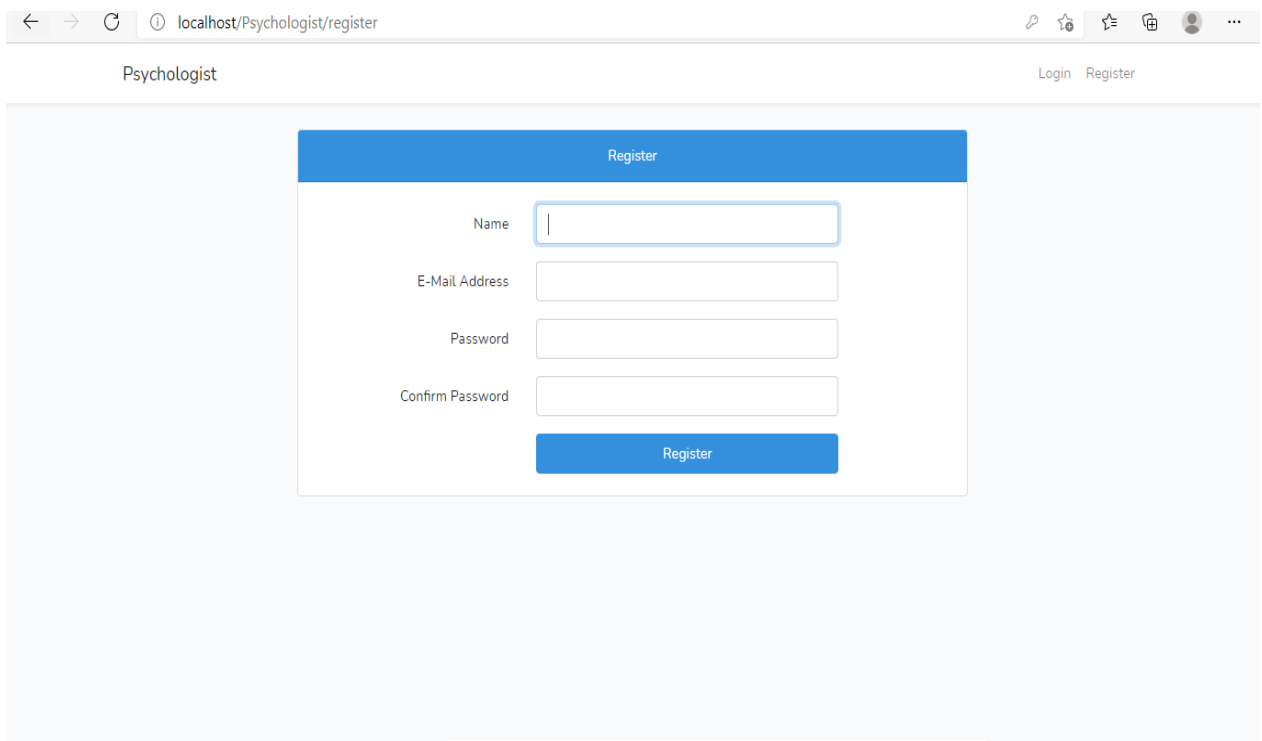
The screenshot shows a web browser window with the URL `localhost/psychologist/register`. The page title is "Psychologist" and there are links for "Login" and "Register". The main content is a "Login" form with a blue header. The form contains the following elements:

- E-Mail Address:
- Password:
- Remember Me
- [Forgot Your Password?](#)

Figure 6.21 Login

6.2.14 Registration

22. Registration Form



The screenshot shows a web browser window with the URL `localhost/psychologist/register`. The page title is "Psychologist" and there are links for "Login" and "Register". The main content is a "Register" form with a blue header. The form contains the following elements:

