LifeCycle: A Blood Donation Management App

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

by

Abdus Samad Mizi

CSE 049 06389

Supervised by: Dr. Kamruddin Md. Nur

Chairman, Associate Professor

Department of Computer Science and Engineering

STAMFORD UNIVERSITY BANGLADESH

November 2017

Abstract

LifeCycle is an android mobile app. It helps needy patients to search blood donors and manage blood easily in their city or area. Users can search donors from their current location or any other location in Bangladesh. According to their preferred distance, available nearest donors will be shown on map or lists. Then users can send blood request to donors and nearest donors will receive notification about the request. Upon acceptance the requester will receive return notification with donor phone number and other details. Besides, users can send blood request to admin, see blood requests' status, write to doctor, call hotline number in emergency situation, search doctor/hospital in their city or area, chat live with admins, share photos while donating blood and, change donation availability. Additionally, there are two other interface for doctors and admins. This app connects blood donors and patients ensuring easy communication. We noticed people looking for blood donors by social media or mobile which sometimes fails to manage blood in time. LifeCycle ensures reliability and several options for patients to manage blood.

Approval

The project report “Blood Donation App” submitted by ABDUS SAMAD MIZI: CSE 049 06389, to the Department of Computer Science & Engineering, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science & Engineering and as to its style and contents.

Board of Examiner’s Name, Signature and Date:

……………………………...	**……………………………..**	**……………………………..**
(Board Member 1)	**(Board Member 2)**	**(Board Member 3)**
Date:	Date:	Date:

Supervisor’s Signature and Date:

……………………………...

Dr. Kamruddin Md. Nur

Date:

Declaration

I, hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of Dr. Kamruddin Md. Nur, Associate Professor & Chairman, Department of Computer Science \& Engineering, Stamford University Bangladesh. I also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Signature and Date:

……………………………..

Student Name: Abdus Samad Mizi

Date:

Dedicated to my beloved parents.

Acknowledgements

First of all, I would like to thank the almighty ALLAH. Today I am successful in completing my work with such ease because he gave me the ability, chance, and cooperating supervisor.

I would like to take the opportunity to express my gratitude to Dr. Kamruddin Nur, my project supervisor. Although he was always engaged with several other activities, he gave me more than enough time in this work. He not only gave me time but also proper guidance and valuable advice whenever I faced with some difficulties. His comments and guidance helped me in preparing my project report.

Last of all, I am grateful to my family members who are always with me in every step of my life.

TABLE OF CONTENTS

List of Tables

List of Figures

1 Introduction

Every time we donate blood, we can save or improve lives. There are lots of generous donors who are always ready to help people. In a critical situation, blood may not be available in the hospital or any blood banks and patients nearest ones' start looking for donors on social media or by calling reliable sources which leads the patient's life in danger for being late or unavailability. The sole purpose of this project which is a blood donation management android app called "LifeCycle" has been presented in this report is to meet the demanding needs of an easy blood donation management app and saving lives by managing blood heroes in time. The report presents all features and procedures assembled to develop the system. These documents specially containing details about objectives, scope, design model, primary requirements and finally monitoring and reporting mechanisms.

LifeCycle is an android mobile app. It is a blood donation management app for the users. Users can manage blood contacting donors or be donors using this app. A user can search nearest donors, send blood request, chat live with admins, search hospital or doctors, call to hotline etc.

1.1 Objective

The aim of this project is to provide a blood donation management app that is user friendly and connects donors and patients in one place. LifeCycle allows users search nearest donors like Uber and send blood requests to available donors. A LifeCycle user can filter their donor search by Map or city. LifeCycle also allows users to chat live with admins or send blood requests to LifeCycle team so that they can manage blood asap.

Here is the overview of the overall objectives that was taken into account while developing the app:

- To provide an easy way to manage blood removing all other so called means that exist today.
- To ensure a reliable communication between donors and patients keeping privacy safe for both.
- To evoke awareness about blood donation among people by sharing the donors' pride story.
- To make it easier to communicate with specialized doctors.
- To provide the users a vast database of hospitals of all cities of Bangladesh.
- To allow users contacting admins directly in emergency.
- To write to available doctors and get prescription or advice.

1.2 Background Study

I was searching for 2 bags o+ blood for my cousin who met an accident and was dying in the hospital a few months ago. I faced trouble finding o+ donors. Even I downloaded some blood donation apps from play store. But all of those don't provide user friendly and reliable system. That is why, I felt necessity of developing LifeCycle. After searching a lot of apps I found some apps which provide some features similar to my app which are given below.

Table 1.2: LifeCycle Features comparison

LifeCycle Features	Search Nearest Donor	Send Blood Request	Doctor Care	Search Doctor	Search Hospital	Hotline
Blood Donation	Yes	No	No	No	No	No
Blood Hero	Yes	Yes	No	No	No	Yes
The Blood Bank	Yes	Yes	No	No	No	No
Find Blood Donors	Yes	Yes	No	No	Yes	No
Blood Donor finder	Yes	No	No	No	Yes	No
Blood friend	Yes	Yes	No	No	No	Yes
Indian Blood Donors	Yes	Yes	No	No	No	No
Simply Blood	Yes	Yes	No	No	No	No
Blood Donor	No	Yes	No	No	No	Yes
Blood Bank Beta, Blood Bangladesh, Blood Link, Blood Donor Finder, Quick Blood, Digital Blood Bank	Yes	No	No	No	No	Yes

Live Chat	My Status	To Be Proud	Admin Panel	Doctor Panel
No	No	Yes	No	No
Yes	Yes	Yes	Yes	No
No	Yes	Yes	No	No
No	Yes	Yes	No	No
No	No	No	No	No
Yes	No	Yes	No	No
No	No	No	No	No
No	No	Yes	No	No
No	Yes	Yes	No	No
No	No	Yes	No	No

These apps give users some special and useful facilities. But I found that these apps don't provide donors privacy or easy blood management system. Then I got a question in my mind. How much helpful will it be if I develop a blood donation app with almost all blood possible blood management options? Then I did a survey among 40 people to ask the question that - do these apps fulfil their need? But the result was not much convincing. 80% of them were not satisfied about these kind of apps. They need an app from which they can effortlessly contact donor and manage blood. In these apps users cannot chat live with admins, write to doctor, search donors from their current location etc.

After considering these issues, I have decided to develop an app which will create a robust blood donation management system. Then I took the project in my hand and named it LifeCycle. This is the history of developing the LifeCycle Android mobile app.

1.3 Overview

Every time you donate blood, you can save or improve lives up to three. There are lots of generous donors who are always ready to help people. In a critical situation, blood may not be available in the hospital or any blood banks and patients nearest ones' start looking for donors on social media or by calling reliable sources which leads the patient's life in danger for being late or unavailability. The sole purpose of this project which is a blood donation management android app called "LifeCycle" has been presented in this report is to meet the demanding needs of a easy blood donation management app and saving lives by managing blood heroes in time. The report presents all features and procedures assembled to develop the system. These documents specially containing details about objectives, scope, design model, primary requirements and finally monitoring and reporting mechanisms.

LifeCycle is an android mobile app. It is a blood donation management app for the users. Users can manage blood contacting donors or be donors using this app. A user can search nearest donors, send blood request, chat live with admins, search hospital or doctors, call to hotline etc.

1.4 Features

To use LifeCycle blood donation application user must log into the system for the first time or register. If user forgets password, can change it by change password option.

Search Nearest Donor: There are two options for searching nearest donors which are “Search by Location” and “Search by Address”. In case of first option, requester has to provide blood group, number of blood bags, distance, hospital, and current or other location. Then, requester will see available donors as marker on Google map filtered by his search criteria. On the other hand, requester can search donors by Division, District, and Upazila.

Send Blood Request: User can send blood request to nearest donors from “Search Nearest Donor” option. The nearest available donors will receive notification about the request which includes the requested blood, number of blood bags, location of the request, contact hospital. The donor then can accept the request or cancel it. If the donor accepts the request, the requester will receive a notification about the acceptance. And only then the requester can see the details of that donor and make phone call. If the donor denies the request, his profile will be hidden for one day. After one day, the profile will be visible automatically. Users can also send blood request to admin only so that admin can take care of it separately. The status of all blood requests will be inside “All Requests” option under Blood Requests.

Doctor Care: Users can directly call to our doctor care hotline number or call for ambulance in emergency situation. Users can also write their problem to available doctors.

Search Doctor: Users can search doctors filtering by Specialty, Division, District, and Upazila.

Search Hospital: Search Hospital filtering by Division, District, and Upazila. Call for serial to a specific hospital.

Hotline: Users can call to complain, get help, give suggestion to LifeCycle authority.

Live Chat: Users can directly chat with admin for any kind blood related help.

My Status: Donors can change their donation availability. If any donor donated blood today and checked the already donated option from My Status his profile will be hidden from donor search for next three months. After that profile will be automatically visible. Also donor can change the status to “Not Available” at any time and set the auto profile visibility date.

To Be proud: Donor can capture photos, choose from gallery while donating blood and his photos will be shared on LifeCycle Web. This inspires others to donate blood.

Admin Panel: Admin can log in using the same app. From admin settings admin can change hotline and ambulance hotline phone numbers, add or remove doctors, add or remove admins. If any donor sends message to admin from Live Chat option, that specific admin will receive notification of that message if his Chat activity is not open. Admin will see the users list who messaged him. The last users will always be on top. Admin can chat selecting any user from that users list. Thirdly, only admin can see all donors' details and make call from "Search Donor by Address" option.

Doctor Panel: Doctor can log in using the same app. From "Doctor Care" option, doctor can see which users wrote him their problems. Doctor can reply to any user selecting a specific user from that list.

1.5 Project Requirements

Tools:

I. An android device with minimum Gingerbread operation system 2.3.1:

Android is a mobile Operation System. The search giant Google developed the Android Operation System for smart phones and many other devices. Now a days Android is a much popular mobile operating system.

II. Android Studio IDE:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

III. Java platform:

Java is a programming language and computing platform first released by Sun Microsystems in 1995. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

IV. JVM:

JVM stands for Java Virtual Machine. Every smart phone runs on a virtual machine which we known as runtime environment. For java dependent phone Android supports 2 JVM which is Dalvik and ART. Before Android 4.4.2 KitKat OS, only Dalvik was available in Android phones. From KitKat OS, Google introduced a new virtual machine named ART which is stand for

Android Runtime. ART runtime environment is better than Dalvik for some special facilities.

