

**PROJECT REPORT**  
**ON**  
**MEDICATE: A Mobile Application to Automate**  
**Patient-Doctor-Hospital Relationships**



**DAFFODIL INTERNATIONAL UNIVERSITY**  
**DHAKA, BANGLADESH**  
**OCTOBER 2020**

**MEDICATE: A MOBILE APPLICATION TO AUTOMATE  
PATIENT-DOCTOR-HOSPITAL RELATIONSHIPS**

**BY**

**FARHANA BINTA SHAHEED**

**ID: 162-5-734**

**AND**

**FAISAL KABIR**

**ID: 162-15-768**

**AND**

**ASIF AL MAHMUD**

**ID: 162-15-771**

This Report Presented in Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

**Md. Nazmul Hoq**

Lecturer & Associate Head

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**OCTOBER 2020**

## **APPROVAL**

This project titled “**MEDICATE: A Mobile Application to Automate Patient-Doctor-Hospital Relationships**”, submitted by Farhana Binta Shaheed (ID 162-15-734) and Faisal Kabir (ID 162-15-768) and Asif Al Mahmud (ID 162-15-771) to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation was held on Wednesday, 08 October, 2020.

## **BOARD OF EXAMINERS**

---

**Dr. Syed Akhter Hossain**  
**Professor and Head**

**Chairman**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

---

**Mr. Narayan Ranjan Chakraborty**  
**Assistant Professor**

**Internal Examiner**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

---

**Mr. Md. Nazmul Hoq**  
**Lecturer**

**Internal Examiner**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

---

**Dr. Mohammad Shorif Uddin**  
**Professor and Chairman**

**External Examiner**

Department of Computer Science and Engineering  
Jahangirnagar University

## DECLARATION

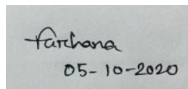
We hereby declare that this project has been done by us under the supervision of **Md. Nazmul Hoq, Lecturer & Associate Head, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

### Supervised by:

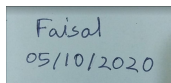


-----  
**Md. Nazmul Hoq**  
Lecturer & Associate Head  
Department of CSE  
Daffodil International University

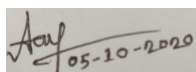
### Submitted by:



-----  
**Farhana Binta Shaheed**  
ID: 162-15-734  
Department of CSE  
Daffodil International University



-----  
**Faisal Kabir**  
ID: 162-15-768  
Department of CSE  
Daffodil International University



-----  
**Asif Al Mahmud**  
ID: 162-15-771  
Department of CSE  
Daffodil International University

## ACKNOWLEDGMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Md. Nazmul Hoq**, Lecturer and Associate Head, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Mobile and Web Application*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Syed Akhter Hossain**, Head, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

## **ABSTRACT**

Nowadays people depend on virtual creation through the phone and the internet. This extensive use of mobile phones creates scope for different mobile applications. Mobile applications have started a revolution, in that they have completely changed the way we live. Actually, in the project, we are trying to build one application that serves various purposes. Our application is mainly focused on automating the medical management system and pharmacy management system. We build both the Android and iOS versions of our application. The main stakeholders of our app are patients, doctors, pharmacy managers, delivery men, etc. Any user can login and search for doctors, search for pharmacies, search for blood donors, and order medicine from home. Our application will make life easier and spontaneous.

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Title Page	i
Approval	ii
Board of examiners	iii
Declaration	iv
Acknowledgments	v
Abstract	vi
List of abbreviations	vii
List of Figures	xii
 <b>CHAPTER 1: INTRODUCTION</b>	 <b>1-3</b>
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcomes	2
1.5 Project Management and Finance	2
1.6 Report Layout	3



## **CHAPTER 2: BACKGROUND**

**4-10**

2.1 Introduction	4
2.2 Preliminaries and Terminologies	4
2.2.1 Software Development Life Cycle	4
2.2.2 Phase of Software Development Life Cycle	6
2.2.3 System Study/Initial Idea	7
2.2.4 Feasibility Study	8
2.3 Related Works	9
2.4 Comparative Analysis	9
2.5 Scope of the Problem	9
2.5.1 List of Actors	9
2.5.1.1 Medical	10
2.5.1.2 Doctor	10
2.5.1.3 User	10
2.5.1.4 Pharmacy	10
2.5.1.5 Delivery man	10
2.5.1.6 Recognizing multiple viewpoint	10

## **CHAPTER 3: REQUIREMENT SPECIFICATION**

**11-18**

3.1 Introduction	11
3.2 Business Process Modeling	11

3.3 Use Case Modeling and Descriptions	12
3.4 Logical Data Model	13
3.5 Design Requirements	14
3.5.1 Permission Requirement	14
3.5.2 Threats of System Security	15
3.5.3 Admin Area	16
3.5.4 Database Security	18
3.6 Conclusion	18
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>19-42</b>
4.1 Front-end Design	19
4.1.1 Design & Prototyping	19
4.1.2 User Interface Prototyping	19
4.2 Back-end Design	20
4.2.1 Describe Database design	20
4.3 Interaction Design and User Experience (UX)	20
4.4 App start	20
4.5 Post	21
4.5.1 Add News	21
4.5.2 Add Post	21
4.6 Send Request	22
4.7 Online Bidding	23
4.8 Blood Donation	23

4.9 Ambulance Service	23
4.10 Real-Time Delivery Update	23
4.11 Search Engine	24
4.12 Final Output Look	24
4.13 Implementation Requirements	40
<b>CHAPTER 5: IMPLEMENTATION &amp; TESTING</b>	<b>43-46</b>
5.1 Introduction	43
5.2 Implementation of Database Design	43
5.2.1 Database	43
5.2.2 Security Testing	43
5.2.3 System Testing	44
5.3 Implementation of Front-end Design	45
5.3.1 Automate User Interface tests	46
5.4 Testing	46
5.5 Test results and Reports	46
<b>CHAPTER 6:IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY</b>	<b>47-48</b>
6.1 Impact on Society	47
6.2 Impact on Environment	47
6.3 Ethical Aspects	47
6.4 Sustainability Plan	48

<b>CHAPTER 7: CONCLUSION &amp; FUTURE SCOPE</b>	<b>49</b>
6.1 Discussion and Conclusion	49
6.2 Scope for Further Developments	49
<b>REFERENCE</b>	<b>50</b>
<b>APPENDICES</b>	<b>51-55</b>
<b>PLAGIARISM RESULT</b>	<b>57-57</b>

## LIST OF FIGURES

FIGURES	PAGE NO
Figure 2.1.1: Software Development Life Cycle	6
Figure 2.2.2: Phases of the Software Development Life Cycle	7
Figure 3.2: Business Process Modeling	12
Figure 3.3: Use Case Modeling	13
Figure 3.4: Logical Data Model	14
Figure 3.5.3: Admin Area	17
Figure 4.5.1: Add News Database	21
Figure 4.5.2: Add Post Database	22
Figure 4.6: Send Request Database	22
Figure 4.12.1: The First Outlook of the Application	25
Figure 4.12.2: Sign Up	26
Figure 4.12.3: Navigation Drawer	27
Figure 4.12.4: Home Page	28
Figure 4.12.5: User Profile	29
Figure 4.12.6: Medical Profile	30
Figure 4.12.7: Medical Service	31
Figure 4.12.8: Add Post	32

Figure 4.12.9: Bidding for Medicine	33
Figure 4.12.10: Get Bidding Notifications from Pharmacies	34
Figure 4.12.11: Accept the Request	35
Figure 4.12.12: Click on the Notification Button	36
Figure 4.12.13: Add Delivery Man	37
Figure 4.12.14: Accept the Delivery Request	38
Figure 4.12.15: Real-time Delivery Process	39
Figure 4.12.16: Real-time Processing System	40
Figure 5.2.3: Testing Pyramid	45

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Introduction**

A mobile application helps us to get news very quickly. People like to make their life comfortable through the internet and mobile. So we tried to make an application for our medical sector that will actually help the people for their every need. When it comes to health people get very much conscious and our application will serve the purpose of facilitating the health sector properly.

“What is Medicate?” Medicate is a Medical Assistant application which is actually an android and iOS application for searching doctors, medicals, pharmacies, blood donors, etc. It also helps people in getting knowledge about their location. The application also shows information about different health cares and health tips.

### **1.2 Motivation**

Suppose suddenly a person is not feeling well or in an emergency case he needs blood, or he needs to purchase medicine, but he can't find it nearby. It is known that during this pandemic situation of corona going outside is not safe at all.

Keep these things in our mind we tried to build the application so that finding doctors, donors, blood groups will be very easy for all users. The main target of the project is to help the aged people who aren't able to go to the pharmacy or the medical when it is urgent. Moreover, the project is built for every user indeed. The app is very user-friendly and understandable for all. We tried to build the app very simple for all users and made sure they can understand each feature perfectly.

### **1.3 Objectives**

In our country, the medical sector is improving day by day. But the number of medical apps are so limited. The primary purpose of making the application is to provide a user-friendly platform to all the users, doctors, pharmacy, and the medicals. In our daily lives, most of the people face the same problem as when we need a doctor, we don't know which medical doctor will be available or when he will come. One other problem is that there is no blood donor available by our side when we need any blood donor. So these problems are very considerable and for these, we build the application. The doctors also share some health tips and news through the application. One main thing is that one can find nearby pharmacies, doctors, medicals, and blood donors using the same app.

### **1.4 Expected Outcome**

Before developing Medicate, we draw a scenario in our mind of what will be the outcome of the application. We wanted to build a platform where the doctors and the patients can communicate, the users get medicine and blood groups easily, users can order medicine from home. We worked on these things and fortunately the outcome is so satisfying. All the outcome of the application is as appropriate as we want.

### **1.5 Project Management and Finance**

This project is developed using the Dart Programming Language, a framework of Flutter. Flutter is a cross-platform which can be used to create Android, IOS, and Web applications.

Medicate app has four kinds of users. They are general users, doctors, medical, and pharmacy. Every user must give some information to get registered in the app. While the registration process the user should select the category among the four types. Once the registration process is done then the user has access to the app.



The general user can post their prescription. The pharmacy user can see the post and contact the general user whether they can deliver the medicine. After getting registered the doctors and medicals can see who is using the app. The medical can send requests to the doctors by seeing their information. After getting the request, if the doctor wants to join the medical, the doctor can accept it. Then the database of the doctor will add to the medical. When the doctor is added the general user can see and also take the medical appointment according to the doctor's schedule. To maintain all the procedures financial support must be needed.

## **1.6 Report Layout**

The project report is organized into 7 chapters. Chapter 1 describes the introduction, project objectives, expected outcomes, and project finance of the application. Chapter 2 describes the background of the whole system. Chapter 3 contains the requirement and specifications of the project. Chapter 4 describes the design specification. Chapter 5 narrates the system test & the system result of this project. Chapter 6 describes the impact on society, environment, and sustainability. Finally Chapter 7 includes the full project with conclusion, limitation & plan of our project as well.

## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 Introduction**

In this chapter, we try to give an overall idea and knowledge about the background of the full project. We try to update the medical and pharmacy system by the app, and there is a need for some data.

To develop the app more and take it to the next level a lot of developers and researchers are needed. It's actually a big project and mainly it's useful. If we can cover a huge place with the app then its market value will be increased. Then the actual purpose of the project will come in reality.

When we decided to do the project there were many difficulties and obstacles we had to face. But we tried hard to complete our project but the area of work is very short. In the future, we will try to overcome all the difficulties and will bring the project in large.

#### **2.2 Preliminaries and Terminologies**

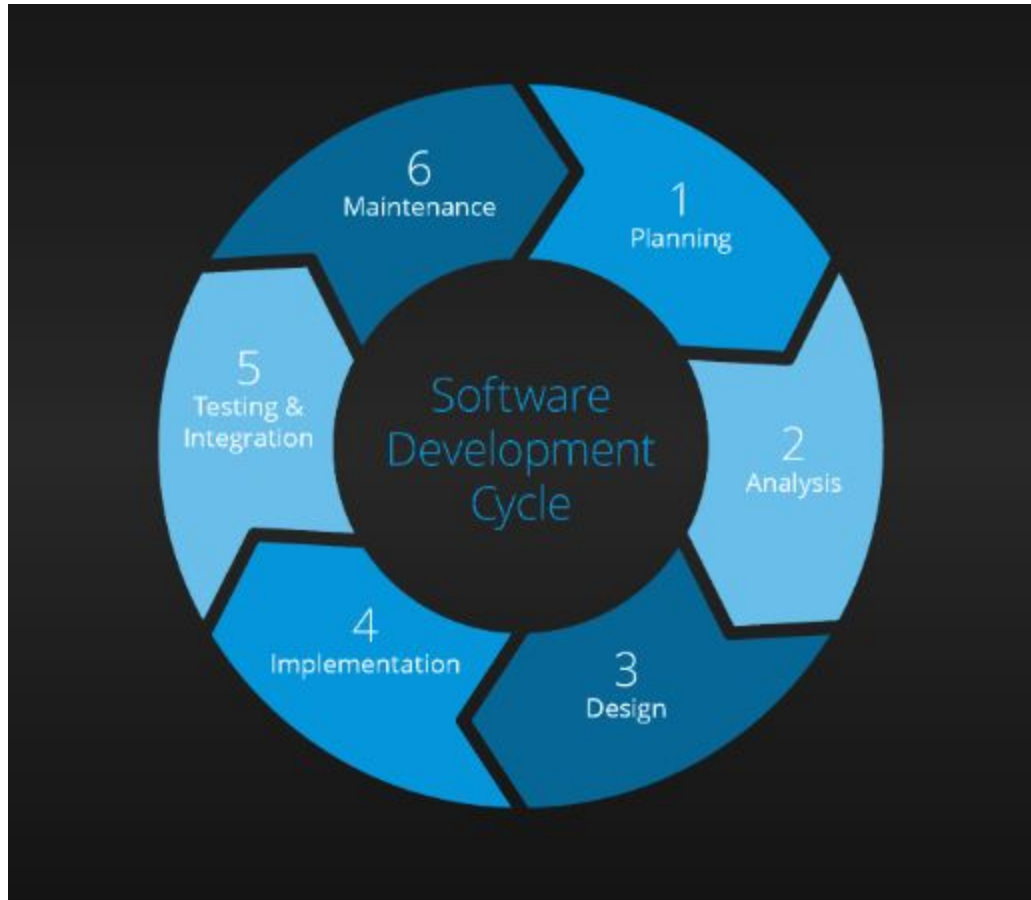
##### **2.2.1 Software Development Life Cycle**

Software Development Life Cycle(SDLC) is a requisite task. To meet or exceed customers' explanations, reach completion within time and cost estimates SDLC is needed. SDLC is a process for designing, testing, and maintaining the software system. So an efficient SDLC will help us to maintain our project and take it to the next level.

To deliver an expected product all the team members should work systematically. But if one of the developers wants to design first, one wants to code and another wants to meet requirements then it will be a great mess and the expected software will be a failure. So the team members should maintain SDLC and have good knowledge.

