NYU POLYTECHNIC SCHOOL OF ENGINEERING

## Computer Science and Engineering

### CS 9033: Mobile Application Development

# B-LIFE

## Final Design Document

## Version 2.0

## Team Members

| Name | Poly Id | Email |
|---|---|---|
| Pranav Bhoraskar | N14624696 | pb1460@nyu.edu |
| Chaitanya Manyam | N10039212 | chaitanya.manyam@nyu.edu |
| Yeshwant Dasari | N11336745 | ykd202@nyu.edu |
| Taranjeet Singh | N13487897 | tts260@nyu.edu |

## Revision History

| Date | Version | Description |
|---|---|---|
| 03/18/2015 | 1.0 | Initial Design Document |
| 05/18/2015 | 2.0 | Final Design Document |

# Table of Contents

# 1. INTRODUCTION

## 1.1 Purpose

Android is the most used and highly power-packed operating system, which is a dominant force to reckon with in the contemporary tech world. In the current world, 46.7 % of phones are android. 60% of Android users are under the age of 34.

In United States for every 2 seconds someone needs blood.  Our application connects blood donors and receivers. Currently more than 41,000 blood donations are needed every day a total of 30 million blood components are transfused each year. In order to support this requirement, we would be requiring a lot of blood and to accommodate this, our application could be a handy one. Cancer patients require blood from 8 people every week and a serious car accident would require amount of blood from 50 people. Our application not only connects receivers but also hospitals that are in dire need of blood.

There are several applications on the Google play, which support similar functionalities but are missing the primary functionality our application supports i.e. connecting donors and receivers by text, calls and sharing pictures between them.

B-Life is a unique application combining all the social elements like email, text, sending request and accepting them onto a social cause - "Blood Donation". Users can access their profiles, send requests to donors, keep track of requests sent and received and finally edit profiles. Users have the option to search for nearby hospitals or donors during the blood crisis period. We keep track of their donation history and give health tips to our users so that they can recover from a blood donation faster than possible. We encourage our users to donate blood regularly and make it a habit. Our application lists all the basic information regarding blood donation, medical facts and answers to some of the FAQ's.

Our application notifies the user whenever a blood drive is taking places nearer to him. Users have the option to edit privacy options such that their profile will be visible to their targeted or interested audience. B-life will try to make an attempt to make this planet a "Blood Immensely Present" planet such that during any emergency or surgery there is always blood present and a decrease in the number of deaths due to blood crisis. We are trying to solve a major problem by taking small steps and hopefully we would be able to solve sooner than we anticipated.

# Source of Information about blood donation and amount of blood required - American Red Cross Society.

## 2. CURRENT SOLUTIONS

1) **Blood app:** -This application supports functionalities like donation history till now and a show reel of your history with the application. Information about blood donation and the medical aspects of it are listed in the application. To encourage blood donation rewards are issued to people who make appointments and donate blood with the app. App allows you to create a team from your contacts whom you can recommend and check their blood donation activities. Application is missing our primary functionality: connection between donor/hospital and receiver.

2) **BloodBro Donor App:** - BloodBro Donor App is an exciting and interesting application. Application supports most of the functionalities listed in the above application. Users have profiles with contact info like phone, email id etc. Search donor options lists all nearest donors located near you, but there is no usage of map API. Search result lists the contact information of donor's with an option to email to the donor. In order to call the donor, we need to copy the number and open the "Caller" app for contacting the donor which is a time-consuming process. In our application, you can just call the donor and the caller application opens up directly with the donor's phone number without the copying overhead.

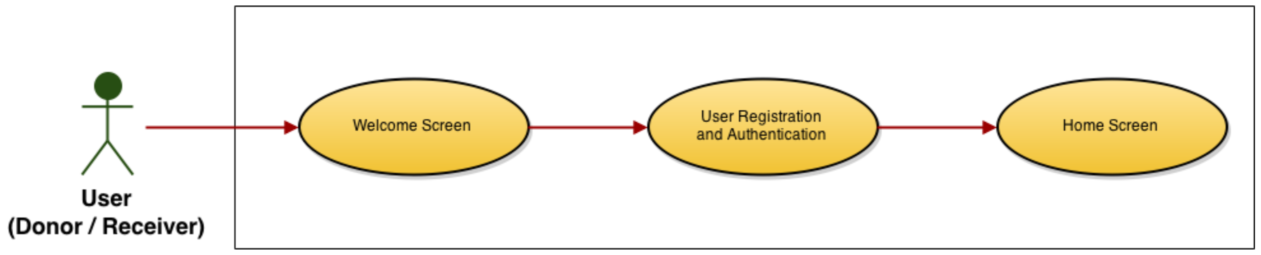
3) **My Blood for You:** - "My Blood For You" application is an improvised version of "Blood Bro Donor" with the caller donor feature added to that. There is no usage of Map API and information about nearer blood drives and hospitals in need of blood are not listed in the application. Sign-up page is simple with basic information to be listed but there is no password retrieval feature. Other Donor profiles are listing last 5 places visited by that donor when he accessed the application that is an unnecessary feature and obstructing privacy.

4) **Live Blood Bank**: - Live Blood Bank implements Map API in their application unlike rest of them, but there is no connectivity between donor and receiver's. Application only lists nearby clinics and blood drives. It is an application only for people who are interested in donating blood.

# 3. SOLUTIONS

## 3.1 USE CASES

### 3.1.1 Welcome Screen



#### 3.1.1.1 Welcome Screen

| | |
|---|---|
| Use Case | **Welcome Screen** |
| Description | This screen is the entry point of the application. |
| Preconditions | Screen appears when the user opens the application. |
| Normal Flows | 1. New users will be redirected to the registration page if the app is opened for the first time.<br>2. For the registered user, this screen will be followed by the Home Screen. |
| Post Conditions | 1. User is directed to the registration screen if the user is new.<br>2. A returning user is directed to the home screen of the app. |

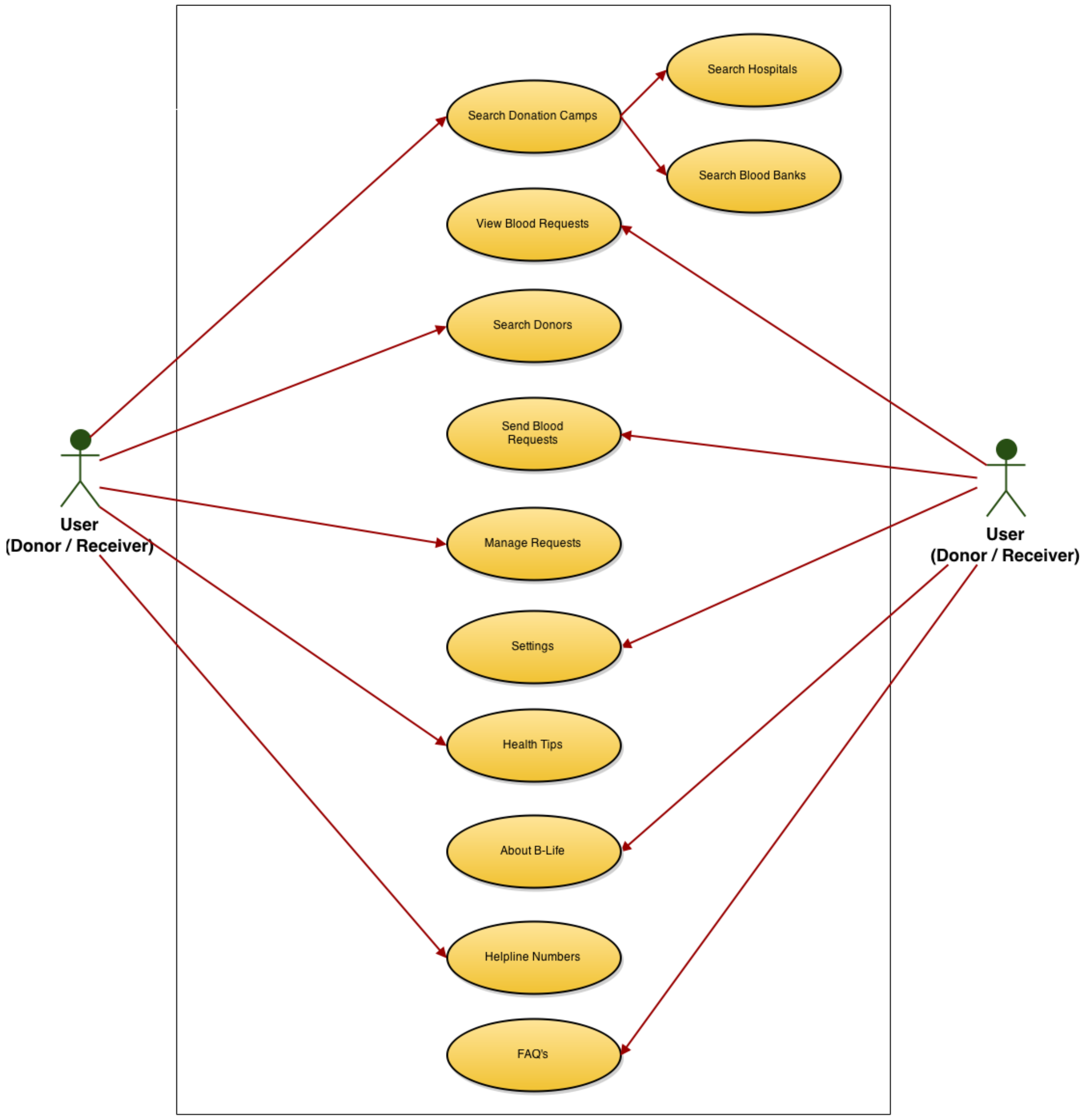
#### 3.1.1.2 User Registration and Authentication

| | |
|---|---|
| Use Case | **User Registration and Authentication** |
| Description | This screen is used for registering the user on the app. |
| Preconditions | 1. The application has installed by the user. |
| Normal Flows | 1. The user is asked to enter his name, phone number, username and other details.<br>2. If details are valid, user receives an authentication code via SMS.<br>3. After entering the code, user uses Register button to register on app. |
| Post Conditions | After the user hits Register button, he/she is directed to Home Screen. |

**3.1.1.3 Home Screen**

| | |
|---|---|
| Use Case | **Home Screen** |
| Description | This screen is the main UI for the application containing various options to navigate to. |
| Preconditions | User has installed the application and has been registered as a valid user. |
| Normal Flows | 1. User should be able to search for hospitals.<br>2. User should be able to send blood requests and view received notifications.<br>3. User should be able to access all the options and settings for the app. |
| Post Conditions | User is directed to the appropriate page on respective button clicks. |