V. PHP:

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

VI. MySql database:

MySQL is a database system that runs on a server and ideal for both small and large applications because of its very fast, reliable, and easy to use features. MySQL is developed, distributed, and supported by Oracle Corporation.

1.6 Motivation

It is common scenario in our country that people are looking for blood of specific group creating threads on social media like Facebook or contacting their friends and family when doctors fails to manage blood. The patient's condition worsens slowly which may lead the patient's life in danger even death if blood is not managed on time. Patient's friends or family become tense and confused about whom to call or their contacting list may be small. In the age of smart phones and apps, the smartest way to fulfill this requirement is to create a hub for both, donors and patients. Although in other developed countries such as U.S.A, Canada, Australia as well as developing countries such as our neighbor India, such kind of apps are already on the run but not much has been undertaken in our country. Therefore, the motivation of this project was to provide an easy to use software to find donors in any city or area. If the target market can be acquainted with this app, both the donors and patients will be beneficial.

1.7 Chapter Summery

This chapter illustrates an introductory section of the app. Here I tried to introduce the users to my app LifeCycle. I tried to share the motivations and the objectives of the app with my beloved users. In the upcoming chapters I will discuss about Literature Review of the Project, Project Outline, Requirement Analysis, Planning The Development Process, Project Requirement and Features and Workflow.

2 Literature Review

In this chapter, I decide to review some of already existing related android applications on Google play store.

2.1 Blood Donor

Blood Donor [2] provides convenient appointment scheduling, rescheduling, reminders facilities. Donors can see how many lives they have helped save, get geo-targeted blood shortage alerts let they know if their blood type is needed in their area, take selfie of their heroic donation and share with friends and family using one of a variety of filters, claim rewards from participating retailers for your donation efforts, track their blood via Blood Journey and get notified when their blood has reached a hospital (when available), join or create a lifesaving team based around their favorite organization (school, business, sports team, etc.) or create their own, recruit others and view rankings on a national scoreboard, earn unique badges to proudly share their achievements via social media, text, or email whether they are a first-timer or a gallon donor, call helpline for any assistance. The main drawback of this application is it is available in USA only. Besides this app doesn't have search donor, search doctor, search hospital, blood request, live chat, location and distance based search, and doctor care options.

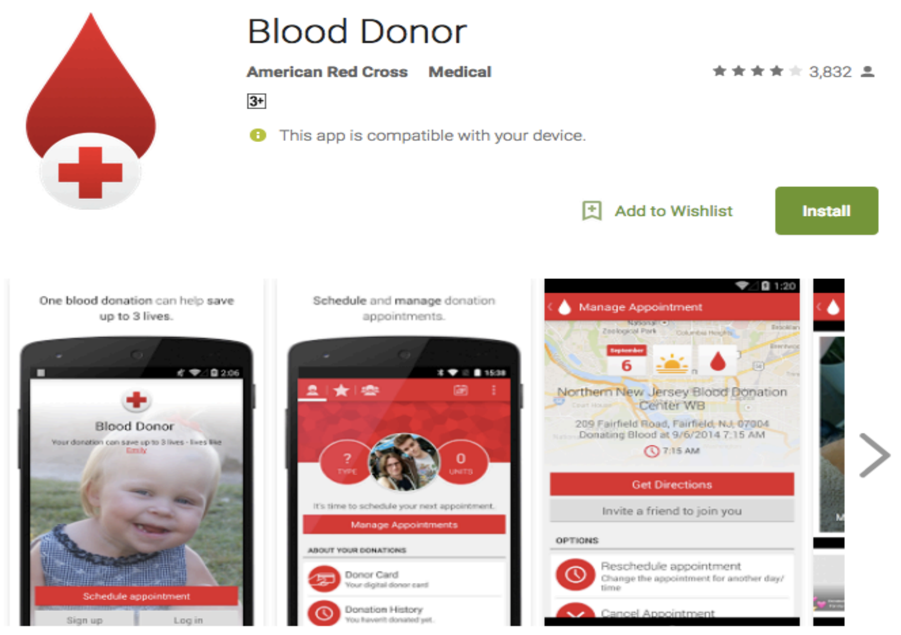


Figure 2.1: Blood Donor

2.2 Simply Blood -Find Blood Donor

Simply Blood [3] uses your phone's network to let you find and connect people for blood requirements. This app does not charge any fees for its services and you don’t have to pay to any blood donor as well for any blood donation. There is no subscription fee. In most of the apps user get to see hundreds of contact numbers which takes hours to find a suitable blood donor. Calling hundreds of blood donors is not required. Just a single search allows you to reach maximum number of blood donors in minimum possible time and that too within just 5 km from where the blood is required. Apart from other blood donation apps available it does not show donor’s contact details to public. Only a verified user can see your contact that too to a limited number of contact details. Blood donors can make special days like Birthday, Anniversary, Festival, Tribute to someone, Holidays, etc. even more special by donating on that day. This application connects the donor to the needy on that particular day to make it special. You can join the Simply Blood Ambassador Program and encourage others to be a part of it. You may integrate your social media accounts to share the requirements through your social media accounts. This app saves your blood donation history, also provides current blood requirements, safe searching, easy Sign Up and Login system. There are some drawbacks in this application too. It has been specially for India. This app does not provide blood request and distance based donors search options.

Simply Blood -Find Blood Donor
Thatspood Medical
353
3+
This app is compatible with your device.
Installed
Welcome to Simply Blood
SIMPLYBLOOD
View and Share all Blood Requests Near You
Keep Tr
Past

Figure 2.2: Simply Blood -Find Blood Donor

2.3 Blood Donor Finder

Blood Donor Finder [4] has a vast database of active donors. User can check the donor list and filter the search according to user ID, blood group, and place. Besides, this application provides details of many blood banks which is helpful if user can't manage blood contacting donors only. There are some limitations of this application too which are it doesn't have distance based donors search, donor registration, hotline, live chat, and blood request options.

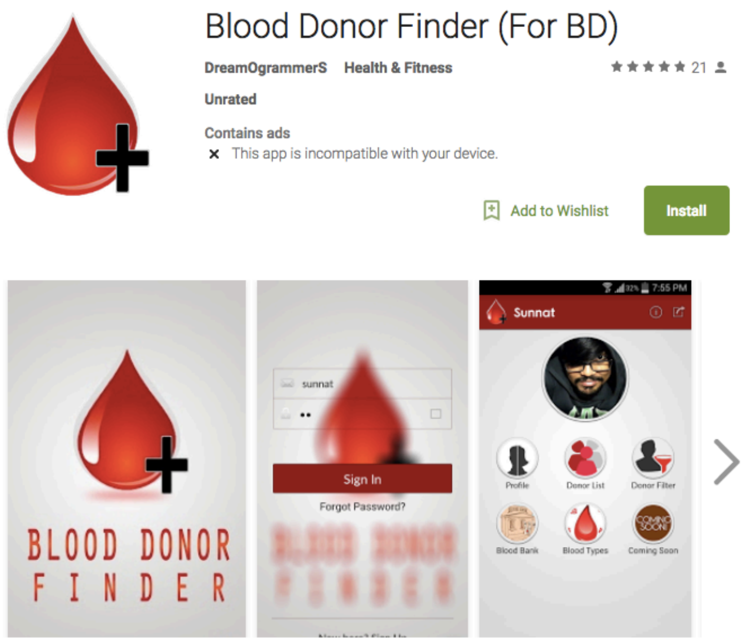


Figure 2.3: Blood Donor Finder

2.4 SocialBlood

User can search for donors and blood banks by SocialBlood [5]. The other features this application provides are convenient and easy blood donors search, blood requests from neighborhood, nearby blood banks to fix blood donation appointments, geo-targeted blood shortage alerts which let users know if blood type is needed in their area, and finding blood donors using your Facebook account and phonebook. But it does not have search doctor, search hospital, blood request, live chat, hotline, doctor care, admin panel, and doctor panel options.

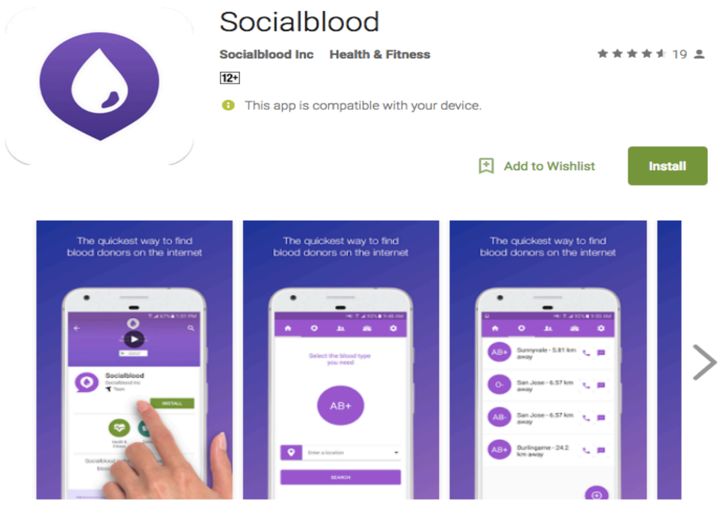

Socialblood
Socialblood Inc
Health & Fitness
19
12+
This app is compatible with your device.
Add to Wishlist
Install
The quickest way to find blood donors on the internet
Select the blood type you need
AB+
SEARCH

Figure 2.4: SocialBlood

2.5 Chapter Summery

This chapter illustrates Literature Review of the Project. In the next chapter I will discuss about Flowchart and Methodology. We will see the full flowchart and the different parts of the mechanism. The methodology will be shown here also.

3 System Design and Methodology

In this chapter, I describe all the components of my proposed blood donation android app. Here I present details of LifeCycle system.

3.1 Project Outline

LifeCycle is an android app that contains search nearest donors, send blood request, doctor care, search doctor, search hospital, hotline, live chat, my status, to be proud, admin Panel, and doctor panel. To develop this app there is a sequence of steps to follow.