- Name:
- E-Mail Address:
- Password:
- Confirm Password:
-

Figure 6.22 Registration

6.3 Admin panel

6.3.1 Admin Dashboard

23. Registration Form

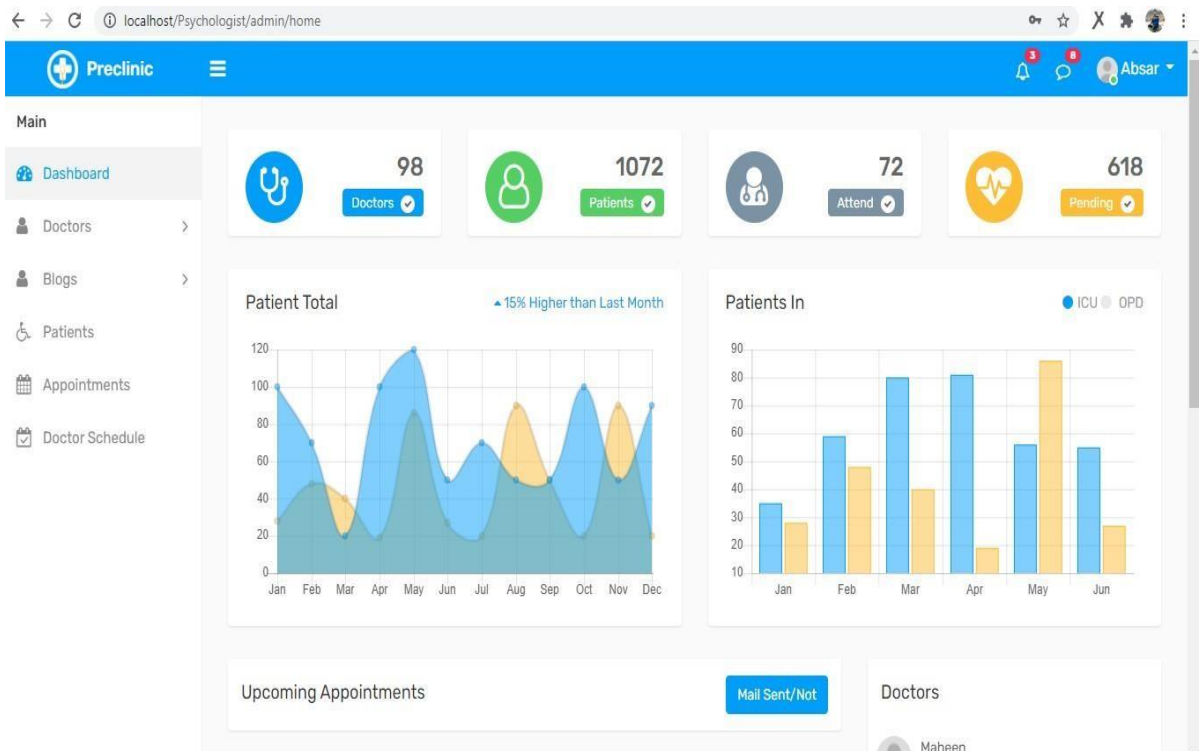


Figure 6.23 Admin Dashboard

6.3.2 Admin Profile

24. Profile

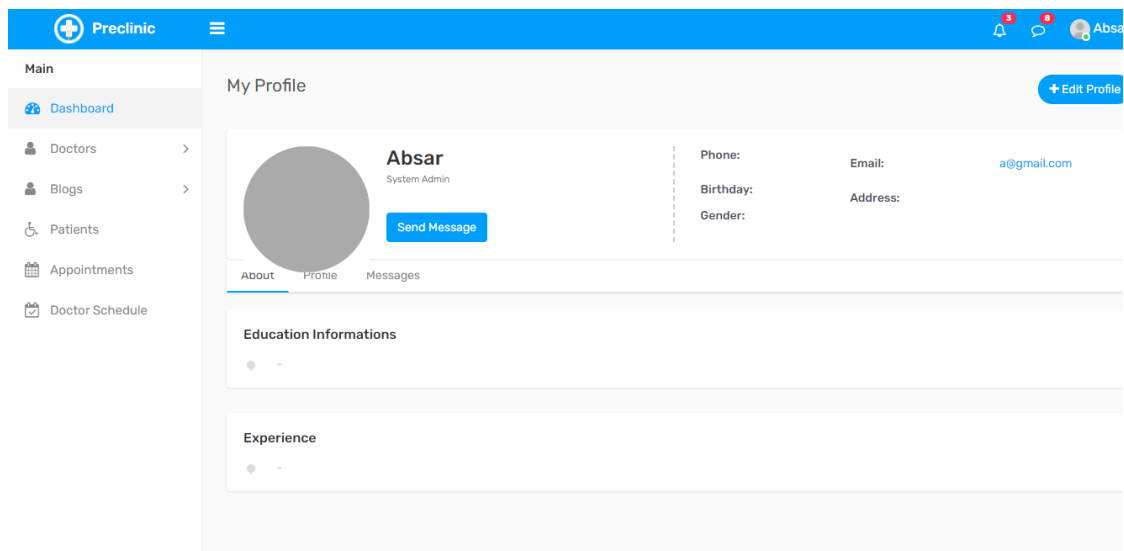


Figure 6.24 Admin Profile

6.3.3 Admin Edit profile

25. Edit form

The screenshot shows the 'Edit Profile' form in the Preclinic system. The form is divided into two main sections: 'Basic Informations' and 'Contact Informations'. The 'Basic Informations' section includes fields for Name (Absar), Email (a@gmail.com), Birth Date (mm/dd/yyyy), Gender (Male), and Profile Image (Choose File). The 'Contact Informations' section includes an Address field.

Figure 6.25 Edit profile

6.3.4 Blog Request

26. Request page

The screenshot shows the 'All Blog Requests' page in the Preclinic system. The page displays a table with columns for Title, Description, Posting Date, and Action. A single request is shown with the title 'Test', description 'Test', and posting date '0000-00-00'. An 'Approve' button is visible next to the request.


