

CS5551 ADVANCED SOFTWARE ENGINEERING

BLOOD DONATION MANAGEMENT

Project Report

Instructor :

Dr. Yugyung Lee

TEAM MATES :

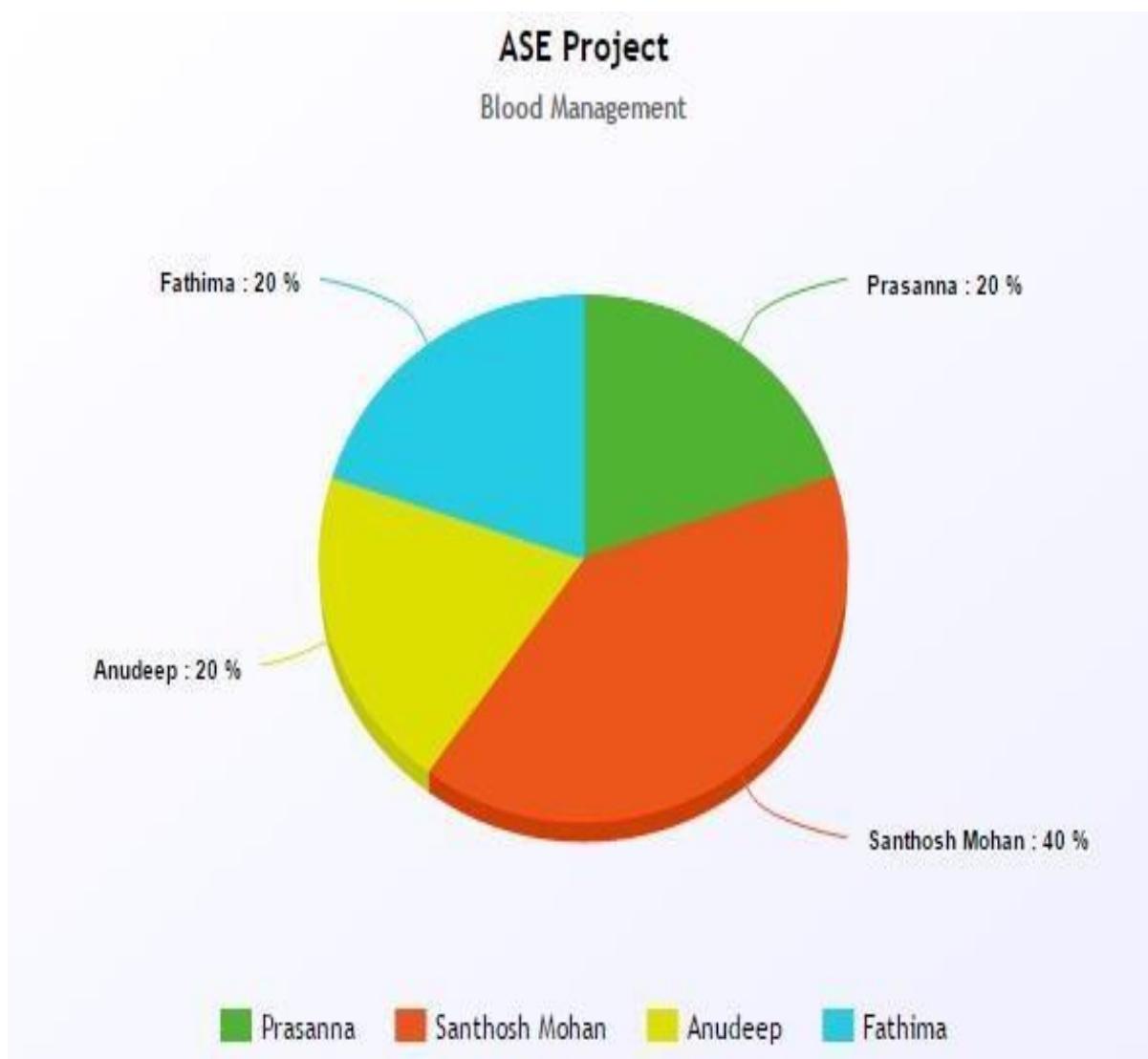
- 1. PRASANNA MUPPIDI (33)**
- 2. SANTHOSH MOHAN(35)**
- 3. ANUDEEP PANDIRI (40)**
- 4. Fathima James S (58)**

Index

1. Overall Contribution	3
2. Proposal	4
3. User Manual	6
4. Project Management Report.....	14
5. Increment 1.....	21
6. Increment 2.....	45
7. Increment 3.....	78
8. Increment 4.....	122

Contribution:

1. Santhosh Mohan - 40%
2. Prasanna Muppidi - 20%
3. Anudeep Pandiri - 20%
4. Fathima James - 20%



PROJECT PROPOSAL

Blood Donation

Overview:

Blood is one of the most critical things people need at emergencies. A lot of times, we have seen people struggle for blood at the times it's most needed. There are lot of blood donation camps available in every city but what happens a lot of times is the wastage of blood because blood needs to be stored in a very significant way and the lifetime of the blood in bottles is less. This made us think about an online application which helps people finding instant blood donors.

Our Android application on blood donation helps the users to find the donor for a particular blood group in a particular location in the easiest way possible. All the donors who wish to donate blood are supposed to enter their details in the application. The details include their Name, Age, Contact number, Email ID, Location (ZIP Code) and most importantly their blood group.

The recipients who wish to receive blood should login to the application with their mobile number and are supposed to search the forum based on the required blood group and the location, they can contact the donor for further details.

It's a simple application which doesn't ask the users for unnecessary information. The donors are sorted based on the blood group and their location which makes the recipients easy while searching.

Technologies Used:

ADT: Android Studio

Programming language: JAVA

Database: SQLite/MySQL

Frontend: XML, JavaScript

There are other applications similar to our application. The following are the links for those applications.

1. <https://play.google.com/store/apps/details?id=com.cube.arc.blood&hl=en>
2. <https://play.google.com/store/apps/details?id=com.ibd.blooddonorapp&hl=en>

References:

1. <http://developer.android.com/sdk/index.html>
2. <http://developer.android.com/training/basics/data-storage/databases.html>
3. <http://projectsgeek.com/2013/10/online-blood-donation-management-system-project.html>
4. <https://guides.github.com/>

User Manual

Motivation & decision:

The main idea of this project is to find the specific blood type donor efficiently. "Donate Blood, Save Life" as per the quote, we are trying to save life by easily search the blood donors in near by geolocation. Even though, there is lot of options to find the donors. We are trying to find a donor in a fast and time efficient manner. The user can get the donor information along with their address, phone no and Google map directions. We are providing lot of choices to find the blood donor or blood banks information. By using our app, the user can search the available blood group type via hospitals, blood drive, blood bank and Facebook share. For that purpose, we have the user friendly functionalities such as,

- ❖ Hospitals
- ❖ Blood drive
- ❖ Blood bank
- ❖ Share on Facebook

By using any of these above functionalities, the user can easily get the donor's information very quickly. The volunteer donors also have an account with us in order to help and support other people as they do. So the donors also search other doors information along with Google direction. Thus, this app is all about the saving a life. Simply, we are making everything easy and friendly here. Just because "Life is beautiful" So Donate blood and save Life. To implement our idea, we used Android studio, Firebase, Java & XML.

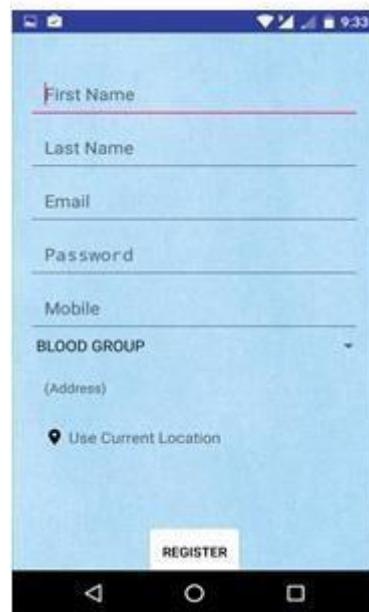
The style of user interaction:

--> The user can interact with our app by login their information. The user should register first, if they don't have an existing account and then they can login through Facebook as well. This below screenshot of our app shows the login page where the user can login through the social media like Facebook. But that option is only for the

existing user. The new user should have to register first and then login via social media. It emphasizes that no need of going to our app every time for login.



--> The registration page contains the user's normal basic information. As shown in the below diagram, user can enter their information along with their credentials. If there is any invalid data entered, then the validation error will be thrown.



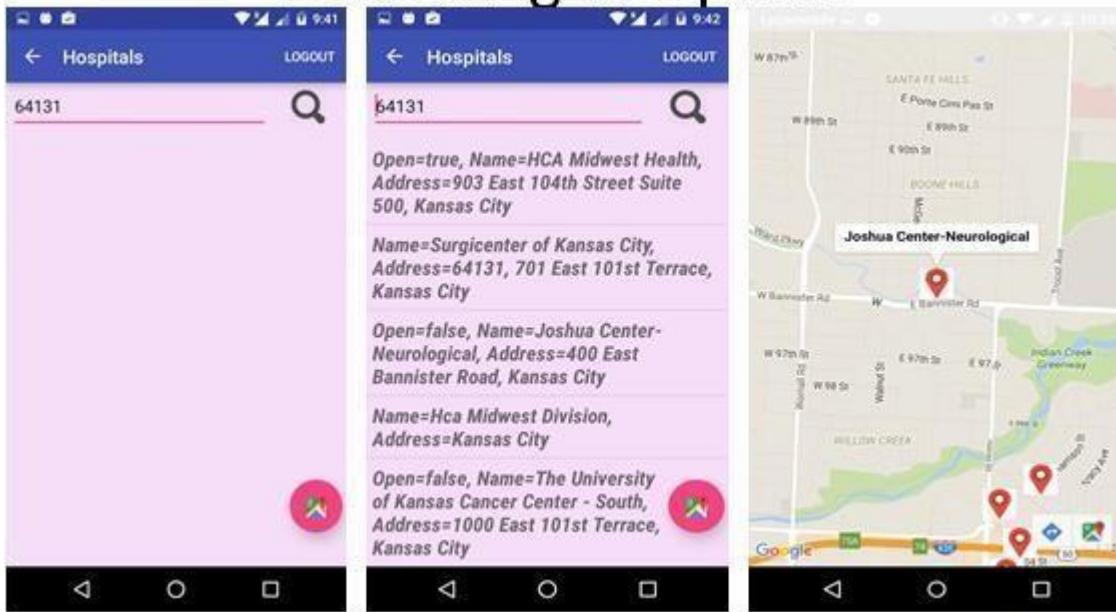
--> After creating an account, the user can easily login through our app as shown in the below screenshot. There is no need of login via username. Instead, the user can login with their phone no as well.



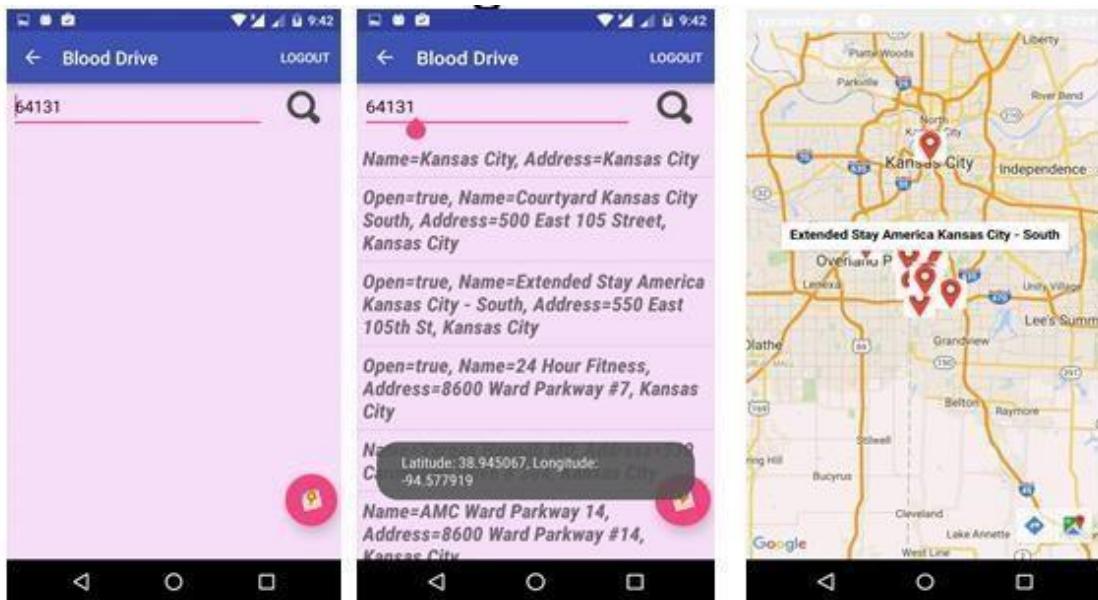
--> After login, the user can search their blood group type in home page. The search result will contain the list of available donors information along with their address, phone no and Google map directions as shown in the below screenshot.



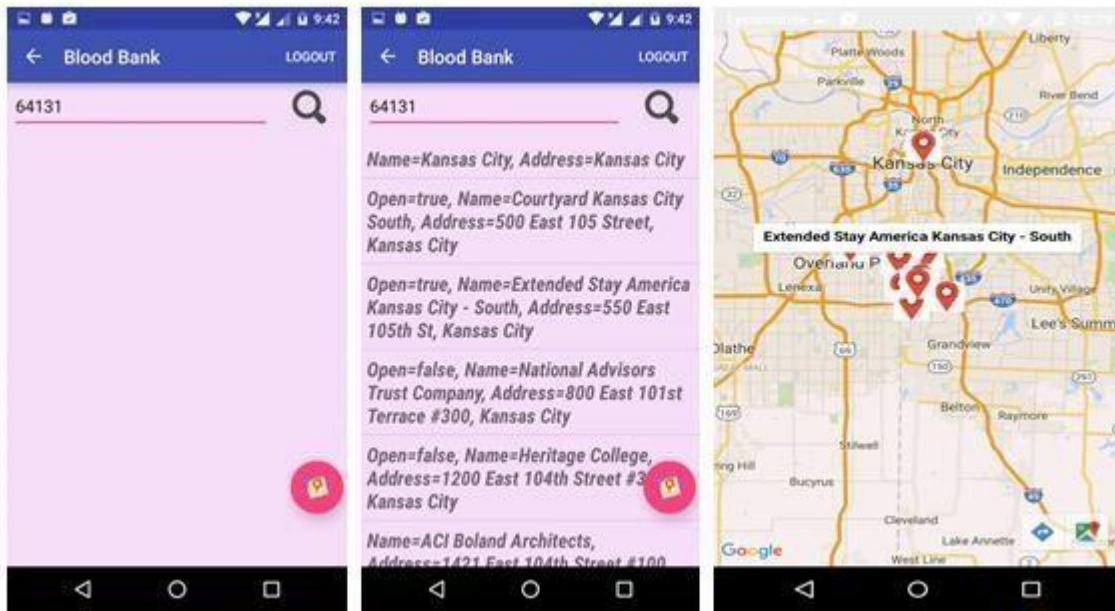
--> The left navigation bar contains the additional functionalities such as Hospitals, Blood drive, Blood bank and share on Facebook. By giving the area zip code, the user can get the near by hospital name, address and Google direction as well. Same like hospital, the user can get the near by Blood drive and Blood bank as well. Finally the user can share the needed blood group information in their Facebook wall as well.



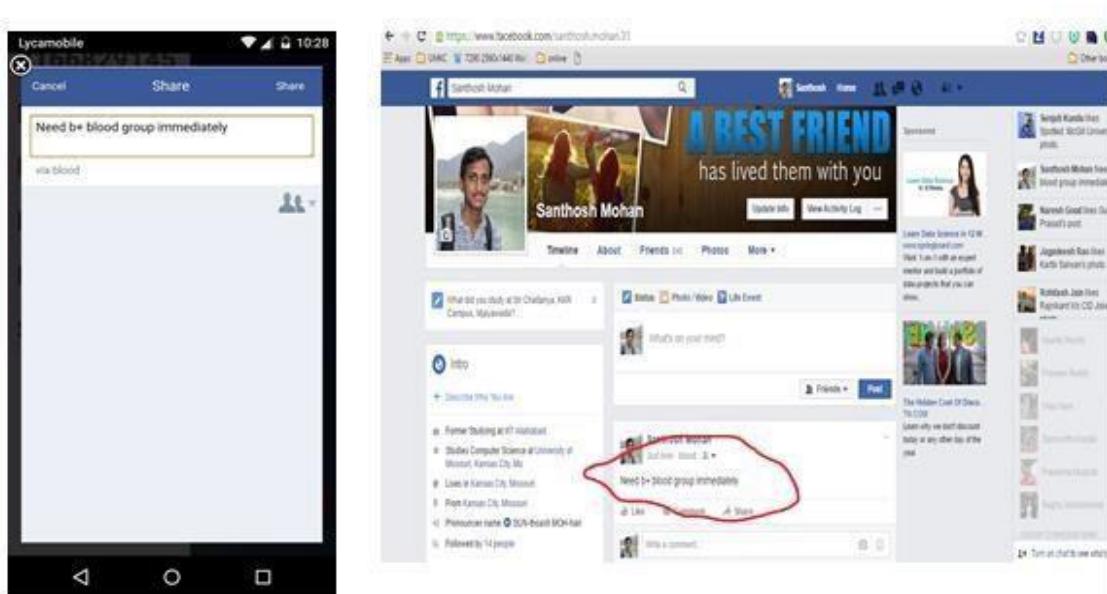
--> As shown in the below diagram, user can search the blood drive information along with address and direction. The one more thing about this functionality is, the user can get to know that the particular blood drive is open or closed.



--> This below screenshot shows the Blood bank information along with Google map direction. By giving the zip code in the search box, the user can get the near by blood banks addresses along with the blood bank closing information.



--> Here is another important functionality of this app. We are providing lot of choices to find the blood donor or blood banks information. The user can share the needed blood group along with their own message in Facebook as shown in the below screenshot.



--> Last but not least, here comes the profile and edit page. As shown in the below diagram, the user can view the profile and also the user can easily edit and update their information, by clicking the right side profile icon in home page.



--> As shown in the figure, currently the user "Santhosh Mohan" is trying to edit his profile page.



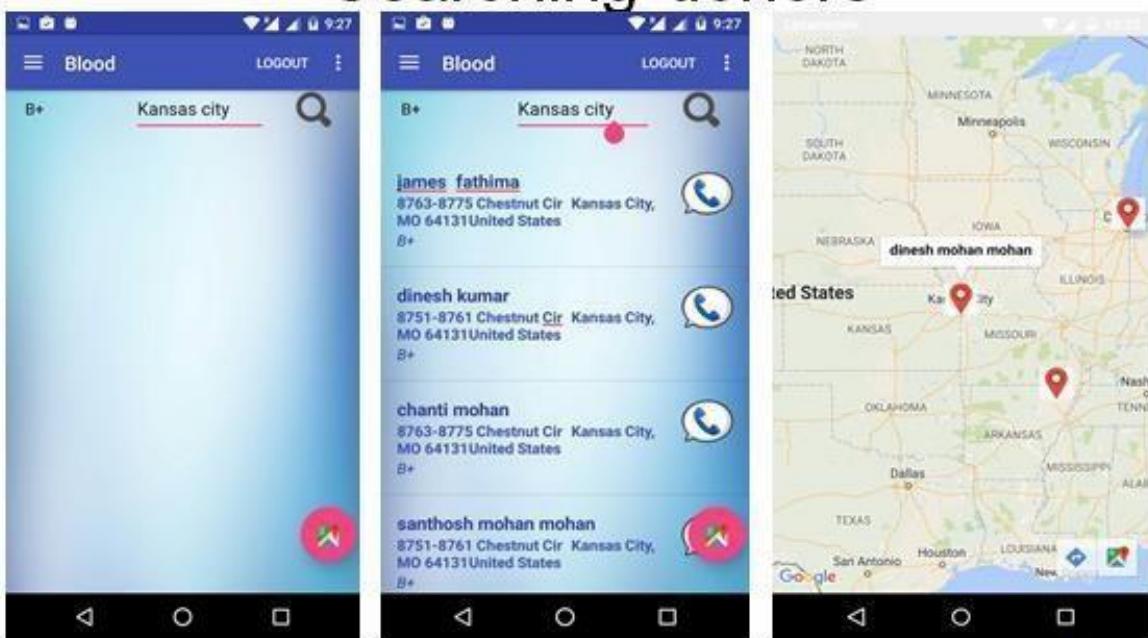
Error recognition and handling:

We have validation in login page and edit page as well. If the user login without giving password then the validation error will be thrown. In the edit page, the user give decimal number for email and character for zip code then the error will be thrown.

Sample interaction:

If the user first select the blood group and search it based on the city, then they will get the list of donors information. Here the user is directly interact with the backend service in order to get the search results.

The screenshot is the best example of the sample interaction. The user is searching with the city name "Kansas city" and then get the available donors information along with Google map directions. We are making everything easy here.



A list of known bugs and deficiencies

We do have some bugs in edit page but it won't affect the over all performance.

Major bug:

- ✓ Map view will be available after the list view. So if someone presses map button before list view, app will crash.

Limitation:

- ✓ There is no free API who provides blood donor list. So limited donors are available in database.

Project Management Report

Introduction:

Our Blood Management application helps the users to find the donor for a particular blood group in a particular location in the easiest way possible. All the donors who wish to donate blood are supposed to enter their details in the application. The details include their Name, Age, Contact number, Email ID, Location (ZIP Code) and most importantly their blood group. The recipients who wish to receive blood should login to the application with their mobile number and are supposed to search the forum based on the required blood group and the location, they can contact the donor for further details.

We have divided the entire project into 4 increments.

Increment 1:

1. Research on importance of Blood Donation Management, requirements for the development.
2. Setup Android Studio, Zenhub and Github for every team member.
3. Design Class diagrams.
4. Design Sequence diagrams.
5. Design State diagrams.
6. Design Wireframes.
7. Design basic UI layouts (Login, Registration, Home)

- Prasanna has done Class diagram and Sequence diagram for the application.
- Santhosh has done State Diagram for the application.
- Anudeep has done the Wireframes of all the pages.
- Fathima has done the sample UI for Login, Registration and Home pages.

Increment 2:

1. Implement Login/logout /Registration/ Registration validation and functionality.
2. Implement home page.
3. Setup Firebase database.
4. Test Login/logout UI/Registration/Registration validation.
5. Test sample user data.
 - Prasanna has done the Login, Registration, Home page validations.
 - Santhosh has done the functionalities for Login, Registration and Home pages.
 - Anudeep has done the UI for all the pages.
 - Fathima has setup the Firebase database.

Increment 3:

1. Implement search functionality
2. Implement View and Edit Profile Functionality
3. Test search functionality
4. Test Edit Profile UI and functionality
5. Implementation of field validations.
 - Prasanna has done the Profile view and validations.
 - Santhosh has done the database queries for the Search functionality and the list view and the map view for the search.
 - Anudeep has done the Edit Profile page and the database.
 - Fathima has done the UI for all the pages and Social Login for the application.

Increment 4:

1. Implementation of Side Navigation Bar.
 2. Implementation of Search for Blood Drives, Hospitals, Blood Banks.
 3. Implementation of Share on FB option.
 4. List View and Map View for all the searches.
 5. Refined UI for all the pages.
- Prasanna has done the map view of Blood Bank search and UI for all the pages.
 - Santhosh has done the Side Navigation Bar, Hospital search and Share on FB.
 - Anudeep has done the list view and map view of Blood drive search.
 - Fathima has done the list view of Blood bank search and UI for the search pages.

Final Project Evaluation:

We have implemented all the functionalities which we thought we should implement.

- We have implemented the search according to the Blood group and City.
- We have implemented the Map view and the list view of the donors.
- We have implemented the Navigation Bar.
- We have implemented the list view and the Map view of Hospital search.
- We have implemented the list view and the Map view of Blood Drive search.
- We have implemented the list view and the Map view of Blood banks search.
- We have implemented the Share on FB option.

Yes, we did stick to our project plan in all our increments.

Agile Process Uses in our project:

- Implementation of project in phases.
 - Design, Development and Testing is done simultaneously.
 - Dynamic changes are implemented easily.
1. We would definitely use Agile process the next time we do a project since it is very easy as changes are done in the requirements generally.
 2. All of our team members have worked really well and we didn't have any issues while working.

Github and Zenhub Images:

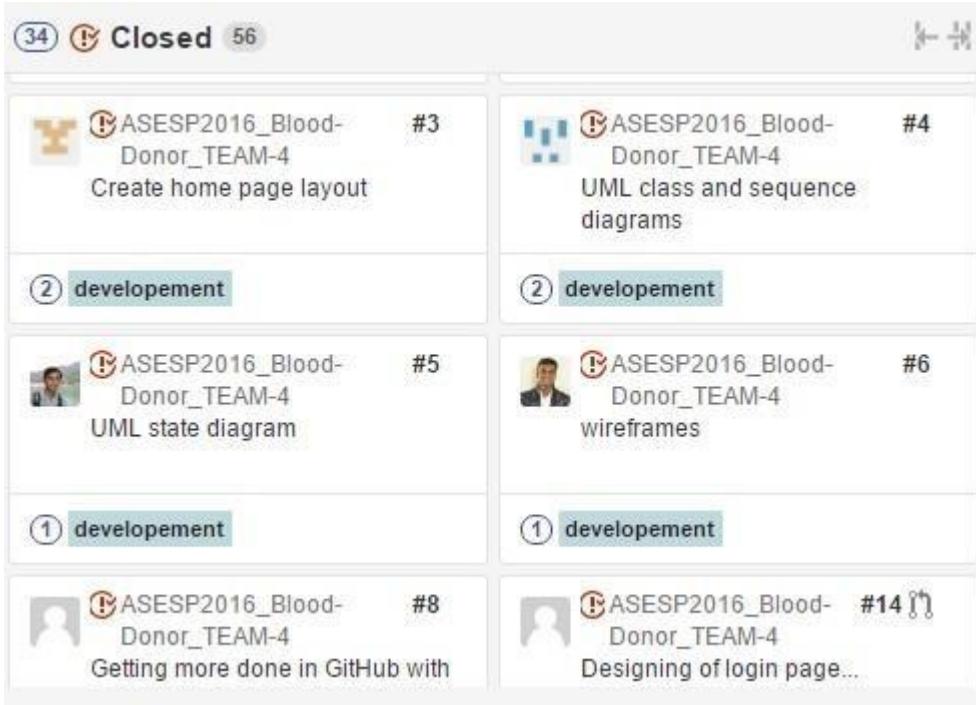


Figure: Increment 1

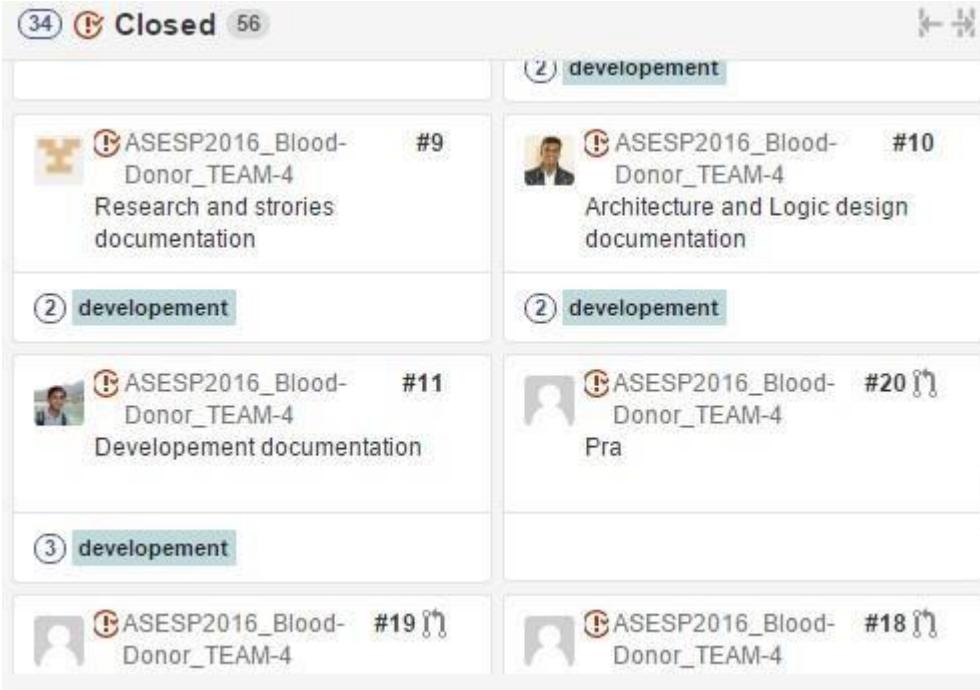


Figure: Increment 2

Closed 56		
 ASESP2016_Blood-#24 Donor_TEAM-4 Anudeep	 ASESP2016_Blood-#15 Donor_TEAM-4 conclusion	#15
	(2) developement	
 ASESP2016_Blood-#23 Donor_TEAM-4 Anudeep	 ASESP2016_Blood-#22 Donor_TEAM-4 Pra	#22
 ASESP2016_Blood-#21 Donor_TEAM-4	 ASESP2016_Blood-#7 Donor_TEAM-4	#7

Figure: Increment 3

Closed 56		
 ASESP2016_Blood-#34 Donor_TEAM-4 login authorization and data retrival for home page	 ASESP2016_Blood-#32 Donor_TEAM-4 Getting more done in GitHub with ZenHub	#32
 ASESP2016_Blood-#31 Donor_TEAM-4 data base activity	 ASESP2016_Blood-#26 Donor_TEAM-4 create a database	#26
	(2) developement	
 ASESP2016_Blood-#27 Donor_TEAM-4	 ASESP2016_Blood-#25 Donor_TEAM-4	#25

Figure: Increment 4

Feb 14, 2016 – May 13, 2016

Contributions: **Commits** ▾

Contributions to master, excluding merge commits

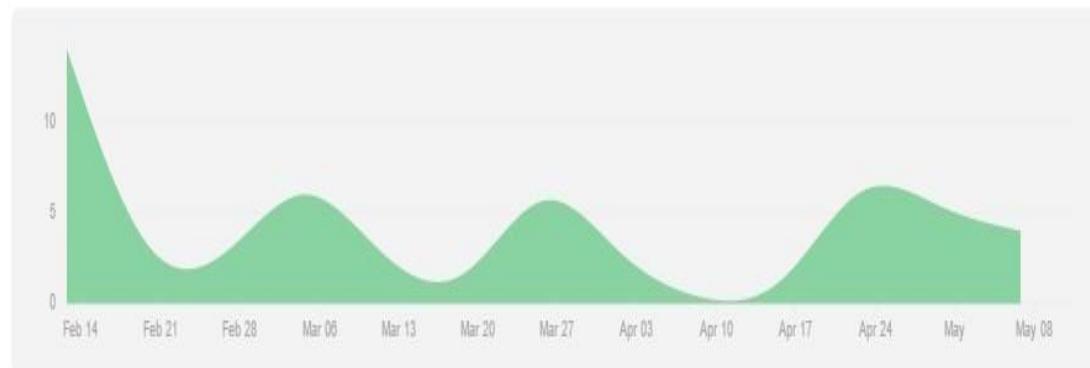


Figure: Overall Contribution Graphs

CS5551 ADVANCED SOFTWARE ENGINEERING

PROJECT REPORT-PHASE 1

BLOOD DONATION MANAGEMENT

TEAM MATES

- 1. PRASANNA MUPPIDI(33)**
- 2. SANTHOSH MOHAN MURARISHETTI(35)**
- 3. ANUDEEP PANDIRI(40)**
- 4. FATHIMA JAMES(58)**

1. Introduction

Our Blood Management application helps the users to find the donor for a particular blood group in a particular location in the easiest way possible. All the donors who wish to donate blood are supposed to enter their details in the application. The details include their Name, Age, Contact number, Email ID, Location (ZIP Code) and most importantly their blood group. The recipients who wish to receive blood should login to the application with their mobile number and are supposed to search the forum based on the required blood group and the location, they can contact the donor for further details.