1. Initial Planning
2. Organizing
3. Developing
4. Maintenance

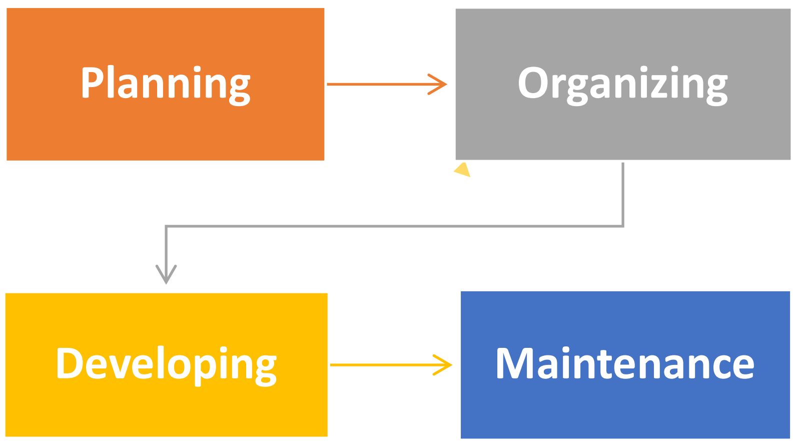


Figure 3.1: Project Outline.

Here the description of these steps:

3.1.1 Initial planning:

3.1.1 Initial planning: To make any project successful planning is a most important part. Without planning no work can be successful. To make my app efficient and more useful I took some steps like –understanding the problem, surveying, analyzing etc. In Bangladesh about 20 percent of people use smart phone. This ratio is increasing rapidly. Almost every family has at least one smart phone. So, to this is a very effective way to connect donors and patients by developing a mobile app. To develop a blood donation management app, we should know about some more other information –

How many people use smart phone?
What are the most common apps they use?
Do the apps fulfill their needs?
Surveying regarding the need of a blood donation management app.
How would it be accepted by the users?

3.1.2 Organizing: Organizing is another important part of the project. In this section I organized the whole part individually, which are planned in the planning section. Here I have gathered all the information that I have collected before. These are –

Donor list database from different internet sources.
Create a volunteer and admin list.
Provide some needed and renowned doctors contact list.
Provide some hospitals information and contact no.

3.1.3 Developing: Lifecylce is an android mobile app. To develop this android mobile app, I needed some developing tools. These are:

Java
Xml
Android studio or any other IDE for develop.
Emulator or a real android device for testing.
PHP
MySql and Sqlite Database

3.1.4 Maintenance: In this section the donor can edit his/her profile details and visibility. Their profile will be hidden for 3 months after successful blood donation. Other features are like –

Editing basic information
Changing donation availability
Sharing donation experience with others

3.2 Requirement Analysis

I did a survey among 40 people about their blood management experience. Here I made some query for analysis the consciousness state. These questions are as follows:

1. How many people would like to see all blood donation management features in one android app?
2. Do they like to search blood donors on social media or via mobile?
3. Do they prefer a system that helps them be donors or find donors easily?

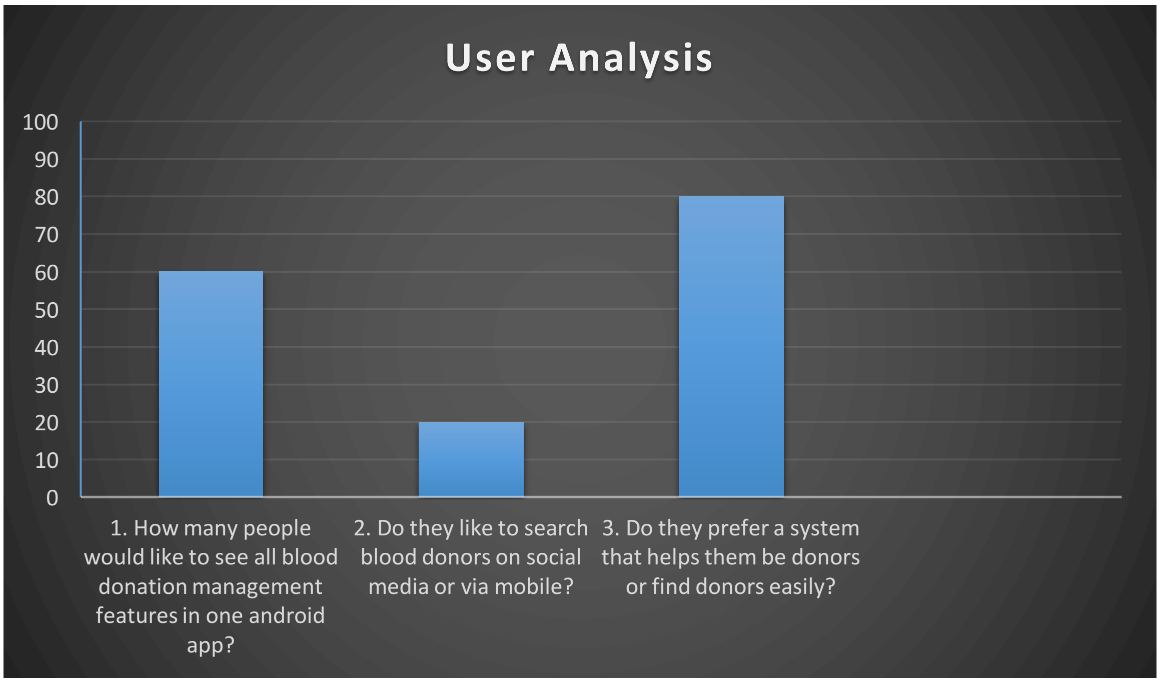


Figure 3.2: Requirement Analysis.

The outcome of the survey is not so convincing. The amount of people who like to search blood donors on social media or via mobile is around 20% where about 60% like to see all blood donation management features in one android app. Around 80% of them prefer a system that helps them be donors or find donors easily.

3.3 Development Process

In every project, planning is the most important part. Without planning a project can never be successful. So to make a project successful planning plays the vital role. In the process of the planning of LifeCyle development I followed some steps. These steps are follows –

1. Requirement Gathering
2. Analysis

3. Design
4. Coding
5. Testing
6. Implementation
7. Documentation

We will see here the whole development process planning with time period in a Gantt Chart [6] below.

Development Phase	120 days						Duration (Day)
	0 to 20 day	21 to 40 day	41 to 60 day	61 to 80 day	81 to 100 day	101 to 120 day	
Requirement Gathering	🟧						0 – 10 (10 days)
Analysis	🟧						11 – 20 (10 days)
Design		🟧	🟧				21 – 50 (30 days)
Coding		🟧	🟧	🟧	🟧		41 – 100 (60 days)
Testing					🟧	🟧	81 – 110 (30 days)
Implementation						🟧	105 – 120 (15 days)
Documentation	🟧	🟧	🟧	🟧	🟧	🟧	11- 120 (110 days)
Total Time (Day)							120 days

Figure 3.3: Development process Gantt chart.

3.4 Flowchart

Here is the flowchart of the app. In these flowchart I have tried to give an overview of the mechanism of the app and how an activity is linked to another activity.

Here is the full flowchart, created by Lucidchart [1], of LifeCycle. In this flowchart you can see all the components together how they work and how they are connected to other components.

