

# **CS5551 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:

- 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

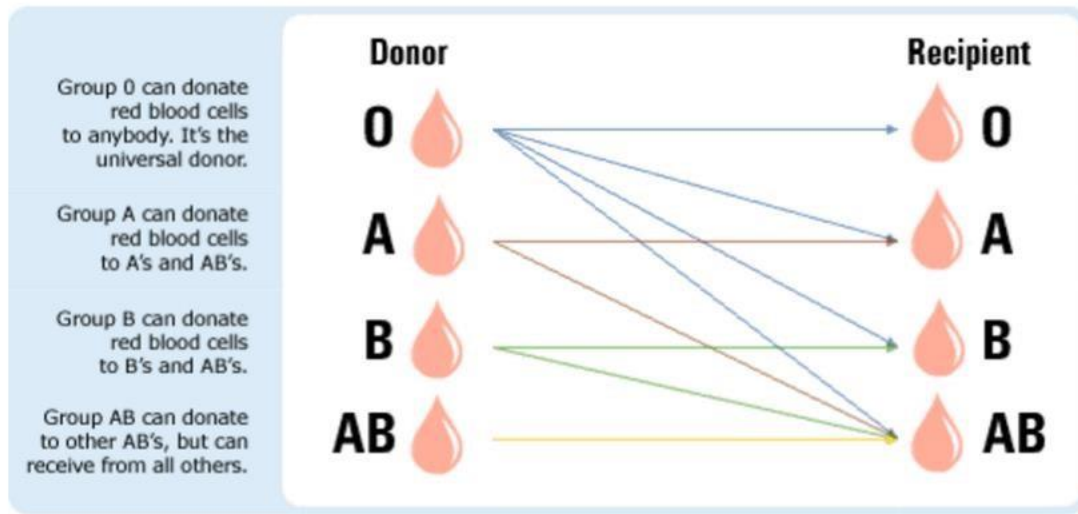
### System Designs:

- 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
- 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.



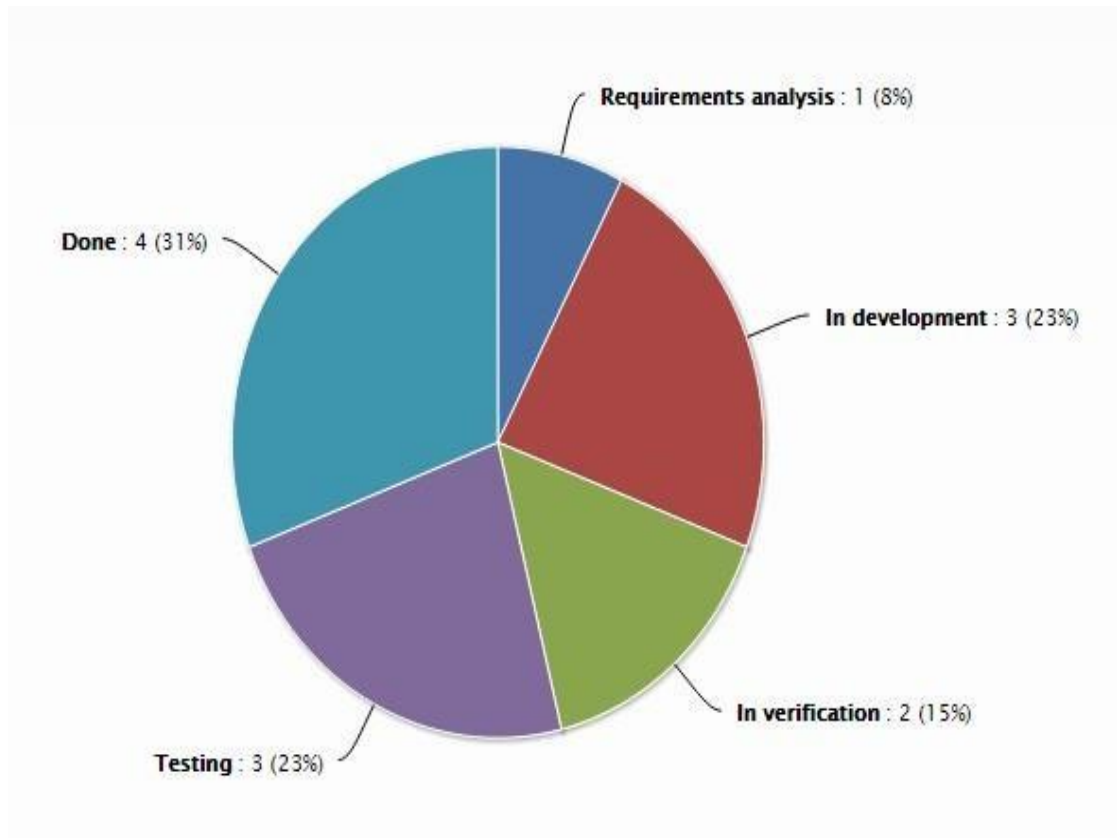
### Explanation:

- ❑ 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.

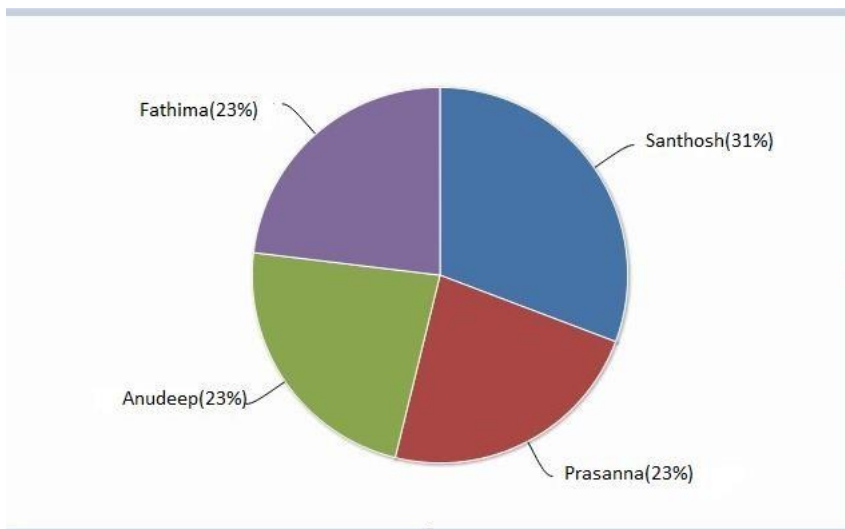
With the above observations, whenever a user wants a particular type of blood, he gets all the blood types who can donate blood to that required type not just the same blood type which the user entered. According to the above algorithm, the user has a bigger platform than just getting one single blood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