Title	Description	Posting Date	Action
 Test	Test	0000-00-00	Approve

Figure 6.26 Blog Request

6.3.5 Doctor Request

27. Request form

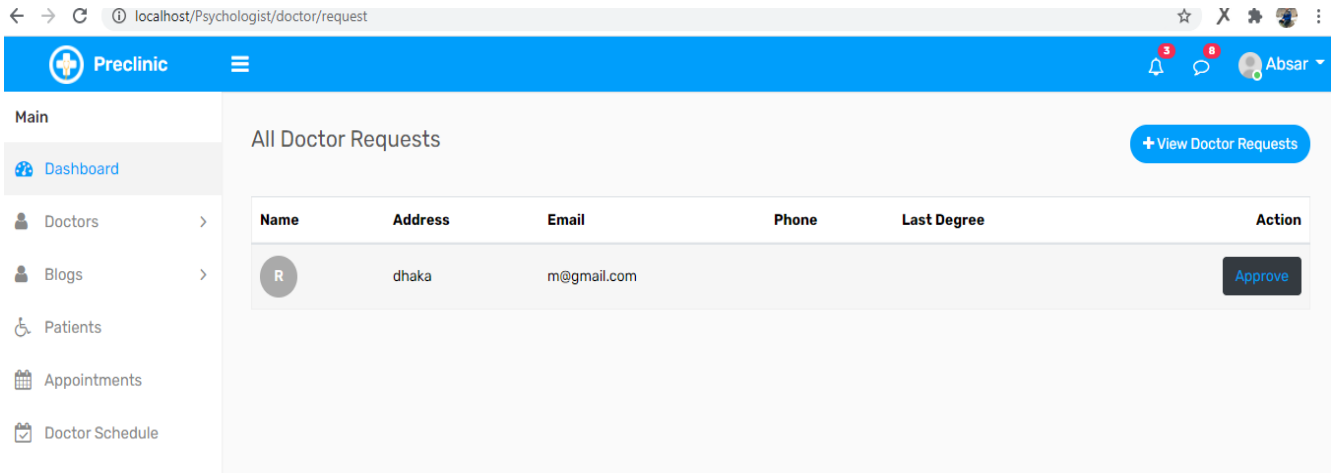


Figure 6.27 Doctor Request

6.3.6 Current Doctor

28. Request form

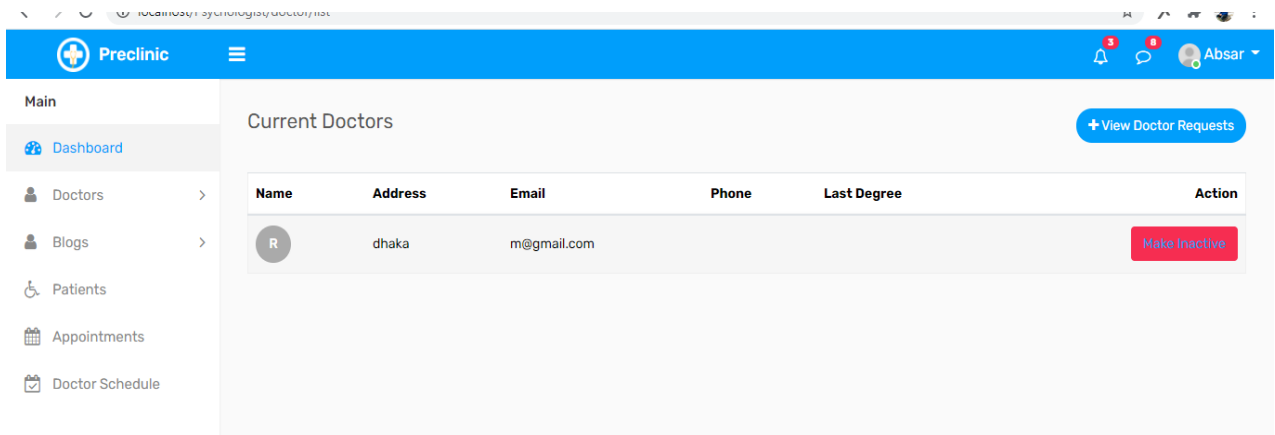


Figure 6.28 Current Doctor

6.3.7 Current Blog

29. Blog page

The screenshot displays the 'Current Blogs' interface. On the left is a sidebar with a 'Main' menu containing 'Dashboard', 'Doctors', 'Blogs', 'Patients', 'Appointments', and 'Doctor Schedule'. The 'Blogs' item is selected. The main content area is titled 'Current Blogs' and features a '+ View Doctor Requests' button. Below this is a table with the following data:

Name	Description	Posting Date	Action
Test	Test	0000-00-00	Make Inactive

Figure 6.29 Current Blog

7. CHAPTER-07 PROJECT SUMMARY

7.1 Limitations

Table 7.1 Summary

Performance	1) Less Performance
Speed	2) Less Speed
Features	1) User cannot go to communication through this system. They have to use 3 rd party software. 2) Doctor cannot take up more than one patient at a time. 3) There have no physical prescription system 4) Doctor cannot enroll to the system without admin approval 5) Survey system is not using AI. So, there have no 100% surety.
Ability	1) Million users can not hit on this system at a time.

7.2 GitHub Link:

<https://github.com/newmaheen/e-psychologist>

7.3 Future Work Plan

- Implement chat
- Implement video call
- Payment system
- Subscription System
- Using AI for Survey
- Profile for patient
- And many more