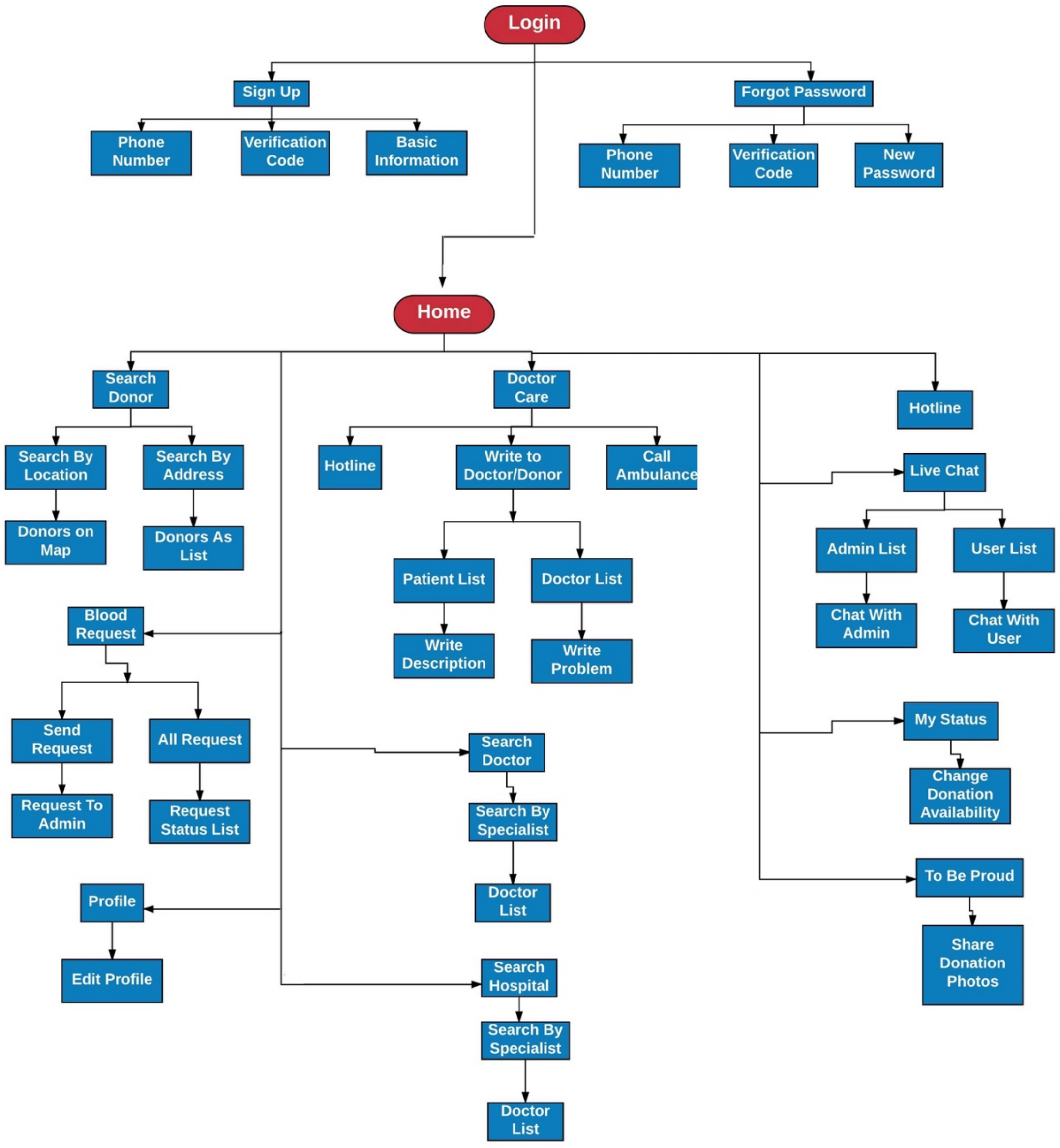
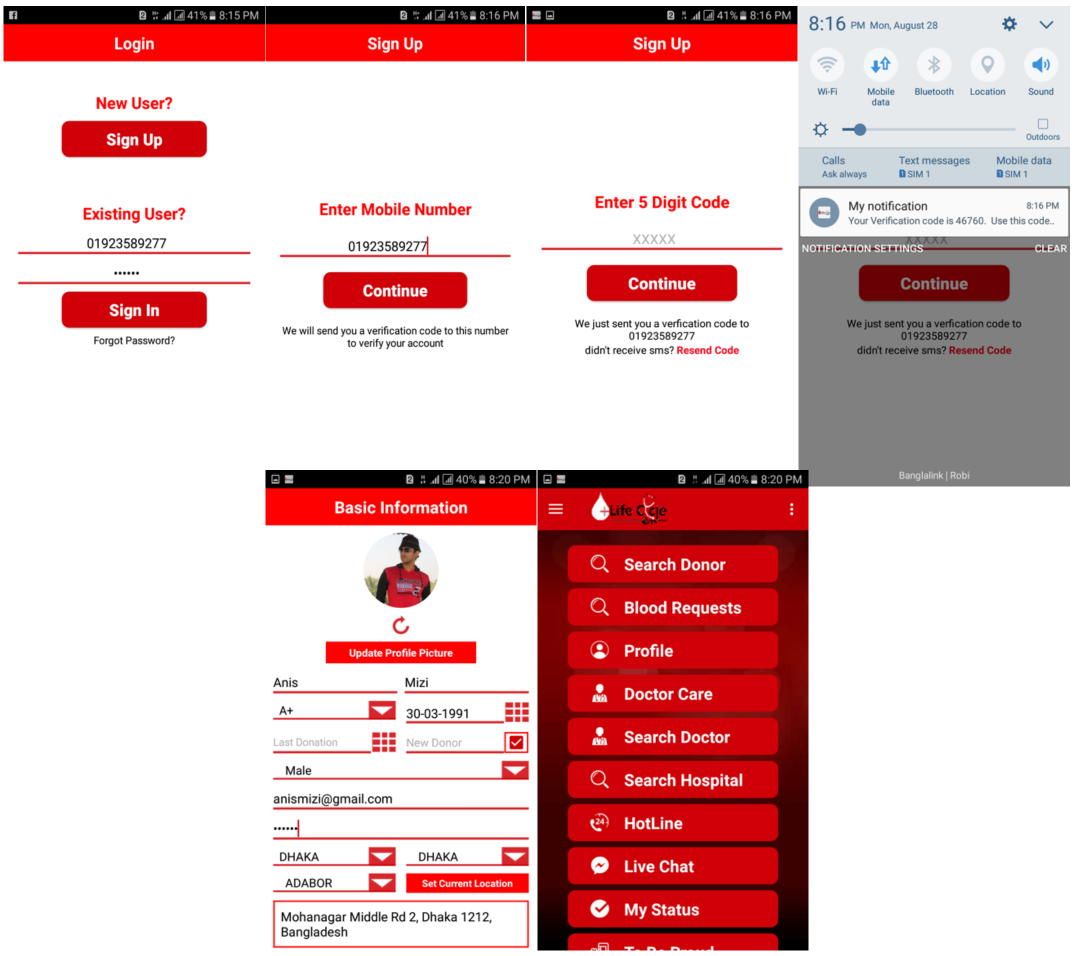


Figure 3.4: LifeCycle flowchart.

3.5 Methodology

3.5.1 Signup Process: To access the app a user has to login with the phone number and password. If user does not have any account, he/she has to sign sign up first. In case user forgets the password, can change it by clicking "forgot password" option. User has to use a valid Bangladeshi phone number starting with 01 to sign up; a verification code will be sent via SMS to the phone number. After receiving the verification code via SMS, user has to verify it. If the user does not receive any code, can request for a new one or retry by "Resend Code" option. User will receive a five-digit verification code to verify number. Then users have to provide their basic information. They can skip some fields if they want to edit them later but can't leave mandatory fields empty like blood group. To be able to chat with admins users must provide a valid email address and password. If the user sets their current location, their profile will be shown in donors list to patients by default. Of course they can hide profile or change donation availability from "My Status" option. If registration is successful user will be taken on Home screen. User can navigate to all available features of LifeCycle from Home screen. Also the top right menu option allows user to log out from the app. There will be different features for admins and doctors on Home screen.



01923589277

01923589277

We will send you a verification code to this number to verify your account

XXXXX

We just sent you a verfication code to 01923589277
didn't receive sms? Resend Code

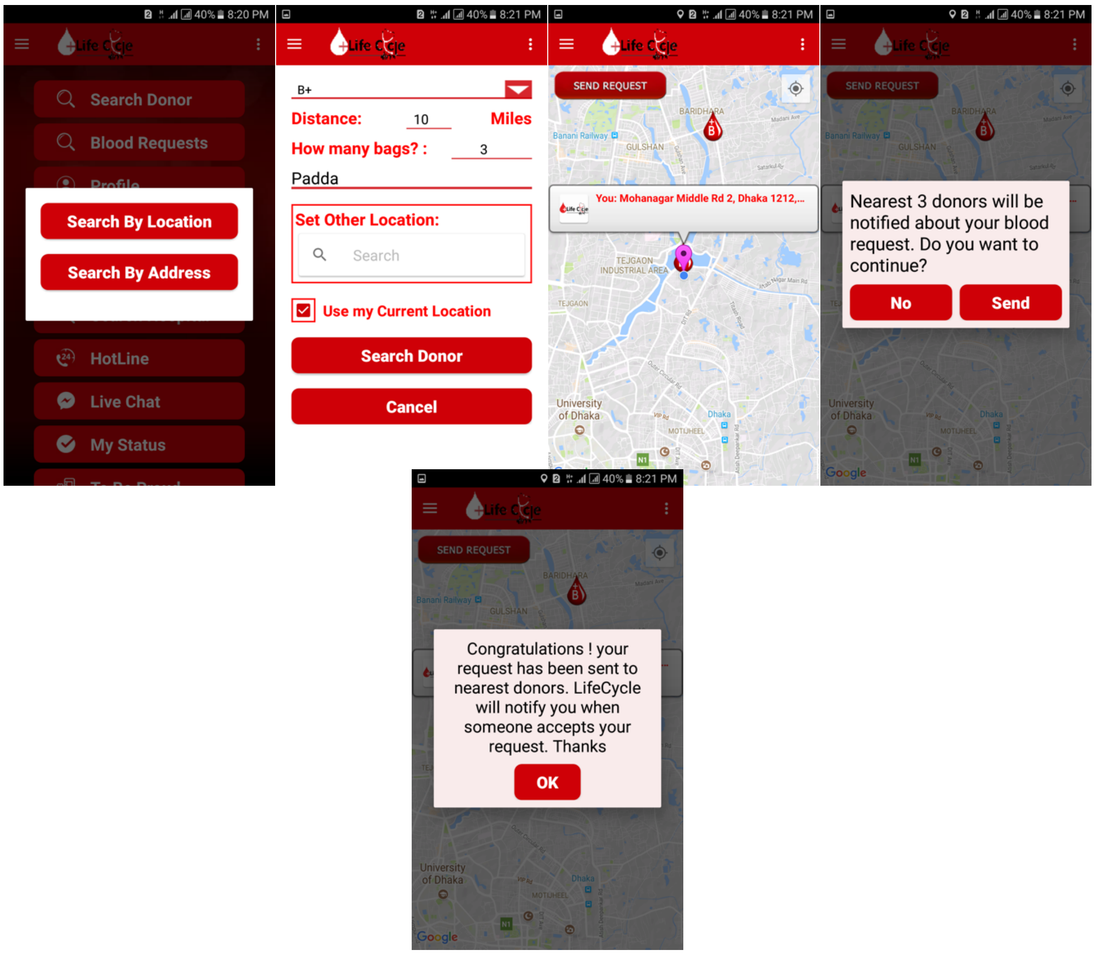
8:16 PM Mon, August 28

My notification 8:16 PM
Your Verification code is 46760. Use this code..

Anis Mizi
A+ 30-03-1991
Male
anismizi@gmail.com
DHAKA DHAKA
ADABOR
Mohanagar Middle Rd 2, Dhaka 1212, Bangladesh

Figure 3.5: Signup Process.

3.5.2 Search by Location: There are two options for searching nearest donors, Search by Location and Search by Address. Upon selecting search by location option user will see a screen to set blood request location. Here user has to provide patient's blood group, distance, number of blood bags, hospital name, and location. Nearest donors will be fetched according to user's search criteria. If user set his/her search criteria similar to figure 3.18, nearest available donors, who are within 10 miles from his current location and has B+ blood group, will be shown on Google Map as markers. User can set another location by "Set Other Location" to search nearby donors from that location. For example, if user set Mouchak, Dhaka and keep other fields as they are in the screenshot below, blood donors will be searched within 10 miles from Mouchak, Dhaka Area. The number of bags and hospital name are necessary to let the donors know, how many bags of blood is needed and which hospital to go for donation. Available donors will be shown on Google Map according to user's donors search criteria. Upon clicking on donor marker user can see it's address only. To send blood request to nearest donors, user has to click "Send Request" button and nearest 3 donors will be notified, if at least 3 available otherwise less, via notification which works as like Uber application. User will see confirmation dialog if donors are notified successfully. Before sending the request user will be prompted to ensure that the request is not being sent by mistake. Upon successful request user will get confirmation.