The wireframe diagram illustrates a login page layout. It features a header section labeled "Login Page". Below the header, there are two input fields: "User Name" and "Password". At the bottom of the form, there are two buttons: "Login" and "Register".

```
graph TD; subgraph "Login Page"; direction TB; subgraph Form; direction TB; Username[User Name]; Password[Password]; end; end; Login[Login]; Register[Register];
```

## 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 shows a registration page with a title bar at the top. Below the title bar, there is a large container box. Inside this box, the following elements are arranged vertically: a text input field labeled 'FIRST NAME', a text input field labeled 'LAST NAME', a text input field labeled 'EMAIL', a text input field labeled 'PASSWORD', a text input field labeled 'MOBILE', a text input field labeled 'BLOOD GROUP' followed by a small dropdown arrow icon, a text input field labeled 'CURRENT LOCATION' preceded by a location pin icon, and a 'Register' button at the bottom.

Registration page

FIRST NAME

LAST NAME

EMAIL

PASSWORD

MOBILE

BLOOD GROUP

CURRENT LOCATION

Register



### 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

Blood Group

▼

Enter city name

Item 1  
Details

Item 2  
Details

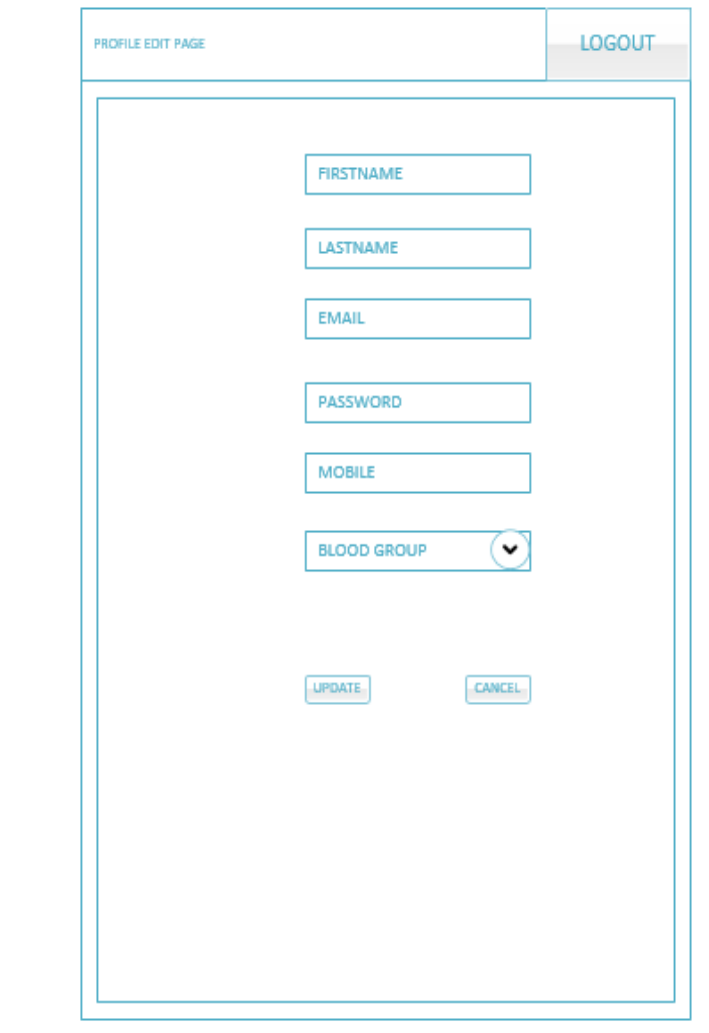
Item 3  
Details

Item 4  
Details

Item 5  
Details

#### 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.

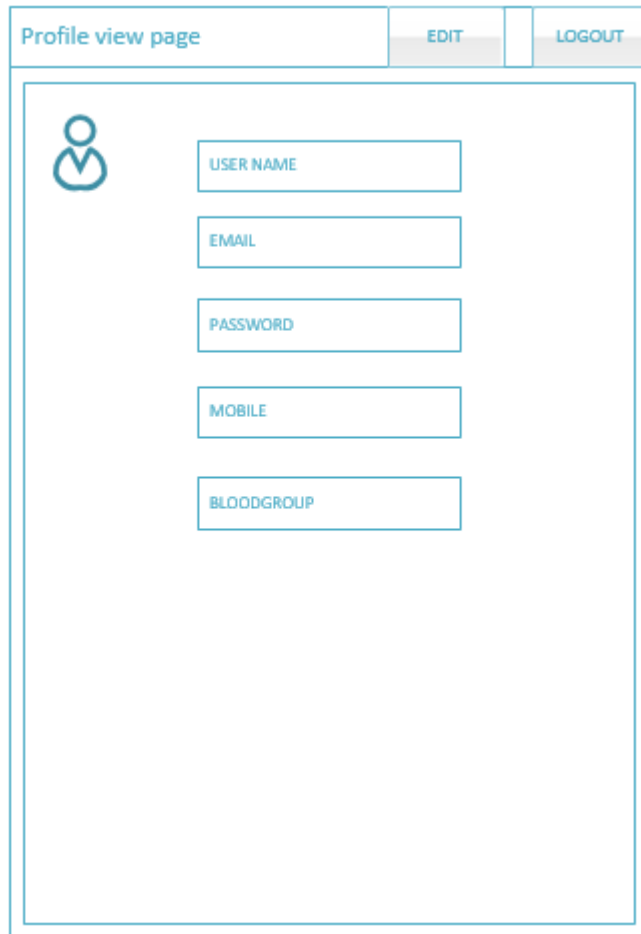


The wireframe illustrates a 'PROFILE EDIT PAGE'. At the top, there is a header bar with 'PROFILE EDIT PAGE' on the left and a 'LOGOUT' button on the right. The main content area is enclosed in a large rectangular frame. Inside this frame, there is a vertical stack of input fields: 'FIRSTNAME', 'LASTNAME', 'EMAIL', 'PASSWORD', 'MOBILE', and 'BLOOD GROUP'. The 'BLOOD GROUP' field is a dropdown menu, indicated by a downward arrow icon. At the bottom of the form, there are two buttons: 'UPDATE' and 'CANCEL'.