APENDIX A

```
<?php
namespace App\Http\Controllers;
use App\Admin;
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
class adminProfileController extends Controller
{
    function __construct()
    {
        $this->middleware('auth:admin');
    }
    public function index(){
        return view('admin.profile.add-profile');
    }
    public function updateProfile(Request $request){
        $profileImage = $request->file('image');
        $user = Admin::findOrFail(Auth::id());
        if($profileImage){
            if( $user->image != null && file_exists($user->image)){
                unlink(asset($user->image));
            }
            $imageName = $profileImage->getClientOriginalName();
            $directory = 'public/admin/profile-image/';
            $imageUrl = $directory.$imageName;
            $profileImage->move($directory , $imageName);

            $user->image= $imageUrl;
            $user->phone = $request->phone;
            $user->address = $request->address;
            $user->gender = $request->gender;
            $user->birthday=$request->birthday;
            $user->institution=$request->institution;
            $user->subject=$request->subject;
            $user->start_date=$request->start_date;
            $user->end_date=$request->end_date;
            $user->degree=$request->degree;
            $user->grade=$request->grade;
            $user->company_name=$request->company_name;
```

```

        $user->location=$request->location;
        $user->job_position=$request->job_position;
        $user->period_form=$request->period_form;
        $user->period_to=$request->period_to;
        $user->save();
    }
    else{
//dd('no imGE');
        $user->phone = $request->phone;
        $user->address = $request->address;
        $user->gender = $request->gender;
        $user->birthday=$request->birthday;
        $user->institution=$request->institution;
        $user->subject=$request->subject;
        $user->start_date=$request->start_date;
        $user->end_date=$request->end_date;
        $user->degree=$request->degree;
        $user->grade=$request->grade;
        $user->company_name=$request->company_name;
        $user->location=$request->location;
        $user->job_position=$request->job_position;
        $user->period_form=$request->period_form;
        $user->period_to=$request->period_to;
        $user->save();
    }
    return redirect('/view-profile');
}
public function viewProfile(){
    $data['admin'] = Auth::user();
    return view ('admin.profile.view-profile', $data);
}
}
}

```

```

<?php
namespace App\Http\Controllers;

use App\Appointment;
use Illuminate\Http\Request;

```

```

use Illuminate\Support\Facades\Mail;

class AppointmentController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    public function newAppointment(Request $request)
    {
        $app = new Appointment();
        $app->name = $request->name;
        $app->email = $request->email;
        $app->phone = $request->phone;
        $app->doctor_id = $request->doctor_id;
        $app->date = $request->date;
        $app->time = $request->time;
        // $app->save();
        // return redirect()->route('success');
        $data['title'] = "Doctor Appointment
Confirmation";
        $data['name'] = $app->name;
        $data['date'] = $app->date;
        $data['time'] = $app->time;
        Mail::send('front-end.email', $data,
function($message) use ($app) {

            $message->to($app->email, $app->name)

                ->subject('Doctor Appointment
Confirmation');
            });

        $app->save();

        return redirect()->back()->with('message',
'Appointment Successful! Please check your email for more
info');
    }
}

```

```
}  
}
```

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Appointment;  
use App\Prescription;  
use App\Doctor;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Auth;  
use DB;  
  
class DoctorController extends Controller  
{  
    function __construct()  
    {  
        $this->middleware('auth:doctor');  
    }  
    public function index()  
    {  
  
        $current_doctor = Auth::user()->id;  
  
        $appointment =  
Appointment::where('is_appointed', '=', 0)  
->where('doctor_id', '=', $current_doctor)  
->get();  
  
        return view('doctor.home.home', [  
            'appointments' => $appointment  
        ]);  
    }  
  
    public function edit(){
```

```

        $data['doctor'] = Auth::user();
        return view('doctor.edit', $data);
    }
    public function update(Request $request){
        $profileImage = $request->file('image');
        $doctor = Doctor::findOrFail(Auth::id());
        if($profileImage) {

            if ($doctor->image != null &&
file_exists($doctor->image)) {
                unlink(asset($doctor->image));
            }
            $imageName = $profileImage-
>getClientOriginalName();
            $directory = 'public/doctor/image/';
            $imageUrl = $directory . $imageName;
            $profileImage->move($directory,
$imageName);
            $doctor->image = $imageUrl;
        }
        $doctor->name = $request->name;
        $doctor->address = $request->address;
        $doctor->phone = $request->phone;
        $doctor->last_degree = $request->last_degree;
        $doctor->save();

        return redirect()->route('doctor.profile');
    }
    public function profile(){
        $data['doctor'] =
Doctor::with(['experiences', 'educations'])-
>where('id', auth()->user()->id)->first();
        return view('doctor.profile', $data);
    }
}

```

```

<?php

namespace App\Http\Controllers;

use App\DoctorSchedule;

```

```

use Illuminate\Http\Request;

class DoctorScheduleController extends Controller
{
    function __construct()
    {
        $this->middleware('auth:doctor');
    }
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $data['schedules'] =
DoctorSchedule::where('doctor_id',auth()->user()->id)-
>orderBy('day')->get();
        return view('doctor.schedule.index',$data);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('doctor.schedule.new');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'day' => 'required',
            'time_from' => 'required',
            'time_to' => 'required',
        ]);
    }
}

```



```

        $doctorSchedule = new DoctorSchedule();
        $doctorSchedule->day = $request->day;
        $doctorSchedule->time_from = $request->time_from;
        $doctorSchedule->time_to = $request->time_to;
        $doctorSchedule->doctor_id = auth()->user()->id;
        $doctorSchedule->save();
        return redirect()->route('doctorSchedule.index');
    }

    /**
     * Display the specified resource.
     *
     * @param \App\DoctorSchedule $doctorSchedule
     * @return \Illuminate\Http\Response
     */
    public function show(DoctorSchedule $doctorSchedule)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param \App\DoctorSchedule $doctorSchedule
     * @return \Illuminate\Http\Response
     */
    public function edit(DoctorSchedule $doctorSchedule)
    {
        return view('doctor.schedule.edit', ['schedule'
=>$doctorSchedule]);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param \App\DoctorSchedule $doctorSchedule
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request,
DoctorSchedule $doctorSchedule)
    {
        $request->validate([
            'day' => 'required',
            'time_from' => 'required',

```

```

        'time_to' => 'required',
    ]);