The project has been divided into four phases with improving implementation features. For the first iteration of our project we want to complete all design section of the application with Login, Registration, Home page designs. We have chosen the android platform to develop our application. For this first phase, we have designed

the UML Class diagram, Sequence diagram, State diagrams along with wireframes. We concentrated mainly on the design part which play a major role in implementing our project.

2. Functions

2.1 Login

Login Page lets the user login to the application. User needs to enter the Username and

Password in order to login. If the user doesn't have an account to Login, he should register first.

2.2 Register

Register page lets the user to register. User needs to provide personal information to create an account.

2.3 Home

In Home Page, already registered user details like Name, Zip Code, Contact Numbers are displayed.

2.4 Search

Once the user registers, all his personal information is stored in SQLite database. User who needs blood should login into the application and search based on the blood group and Zip code and he can then contact the donor based on the contact information present in the application.

2.5 Admin

Users can anytime contact Admin for emergency situations. If any of the contact details or the Blood Group is different from what is required, Admin helps the user for more information.

3. Proposed System

1. Requirement Specification:

□ Functional Requirements:

- i. User should have a Sign in.
- ii. If user is new, he should have a Register. iii. User should be able to provide his personal information.
- iv. User details should be valid.
- v. User should be able to search based on the Blood Group. vi. User should be able to search based on the Zip Code.
- vii. User should be able to view the corresponding contact details. viii. User should be able to contact the Admin in emergency situations.
- ix. User should be able to contact the donor whose details user found on the application.
- x. User should be able to close the session. xi. User should be able to logout.

4. Technologies Used

- i. ADT: Android Studio ii. Programming language: JAVA iii. Database: SQLite
- iv. Frontend: XML, JavaScript

5. Development

System Designs:

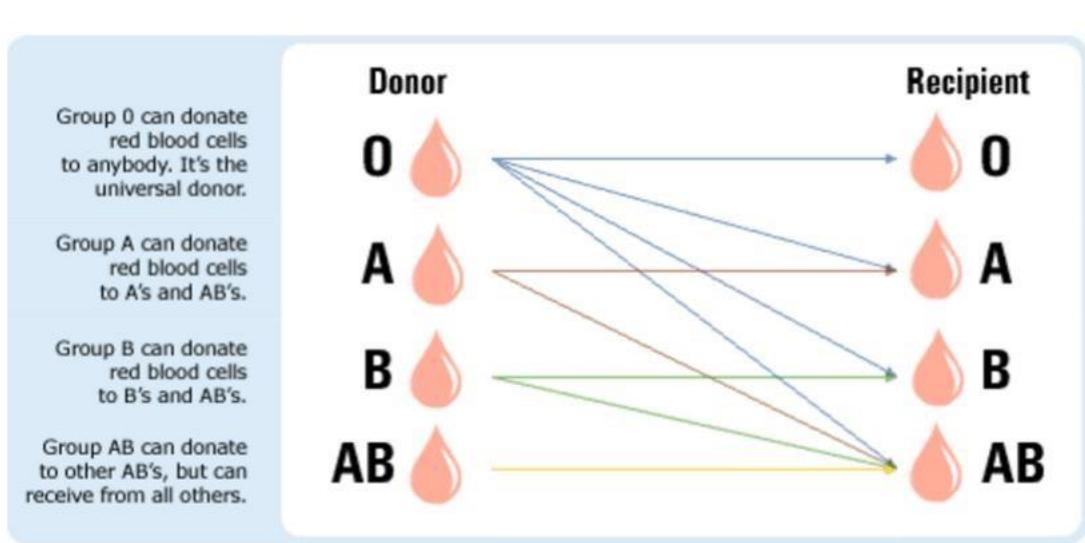
- 1) Login: Allow users to login into their account. Users use email address and Password to identify themselves.

- 2) Registration: Allow new users to create an account. User information collected during this process:
 - First name
 - Last name
 - Date of birth
 - Email address
 - Password
 - Blood Type
 - Zip Code
 - Mobile Number
 - Gender
- 3) Logout: Allow users to logout of their account
- 4) Registration validation: All fields should be valid.
- 5) Blood donation: The registration form takes the blood group of the user while registering. The entered value should be a valid blood type. It's a drop down system where the user needs to select one of the available blood types. The user should have a valid email ID and a Mobile Number.
- 6) Contacts display: Once the user enters the blood group he wants, he gets a list of all the corresponding blood types and all the contact details

UIs:

- 1) Login UI: Implement login activity UI design
- 2) Register UI: Implement register activity UI design
- 3) Search UI: Implement search activity UI design
- 4) Blood Group Selection UI: Implement Blood group activity UI design
- 5) Donor Details UI: Implement Donor details UI design

The below picture illustrates the specific ways in which blood can be donated.



Explanation:

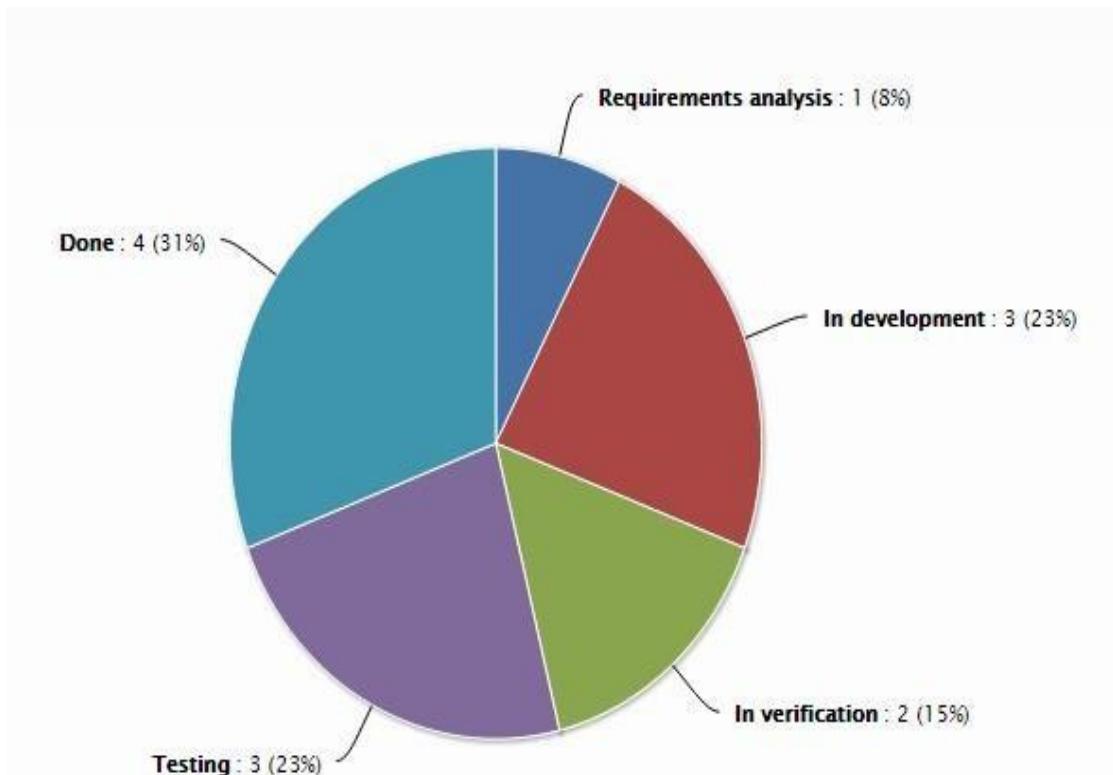
- Donors with blood type O can donate blood to Recipients with blood types O, A, B, AB. That's why people with blood group O are called Universal Donors.
- Donors with blood type A can donate blood to Recipients with blood groups A, AB.
- Donors with blood type B can donate blood to Recipients with blood groups B,
- AB donors with blood type B can donate blood to Recipients with blood groups B, AB.
- Donors with blood type AB can donate blood to Recipients with blood group AB. People with blood type AB are called the Universal Recipients. That's why they are called the Universal Recipients.

With the above observations, whenever a user wants a particular type of blood, he gets all the blood types who can donate blood to that required type not just the same blood type which the user entered. He gets to see a higher number of donor information and the probability of the correct match is higher. According to the above algorithm, the user has a bigger platform than just getting one single blood type. This improvises the typical blood donation system.

He gets to see a higher number of donor information and the probability of the correct match is higher. This improvises the typical blood donation system.

6. Analysis Graphs

- Increment Analysis graph:



- Assignment division graph

7. Wireframe

7.1 Login Page

The login page has two buttons such as Submit and Register. If the user is already having an account, they can easily login by clicking the submit button. Otherwise all the users should create an account by giving their basic information.

The below screenshot shows the wireframe diagram of the login page.



7.2. Registration Page

By clicking the registration button, the user will redirect to the registration page. The registration page has all the basic information details. By entering and clicking the submit confirmation button, an account will be created for that specific user. After that the user can go back to the login and then login with their user name and password.

The below screenshot shows the wireframe registration page.

Registration Page

7.3. Home Page

Once the login is done, the user will be redirected to home page. The home page contains the information about blood donor. To getting the particular blood group donor information, the user should have to select the blood type by clicking the drop down list and type the zip code in the corresponding textbox.

As the screenshot shows, the donor name, zip code and phone number will be displayed.

Home Page

Logout

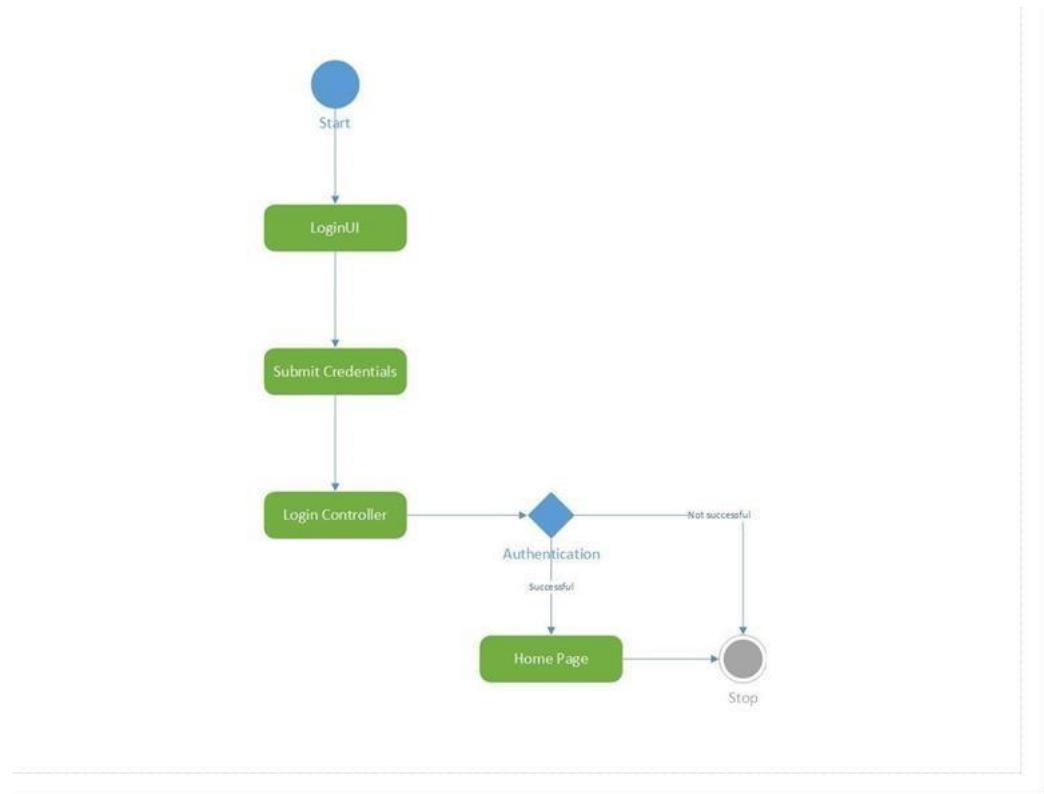
Blood Group: Zipcode: Search

User 1(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 2(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 3(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 4(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 5(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>

8.State Diagrams

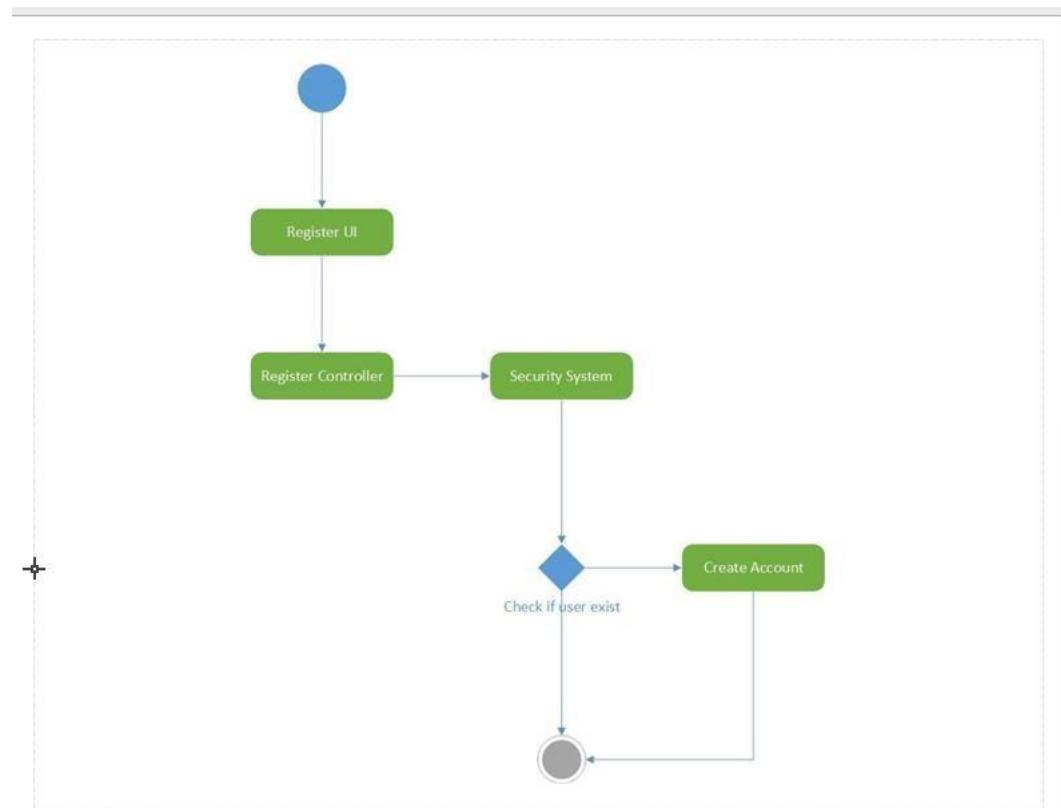
8.1. Login Page

The login page state diagram describes the login page process. If the login credentials are valid and unique then the page will be redirected to the home page by the login controller. The login process will be successful if the login controller authentication is successful otherwise the process will stop automatically.



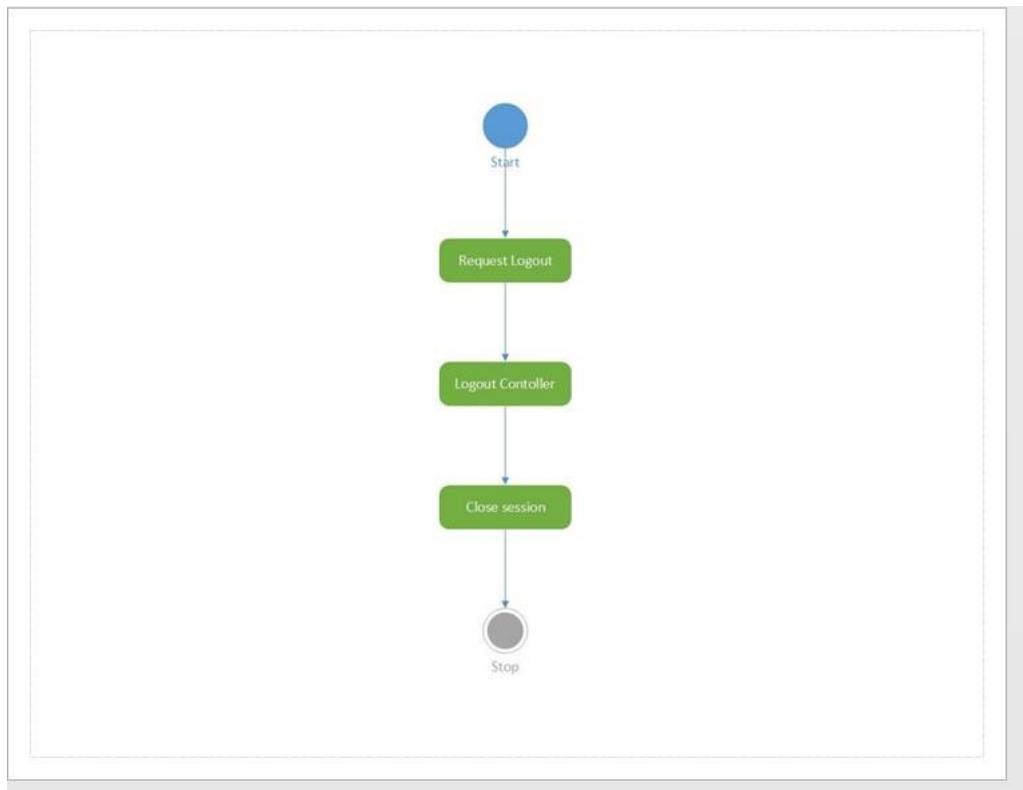
8.2. Registration Page

The state diagram of registration page verifies all the basic information about the user and then will create an account for that specific user. The registration UI will get all the basic information from the user and then pass them to the register controller. The security system will verify the user basic information. If all the values valid and unique, the new account will be created for that specific user. The below screenshot shows the State diagram of registration page.



8.3. Home Page

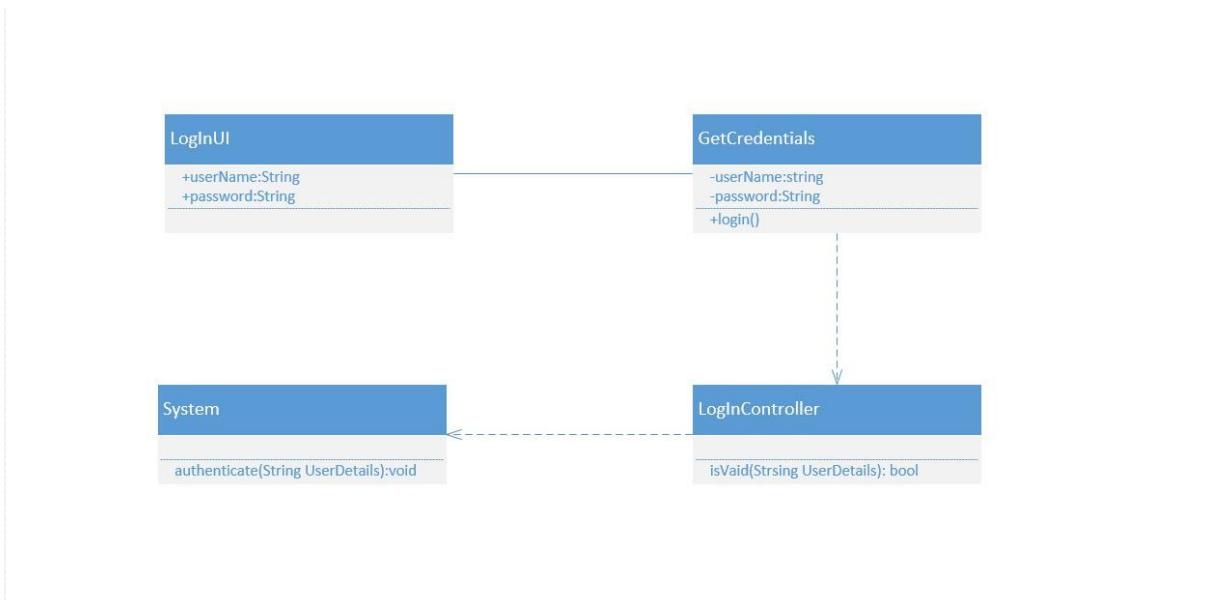
The home page displays the blood donor information. Once the user gets the information about the donor, they can logout the home page. The state diagram shows the logout steps clearly in the below screenshot. The request logout passes the request to the logout controller. The logout controller closes the session.



9. Class diagrams

9.1. Login Page

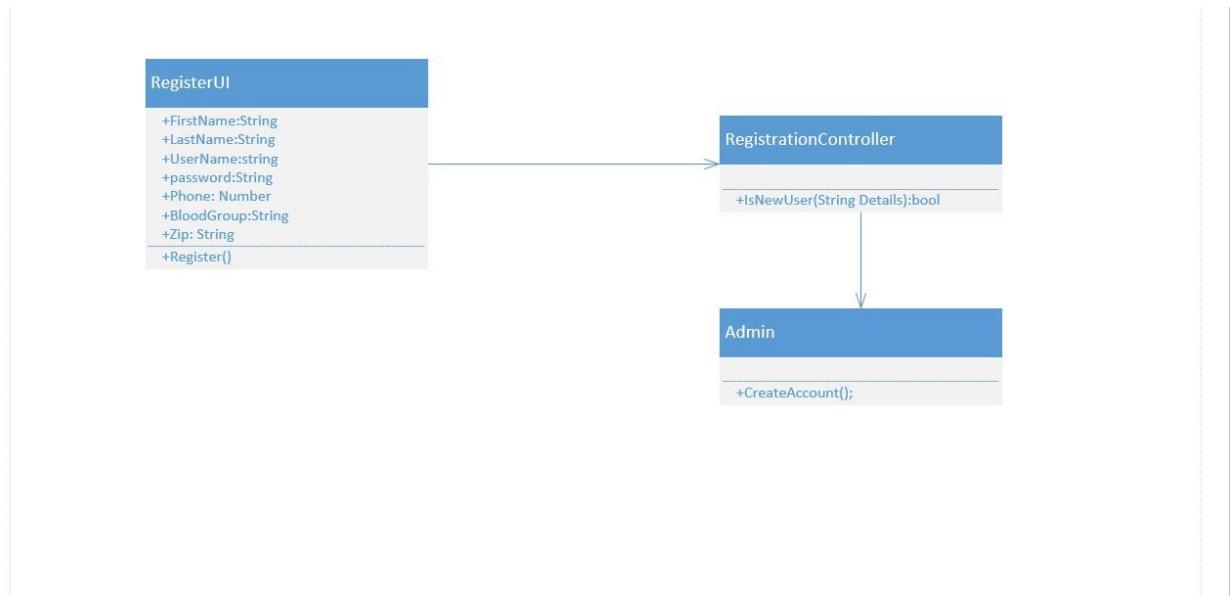
The login of the class diagram describes the flow of the system and process. The loginUI has the Username and password variables. Those variables are passing to the login controller through the GetCredentials. The system verifies the login credentials by using the authenticate function.



8.2. Registration Page

Similar to login page, the registrationUI takes all the necessary variables and pass them to the registration controller. The registration controller verifies the basic details and then the new account creates by the Admin.

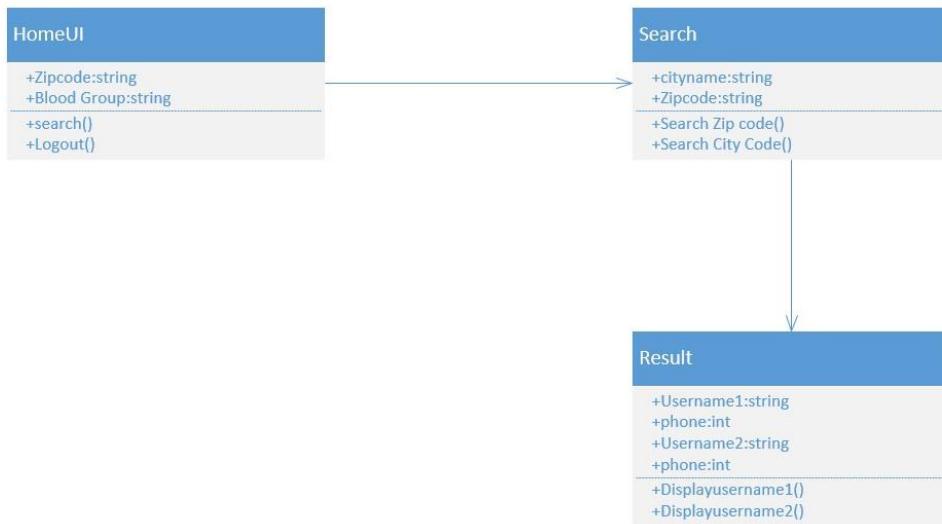
The below screenshot shows the class diagram of registration page.



8.3. Home Page

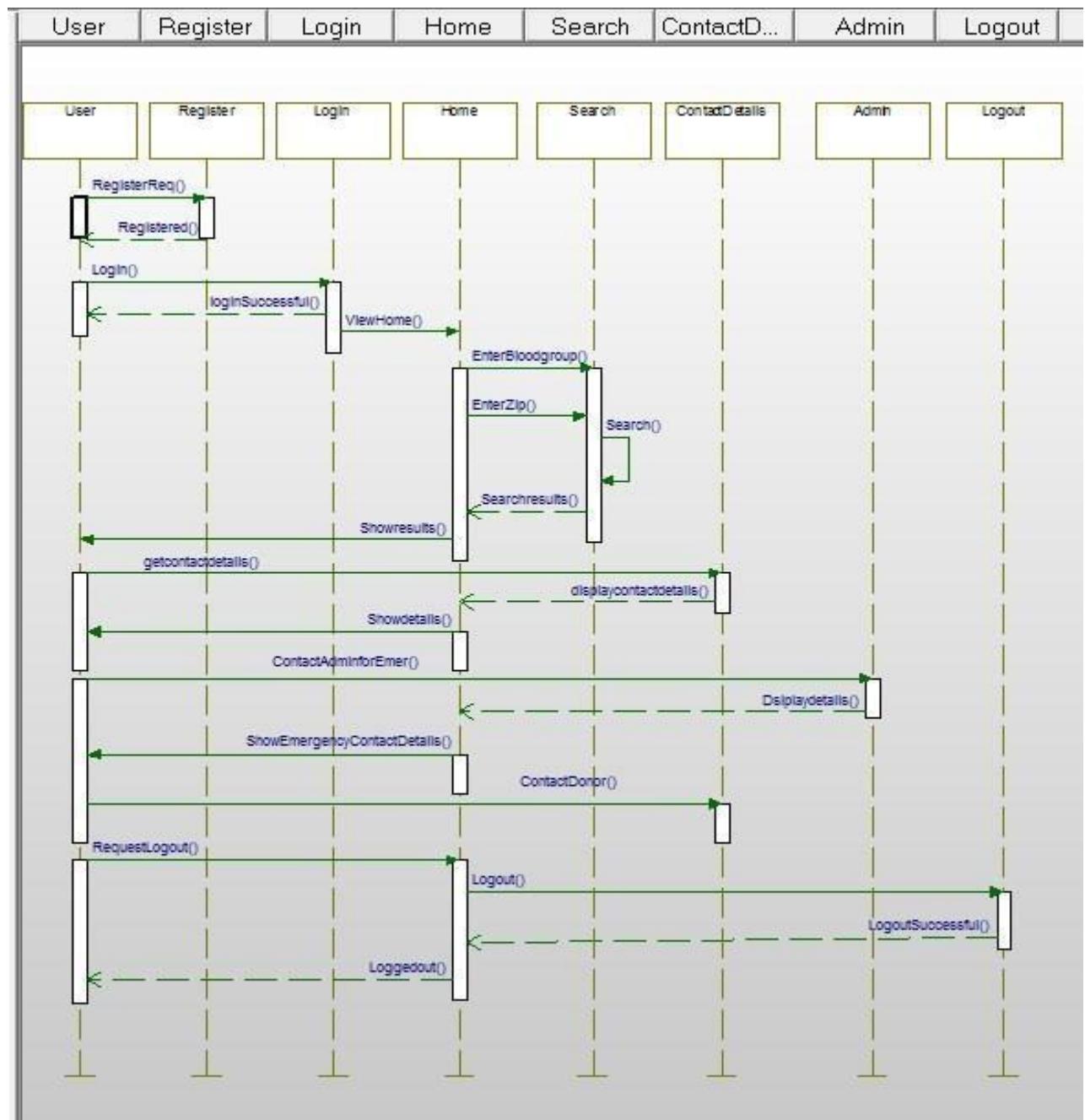
The home page donor search will take places by the search variables from the HomeUI. The search operation will happen based on the zip code and the blood type. Finally, the result will be displayed based on the blood type and zip code.

The below screenshot shows the class diagram of home page.