PROFILE EDIT PAGE		LOGOUT
<div>FIRSTNAME</div> <div>LASTNAME</div> <div>EMAIL</div> <div>PASSWORD</div> <div>MOBILE</div> <div>BLOOD GROUP ▼</div> <div>UPDATE CANCEL</div>		

### 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.

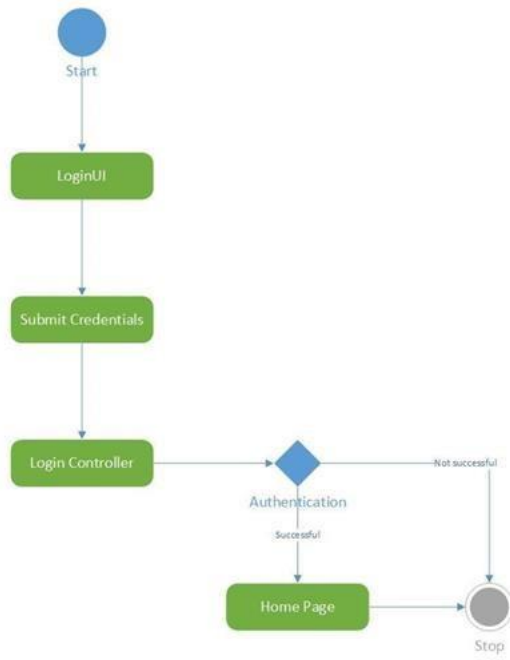


The screenshot displays a web interface for a 'Profile view page'. At the top, there is a header bar with the title 'Profile view page' on the left, and two buttons, 'EDIT' and 'LOGOUT', on the right. Below the header, the main content area is enclosed in a large rectangular frame. On the left side of this frame is a circular icon representing a user profile. To the right of the icon, there are five vertically stacked text input fields, each with a label: 'USER NAME', 'EMAIL', 'PASSWORD', 'MOBILE', and 'BLOODGROUP'.

## 8.State Diagrams

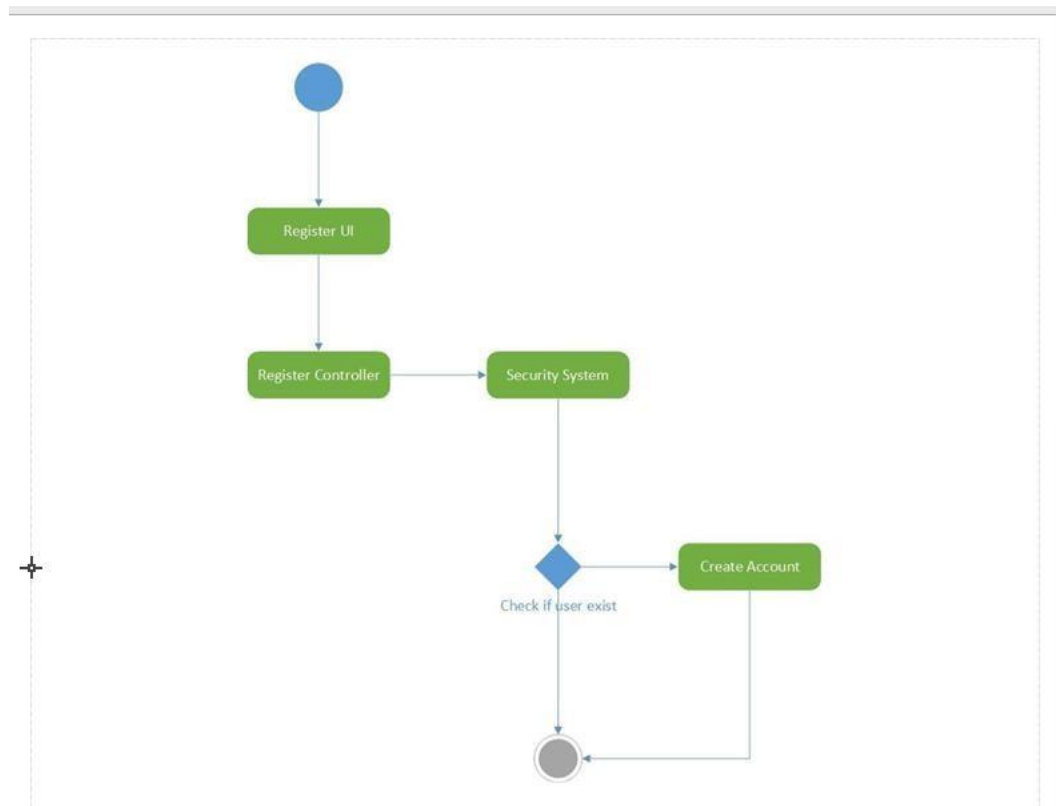
### 8.1. Login Page

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.