### 3.1.2 Home Screen



#### 3.1.2.1 Search Donation Camps

| | |
|---|---|
| Use Case | **Search Donation Camps** |
| Description | This screen helps the user to search for the hospitals and the blood donation camps with the help of various parameters. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | 1. User is directed to the screen where user can search for the hospitals and blood donation camps<br>2. User can search for hospitals with the help of his location. |
| Post Conditions | User will be directed to a screen with list containing the results of his queries. |

#### 3.1.2.2 Search Blood Donors

| | |
|---|---|
| Use Case | **Search Blood Donors** |
| Description | This screen gives users the facility of searching for the blood donors located in their areas. They can be searched by entering details about the blood group required, city or the zip code. The donors which will be registered on the app will be displayed as the result of the search. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | 1. User searches for the blood donors by entering the search parameters.<br>2. The query is made to the database.<br>3. User can see the details about the blood donors and can contact them. |
| Post Conditions | The requested blood donors are displayed on the screen. User can contact the blood donors displayed on the screen. |

#### 3.1.2.3 View Blood Requests

| | |
|---|---|
| Use Case | **View Blood Requests** |
| Description | This screen contains the list of the blood requests received by the user. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | 1. User can view the blood requests displayed by the application.<br>2. User can view the details about the various requests and the senders. |
| Post Conditions | User can select a request to view its details and contact the sender. |

#### 3.1.2.4 Send Blood Requests

| | |
|---|---|
| Use Case | **Send Blood Requests** |
| Description | User can create a blood request and post it on the app and ask for donors. User should be able to enter name, contact details, Blood group required. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | 1. User who wants to make request can fill up the required forms and post the requests on the app.<br>2. User can set the deadline till the day the request has to be completed. |
| Post Conditions | The request has been successfully saved in the database. |

#### 3.1.2.5 Manage Requests

| | |
|---|---|
| Use Case | **Manage Requests** |
| Description | This screen deals with the blood requests that are made by the user. The user can manage the request posted by him/her and can delete them when no longer needed. |
| Preconditions | 1. User has installed the app and is registered on the app.<br>2. User has successfully created a blood request. |
| Normal Flows | 1. User can basically manage the blood request he posted.<br>2. If user deletes the request, he is directed to the home screen. |
| Post Conditions | NA |

#### 3.1.2.6 Settings

| | |
|---|---|
| Use Case | **Settings** |
| Description | This screen deals with the settings for the app and the user. User can change his profile settings and notification settings. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | 1. User can go to the settings page to change various settings in his profile and also to the app.<br>2. User can also change the notification settings from the edit notifications option. |
| Post Conditions | The change in settings is now visible and applied to the app. |

#### 3.1.2.7 Health Tips

| | |
|---|---|
| Use Case | **Health Tips** |
| Description | This screen shows various health tips for the app users. General Health tips related to blood donations along with those specifically related to donors and receivers. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | 1. Users can search for various health tips by clicking the specific sections. |
| Post Conditions | NA |

#### 3.1.2.8 Helpline Numbers

| | |
|---|---|
| Use Case | **Helpline Numbers** |
| Description | This screen contains the various helpline numbers related health care. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | Users can use this activity for searching various helpline numbers. User can be directed to this screen by clicking "helpline numbers" button from the home screen. |
| Post Conditions | NA |

#### 3.1.2.9 About B-Life

| | |
|---|---|
| Use Case | **About  B-Life** |
| Description | This screen contains the information about the android app. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | This activity can be accessed after clicking the about us button on the home screen. |
| Post Conditions | NA |

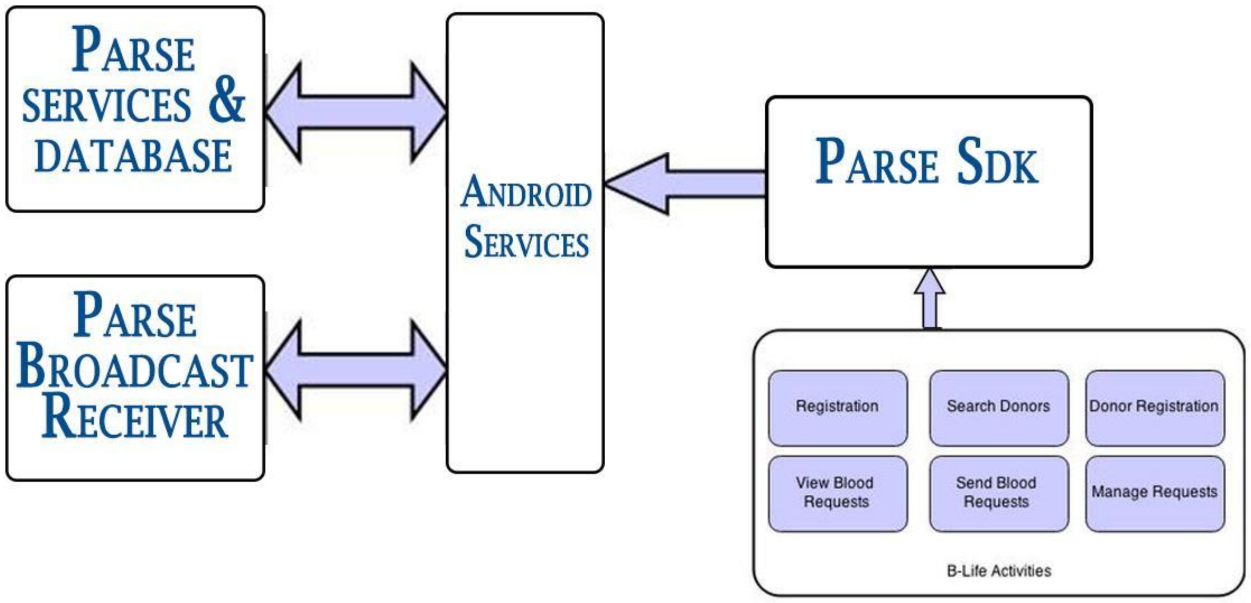
#### 3.1.2.10 FAQs

| | |
|---|---|
| Use Case | **FAQs** |
| Description | This screen contains the basic FAQ's about the blood donations. |
| Preconditions | 1. User has installed the app and is registered on the app. |
| Normal Flows | This activity can be accessed after clicking the FAQ button on the home screen. |
| Post Conditions | NA |

## 3.2 MVC Framework

### 3.2.1 Architecture Diagram

This diagram shows how our android app will be interacting with the web servers, APIs, and other android services.



### 3.2.2 MVC Framework Details

#### 3.2.2.1 Models

- **User:** Stores the user session information like user id, username, password as well as the personal details of the user like first name, last name, phone number etc., depending upon whether the user is registered as a receiver only or as a donor too. The specific data fields are as follows:

| | | |
|---|---|---|
| ○ userID | : | int |
| ○ username | : | String |
| ○ password | : | String |
| ○ phoneNumber | : | int |
| ○ firstName | : | String |
| ○ lastName | : | String |
| ○ city | : | String |
| ○ zipCode | : | int |
| ○ isDonor | : | boolean |
| ○ DOB | : | Date |
| ○ bloodGroup | : | String |
| ○ weight | : | int |
| ○ hasDisease | : | boolean |
| ○ gender | : | String |
| ○ isPregnant | : | boolean |

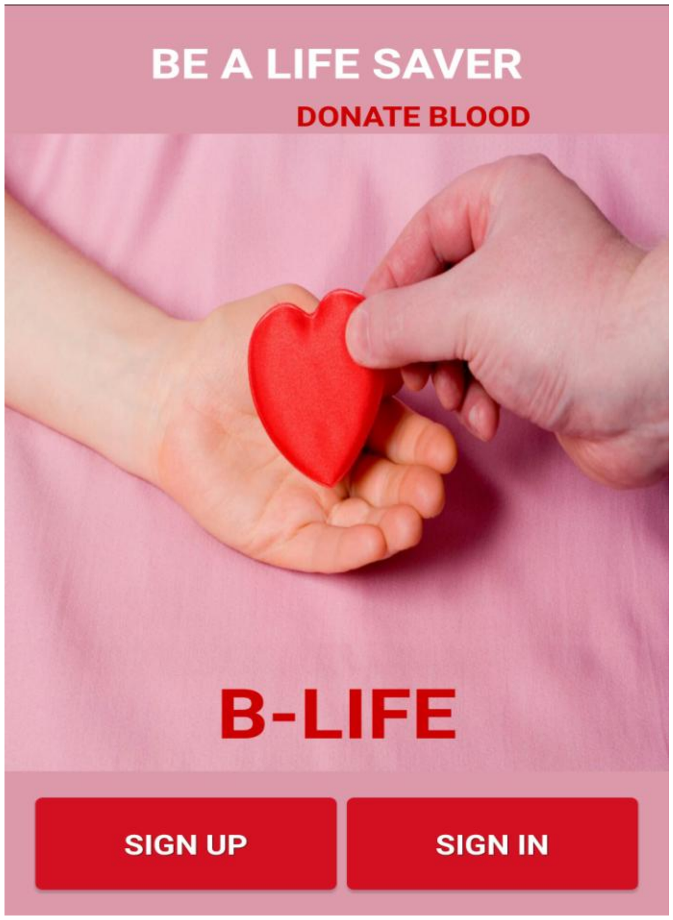
- **BloodRequest:** The request object that stores the information about the blood requirements like receiver (or patient) name, blood group, location, message, required on/before etc. The specific data fields are as follows:

| | | |
|---|---|---|
| ○ receiverName | : | String |
| ○ bloodGroup | : | String |
| ○ location | : | String |
| ○ city | : | String |
| ○ message | : | String |
| ○ requiredBefore | : | Datetime |
| ○ cellNumber | : | int |
| ○ requestStatus | : | String |
| ○ userID | : | int (User’s ID who submitted the request) |

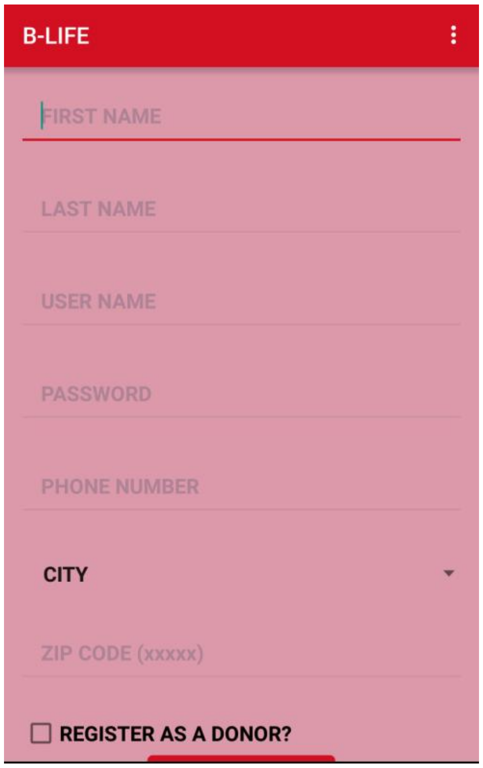
### 3.2.2.2 Views (UI Design)