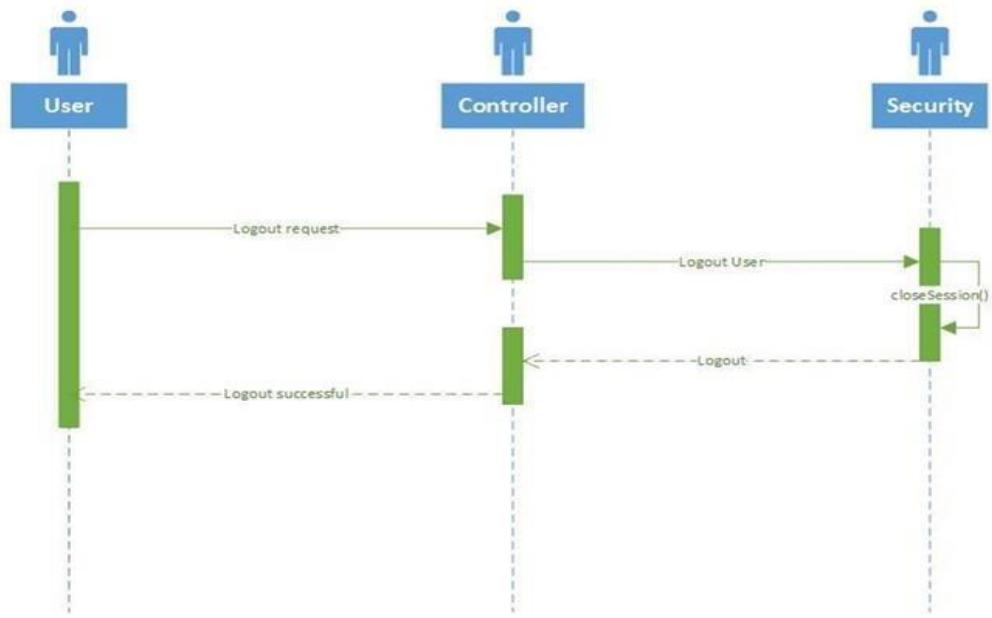


9.Sequence Diagram

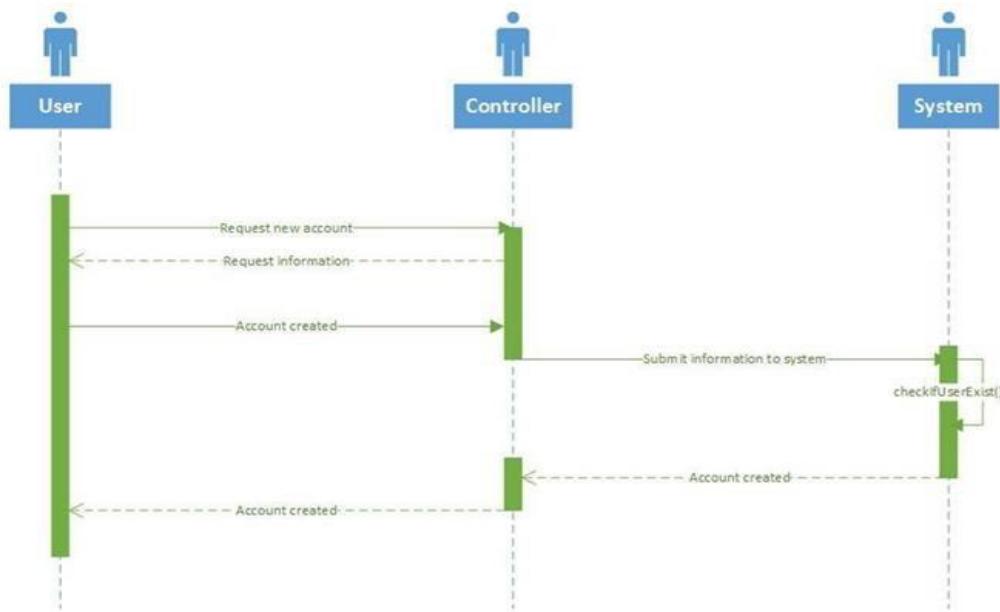
The sequence diagram describes the complete process of this project. As the screenshot shows, the registration process takes place first and the login function takes place. If the login is successful, then the view home function takes place. Sequentially, the enterbloodgroup(), enterzipcode() functions are taking place for the search. Once the search is successful, the searchresults () and showresults () functions are taking place to display the results. Finally, the logout () function takes place to logout the page.



9.1. Login Sequence Page



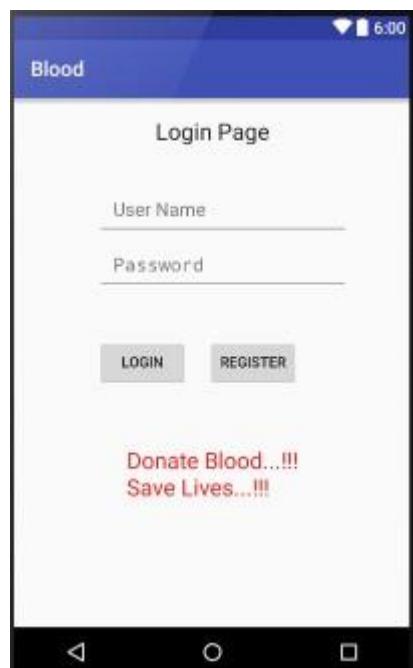
9.2. Registration Sequence Page:



10.The screenshots for the final all xml pages:

10.1. Login Page

The below screenshot shows the login page of the uml output. The user will enter their username and password, if they already have an account. Otherwise, they should create an account by clicking the registration button.



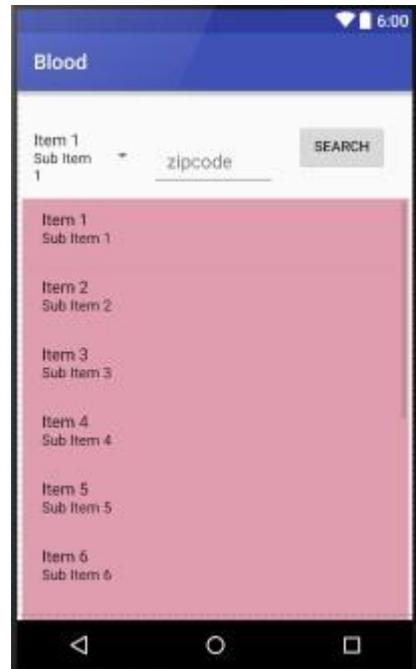
10.2.Registration page:

The Registrstion page has all the basic information tabs. After giving and entering all the basic information, the new account will be created for that specoifc user. The below screenshot shows the registration page.



10.3.Home Page:

The home page provides the search results based on the search conditions. To get the search results, the search information has been entered in the corresponding search boxes. The below screenshot shows the home page of xml output.



11. Architecture/Design

UI Design:

We have the following UIs in our project.

- **Login Page:** This is an xml layout with two text fields and two buttons. User Details will be authenticated by using the text from those text fields. If the user is new he has to register, so by clicking register button he will be redirected to registration page. If the user is successfully logged in, he will be redirected to home page.
- **Registration Page:** This is an xml layout with 6 text fields, 1 radio button and register button. User has to give name, mobile number, username, zip code and more importantly blood group name. If all the required fields are validated once user clicks register button. If user has account already, an error message will be thrown with message as “account already exists, please go to login page”. If he is a new user, all the details will be stored in our database and he will be redirected to Home Page.
- **Home page:** This is the main page where user will be redirected after successful login. In this page, user can search donors who are available in the location he provided. He can view the contact details of the donors. We will extend this functionality to google maps where all the donors are seen in the maps. User can also update profile information. There is a logout button where user can close the session.

12. Database

We are planning to use MySQL database or SQLite database to store user details. If we use social logins, we get required details from the corresponding apis.

We have only one table named “User”.

The table will consist of following columns.

- First Name : Char
- Last Name : Char
- Mobile : Number
- User Name : Char
- Password : Char
- Zip code : Char

Blood group : Char
o Created at:
DateTime o Id :
Number o Updated
at : DateTime

13. Four Different Increments:

13.1 Increment 1-- Requirement Gathering and Designing the application

1. Research on importance of blood donation management, requirements for the development.
2. Setup android studio, zenhub and Github for every team member.
3. Design class diagrams.
4. Design sequence diagrams.
5. Design state diagrams.
6. Design WireFrames.
7. Design basic UI layouts (Login, Registration, Home)

13.2 Increment 2—Coding/Testing

- 1) Implement Login/logout /Registration/ Registration validation and functionality.
- 2) Implement home page.
- 3) Setup local databases.
- 4) Test Login/logout UI/Registration/Registration validation.
- 5) Test sample user data.

13.3 Increment 3 – Coding/Testing 1)

- Implement search functionality
- 2) Implement map functionality
 - 3) Test search functionality
 - 4) Test map UI and functionality
 - 5) Extend to profile updation stage.

13.4 Increment 4 -- Refine GUI

- 1) Refine GUI for Login/logout UI/Registration/Registration validation.
- 2) Refine GUI for Home
- 3) Refine GUI for Profile
- 4) Refine GUI for Map.

14. Project Timelines, Members, *Task Responsibility*

Member and Responsibilities:

Artifacts	Members : Santhosh, Anudeep, Prasanna, FathimaJ
Research	All
Project Plan	Santhosh mohan
UML Diagrams	Prasanna, Anudeep
Modeling Database	Anudeep, Prasanna
Initial Mockup	FathimaJ
Reports	All
Development	Santhosh Mohan, Prasanna
Testing	Anudeep, FathimaJ
Maintenance	Santhosh Mohan

Project Timelines:

Increments	Tasks
Increment 1	Initial setup of environments, uml diagrams and code (Login UI, Registration UI, Home UI)
Increment 2	Code (Login, Register, Home Java code functionalities basic)
Increment 3	Code + Test (Profile functionality, map functionality, testing) and use case execution
Increment 4	Deployment to play store, final report.

15. Bibliography

- www.github.com
- www.developer.android.com
- www.developer.google.com
- <http://www.redcrossblood.org/learn-about-blood/blood-types>
- https://en.wikipedia.org/wiki/Blood_type

CS5551 ADVANCED SOFTWARE ENGINEERING

PROJECT REPORT-PHASE 2

BLOOD DONATION MANAGEMENT

TEAM MATES

- 1. PRASANNA MUPPIDI(33)**
- 2. SANTHOSH MOHAN MURARISHETTI(35)**
- 3. ANUDEEP PANDIRI(40)**
- 4. FATHIMA JAMES(58)**

1. Introduction

Our Blood Management application helps the users to find the donor for a particular blood group in a particular location in the easiest way possible. All the donors who wish to donate blood are supposed to enter their details in the application. The details include their Name, Age, Contact number, Email ID, Location (ZIP Code) and most importantly their blood group. The recipients who wish to receive blood should login to the application with their mobile number and are supposed to search the forum based on the required blood group and the location, they can contact the donor for further details.

The project has been divided into four phases with improving implementation features. For the first iteration of our project we want to complete all design section of the application with Login, Registration, Home page designs. We have chosen the

android platform to develop our application. For this first phase, we have designed the UML Class diagram, Sequence diagram, State diagrams along with wireframes. We concentrated mainly on the design part which play a major role in implementing our project.

2. Functions

2.1 Login

Login Page lets the user login to the application. User needs to enter the Username and

Password in order to login. If the user doesn't have an account to Login, he should register first.

2.2 Register

Register page lets the user to register. User needs to provide personal information to create an account.

2.3 Home

In Home Page, already registered user details like Name, Zip Code, Contact Numbers are displayed.

2.4 Search

Once the user registers, all his personal information is stored in SQLite database. User who needs blood should login into the application and search based on the blood group and Zip code and he can then contact the donor based on the contact information present in the application.

2.5 Admin

Users can anytime contact Admin for emergency situations. If any of the contact details or the Blood Group is different from what is required, Admin helps the user for more information.

3. Proposed System

1. Requirement Specification:

□ Functional Requirements:

- i. User should have a Sign in.
- ii. If user is new, he should have a Register. iii. User should be able to provide his personal information.
- iv. User details should be valid.
- v. User should be able to search based on the Blood Group. vi. User should be able to search based on the Zip Code.
- vii. User should be able to view the corresponding contact details. viii. User should be able to contact the Admin in emergency situations.
- ix. User should be able to contact the donor whose details user found on the application.
- x. User should be able to close the session. xi. User should be able to logout.

4. Technologies Used

- i. ADT: Android Studio ii. Programming language: JAVA iii. Database: SQLite
- iv. Frontend: XML, JavaScript

5. Development

System Designs:

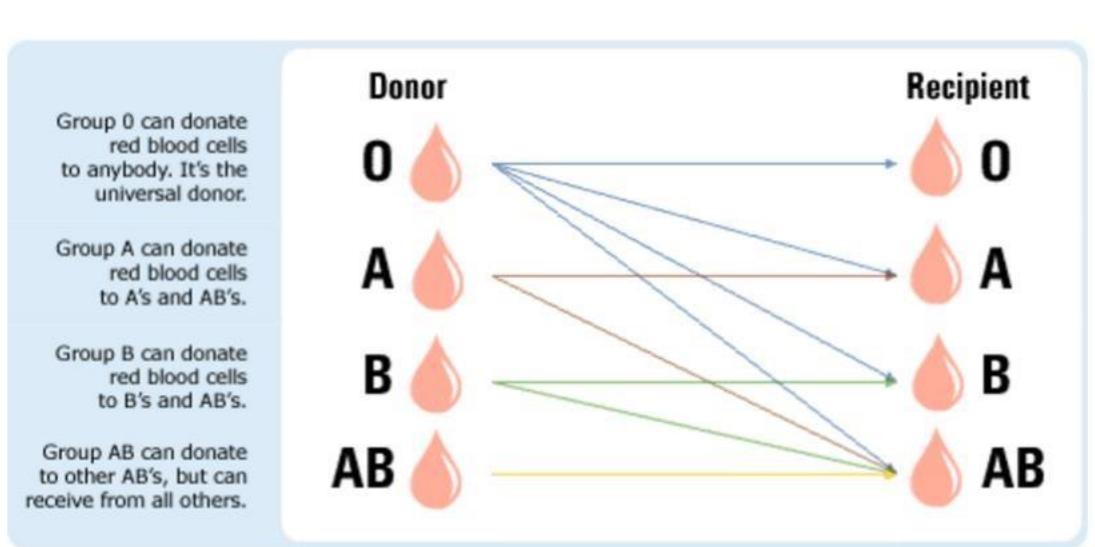
- 1) Login: Allow users to login into their account. Users use email address and Password to identify themselves.
- 2) Registration: Allow new users to create an account. User information collected during this process:
 - First name
 - Last name
 - Date of birth
 - Email address
 - Password
 - Blood Type
 - Address
 - Mobile Number
 - Gender
- 3) Logout: Allow users to logout of their account
- 4) Registration validation: All fields should be valid.
- 5) Blood donation: The registration form takes the blood group of the user while registering. The entered value should be a valid blood type. It's a drop down system where the user needs to select one of the available blood types. The user should have a valid email ID and a Mobile Number.
- 6) Contacts display: Once the user enters the blood group he wants, he gets a list of all the corresponding blood types and all the contact details

UIs:

- 1) Login UI: Implement login activity UI design
- 2) Register UI: Implement register activity UI design
- 3) Search UI: Implement search activity UI design
- 4) Blood Group Selection UI: Implement Blood group activity UI design

4) Donor Details UI: Implement Donor details UI design

The below picture illustrates the specific ways in which blood can be donated.



Explanation:

- Donors with blood type O can donate blood to Recipients with blood groups O, A, B, AB. That's why people with blood group O are called Universal Donors.
- Donors with blood type A can donate blood to Recipients with blood groups A, AB.
- Donors with blood type B can donate blood to Recipients with blood groups B, AB.
- AB donors with blood type B can donate blood to Recipients with blood groups B, AB.
- Donors with blood type AB can donate blood to Recipients with blood group AB. People with AB blood type can receive blood from all the other blood types. That's why they are called the Universal Recipients.

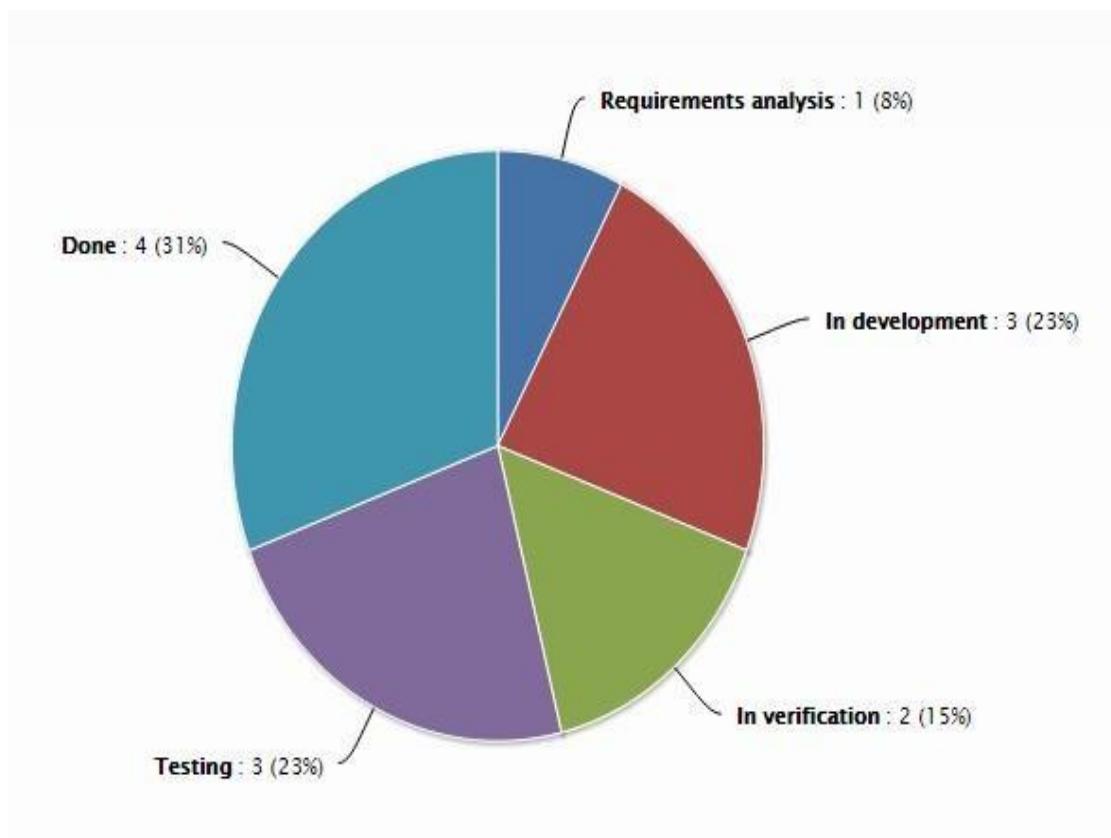
With the above observations, whenever a user wants a particular type of blood, he gets all the blood types who can donate blood to that required type not just the same blood type which the user entered. The user has a bigger platform than just getting one single blood type. He gets to see a higher number of donor information and the probability of the correct match is higher. This improvises the typical blood donation system.

According to the above algorithm, the user has a bigger platform than just getting one single blood type.

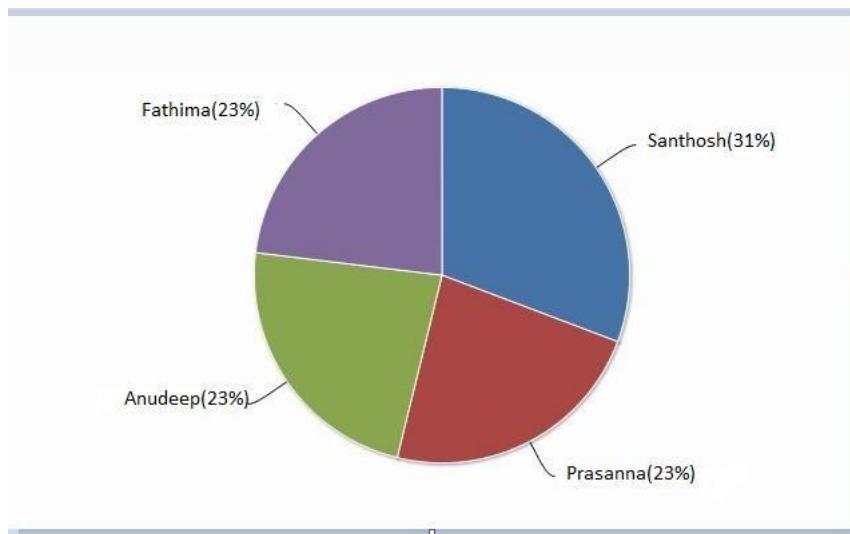
He gets to see a higher number of donor information and the probability of the correct match is higher. This improvises the typical blood donation system.

6. Analysis Graphs

- Increment Analysis graph:



- Assignment division graph



The above diagram describes about the tasks done by each group member.

7. Wireframe

7.1 Login Page

The login page has two buttons such as Submit and Register. If the user is already having an account, they can easily login by clicking the submit button. Otherwise all the users should create an account by giving their basic information.

The below screenshot shows the wireframe diagram of the login page.



7.2. Registration Page

By clicking the registration button, the user will redirect to the registration page. The registration page has all the basic information details. By entering and clicking the submit confirmation button, an account will be created for that specific user. After that the user can go back to the login and then login with their user name and password.

The below screenshot shows the wireframe registration page.

Registration Page

7.3. Home Page

Once the login is done, the user will be redirected to home page. The home page contains the information about blood donor. To getting the particular blood group donor information, the user should have to select the blood type by clicking the drop down list and type the zip code in the corresponding textbox.

As the screenshot shows, the donor name, zip code and phone number will be displayed.

Home Page

Logout

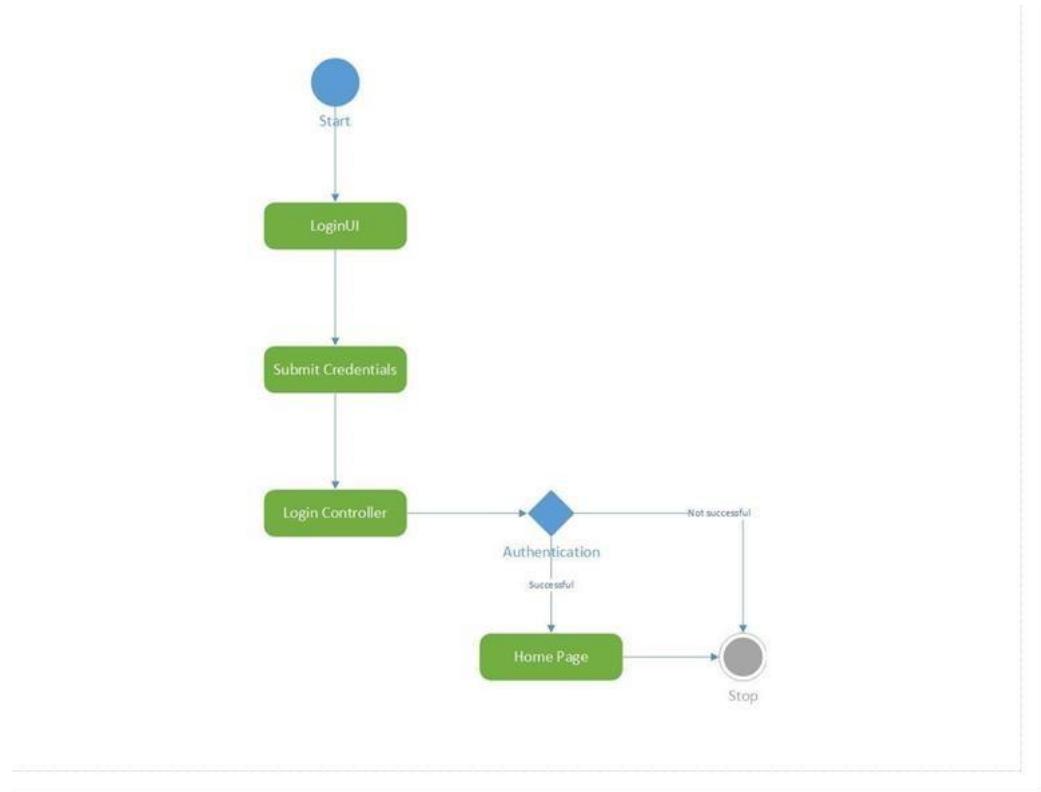
Blood Group: Zipcode: Search

User 1(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 2(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 3(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 4(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>
User 5(name): <input type="text"/>	City: <input type="text"/>	Number: <input type="text"/>

8.State Diagrams

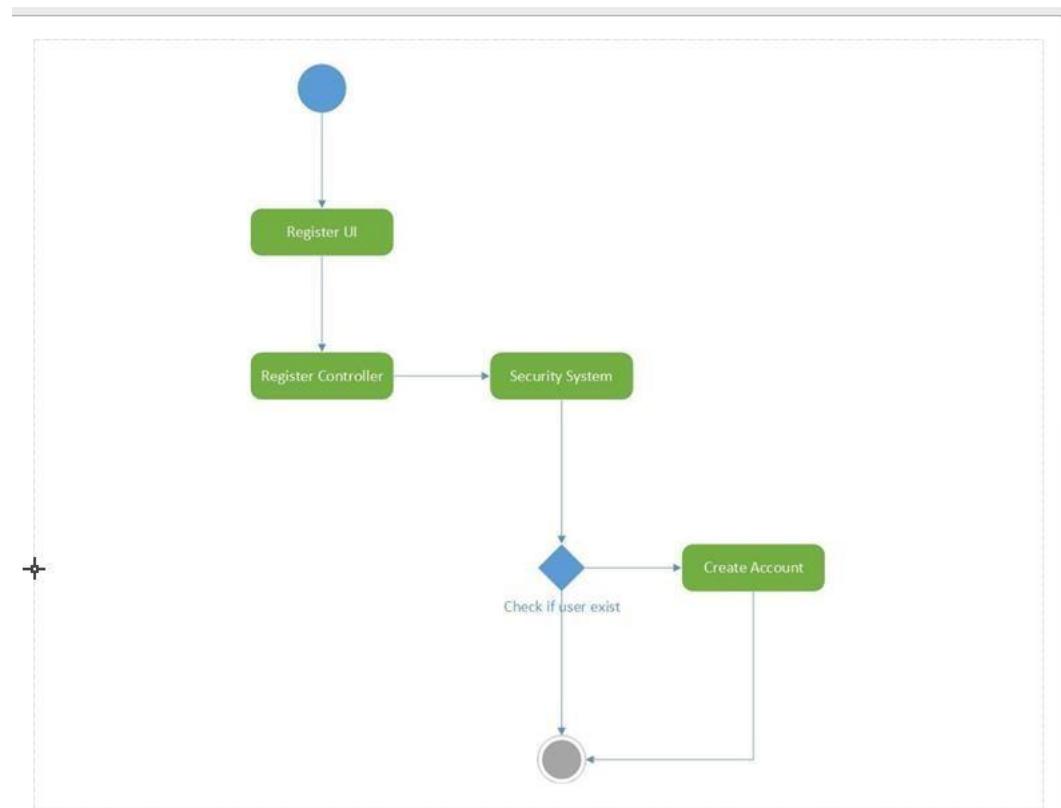
8.1. Login Page

The login page state diagram describes the login page process. If the login credentials are valid and unique then the page will be redirected to the home page by the login controller. The login process will be successful if the login controller authentication is successful otherwise the process will stop automatically.



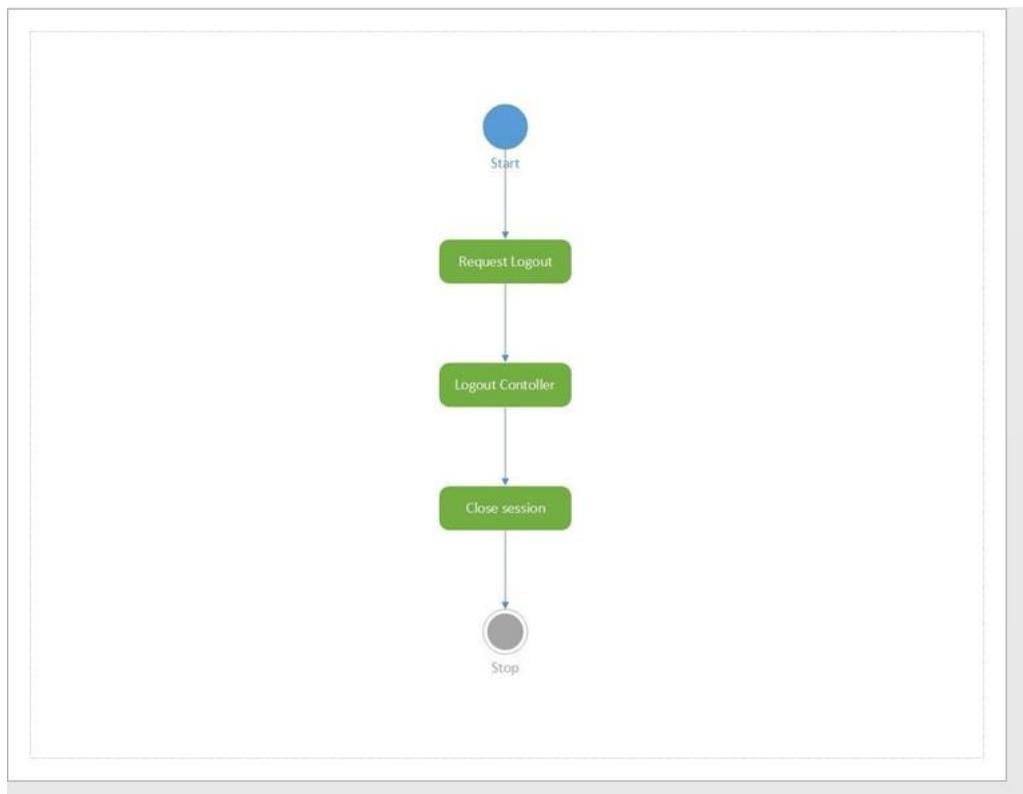
8.2. Registration Page

The state diagram of registration page verifies all the basic information about the user and then will create an account for that specific user. The registration UI will get all the basic information from the user and then pass them to the register controller. The security system will verify the user basic information. If all the values valid and unique, the new account will be created for that specific user. The below screenshot shows the State diagram of registration page.



8.3. Home Page

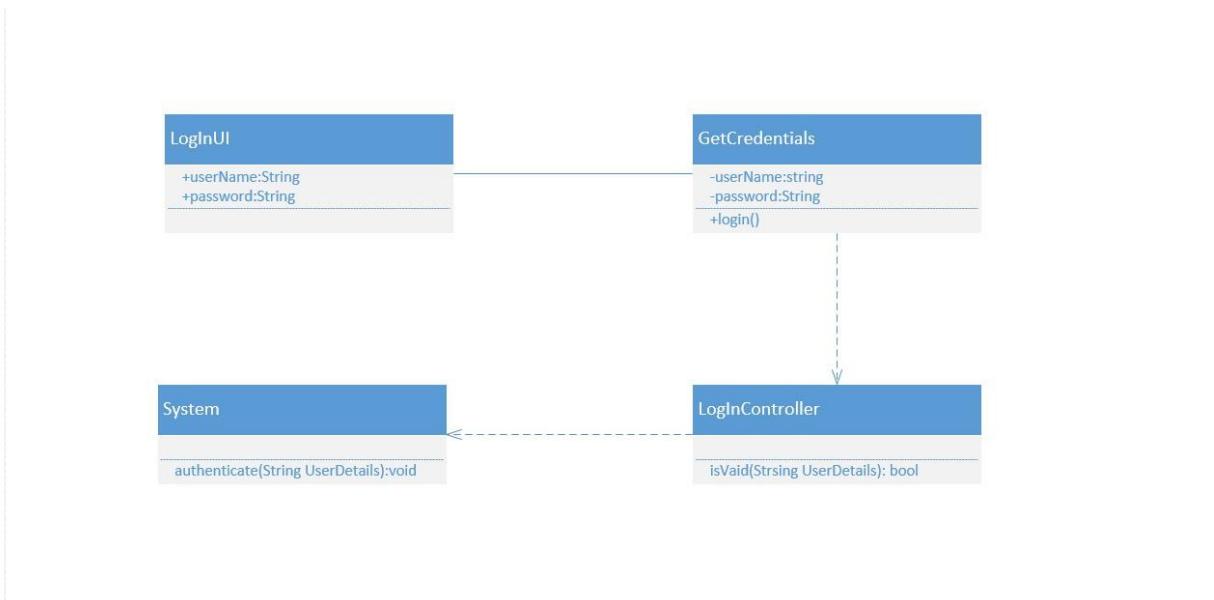
The home page displays the blood donor information. Once the user gets the information about the donor, they can logout the home page. The state diagram shows the logout steps clearly in the below screenshot. The request logout passes the request to the logout controller. The logout controller closes the session.