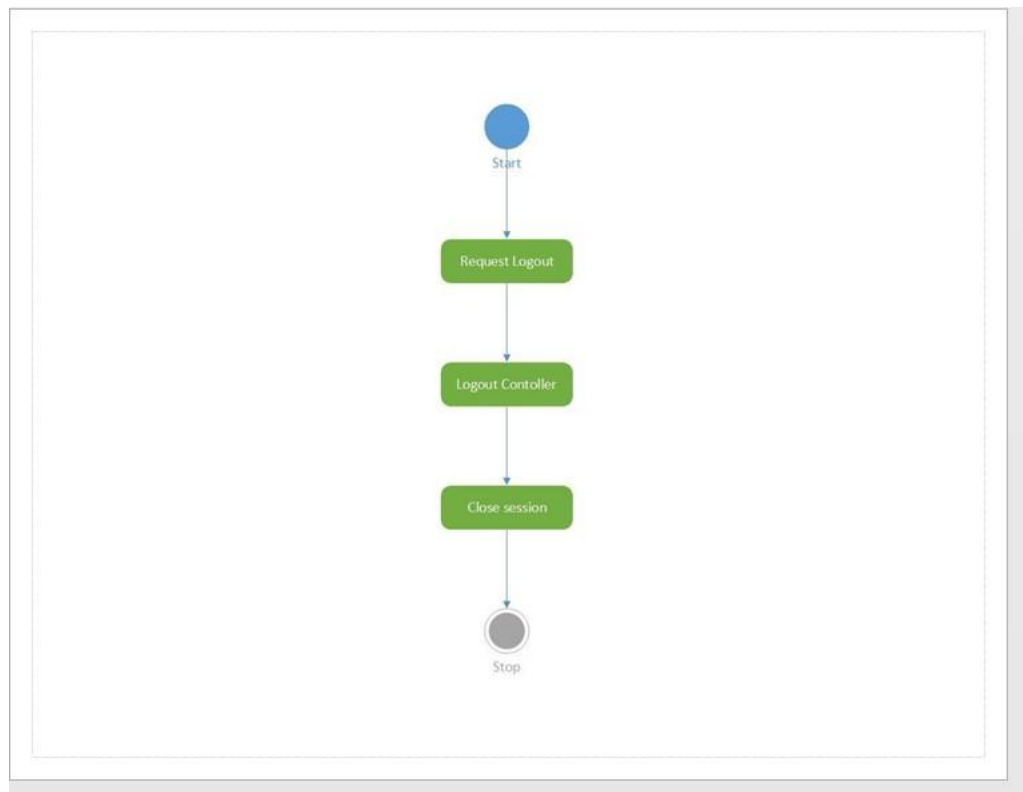
## 8.2. Registration Page

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

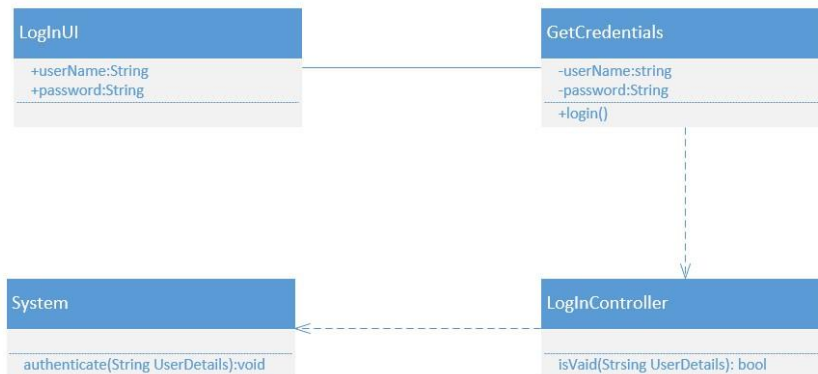
The home page displays the blood donor information. Once the user gets the information about the donor, they can logout the home page. The state diagram shows the logout steps clearly in the below screenshot. The request logout passes the request to the logout controller. The logout controller closes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



## 9. Class diagrams

### 9.1. Login Page

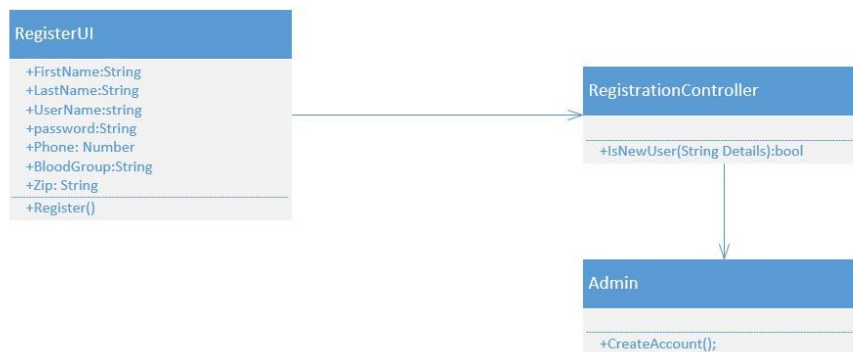
The login of the class diagram describes the flow of the system and process. The loginUI has the Username and password variables. Those variables are passing to the login controller through the GetCredentials. The system verifies the login credentials by using the authenticate function.



## 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

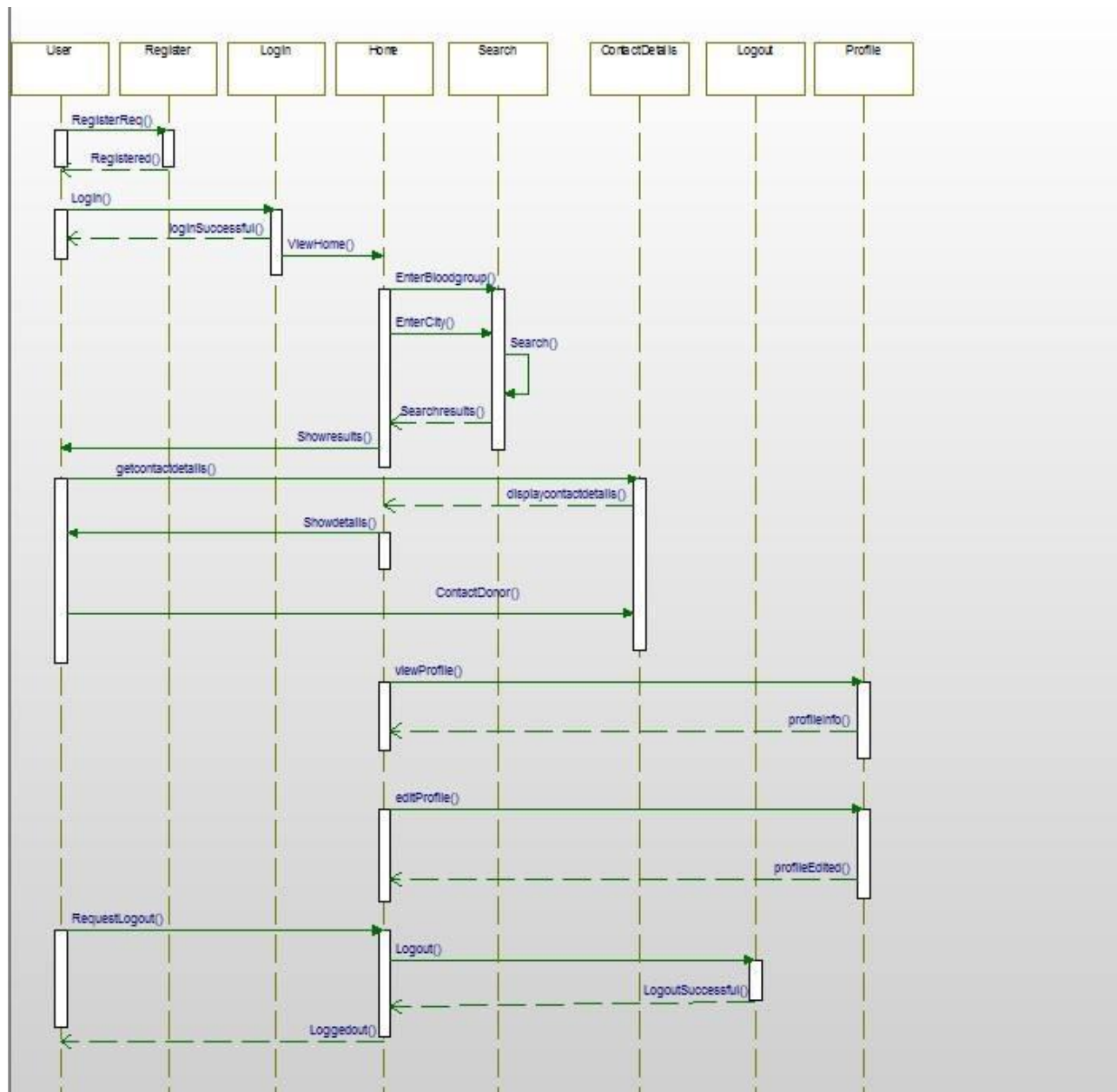
The home page donor search will take places by the search variables from the HomeUI. The search operation will happen based on the zip code and the blood type. Finally, the result will be displayed based on the blood type and zip code.