B+
Distance: 10 Miles
How many bags? : 3
Padda
Set Other Location:
Search
☑ Use my Current Location

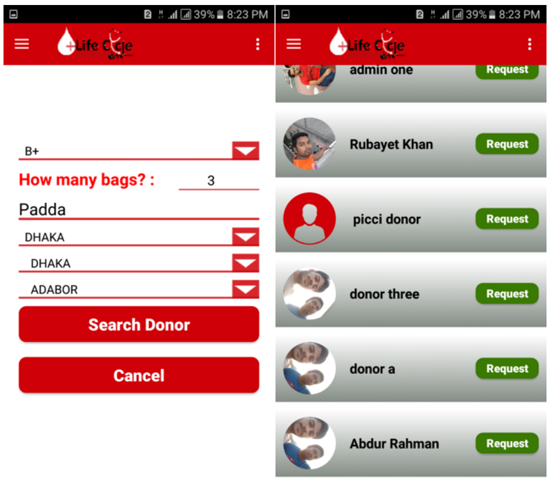
You: Mohanagar Middle Rd 2, Dhaka 1212,...

Nearest 3 donors will be notified about your blood request. Do you want to continue?

Congratulations ! your request has been sent to nearest donors. LifeCycle will notify you when someone accepts your request. Thanks

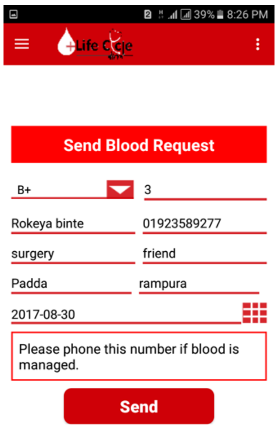
Figure 3.6: Search by Location.

3.5.3 Search by Address: This works as like "Search by Location" feature except user can see available donors in a Thana/Upazila and a list of donors will be returned with their name and profile picture only. User can send blood request to a specific donor by clicking request button and the request feature works same as before but the difference is blood request will be sent only to that specific user.



B+
How many bags? : 3
Padda
DHAKA
DHAKA
ADABOR

admin one
Rubayet Khan
picci donor
donor three
donor a
Abdur Rahman

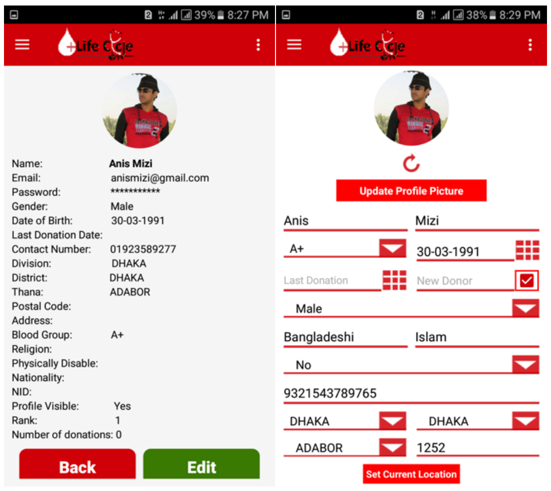


B+ 3
Rokeya binte 01923589277
surgery friend
Padda rampura
2017-08-30
Please phone this number if blood is managed.

Figure 3.7: Search by Address. Figure 3.8: Send Request.

3.5.4 Send Request: Send Request, Figure 3.8, works as like “Search by Location” feature except user can see available donors in a Thana/Upazila and a list of donors will be returned with their name and profile picture only.

3.5.5 Profile Details: Here user can see the details of their profile, some of which has already been fulfilled by them while at the time of sign of and other non-mandatory information is empty. They can edit all or any part of the details by clicking edit button. User can edit their profile details from edit profile option.



Name:	**Anis Mizi**
Email:	anismizi@gmail.com
Password:	***********
Gender:	Male
Date of Birth:	30-03-1991
Last Donation Date:	
Contact Number:	01923589277
Division:	DHAKA
District:	DHAKA
Thana:	ADABOR
Postal Code:	
Address:	
Blood Group:	A+
Religion:	
Physically Disable:	
Nationality:	
NID:	
Profile Visible:	Yes
Rank:	1
Number of donations:	0

Anis	Mizi
A+	30-03-1991
Last Donation	New Donor ✓
Male	
Bangladeshi	Islam
No	
9321543789765	
DHAKA	DHAKA
ADABOR	1252

Figure 3.9: Profile Details.

3.5.6 Doctor Care Options: If the user press on “Doctor Care” button from home menu and he/she is a donor or normal user, will see this screen. Here “Direct Doctor Call” and “Ambulance” both are two hotline numbers. If the user is ill and need prescription or consultation of LifeCycle’s available doctor, all they need is to go to “Write to Doctor” option, select any doctor they like, and write their problem briefly. After pressing on “Doctor Care” button user will see a list of available doctors to communicate. To write their problem to any specific doctor, they need to press on “Message” button and “Empty” button to delete the whole conversation with any specific doctor. Then user will see a message box screen where user can see their messages as well as the doctor’s reply he is getting consultation from. Each message contains message time and date, so that the user or doctor can easily identify when actually the message was posted.

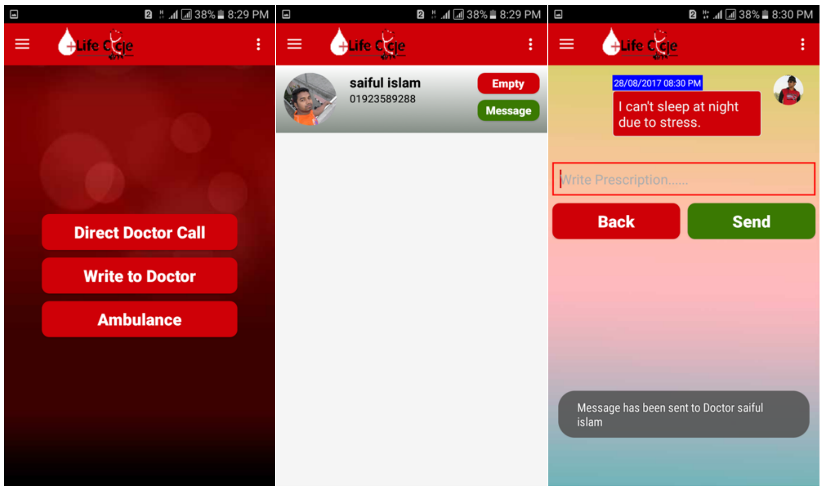


Figure 3.10: Doctor Care Options.

3.5.7 Search Doctor: Users can search for doctor filtered by their specialty, Division, District, Thana/Upazila. This search will load a list of doctors of different hospitals in that Thana. Next screen will present a list of doctors filtered by their specialty, Division, District, and Thana. Pressing "Details" button, user can see that doctor's full information. Users can make call from this screen as well as from details screen. This screen shows the details of doctor and user can make call from here too.

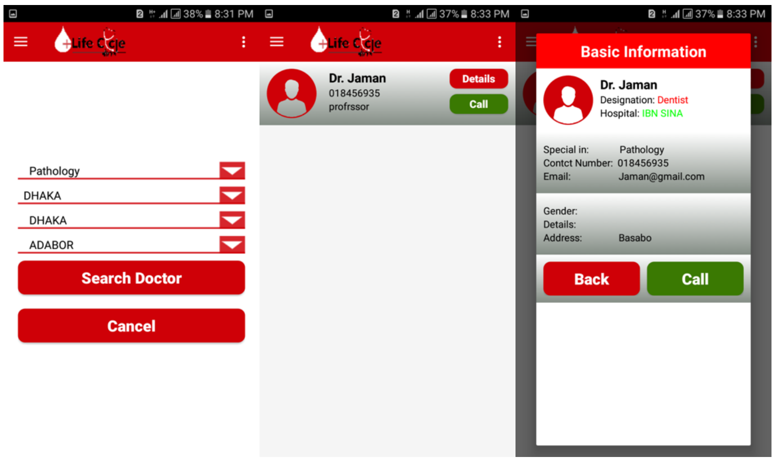
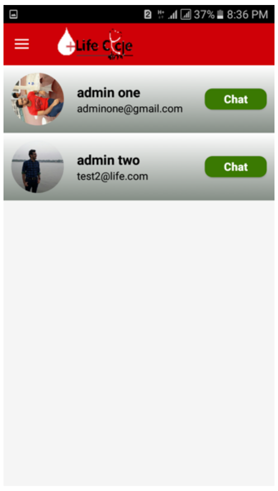


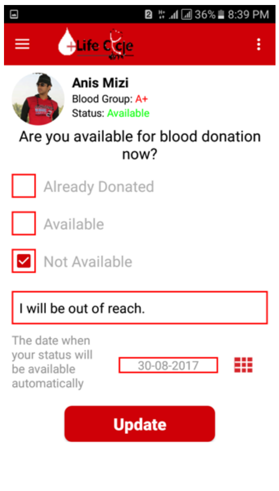
Figure 3.11: Search Doctor.

3.5.8 Admin List (Live Chat): In figure 3.12, if the users fail to manage blood, face any other difficulty, or have any query, can directly contact with LifeCycle's admin via "Live Chat" feature from home menu. The admin will receive a notification if his LifeCycle app is not open or a live session with that user is not established. If user finds any admin unresponsive, can contact with other available one. However, users can only see those admins or contact who have set their live chat availability on.



admin one
adminone@gmail.com

admin two
test2@life.com



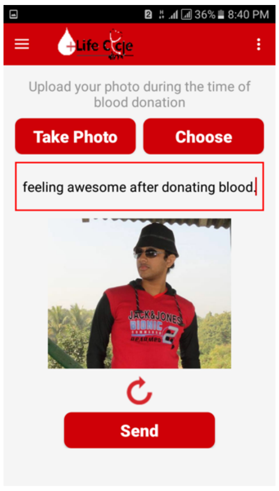
Anis Mizi
Blood Group: A+
Status: Available

Are you available for blood donation now?

☐ Already Donated
☐ Available
☑ Not Available

I will be out of reach.