9.Class diagrams

9.1. Login Page

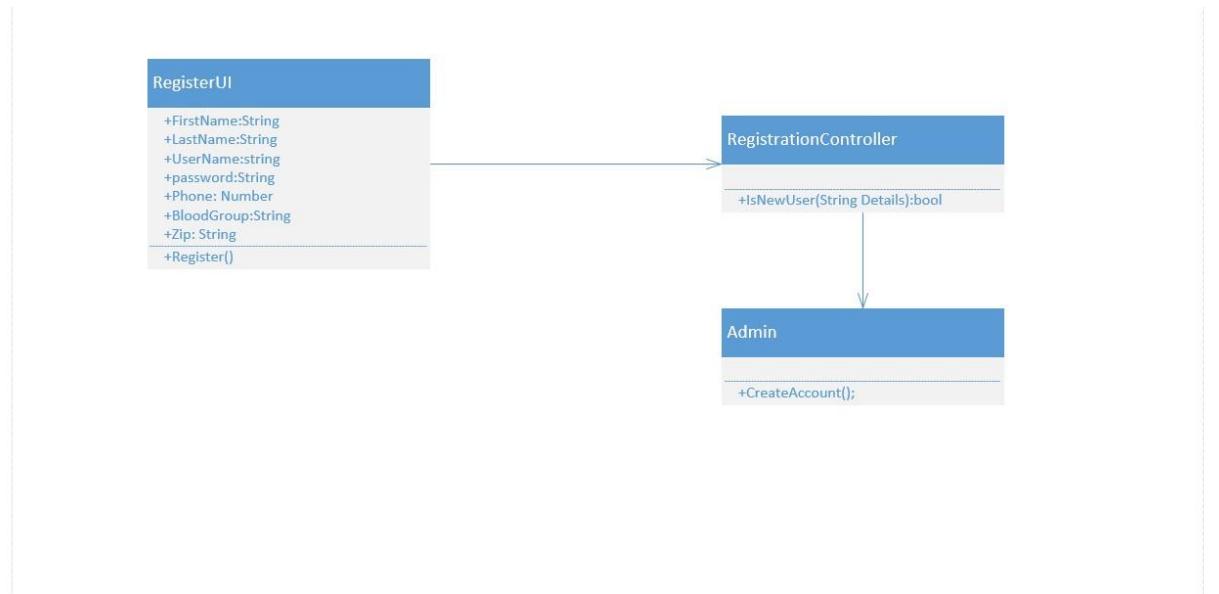
The login of the class diagram describes the flow of the system and process. The loginUI has the Username and password variables. Those variables are passing to the login controller through the GetCredentials. The system verifies the login credentials by using the authenticate function.



8.2. Registration Page

Similar to login page, the registrationUI takes all the necessary variables and pass them to the registration controller. The registration controller verifies the basic details and then the new account creates by the Admin.

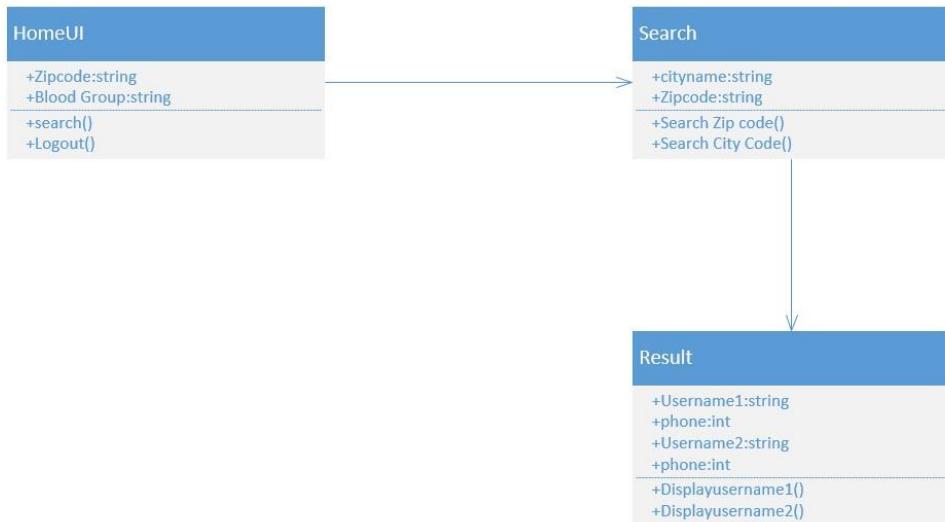
The below screenshot shows the class diagram of registration page.



8.3. Home Page

The home page donor search will take places by the search variables from the HomeUI. The search operation will happen based on the zip code and the blood type. Finally, the result will be displayed based on the blood type and zip code.

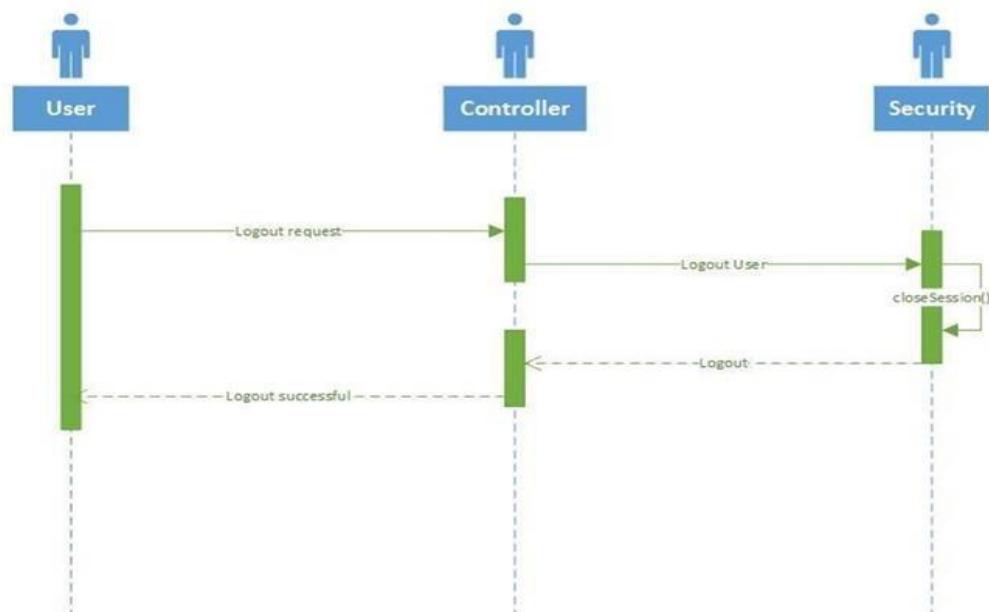
The below screenshot shows the class diagram of home page.



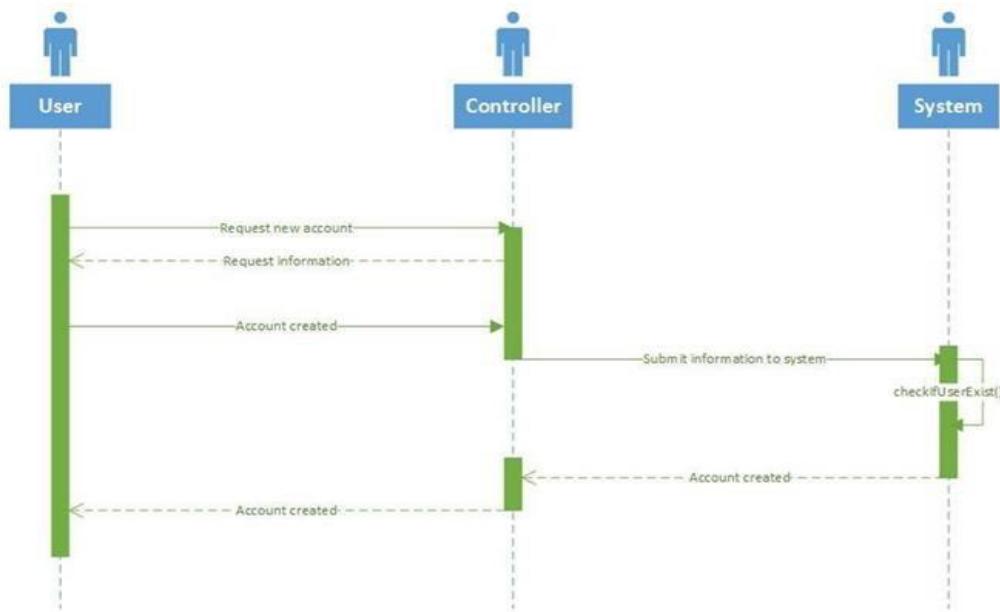
9.Sequence Diagram

The sequence diagram describes the complete process of this project. As the screenshot shows, the registration process takes place first and the login function takes place. If the login is successful, then the view home function takes place. Sequentially, the enterbloodgroup(), enterzipcode() functions are taking place for the search. Once the search is successful, the searchresults () and showresults () functions are taking place to display the results. Finally, the logout () function takes place to logout the page.

9.1. Login Sequence Page



9.2. Registration Sequence Page:



10. Layouts:

10.1. Welcome Page

The below screenshot shows the Welcome page of the uml output. The user can **Login**, if they already have an account. Otherwise, they should create an account by clicking the **register** button.

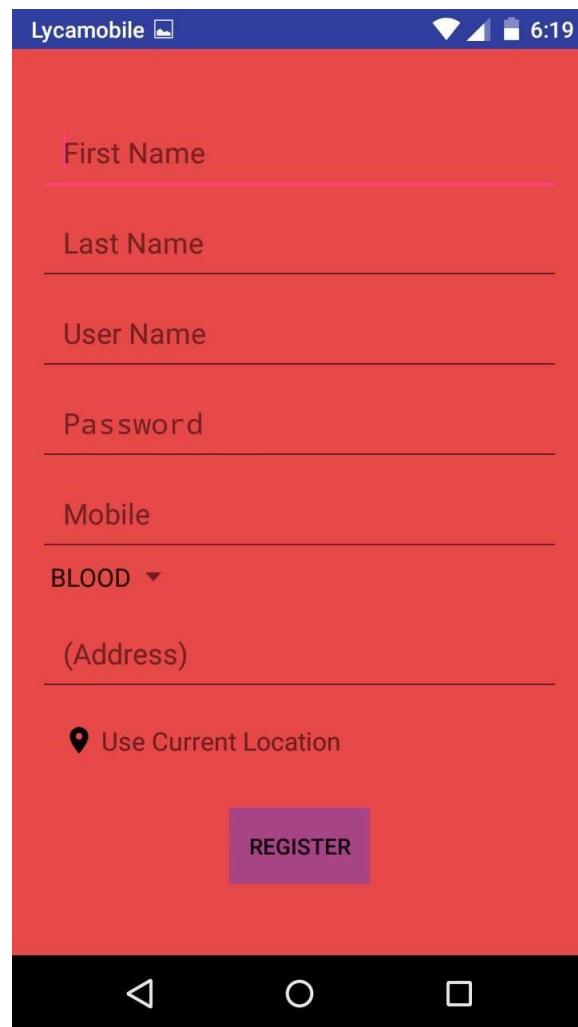


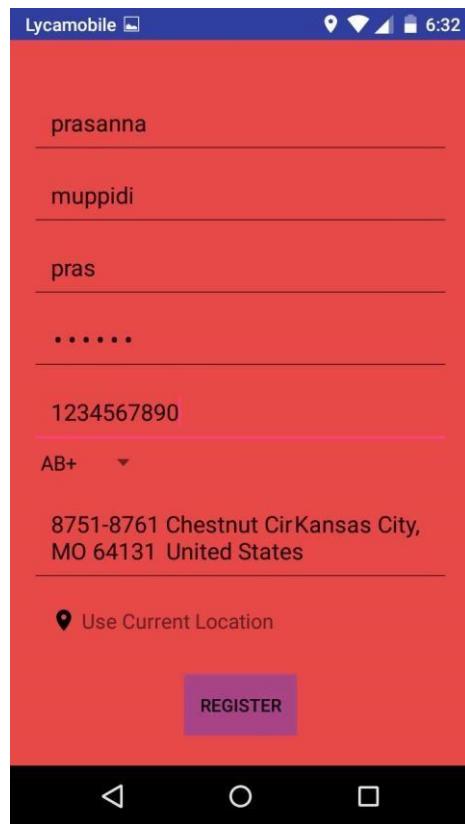
In future, we are planning to increment Social Login such as Facebook or Gmail in the center of the layout.

Note: If the user is already logged in, application doesn't ask him to login again and Welcome Page is not displayed. Session expires when the user clicks the logout button and will be redirected to the Login page.

10.2.Registration page:

The Registration page has all the basic information tabs. After giving and entering all the basic information, new account will be created for that specific user. The below screenshot shows the registration page.





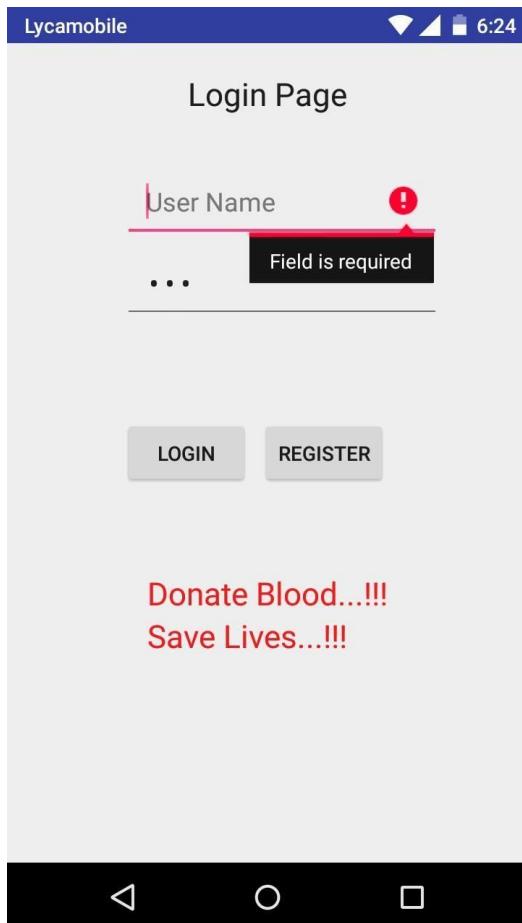
Address can be entered manually or by clicking the Use current location marker.

Validations have been added for First name, Last name and Mobile Number.

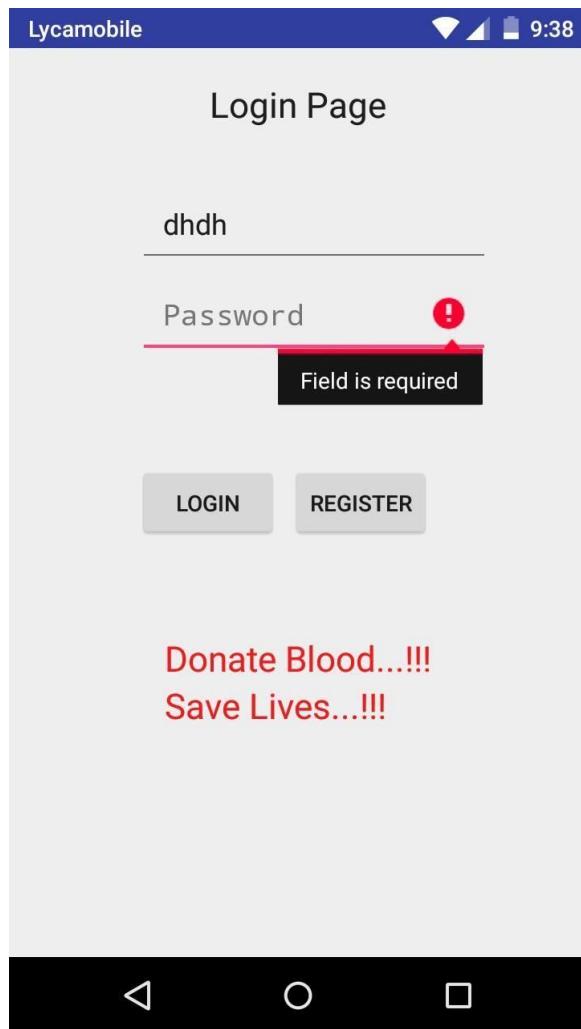
First name and Last name can't have numbers and Mobile Number can't have characters and it takes exactly 10 numbers.

10.3. Login Page:

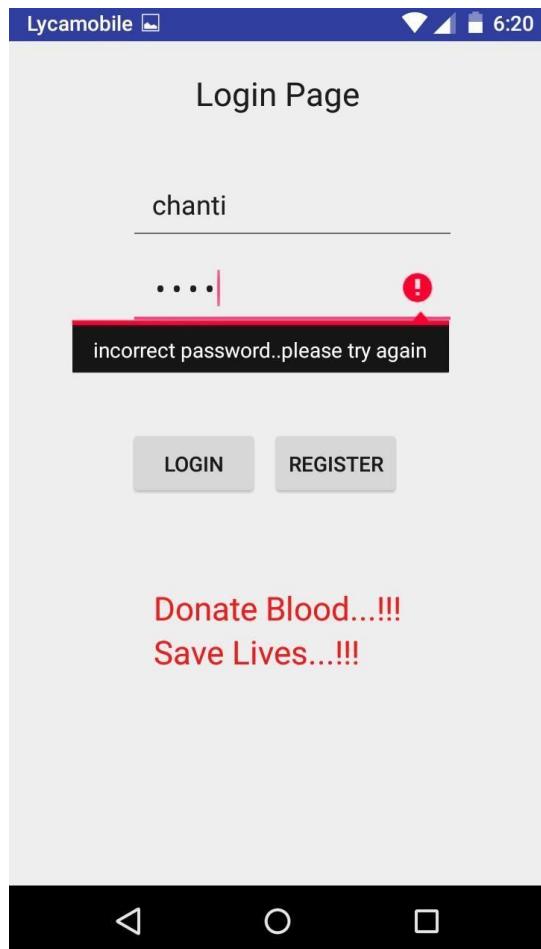
- If the Username field is empty, an error message pops up telling the Username field is required.



- If the password field is empty, an error message pops up telling the Password field is required.



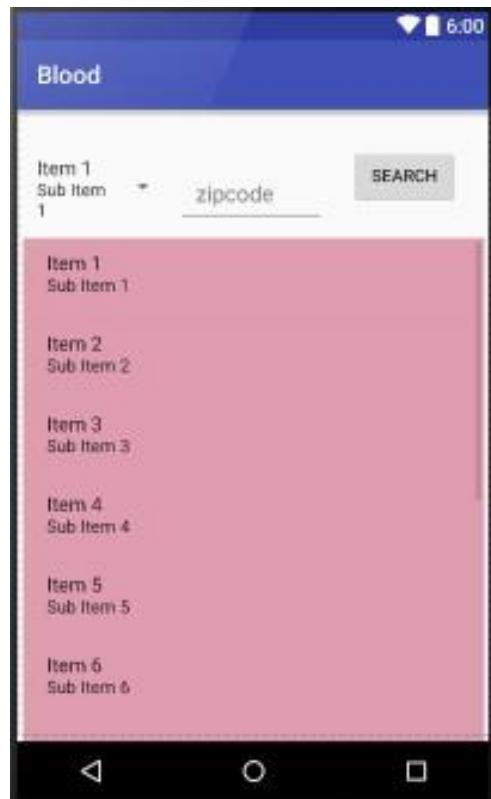
- If the user credentials are not in the database, an error message pops up telling the credentials are invalid.



- If the user credentials are valid, user is redirected to the home page.

10.4.Home Page:

The home page provides the search results based on the search conditions. To get the search results, the search information has been entered in the corresponding search boxes. The below screenshot shows the home page of xml output.



11. Architecture/Design

UI Design:

We have the following UIs in our project.

- **Login Page:** This is an xml layout with two text fields and two buttons. User Details will be authenticated by using the text from those text fields. If the user is new he has to register, so by clicking register button he will be redirected to registration page. If the user is successfully logged in, he will be redirected to home page.
- **Registration Page:** This is an xml layout with 6 text fields, 1 radio button and register button. User has to give name, mobile number, username, zip code and more importantly blood group name. If all the required fields are validated once user clicks register button. If user has account already, an error message will be thrown with message as “account already exists, please go to login page”. If he is a new user, all the details will be stored in our database and he will be redirected to Home Page.
- **Home page:** This is the main page where user will be redirected after successful login. In this page, user can search donors who are available in the location he provided. He can view the contact details of the donors. We will extend this functionality to google maps where all the donors are seen in the maps. User can also update profile information. There is a logout button where user can close the session.

12. Database

We are planning to use MySQL database or SQLite database to store user details. If we use social logins, we get required details from the corresponding APIs.

We have only one table named “User”.

The table consists of following columns.

- First Name : Char
- Last Name : Char
- Mobile : Number
- User Name : Char
- Password : Char ○

- Address : Char ○
- Blood group : Char
- Created at:
- DateTime ○
- Updated at :
- DateTime

The below is the screenshot for the Firebase database.
It's a NoSQL Database where data is stored as JSON objects.

The screenshot shows the Firebase Realtime Database dashboard with the following structure:

```

bloodmanagement
  |
  +-- Users
      |
      +-- chanti
          |
          +-- chanti1913
              |
              +-- address: "8763-8775 Chestnut Cir\nKansas City, MO 64131\Un..."
              |
              +-- blood_group: "A+"
              |
              +-- first_name: "chanti"
              |
              +-- last_name: "mohan"
              |
              +-- mobile: "949"
              |
              +-- password: "msm"
              |
              +-- user_name: "chanti1913" ✎
      |
      +-- pras
      |
      +-- raviteja83
      |
      +-- surya
  
```

The legend indicates the following changes:

- Changed (Yellow)
- Added (Green)
- Deleted (Red)
- Moved (Blue)

13. *Four Different Increments:*

13.1 Increment 1-- Requirement Gathering and Designing the application

1. Research on importance of blood donation management, requirements for the development.
2. Setup android studio, zenhub and Github for every team member.
3. Design class diagrams.
4. Design sequence diagrams.
5. Design state diagrams.
6. Design WireFrames.
7. Design basic UI layouts (Login, Registration, Home)

13.2 Increment 2—Coding/Testing

- 1) Implement Login/logout /Registration/ Registration validation and functionality.
- 2) Implement home page.
- 3) Setup local databases.
- 4) Test Login/logout UI/Registration/Registration validation.
- 5) Test sample user data.

13.3 Increment 3 – Coding/Testing 1)

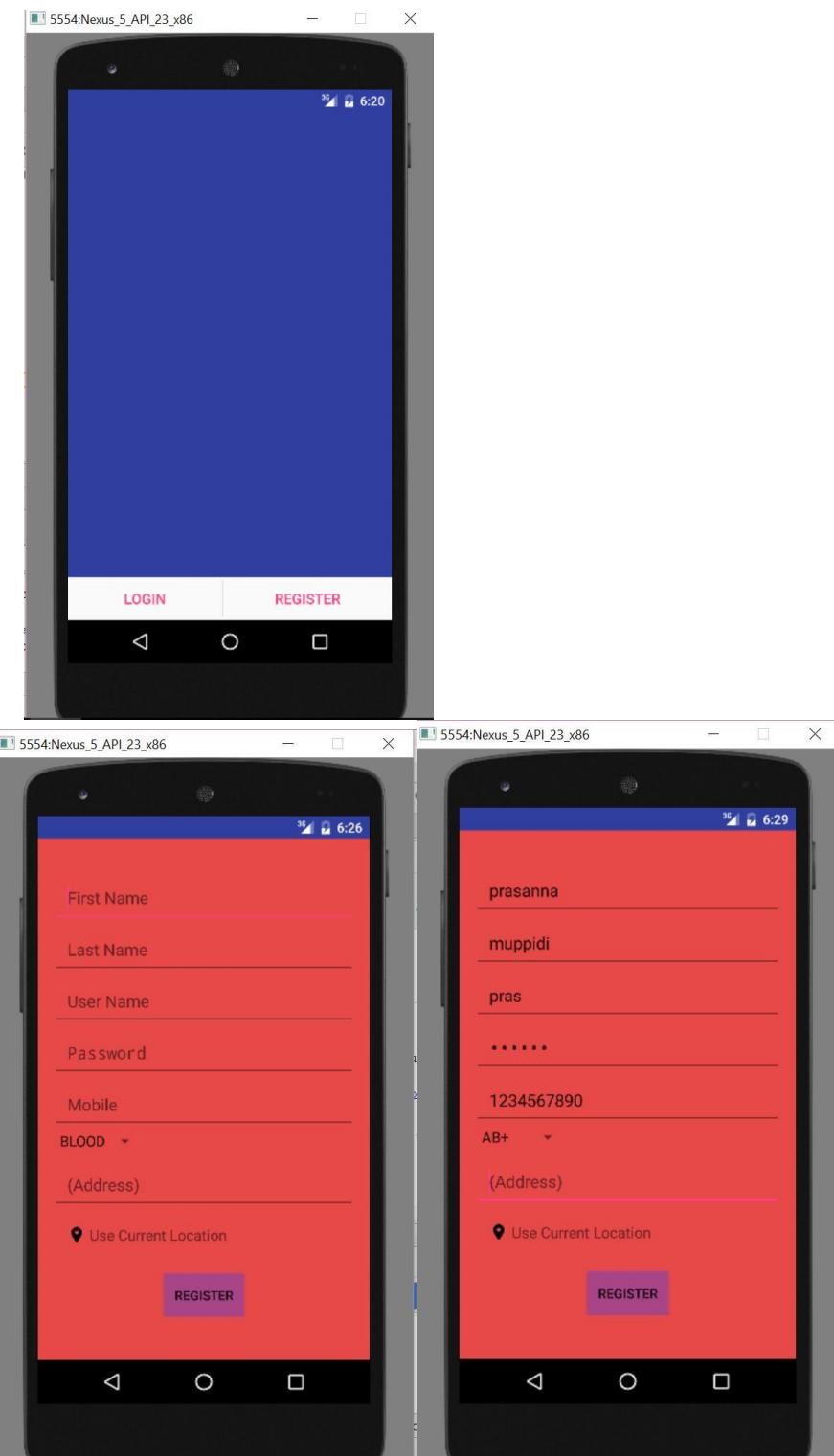
- Implement search functionality
- 2) Implement map functionality
 - 3) Test search functionality 4) Test map UI and functionality
 - 5) Extend to profile updation stage.

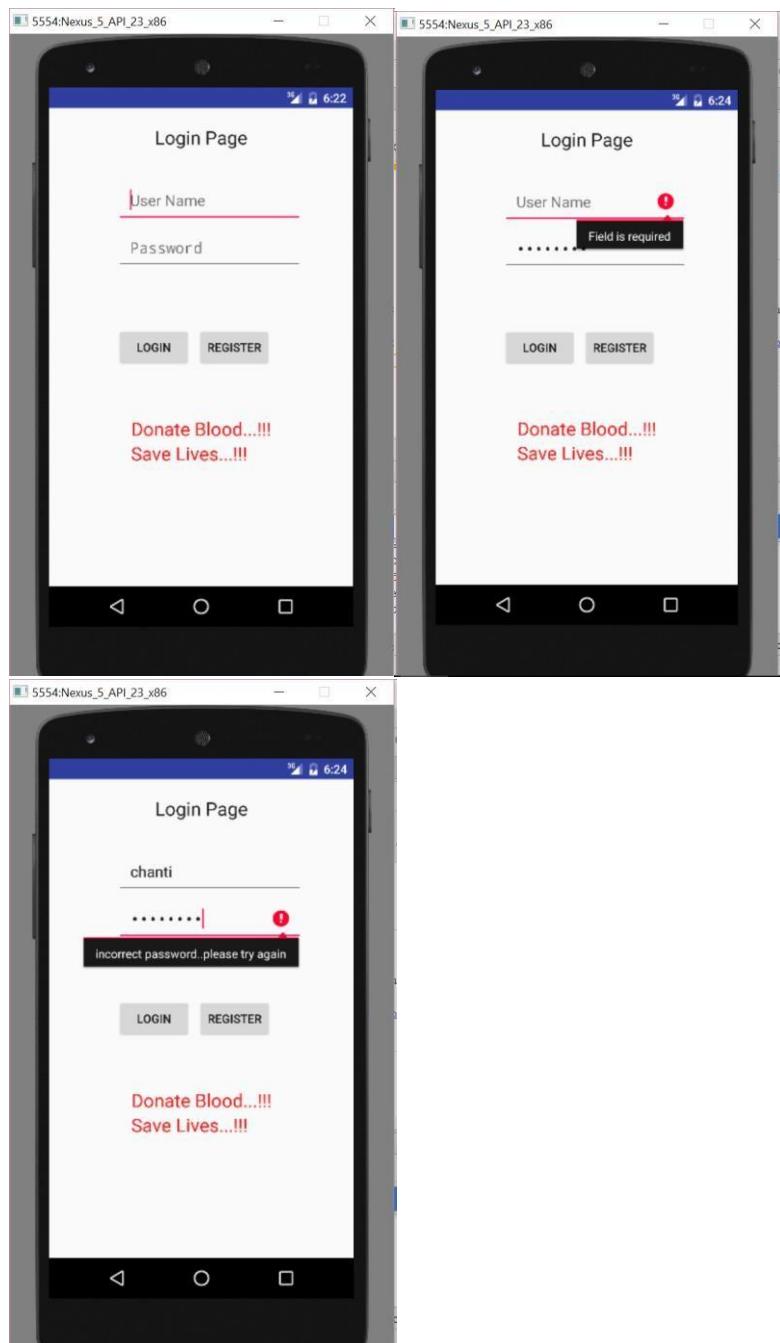
13.4 Increment 4 -- Refine GUI

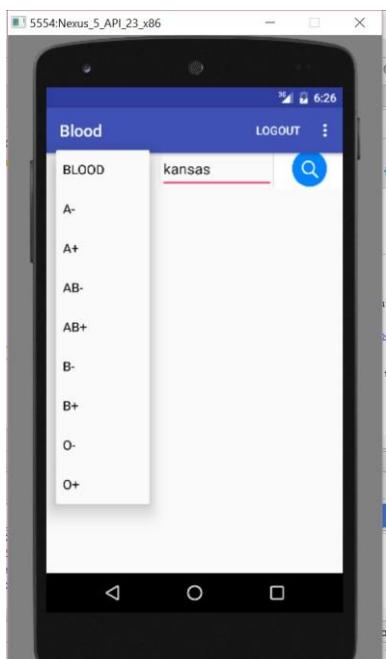
- 1) Refine GUI for Login/logout UI/Registration/Registration validation.
- 2) Refine GUI for Home
- 3) Refine GUI for Profile 4) Refine GUI for Map.