The below screenshot shows the class diagram of home page.

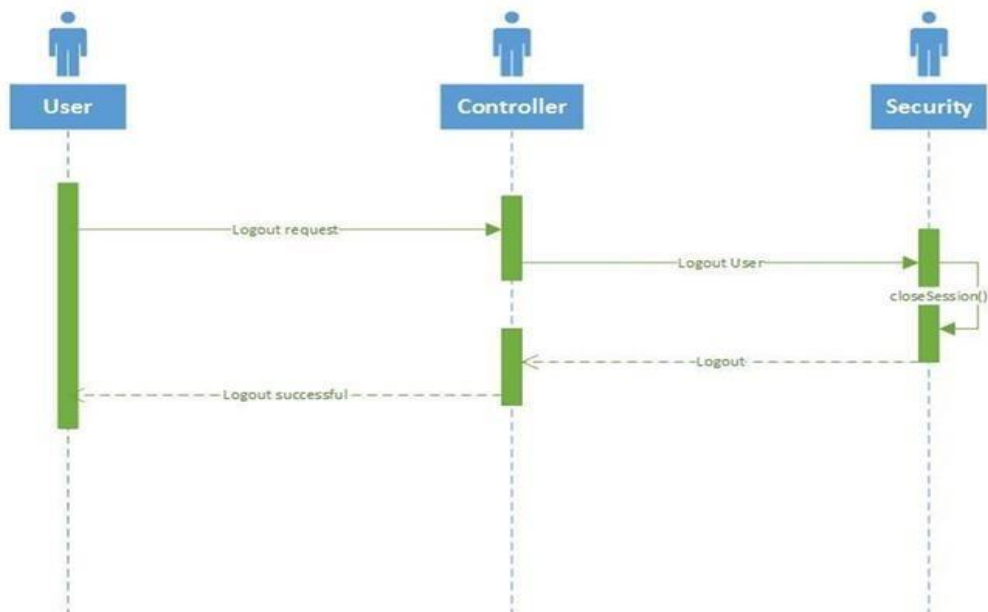


## 9. Sequence Diagram

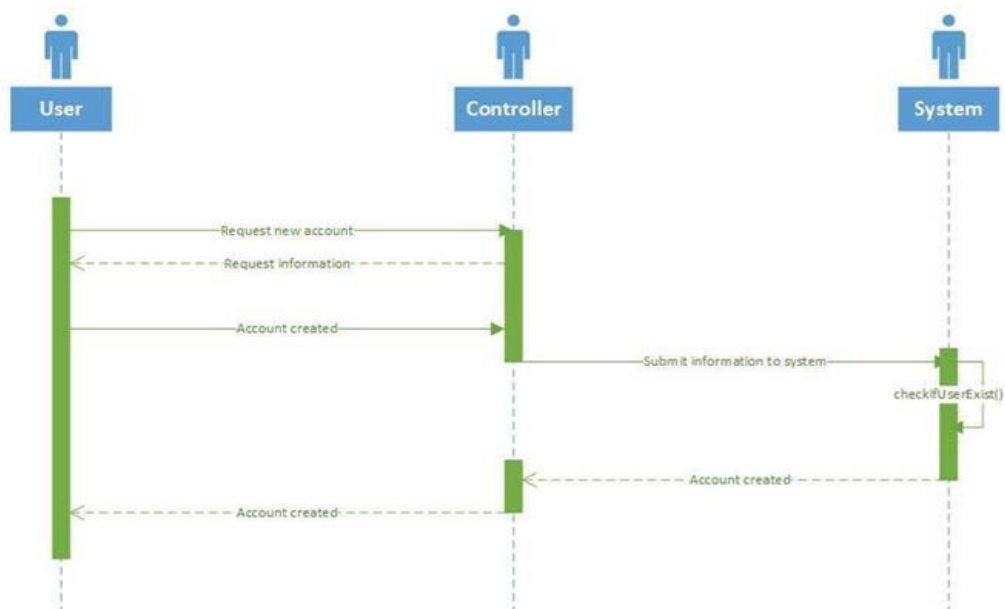
The sequence diagram describes the complete process of this project. As the screenshot shows, the registration process take place first and the login function takes place. If the login is successful, then the view home function takes place. Sequentially, the enterbloodgroup(), enterzipcode() functions 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.