Following are the different phases of the software development cycle:

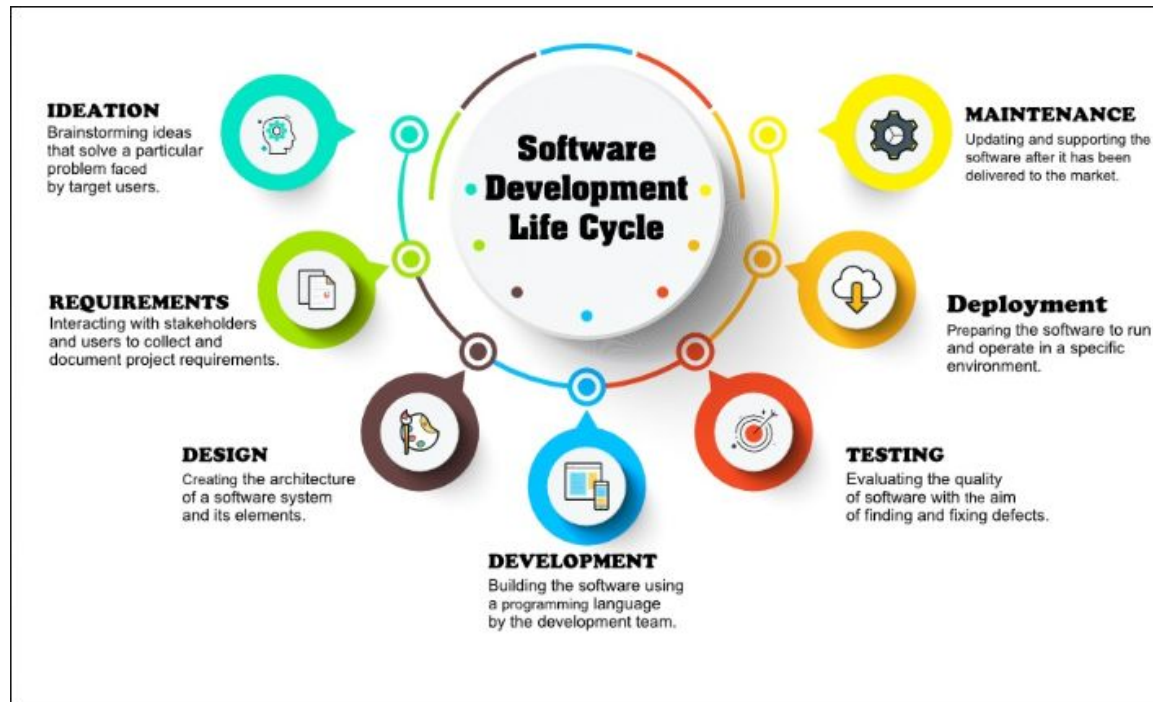
- Preliminary study
- Feasibility study
- Detailed system study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance
- Review



**Figure 2.2.1:** Software Development Life Cycle

### **2.2.2 Phases of the Software Development Life Cycle**

In this section here we will discuss the different phases of SDLC and their details. We will get to know about each phase better by the description. Here the figure 2.2 that will show the system development life cycle detail.



**Figure 2.2.2:** Phases of the Software Development Life Cycle

### 2.2.3 System Study/Initial Idea

System study is the initial stage of the System Development Life Cycle. This stage helps us to know what we want to do in our project and what will be the outcome. When anyone wants to make something initially, he finds some idea and then executes this before executing the project's initial idea. Like before making Medicate we thought that we want to make it and what the features we want in our app, what will the result and what is the purpose. So all these things are the initial idea of our project.

We would say that system study phases through the following steps to describe the initial study more effectively:

- Problem identification & project initiation
- Background analysis

- Interface of findings

#### **2.2.4 Feasibility Study**

To elaborate the Initial idea feasibility study is an imposition of our project. The main purpose of the feasibility study is to when we initiate our idea then what is the strength and weakness, what can the merits and demerits in the natural environment, and how justified in terms of economic and organizational terms. As well as there are some more feasibilities like legal feasibility, time feasibility, operational feasibility, resource feasibility, and financial feasibility to enrich our project further.

There are many influential things in our project. It will help to connect people in their crisis moments like finding donors or hospitals. So the future will be very effective and useful if the analysts take it to the high point. The platforms we have chosen for my project are the android studio, Dart programming language, and MySQL to store databases.

There are some steps in the feasibility study to develop our project and maintain it. Do follow the following steps and it will be all clear about the app to the client and the analyst.

- Project Description
- Goals
- Timeline
- Cost and Budgeting
- Purpose
- Market Analysis
- Resource

## **2.3 Related Works**

In the Play Store or App Store, we can find some apps for blood donation, search nearby hospitals, pharmacies or doctors, and many like this. But the main fact is that all the activities are separate. But in our app, we try to combine all the things for the user. It will be very useful if they can find many things on one platform.

In our app, we try to put some new things like news reading and sharing so it will be very useful for the user. Now we find the news for health in a newspaper app, but we have our own. So people can find many solutions in Medicate and that's the plus point for us.

## **2.4 Comparative Analysis**

In the Play Store or App Store, we can find some apps for blood donation, search nearby hospitals, pharmacies or doctors, and many like this. But the main fact is that all the activities are separate. But in our app, we try to combine all the things for the user. It will be very useful if they can find many things on one platform.

In our app, we try to put some new things like news reading and sharing so it will be very useful for the user. Now we find the news for health in a newspaper app, but we have our own. So people can find many solutions in Medicate and that's the plus point for us.

## **2.5 Scope of the Problem**

While establishing the project we need to work with the following factors and related to the scope of problems

### **2.5.1 List of Actors**

#### **2.5.1.1 Medical**

Admin from the medical hospitals can see the doctors list and can send them requests to join their hospital. They can view and give the news.

#### **2.5.1.2 Doctor**

Doctors can share advice or news to all users. They can get requests from the hospital. If they like the hospital management system and schedule, then they will accept it or ignore it.

#### **2.5.1.3 User**

Users can see the doctors list and nearby medical or pharmacy. They can also get registered as blood donors. So users play a vital role in the system. They can also share their prescription to the pharmacy.

#### **2.5.1.4 Pharmacy**

The pharmacy can get notifications and can add news. They can also tell the user which medicines are available or not.

#### **2.5.1.5 Delivery Man**

For the benefit of the customers, there is an option for a delivery system. Actually, each pharmacy can have its own delivery man for the delivery system. The delivery man picks up the parcel and drops it to the accurate location. So the customers or buyers can order from home and get the delivery in the door.

#### **2.5.1.6 Recognizing Multiple Viewpoint**

Different actors have different work and different platforms. So sometimes they can read the same thing and sometimes not. But their wants is the same and that is:

- User-friendly System
- Error friendly System
- Get result what they want.



## **CHAPTER 3**

### **Requirement Specification**

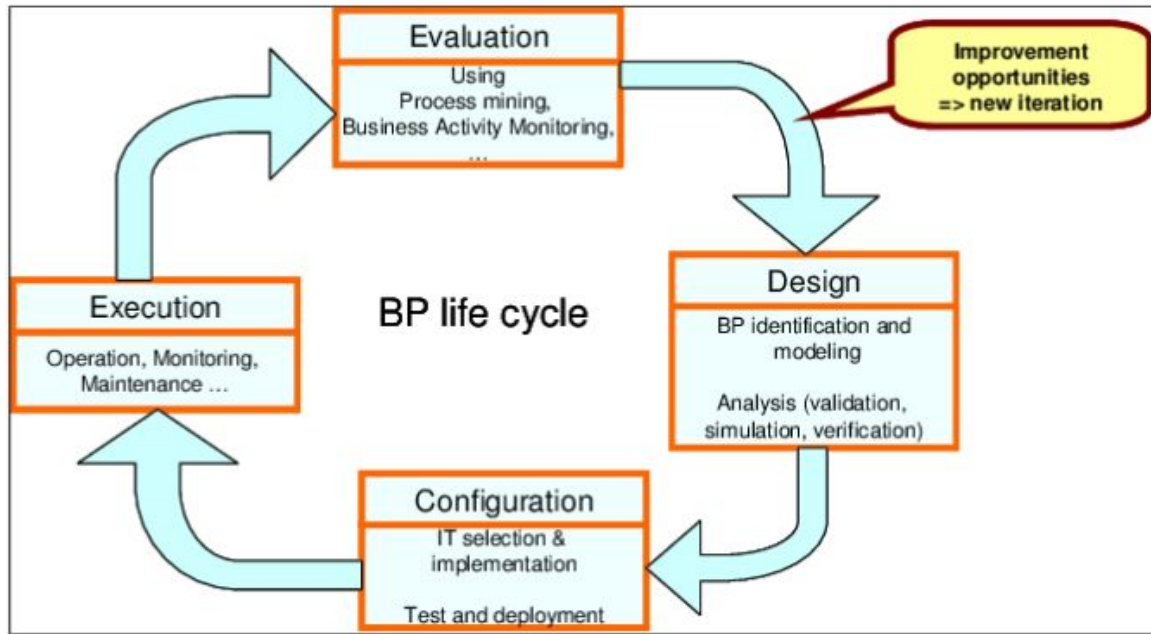
#### **3.1 Introduction**

For every project, there are some requirements for development. So our project Medicate has also some requirement specifications. As we already know that medicate is an Android and iOS Application, so we just need a smartphone or tab. We know that there are many Operating Systems available in the global market. Among them, Android, iOS, Windows, and blackberry are very popular. But in the initial stage, our project Medicate will work on Android and iOS Operating Systems. The user must have an internet connection on the device to run the application. So the user must have all these requirements. There are some other requirements for the admin panel. The admin must have enough knowledge about the database and maintenance of the doctor's information. Our future goal is to work with the project so vastly so it's not that easy. So the admin will also need vast knowledge. For the project, we use our hosting for the database. There is no limit for storage and for this could be an advantage for further use. In the future, many things may change, and we may also change our requirements on user demand.

#### **3.2 Business Process Modeling**

The business process model represents the graphical representation of a business process. If a project wants to business then there are two possible ways for business process modeling. One is the 'As-is' state which is used for the state of the process as it is right now without making any change and the other is the 'To-be' state which is used for future process.

Figure 3.2 shows the idea of the business process modeling and cycle.

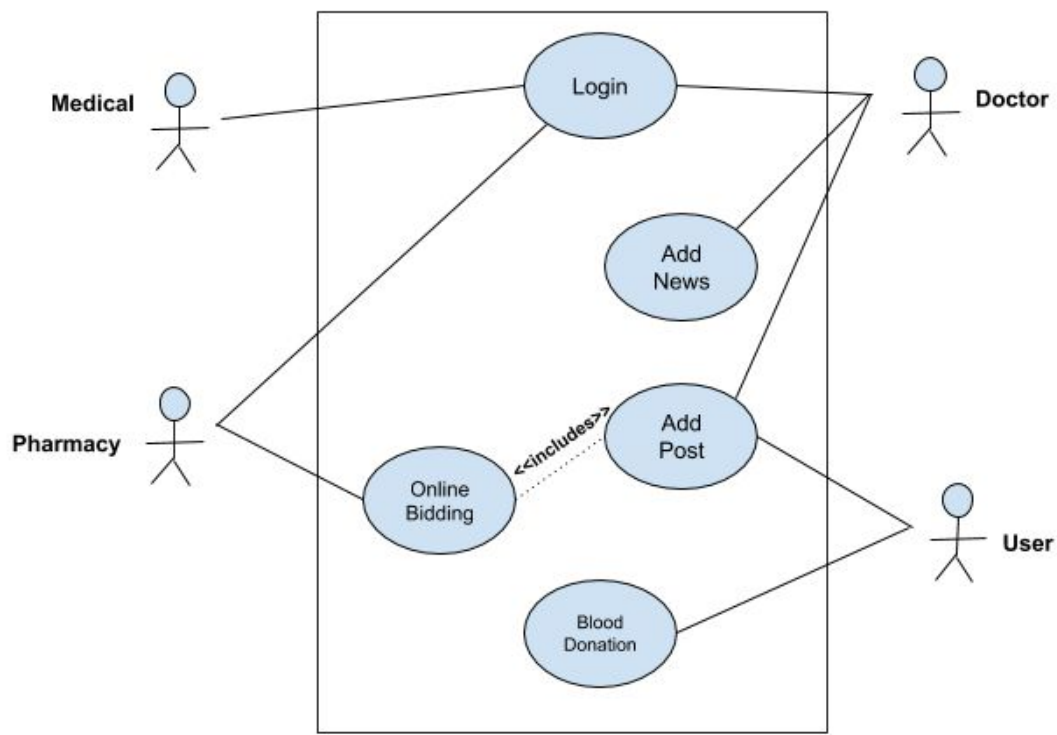


**Figure 3.2:** Business Process Modeling

### 3.3 Use Case Modeling and Descriptions

A use case model is a model where many actors can interact with each other and can provide a complete functional and technical view of the system. In this project, there are some actors like doctors, users, pharmacies, and blood donors. How they connected to each other and with the whole system that can be seen by the use case model.

Figure 3.3 shows the use case model of the project on a small scale.

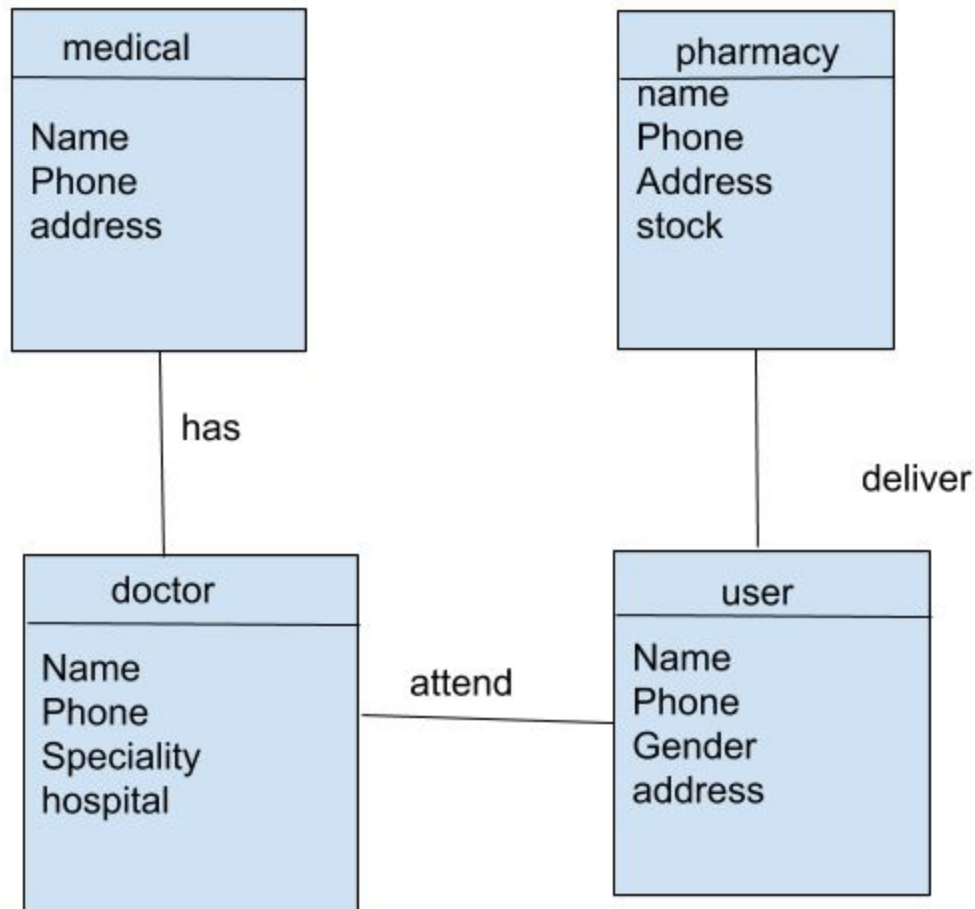


**Figure 3.3:** Use Case Model

### 3.4 Logical Data Model

The logical data model is used to identify the structure of data elements and to set the relation between them. It is a logical data model that does not represent the physical structure.