14. TESTING:

14.1 Emulator testing:







15. Project Timelines, Members, Task Responsibility

Member and Responsibilities:

Artifacts	Members : Santhosh, Anudeep, Prasanna, FathimaJ
Research	All
Project Plan	Santhosh mohan
UML Diagrams	Prasanna, Anudeep
Modeling Database	Anudeep, Prasanna
Initial Mockup	FathimaJ
Reports	All
Development	Santhosh Mohan, Prasanna, Anudeep
Testing	Anudeep, FathimaJ
Maintenance	Santhosh Mohan

Project Timelines:

Increments	Tasks
Increment 1	Initial setup of environments, uml diagrams and code (Login UI, Registration UI, Home UI)
Increment 2	Code (Login, Register, Home Java code functionalities basic)
Increment 3	Code + Test (Profile functionality, map functionality, testing) and use case execution
Increment 4	Deployment to play store, final report.

16. Bibliography

- www.github.com
- www.developer.android.com
- www.developer.google.com
- <http://www.redcrossblood.org/learn-about-blood/blood-types>
- https://en.wikipedia.org/wiki/Blood_type
- <https://www.firebaseio.com/>
- <https://www.firebaseio.com/docs/android/guide/>
- <https://www.firebaseio.com/blog/>

CS551 ADVANCED SOFTWARE ENGINEERING

PROJECT REPORT-PHASE 3

BLOOD DONATION MANAGEMENT

TEAM MATES

- 1. PRASANNA MUPPIDI(33)**
- 2. SANTHOSH MOHAN MURARISHETTI(35)**
- 3. ANUDEEP PANDIRI(40)**
- 4. FATHIMA JAMES(58)**

1.Introduction

Our Blood Management application helps the users to find the donor for a particular blood group in a particular location in the easiest way possible. All the donors who wish to donate blood are supposed to enter their details in the application. The details include their Name, Age, Contact number, Email ID, Location (ZIP Code) and most importantly their blood group. The recipients who wish to receive blood should login to the application with their mobile number and are supposed to search the forum based on the required blood group and the location, they can contact the donor for further details.

The project has been divided into four phases with improving implementation features. For the first iteration of our project we want to complete all design section of the application with Login, Registration, Home page designs. We have chosen the android platform to develop our application. For this first phase, we have designed the UML Class diagram, Sequence diagram, State diagrams along with wireframes. We concentrated mainly on the design part which play a major role in implementing our project.

2.Functions:

2.1 Login

Login Page lets the user login to the application. User needs to enter the Username and Password in order to login. If the user doesn't have an account to Login, he should register first.

2.2 Register

Register page lets the user to register. User needs to provide personal information to create an account.

2.3 Home

In Home Page, already registered user details like Name, Zip Code, Contact Numbers are displayed.

2.4 Search

Once the user registers, all his personal information is stored in SQLite database. User who needs blood should login into the application and search based on the blood group and Zip code and he can then contact the donor based on the contact information present in the application.

2.5 Admin

Users can anytime contact Admin for emergency situations. If any of the contact details or the Blood Group is different from what is required, Admin helps the user for more information.

3. Proposed System

1. Requirement Specification:

- Functional Requirements:
 - 1. User should have a Sign in.
 - 2. If user is new, he should have a Register. iii. User should be able to provide his personal information.
 - iv. User details should be valid.
 - v. User should be able to search based on the Blood Group. vi. User should be able to search based on the Zip Code. vii. User should be able to view the corresponding contact details. viii. User should be able to contact the Admin in emergency situations.
 - ix. User should be able to contact the donor whose details user found on the application.
 - x. User should be able to close the session. xi. User should be able to logout.

4. Technologies Used:

- i. ADT: Android Studio
- ii. Programming language: JAVA
- iii. Database: SQLite
- iv. Frontend: XML, JavaScript

5.Development

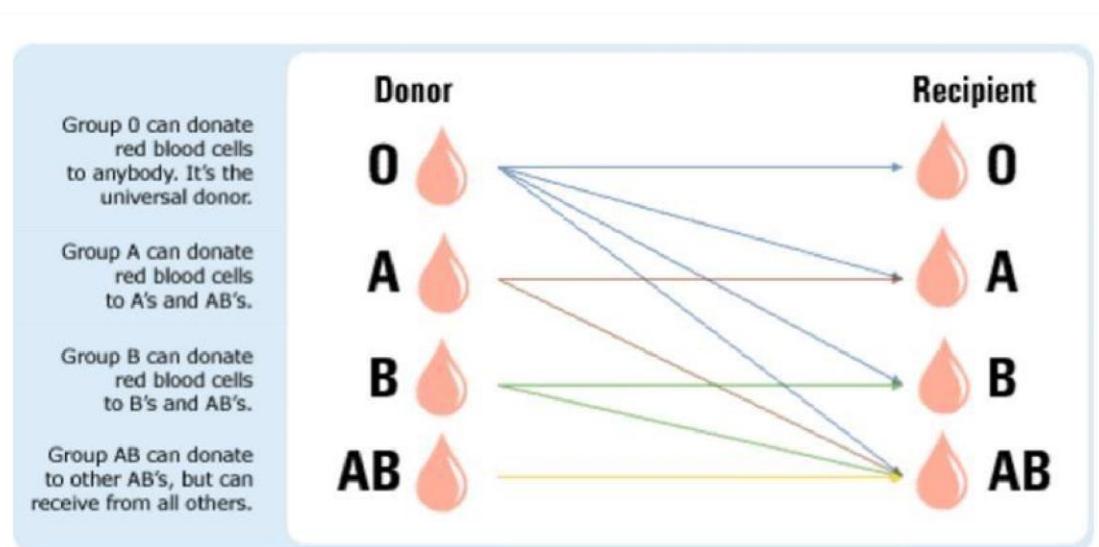
System Designs:

- 1) Login: Allow users to login into their account. Users use email address and Password to identify themselves.
- 2) Registration: Allow new users to create an account. User information collected during this process:
 - First name
 - Last name
 - Date of birth
 - Email address
 - Password
 - Blood Type
 - Address
 - Mobile Number
 - Gender
- 3) Logout: Allow users to logout of their account
- 4) Registration validation: All fields should be valid.
- 5) Blood donation: The registration form takes the blood group of the user while registering. The entered value should be a valid blood type. It's a drop down system where the user needs to select one of the available blood types. The user should have a valid email ID and a Mobile Number.
- 6) Contacts display: Once the user enters the blood group he wants, he gets a list of all the corresponding blood types and all the contact details

UIs:

1. Login UI: Implement login activity UI design
2. Register UI: Implement register activity UI design
- 7) Search UI: Implement search activity UI design
- 3) Blood Group Selection UI: Implement Blood group activity UI design
- 4) Donor Details UI: Implement Donor details UI design

The below picture illustrates the specific ways in which blood can be donated.



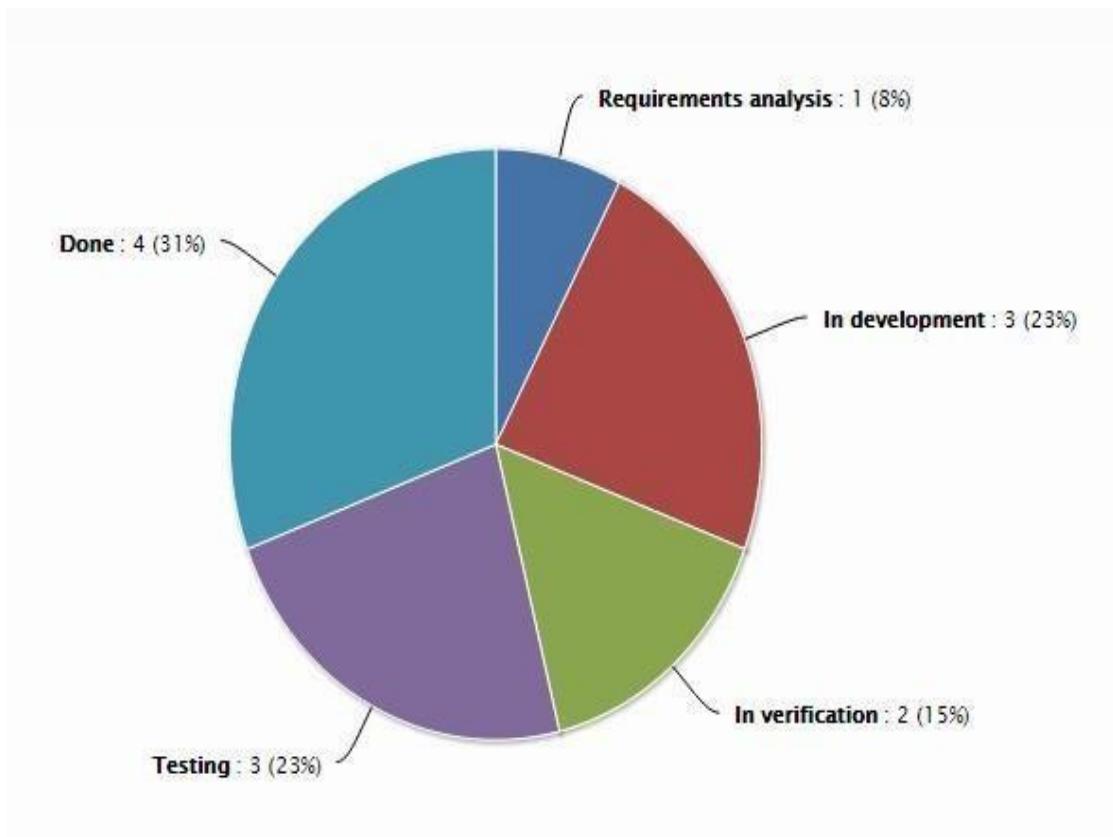
Explanation:

- Donors with blood type O can donate blood to Recipients with blood groups O, A, B, AB. That's why people with blood group O are called Universal Donors.
- Donors with blood type A can donate blood to Recipients with blood groups A, AB.
- Donors with blood type B can donate blood to Recipients with blood groups B, AB.
- Donors with blood type AB can donate blood to Recipients with blood group AB. People with blood type AB can receive blood from all the other blood types. That's why they are called the Universal Recipients.

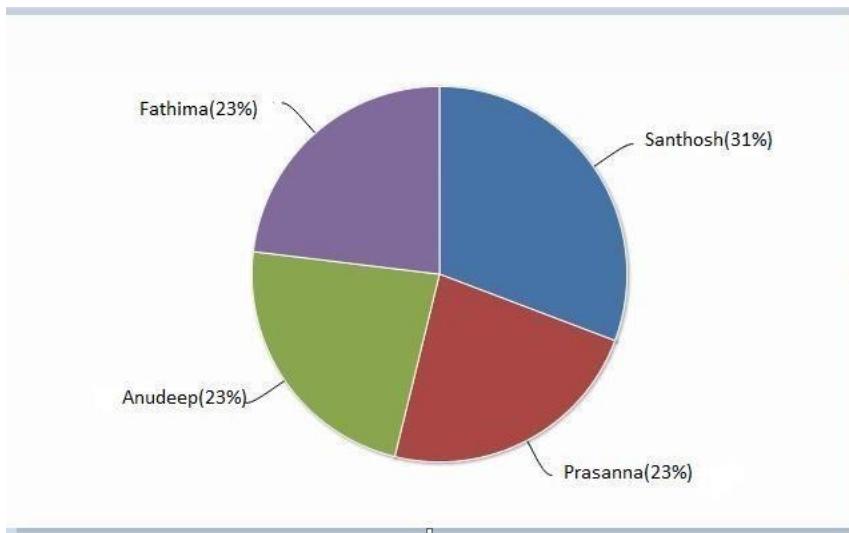
With the above observations, whenever a user wants a particular type of blood, he gets all the blood types who can donate blood to that required type not just the same blood type which the user entered. According to the above algorithm, the user has a bigger platform than just getting one single blood type. He gets to see a higher number of donor information and the probability of the correct match is higher. This improvises the typical blood donation system.

6: Analysis Graphs

- Increment Analysis graph:



- Assignment division graph



The above diagram describes about the tasks done by each group member.

7. Wireframe

7.1 Login Page

The login page has two buttons such as Submit and Register. If the user is already having an account, they can easily login by clicking the submit button. Otherwise all the users should create an account by giving their basic information.

The below screenshot shows the wireframe diagram of the login page.



7.2. Registration Page

By clicking the registration button, the user will redirect to the registration page. The registration page has all the basic information details. By entering and clicking the submit confirmation button, an account will be created for that specific user. After that the user can go back to the login and then login with their user name and password.

The below screenshot shows the wireframe registration page.

The wireframe diagram illustrates a registration page with the following components:

- A title bar at the top labeled "Registration page".
- Seven input fields arranged vertically:
 - FIRST NAME
 - LAST NAME
 - EMAIL
 - PASSWORD
 - MOBILE
 - BLOOD GROUP (with a dropdown arrow icon)
 - CURRENT LOCATION (with a location pin icon)
- A "Register" button located at the bottom center.

7.3. Home Page

Once the login is done, the user will be redirected to home page. The home page contains the information about blood donor. To getting the particular blood group donor information, the user should have to select the blood type by clicking the drop down list and type the zip code in the corresponding textbox.

As the screenshot shows, the donor name, zip code and phone number will be displayed.

The screenshot displays a user interface titled "Home page". At the top left is a dropdown menu labeled "Blood Group" with a downward arrow icon. Next to it is a search bar containing the placeholder "Enter city name" and a magnifying glass icon. Below this header, there is a list of five items, each consisting of a text input field and a small blue icon. The items are labeled "Item 1 Details", "Item 2 Details", "Item 3 Details", "Item 4 Details", and "Item 5 Details". The blue icons next to each item appear to be edit or details buttons.

7.4. Edit Page

The edit page is mainly used to do any user information update. The user wants to edit or update their old information. They can easily do it by clicking the edit button. The below screenshot shows the edit page wireframe. Simply, we can edit the home page and then update the changes.

The wireframe illustrates a 'PROFILE EDIT PAGE' interface. At the top, there is a 'LOGOUT' button. Below the header, there is a large rectangular form area containing the following fields:

- FIRSTNAME**: An input field for entering the first name.
- LASTNAME**: An input field for entering the last name.
- EMAIL**: An input field for entering the email address.
- PASSWORD**: An input field for entering the password.
- MOBILE**: An input field for entering the mobile number.
- BLOOD GROUP**: A dropdown menu indicated by a button with a downward arrow.

At the bottom of the form, there are two buttons: **UPDATE** and **CANCEL**.

7.5. Profile Preview Page

By using the profile preview page, we can preview the profile page. The below screenshot shows the profile preview page.

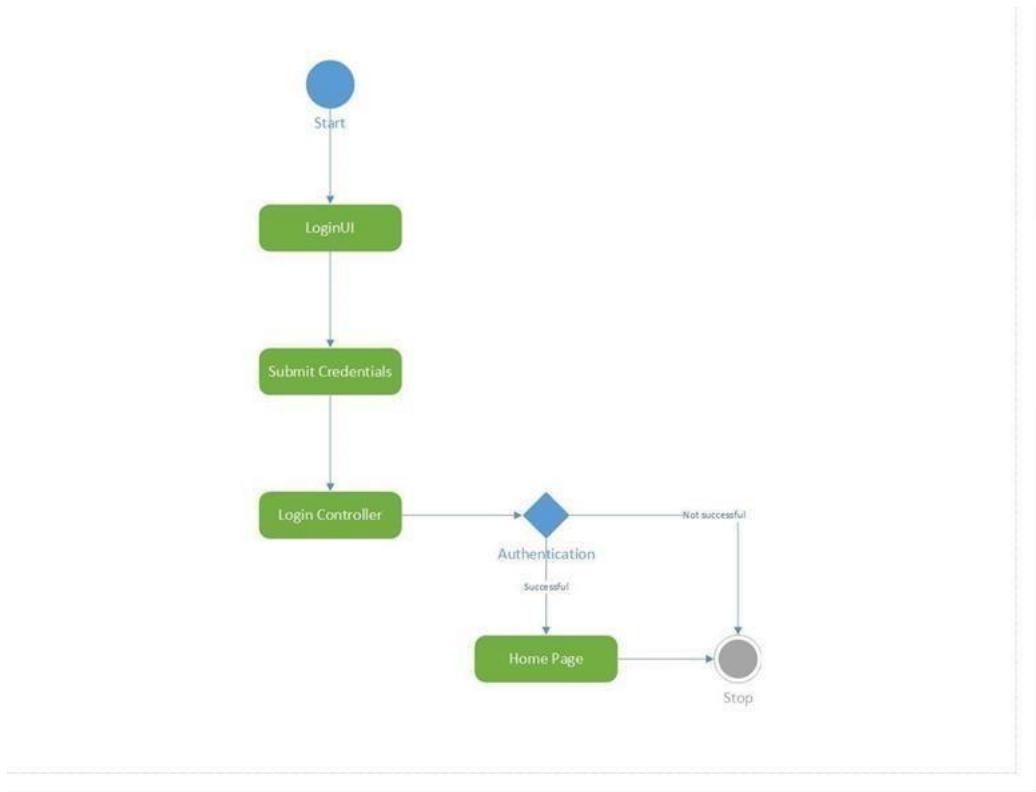


The screenshot displays a 'Profile view page' interface. At the top, there is a header bar with the title 'Profile view page' on the left, and 'EDIT' and 'LOGOUT' buttons on the right. Below the header is a large input field containing a user icon. To the right of the icon are five smaller input fields labeled 'USER NAME', 'EMAIL', 'PASSWORD', 'MOBILE', and 'BLOODGROUP' respectively.

8.State Diagrams

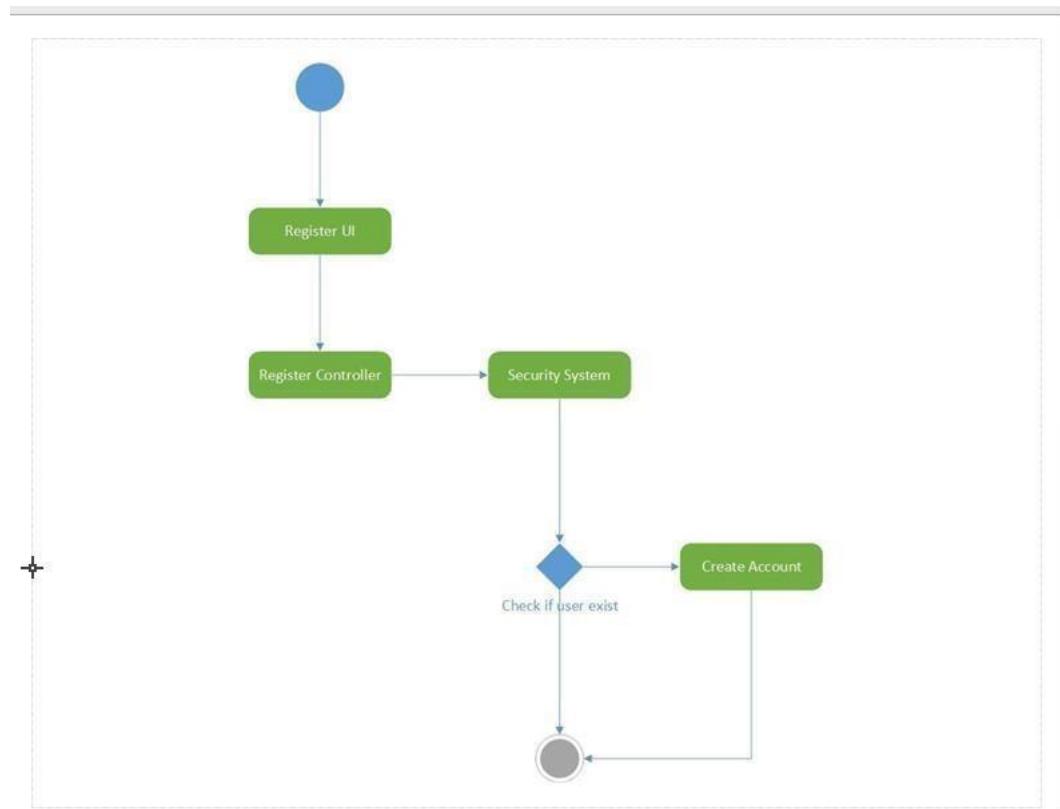
8.1. Login Page

The login page state diagram describes the login page process. If the login credentials are valid and unique then the page will be redirected to the home page by the login controller. The login process will be successful if the login controller authentication is successful otherwise the process will stop automatically.



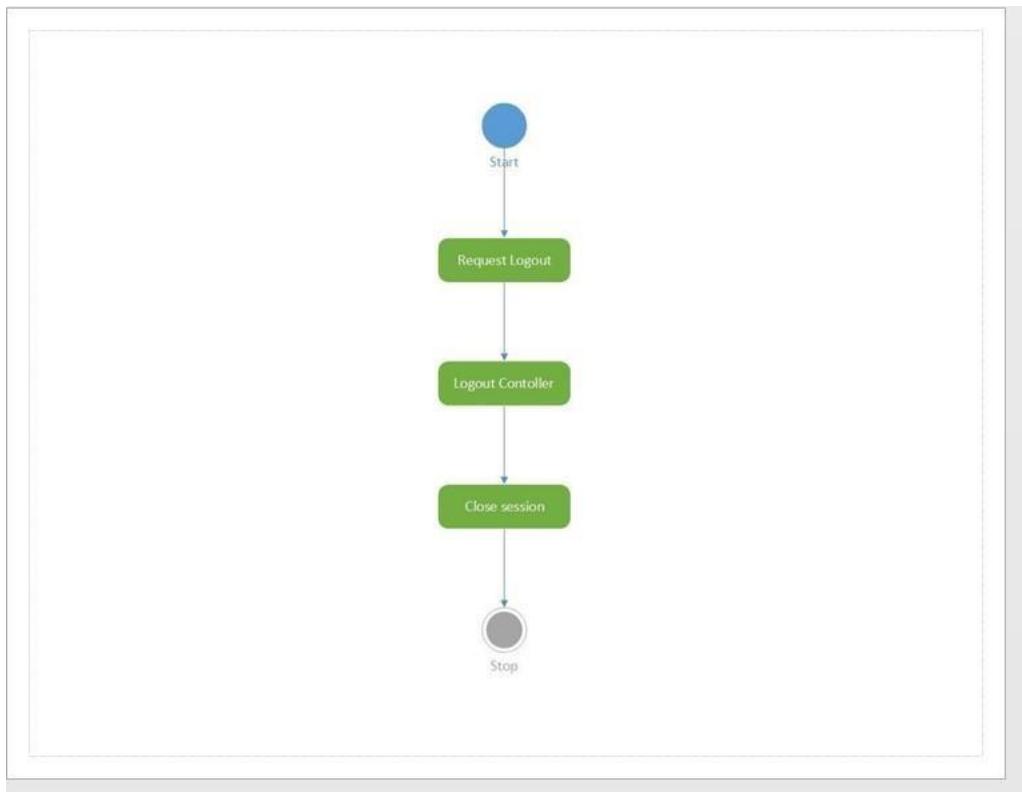
8.2. Registration Page

The state diagram of registration page verifies all the basic information about the user and then will create an account for that specific user. The registration UI will get all the basic information from the user and then pass them to the register controller. The security system will verify the user basic information. If all the values valid and unique, the new account will be created for that specific user. The below screenshot shows the State diagram of registration page.



8.3. Home Page

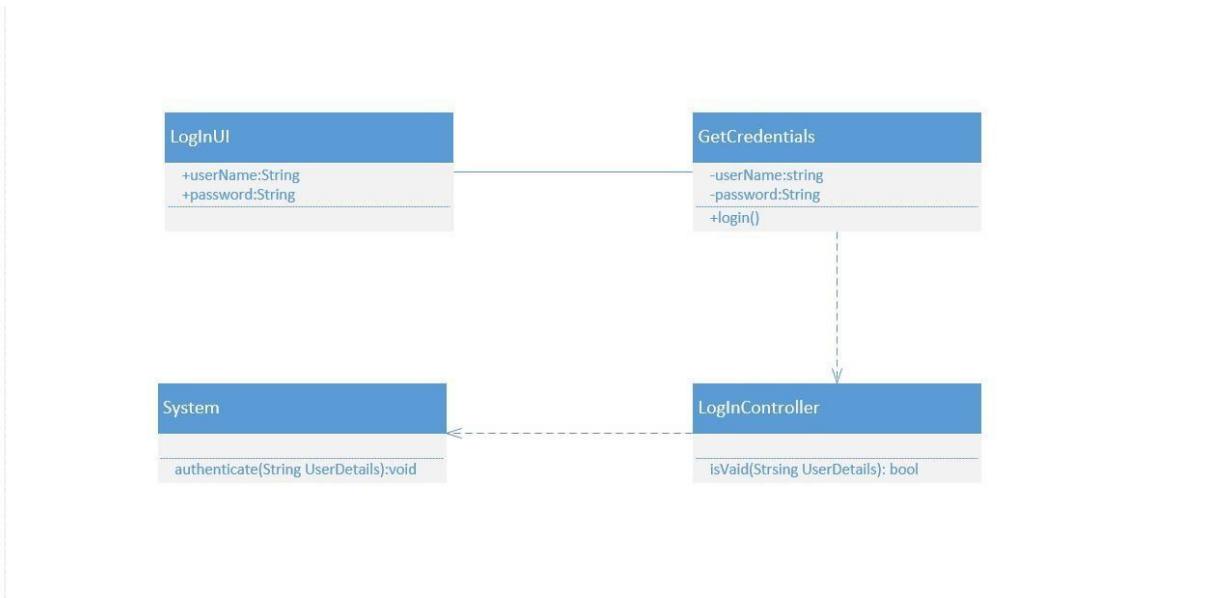
The home page displays the blood donor information. Once the user gets the information about the donor, they can logout the home page. The state diagram shows the logout steps clearly in the below screenshot. The request logout passes the request to the logout controller. The logout controller closes the session.



9. Class diagrams

9.1. Login Page

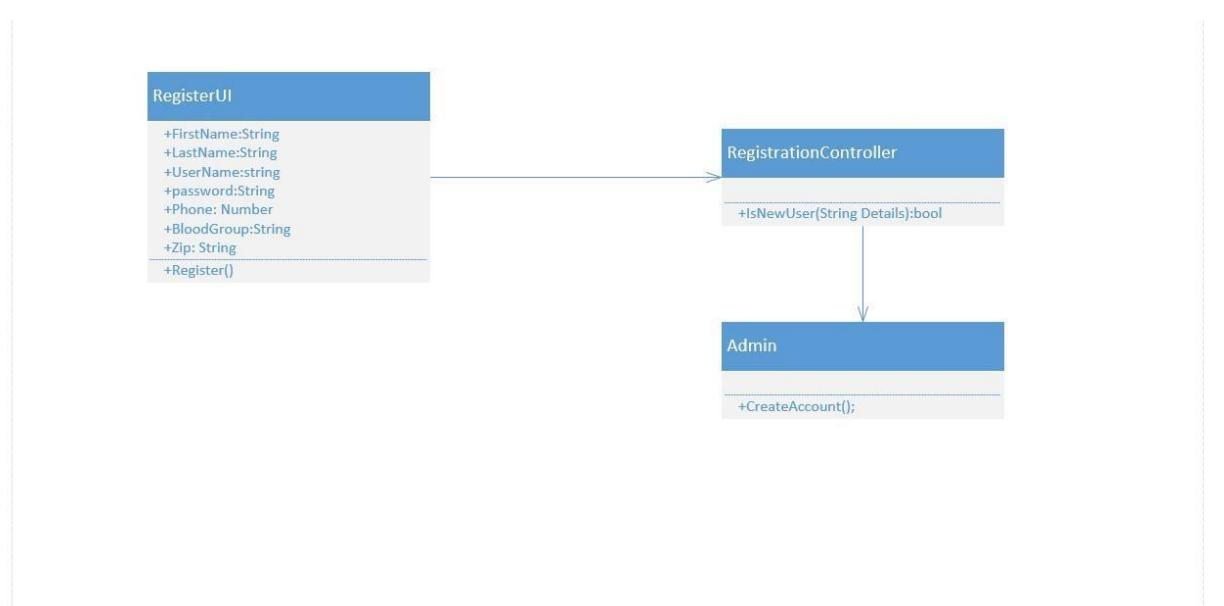
The login of the class diagram describes the flow of the system and process. The loginUI has the Username and password variables. Those variables are passing to the login controller through the GetCredentials. The system verifies the login credentials by using the authenticate function.



9.2. Registration Page

Similar to login page, the registrationUI takes all the necessary variables and pass them to the registration controller. The registration controller verifies the basic details and then the new account creates by the Admin.

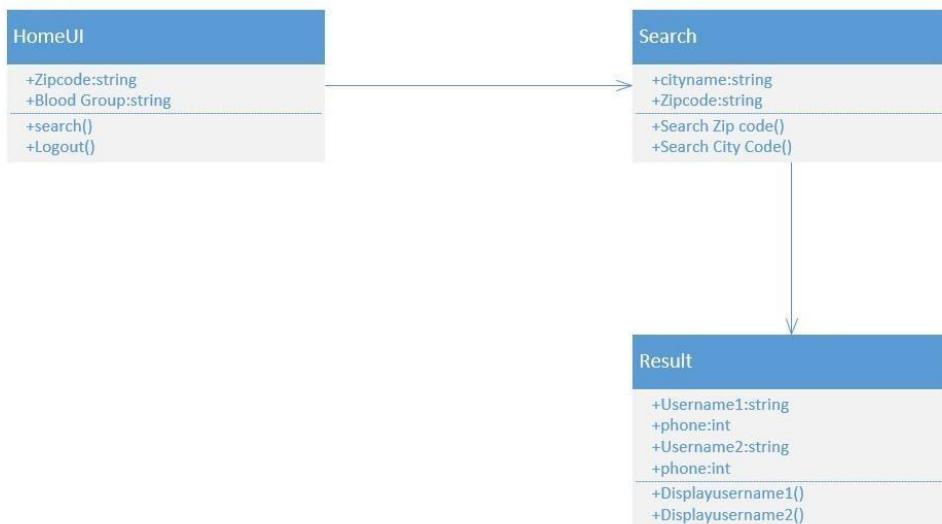
The below screenshot shows the class diagram of registration page.