The B-Life application UI consists of the following views or screens. The initial UI drawings for some of these views are shown below:

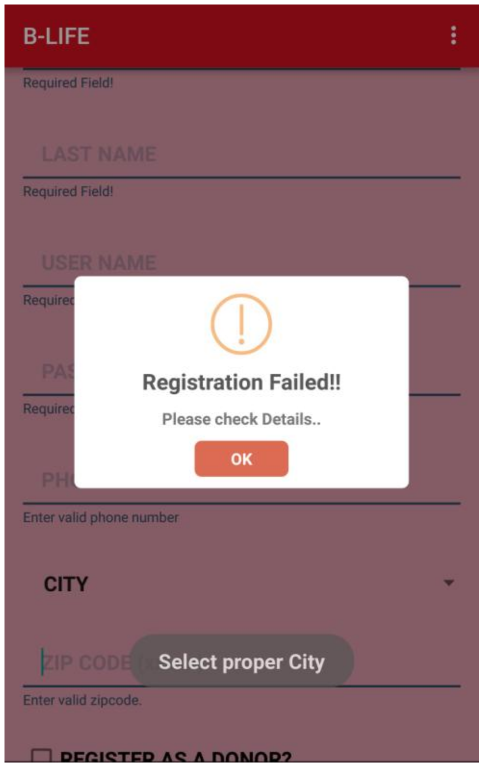
**Welcome Screen**



## Registration Screen



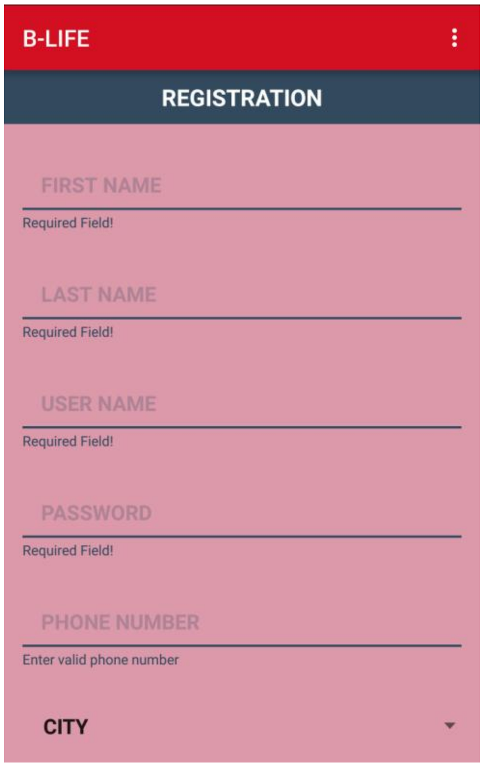
Validations have been performed on the registration page for ensuring proper user inputs and valid user authentication.



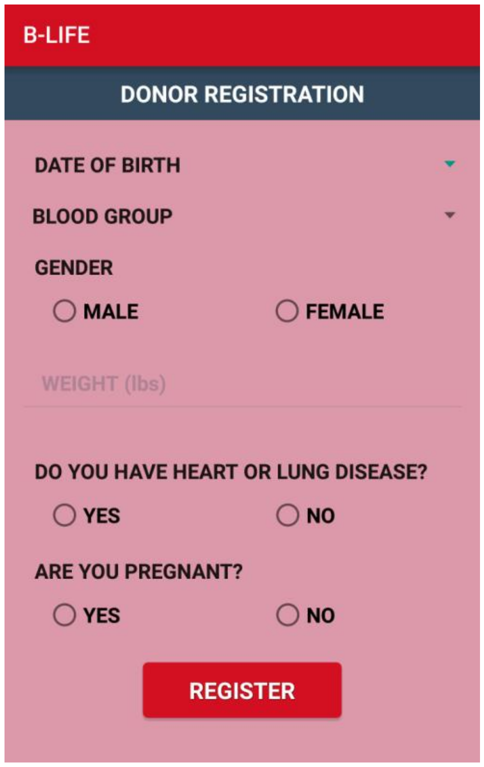
Registration Failed!!

Please check Details..

Select proper City



**Donor Registration Screen**


B-LIFE

DONOR REGISTRATION

DATE OF BIRTH

BLOOD GROUP

GENDER

MALE FEMALE

WEIGHT (lbs)

DO YOU HAVE HEART OR LUNG DISEASE?

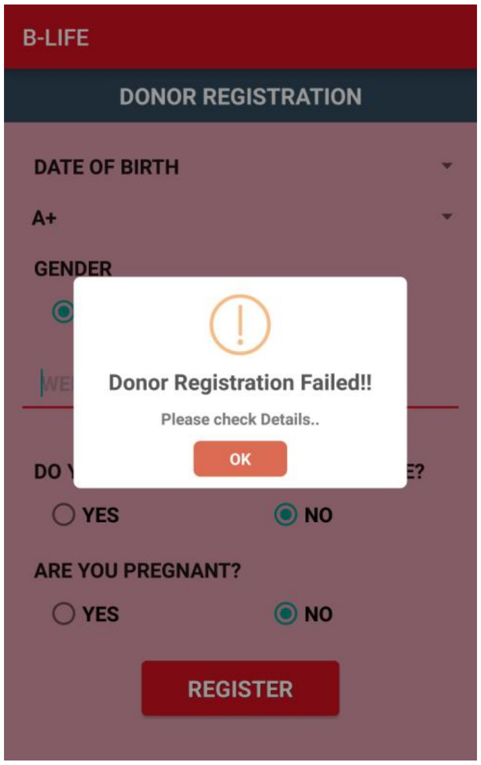
YES NO

ARE YOU PREGNANT?

YES NO

REGISTER

Validations have been performed and proper checks have been established in order to ensure that the user registering as a donor is eligible for blood donation.


B-LIFE

DONOR REGISTRATION

DATE OF BIRTH

A+

GENDER

Donor Registration Failed!!

Please check Details..

OK

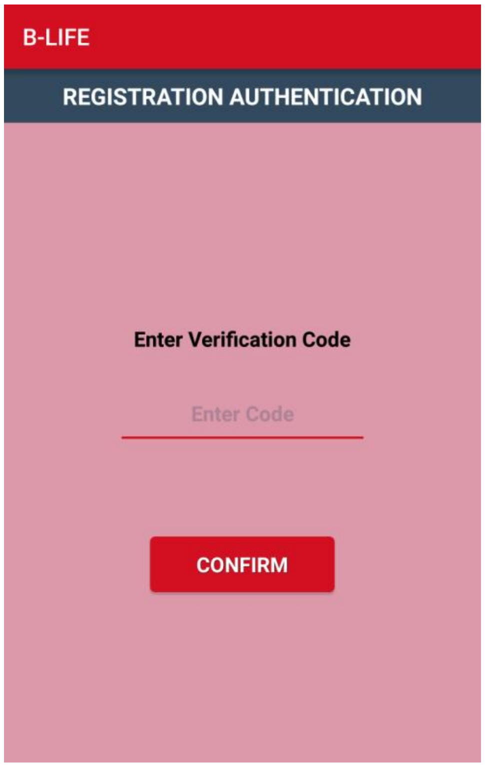
YES [✓ NO]

ARE YOU PREGNANT?

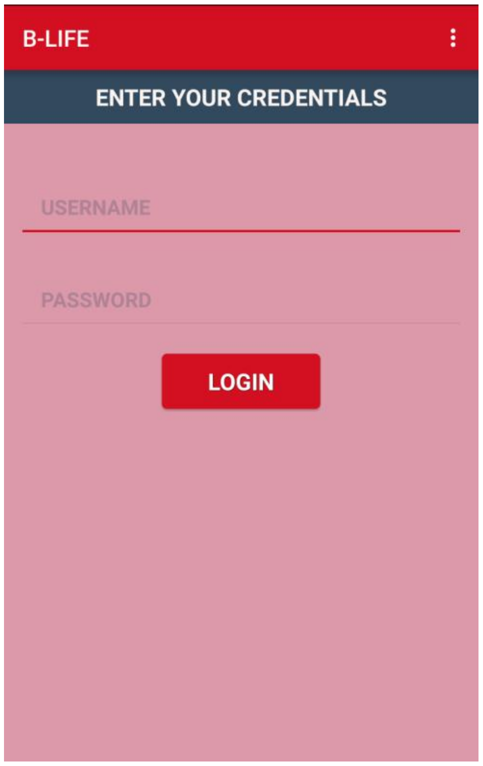
YES [✓ NO]