Figure 3.4 shows the logical data model of the project.



**Figure 3.4:** Logical Data Model

### 3.5 Design Requirements

#### 3.5.1 Permission Requirement

To use any Android or iOS application, the user must need to provide some permission. One thing is very important and that is, Android has launched a total of 16 versions. Android (A-P). From Android M (Marshmallow) there is a new option added which is called android runtime permission. But the lower version from android' this option was not available. When any user was installing an app then all permission was automatically granted. But from Android M, users must provide all permission individually. And all the state has followed perfectly in Medicate.

The set of permissions are need for using Helpmate is listed below:

- Internet Permission
- Read Internal Storage
- Write Internal Storage
- Access user's COARSE Location
- Access user's FINE Location
- Call Permission
- Read Wi-Fi state permission
- Read phone state

If any user rejects any permission any of these then the particular feature will not be available for that user.

### **3.5.2 Threats to System Security**

In 2020 most people are using smartphones. As the users are increasing so the number of threats is also increasing day by day. The main concern of a project is the security system. If we want to make it secure then we have to try our best to overcome the threats.

- It is the most worrisome threat to enterprise security in 2019. Actually, in our project, there are some datasets. So we have to keep the dataset safe and secure. The data security system must be very strong and unfortunately, if any hacker wants to take access to the database then we must utilize encryption.
- There is another thesis that is if the admin of the hospital is not experienced about the system and the database of the doctors then he would not able to add any doctor to their list. If any fake doctor got access to the hospital, this will be a very big mess. On the other hand, when any doctor receives any request from any hospital, he must also know about the hospital.
- Creating and spreading rumors are very common nowadays. Actually fake users are doing this and for this general people get scared very easily. So we have to keep concerned about the matter and ban fake users from the app.

- We should also stop the negative use of the application. If a blood donor wants to give blood, we must make sure that the blood is used for the right purpose. Then how's the behavior of the donor or is there any disprove intention.

### **3.5.3 Admin Area**

Admin plays the most significant role in the system. Admin has all the right to observe all the points of all the users. Even the admin has the power to deactivate or delete any account at any time. The very first stage of the project is started by the admin. The rest of the users must get registered by giving some valid information depends on their term. When a doctor wants to get registered then he must give valid information about his degree and career with his basic info. Like the doctor, the hospital, and the pharmacy also need to give some valid information. The normal people must give the name and a valid email to get registered. If the admin finds any doubt with any of the information then he can do anything with the user id.

Admin can see all the data and activity of all the users. If he finds that any of the users are creating violence or does not maintain all the terms and conditions, he can block him. So admin is the most powerful person in the app. We already know that the pharmacy can deliver medicine to the customer but if any customer gets medicine without the expiry date then he can report it to the admin.

If any user complains about any hospital or doctor and the admin found it valid then he can take action against the hospital or doctor.

id:	82
name:	"Faisal Kabir"
username:	"faisalkabir"
email:	"faisalkabir95@gmail.com"
firebase_id:	null
phone:	"01764993083"
gender:	"Male"
image:	"http://faisal.thesoftking.com/Medicate/assets/images/user/1573248948_faisalkabir.jpg"
category:	"medical"
blood_group:	"A+"
is_bld_dnr:	"1"
phone_verify:	"0"
email_verify:	"0"
status:	"1"
address:	"Uttara, Sector 11"
location:	null
city:	"Dhaka"
zip_code:	"1230"
country_id:	null
date_of_birth:	"20-10-1995"
nid:	null
doc_verify:	"0"

**Figure 3.5.3:** Admin Area

In the image, there is an option called status.

If status = 1; then the user can use his account after getting registered and can keep using it.

If status = 0; then the user can not use his id because the admin deactivated his id.

If status = -1; then the user id got deleted by the admin and there will be no chance to get back that id.

#### **3.5.4 Database Security**

In the project, we use only an online database and for this database, we use the MySQL database. While using the database there are security issues. For security purposes, we use the rest API. Nowadays, security is the most important thing, and we try to maintain security facts. Without getting registered no one can see any post or can't find anything. So for security purposes, we keep proper registration.

#### **3.6 Conclusion**

When all requirement specification is done hopefully, our application will work purely. We tried to keep the medical system very quickly to the aged people through the application. Medicate is that kind of application which will be needed in our daily life. So if we can work on this sincerely the future scope will be very bright.



## **Chapter 4**

### **DESIGN SPECIFICATION**

#### **4.1 Front-end Design**

##### **4.1.1 Design & Prototyping**

Nowadays smartphones are a part of our life. The users vary from age to age. So before designing the app we must keep one thing in our mind that whatever we try to make it must be very simple for all kinds of users. The user interface(UI) is the point where the device and the user interact and communicate. So base on user experience we try to keep it simple.

As our mother tongue is Bengali, so we keep two languages English and Bengali. The user can choose whatever he likes. Even there are two moods that are light mood and dark mood. Depend on the comfort of their eyesight they can change. Mainly we try to build the design so user-friendly.

##### **4.1.2 User Interface Prototyping**

Prototyping is a process of building a model before making a system. Every type of experienced artificer and engineer trying to build a prototype before making their project. It is actually a system development method. There are some general prototype industry processes by which we are following to build our project.

- Requirement Gathering
- Quick Design
- Building Prototype
- Engineer Product
- Refining Prototype

- Customer Evaluation.

The prototype is mainly used to reduce time and cost, improved, and increased user involvement. For successful prototyping identifies and corrects problems long before they can negatively impact and irreparable harm to the final finished project.

## **4.2 Back-end Design**

### **4.2.1 Describe Database Design**

Database design is the organization of all data of a project according to a database model. What data need to store and how the data interrelated with each other is known by the database design. in Medicate there are 27 tables of data. There are different name for each data table and in the table, there are stored data.

### **4.3 Interaction Design and User Experience (UX)**

Interaction design is a design that interacts between users and products. Interaction design enables the user to achieve their objectives in the best way possible.

User experience is the process of supporting user behavior through usability, usefulness, and desirability provided in the interaction with a product.

To deliver the best service to the user both interaction design and user experience are very essential.

### **4.4 APP Start**

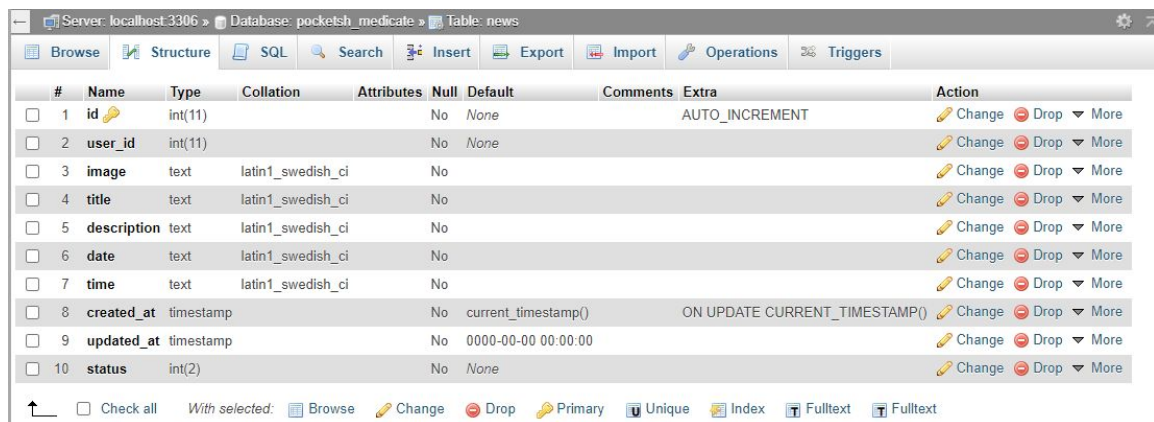
We try to make an application when facing any problem, or we want to make that easy. When \we started to develop the app there are sequences from first to last. Actually, for building any app these sequences are necessary. At first, we need to think about why and for whom we are developing this. Then we must give that a design which is user-friendly. We must select the right approach for selecting the app. There are many types of approaches so the developers must choose this according to their preference. Selecting an appropriate tool is very important. We use flutter as a tool in our project. After that, we

need to apply for the test and it's very necessary. When we saw that the output of the application is matching our expectations then we will be successful.

## 4.5 Post

### 4.5.1 Add News

In our application, there is an option called 'Add News'. All the types of users can add any news. For adding news one must give the title and write the description. To grab attention, he can also add an image related to the news but it's optional. The doctor can add news related to health issues and cures so that it can help others. The user, pharmacy, and medical can also add different types of news.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int(11)			No	None			Change Drop More
3	image	text	latin1_swedish_ci		No				Change Drop More
4	title	text	latin1_swedish_ci		No				Change Drop More
5	description	text	latin1_swedish_ci		No				Change Drop More
6	date	text	latin1_swedish_ci		No				Change Drop More
7	time	text	latin1_swedish_ci		No				Change Drop More
8	created_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More
9	updated_at	timestamp			No	0000-00-00 00:00:00			Change Drop More
10	status	int(2)			No	None			Change Drop More

Figure 4.5.1: Add News Database

### 4.5.2 Add Post

There is another option called 'Add Post' but this option is only available for normal users or patients. If a patient wants to add a post then he must add an image, location, and description. He can also post his prescription for purchasing medicine. By seeing the post any pharmacy can contact them and send them medicines.

Server: localhost:3306 » Database: pocketsh_medicate » Table: post										
<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">SQL</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Export</a> <a href="#">Import</a> <a href="#">Operations</a> <a href="#">Triggers</a>										
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 2	user_id	int(11)			No	None			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 3	address_id	int(11)			No	None			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 4	images	longtext	utf8mb4_bin		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 5	description	text	latin1_swedish_ci		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 6	info	text	latin1_swedish_ci		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 7	date	text	latin1_swedish_ci		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 8	time	text	latin1_swedish_ci		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 9	status	int(1)			No	None			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 10	created_at	timestamp			No	current_timestamp()			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 11	updated_at	timestamp			No	current_timestamp()			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Fulltext](#)

Figure 4.5.2: Add Post Database

## 4.6 Send Request

Send request is another exiting option in the app. We already know that when a doctor gets registered in the app he must give some information. Medicals can see the information of the doctors. If the medical like the profile of the doctor and wants to allot him then the medical can send the doctor request. The doctor gets a notification that the medical send him a request. After getting the request the doctor can see the circumstances of the medical. If he likes the medical then he can accept the request. Even a doctor can also send requests to a medical.

Server: localhost:3306 » Database: pocketsh_medicate » Table: request_doctor										
<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">SQL</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Export</a> <a href="#">Import</a> <a href="#">Operations</a> <a href="#">Triggers</a>										
<a href="#">Table structure</a> <a href="#">Relation view</a>										
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 2	doctor_id	int(11)			No	None			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 3	medical_id	int(11)			No	None			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 4	start_time	text	latin1_swedish_ci		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 5	end_time	text	latin1_swedish_ci		No				<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 6	is_read	int(2)			No	0			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 7	status	int(2)			No	0			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 8	created_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 9	updated_at	timestamp			No	0000-00-00 00:00:00			<a href="#">Change</a>	<a href="#">Drop</a> <a href="#">More</a>

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Fulltext](#)

Figure 4.6: Send Request Database

#### **4.7 Online Bidding**

When a patient gives a post that he needs some medicine, many pharmacies see the post. These pharmacies can beat online then. When the customer feels comfort and justice with the price then he'll choose one.

#### **4.8 Blood Donation**

Blood donation is one of the most significant contributions that a person can make towards the society. So we try to do the work by our app. When any user at first wants to sign in then there is an option for blood donation. If he wants to donate blood then he will choose yes otherwise no. He must also give his blood group. If he wants to give blood then his name and address will show on the donation list.

#### **4.9 Ambulance Service**

When any medical wants to sign in like blood donation there is also an option if there is an ambulance service in the hospital or not. There are two options yes or not. If the medical chooses yes then the name will add to the ambulance service list. When the patient wants any ambulance service then he can see the list.

#### **4.10 Real-Time Delivery Update**

It is another exciting feature to know the real update of your medicine delivery. When the delivery man picks up any medicine he can add this in the app. By this, the customer or buyer can see the fact and see where the delivery man is. The buyer can also see how much time it takes to arrive at the delivery point and how far the delivery man is. So it is very helpful for both parties.