9.3. Home Page

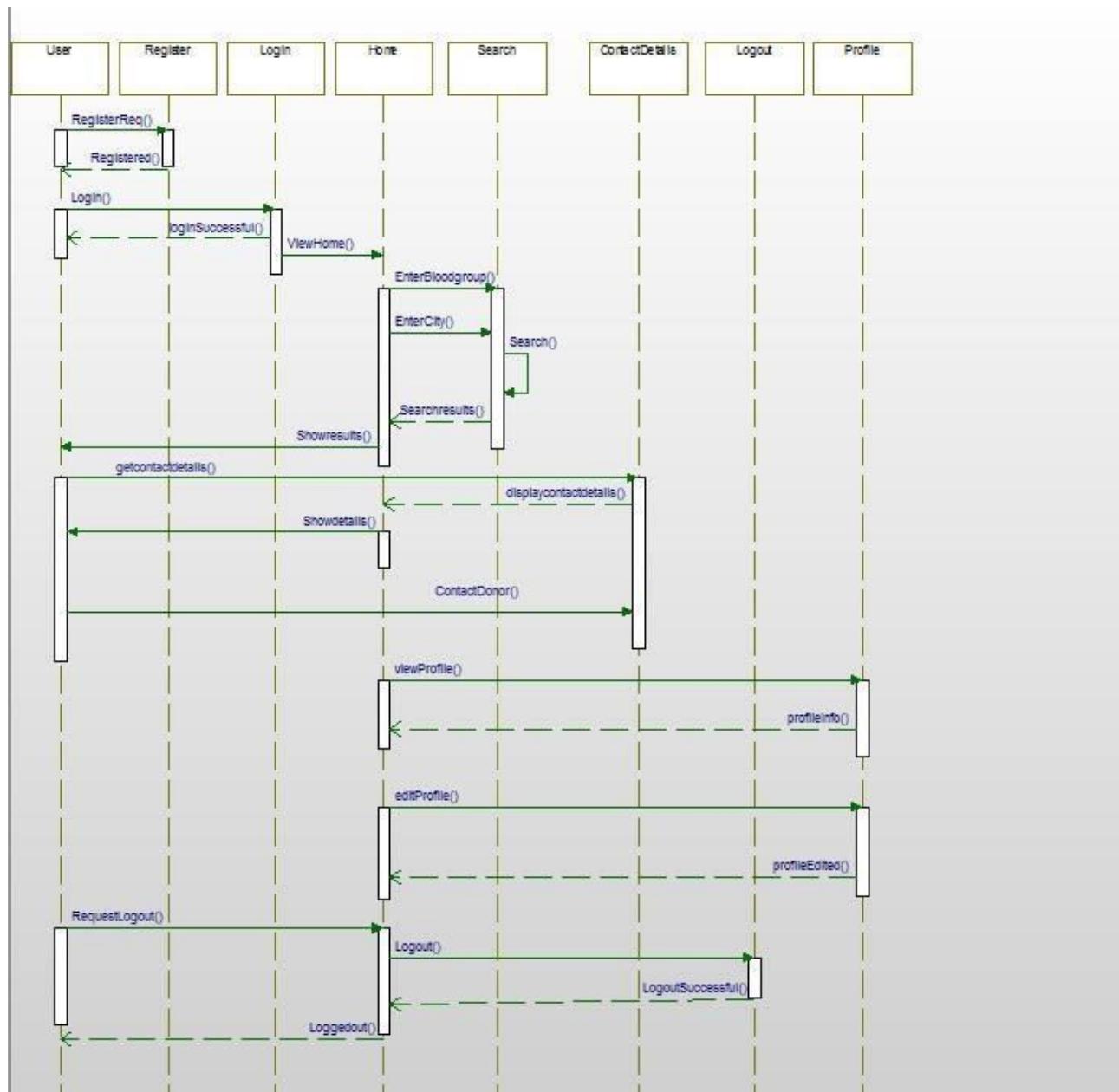
The home page donor search will take places by the search variables from the HomeUI. The search operation will happen based on the zip code and the blood type. Finally, the result will be displayed based on the blood type and zip code.

The below screenshot shows the class diagram of home page.

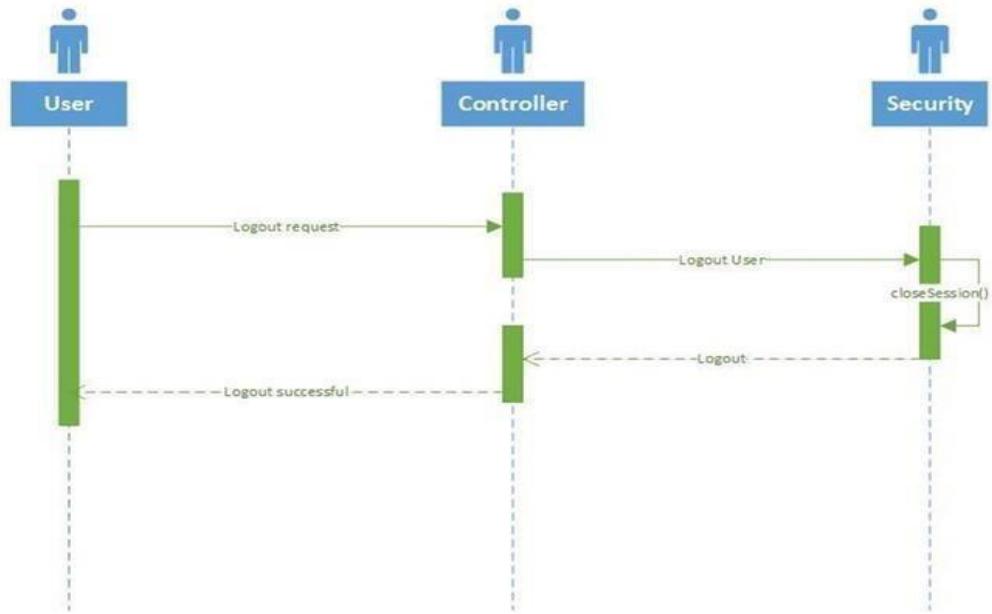


9.Sequence Diagram

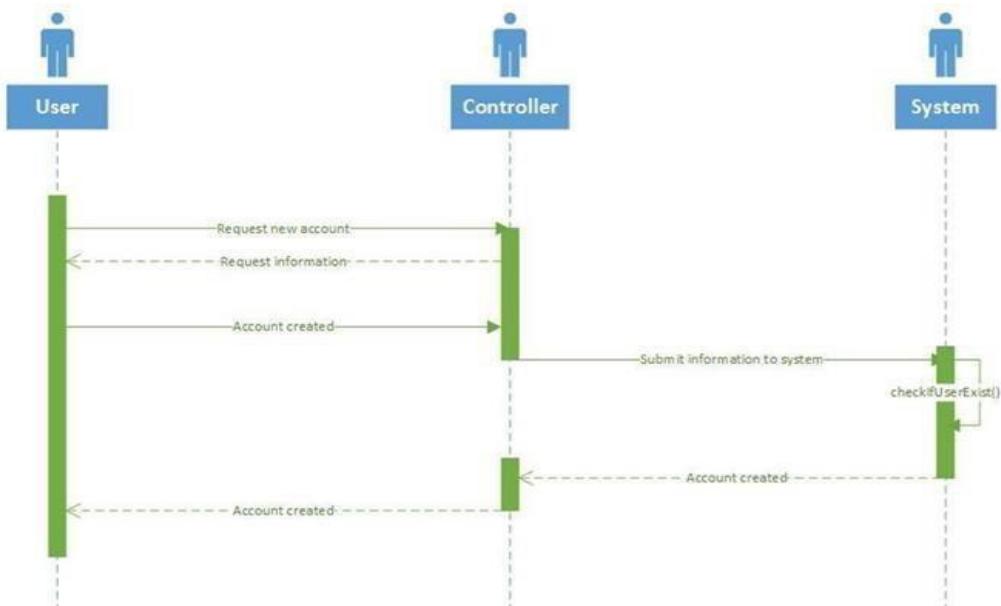
The sequence diagram describes the complete process of this project. As the screenshot shows, the registration process takes place first and the login function takes place. If the login is successful, then the view home function takes place. Sequentially, the enterbloodgroup(), enterzipcode() functions are taking place for the search. Once the search is successful, the searchresults () and showresults () functions are taking place to display the results. Finally, the logout () function takes place to logout the page.



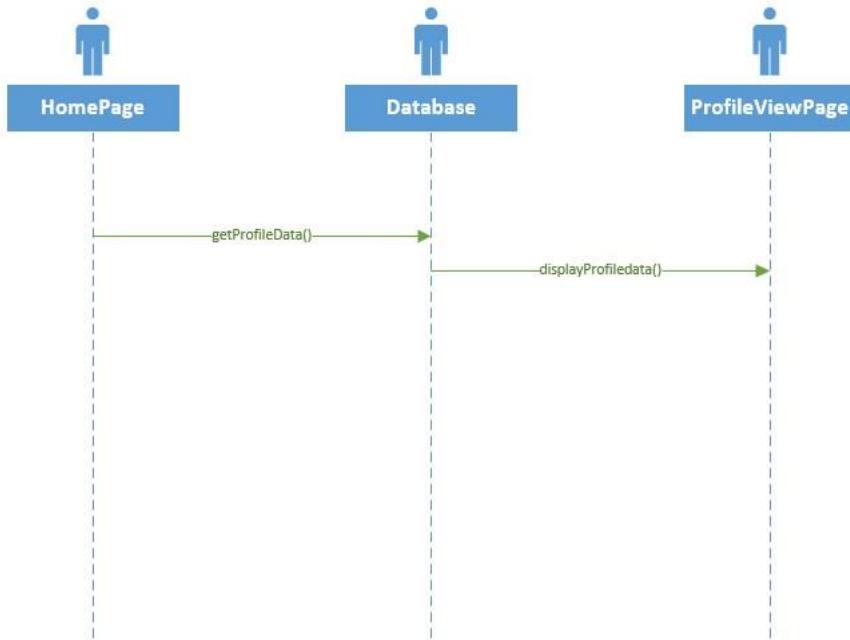
9.1. Login Sequence Page



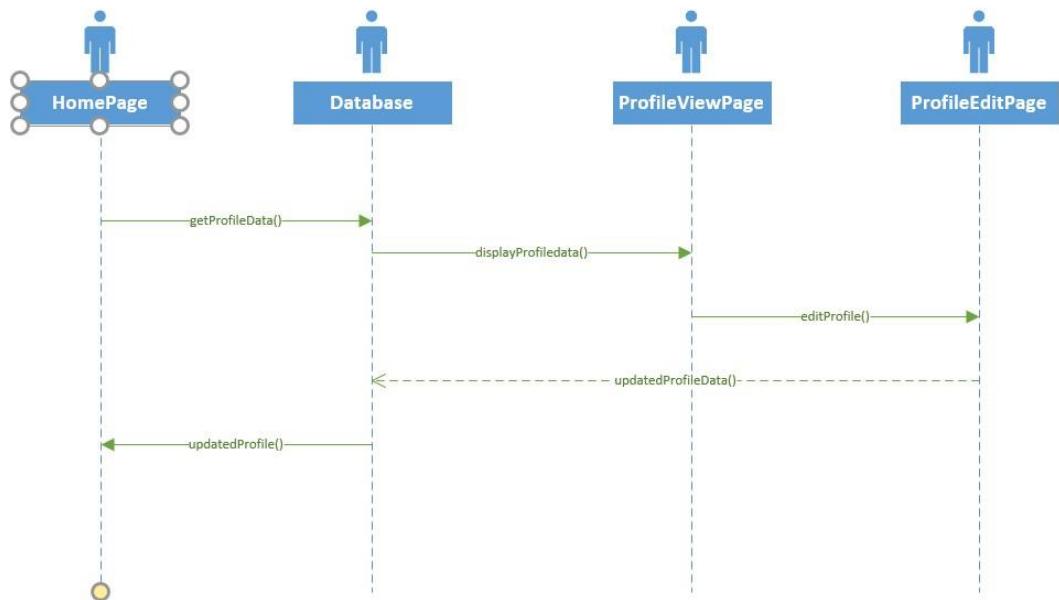
9.2. Registration Sequence Page:



9.3. Profile View Page Sequence Diagram



9.4. Edit Profile Page Sequence Diagram



10. Layouts:

10.1. Welcome Page

The below screenshot shows the Welcome page of the uml output. The user can **Login**, if they already have an account. Otherwise, they should create an account by clicking the **register** button.



In future, we are planning to increment Social Login such as Facebook or Gmail in the center of the layout.

Note: If the user is already logged in, application doesn't ask him to login again and Welcome Page is not displayed. Session expires when the user clicks the logout button and will be redirected to the Login page.

10.2.Registration page:

The Registration page has all the basic information tabs. After giving and entering all the basic information, new account will be created for that specific user. The below screenshot shows the registration page.





Login Page



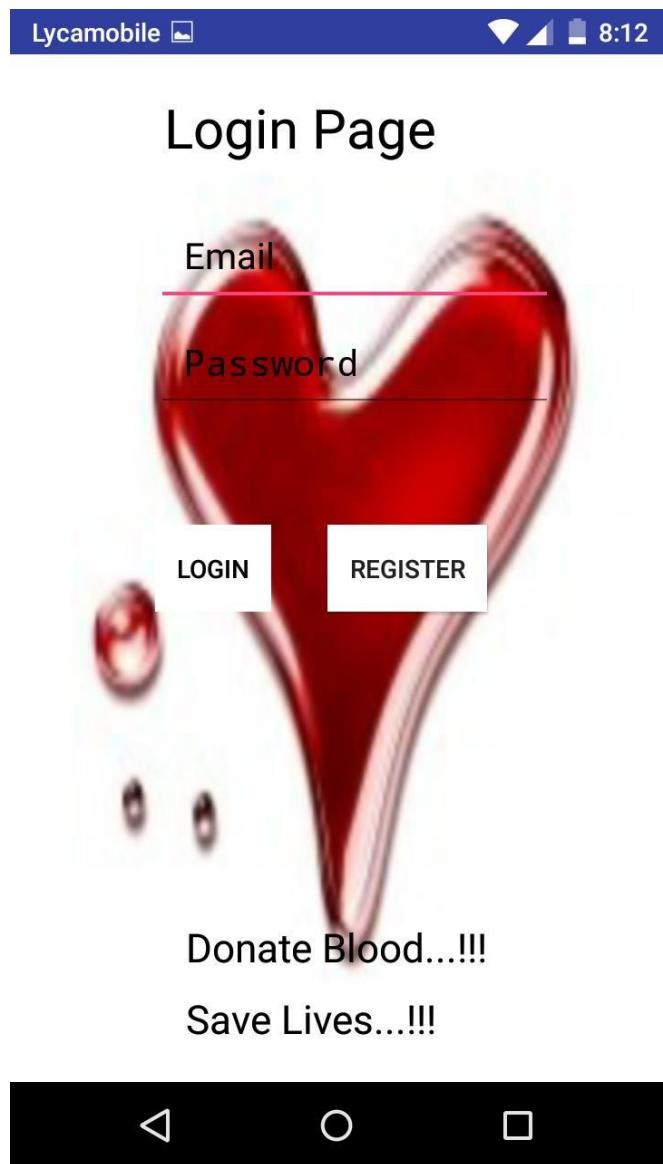
Address can be entered manually or by clicking the Use current location marker.

Validations have been added for Firstname, Lastname and Mobile Number.

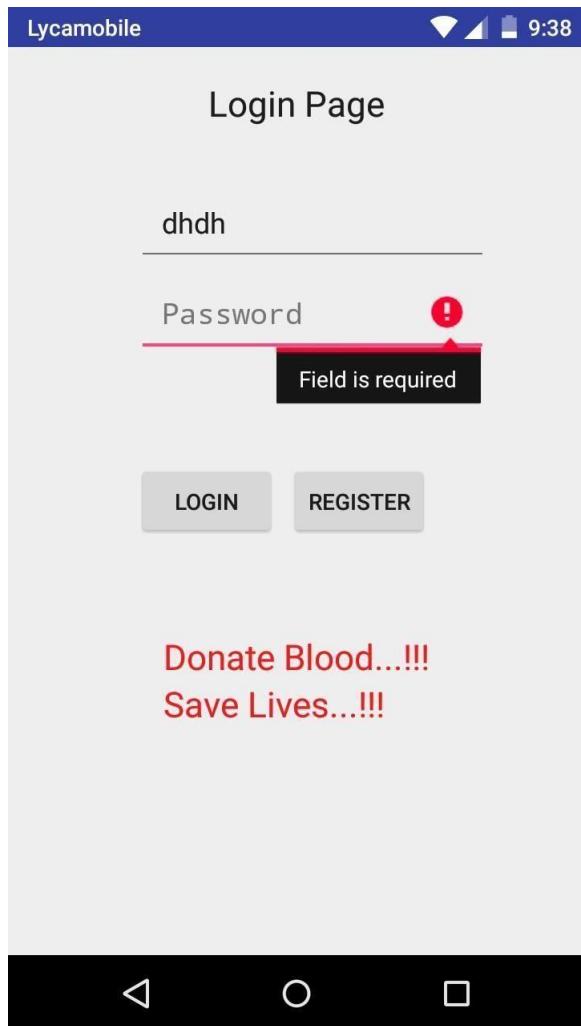
Firstname and Lastname can't have numbers and Mobile Number can't have characters and it takes exactly 10 numbers.

10.3. Login Page:

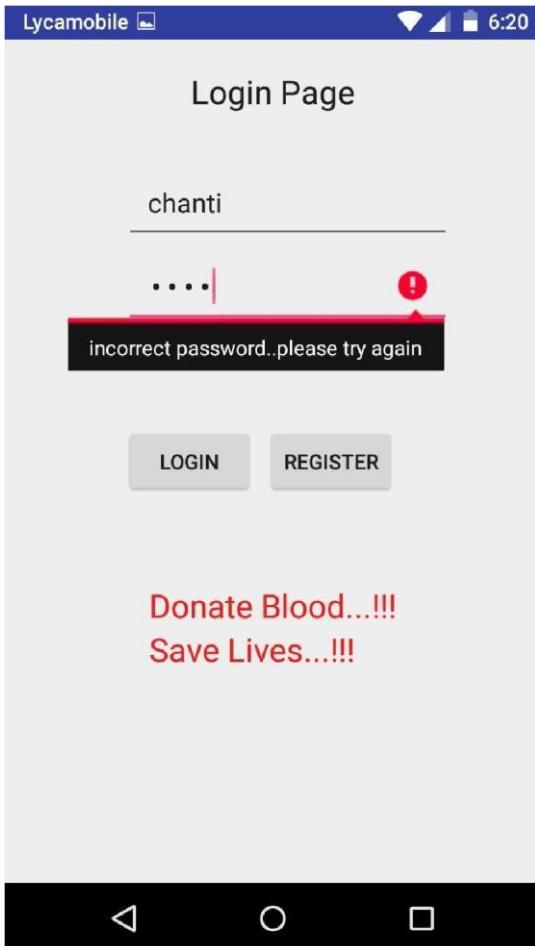
- If the Username field is empty, an error message pops up telling the Username field is required.



- If the password field is empty, an error message pops up telling the Password field is required.



- If the user credentials are not in the database, an error message pops up telling the credentials are invalid.



- If the user credentials are valid, user is redirected to the home page.

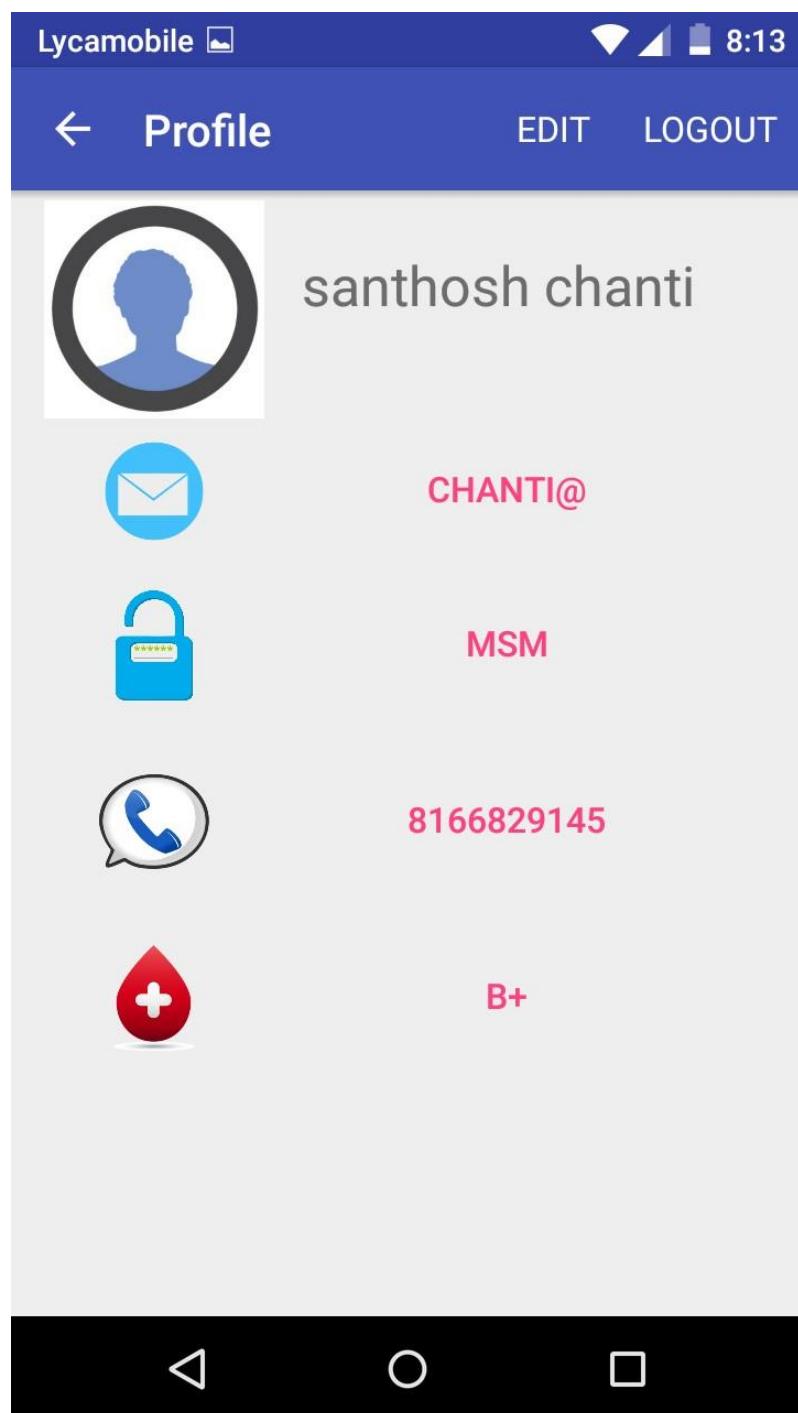
10.4. Home Page:

The home page provides the search results based on the search conditions. To get the search results, the search information has been entered in the corresponding search boxes. The below screenshot shows the home page of xml output.





Profile Edit Page



Edit Page

Lycamobile 8:13

Edit LOGOUT

FIRST NAME	santhosh
LAST NAME	chanti
EMAIL	chanti@
PASSWORD	msm
MOBILE	8166829145
BLOOD GROUP	B-

UPDATE CAN CEL

◀ O □

11. Architecture/Design

UI Design:

We have the following UIs in our project.

- **Login Page:** This is an xml layout with two text fields and two buttons. User Details will be authenticated by using the text from those text fields. If the user is new he has to register, so by clicking register button he will be redirected to registration page. If the user is successfully logged in, he will be redirected to home page.
- **Registration Page:** This is an xml layout with 6 text fields, 1 radio button and register button. User has to give name, mobile number, username, zip code and more importantly blood group name. If all the required fields are validated once user clicks register button. If user has account already, an error message will be thrown with message as "account already exists, please go to login page". If he is a new user, all the details will be stored in our database and he will be redirected to Home Page.
- **Home page:** This is the main page where user will be redirected after successful login. In this page, user can search donors who are available in the location he provided. He can view the contact details of the donors. We will extend this functionality to google maps where all the donors are seen in the maps. User can also update profile information. There is a logout button where user can close the session.

12. Database

We are planning to use MySQL database or SQLite database to store user details. If we use social logins, we get required details from the corresponding APIs.

We have only one table named “User”.

The table consists of following columns.

- First Name : Char ○
- Last Name : Char ○
- Mobile : Number ○ User
- Name :
- Char ○ Password :
- Char ○ Address :
- Char ○ Blood group :
- Char ○ Created at:
- DateTime ○ Updated at :
- DateTime

The below is the screenshot for the Firebase database.

It's a NoSQL Database where data is stored as JSON objects.

The screenshot shows the Firebase Realtime Database dashboard. On the left, there is a sidebar with icons for Data, Security & Rules, Simulator, Analytics, Login & Auth, Hosting, and Secrets. The 'Data' icon is selected. The main area shows a hierarchical tree structure under the 'bloodmanagement' database. The 'Users' node has four children: 'chanti', 'chanti1913', 'pras', and 'raviteja83'. The 'chanti1913' node contains several fields: 'address' (with a long address), 'blood_group' ('A+'), 'first_name' ('chanti'), 'last_name' ('mohan'), 'mobile' ('049'), 'password' ('msm'), and 'user_name' ('chanti1913'). A legend on the right indicates that yellow squares represent 'Changed' data, green squares represent 'Added' data, red squares represent 'Deleted' data, and blue squares represent 'Moved' data. There is also an 'Import Data' and 'Export Data' button at the top right.

13. Four Different Increments:

13.1 Increment 1-- Requirement Gathering and Designing the application

1. Research on importance of blood donation management, requirements for the development.
2. Setup android studio, zenhub and Github for every team member.
3. Design class diagrams.
4. Design sequence diagrams.
5. Design state diagrams.
6. Design WireFrames.
7. Design basic UI layouts (Login, Registration, Home)

13.2 Increment 2—Coding/Testing

- 1) Implement Login/logout /Registration/ Registration validation and functionality.
- 2) Implement home page.
- 3) Setup local databases.
- 4) Test Login/logout UI/Registration/Registration validation.
- 5) Test sample user data.

13.3 Increment 3 - Coding/Testing 1)

- 1) Implement search functionality
- 2) Implement View and Edit Profile Functionality
- 3) Test search functionality
- 4) Test Edit Profile UI and functionality
- 5) Implementation of field validations.

13.4 Increment 4 -- Refine GUI

- 1) Refine GUI for Login/Registration.
- 2) Refine GUI for Home Page
- 3) Refine GUI for Profile Page
- 4) Refine GUI for Welcome Page.

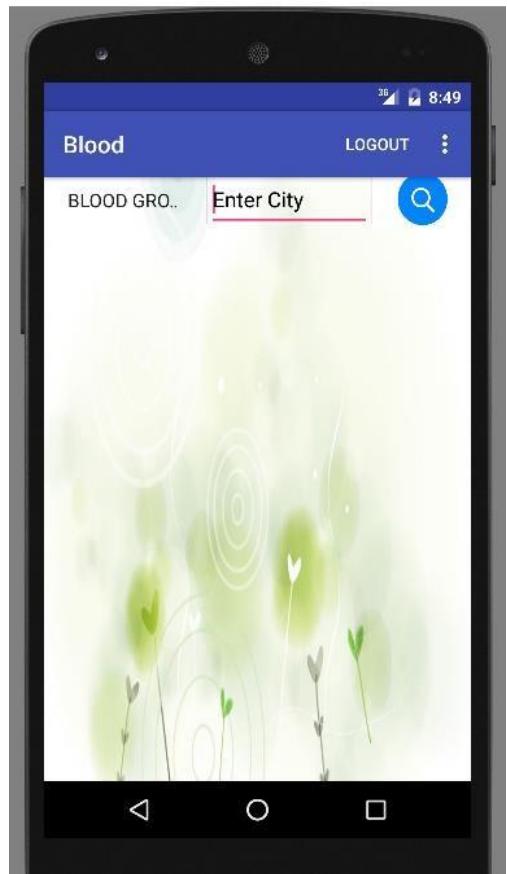
14. TESTING:

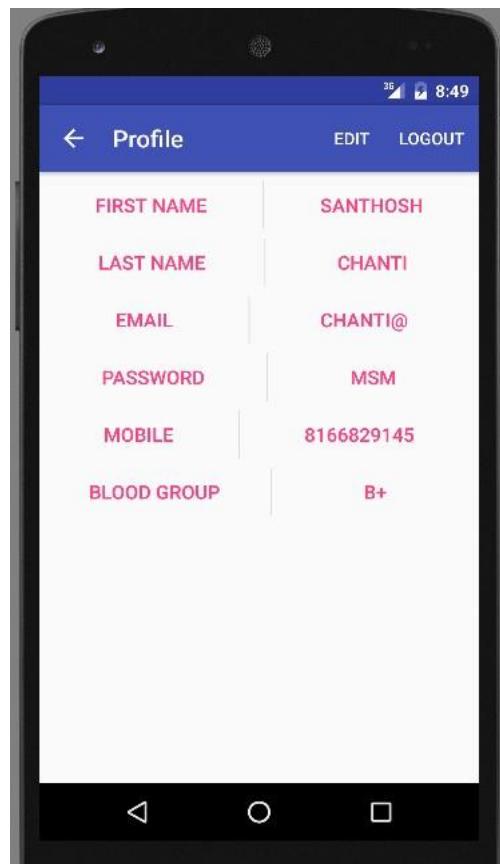
14.1 Emulator testing:

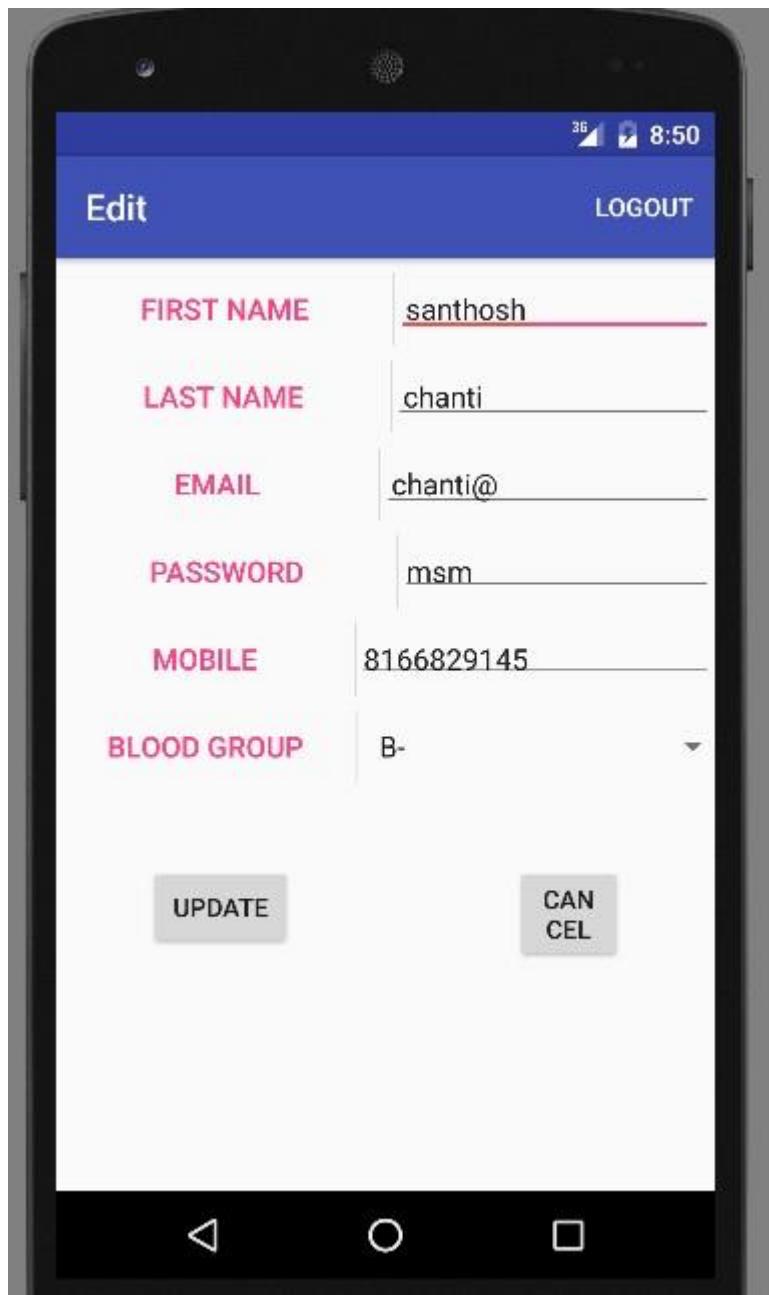












15. Project Timelines, Members, *Task Responsibility*

Member and Responsibilities:

Artifacts	Members: Santhosh, Anudeep, Prasanna, FathimaJ
Research	All
Project Plan	Santhosh mohan
UML Diagrams	Prasanna, Anudeep
Modeling Database	Anudeep, Prasanna
Initial Mockup	FathimaJ
Reports	All
Development	Santhosh Mohan, Prasanna, Anudeep
Testing	Anudeep, FathimaJ
Maintenance	Santhosh Mohan

Project Timelines:

Increments	Tasks
Increment 1	Initial setup of environments, uml diagrams and code (Login UI, Registration UI, Home UI)
Increment 2	Code (Login, Register, Home Java code functionalities basic)
Increment 3	Code + Test (Profile functionality, testing) and use case execution
Increment 4	Map functionality, Deployment to play store, final report.