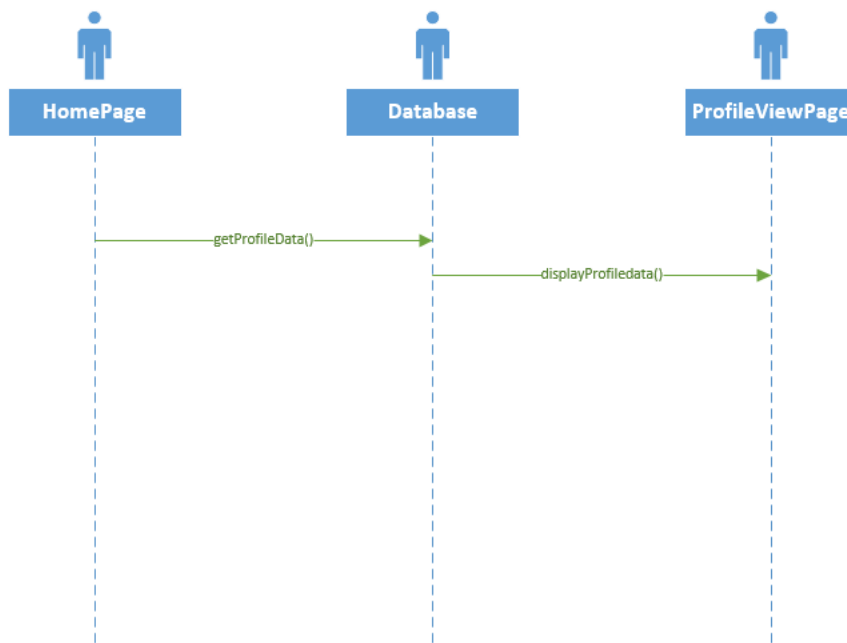
### 9.1. Login Sequence Page



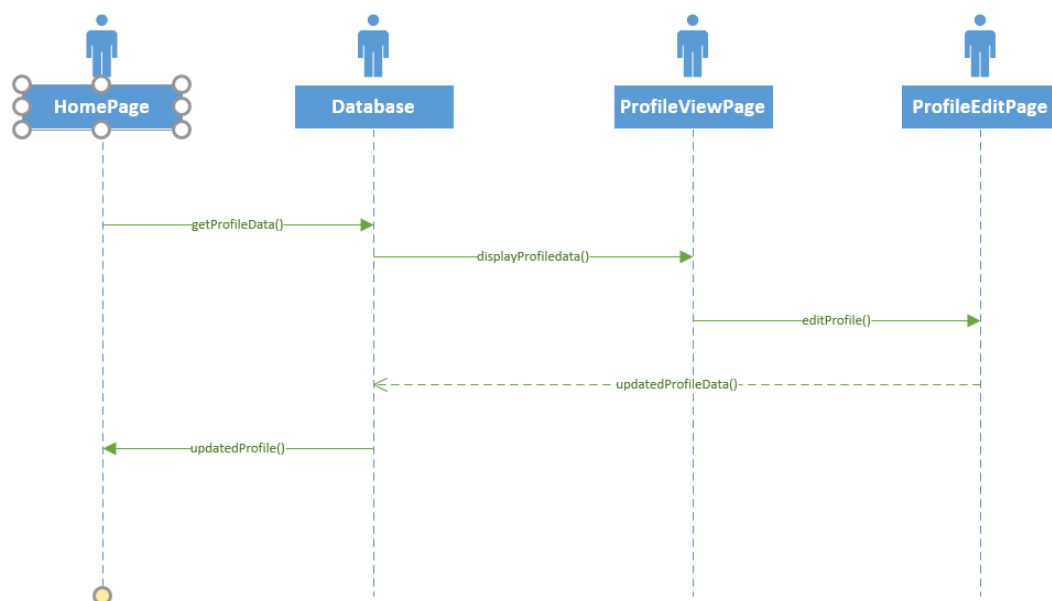
### 9.2. Registration Sequence Page:



### 9.3. Profile View Page Sequence Diagram



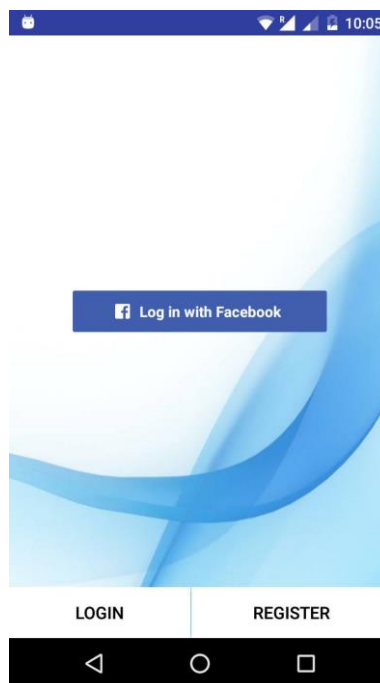
### 9.4. Edit Profile Page Sequence Diagram



## 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.

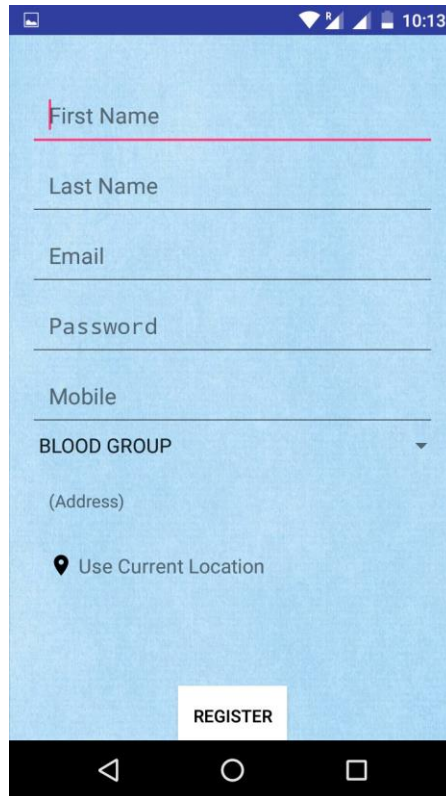


In future, we are planning to increment Social Login such as Facebook or Gmail in the center of the layout.

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.



The screenshot displays a mobile application's registration page. The page has a light blue background. At the top, there is a dark blue status bar with icons for signal, Wi-Fi, and battery, and the time 10:13. The form consists of several input fields with labels: 'First Name' (with a red vertical line on the left), 'Last Name', 'Email', 'Password', 'Mobile', 'BLOOD GROUP' (with a dropdown arrow), and '(Address)'. Below the address field is a location pin icon followed by the text 'Use Current Location'. At the bottom of the form is a white button with the text 'REGISTER'. The bottom of the screen shows a black navigation bar with three white icons: a back arrow, a circle, and a square.

prasanna

mup

Muppidi.prasanna@gmail.com

.....