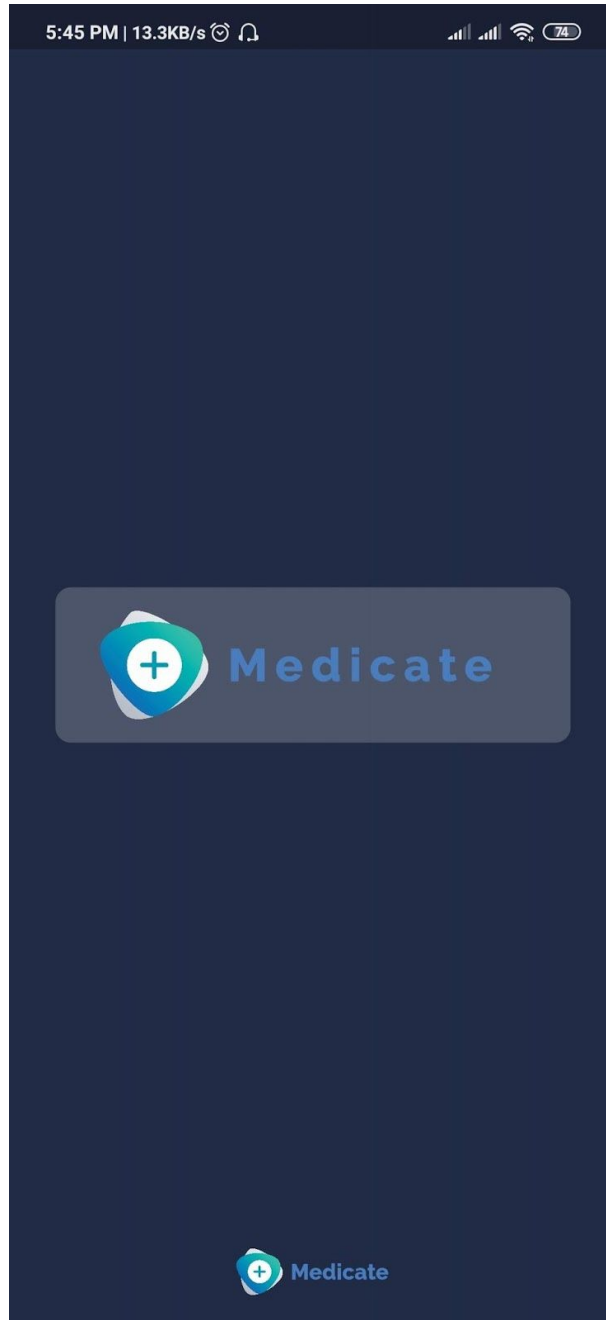
#### **4.11 Search Engine**

In the application, there will be many doctors, pharmacies, medicals, and blood group. So sometimes if you want to find any specific thing then it can be time-consuming. For that problem, we built a search engine in the application. Users can search their specific doctors, medicine, pharmacy, and blood group. This will be very helpful for them, and they just get the result instantly.

#### **4.12 Final Output Look**

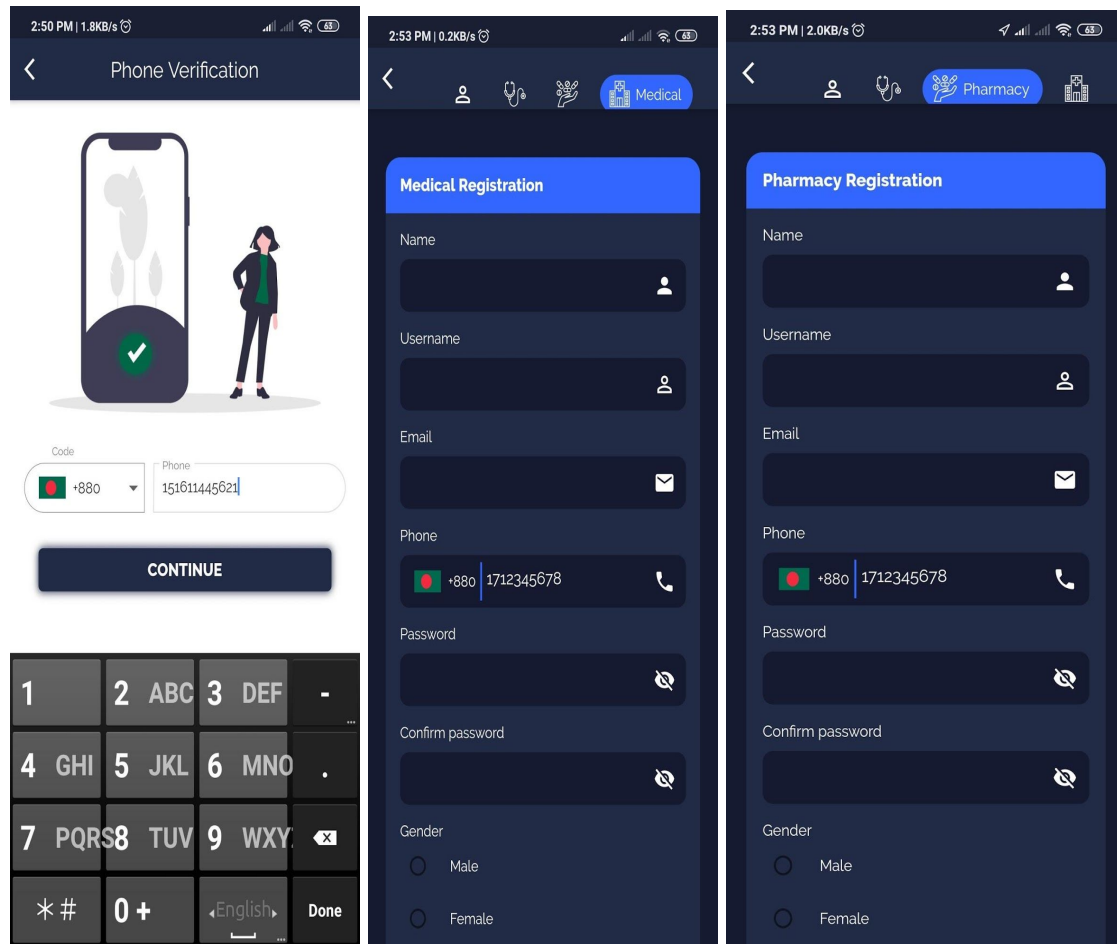
Figure 4.12.1 shows the first starting of the application.

This screen is a splash screen that means the page will vanish after a brief period and shows the next page. The next page is the login page.



**Figure 4.12.1:** The First Outlook of the Application.

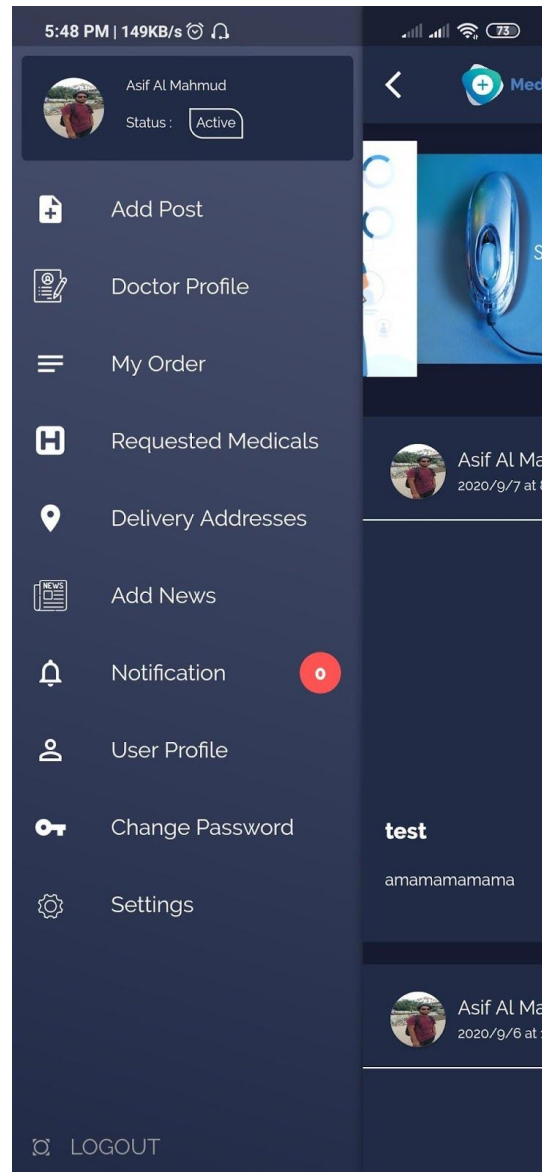
Figure 4.12.2 will show the Sign-up page. By inputting name, email, phone number, gender, and password one can sign up in the app.



**Figure 4.12.2: Sign Up**

Fig 4.12.3 will show the navigation drawer of the application. Some options that vary from user to user. There are some common options like add posts, setting, change passwords.





**Figure 4.12.3:** Navigation Drawer

Fig 4.12.4 will show the home page of the application. There the user can see the post like health tips, advertisement, search option, and many more.

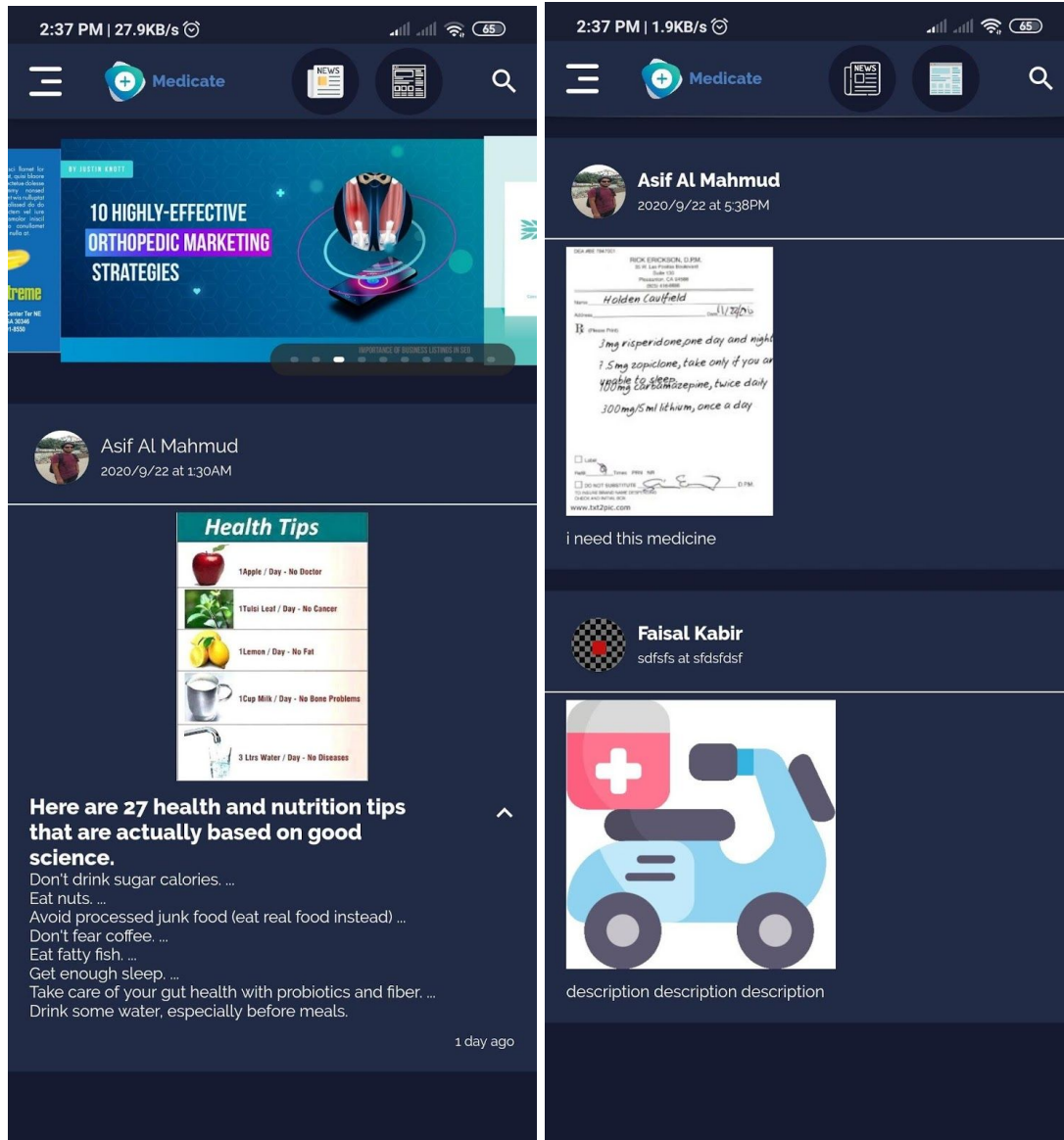


Figure 4.12.4: Home page

Figure 4.12.5 will show how the user profile is. There are some basic pieces of information about the user like name, phone number, address, picture, etc.

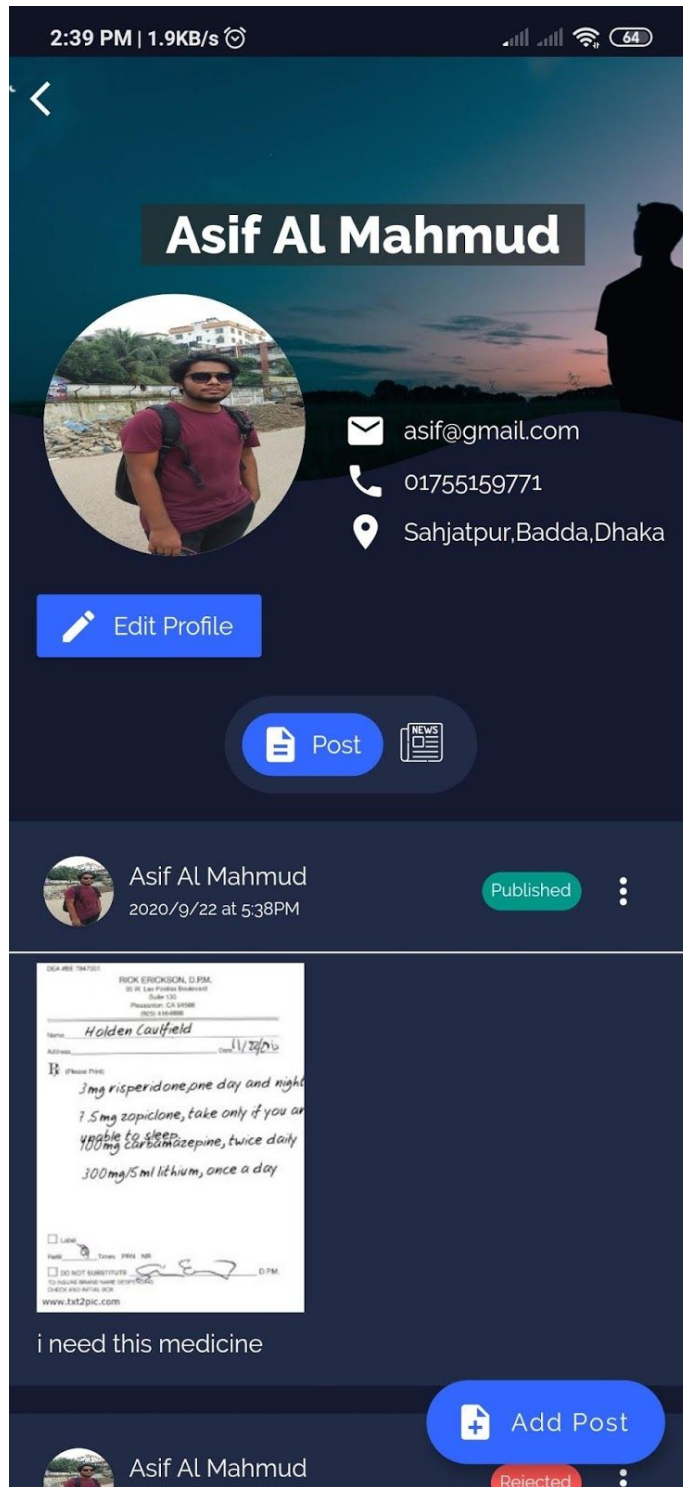
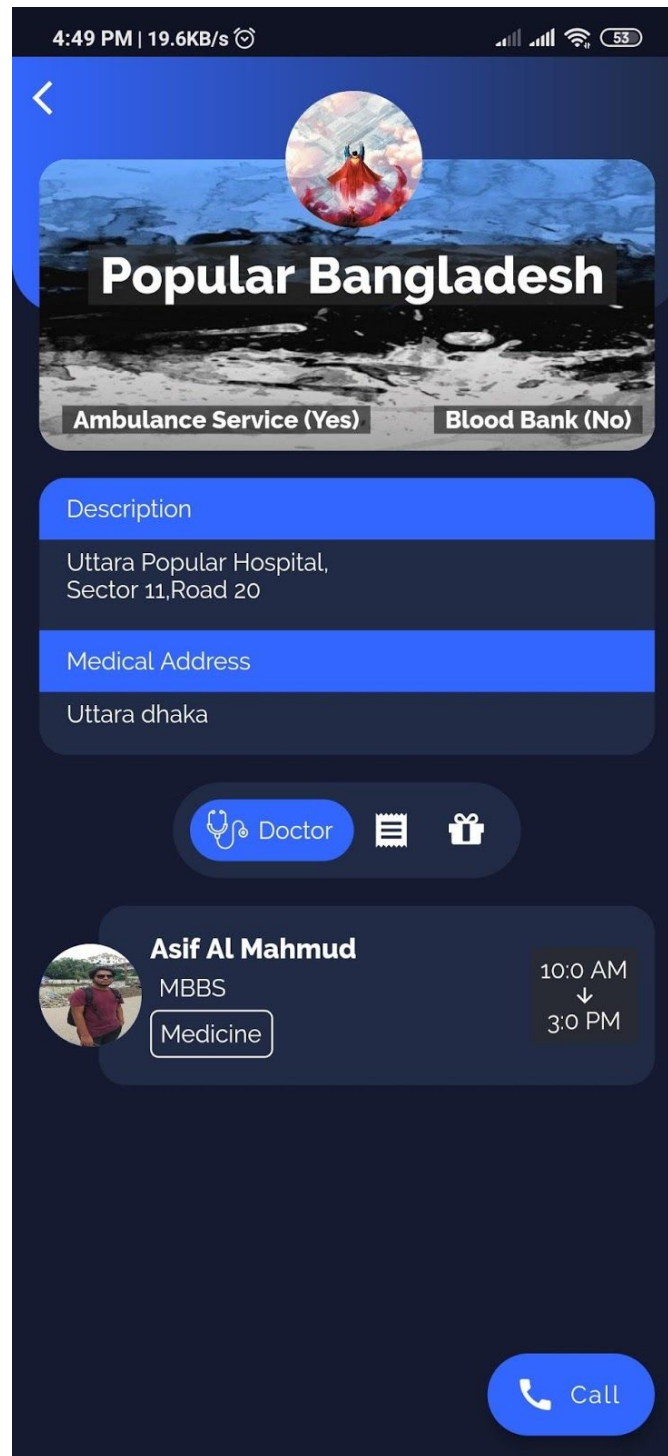