REGISTER

## Registration Authentication Screen



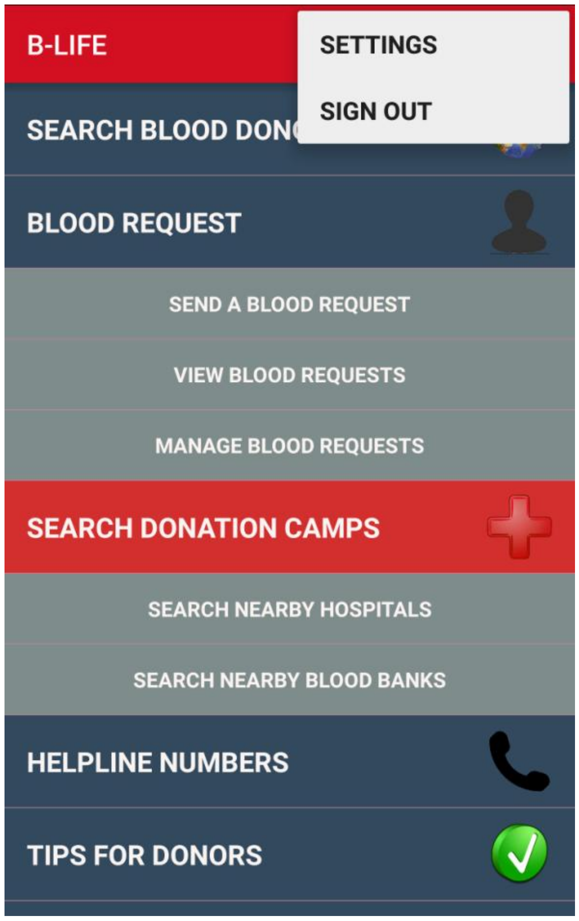
## Login Screen



**Home Screen**

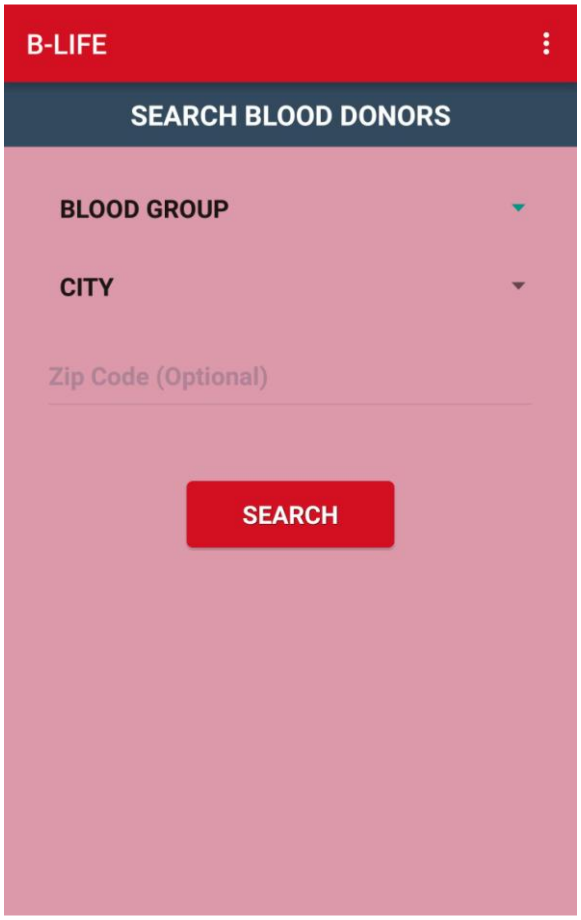
This is the home page for the application that helps user to navigate through the app fluently and smoothly. It provides links for a number of tasks such as -

| | |
|---|---|
| Search Blood Donors | Send A Blood Request |
| View Blood Requests | Manage Blood Requests |
| Search Nearby Hospitals | Search Blood Banks |
| Important Helpline numbers | Tips for Donors |



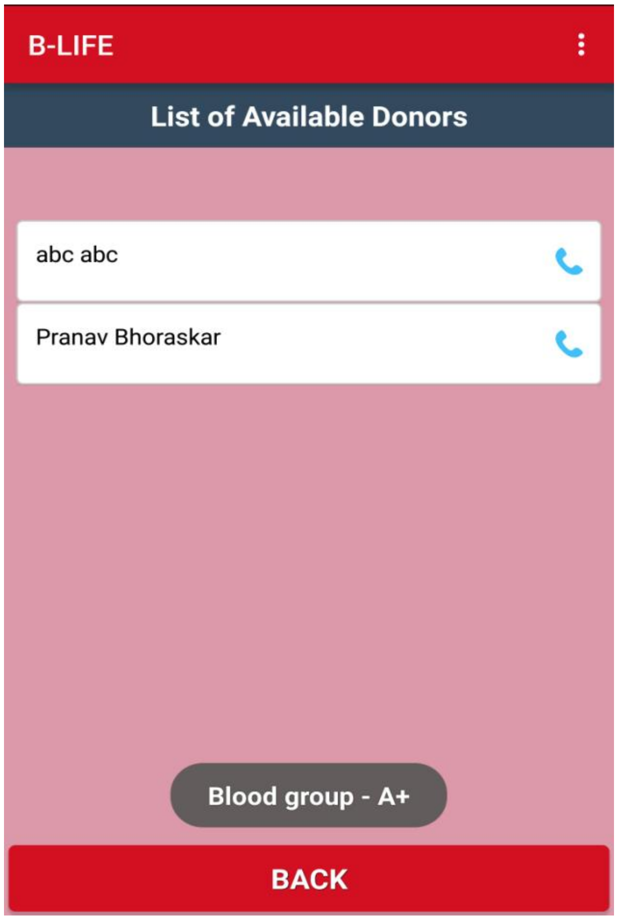
**Search Donors Screen**

This page provides a platform to search blood donors based on the parameters of Blood Group and City. a user can search blood donors as displayed in the following screen and the  results are displayed on the Donors List page.



**Donors List Screen**

This screen provides a list of donors based on the search performed by the user. Blood donors in a particular location and of particular blood group are displayed as shown below -

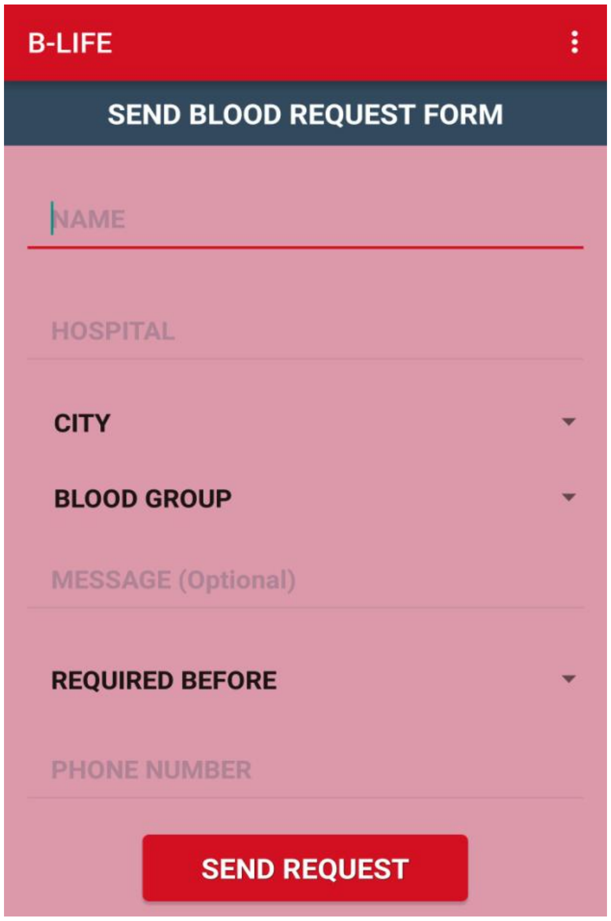


abc abc

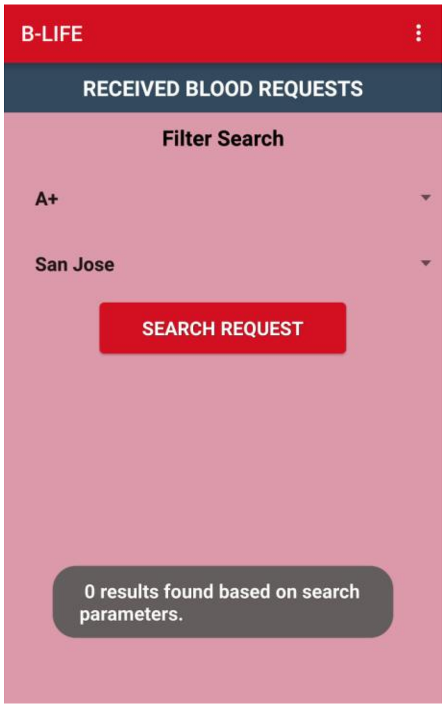
Pranav Bhoraskar

**Send Blood Request Screen**

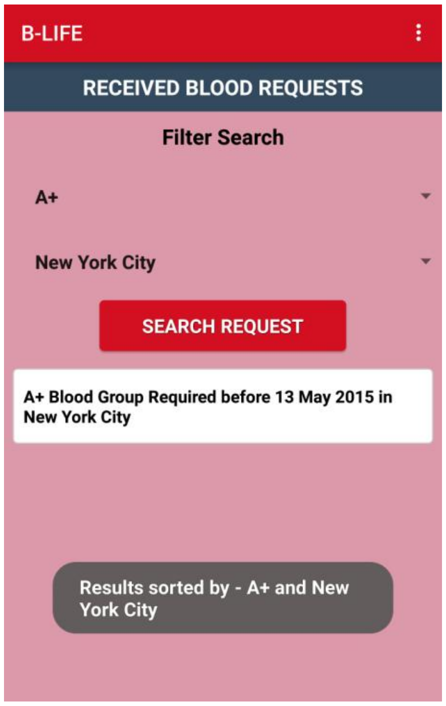
A user who is in need of blood can post a blood request on the app server so that the person who is eligible for blood donation can respond to the request. This blood request which is sent is received by the users who are located in the region where the blood request has been posted.



## View Blood Requests Screen

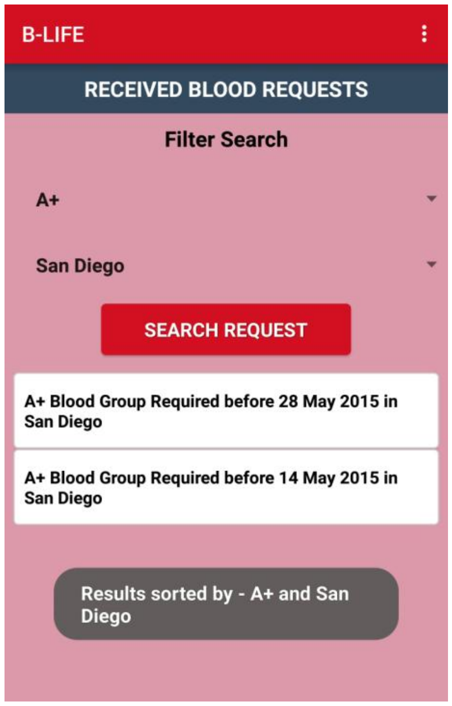


0 results found based on search parameters.