The date when your status will be available automatically: 30-08-2017



Upload your photo during the time of blood donation

feeling awesome after donating blood

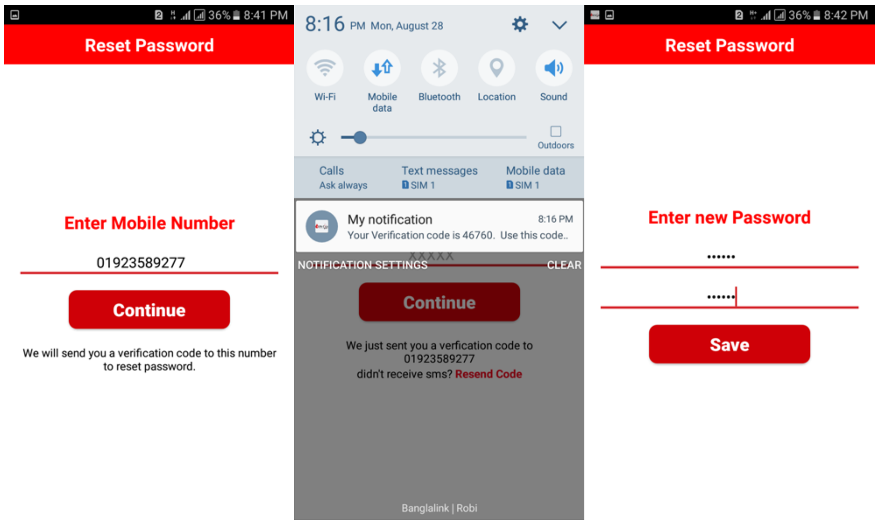
Figure 3.12: Admin Lists   Figure 3.13: My Status   Figure 3.14: To Be Proud

3.5.9 My Status: In Figure 3.13, if any donor user of lifecycle has recently donated blood and do not want to be contacted by patients and do not want to be available on donors search list, can change their donation or profile status from this screen. In case the donor checks “Already Donated” checkbox, he/she will see an automatic donation availability date which is exactly 90 days after from current date. Upon updating status with the above setting, user will not be able to edit this “My Status” screen before 90 days. Of course he/she will receive a confirmation dialog before updating “Already Donated” option. “Available” checkbox is automatically selected by default. In case donor is receiving too many blood requests and is not willing to donate blood in spite of having the donation availability or do not want to be disturbed, can check “Not Available” option leaving a reason(optional) and setting auto available date(optional) from date chooser. The donor’s profile will be visible automatically by system on that specific date. Of course, user can set available at any time in this situation.

3.5.10 To Be Proud: In Figure 3.14, donors can upload their photos and share on social media during the time of blood donation. They can share their experience or feeling too by writing comment. There are two options for taking photos, phone camera and gallery. To take photo using camera user must allow camera permission for the first time. In case of sharing photos from gallery, user has to allow gallery read permission. Shared photos will be saved on server and shown on website and linked social media pages.

3.5.11 Reset password phone number: In case user forgets their password, can change by pressing “Forgot Password?” option from Login screen. User has to provide their phone number to verify the account. If the account exists, user will receive a verification code SMS to verify account’s ownership. If verification is successful user will be taken to “New Password” screen to change the password. Here user will change the account password. This will change only account password, not “Live Chat” login password. To change Live

Chat login password, user has to do it from Live Chat login screen. When user log in into the app or sign up, Live Chat credentials creation and login are performed automatically by the system and user won't face Live Chat login screen. In case the account password is changed or Live Chat credentials creation and Live Chat login were not successful during sign up or login, user will see Live Chat login screen upon pressing on "Live Chat" feature from home menu. Next screen verifies the account ownership of user before taking him to "Change Password" screen.



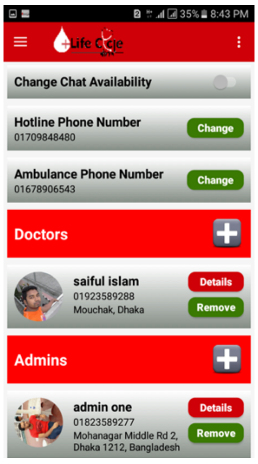
We will send you a verification code to this number to reset password.

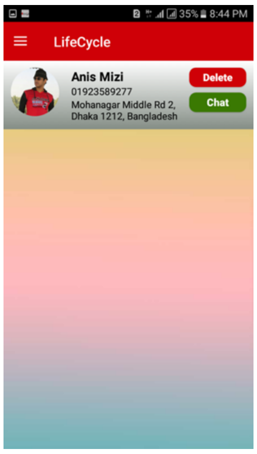
My notification 8:16 PM
Your Verification code is 46760. Use this code..

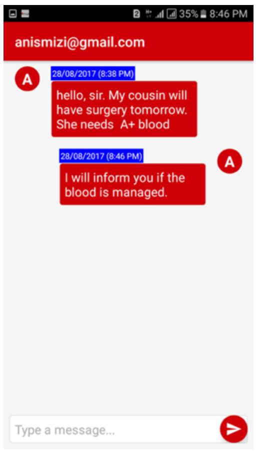
We just sent you a verfication code to 01923589277
didn't receive sms? Resend Code

Figure 3.15: Reset password.

3.5.12 Admin Settings: This, Figure 3.16, feature is only available to admin users. They can change Chat availability, Hotline phone number, Ambulance hotline number, add new doctor, remove existing doctor, add new admin, and remove existing admin from this screen.







28/08/2017 (8:38 PM)
hello, sir. My cousin will have surgery tomorrow. She needs A+ blood

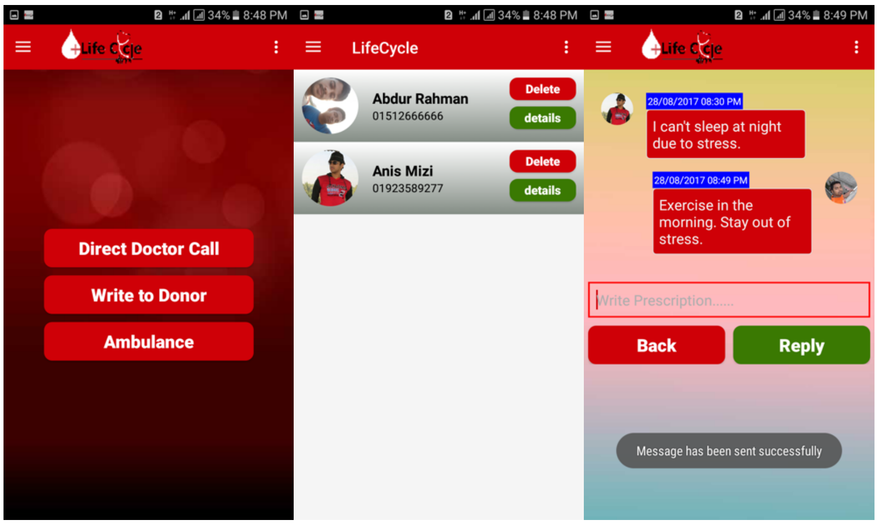
28/08/2017 (8:46 PM)
I will inform you if the blood is managed.

Figure 3.16: Admin Settings. Figure 3.17: Admin Chat list. Figure 3.18: Admin Chat with user.

3.5.13 Admin Chat list: This screen, Figure 3.17, fetches list of users who messaged admin via live chat. Any admin can only see those users who messaged him. The user who messaged last will be shown first. Admin can delete the chat history or start chat with the user from this screen.

3.5.14 Admin Chat with user: The chat screen, Figure 3.18, for admin and user is same. Previous chat history will also be loaded.

3.5.15 Write to Donor option: If the user is doctor and press “Doctor Care” option from home menu will be taken to this screen. Upon clicking on “Write to Donor” option the doctor will see patients list who wrote to him/her. This screen retrieves list of patients who wrote to the doctor. The doctor can delete any message history or see conversation clicking on “details” button. From this screen doctor can write prescription to patient.



Abdur Rahman 01512666666

Anis Mizi 01923589277

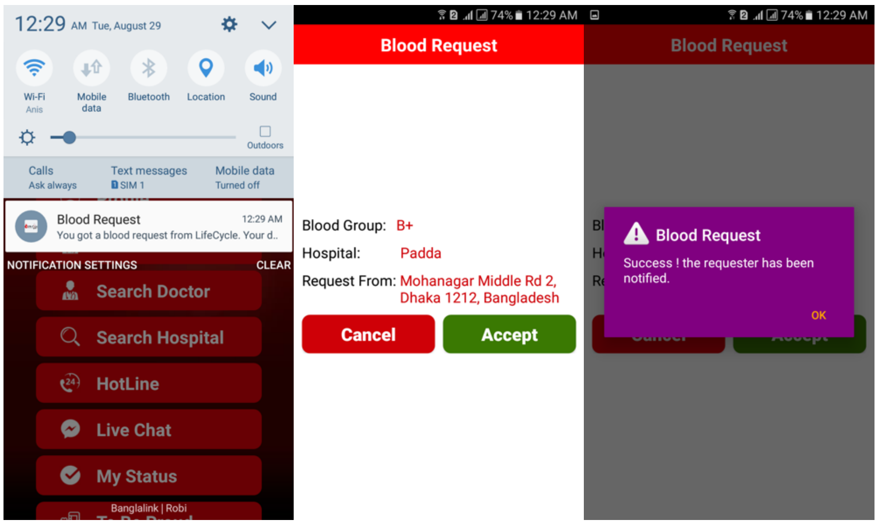
28/08/2017 08:30 PM
I can't sleep at night due to stress.

28/08/2017 08:49 PM
Exercise in the morning. Stay out of stress.

Message has been sent successfully

Figure 3.19: Write to Donor option.

3.5.16 Blood Request to Donors: If any user sends blood request to nearest donors, available nearest donors will receive this notification. Of course, only those donors will be notified who have match according to the users’ search criteria. Upon clicking on blood request, donor will see the blood request details which includes contact hospital name and the location from where the request has been made. The donor can either accept or deny the blood request. If he/she accepts, it means he/she is available to donate. In case the donor denies the request, it means he/she is available but doesn’t want to donate now and the donor’s profile will be hidden for 1 day automatically so that he/she is not disturbed by other requesters on that day. Upon acceptance, donor will see a confirmation dialog if the requester has been notified about his/her availability. If any user gets too many blood requests in a day or wants to hide his/her profile, all he/she needs is to open “My Status” option and set unavailable for a specific period. If users set automatic profile visible date, the profile will be visible on donor search results exactly from that day. Users can also hide their profile for 24 hours by canceling any blood request. After 24 hours, the profile will be automatically visible again by the system.