Figure 4.12.5: User Profile

Figure 4.12.6 will show the medical user profile. There are some pieces of information like medical name, address, ambulance, blood bank, doctors, etc.



**Figure 4.12.6:** Medical Profile

Figure 4.12.7 will show the services of the medical such as tests.

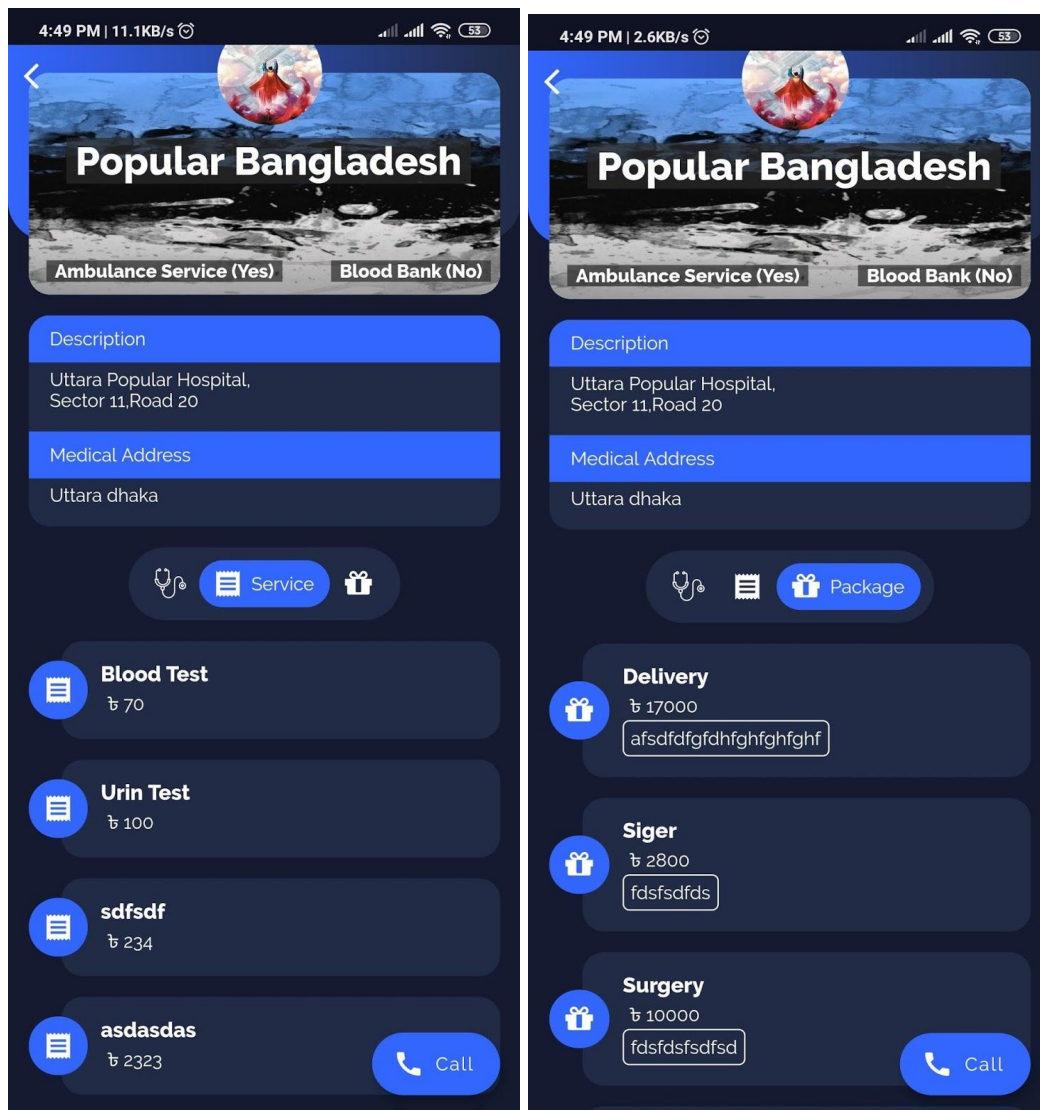
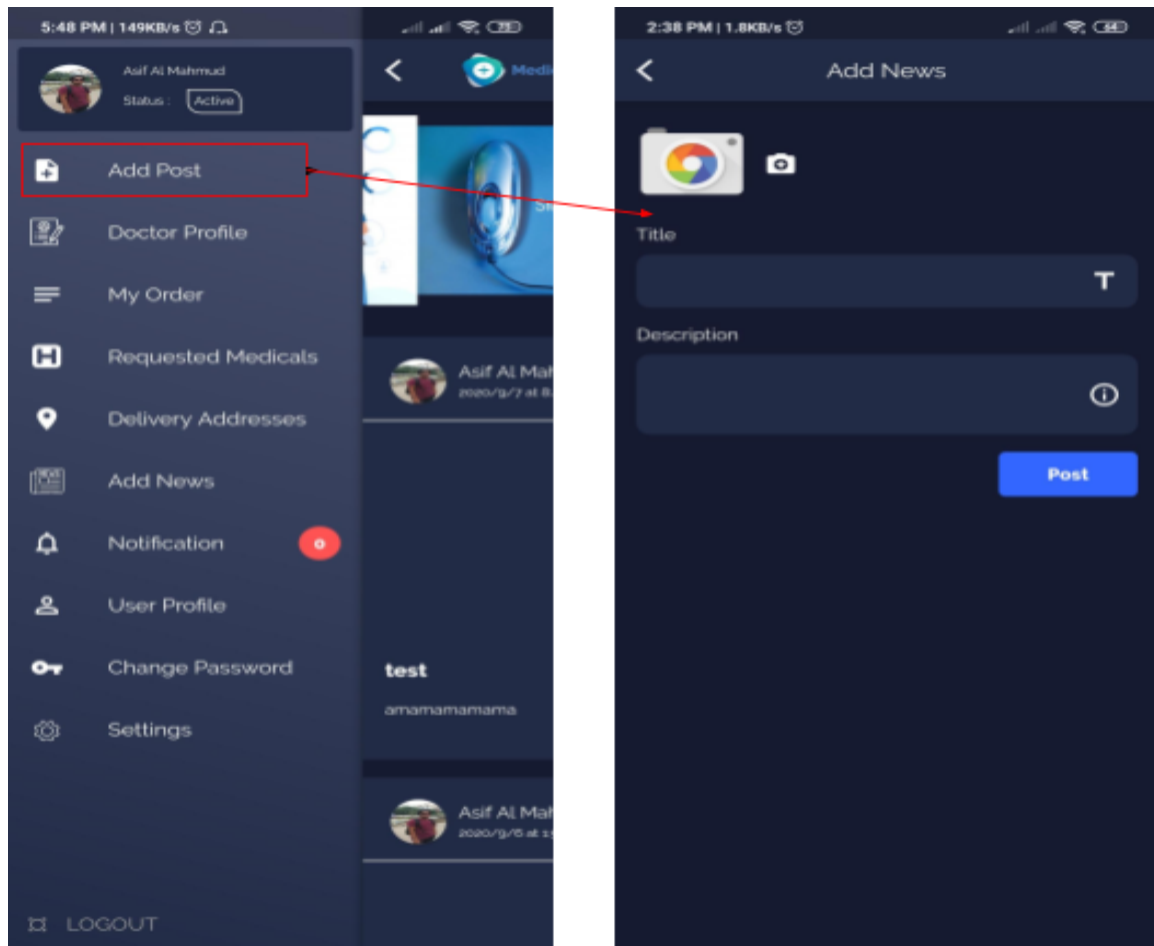


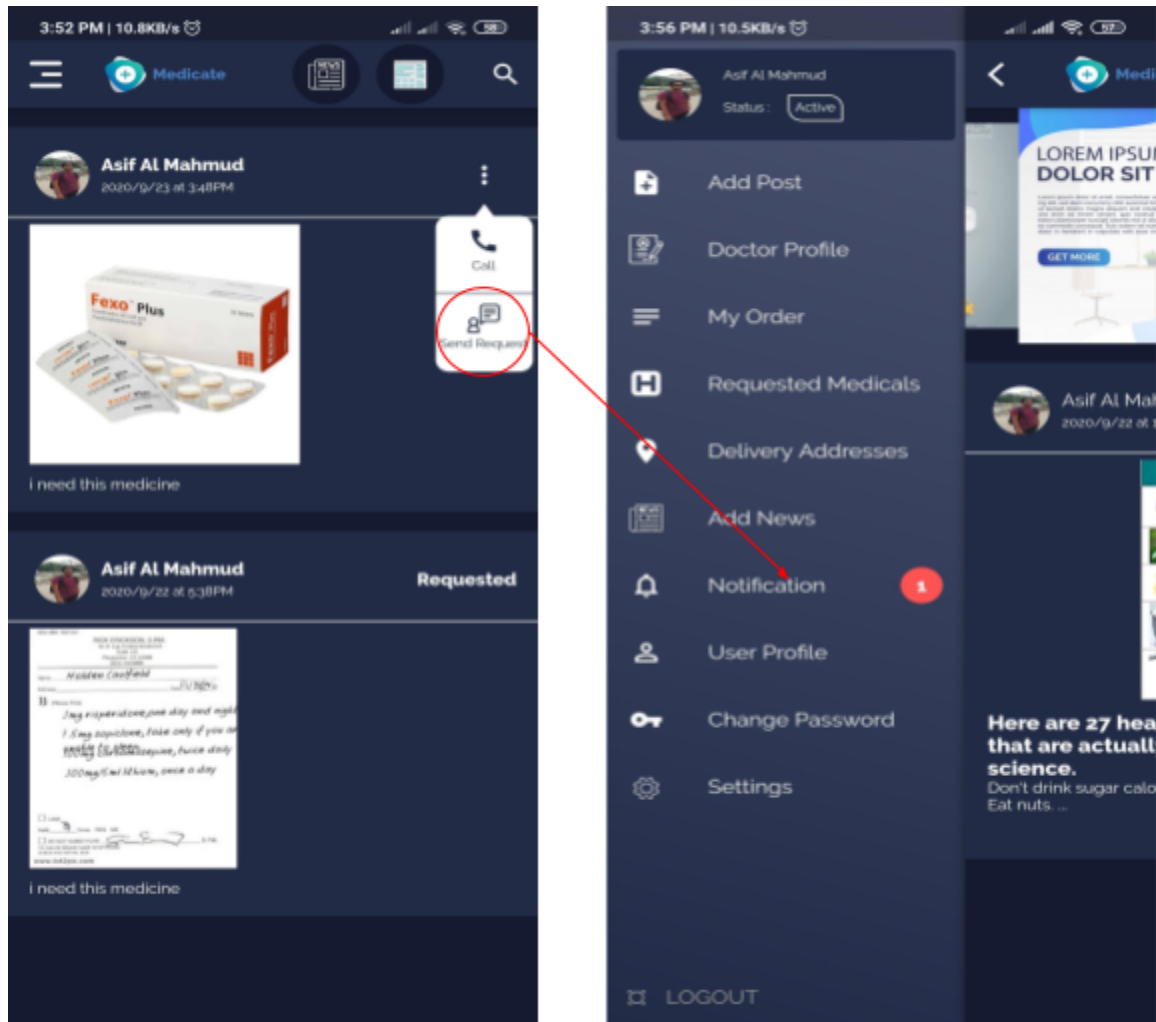
Figure 4.12.7: Medical Service

Fig 4.12.8 will show how the doctor adds a post for the medical.