//     $doctorSchedule = new DoctorSchedule();
    $doctorSchedule->day = $request->day;
    $doctorSchedule->time_from = $request->time_from;
    $doctorSchedule->time_to = $request->time_to;
    $doctorSchedule->save();
    return redirect()->route('doctorSchedule.index');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\DoctorSchedule $doctorSchedule
 * @return \Illuminate\Http\Response
 */
public function destroy(DoctorSchedule
$doctorSchedule)
{
    $doctorSchedule->delete();
    return redirect()->back();
}
}

```

```

<?php

namespace App\Http\Controllers;

use App\Doctor;
use Illuminate\Http\Request;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */

    /**
     * Show the application dashboard.
     */
}

```

```

        * @return \Illuminate\Contracts\Support\Renderable
        */
        public function index()
        {
            $data['doctors'] =
Doctor::where('is_doctor', '=', 1)->get();
            $data['message'] = "";
            // $data['doctors'] =
Doctor::where('is_doctor', '=', 1)->get();
            // dd($data);
            return view('front-end.home.home', $data);
        }
        public function success()
        {
            $data['doctors'] =
Doctor::where('is_doctor', '=', 1)->get();
            $data['message'] = "Appointment Successful! Please
check your email for more info";

            return view('front-end.home.home', $data);
        }
    }
}

```

```

<?php

namespace App\Http\Controllers;

use App\Prescription;
use App\Appointment;

use Illuminate\Http\Request;
use Mail;

class PrescriptionController extends Controller
{
    function __construct()
    {
        $this->middleware('auth:doctor');
    }
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response

```

```

*/
public function index()
{
    $data['blog_information'] = Blog::
        where('doctor_id',auth()->user()->id)
        ->get();
    return view('doctor.blogInformation.index',$data);
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    // return
view('doctor.prescription.prescription');
}
public function view($appointment_id)
{
    return view('doctor.prescription.prescription')-
>with('appointment_id',$appointment_id);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'disorder_history' => 'required',
        'prescription' => 'required',
        'test' => 'required',
    ]);

    $prescription = new Prescription();
    $prescription->disorder_history = $request-
>disorder_history;
}

```

```

        $prescription->prescription = $request-
>prescription;
        $prescription->test = $request->test;
        $prescription->is_prescribed = 1;
        $prescription->doctor_id = auth()->user()->id;
        $prescription->appointment_id = $request-
>patient_id;

        $data['title'] = "Doctor Prescription
Confirmation";
        $data['history'] = $request->disorder_history;
        $data['prescription'] = $request->prescription;
        $data['test'] = $request->test;

        $appointment_info =
Appointment::select('email', 'name')
        ->where('id', $request->patient_id)->get();
        $data['name'] = $appointment_info[0]->name;

        Mail::send('front-end_email_prescription', $data,
function($message) use ($appointment_info) {

            $message->to($appointment_info[0]-
>email, $appointment_info[0]->name)

                ->subject('Doctor Prescription
Confirmation');
        });

        if($prescription->save())
        {
            Appointment::where('id', $request-
>patient_id)->update(array('is_appointed' => 1));
        }
        return redirect()->route('doctor-home');
    }

    /**
     * Display the specified resource.
     *
     * @param \App\DoctorSchedule $doctorSchedule
     * @return \Illuminate\Http\Response
    */

```

```

*/
public function show(DoctorSchedule $doctorSchedule)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\DoctorSchedule $doctorSchedule
 * @return \Illuminate\Http\Response
 */
public function edit(DoctorSchedule $doctorSchedule)
{
    return view('doctor.schedule.edit', ['schedule'
=>$doctorSchedule]);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\DoctorSchedule $doctorSchedule
 * @return \Illuminate\Http\Response
 */
public function update(Request $request,
DoctorSchedule $doctorSchedule)
{
    $request->validate([
        'day' => 'required',
        'time_from' => 'required',
        'time_to' => 'required',
    ]);

    // $doctorSchedule = new DoctorSchedule();
    $doctorSchedule->day = $request->day;
    $doctorSchedule->time_from = $request->time_from;
    $doctorSchedule->time_to = $request->time_to;
    $doctorSchedule->save();
    return redirect()->route('doctorSchedule.index');
}

/**
 * Remove the specified resource from storage.
 *

```

```

* @param \App\DoctorSchedule $doctorSchedule
* @return \Illuminate\Http\Response
*/
public function destroy(DoctorSchedule
$doctorSchedule)
{
    $doctorSchedule->delete();
    return redirect()->back();
}
public function blogList(){
    $data['blogs'] = Blog::orderBy('id','DESC')-
>where('is_blog',true)->paginate(15);
    return view('admin.blogs.index',$data);
}
// public function requests(){
//     $data['doctors'] =
Doctor::orderBy('id','DESC')->where('is_doctor',false)-
>paginate(15);
//     return view('admin.doctors.requests',$data);
// }
}

```

```

<?php
namespace App\Http\Controllers;

use App\Appointment;
use App\Doctor;
use App\Blog;
use Illuminate\Http\Request;
use PhpParser\Comment\Doc;

class PsychologistController extends Controller
{
    function __construct()
    {
        $this->middleware('auth:admin');
    }

    public function index(){
        $appointment = Appointment::orderBy('id','DESC')-
>paginate(5);
        $doctor = Doctor::orderBy('id','DESC')-

```