8472620316

B-

(Address)

📍 Use Current Location

REGISTER

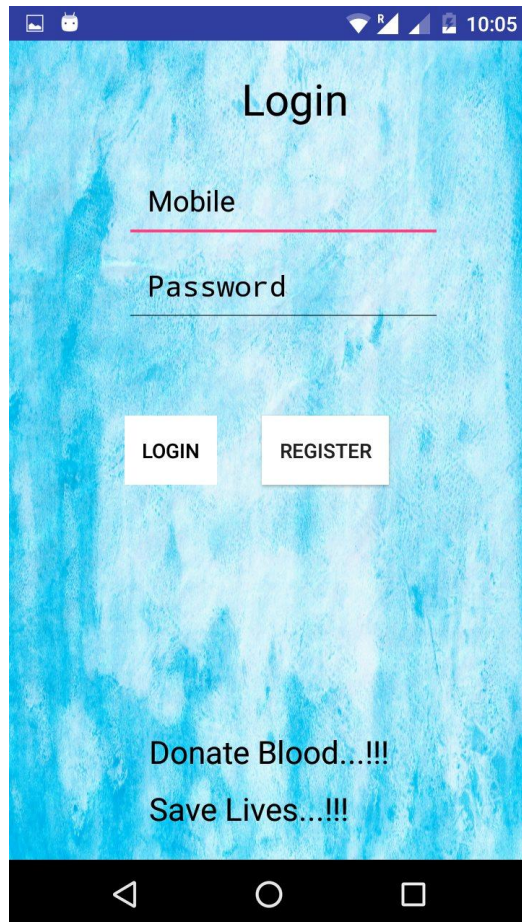
Address can be entered manually or by clicking the Use current location marker.

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.



Login

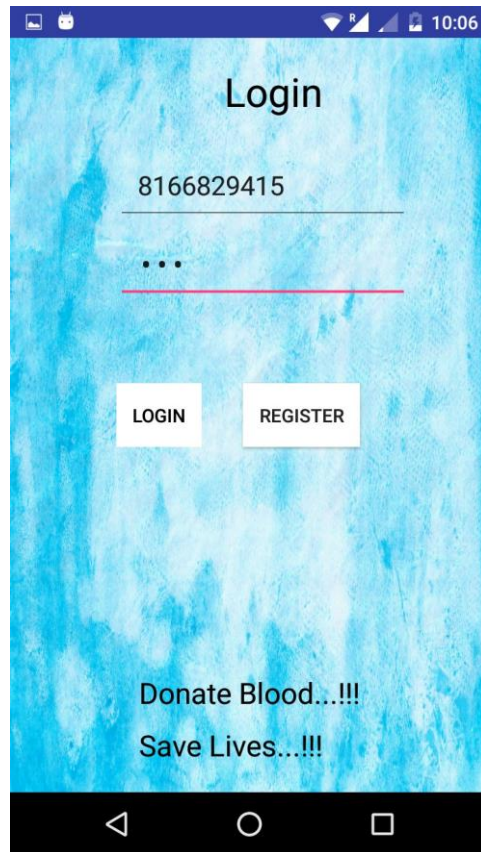
Mobile

Password

LOGIN REGISTER

Donate Blood...!!!  
Save Lives...!!!



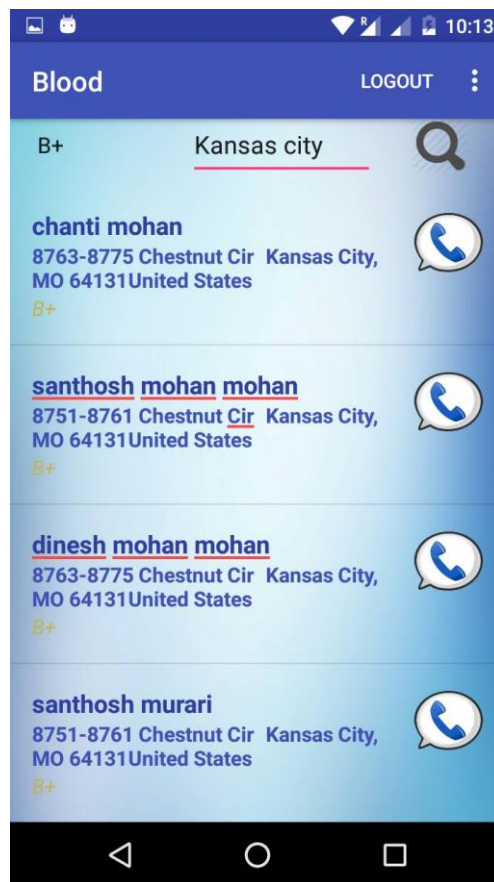


- If the user credentials are valid, user is redirected to the home page.

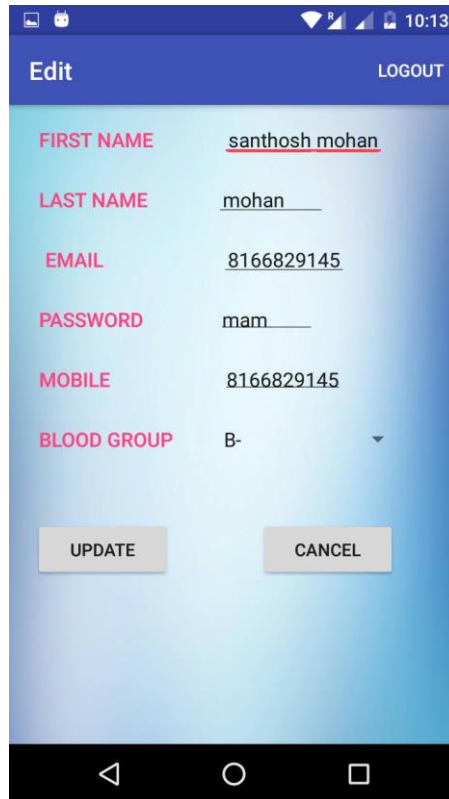
#### 10.4. Home Page:

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



A screenshot of a mobile application's "Edit" profile page. The page has a blue header with the word "Edit" on the left and "LOGOUT" on the right. Below the header, there are several form fields with labels in red: "FIRST NAME" (value: santhosh mohan), "LAST NAME" (value: mohan), "EMAIL" (value: 8166829145), "PASSWORD" (value: mam), "MOBILE" (value: 8166829145), and "BLOOD GROUP" (value: B- with a dropdown arrow). At the bottom of the form are two buttons: "UPDATE" and "CANCEL". The background is a light blue gradient. The Android navigation bar is visible at the very bottom.