16. Bibliography

- www.github.com
- www.developer.android.com
- www.developer.google.com
- <http://www.redcrossblood.org/learn-about-blood/blood-types>
- https://en.wikipedia.org/wiki/Blood_type
- <https://www.firebaseio.com/>
- <https://www.firebaseio.com/docs/android/guide/>
- <https://www.firebaseio.com/blog/>

CS5551 ADVANCED SOFTWARE ENGINEERING

PROJECT REPORT-PHASE 4

BLOOD DONATION MANAGEMENT

TEAM MATES

- 1. PRASANNA MUPPIDI(33)**
- 2. SANTHOSH MOHAN MURARISHETTI(35)**
- 3. ANUDEEP PANDIRI(40)**
- 4. FATHIMA JAMES(58)**

1.Introduction

Our Blood Management application helps the users to find the donor for a particular blood group in a particular location in the easiest way possible. All the donors who wish to donate blood are supposed to enter their details in the application. The details include their Name, Age, Contact number, Email ID, Location (ZIP Code) and most importantly their blood group. The recipients who wish to receive blood should login to the application with their mobile number and are supposed to search the forum based on the required blood group and the location, they can contact the donor for further details.

The project has been divided into four phases with improving implementation features. For the first iteration of our project we want to complete all design section of the application with Login, Registration, Home page designs. We have chosen the android platform to develop our application. For this first phase, we have designed the UML Class diagram, Sequence diagram, State diagrams along with wireframes. We concentrated mainly on the design part which play a major role in implementing our project.

2.Functions:

2.1 Login

Login Page lets the user login to the application. User needs to enter the Username and Password in order to login. If the user doesn't have an account to Login, he should register first.

2.2 Register

Register page lets the user to register. User needs to provide personal information to create an account.

2.3 Home

In Home Page, already registered user details like Name, Zip Code, Contact Numbers are displayed.

2.4 Search

Once the user registers, all his personal information is stored in SQLite database. User who needs blood should login into the application and search based on the blood group and Zip code and he can then contact the donor based on the contact information present in the application.

2.5 Admin

Users can anytime contact Admin for emergency situations. If any of the contact details or the Blood Group is different from what is required, Admin helps the user for more information.

3. Proposed System

1. Requirement Specification:

- Functional Requirements:
 - i. User should have a Sign in.
 - ii. If user is new, he should have a Register. iii. User should be able to provide his personal information.
 - iv. User details should be valid.
 - v. User should be able to search based on the Blood Group. vi. User should be able to search based on the Zip Code. vii. User should be able to view the corresponding contact details. viii. User should be able to contact the Admin in emergency situations.
 - ix. User should be able to contact the donor whose details user found on the application.
 - x. User should be able to close the session. xi. User should be able to logout.

4. Technologies Used:

- i. ADT: Android Studio
- ii. Programming : JAVA
- iii. Database: SQLite
- iv. Frontend: XML, JavaScript

5.Development

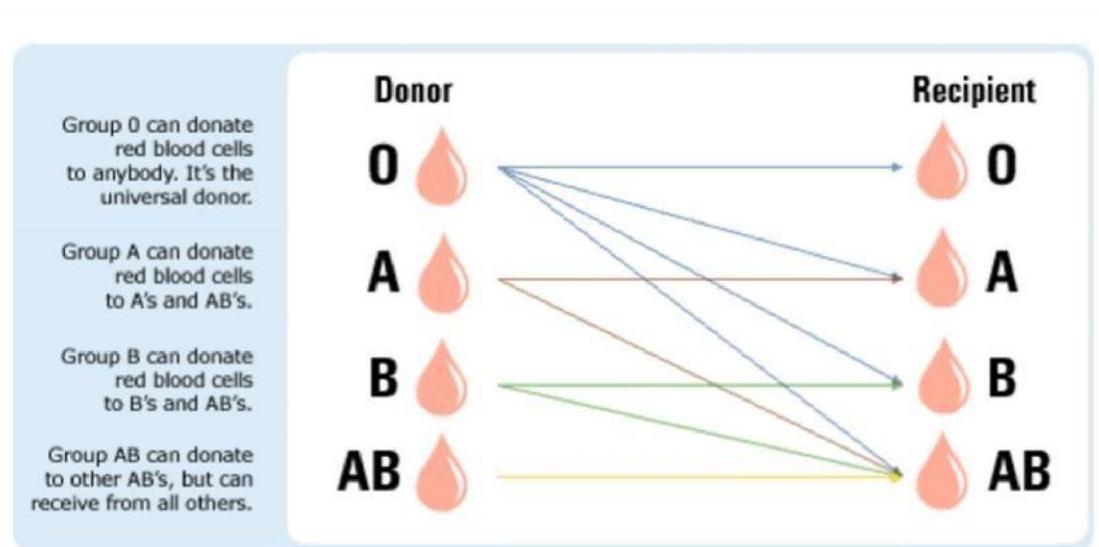
System Designs:

- 1) Login: Allow users to login into their account. Users use email address and Password to identify themselves.
- 2) Registration: Allow new users to create an account. User information collected during this process:
 - First name
 - Last name
 - Date of birth
 - Email address
 - Password
 - Blood Type
 - Address
 - Mobile Number
 - Gender
- 3) Logout: Allow users to logout of their account
- 4) Registration validation: All fields should be valid.
- 5) Blood donation: The registration form takes the blood group of the user while registering. The entered value should be a valid blood type. It's a drop down system where the user needs to select one of the available blood types. The user should have a valid email ID and a Mobile Number.
- 6) Contacts display: Once the user enters the blood group he wants, he gets a list of all the corresponding blood types and all the contact details

UIs:

- 1) Login UI: Implement login activity UI design
- 2) Register UI: Implement register activity UI design
- 7) Search UI: Implement search activity UI design
- 3) Blood Group Selection UI: Implement Blood group activity UI design
- 4) Donor Details UI: Implement Donor details UI design

The below picture illustrates the specific ways in which blood can be donated.



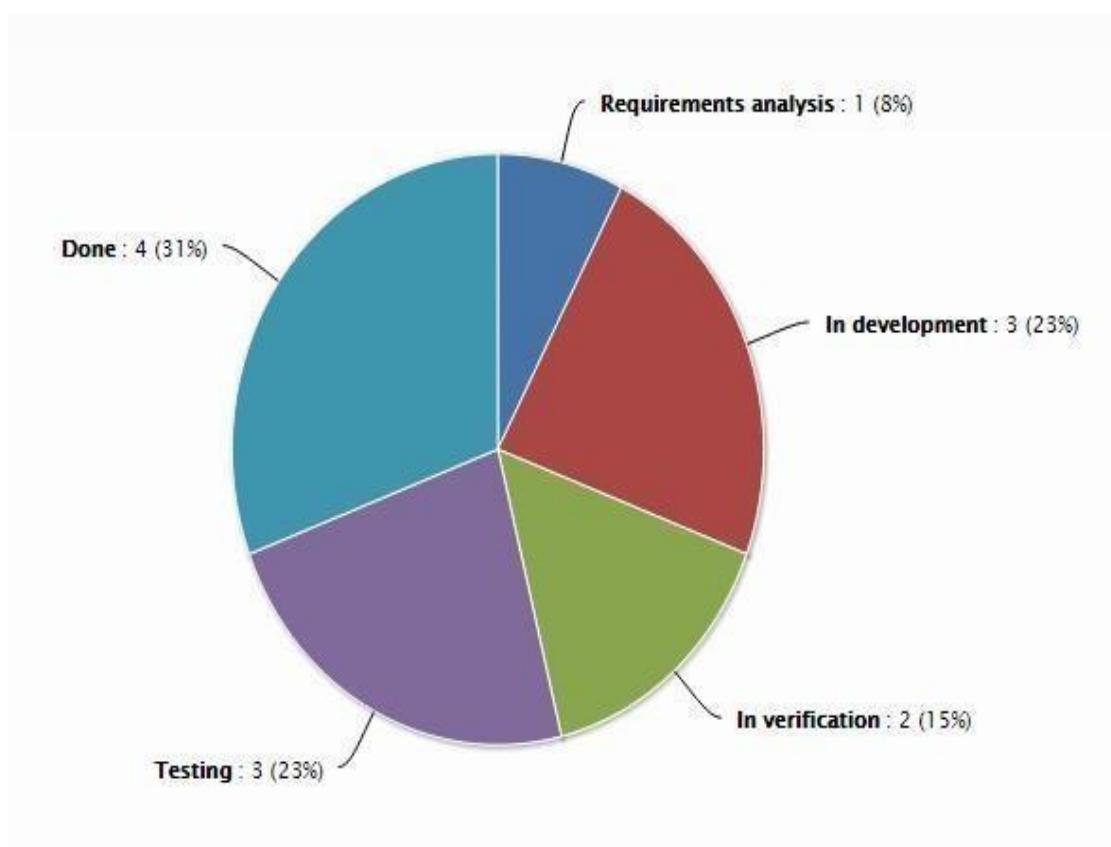
Explanation:

- Donors with blood type O can donate blood to Recipients with blood groups O, A, B, AB. That's why people with blood group O are called Universal Donors.
- Donors with blood type A can donate blood to Recipients with blood groups A, AB.
- Donors with blood type B can donate blood to Recipients with blood groups B, AB.
- Donors with blood type AB can donate blood to Recipients with blood group AB. People with blood type AB can receive blood from all the other blood types. That's why they are called the Universal Recipients.

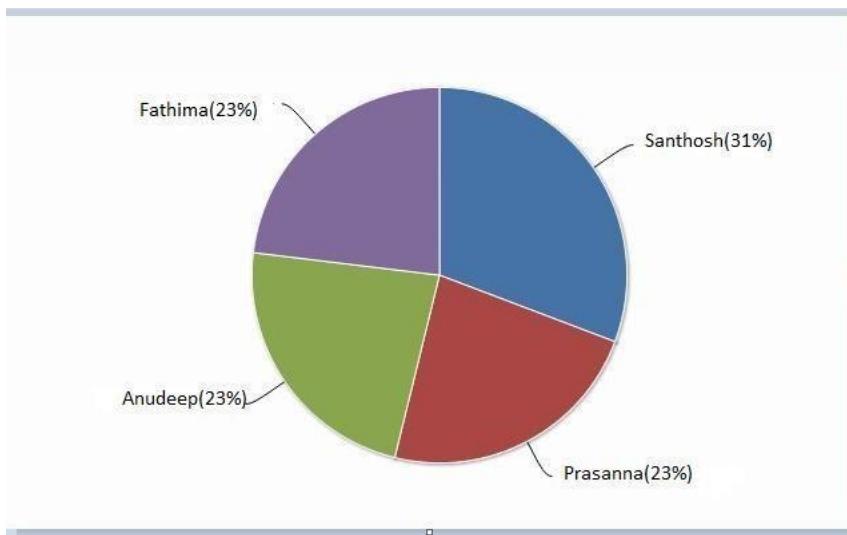
With the above observations, whenever a user wants a particular type of blood, he gets all the blood types who can donate blood to that required type not just the same blood type which the user entered. According to the above algorithm, the user has a bigger platform than just getting one single blood type. He gets to see a higher number of donor information and the probability of the correct match is higher. This improvises the typical blood donation system.

6. Analysis Graphs

- Increment Analysis graph:



- Assignment division graph



The above diagram describes about the tasks done by each group member.

7. Wireframe

7.1 Login Page

The login page has two buttons such as Submit and Register. If the user is already having an account, they can easily login by clicking the submit button. Otherwise all the users should create an account by giving their basic information.

The below screenshot shows the wireframe diagram of the login page.



7.2. Registration Page

By clicking the registration button, the user will redirect to the registration page. The registration page has all the basic information details. By entering and clicking the submit confirmation button, an account will be created for that specific user. After that the user can go back to the login and then login with their user name and password.

The below screenshot shows the wireframe registration page.

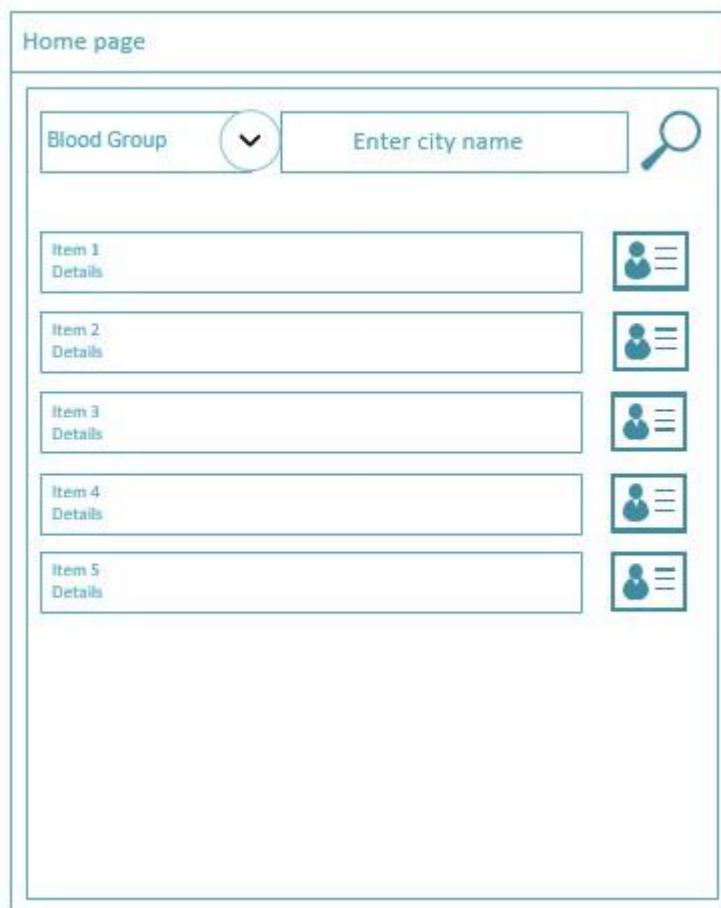
The wireframe illustrates a registration form titled "Registration page". It contains the following fields:

- First Name
- Last Name
- Email
- Password
- Mobile
- Blood Group (with a dropdown arrow icon)
- Current Location (with a location pin icon)
- Register button

7.3. Home Page

Once the login is done, the user will be redirected to home page. The home page contains the information about blood donor. To getting the particular blood group donor information, the user should have to select the blood type by clicking the drop down list and type the zip code in the corresponding textbox.

As the screenshot shows, the donor name, zip code and phone number will be displayed.



7.4. Edit Page

The edit page is mainly used to do any user information update. The user wants to edit or update their old information. They can easily do it by clicking the edit button. The below screenshot shows the edit page wireframe. Simply, we can edit the home page and then update the changes.

The wireframe illustrates the layout of the Profile Edit Page. At the top left is the title "PROFILE EDIT PAGE" and at the top right is a "LOGOUT" link. The main content area contains six input fields: "FIRSTNAME", "LASTNAME", "EMAIL", "PASSWORD", and "MOBILE". Below these is a dropdown menu labeled "BLOOD GROUP" with a downward arrow icon. At the bottom of the form are two buttons: "UPDATE" on the left and "CANCEL" on the right.

7.5. Profile Preview Page

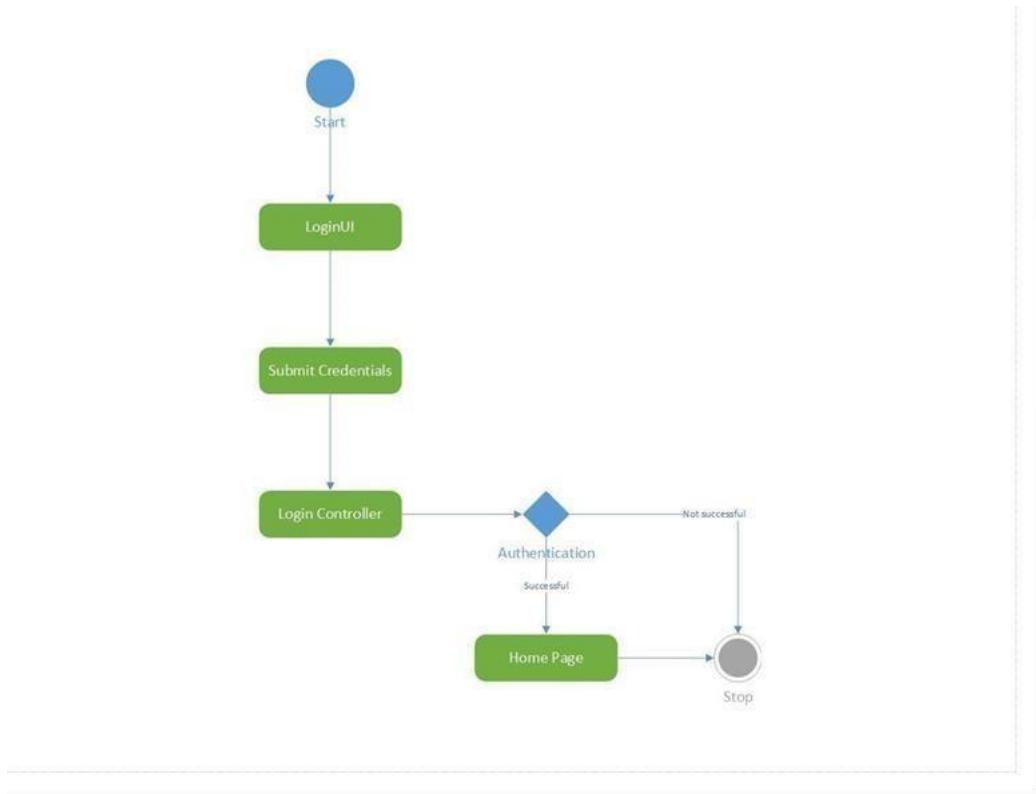
By using the profile preview page, we can preview the profile page. The below screenshot shows the profile preview page.

Profile view page		EDIT	LOGOUT
	USER NAME		
	EMAIL		
	PASSWORD		
	MOBILE		
	BLOODGROUP		

8.State Diagrams

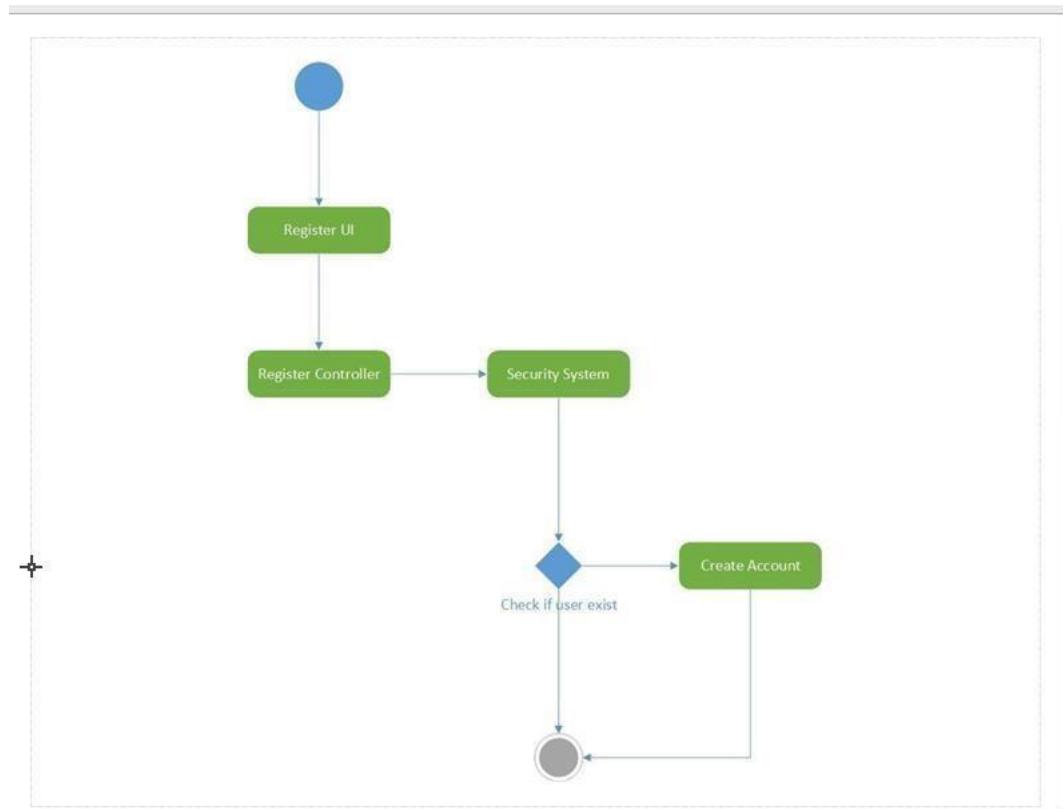
8.1. Login Page

The login page state diagram describes the login page process. If the login credentials are valid and unique then the page will be redirected to the home page by the login controller. The login process will be successful if the login controller authentication is successful otherwise the process will stop automatically.



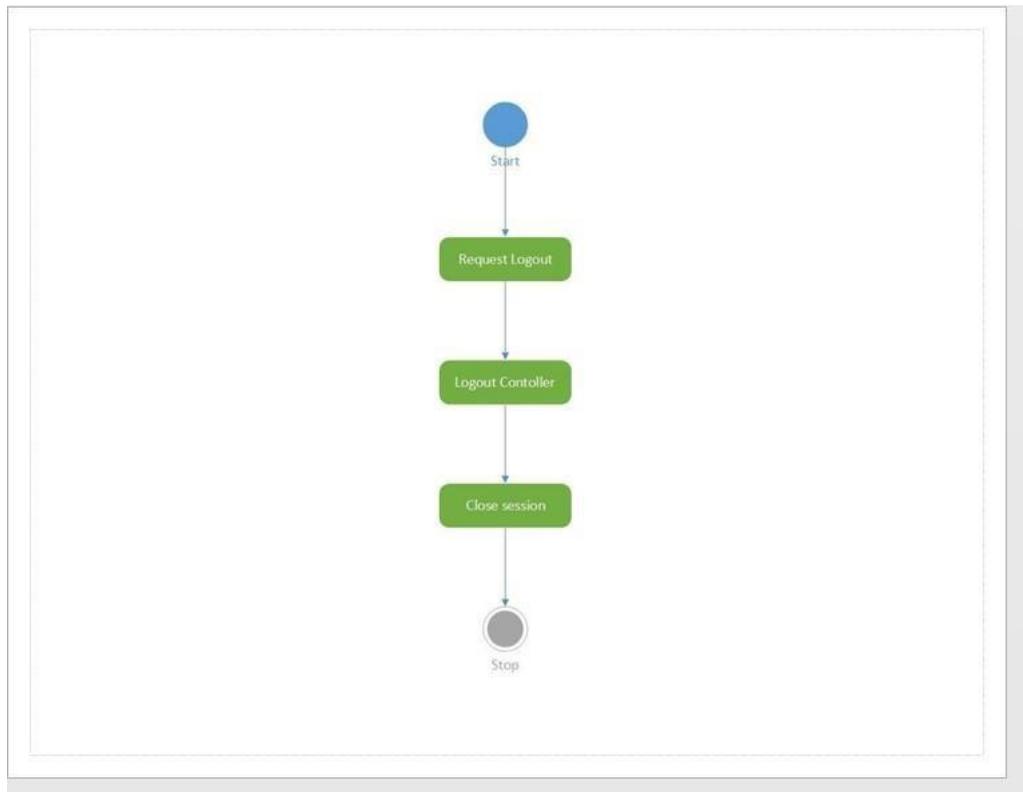
8.2. Registration Page

The state diagram of registration page verifies all the basic information about the user and then will create an account for that specific user. The registration UI will get all the basic information from the user and then pass them to the register controller. The security system will verify the user basic information. If all the values valid and unique, the new account will be created for that specific user. The below screenshot shows the State diagram of registration page.



8.3. Home Page

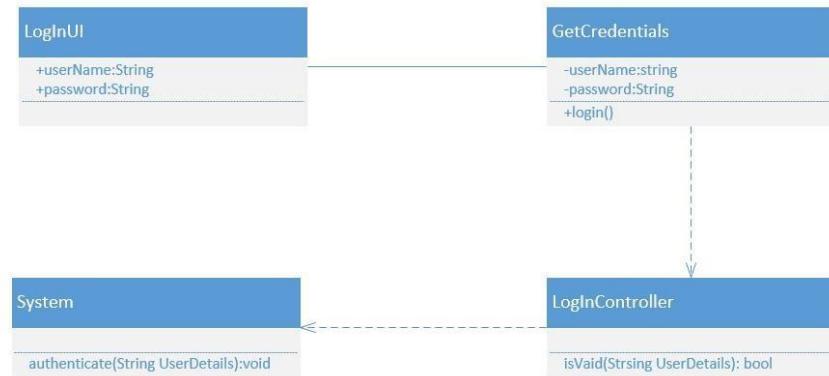
The home page displays the blood donor information. Once the user gets the information about the donor, they can logout the home page. The state diagram shows the logout steps clearly in the below screenshot. The request logout passes the request to the logout controller. The logout controller closes the session.



9.Class diagrams

9.1. Login Page

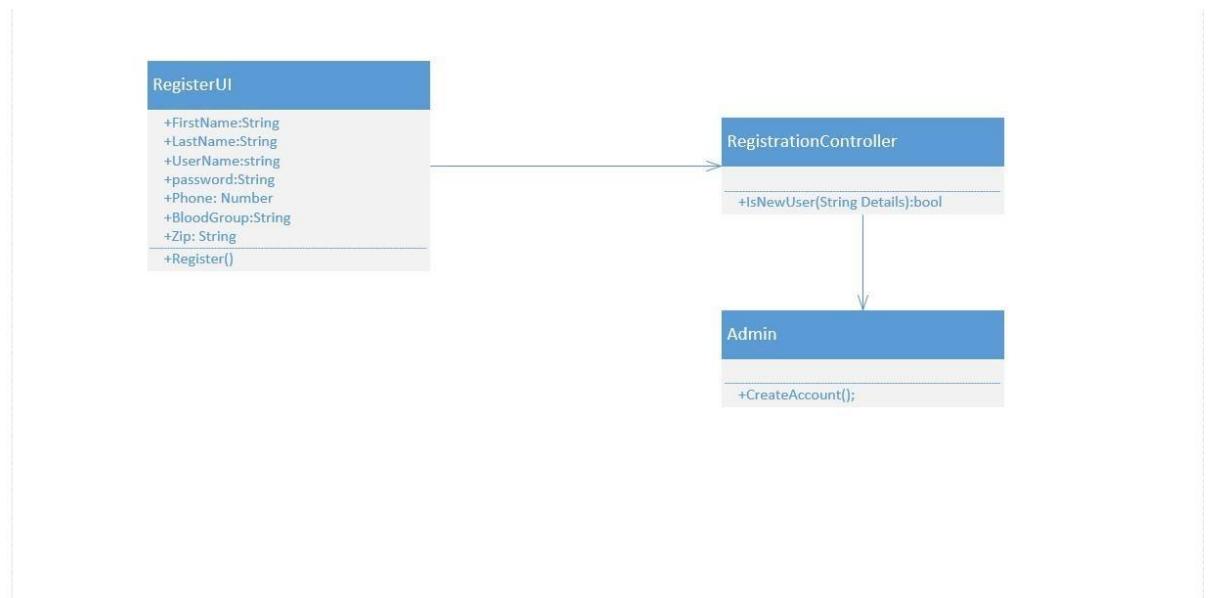
The login of the class diagram describes the flow of the system and process. The loginUI has the Username and password variables. Those variables are passing to the login controller through the GetCredentials. The system verifies the login credentials by using the authenticate function.



8.2. Registration Page

Similar to login page, the registrationUI takes all the necessary variables and pass them to the registration controller. The registration controller verifies the basic details and then the new account creates by the Admin.