```

>where('is_doctor', true)->paginate(5);

    return view('admin.home.home', [
        'appointments' => $appointment,
        'doctors'       => $doctor
    ]);
}

public function doctorList(){
    $data['doctors'] = Doctor::orderBy('id', 'DESC')-
>where('is_doctor', true)->paginate(15);
    return view('admin.doctors.index', $data);
}
public function blogList(){
    $data['blogs'] = Blog::orderBy('id', 'DESC')-
>where('is_blog', true)->paginate(15);
    return view('admin.blogs.index', $data);
}
public function requests(){
    $data['doctors'] = Doctor::orderBy('id', 'DESC')-
>where('is_doctor', false)->paginate(15);
    return view('admin.doctors.requests', $data);
}
public function requestsBlog(){
    $data['blogs'] = Blog::orderBy('id', 'DESC')-
>where('is_blog', false)->paginate(15);
    return view('admin.blogs.requests', $data);
}
public function approveDoctor($id){
//      dd($id);
    $doc = Doctor::findOrFail($id);
    if($doc->is_doctor == false){
        $doc->is_doctor = true;
    }else{
        $doc->is_doctor = false;
    }
    $doc->save();
    return redirect()->back();
}

public function approveBlog($id){
//      dd($id);
    $doc = Blog::findOrFail($id);
    if($doc->is_blog == false){
        $doc->is_blog = true;
    }
}

```



```

        }else{
            $doc->is_blog = false;
        }
        $doc->save();
        return redirect()->back();
    }
}

```

```

<?php

namespace App\Http\Controllers;

use App\Blog;
use Illuminate\Http\Request;

class BlogInformationController extends Controller
{
    function __construct()
    {
        $this->middleware('auth:doctor');
    }
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $data['blog_information'] = Blog::
            where('doctor_id', auth()->user()->id)
            ->get();
        return view('doctor.blogInformation.index', $data);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('doctor.blogInformation.new');
    }
}

```

```

}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'blogTitle' => 'required',
        'blogDescription' => 'required',
        'date' => 'required',
    ]);

    $blog = new Blog();
    $blog->name = $request->blogTitle;
    $blog->description = $request->blogDescription;
    $blog->posting_date = $request->date;
    $blog->doctor_id = auth()->user()->id;
    $blog->save();
    return redirect()->route('blogInformation.index');
}

/**
 * Display the specified resource.
 *
 * @param \App\DoctorSchedule $doctorSchedule
 * @return \Illuminate\Http\Response
 */
public function show(DoctorSchedule $doctorSchedule)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\DoctorSchedule $doctorSchedule
 * @return \Illuminate\Http\Response
 */
public function edit(DoctorSchedule $doctorSchedule)
{
    return view('doctor.schedule.edit', ['schedule'

```

```

=>$doctorSchedule]);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param \App\DoctorSchedule $doctorSchedule
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request,
DoctorSchedule $doctorSchedule)
    {
        $request->validate([
            'day' => 'required',
            'time_from' => 'required',
            'time_to' => 'required',
        ]);

        // $doctorSchedule = new DoctorSchedule();
        $doctorSchedule->day = $request->day;
        $doctorSchedule->time_from = $request->time_from;
        $doctorSchedule->time_to = $request->time_to;
        $doctorSchedule->save();
        return redirect()->route('doctorSchedule.index');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\DoctorSchedule $doctorSchedule
     * @return \Illuminate\Http\Response
     */
    public function destroy(DoctorSchedule
$doctorSchedule)
    {
        $doctorSchedule->delete();
        return redirect()->back();
    }
    public function blogList(){
        $data['blogs'] = Blog::orderBy('id', 'DESC')-
>where('is_blog', true)->paginate(15);
        return view('admin.blogs.index', $data);
    }
    // public function requests(){

```