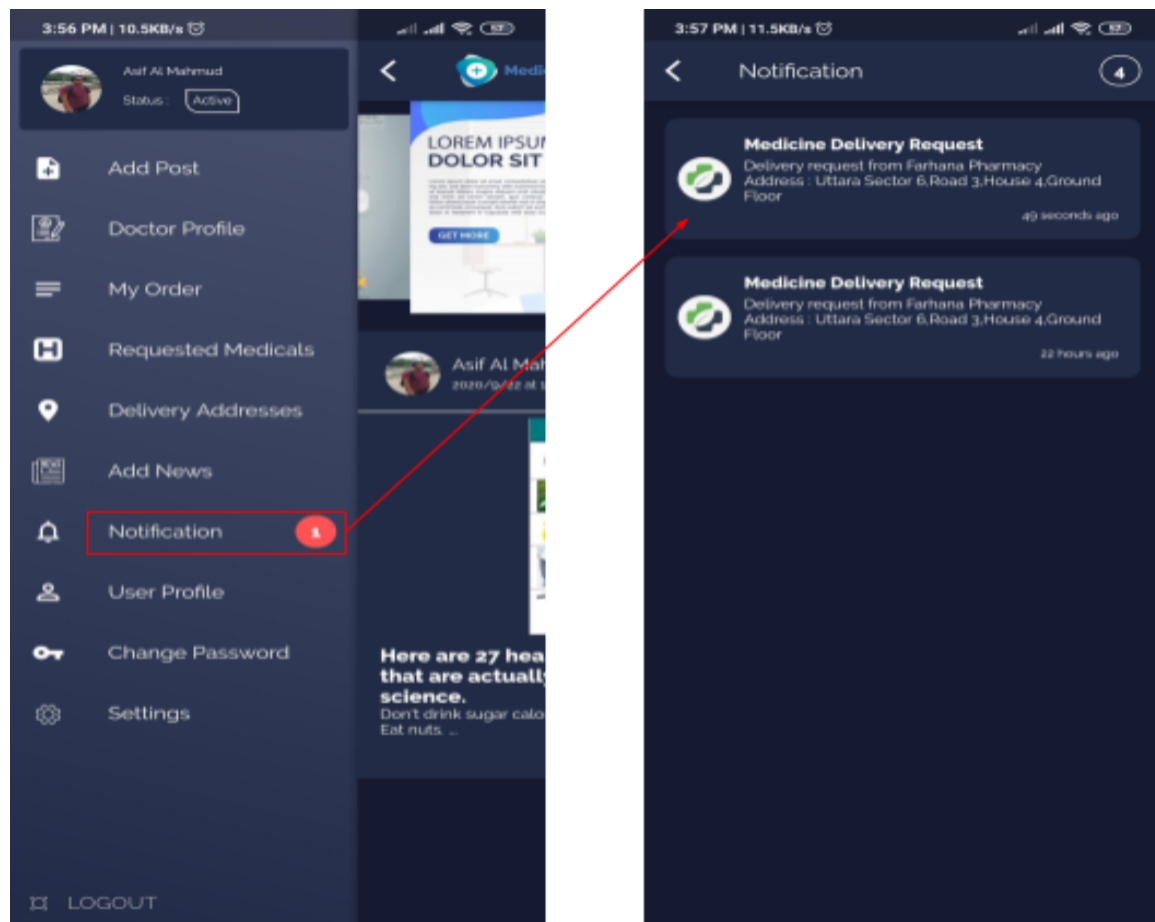
**Figure 4.12.8:** Add Post

Fig 4.12.9 will show us how to send requests from pharmacies to users for bidding. When any user wants medicine, and he will post it on the app then the pharmacies can bid it and send a request to the user. The user will get a notification for the request.



**Figure 4.12.9:** Bidding for Medicine

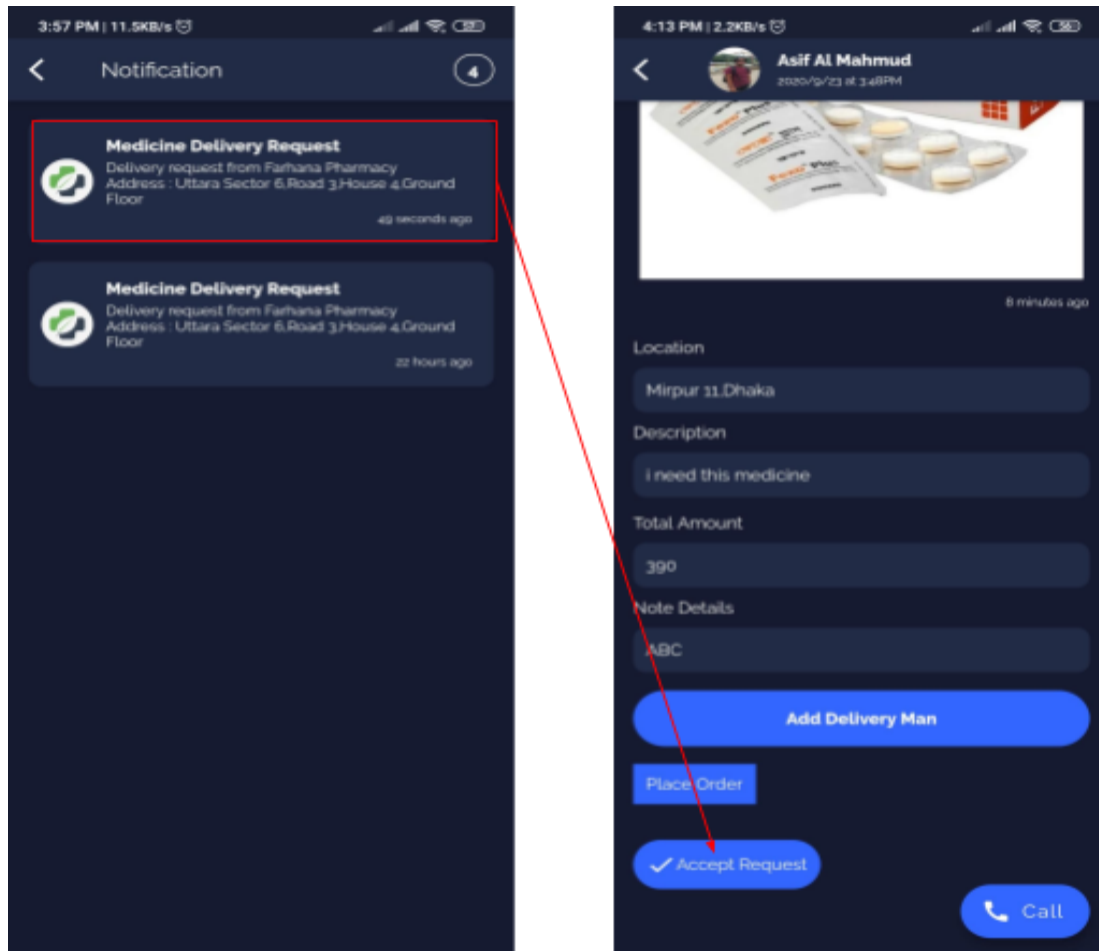
Figure 4.12.10 will show, how the user gets notifications from the pharmacies after the bidding.



**Figure 4.12.10:** Get Bidding Notifications from Pharmacies

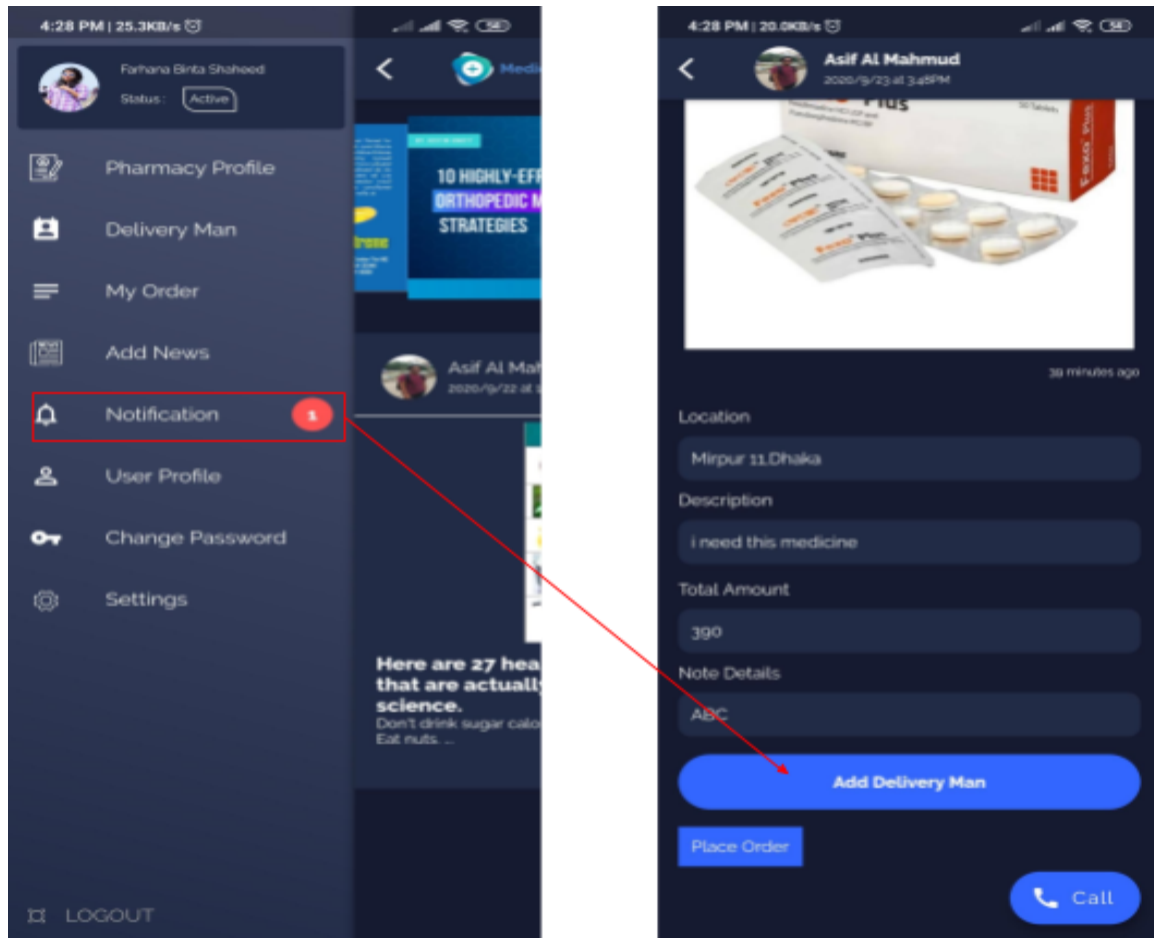
Fig 4.12.11 will show the accepted procedure of the request.





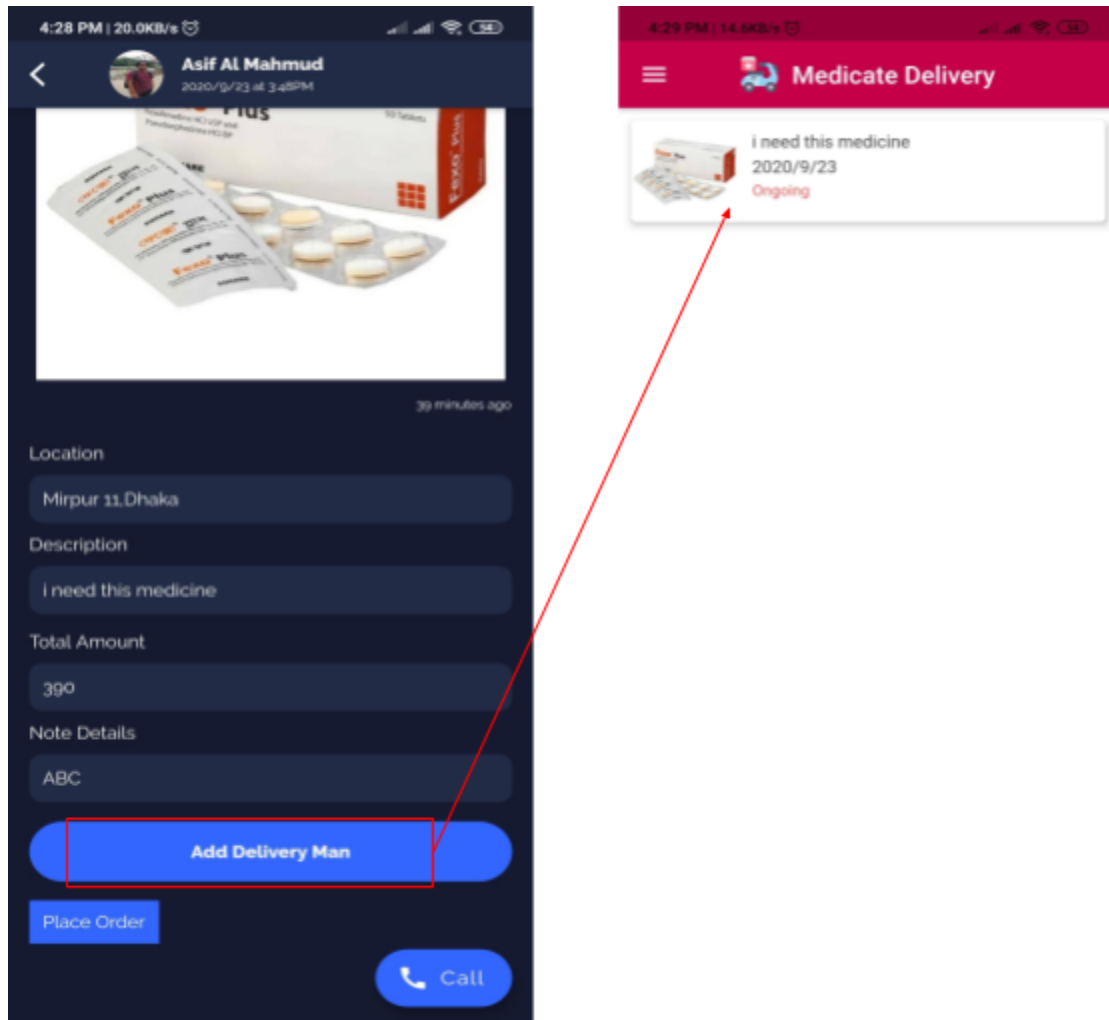
**Figure 4.12.11:** Accept the Request

Fig 4.12.12 will show when the pharmacy clicks on the notification then the page will go to the next page. On the next page, there is an option to add a delivery man.