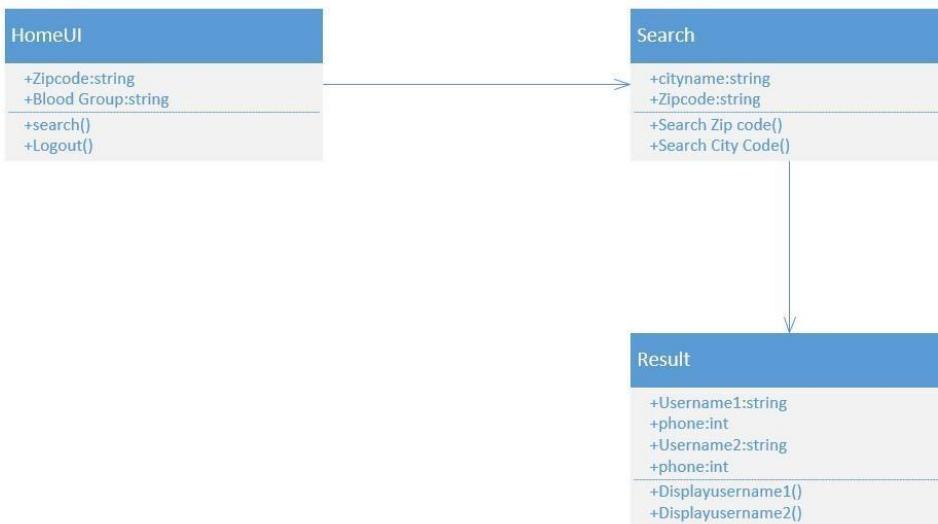
The below screenshot shows the class diagram of registration page.



8.3. Home Page

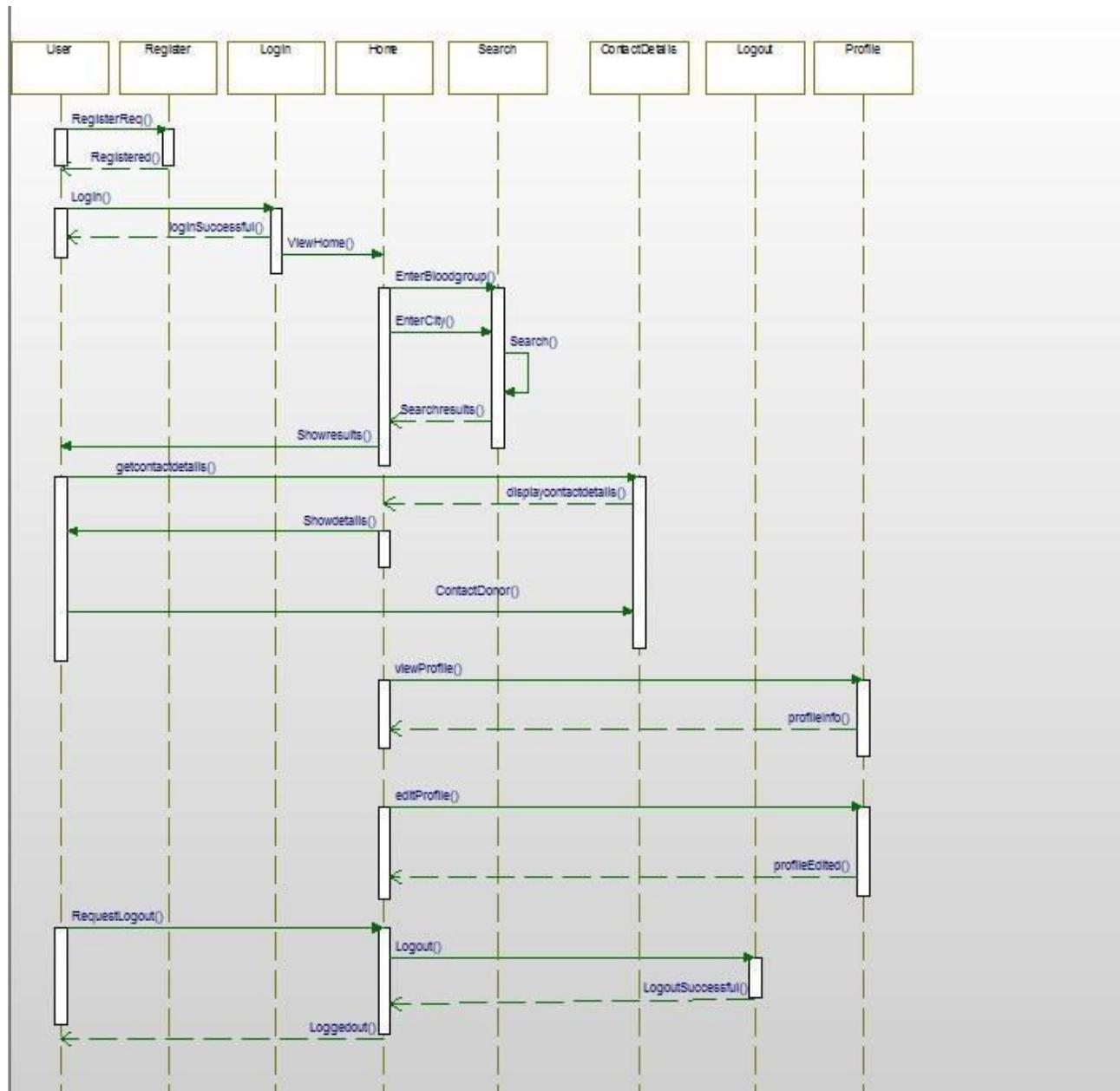
The home page donor search will take places by the search variables from the HomeUI. The search operation will happen based on the zip code and the blood type. Finally, the result will be displayed based on the blood type and zip code.

The below screenshot shows the class diagram of home page.

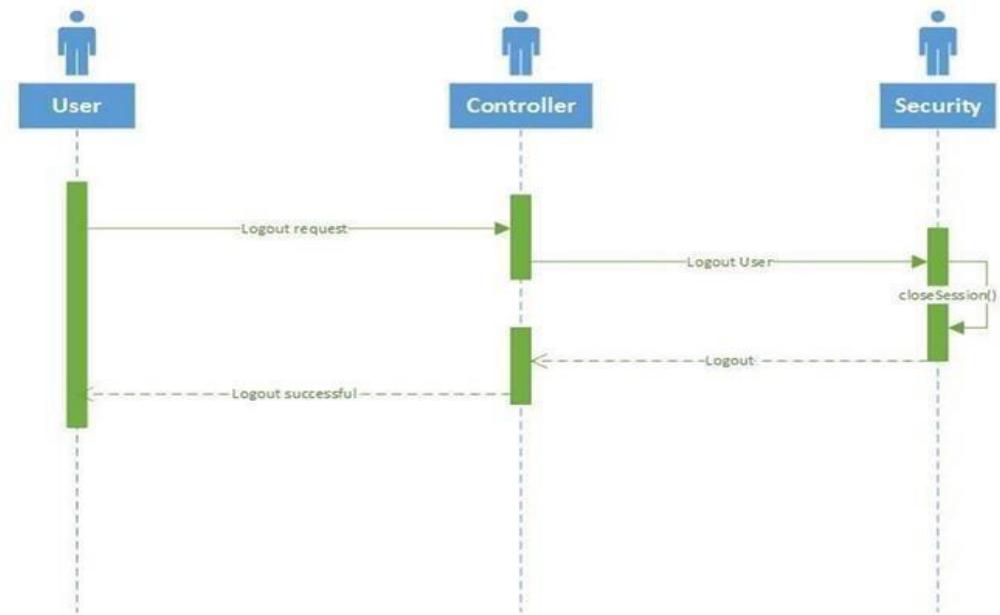


9.Sequence Diagram

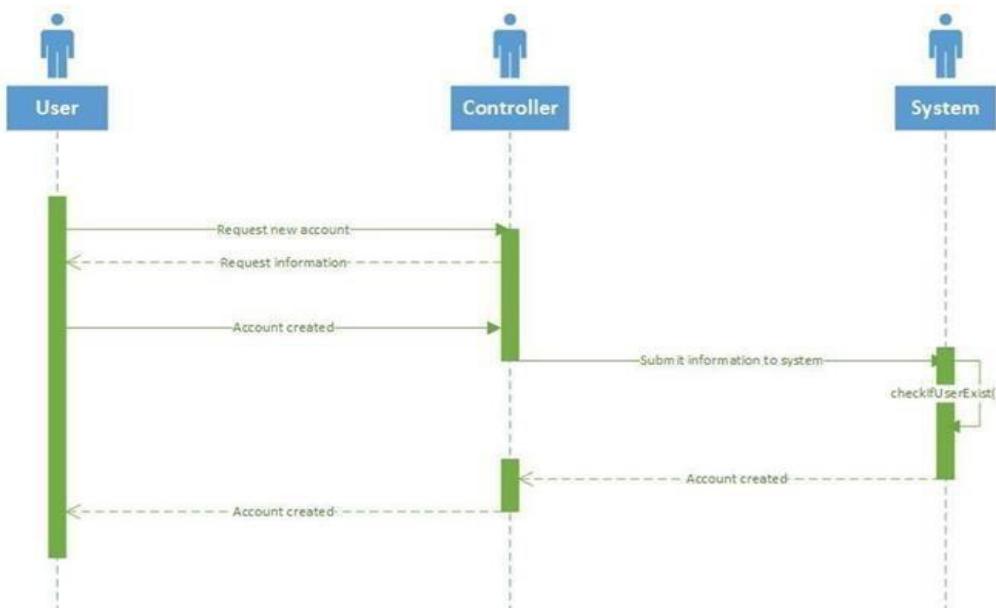
The sequence diagram describes the complete process of this project. As the screenshot shows, the registration process takes place first and the login function takes place. If the login is successful, then the view home function takes place. Sequentially, the enterbloodgroup(), enterzipcode() functions are taking place for the search. Once the search is successful, the searchresults () and showresults () functions are taking place to display the results. Finally, the logout () function takes place to logout the page.



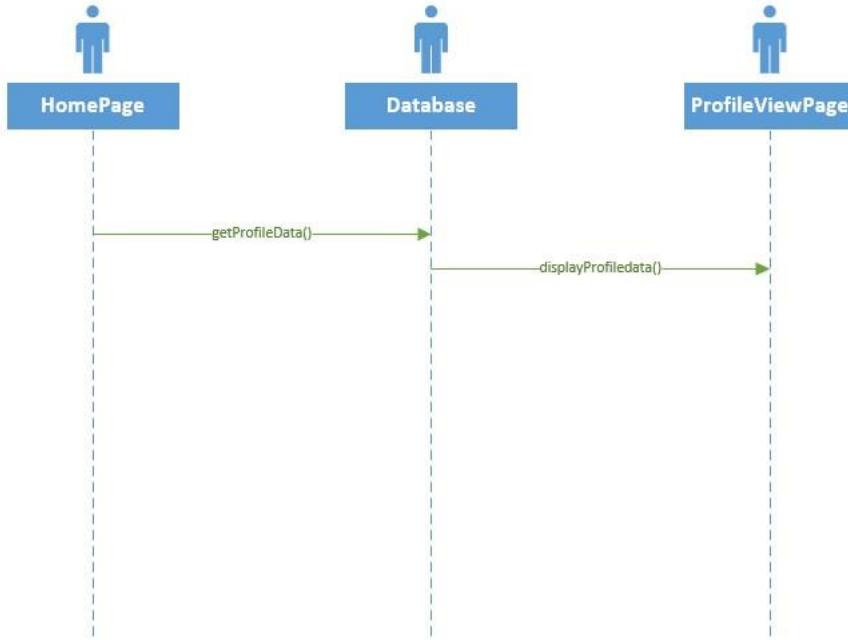
9.1. Login Sequence Page



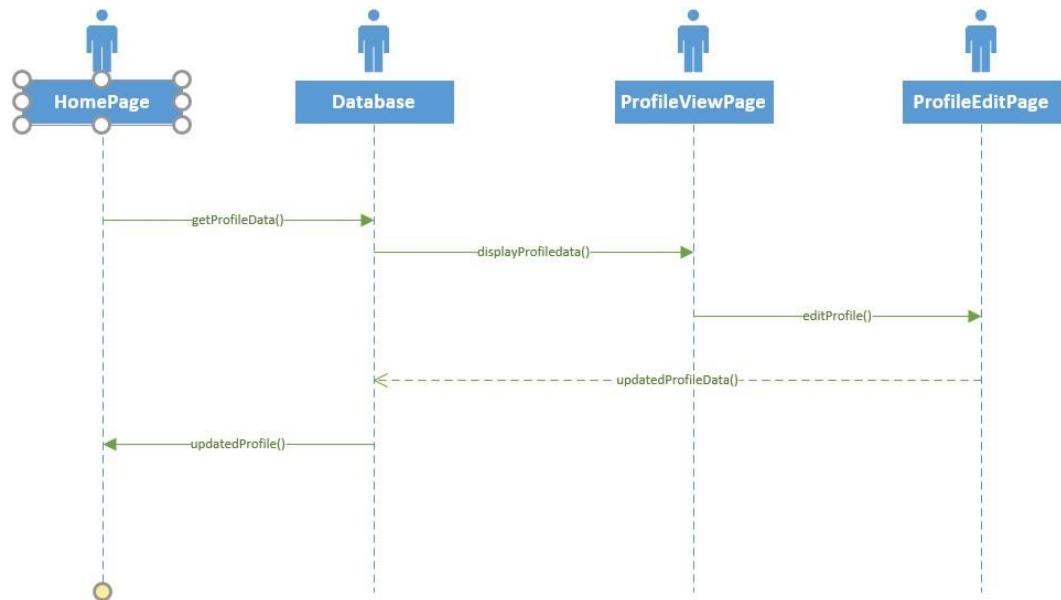
9.2. Registration Sequence Page:



9.3. Profile View Page Sequence Diagram



9.4. Edit Profile Page Sequence Diagram



10. Layouts:

10.1. Welcome Page

The below screenshot shows the Welcome page of the uml output. The user can **Login**, if they already have an account. Otherwise, they should create an account by clicking the **register** button.

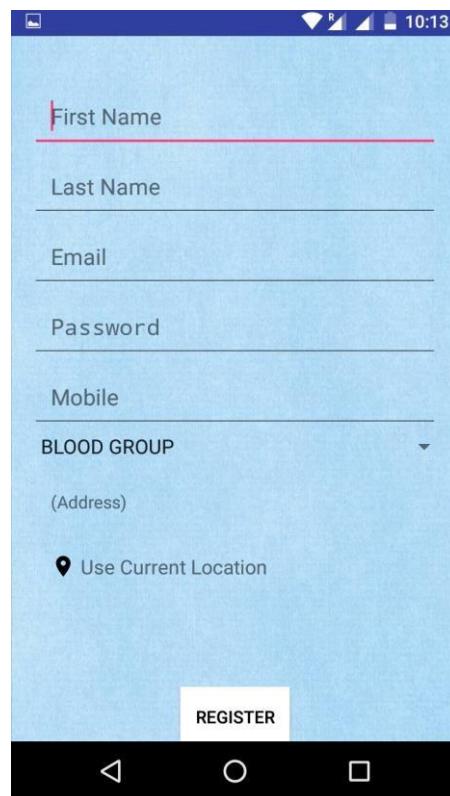


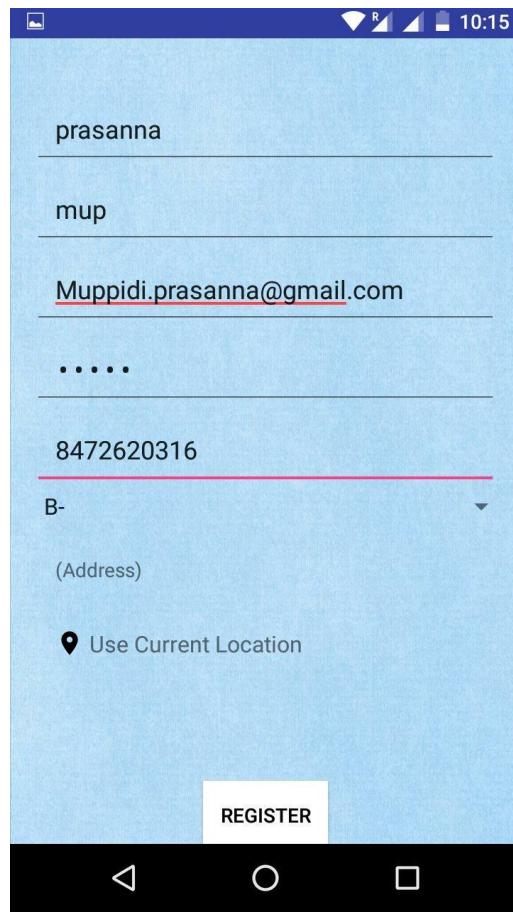
In future, we are planning to increment Social Login such as Facebook or Gmail in the center of the layout.

Note: If the user is already logged in, application doesn't ask him to login again and Welcome Page is not displayed. Session expires when the user clicks the logout button and will be redirected to the Login page.

10.2.Registration page:

The Registration page has all the basic information tabs. After giving and entering all the basic information, new account will be created for that specific user. The below screenshot shows the registration page.





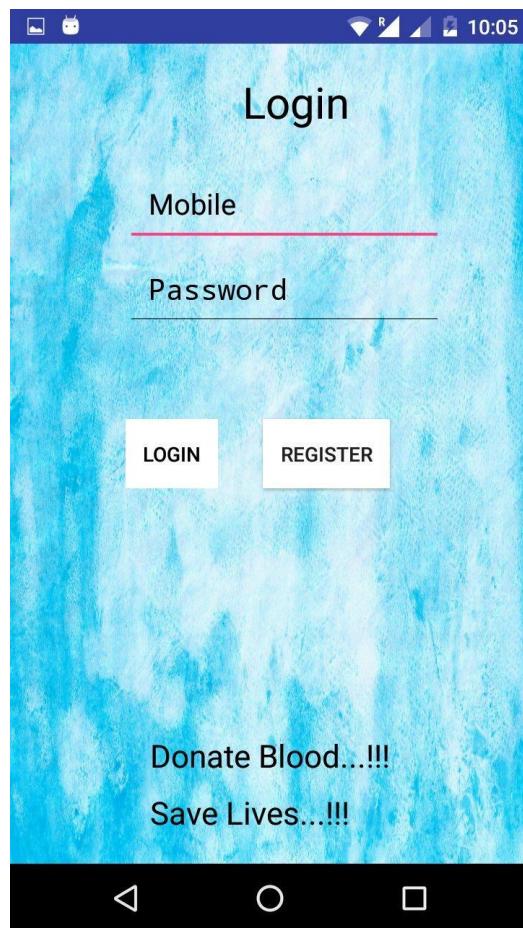
Address can be entered manually or by clicking the Use current location marker.

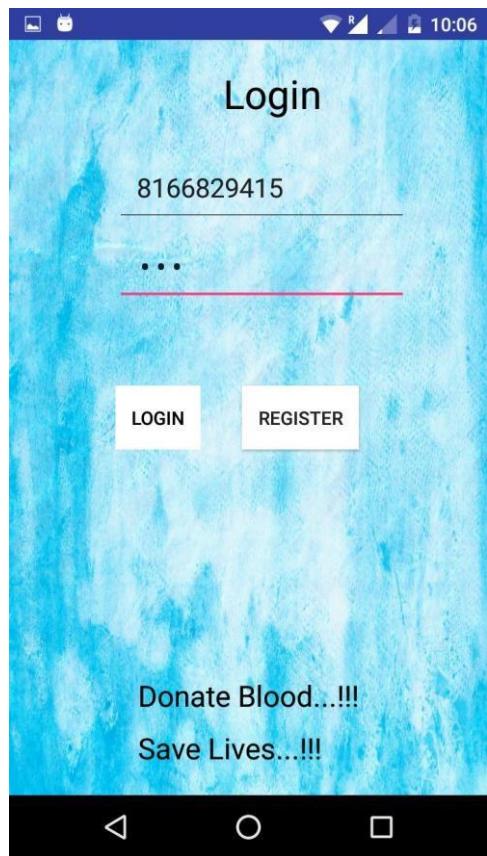
Validations have been added for Firstname, Lastname and Mobile Number.

Firstname and Lastname can't have numbers and Mobile Number can't have characters and it takes exactly 10 numbers.

Login Page:

- If the Username field is empty, an error message pops up telling the Username field is required.



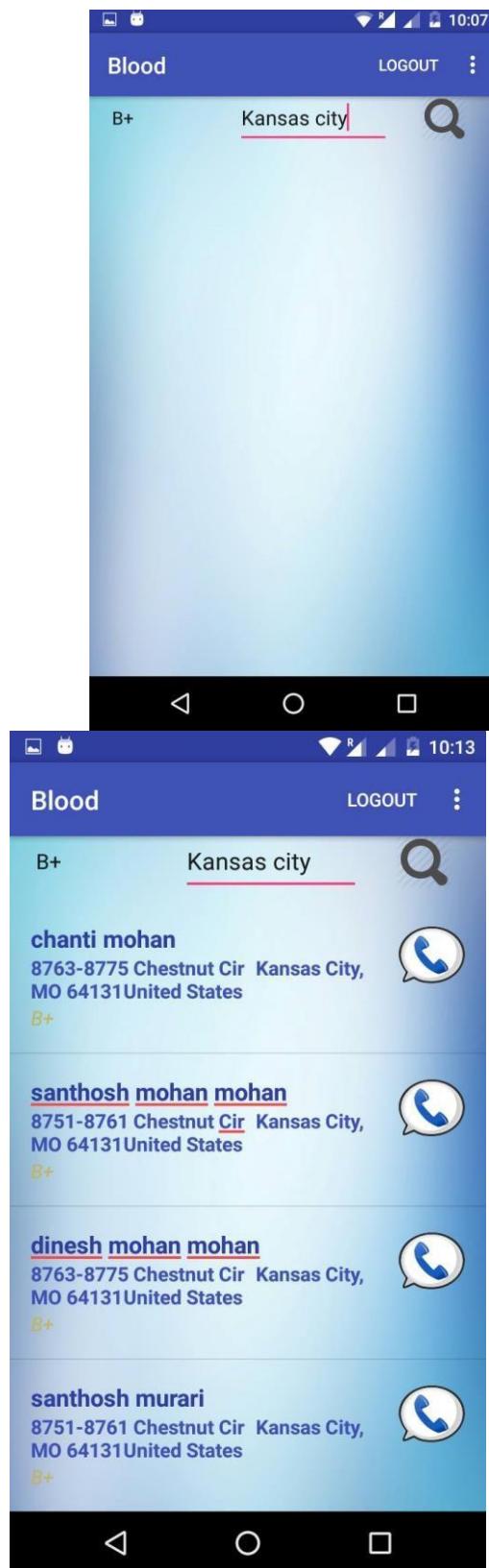


- If the user credentials are valid, user is redirected to the home page.

10.4. Home Page:

The home page provides the search results based on the search conditions. To get the search results, the search information has been entered in the corresponding search boxes. The below screenshot shows the home page of xml output.





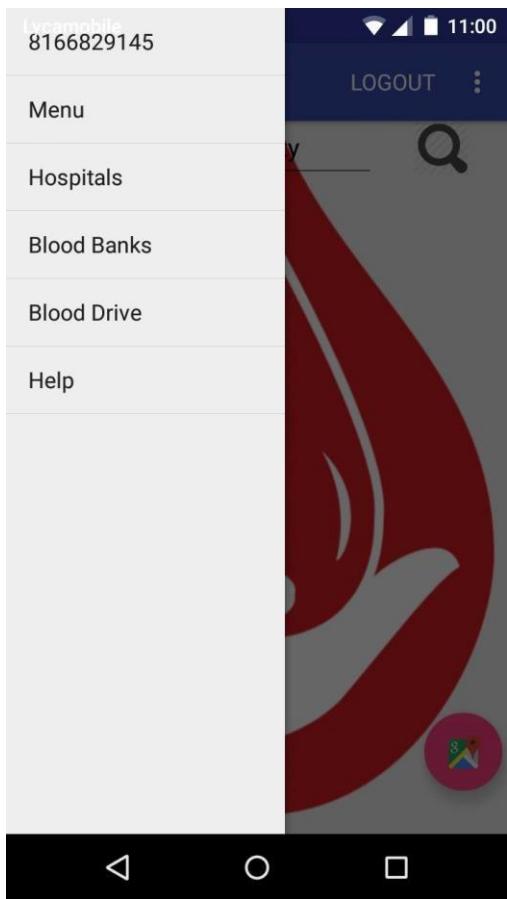
Profile Edit Page



[View Page](#)



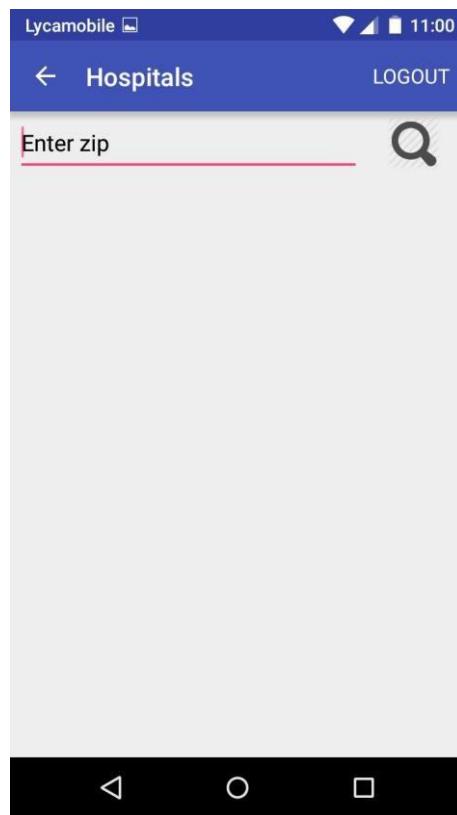
Left Navigation Bar:



Map View:



Search Nearby Hospitals:



11. Architecture/Design

UI Design:

We have the following UIs in our project.

- **Login Page:** This is an xml layout with two text fields and two buttons. User Details will be authenticated by using the text from those text fields. If the user is new he has to register, so by clicking register button he will be redirected to registration page. If the user is successfully logged in, he will be redirected to home page.
- **Registration Page:** This is an xml layout with 6 text fields, 1 radio button and register button. User has to give name, mobile number, username, zip code and more importantly blood group name. If all the required fields are validated once user clicks register button. If user has account already, an error message will be thrown with message as “account already exists, please go to login page”. If he is a new user, all the details will be stored in our database and he will be redirected to Home Page.
- **Home page:** This is the main page where user will be redirected after successful login. In this page, user can search donors who are available in the location he provided. He can view the contact details of the donors. We will extend this functionality to google maps where all the donors are seen in the maps. User can also update profile information. There is a logout button where user can close the session.

12. Database

We are planning to use MySQL database or SQLite database to store user details. If we use social logins, we get required details from the corresponding APIs.

We have only one table named “User”.

The table consists of following columns.

- First Name : Char ○
- Last Name : Char ○
- Mobile : Number ○ User
- Name :
- Char ○ Password :
- Char ○ Address :
- Char ○ Blood group :
- Char ○ Created at:
- DateTime ○ Updated at :
- DateTime

The below is the screenshot for the Firebase database.

It's a NoSQL Database where data is stored as JSON objects.

The screenshot shows the Firebase Realtime Database dashboard. On the left, there is a sidebar with icons for Data, Security & Rules, Simulator, Analytics, Login & Auth, Hosting, and Secrets. The main area displays a tree view of the database structure under a project named "VIEWING BLOODMANAGEMENT". The root node "bloodmanagement" has a child node "Users". Inside "Users", there are four child nodes: "chanti", "chanti1913", "pras", and "raviteja83". The "chanti1913" node contains the following data:

```
address: "8763-8775 Chestnut Cir\nKansas City, MO 64131\nUnit 100"
blood_group: "A+"
first_name: "chanti"
last_name: "mohan"
mobile: "949"
password: "msm"
user_name: "chanti1913"
```

A legend on the right indicates changes: yellow for Changed, green for Added, red for Deleted, and blue for Moved. The "user_name" field for the "chanti1913" node is highlighted with a red border and a delete icon.

13. Four Different Increments:

13.1 Increment 1-- Requirement Gathering and Designing the application

1. Research on importance of blood donation management, requirements for the development.
2. Setup android studio, zenhub and Github for every team member.
3. Design class diagrams.
4. Design sequence diagrams.
5. Design state diagrams.
6. Design WireFrames.
7. Design basic UI layouts (Login, Registration, Home)

13.2 Increment 2—Coding/Testing

- 1) Implement Login/logout /Registration/ Registration validation and functionality.
- 2) Implement home page.
- 3) Setup local databases.
- 4) Test Login/logout UI/Registration/Registration validation.
- 5) Test sample user data.

13.3 Increment 3 - Coding/Testing 1)

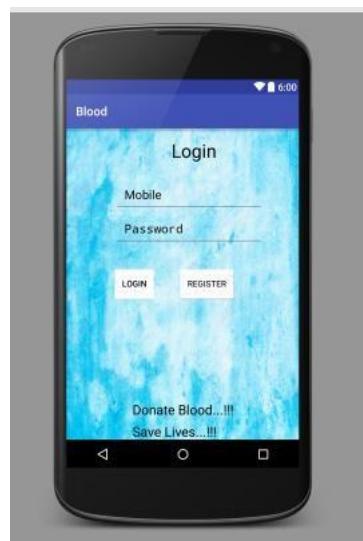
- 1) Implement search functionality
- 2) Implement View and Edit Profile Functionality
- 3) Test search functionality
- 4) Test Edit Profile UI and functionality
- 5) Implementation of field validations.

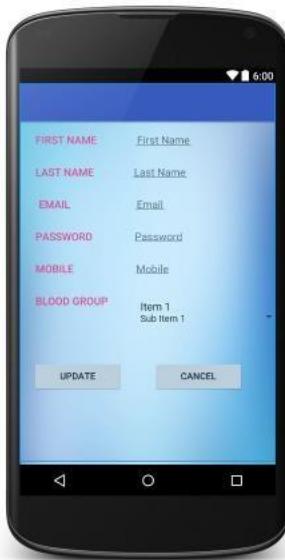
13.4 Increment 4 -- Refine GUI

- 1) Refine GUI for Login/Registration.
- 2) Refine GUI for Home Page
- 3) Refine GUI for Profile Page
- 4) Refine GUI for Welcome Page.

14. TESTING:

14.1 Emulator testing:





15. Project Timelines, Members, Task Responsibility

Member and Responsibilities:

Artifacts	Members: Santhosh, Anudeep, Prasanna, FathimaJ
Research	All
Project Plan	Santhosh mohan
UML Diagrams	Prasanna, Anudeep
Modeling Database	Anudeep, Prasanna
Initial Mockup	FathimaJ
Reports	All
Development	Santhosh Mohan, Prasanna, Anudeep ,Fathima J
Testing	Anudeep, Fathima J
Maintenance	Santhosh Mohan

Project Timelines:

Increments	Tasks
Increment 1	Initial setup of environments, uml diagrams and code (Login UI, Registration UI, Home UI)
Increment 2	Code (Login, Register, Home Java code functionalities basic)
Increment 3	Code + Test (Profile functionality, testing) and use case execution
Increment 4	Map functionality, Deployment to play store, final report.

16. Bibliography

- www.github.com
- www.developer.android.com
- www.developer.google.com
- <http://www.redcrossblood.org/learn-about-blood/blood-types>
- https://en.wikipedia.org/wiki/Blood_type
- <https://www.firebaseio.com/>
- <https://www.firebaseio.com/docs/android/guide/>
- <https://www.firebaseio.com/blog/>