```

//      $data['doctors'] =
Doctor::orderBy('id', 'DESC')->where('is_doctor', false)-
>paginate(15);
//      return view('admin.doctors.requests', $data);
// }
}

```

APPENDIX B

Taste Case

Taste Case-01

Table1: Taste Case 1 (Login)

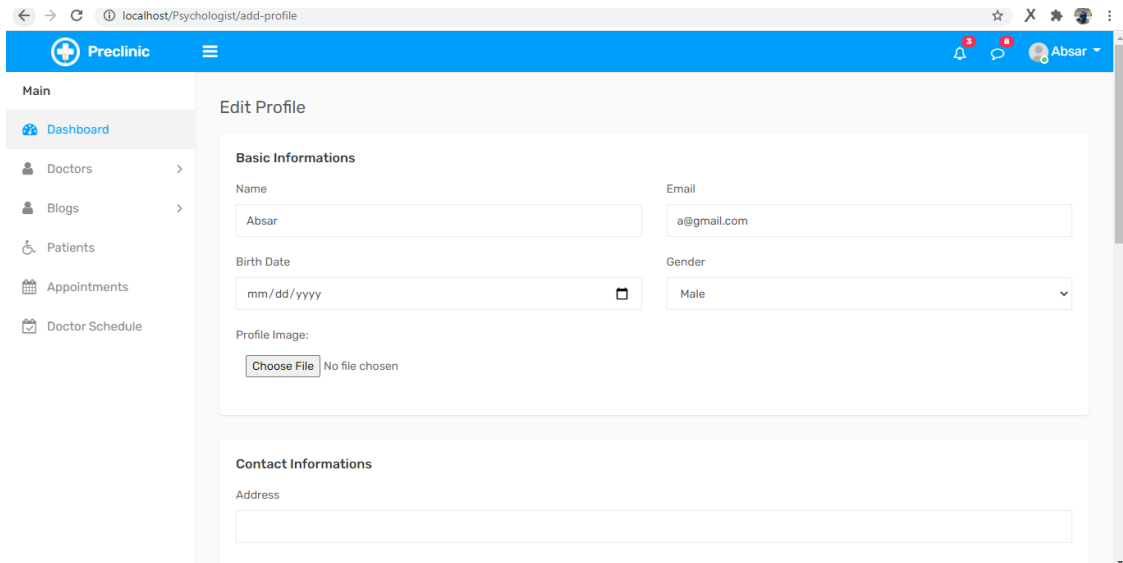
Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Login	Check login with valid data	1. Browse site 2. Input Email 3. Input Password 4. Press the login button	Email: maheen@gmail.com Password: 12345678	Logged in successfully	Pass
TU 02	Login	Check login with invalid data	1. Browse site 2. Input Email 3. Input Password 4. Press the login button	Email: maheen12@gmail.com Password: 12345445	Wrong email/Password and don't login	Pass



Taste Case-02

Table-2: Taste Case 2 (Manage profile)

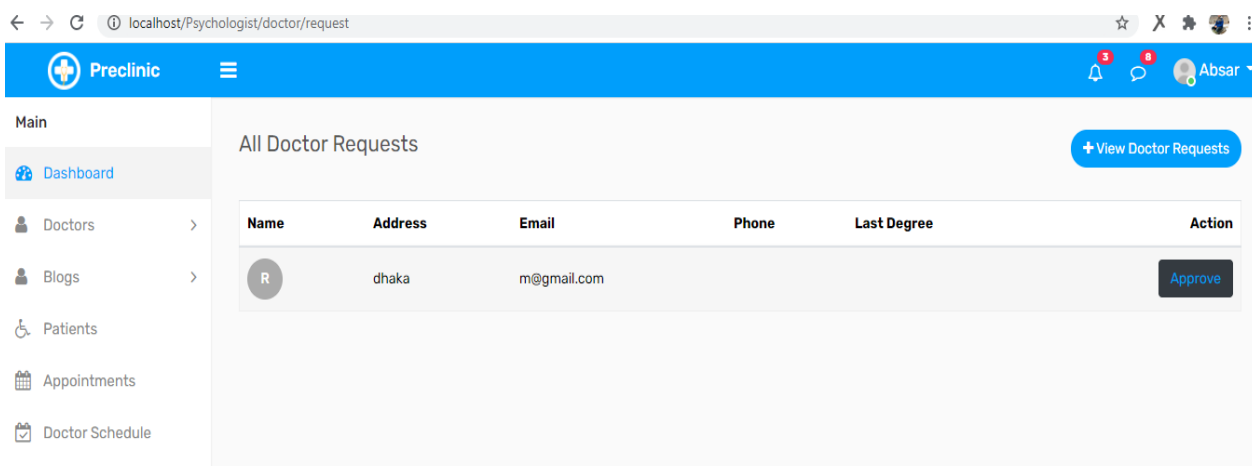
Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Manage profile	Check edit/update with valid data	<ol style="list-style-type: none"> 1. Login 2. My Profile 3. Edit/Update with suitable data 	Degree: MBBS Home: Dhaka Experience: 5 years Etc.	Registration successfully	Pass
TU 02	Manage profile	Check edit/update with invalid data	<ol style="list-style-type: none"> 1. Login 2. My Profile 3. Edit/Update with suitable data 	Degree: MBBS Home: Dhaka Experience: %/*& Etc.	Must be an number form	Pass



Taste Case-03

Table-3: Taste Case 3 (Approve doctor)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Approve doctor	Check Doctor request which contain valid data	1. Login 2. Go to Doctor Request 3. Check info	Name: Absar Degree: MBBS Experience: ok Etc.	Registration successfully	Pass
TU 02	Approve doctor	Check Doctor request which contain invalid data	1. Login 2. Go to Doctor Request 3. Check info	Name: john Degree: FCPS Experience: nil	Empty field	Pass



Taste Case-04

Table-4: Taste Case 4 (Make Appointment)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Make Appointment	Check appointment request which contain valid data	<ol style="list-style-type: none"> 1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit 	Name: Maheen Email: maheen@gmail.com Doctor: Selected Time: Selected	Make appointment successfully	Pass
TU 02	Make Appointment	Check appointment request which contain invalid data	<ol style="list-style-type: none"> 1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit 	Name: Maheen Email: maheen.gmail.com Doctor: Selected Time: Selected	Appointment denied due to false email	Pass

Make an Appointment

Exercitation cupim ex, short ribs cow in ullamco corned beef veniam kevin.
Eu frankfurter ham hock ball tip reprehenderit adipiscing ipsum

Your Name Email Address

Select Doctor Phone

mm/dd/yyyy

Taste Case-05

Table-5: Taste Case 05 (Make a post)

Test case ID	Test case name	Test scenario	Test step	Test data	Expected result	Pass/Fail
TU 01	Make a post	Check appointment request which contain valid data	1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit	Name: Maheen Email: maheen@gmail.com Doctor: Selected Time: Selected	Make appointment successfully	Pass
TU 02	Make a post	Check appointment request which contain invalid data	1. Enter site 2. Go to appointment 3. Fill the box using required data 4. Submit	Name: Maheen Email: maheen.gmail.com Doctor: Selected Time: Selected	Appointment denied due to false email	Pass

Psychologist ☰ 🔔 💬 👤 Maheen

Main

- Dashboard
- Education >
- Experience >
- Schedule >
- Add Information >
- Departments

Add New Blog Information

Title*
Diu

Description*
Daffodil

Posting Date*
05/25/2021

SAVE

Turnitin Originality Report

Processed on: 19-Jun-2021 15:09 +06
ID: 1608975939
Word Count: 10683
Submitted: 1

171-35-1961 By Maheen Absar

Similarity Index 17%	Similarity by Source
	Internet Sources: 16% Publications: 2% Student Papers: 13%

4% match (student papers from 28-Mar-2018) Class: Article 2018 Assignment: Journal Article Paper ID: 937594737
2% match (student papers from 10-Nov-2020) Submitted to nsbm on 2020-11-10
2% match (student papers from 05-May-2021) Submitted to Anglia Ruskin University on 2021-05-05
1% match (Internet from 28-Jul-2019) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2940/P12395%20%285%25%29.pdf?isAllowed=y&sequence=1