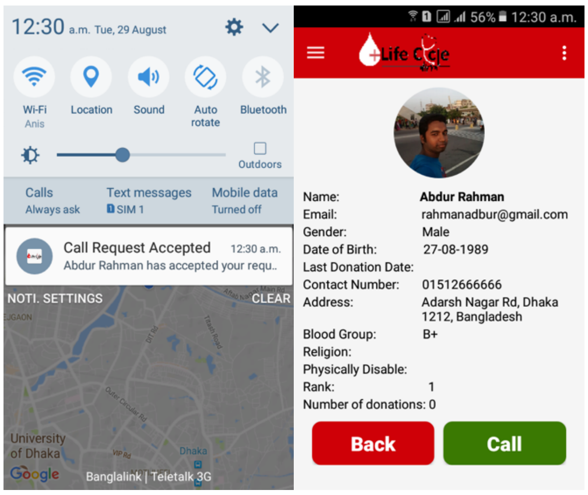
Blood Request 12:29 AM
You got a blood request from LifeCycle. Your d..

Blood Group: B+
Hospital: Padda
Request From: Mohanagar Middle Rd 2, Dhaka 1212, Bangladesh

Blood Request
Success ! the requester has been notified.

Figure 3.20: Blood Request to Donors.

3.5.17 Donor Details: If any donor accepts the blood request, the requester will get this notification. Upon clicking on the notification the requester will be taken to this donor's profile details screen which includes visible phone number of that donor. The requester will see this screen if any donor accepts the blood donation request and can make call by pressing "Call" button.



Call Request Accepted 12:30 a.m.
Abdur Rahman has accepted your requ..

Name:	Abdur Rahman
Email:	rahmanadbur@gmail.com
Gender:	Male
Date of Birth:	27-08-1989
Last Donation Date:	
Contact Number:	01512666666
Address:	Adarsh Nagar Rd, Dhaka 1212, Bangladesh
Blood Group:	B+
Religion:	
Physically Disable:	
Rank:	1
Number of donations:	0

Figure 3.21: Donor Details.

3.6 Chapter Summery

This chapter illustrates the Flowchart and Methodology of LifeCycle. Here I tried to elaborate how LifeCycle works with specific figures. Also a sample outline shown by the flowchart here. The next chapter will illustrate about the implementation of LifeCycle.

4 Implementation

In this chapter, I describe the implementation details of the project. Here I present Android Studio [7], Android Volley [8], JSON [9], XML [10], Google Maps and Places APIs [11], Firebase [12], PHP [13] and MySql [14]. Here we have written our program with Android Studio which runs on JVM [15] and PHP which runs on a server. Here PHP saves and fetches data from MySql database.

4.1 Installing Android Studio

Hardware and Software Required:

1. Windows/Mac/Linux operating system
2. JDK
3. Android SDK
4. Android Studio

To run android studio, the system must have jdk(Jave Development Kit) and android sdk installed. Before installing android studio, we must create java environment variable. Alternatively, we can install Android Studio, The Official IDE for Android, but jdk is must.

4.2 Network Operations with Android Volley

Package Required:

1. Android volley library, version 1.0.19

We can include android volley in android studio package by two ways: download the package and add it as library OR simply compile the project putting this line of code “compile 'com.mcxiaoke.volley:library:1.0.19'” in dependencies section of app based .gradle file.

4.2.1 Implementation

Listing 4.1: Network Operations with Android Volley Code

```
String url = getString(R.string.server_url)+"retreive_nearest_donors.php";
StringRequest stringRequest = new StringRequest(Request.Method.POST,
        url, new Response.Listener<String>() {
    @Override
```

```
        public void onResponse(String response) {
            try {
                JSONArray jsonArray = new JSONArray(response);
                JSONObject mJsonObject;

                if(jsonArray.length() == 0){

Toast.makeText(SearchDonorFragment.this,".",Toast.LENGTH_SHORT).show();
                    new AlertDialog.Builder(SearchDonorFragment.this,
R.style.MyAlertDialogStyle)
                            .setIcon(android.R.drawable.ic_dialog_alert)
                            .setTitle("Not found")
                            .setMessage("No Donors found. Try with different search.")
                            .setPositiveButton("OK", new DialogInterface.OnClickListener() {

                                @Override
                                public void onClick(DialogInterface dialog, int which) {

                                }

                            }).show();
                }
                for (int i = 0; i < jsonArray.length(); i++) {
                    mJsonObject = jsonArray.getJSONObject(i);
                    DataLoader.UserInfo userInfo = new DataLoader.UserInfo();
                    userInfo.phone = mJsonObject.getString("phone");
                    userInfo.blood_group = mJsonObject.getString("blood_group");
                    userInfo.birth_date = mJsonObject.getString("birth_date");
                    userInfo.age = mJsonObject.getString("age");
                    userInfo.last_donation = mJsonObject.getString("last_donation");
                    userInfo.email = mJsonObject.getString("email");
                    userInfo.division = mJsonObject.getString("division");
                    userInfo.district = mJsonObject.getString("district");
                    userInfo.upazila = mJsonObject.getString("upazila");
                    userInfo.address = mJsonObject.getString("address");
                    userInfo.donations_number = mJsonObject.getString("donations_number");
                    userInfo.gender = mJsonObject.getString("gender");
                    userInfo.already_donated = mJsonObject.getString("already_donated");
                    Double lat = Double.parseDouble(mJsonObject.getString("lastLat"));
                    Double lng = Double.parseDouble(mJsonObject.getString("lastLng"));
                    userInfo.latitude = lat;
                    userInfo.longitude = lng;
                    DataLoader.userDetails.add(userInfo);
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(SearchDonorFragment.this,"Network Error. Please try
again.",Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("phone", DataLoader.getUserPhone());
            params.put("filterBlood", SearchByLocation.searchByLocBlood);
            params.put("filterMile", "" + SearchByLocation.searchByLocMile);
            params.put("latitude", "" + latitude);
            params.put("longitude", "" + longitude);
            return params;
        }
    };
```

4.3 Sending and Receiving data using JSON

Package Required:

1. import org.json.JSONArray;
2. import org.json.JSONException;
3. import org.json.JSONObject;

Json is very useful to send or recive large data in array name/value pairs. To add json in android studio project, we need to import these three libraries.

4.3.1 Implementation

Listing 4.2: Creating Json Object Code

```
    JSONObject blood_request = new JSONObject();
    try {
        blood_request.put("requester_phone", DataLoader.getUserPhone());
        blood_request.put("requested_blood", SearchByLocation.searchByLocBlood);
        blood_request.put("bags", SearchByLocation.total_bags);
        blood_request.put("hospital", SearchByLocation.hospital);
        blood_request.put("address", SearchByLocation.request_address);

    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
```

Listing 4.3: Fetching data from Json Object

```
JSONArray jsonArray = new JSONArray(response);
JSONObject mJsonObject;
mJsonObject = jsonArray.getJSONObject(i);
DataLoader.UserInfo userInfo = new DataLoader.UserInfo();
userInfo.phone = mJsonObject.getString("phone");
userInfo.blood_group = mJsonObject.getString("blood_group");
userInfo.birth_date = mJsonObject.getString("birth_date");
userInfo.age = mJsonObject.getString("age");
userInfo.last_donation = mJsonObject.getString("last_donation");
userInfo.email = mJsonObject.getString("email");
userInfo.division = mJsonObject.getString("division");
userInfo.district = mJsonObject.getString("district");
userInfo.upazila = mJsonObject.getString("upazila");
userInfo.address = mJsonObject.getString("address");
userInfo.donations_number = mJsonObject.getString("donations_number");
userInfo.gender = mJsonObject.getString("gender");
userInfo.already_donated = mJsonObject.getString("already_donated");
Double lat = Double.parseDouble(mJsonObject.getString("lastLat"));
Double lng = Double.parseDouble(mJsonObject.getString("lastLng"));
userInfo.latitude = lat;
userInfo.longitude = lng;
```

4.4 Declaring UI elements in XML

A XML layout in android studio defines the visual structure of a user interface, such as the UI for an activity or app widget. we can declare a xml layout in two ways: 1. Creating a xml file 2. Instantiating layout elements in runtime

When we create an activity in android studio, a xml layout file is created automatically. Otherwise we can generate the layout file manually.

4.4.1 Implementation

Listing 4.4: Declaring UI elements in XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="horizontal"
   android:layout_marginBottom="15dp">

   <ImageView
      android:id="@+id/listIcon"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@mipmap/ic_launcher"
      android:layout_marginLeft="20dp"
      android:layout_centerVertical="true" />

   <TextView
      android:id="@+id/listTitle"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_vertical"
      android:layout_marginLeft="20dp"
      android:text="Home Options"
      android:textColor="#000"
      android:textSize="15sp" />

</LinearLayout>
```

4.5 Google Maps and Location APIs

Package Required:

1. Play services maps, version 11.0.1
2. Play services location, version 11.0.1

We need google map to show nearest donors as markers and their distances from patient's location. We can include Google Play services maps and location in android studio package by two ways: download the package and add it as library OR simply compile the project putting this lines of code "compile 'com.google.android.gms:play-services-maps:11.0.1'" and "compile 'com.google.android.gms:play-services-location:11.0.1'" in dependencies section.

4.5.1 Implementation

Listing 4.5: Adding markers on Google Maps

```
MarkerOptions markerOptions = new MarkerOptions();
markerOptions.position(latLng);
markerOptions.title(userInfo.fname+" "+userInfo.lname);
markerOptions.snippet(userInfo.address);
switch (userInfo.blood_group){
case "A+":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.apositive));
break;
case "A-":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.anegative));
break;
case "B+":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.bpositive));
break;
case "B-":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.bnegative));
break;
case "O+":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.opositive));
break;
case "O-":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.onegative));
break;
case "AB+":
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.abpositive));
break;
case "AB-":
```

```
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.abnegative));
break;
default:
markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE
_MAGENTA));
break;


}
mMap.addMarker(markerOptions);
```

4.6 Implementing Live Chat using Firebase

Package Required:

1. Firebase Messaging, version 11.0.1

To add firebase in our project we need to, first, add rules to root-level build.gradle file which are: classpath 'com.google.gms:google-services:3.1.1' // google-services plugin
And
url "https://maven.google.com" // Google's Maven repository

Then, in module Gradle file (usually the app/build.gradle), we need to add the apply plugin line at the bottom of the file to enable the Gradle plugin: “apply plugin: 'com.google.gms.google-services'”
and “compile 'com.google.firebase:firebase-core:11.4.2'” in dependencies.

4.6.1 Implementation

Listing 4.6: Implementing Live Chat using Firebase