A+ Blood Group Required before 13 May 2015 in New York City

Results sorted by - A+ and New York City



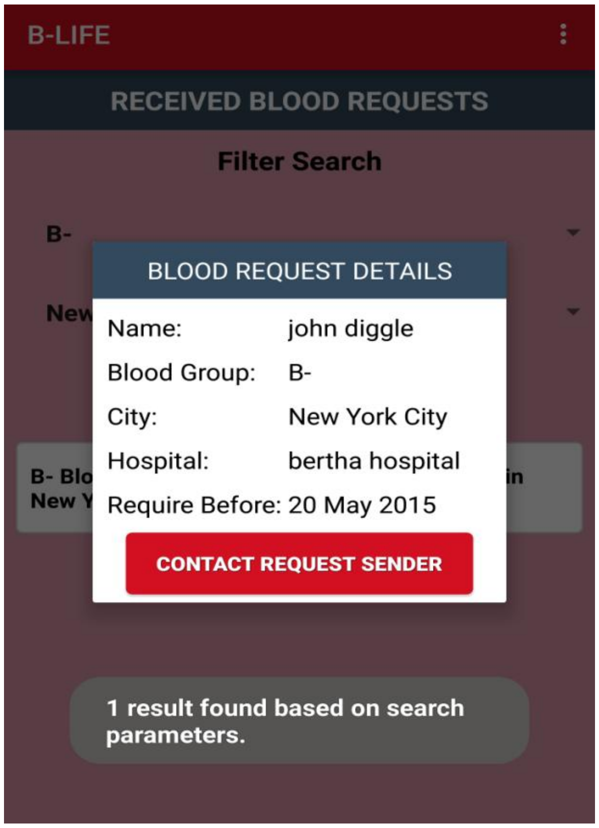
A+ Blood Group Required before 28 May 2015 in San Diego

A+ Blood Group Required before 14 May 2015 in San Diego

Results sorted by - A+ and San Diego

VBR screen provides user to search for various blood requests that has been posted on the application. User can narrow down the search results with the help of the search parameters provided, i.e, City and Blood Group.

**Blood Required Screen**


B-LIFE

RECEIVED BLOOD REQUESTS

Filter Search

B-

New

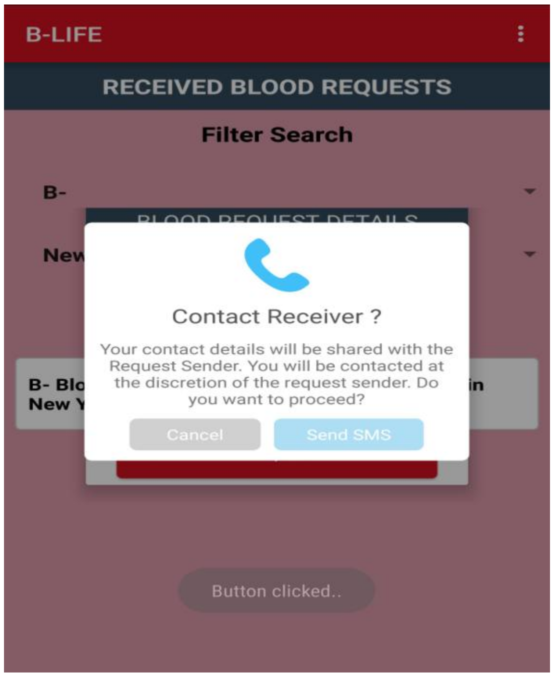
BLOOD REQUEST DETAILS

| | |
|---|---|
| Name: | john diggle |
| Blood Group: | B- |
| City: | New York City |
| Hospital: | bertha hospital |
| Require Before: | 20 May 2015 |

CONTACT REQUEST SENDER

1 result found based on search parameters.

This screen provides with the basic details which are required in order to identify the blood request. A donor can contact the blood request sender or by contacting with the help of "Contact Request Sender" button. An SMS is sent to the request sender and then the two parties can communicate at their discretion.


B-LIFE

RECEIVED BLOOD REQUESTS

Filter Search

B-

New

Contact Receiver ?

Your contact details will be shared with the Request Sender. You will be contacted at the discretion of the request sender. Do you want to proceed?

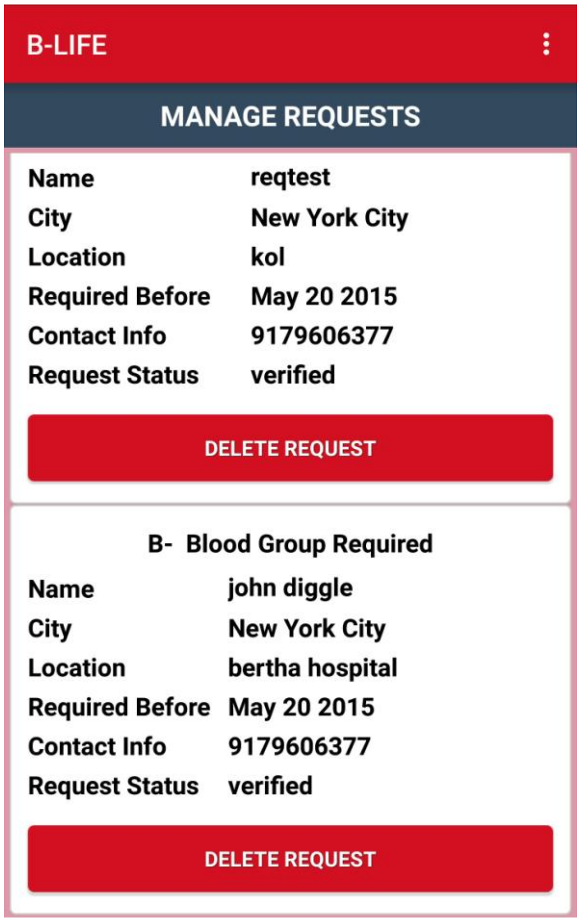
Cancel

Send SMS

Button clicked..

**Manage Requests Screen**

This screen helps the user to manage the blood requests that he has posted. The user has to verify the request by the verification code he receives via SMS. If the user status is "Awaiting Confirmation" then he can be redirected to the request verification page and then he can verify his request.



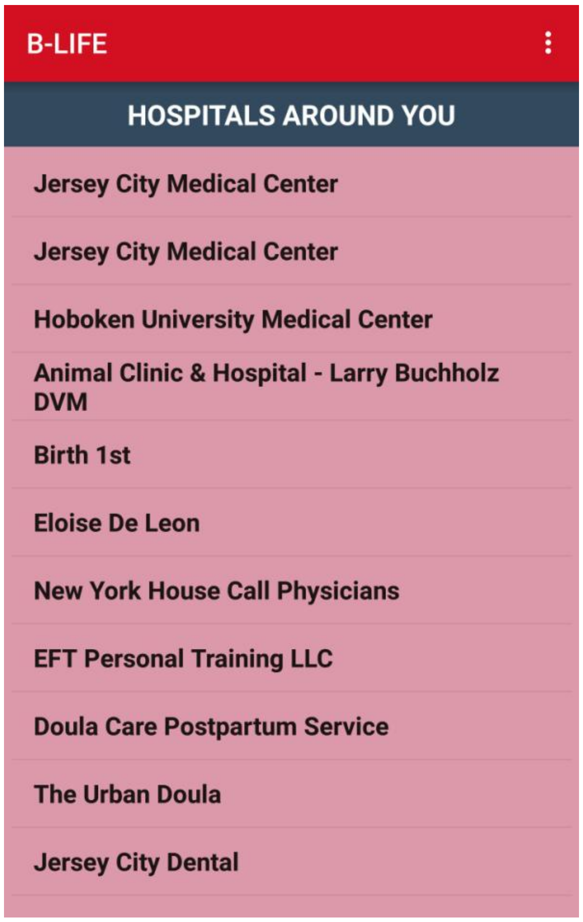
| | |
|---|---|
| Name | reqtest |
| City | New York City |
| Location | kol |
| Required Before | May 20 2015 |
| Contact Info | 9179606377 |
| Request Status | verified |

B- Blood Group Required

| | |
|---|---|
| Name | john diggle |
| City | New York City |
| Location | bertha hospital |
| Required Before | May 20 2015 |
| Contact Info | 9179606377 |
| Request Status | verified |

**Search Hospitals Screen**

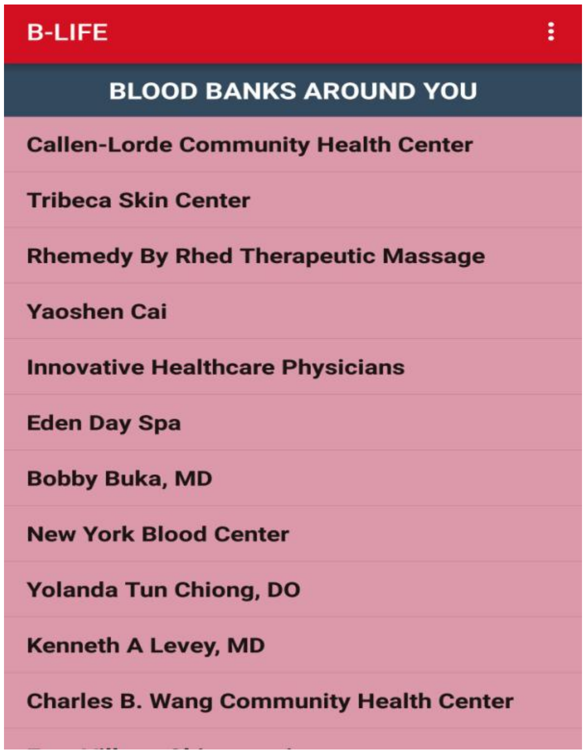
This screen helps the user by listing all the nearby hospitals nearer to the user's location. We have used the Yelp API for getting the information of the hospitals nearby. We have used a simple list view for showing the hospitals names. If we click on any hospital, it takes us to the selected hospital screen and lists the address and phone number of the specified hospital.



Jersey City Medical Center
Jersey City Medical Center
Hoboken University Medical Center
Animal Clinic & Hospital - Larry Buchholz DVM
Birth 1st
Eloise De Leon
New York House Call Physicians
EFT Personal Training LLC
Doula Care Postpartum Service
The Urban Doula
Jersey City Dental

**Search Blood Banks Screen**

This screen helps the user by listing all the nearby blood banks nearer to the user's location. We have used the Yelp API for getting the information of the blood banks nearby. We have used a simple list view for showing the blood banks names. If we click on any blood bank, it takes us to the selected hospital screen and lists the address and phone number of the specified blood bank.