1% match (Internet from 05-Jun-2020) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3927/P148856%20%2828%25%29.pdf?isAllowed=y&sequence=1
1% match (student papers from 16-May-2021) Submitted to Arab Open University on 2021-05-16
1% match (Internet from 01-Apr-2020) https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report
1% match (student papers from 13-Jun-2021) Submitted to University of Northampton on 2021-06-13
1% match (Internet from 13-Jun-2021)

1% match (student papers from 13-Jun-2021) Submitted to University of Northampton on 2021-06-13
1% match (Internet from 13-Aug-2019) https://monkeywebstudio.com/laravel-5-7-basic-crud-tutorial/
< 1% match (student papers from 10-Apr-2018) Class: April 2018 Project Report Assignment: Student Project Paper ID: 944219469
< 1% match (Internet from 10-Jan-2020) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3551/P13655%2823%25%29.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 02-Apr-2021) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5355/162-15-8132%20%2811_%29.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 11-Nov-2020) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2176/P13007%20%2824%25%29.pdf?sequence=
< 1% match (student papers from 05-May-2019) Submitted to University of Northampton on 2019-05-05

https://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=10&oid=1608975939&sid=0&n=0&m=2&svr=36&r=10.33709662417115&lang=e... 1/13

6/19/2021	Turnitin
< 1% match (student papers from 04-Jan-2020) Submitted to Bahrain Polytechnic on 2020-01-04	
< 1% match (student papers from 29-Jun-2017) Submitted to Bahrain Polytechnic on 2017-06-29	
< 1% match (Internet from 09-Jun-2021) https://tugasakhir.id/crud-gambar-laravel/	
< 1% match (student papers from 27-Apr-2018) Submitted to Glasgow Clyde College on 2018-04-27	
< 1% match (Internet from 21-May-2019) https://gitlab.solve-x.net/bioles-horizont/CrmBioles/commit/69992cf8a782d06735a3402c967ee865609998bf	

Submitted to Glasgow Clyde College on 2018-04-27
< 1% match (Internet from 21-May-2019) https://gitlab.solve-x.net/bioles-horizont/CrmBioles/commit/69992cf8a782d06735a3402c967ee865609998bf
< 1% match (Internet from 22-Feb-2019) https://quantizd.com/facebook-php-sdk-with-laravel/
< 1% match (student papers from 10-May-2021) Submitted to University of Ulster on 2021-05-10
< 1% match (student papers from 09-Jun-2015) Submitted to University of Greenwich on 2015-06-09
< 1% match (Internet from 15-Nov-2020) https://www.guru99.com/test-case.html
< 1% match (Internet from 17-Dec-2016) http://ro.uow.edu.au/cgi/viewcontent.cgi?article=2016&context=theses
< 1% match (student papers from 30-Aug-2007) Submitted to University of Sheffield on 2007-08-30
< 1% match (Internet from 10-Jan-2021) https://www.coursehero.com/file/67301064/Lab-Assignment-1pdf/
< 1% match (student papers from 29-Sep-2020) Submitted to University of Wales Swansea on 2020-09-29
<p>TITLE OF THE PROJECT E-psychologist.com Submitted By Maheen Absar (ID: 171-35-1961) Supervised By Mr. Khalid Been Badruzzaman Biplob Lecturer (Senior Scale) Faculty of Science and Information Technology Department of Software Engineering Daffodil International University Fall-2020 A project (SWE 431) submitted in fulfillment of the Requirements for the degree of BSc in Software Engineering i © Daffodil International University APPROVAL i © Daffodil International University DECLARATION I, the author, hereby declare that the project becomes the property of Daffodil International University (DIU). I give permission to the DIU that they can change in whole or in part for the purpose of research or academic exchange only. Supervised By Mr. Khalid Been Badruzzaman Biplob HASH(0x7fd04d3960e8)-1961 Department of Software Engineering Daffodil International University ii © Daffodil International University ACKNOWLEDGEMENT At first, HASH(0x7fd04d373208) project by keeping me mentally strong and physically fit. I am also very grateful to my parents who always inspire me to do good things, study attentively and do something for my country. They inspired me a lot to complete this project during this hard situation of covid-19. My teachers helped me a lot to achieve the qualifications to</p>

Accounts Clearance