```
@Override
public void sendMessageToFirebaseUser(final Context context, final Chat chat, final String
receiverFirebaseToken) {
final String room_type_1 = chat.senderUid + "_" + chat.receiverUid;
final String room_type_2 = chat.receiverUid + "_" + chat.senderUid;
Log.d("chatTrace","Inside sendMessageToFirebaseUser ChatInteractor");

final DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference();
```

```
databaseReference.child(Constants.ARG_CHAT_ROOMS).getRef().addListenerForSingleV
alueEvent(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if (dataSnapshot.hasChild(room_type_1)) {
Log.d("chatTrace","Inside sendMessageToFirebaseUser ChatInteractor room_type_1
"+room_type_1);
Log.e(TAG, "sendMessageToFirebaseUser: " + room_type_1 + " exists");
databaseReference.child(Constants.ARG_CHAT_ROOMS).child(room_type_1).child(String.
valueOf(chat.timestamp)).setValue(chat);
} else if (dataSnapshot.hasChild(room_type_2)) {
Log.d("chatTrace","Inside sendMessageToFirebaseUser ChatInteractor room_type_2
"+room_type_2);
Log.e(TAG, "sendMessageToFirebaseUser: " + room_type_2 + " exists");
databaseReference.child(Constants.ARG_CHAT_ROOMS).child(room_type_2).child(String.
valueOf(chat.timestamp)).setValue(chat);
} else {
Log.d("chatTrace","Inside sendMessageToFirebaseUser ChatInteractor success else");
Log.e(TAG, "sendMessageToFirebaseUser: success");
databaseReference.child(Constants.ARG_CHAT_ROOMS).child(room_type_1).child(String.
valueOf(chat.timestamp)).setValue(chat);
getMessageFromFirebaseUser(chat.senderUid, chat.receiverUid);
}
//send push notification to the receiver
sendPushNotificationToReceiver(chat.sender,
chat.message,
chat.senderUid,
new SharedPrefUtil(context).getString(Constants.ARG_FIREBASE_TOKEN),
receiverFirebaseToken);
mOnSendMessageListener.onSendMessageSuccess();
}

@Override
public void onCancelled(DatabaseError databaseError) {
mOnSendMessageListener.onSendMessageFailure("Unable to send message: " +
databaseError.getMessage());
}
});
}

private void sendPushNotificationToReceiver(String username,
String message,
String uid,
String firebaseToken,
String receiverFirebaseToken) {
FcmNotificationBuilder.initialize()
.title(username)
.message(message)
.username(username)
.uid(uid)
.firebaseToken(firebaseToken)
.receiverFirebaseToken(receiverFirebaseToken)
.send();
}

@Override
public void getMessageFromFirebaseUser(String senderUid, String receiverUid) {
final String room_type_1 = senderUid + "_" + receiverUid;
final String room_type_2 = receiverUid + "_" + senderUid;
Log.d("chatTrace","Inside sendMessageToFirebaseUser ChatInteractor
getMessageFromFirebaseUser");

final DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference();

databaseReference.child(Constants.ARG_CHAT_ROOMS).getRef().addListenerForSingleV
alueEvent(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if (dataSnapshot.hasChild(room_type_1)) {
Log.d("chatTrace","Inside sendMessageToFirebaseUser ChatInteractor
getMessageFromFirebaseUser room_type_1 "+room_type_1);
Log.e(TAG, "getMessageFromFirebaseUser: " + room_type_1 + " exists");
FirebaseDatabase.getInstance()
.getReference()
.child(Constants.ARG_CHAT_ROOMS)
.child(room_type_1).addChildEventListener(new ChildEventListener() {
@Override
public void onChildAdded(DataSnapshot dataSnapshot, String s) {
Chat chat = dataSnapshot.getValue(Chat.class);
mOnGetMessagesListener.onGetMessagesSuccess(chat);
}
```

4.7 Interacting with MySql database using PHP

Hardware/Software Required:

1. Web Hosting
2. PHP
3. MySql Database

We need a Linux or Windows based server to run php scripts and interact with MySql database. Our Android application will save or fetch data from the MySql database running php scripts on server.

4.7.1 Implementation

Listing 4.7: Saving into database

```
$sql = "insert into
donors(phone,password,pic_path,fname,lname,blood_group,birth_date,age,last_donation,new
_donor,email,division,district,thana,address,latitude,longitude,code,verification,lastLat,lastLn
g,fcm_email,fcm_uid,fcm_token,pro_visible,called_date,called_today,donations_number,use
r_type,gender,already_donated,autopro_visible,singup_steps,post_code,rank,web_url,fb_url,p
rofile_photo,religion,is_physically_disble,nationality,nid,status,updated_at,created_at,update
d_by,created_by)
	values('".$phone."','".$password."','".$pic_path."','".$fname."','".$lname."','".$blood_gr
oup."','".$birth_date."','".$age."','".$last_donation."','".$new_donor."','".$email."','".$division."
','".$district."','".$thana."','".$address."','".$latitude."','".$longitude."','".$code."','".$verification
."','".$lastLat."','".$lastLng."','".$fcm_email."','".$fcm_uid."','".$fcm_token."','".$pro_visible."'
,'".$called_date."','".$called_today."','".$donations_number."','".$user_type."','".$gender."','".$
already_donated."','".$autopro_visible."','".$singup_steps."','".$post_code."','".$rank."','".$we
b_url."','".$fb_url."','".$profile_photo."','".$religion."','".$is_physically_disble."','".$nationalit
y."','".$nid."','".$status."','".$updated_at."','".$created_at."','".$updated_by."','".$created_by."')
;";
	if(mysqli_query($conn,$sql)){
		echo "inserted";
		mysqli_close($conn);
	}
```

Listing 4.8: Retrieving from database

```
	require_once("database_connect.php");

	$all_user = array();
	$i = 0;
	$sql = "select * from donors";
```

```
$total = mysqli_query($conn, $sql);
if(mysqli_num_rows($total)>=1){
        while($rows = mysqli_fetch_assoc($total)){

                        $user = new User;
                        $user->pic_path = $rows['pic_path'];
                        $user->fname = $rows['fname'];
                        $user->lname = $rows['lname'];
                        $user->email = $rows['email'];
                        $user->phone = $rows['phone'];
                        $user->birthDate = $rows['birthDate'];
                        $user->bloodgroup = $rows['bloodgroup'];
                        $user->location = $rows['location'];
                        $user->latitude = $rows['latitude'];
                        $user->longitude = $rows['longitude'];
                        //$all_usr[$i] = $user;
                        //echo json_encode($user);
                        array_push($all_user, $user);
                $i++;
        }

}

echo json_encode($all_user);

mysqli_close($conn);

class User{
        var $pic_path;
        var $fname;
        var $lname;
        var $email;
        var $phone;
        var $birthDate;
        var $bloodgroup;
        var $location;
        var $latitude;
        var $longitude;
}
```

5 Conclusion

In this very last chapter of the book we conclude all of our introduction, review, design and implementations. However, for this last chapter we would like to discuss about those problems we face during the development process of the project and also like to talk about our future goals concerning this very project.

5.1 Limitation

In the world nothing is perfect. Everything has some limitations. LifeCycle has also some limitations. Here are some limitations which I found after using the app.

I. User cannot use the application without internet connection.
II. Volunteers need to populate the hospitals and doctors list in database manually.
III. Though the application has rich functionalities but the UI design is a bit old fashioned comparing with other latest applications in 2017.

I have some future plan to resolve these limitations which I will describe in my Future Plan section.

5.2 Future Plan

There are some limitations in this project which have been included as future plans to be considered.

1. Maintaining a local database so that user can access some features, like “Doctor Search” and “Hospital Search”, offline and that database will be updated when connected to internet.

2. Implementing social login and signup system which will improve user experience. User will be able to sign up using their Facebook, twitter, or other social account instead of filling up forms.

3. Replacing the whole UI design with Google material design, is used by most of the android applications available in Google play store.

5.3 Conclusion

I have tried to make this app simple and efficient to use for two type of users, patients and donors. I hope people of Bangladesh will be able to manage blood for their friends or family in time if they use our application. In future, I have a lot of dreams about this app. To make this app more efficient, any kind of constructive idea will be mostly appreciated and I will be grateful.

References

[1] https://www.lucidchart.com/, "flowchart maker", Draw flow charts & collaborate with others online

[2] Blood Donor, "American Red Cross", https://play.google.com/store/apps/details?id= com.cube.arc.blood, Updated On August 29, 2017, Current Version 1.4.6.

[3] Simply Blood -Find Blood Donor, "Thatspood", https://play.google.com/store/apps/ details?id=com.simplyblood, Updated On October 8, 2017, Current Version 2.0.

[4] Blood Donor Finder, "Thatspood", https://play.google.com/store/apps/details?id=com .dos.sunnat.blooddonorfinder, Updated On January 2, 2015, Current Version 2.5.

[5] Socialblood, "Socialblood Inc", https://play.google.com/store/apps/details?id= org.socialblood.app, Updated On January October 26, 2016, Current Version 1.0.5.

[6] Henry Gantt, "Gantt chart", http://www.gantt.com/, © 2016 Gantt.com

[7] Android Studio, "The Official IDE for Android", https://developer.android.com/ studio/ index.html, Current Version 2.3.3

[8] Android Volley, "HTTP library that makes networking for Android apps easier", https:// developer.android.com/training/volley/index.html, Current Version 1.0.19

[9] JSON, "JSON (JavaScript Object Notation) is a lightweight data-interchange format", http://www.json.org/

[10] XML, "Extensible Markup Language (XML)", https://en.wikipedia.org/wiki/XML

[11] Google Places API, "Google Places APIs belong to the family of Google Maps APIs", https://developers.google.com/places/

[12] Firebase, "Firebase is Google's mobile platform", https://firebase.google.com/

[13] Firebase, "a server scripting language", https://secure.php.net/, Current Version 7.2.0

[14] MySQL, "MySQL is an open-source relational database management system (RDBMS)", https://www.mysql.com/

[15] JVM, "A Java virtual machine", https://en.wikipedia.org/wiki/Java_virtual_machine