B-LIFE

BLOOD BANKS AROUND YOU

Callen-Lorde Community Health Center

Tribeca Skin Center

Rhemedy By Rhed Therapeutic Massage

Yaoshen Cai

Innovative Healthcare Physicians

Eden Day Spa

Bobby Buka, MD

New York Blood Center

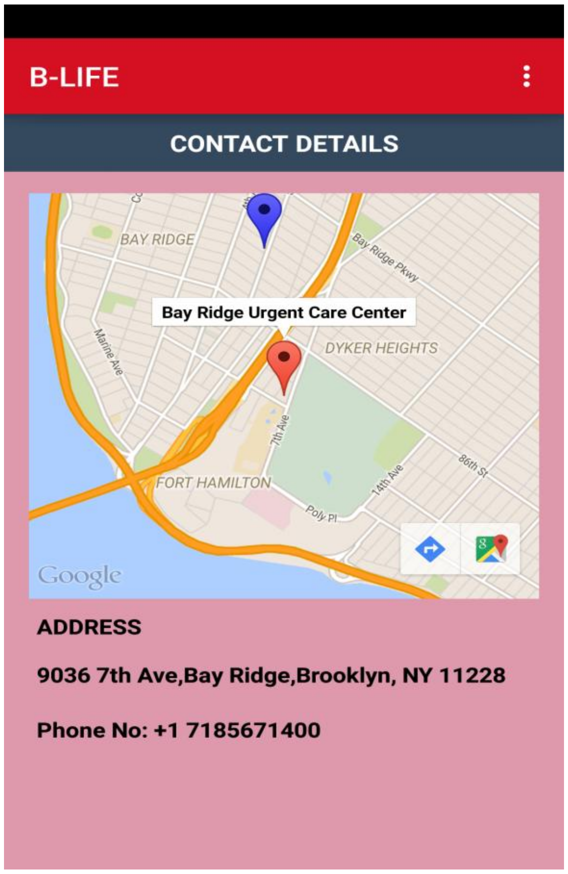
Yolanda Tun Chiong, DO

Kenneth A Levey, MD

Charles B. Wang Community Health Center

**Selected Hospitals Screen**

This screen shows the address and phone number of the selected hospital/blood bank/hotel. It has a map fragment with the location of the selected hospital/blood bank/hotel and the location of the user with the blue marker. If you click on the direction's button on the map fragment, it takes us to the google maps application and navigates us from our location to the destination. If we click on the phone number, dialer application is opened automatically with the phone number on it.

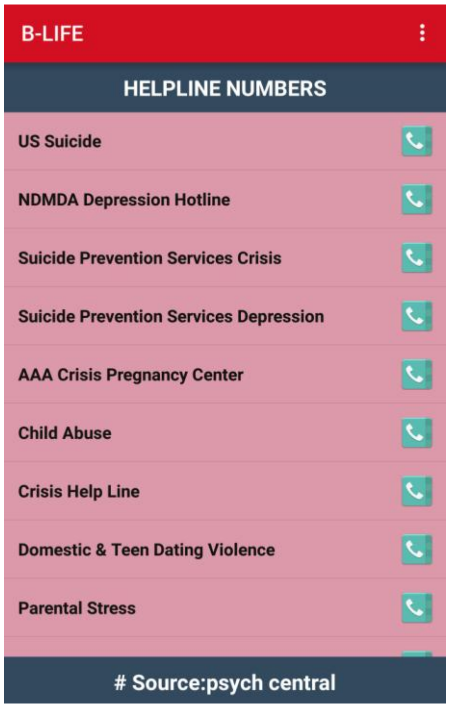


ADDRESS

9036 7th Ave,Bay Ridge,Brooklyn, NY 11228

Phone No: +1 7185671400

## Helpline Numbers Screen



HELPLINE NUMBERS

US Suicide

NDMDA Depression Hotline

Suicide Prevention Services Crisis

Suicide Prevention Services Depression

AAA Crisis Pregnancy Center

Child Abuse

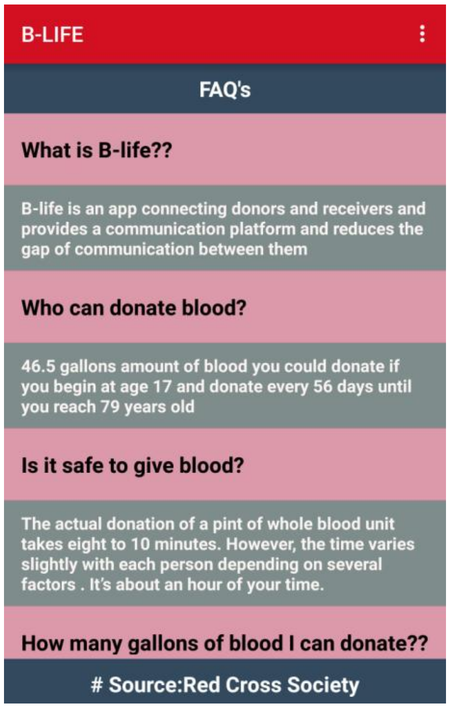
Crisis Help Line

Domestic & Teen Dating Violence

Parental Stress

# Source:psych central

## FAQs Screen



FAQ's

What is B-life??

B-life is an app connecting donors and receivers and provides a communication platform and reduces the gap of communication between them

Who can donate blood?

46.5 gallons amount of blood you could donate if you begin at age 17 and donate every 56 days until you reach 79 years old

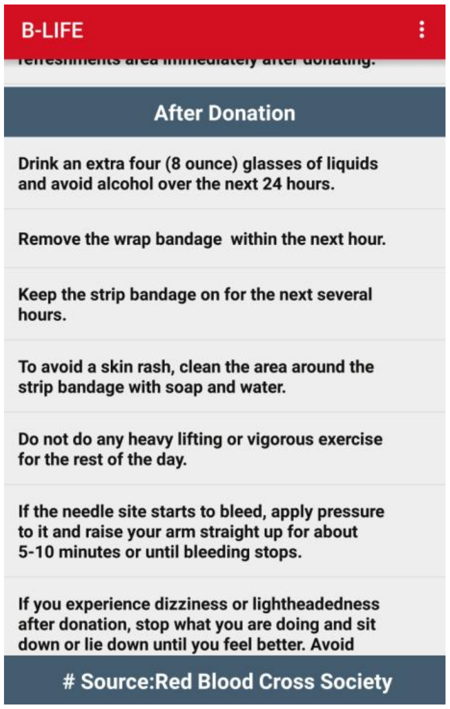
Is it safe to give blood?

The actual donation of a pint of whole blood unit takes eight to 10 minutes. However, the time varies slightly with each person depending on several factors . It's about an hour of your time.

How many gallons of blood I can donate??

# Source:Red Cross Society

## Health Tips Screen



B-LIFE

refreshments area immediately after donating.

After Donation

Drink an extra four (8 ounce) glasses of liquids and avoid alcohol over the next 24 hours.

Remove the wrap bandage  within the next hour.

Keep the strip bandage on for the next several hours.

To avoid a skin rash, clean the area around the strip bandage with soap and water.

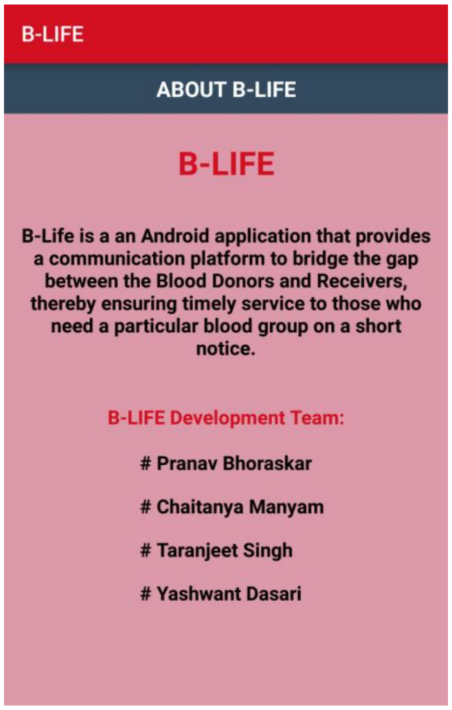
Do not do any heavy lifting or vigorous exercise for the rest of the day.

If the needle site starts to bleed, apply pressure to it and raise your arm straight up for about 5-10 minutes or until bleeding stops.

If you experience dizziness or lightheadedness after donation, stop what you are doing and sit down or lie down until you feel better. Avoid

# Source:Red Blood Cross Society

## About B-Life Screen



B-LIFE

ABOUT B-LIFE

B-LIFE

B-Life is a an Android application that provides a communication platform to bridge the gap between the Blood Donors and Receivers, thereby ensuring timely service to those who need a particular blood group on a short notice.

B-LIFE Development Team:

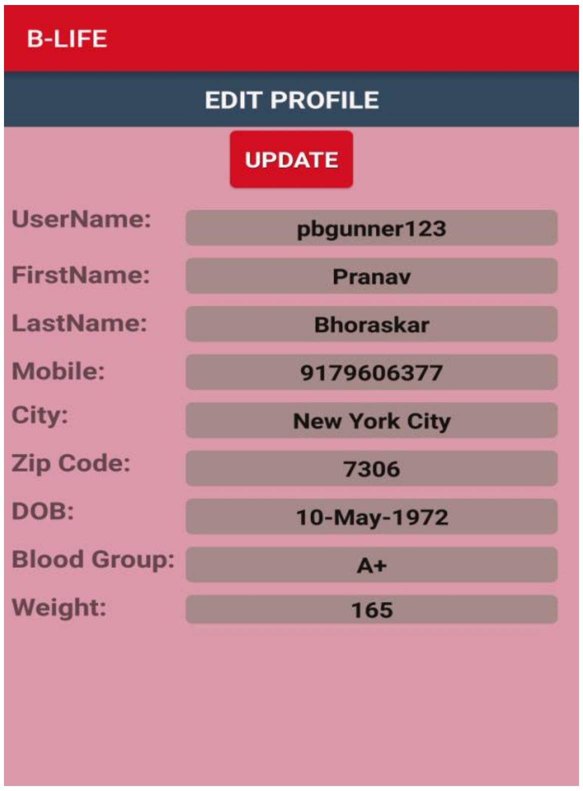
# Pranav Bhoraskar

# Chaitanya Manyam

# Taranjeet Singh

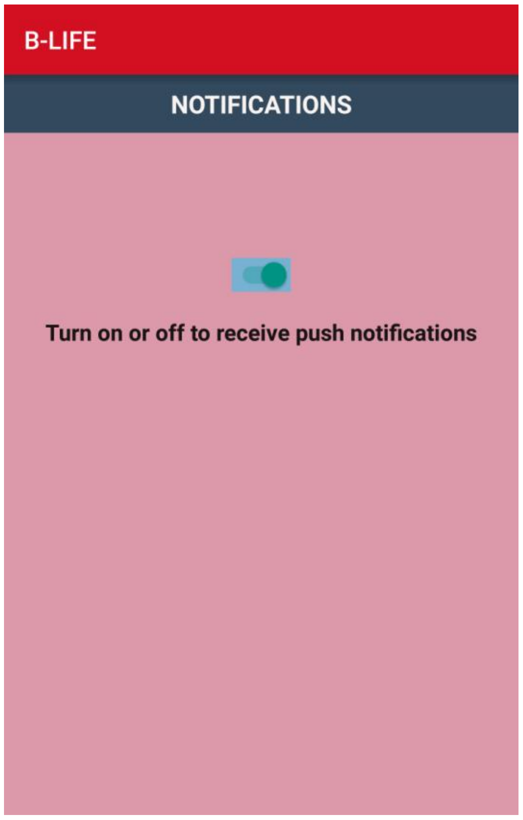
# Yashwant Dasari

**Settings Screen**
**Edit Profile Screen**



| | |
|---|---|
| UserName: | pbgunner123 |
| FirstName: | Pranav |
| LastName: | Bhoraskar |
| Mobile: | 9179606377 |
| City: | New York City |
| Zip Code: | 7306 |
| DOB: | 10-May-1972 |
| Blood Group: | A+ |
| Weight: | 165 |

**Edit Notifications Screen**



Turn on or off to receive push notifications

**3.2.2.3 Controllers (Activities)**
Every activity will control the corresponding view. It will be manipulating the data in local and/or server database and displaying the results on the UI. Apart from this, we will be implementing helper classes (as the need be) for controllers.