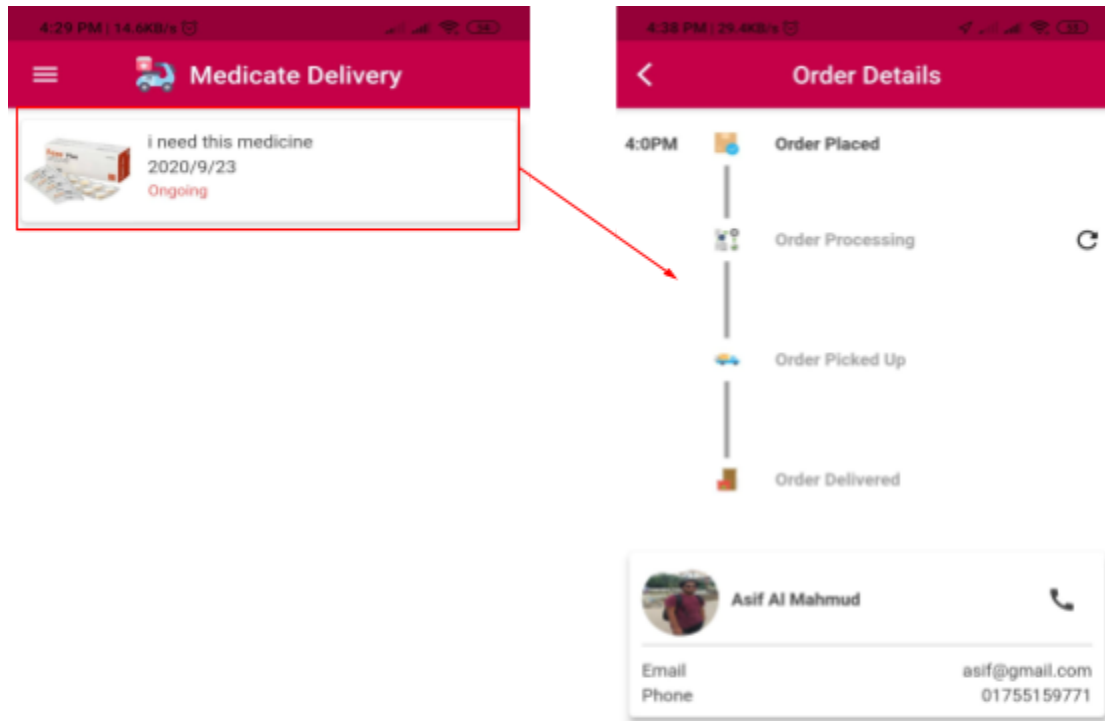
**Figure 4.12.12:** Click on the Notification Button.

Figure 4.12.13 will show how the pharmacy can add the delivery man for the medicine delivery.



**Figure 4.12.13:** Add Delivery Man

Figure 4.12.14 will show when the delivery man accepts the request then the details are also seen on the next page.



**Figure 4.12.14:** Accept the Delivery Request

Figure 4.12.15 will show the real-time delivery process. That means when the delivery man picks up the medicine or on the way or delivers the medicine it will be shown in the app.

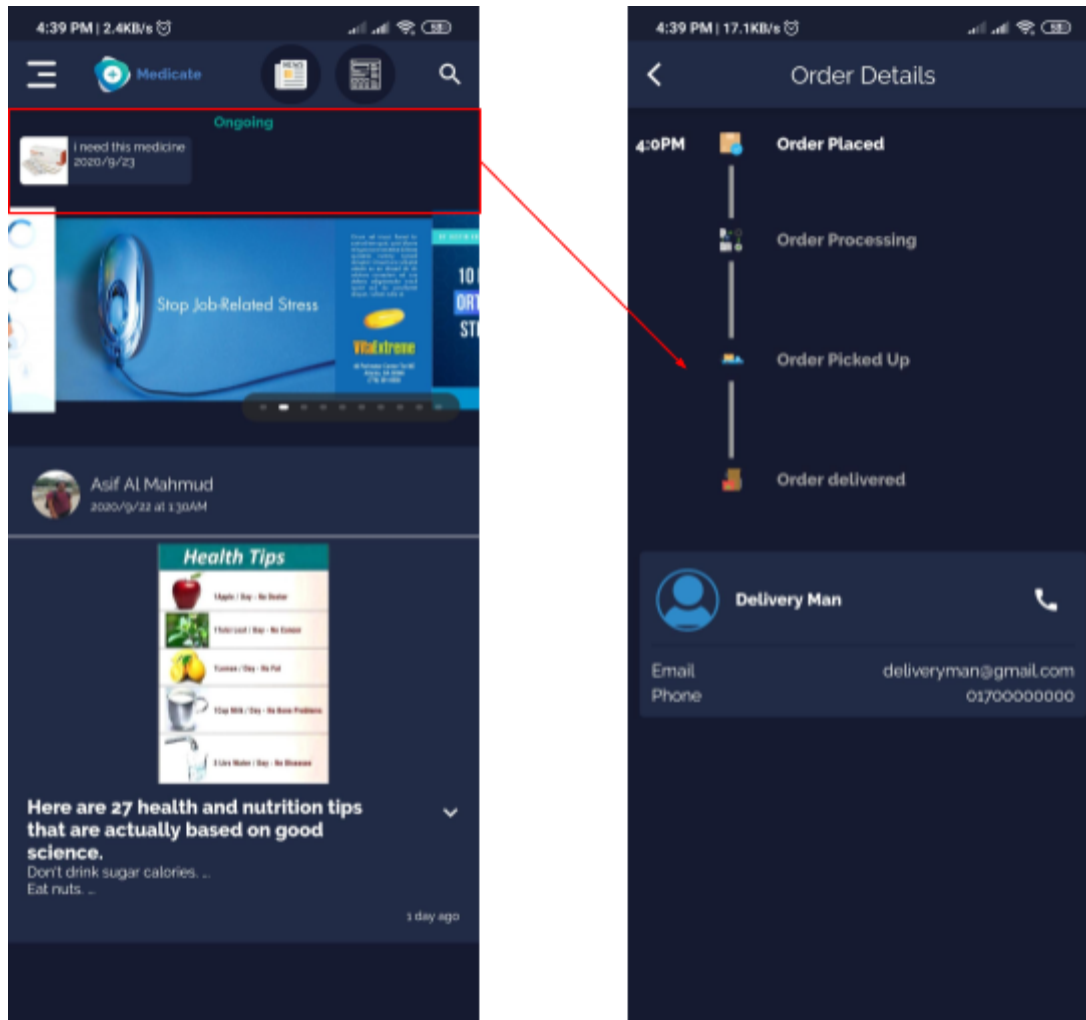
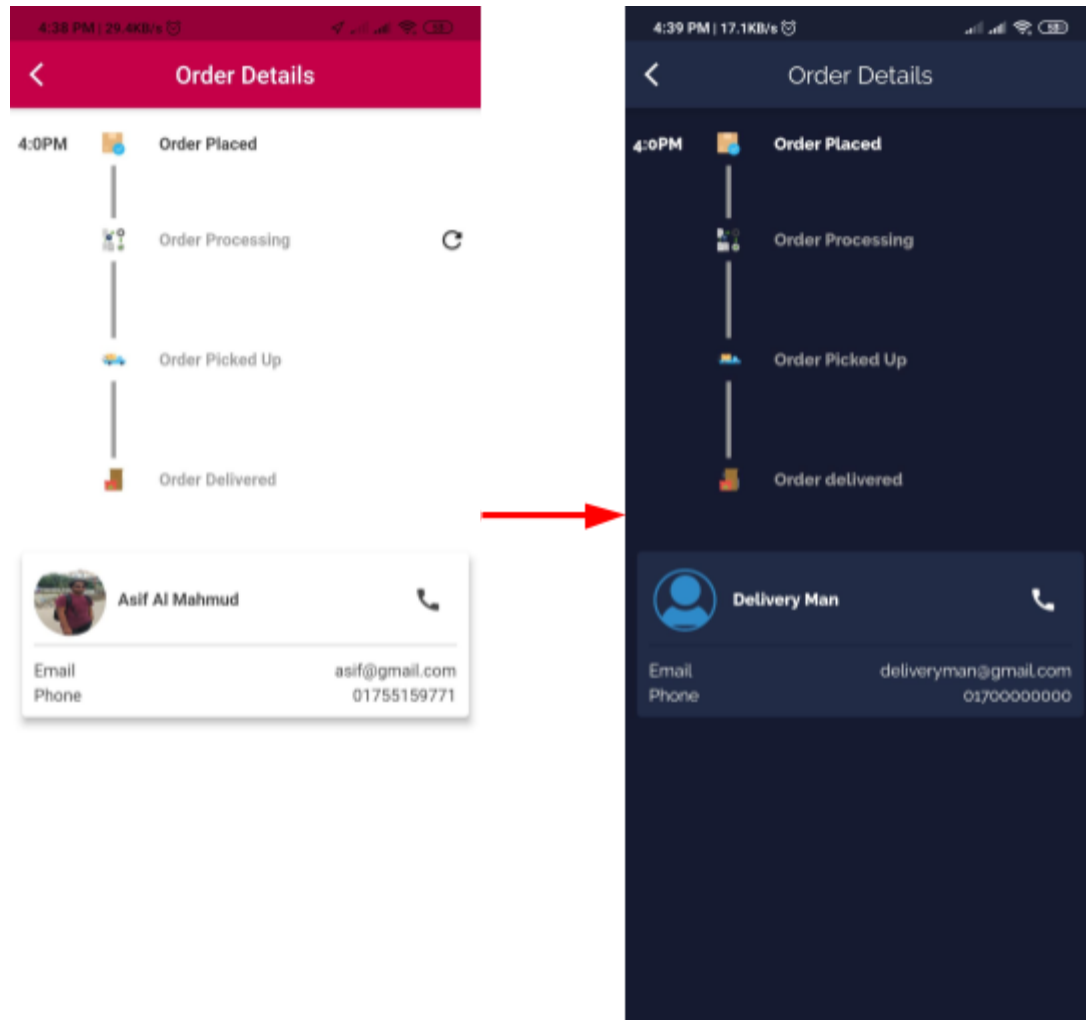


Figure 4.12.15: Real-time Delivery Process

Figure 4.12.16 will show the real-time processing system.



**Figure 4.12.16:** Real-time Processing System

### 4.13 Implementation Requirements

For developing a mobile application there are some technology and technical requirements.

- Android Studio [1]: Android Studio is that the official integrated development environment (IDE) for Google's Android package, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. In the android studio, we are able to work with the flutter framework by adding the flutter plugin. It's available for download on macOS, Windows, Linux based Operating System.
- AVD Manager [2]: An Android Virtual Device (AVD) could be a device configuration that's run with the Android emulator. It works with the emulator to supply a virtual device-specific environment to install and run Android apps. The Android SDK provides the AVD Manager tool for creating and otherwise managing AVD.
- Dart [3]: Dart is a client-optimized programming language for fast apps on multiple platforms. It is developed by Google and accustomed to building mobile, desktop, backend, and web applications. Dart is an object-oriented, class defined, garbage-collected language employing a C-style syntax that compiles optionally into JavaScript as we build the application for both Android and IOS, so we chose the dart language program.
- MySQL [4]: MySQL is an Oracle-backed open-source computer database management system (RDBMS) supported Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most frequently related to web applications and online publishing. MySQL is a vast component of an open-source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, and MySQL as the computer database management system, and PHP as the object-oriented scripting language. (Sometimes Perl or Python is employed rather than PHP.).

- Laravel [5]: Laravel is a web application framework with an expressive, elegant syntax. We believe development must be a pleasant, creative experience to be truly fulfilling. Laravel attempts to require the pain out of development by easing common tasks used in the majority of web projects, like authentication, routing, sessions, and caching. Laravel aims to create the event process a delightful one for the developer without sacrificing application functionality. Happy developers make the most effective code. To this end, we've attempted to mix the very best of what we have seen in other web frameworks, including frameworks implemented in other languages, like Ruby on Rails, ASP.NET MVC, and Sinatra.
- Firebase [6]: Firebase is a kind of Google's mobile application development platform that helps us to build, grow, and improve our app. We use push notifications and phone authentication for our project.



## **CHAPTER 5**

### **IMPLEMENTATION & TESTING**

#### **5.1 Introduction**

Implementation is a very necessary thing for a project. Implementation is the action that must follow any preliminary thinking for something to happen. For an implementation process to be successful many tasks need to be done in sequence. When a project is completed then there must be some scope of implementation. Today we tried hard to build up the application and partially we succeeded. But in the future, we also have a plan to build the app more extensively. Testing also plays a vital role in application development. If we don't test the things then we cannot get real-life problems. So for developing our application both sectors are needed.

#### **5.2 Implementation of database design**

##### **5.2.1 Database**

For the database, we used the MySQL database management system. There are 27 tables in our project for the database. The different table has a different name for identity.

##### **5.2.2 Security Testing**

Security is the most concerning topic in the project. As there are doctors, pharmacies, donors, and delivery boy they all have some personal information and their safety is very important. If something goes wrong then all the information will vanish. So we have to be very careful about the security system. We use flutter for our project and for that each and every data is secured here.

If anyone tries to decompile it even he will not get any chance to destroy this.

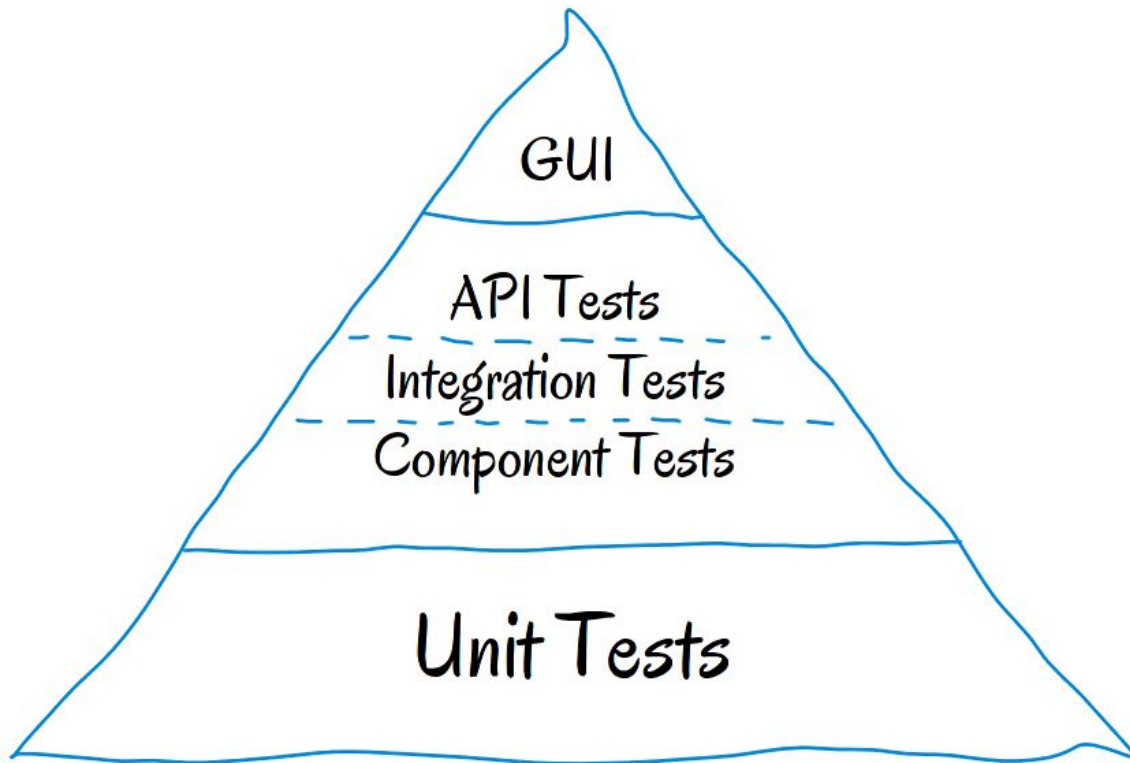
### 5.2.3 System Testing

System testing is a kind of testing that validates a complete and fully integrated system. So system testing is very important for every project. System testing checks how components are interacting with one another and with the whole system. Like we are getting the exact output what we expected but it's not working for android and iOS. On the other hand, it may seem different when we try to install it. By system testing, we try to solve them.