FIRST NAME	<u>santhosh mohan</u>
LAST NAME	<u>mohan</u>
EMAIL	<u>8166829145</u>
PASSWORD	<u>mam</u>
MOBILE	<u>8166829145</u>
BLOOD GROUP	<u>B-</u>


UPDATE CANCEL


## View Page




A screenshot of a mobile application's "Profile" view page. The page has a blue header with a back arrow on the left, the word "Profile" in the center, and "EDIT" and "LOGOUT" on the right. Below the header, there is a profile card. The card features a circular profile picture placeholder (a blue silhouette) and the name "santhosh mohan" followed by "mohan". Below the name are four rows of contact information, each with an icon on the left and a value on the right: an envelope icon for the email "8166829145", a padlock icon for the name "MAM", a telephone icon for the mobile number "8166829145", and a red blood drop icon for the blood group "B+". The background is a light blue gradient with abstract white curves. The Android navigation bar is visible at the very bottom.


Profile EDIT LOGOUT

 santhosh mohan  
mohan

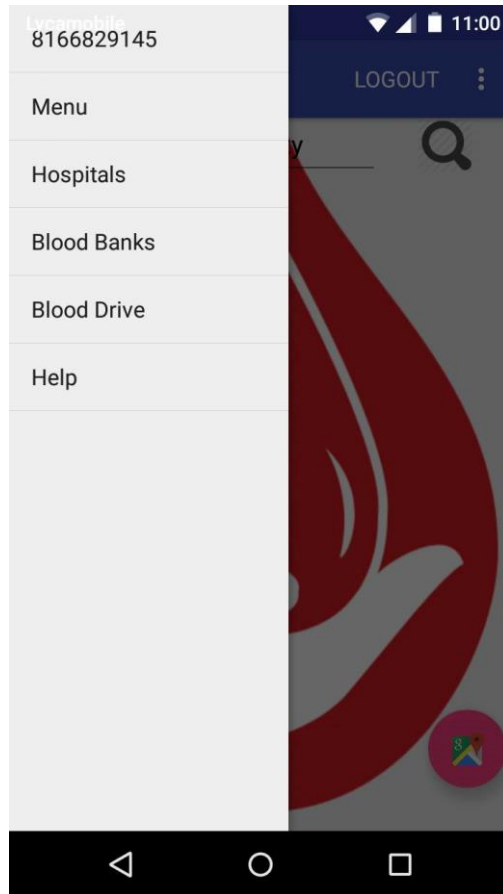
 8166829145

 MAM

 8166829145

 B+

### Left Navigation Bar:



### Map View:



### Search Nearby Hospitals:

Enter zip



## 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.

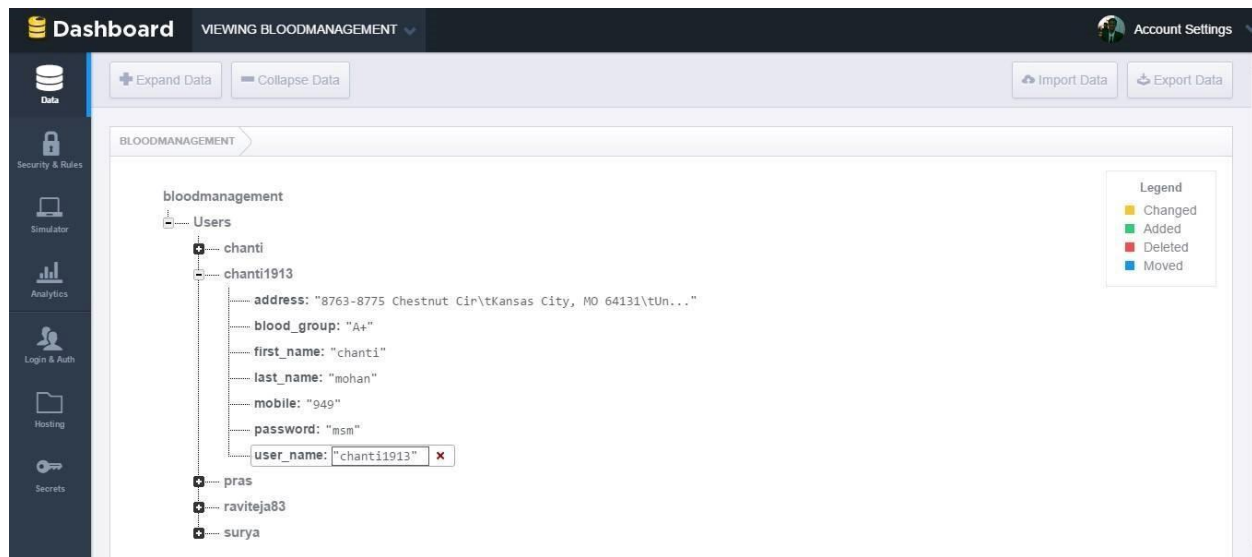
We have only one table named “User”.

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.



### 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.

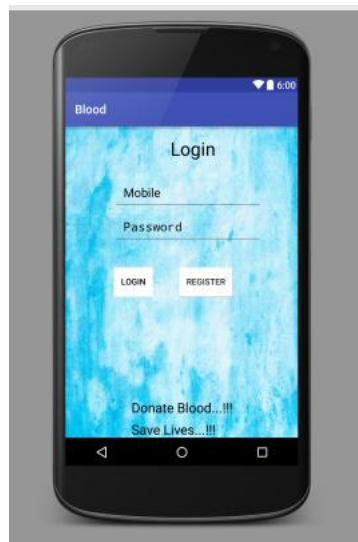
#### 13.4 Increment 4 -- Refine GUI

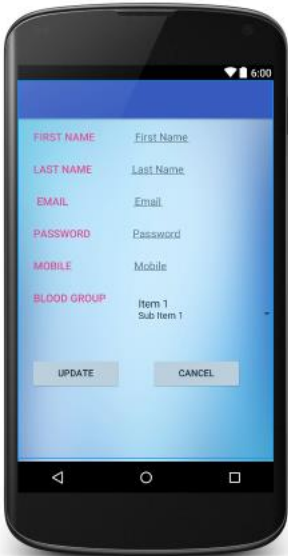
- 1) Refine GUI for Login/Registration.
- 2) Refine GUI for Home Page
- 3) Refine GUI for Profile Page
- 4) Refine GUI for Welcome Page.



## 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](http://www.github.com)
- [www.developer.android.com](http://www.developer.android.com)
- [www.developer.google.com](http://www.developer.google.com)
- <http://www.redcrossblood.org/learn-about-blood/blood-types>
- [https://en.wikipedia.org/wiki/Blood\\_type](https://en.wikipedia.org/wiki/Blood_type)
- <https://www.firebase.com/>
- <https://www.firebase.com/docs/android/guide/>
- <https://www.firebase.com/blog/>