- **WelcomeActivity**
  This activity welcomes the user to the application and provides the user the option to either login to the application if he has the required credentials or sign up with the application if he is a new user.

- **RegistrationActivity**
  This activity is invoked when the user selects the sign up option from the Welcome screen. This activity is used to register a new user with server first time the user installs the application. A new user submits mandatory information like username, password, unique telephone number, first name, last name, city and zipcode. User is also provided an option to register as a donor on this activity. If the user selects that option, he moves to the DonorRegistrationActivity, otherwise the user submits the information. The server validates if telephone number is already registered with the server. If yes, then an error is generated where the user can choose to input new telephone number. If no error is generated then server sends an authentication code on mentioned telephone number.

- **DonorRegistrationActivity**
  This activity is used to register the user as a donor if he selects the option to register as a donor either from the RegistrationActivity or from the fragment displayed on any activity accessible from home screen (where the option is selectable anytime until a user is registered as a donor)
  A new donor submits the mandatory information like DOB, blood group, weight, whether he has a lung or heart disease, gender, whether the user is pregnant or not. Based on this information, either the user is accepted as a donor or rejected by displaying the relevant ineligibility message where the user chooses to continue as a receiver only. The server validates if telephone number is already registered with the server. If yes, then an error is generated where the user can choose to input enter new telephone number. If no error is generated then server sends an authentication code on mentioned telephone number.

- **RegistartionAuthenticationActivity**
  This activity allows the user to input authentication code received on his phone number to validate his authenticity and finally move to the home screen.
- **LoginActivity**
  This activity is invoked when the user selects the sign in option from the welcome screen. This activity allows the user to login with his credentials and move to the home screen if the credentials are valid or generates suitable error messages in case of invalid credentials.

- **HomeActivity**
  This activity is used to handle home screen view, which contains the following options:

  - Search Donors
  - Blood Requests (Send, View and Manage blood requests)

  - Search Donation Camps (Hospitals/Blood Banks)
  - Helpline
  - Heath Tips
  - FAQs
  - About B-Life

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **SearchDonorsActivity**
  This activity is invoked when the user selects the option to search for blood donors from the home screen and is used to handle the search for blood donors. User can search for a blood donor based on required blood group and city or zip code.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **DonorsListActivity**
  This activity is used to display the list of donors matching the search criteria entered by the user on the SearchDonorsActivity. User can select any donor and call the donor by using his listed cell number via phone dialer app.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **SendBloodRequestActivity**
  This activity is invoked when the user selects the option to send a blood request from the home screen. It allows the user to submit a blood request by entering the required information like name of the patient/receiver, blood group required, location, city, message, required before date and time, and cell number. On submitting the request, the server generates a verification code to confirm the entered cell number and the user moves to the Manage Requests where in the user can view his submitted request with the relevant status (like awaiting verification). To finally submit a request successfully, the user must enter the received verification code to verify his cell number and then the request gets saved in the server database.

This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **ViewBloodRequestsActivity**
  This activity is invoked when the user selects the option to view blood requests from the home screen. Using this activity, the user can view the valid blood requests retrieved from the server database based on the selected blood group or city as a list view. User can select any blood request to move to the Blood Required screen where he can view request details and has the option to contact the requestor via cell phone or share his photo via phone camera if he is interested to respond to this request.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **ManageRequestsActivity**
  This activity is invoked when the user selects the option to manage blood requests from the home screen. It provides the user the options to verify and finally submit a request awaiting verification and to delete any request submitted by him.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **SearchHospitalsActivity**
  This activity is invoked when the user selects the search hospitals option from the home screen. The user can search for any hospitals around his location or any other location of interest through Yelp API which will be used to handle hospital search by this activity. User can also select any hospital retrieved by the Yelp API to view its location on the map, which will be handled by the Google Maps API.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **SearchBloodBanksActivity**
  This activity is invoked when the user selects the search blood banks option from the home screen. The user can search for any blood banks around his location or any other location of interest through Yelp API which will be used to handle blood banks search by this activity. User can also select any blood bank retrieved by the Yelp API to view its location on the map, which will be handled by the Google Maps API.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **HelplineNumbersActivity**
  This activity is invoked when the user selects the helpline option from the home screen. It displays the list of all the important helpline numbers related to blood donation. The user has the option to call any helpline number via phone dialer app.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **FAQsActivity**
  This activity is invoked when the user selects the FAQs option from the home screen. It displays the static information about blood donation in the form of FAQs for the blood donors as well as for the blood receivers.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **HealthTipsActivity**
  This activity is invoked when the user selects the health tips option from the home screen. It displays the static health tips information for the blood donors as well as for the blood receivers.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings

  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **AboutBLifeActivity**
  This activity is invoked when the user selects the about B-Life option from the home screen. It displays the information about the B-Life application and its developers.

  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **SettingsActivity**
  This activity is invoked when the user selects the setting option from the fragment which is accessible from any activity of the application.
  The user can choose to edit profile settings or notification settings using this activity.

- **EditProfileActivity**
  This activity is invoked when the user selects the option to edit profile settings from the Settings screen. It allows the user to edit his profile details. User is also provided the flexibility to set his availability status in the profile settings based on which he will be either displayed on the Donors List screen or not.

- **EditNotificationsActivity**
  This activity is invoked when the user selects the option to edit notifications from the Settings screen. It allows the user to edit the notification setting based on which the user will either receive the push notifications for the new blood requests or not.

- **SearchHotelScreen**
  This activity is invoked when the user selects the search hotels option from the home screen. The user can search for any hotels around his location or any other location of interest through Yelp API which will be used to handle hotels search by this activity. User can also select any hotel retrieved by the Yelp API to view its location on the map, which will be handled by the Google Maps API.
  This activity will also contain a fragment through which the user can select the following options:

  - Settings
  - Register as a donor (if the user is not registered as a donor yet)
  - Logout

- **SelectedHospitalScreen**
  This activity is invoked when the user selects any hospital / blood bank / hotel from their respective search activities. This activity has a small map fragment with the specified location

marker as well as the user's device location marker which we got from the google maps API. It also lists the address and phone number of the selected hospital / blood bank / hotel.
This activity will also contain a fragment through which the user can select the following options:

- Settings
- Register as a donor (if the user is not registered as a donor yet)
- Logout

### 3.2.2.4 Parse Broadcast Receiver

Push notifications are a great way to keep the users engaged and informed about the app. The parse library provides push notifications using google cloud messaging (GCM). Every parse application installed on a device registered for push notifications has an associated installation object. In android, installation objects are available through ParseInstallation.
Push notifications can be sent into two ways using channels and advanced targeting. In our app, we have used advanced targeting; in advanced targeting, we can write complex queries according to our requirements with the help of customized Broadcast receiver. According to application's requirements, Broadcast receiver has been modified which extends ParsePushBroadcastReceiver.

```
public class NotificationReceiver extends ParsePushBroadcastReceiver {
  private static final String TAG = "NotificationReceiver";
  private static final String PARSE_JSON_NOTIFICATION = "com.parse.Data";

  //override the onPushOpen() method to open "ViewBloodRequestActivity" instead of app's launcher activity
when user taps on a Notification
  @Override
  protected void onPushOpen(Context context, Intent intent){
    //since we override onPushOpen() method, we need to track our app's push open event manually
    ParseAnalytics.trackAppOpenedInBackground(intent);

    String targetActivity = null;

    //get the value of key "target" from the JSON data received in Parse Push notification
    try
   {
      JSONObject pushData = getJSONDataFromIntent(intent);
      targetActivity = pushData.getString("target");
    }
   catch (JSONException e)
   {
      Log.e(TAG, "Unexpected JSONException when receiving push data: ", e);
    }

    //get the launcher activity class for the app from ParsePushBroadcastReceiver's getActivity() method
    Class<? extends Activity> cls = getActivity(context, intent);

    Intent activityIntent;
    if (targetActivity != null && !targetActivity.isEmpty())
   {
      //send an explicit intent to open ViewBloodRequestActivity on tapping a push notification
      activityIntent = new Intent(context, ViewBloodRequestActivity.class);
      Log.d("notttt - ", "entering if");
```

```
        }
       else
       {
           //open the launcher activity of the app
           activityIntent = new Intent(context, cls);
           Log.d("notttt - ", "entering else");
        }

        if (Build.VERSION.SDK_INT >= 16) {
           TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);
           stackBuilder.addParentStack(cls);
           stackBuilder.addNextIntent(activityIntent);
           stackBuilder.startActivities();
        }
       else
       {
           activityIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
           activityIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
           context.startActivity(activityIntent);
        }
    }

    private JSONObject getJSONDataFromIntent(Intent intent) {
        JSONObject jsonData = null;
       try
        {
           jsonData = new JSONObject(intent.getExtras().getString(PARSE_JSON_NOTIFICATION));
        }
      catch (JSONException e)
        {
           Log.e(TAG, "Unexpected JSONException when receiving push data: ", e);
        }
        return jsonData;
    }
}
```

## 3.3 Other Android Components

- **Database (Parse) -** Parse is cloud platform for storing data securely and effectively. Parse works on PAAS (platform as a service) concept which means parse provides a single server which stores basic data types, including locations and photos and query across them without spinning up a single server. Parse allows focusing on creating amazing user experiences and forgetting complex infrastructure. Parse also sends targeted push notifications across devices and platforms with ease.

  Setting the Parse platform is very easy; following are the steps to set up:
  1. Sign up with parse and create the new android app.
  2. Install the parse sdk inside the lib directory of the app.
  3. Initialize the application with parse generated keys.
  4. For push notifications, get current installation after initializing.

- **APIs -** 3rd party APIs can be used for searching options - such as Searching Hospitals, donors etc. Hospitals can also be searched with the help of Google-Maps API. Yelp API can be used for searching hospitals in a particular region.