While testing the full process we should know about the testing pyramid. There are three types of testing in the pyramid such as small testing, medium testing, and big testing.

- Small testing is actually the unit testing where individual units or components are tested. When we were developing our application then each point is tested by small testing.
- Medium testing is a kind of testing between small and big testing. It completes different components and runs on emulators or real devices.
- Big tests are done when the full project is completed. Big tests run by completing a UI workflow. They confirm that key and other outputs are working as expected on real devices.

Figure 5.2.3 will help to get an idea about pyramid testing. The bottom part is the small testing and the top part is the big testing.



**Figure 5.2.3:** Testing Pyramid

### **5.3 Implementation of Front-end Testing**

To be honest when we get the idea about Medicate we think we will not face any hard problems while developing that. After that, we started our project, and step by step we were developing it with different types of testing. That was a plus point for us actually. At every point of any error, we tried to fix it. We asked for help from experts, on the internet and also from our teachers. As testing and implementation depend on each other so for an accurate result we tried our best. When we thought that a specific part of our application is giving the exact out what is expected then we moved on to the next part.

### **5.3.1 Automate User Interface Tests**

User Interface(UI) testing is the process of testing the visual elements of the application. We know that there are so many variations in smartphones. So we tried to make the application for the android and iOS users. In android and iOS phone users can use our application very easily also in tab or iPad they can use this. As we use flutter so there is no difference between android and iOS.

For each of the parts, we applied testing and overcame all the problems. We used many versions of design and various resources files to improve our design and also used Google's latest material design concept.

### **5.4 Testing**

Actually, software testing is a kind of method whether the output is matching our expectations and to ensure that the application is error-free. So testing plays a vital role in the project to get the desired result. There are many kinds of testing for different purposes but in our project verification and validation testing is focused. Verification testing helps us to build the application according to the requirements and design. Validation testing helps us to ensure things about our users. We also have to check every data for the accuracy and satisfaction of the users.

### **5.6 Test Result and Reports**

After applying some kinds of testing we got the expected result of what we wanted. There is no error or defect in the application so the test result is positive.

## **CHAPTER 6**

### **IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY**

#### **6.1 Impact on Society**

Medicate can create a vast change in society. Communication between the doctors and the medical, the doctors, and patients, the users, and blood donors can always keep advance. The medicine delivery system also plays a great role in the society. In this pandemic situation, everyone wants to stay home and keep a social distance. By using the app everyone can get the medical updates in their hand.

#### **6.2 Impact on Environment**

Medicate is an application that can be used for medicals, pharmacies, doctors, patients, and also donors. So there is no negative impact on the environment. It is a safer platform for all types of users.

#### **6.3 Ethical Aspects**

After developing any application there are some ethical aspects. Medicate has also some ethical aspects like respect for autonomy, non-maleficence, beneficence, and justice. It is based on a set of values that professionals can refer to in the case of any confusion and conflict.

## **6.4 Sustainability Plan**

Sustainability for Medicate is a judgment about whether the program, as currently constructed, will meet the demands of all users today and in the future. The evaluation of Medicate sustainability requires examination from more than one standpoint. For example, while the projected future cost of Medicate is the most widely raised concern about the program's sustainability, it is also important to consider whether the program's design continues to meet the needs of society.

## **CHAPTER 7**

### **CONCLUSION & FUTURE IMPLEMENTATION**

#### **7.1 Conclusion**

Finally, we can say that medicate is a very useful application for all. We tried to build a platform where people can search for their medical related problems. Here patients will find the exact doctor according to their problems and also find medicine staying at home. Hospitals and doctors can find each other. So it's all about communication.

The Internet has made our life easy, and we tried to make it easier by the application. Finding donors, doctors, pharmacies, hospitals, and news. So it encrypts our time and makes us self dependent. Doctors can also find suitable hospitals to serve people. So it's beneficial for all. As Medicate is very user-friendly so it can be a blessing for all.

#### **7.2 Scope of Future Implementation**

Medical Assistant is a mobile application with limited data so there are many things we want to do in the future. May these things take the Medicate to a high level. We try to give a shortlist below:

- A complete web application.
- Try to cover a large area.
- Admin system with 24x7 helping support.
- Some extra facilities for the regular user.
- More security options.

## REFERENCES

- [1] Learn about System Development Life Cycle, available at  
<<[https://www.tutorialspoint.com/sdlc/sdlc\\_overview](https://www.tutorialspoint.com/sdlc/sdlc_overview)>> last accessed on 3-9-2020 at 5:00 PM.
- [2] Learn about the phases of the System Development Life Cycle, available at  
<<<https://ncube.com/blog/software-development-life-cycle-guide>>> last accessed on 4-9-2020 PM.
- [3] Learn about Business Process Modeling, available at  
<<<https://kissflow.com/bpm/business-process-modeling/>>> last accessed on 2-10-2020 at 9:00 PM.
- [4] Learn about MySQL, available at  
<<<https://en.wikipedia.org/wiki/MySQL>>> last accessed on 2-10-2020 at 9:30 PM.
- [5] Learn about testing, available at  
<<<https://code.tutsplus.com/tutorials/deciphering-testing-jargon--net-27513>>> last accessed on 2-10-2020 at 10:30 PM.



## APPENDIX

### APPENDIX A: Model Class

#### Main Activity

```
void main(){
  runApp(
    /*...*/
    MyApp()
  );
}

var Language;
String FirebaseToken;
double mainHeight, mainWidth, paddingTop, appBarHeight;

class MyApp extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    // TODO: implement createState
    return Show_MyApp();
  }
}

class Show_MyApp extends State<MyApp>{

  final FirebaseMessaging _firebaseMessaging = FirebaseMessaging();

  @override
  void initState() {
    // TODO: implement initState
    super.initState();
    register();
  }

  @override
  Widget build(BuildContext context) {
    SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
      statusBarColor: Themes.Primary_Dark,
    ));

    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        pageTransitionsTheme: PageTransitionsTheme(builders: {...}), // PageTransitionsTheme
        fontFamily: Save.Raleway,
      ), // ThemeData
      routes: <String, WidgetBuilder>{...},
      initialRoute: SPLASH_SCREEN,
    ); // MaterialApp
  }
}

GetLanguage((String ln, BuildContext context))async{
  Language = json.decode(await DefaultAssetBundle.of(context).loadString('assets/Language/$ln.json'));
}
```

## User Sign-Up Activity

```
class User_Signup extends StatefulWidget {
  int pos;

  User_Signup(this.pos);

  @override
  State<StatefulWidget> createState() {...}
}

class Show_User_Signup extends State<User_Signup> {
  bool loading_state=false;

  int index;
  static int pos=0;
  static BuildContext bcontext;

  Country _selectedFilteredDialogCountry ;

  Show_User_Signup(this.index);

  TextStyle _style_heading =
    TextStyle(color: Themes.Text_Color, fontSize: Dimension.Text_Size);

  @override
  void initState() {
    super.initState();
  }
}
```

```
Future request_Registration() async {...}

void GO(BuildContext context,var _scaffoldKey_User) async {

  if (date == 'Date' || date == '') {
    Show_Snackbar(Language['date_of_birth_alert'], _scaffoldKey_User);
  }
  else if (selectedGender==null) {
    Show_Snackbar(Language['gender_alert'], _scaffoldKey_User);
  }
  else if (_blood_name==null) {
    Show_Snackbar(Language['blood_group_alert'], _scaffoldKey_User);
  }
  else {
    actionRegistration();
  }
}

actionRegistration()async{
  var _scaffoldKey_User = getScaffoldKy();
  if (loading_state == false) {
    loading_state = true;
    setState(() {
  }
}
```

## Doctor Sign-up

```
class Doctor extends StatefulWidget {...}

class Show_Doctor extends State<Doctor>{
  TextStyle style_heading =
    TextStyle(color: Themes.Text_Color, fontSize: Dimension.Text_Size);

  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return Container(
      child: ListView(
        shrinkWrap: true,
        physics: NeverScrollableScrollPhysics(),
        children: <Widget>[
          Column(
            mainAxisAlignment: MainAxisAlignment.start,
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisAlignment: MainAxisAlignment.min,
            children: <Widget>[
              Padding(...), // Padding
              Container(
                margin: EdgeInsets.only(
                  left: Dimension.Padding,
                  right: Dimension.Padding,
                  top: 10), // EdgeInsets.only
                padding: EdgeInsets.only(left: 10),
                decoration: BoxDecoration(
                  color: Themes.Background,
                  borderRadius: BorderRadius.all(
                    Radius.circular(10.0))), // BorderRadius.all, BoxDecoration
              child: Center(
                child: TextFormField(
                  controller: edu_backgrnd,
                  style: textStyle,
                  keyboardType: TextInputType.text,
                  maxlines: 3,
                  validator: (String value) {
                    if (value.isEmpty) {
                      return '${Language['education_background']}'${Language['required']}';
                    } else {
                      return null;
                    }
                  },
                decoration: InputDecoration(
                  border: InputBorder.none,
                  suffixIcon: Icon(
                    Custom_Icon.MyFlutterApp.open_book,
```

```
/////Search
Route::post('search-doctor', 'Api\SearchController@Search_doctor');
Route::post('search-medical', 'Api\SearchController@Search_medical');
Route::post('search-pharmacy', 'Api\SearchController@Search_pharmacy');
Route::post('search-bloodonor', 'Api\SearchController@Search_bloodonor');

////Notification
Route::get('notification/{id}', 'Api\NotificationController@notification');
Route::get('my-notification/{id}', 'Api\NotificationController@my_notification');
Route::post('delivery-request', 'Api\NotificationController@Medicine_Delivery_Request');
Route::post('response-delivery-request', 'Api\NotificationController@Response_Delivery_Request');

////DeliveryBoy
Route::post('deliveryBoyLogin', 'Api\DeliveryBoyController@login');
Route::post('add-deliveryboy', 'Api\DeliveryBoyController@addDeliveryBoy');
Route::get('mydeliveryboys/{id}', 'Api\DeliveryBoyController@getMyDeliveryBoys');
Route::post('updateDeliveryBoy', 'Api\DeliveryBoyController@editDeliveryBoy');
Route::post('changeDeliveryBoyPassword/{id}', 'Api\DeliveryBoyController@change_password');
Route::post('orderPlaced', 'Api\MedicineDeliveryController@orderPlaced');
Route::post('changeDeliveryStatus/{id}', 'Api\MedicineDeliveryController@changeDeliveryStatus');
Route::post('updateDeliveryLocation/{id}', 'Api\MedicineDeliveryController@updateDeliveryLocation');
Route::post('cancelDelivery', 'Api\MedicineDeliveryController@cancelDelivery');
Route::get('myOngoingDeliveries/{id}', 'Api\MedicineDeliveryController@myOngoingDeliveries');
Route::post('myOrders', 'Api\MedicineDeliveryController@myOrders');
Route::post('myCurrentOrders', 'Api\MedicineDeliveryController@myCurrentOrders');
Route::get('singleOrder/{id}', 'Api\MedicineDeliveryController@singleOrder');
Route::get('removeDeliveryBoy/{id}', 'Api\DeliveryBoyController@removeDeliveryBoy');

////User Address
Route::get('getAddress/{id}', 'Api\UserAddressesController@getAddress');
Route::post('updateAddress/{id}', 'Api\UserAddressesController@updateAddress');
Route::get('deletAddress/{id}', 'Api\UserAddressesController@deletAddress');
Route::post('addAddress', 'Api\UserAddressesController@addAddress');

////Ads
Route::get('ads', 'Api\AdController@All_Ads');
```



```

/////User
Route::post('login', 'Api\UserController@login');
Route::post('register', 'Api\UserController@register');
Route::post('send-mail-code', 'Api\UserController@sendMailCode');
Route::post('profileUpdate/{id}', 'Api\UserController@profile_update');
Route::post('change-password/{id}', 'Api\UserController@change_password');
Route::get('user/{id}', 'Api\UserController@get_user');
Route::get('logout/{id}', 'Api\UserController@logout');

/////Medical
Route::post('medical-update/{id}', 'Api\MedicalController@UpdateMedicalDetails');
Route::get('allmedical', 'Api\MedicalController@Medicals');
Route::get('details-medical/{id}', 'Api\MedicalController@DetailsMedicals');
Route::get('requested-medicals/{id}', 'Api\MedicalController@requested_medical');

/////Pharmacy
Route::post('pharmacy-update/{id}', 'Api\PharmacyController@UpdatePharmacyDetails');
Route::get('allpharmacy', 'Api\PharmacyController@update_package');
Route::get('details-pharmacy/{id}', 'Api\PharmacyController@DetailsPharmacy');

/////Packagee
Route::post('add-package', 'Api\PackageController@create');
Route::post('update-package/{id}', 'Api\PackageController@update_package');
Route::get('remove-package/{id}', 'Api\PackageController@remove_package');

/////Service
Route::post('add-service', 'Api\ServiceController@create');
Route::post('update-service/{id}', 'Api\ServiceController@update_service');
Route::get('remove-service/{id}', 'Api\ServiceController@remove_service');

```

## Plagiarism Result

162-15-734

### ORIGINALITY REPORT

6%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

5%

STUDENT PAPERS

### PRIMARY SOURCES

1

[dspace.daffodilvarsity.edu.bd:8080](https://dspace.daffodilvarsity.edu.bd:8080)

Internet Source

1%

2

[www.allbusiness.com](http://www.allbusiness.com)

Internet Source

1%

3

Submitted to Daffodil International University

Student Paper

1%

4

[www.al-islam.org](http://www.al-islam.org)

Internet Source

<1%

5

Submitted to Texas A&M University - Commerce

Student Paper

<1%

6

Submitted to Universal College of Learning

Student Paper

<1%

7

[collegeresearchpaper667.blogspot.com](http://collegeresearchpaper667.blogspot.com)

Internet Source

<1%

8

[rapidsrepro.com](http://rapidsrepro.com)

Internet Source

<1%

9

Submitted to Aspen University

Student Paper

<1%

10	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
11	Submitted to Milton Keynes College Student Paper	<1%
12	Submitted to University of Greenwich Student Paper	<1%
13	Submitted to University of Portsmouth Student Paper	<1%
14	insis.vse.cz Internet Source	<1%
15	dahlan.id Internet Source	<1%
16	www.17educations.com Internet Source	<1%
17	en.wikipedia.org Internet Source	<1%
18	oer.nios.ac.in Internet Source	<1%

Exclude quotes On  
Exclude bibliography On

Exclude matches < 10 words