**Yelp API :-** Yelp API is easy to use and to implement. By using Yelp API, we can gain easy access to search results and local business information from over 50 million businesses in 27 countries. We can find up to 40 best results for a geographically oriented search. Secondly, Yelp sort results by the best match for the query, highest ratings, or distance. We can limit results to those businesses offering a Yelp Deal, and display information about the deal like the title, savings, and purchase URL. We can identify and display whether a business has been claimed on Yelp.com. The API uses a standard, secure authorization protocol (OAuth 1.0a, xAuth) for authenticating requests and offers various API methods. For using Yelp API, it uses OAuth 1.0a for authenticating API requests as per the OAuth specification. Each Yelp request contains these parameters: oauth_consumerkey, oauth_token, oauth_signature_method, oauth_signature, oauth_timestamp, oauth_noncw. These parameters may be passed in the HTTP (Authorization) header as URL query keys or in the POST data. Generating the OAuth signature is done by applying the HMAC-SHA1 with the oauth_token_secret.

We have to create an account with Yelp developers to get the consumer key and token to interact with the Yelp API. While Interacting with the Yelp API, we have to send a JSON request with the search keyword and device's location (latitude and longitude). We get a JSON array with all the relevant information of locations near you with information like address, phone number, reviews, url etc.

For eg:-
We have to add the Yelp.java in our project and whenever we are using the yelp on any specific activity, we have to make an instance of the yelp class and use it in that specific activity. Let us make get_blood_bank a Yelp object.

```
String blood_bank = get_blood_bank.search("blood banks", lat, lng);
```

After running this statement, we would be getting a list of all blood banks nearer to the lat and lng declared in the above statement. Later, we need process that string with the help of Json objects and get the specific fields we are interested.

**Google Maps API :-**
The Google Maps JavaScript API v3 is a powerful, popular mapping API. It's simple to use to add maps to your website, or web or mobile application, and provides a wide range of services and utilities for data visualization, map manipulation, directions, and more. We have to create an account with Google developer's site and generate API key and token by giving system's SHA-1 Key. In our application, we used map fragment in the activity page for displaying Google maps. Later, we can add markers by listing co-ordinates of the location we would like to show on the map with title name and marker. By clicking on the marker, we can navigate from our location. We can initially set the camera whether we want it in zoomed mode or animated mode.

For example :-
We have to create a fragment on the xml page of the specified activity for displaying the google map.

```
<android:name="com.google.android.gms.maps.MapFragment" >
```

This specific command will generate the google maps on the fragment. Any changes to the maps can be done dynamically by the class file of the specified activity whether it is adding marker, camera options, zoom in and zoom out options etc.
In order to use the google maps API, we have declare the API key in the manifest file.

```
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="specific to each system/account" />
```

API value is unique for each system and the same value cannot be used on other systems for interaction with Google maps API.

**GPS Tracker :-**
It checks periodically whether the device is connected to the wifi or not as well as the GPS is enabled or not. If the device is not connected to either one of them, a sweet alert dialog box is opened and it opens up the setting's page to enable that specific functionality. Device location is obtained by using the location manager class and is stored in a Location variable. We can obtain the latitude and longitude of the device by calling the location methods i.e getLatitude and getLongitude methods. We try to calculate the last known location of the device. Once we are done in getting the device co-ordinates, we can disable the GPS functionality by calling the stopUsingGps method. GPS Tracker in the application when the device location is required whether for finding nearby hospitals etc. GPS Tracker must be called periodically as it is an Async task rather than an service when device location should be updated periodically.

- **Adapter classes**
  - **<u>CustomAdapter</u>**

    This adapter extends the parse query adapter. The main purpose of this adapter is to query the parse database with user inputs and list the results in the list view. Custom Adapter is used to get the list of donors and display it on list view.

  - **<u>ExpandableListAdapter</u>**

    This class displays the list of items in a list view and each item on click will expand with more list items. This adapter extends BaseExpandableListAdapter and has been used on multiple screens like Faq's, Health Tips and etc.

  - **<u>TabsPageAdapter</u>**

    This class extends the FragmentStatePagerAdapter and is used for creating the fragments for swiping between two screens. This adapter is used for Edit Settings screen.

# 4. ADVANCED ANDROID FEATURES

## 4.1 Android Lollipop 5.0 Integration -

### 4.1.1 Recycler Views -

Recycler Views are the new improved feature that are introduced in the Android Lollipop 5.0. They are a new and improved feature for the classical 'ListView' of the older android versions which are soon going to be deprecated. Recycler Views, in our project, provided us with a more interactive way of displaying contents in a form of Lists. Its adapter usually contains OnBindViewHolder and OnCreateHolder functions. It also has a ViewHolder class basically used for interacting with the contents in the layout of recycler view.

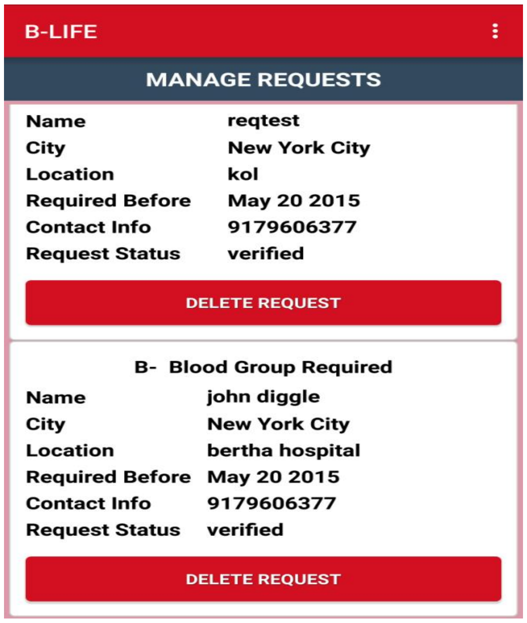
### 4.1.2 Card View -

Card View is a new feature that is introduced in the Android Lollipop 5.0. They are a new and improved feature used for displaying contents in the form of cards. Card Views, in our project, provided us with a more interactive way of displaying contents in a form of cards on the 'Manage Request Screen'. Its adapter usually contains OnBindViewHolder and OnCreateHolder functions. It also has a ViewHolder class basically used for interacting with the contents in the layout of card view. The adapter of card view is similar to that of Recycler View.

Code snippet for Card Views as used in our project -

```
@Override
  public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    // create a new view
    View itemLayoutView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.mrequest_cardview_item, parent, false);

    ViewHolder viewHolder = new ViewHolder(itemLayoutView);
    return viewHolder;
  }
```



| | |
|---|---|
| Name | reqtest |
| City | New York City |
| Location | kol |
| Required Before | May 20 2015 |
| Contact Info | 9179606377 |
| Request Status | verified |

B- Blood Group Required

| | |
|---|---|
| Name | john diggle |
| City | New York City |
| Location | bertha hospital |
| Required Before | May 20 2015 |
| Contact Info | 9179606377 |
| Request Status | verified |

### 4.1.3 Material Edit Texts -

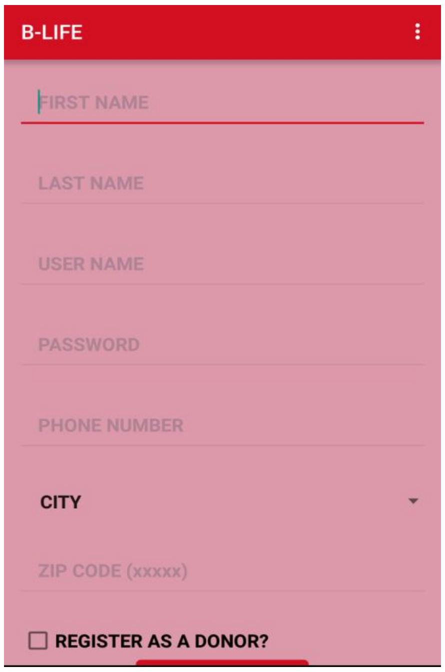
Material Edit Texts are a feature of Android 5.0 Material Design. With more animations integrated with more smooth and fluent UI, this feature enables a more user friendly and more interactive UI to the user. As our application is backward compatible with older versions of Android, we simply integrated the online library dependency in the 'app-gradle' file in order to utilize this feature.
These edittexts provided a a number of attributes such as color for the hints and the errors that have to be displayed when the user enters some wrong inputs. This features is used mainly in the places where the user has to fill forms, such as - Registration page, Donor Registration form and send a blood request form.

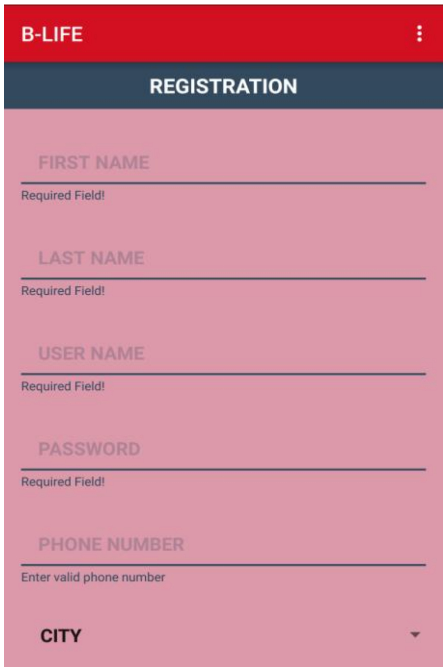
Dependency Library used -

```
dependencies {  compile 'com.rengwuxian.materialedittext:library:2.0.3' }
```

Code Snippet for Material Edit-Texts as used in our project -

```
<com.rengwuxian.materialedittext.MaterialEditText
        android:textColorHint="#34495e"
        app:met_errorColor="#34495e"
        app:met_baseColor="#34495e"
        app:met_primaryColor="#FFD41022"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:padding="16dp"
        android:inputType="number"
        android:id="@+id/editWeight"
        android:layout_marginTop="10dp"
        android:gravity="bottom|left"
        android:hint="@string/register_donor_weight"
    />
```





## 4.2 Sweet Alert Dialog Box -

This feature is basically an advanced version of the default and classical 'Alert Dialog Box' which provides a more interactive UI for a otherwise "boring" Alert Dialog box. With this Sweet alert dialog box animations in the alert dialog box were made smoother. It came with various types of alerts such as - WARNING_TYPE, SUCCESS_TYPE, CUSTOM_IMAGE_TYPE etc. This dialog box was easily accessible with a library dependency that was added in the 'app-gradle' file.

Dependency Library used - dependencies { compile 'cn.pedant.sweetalert:library:1.3' }

Code Snippet for Sweet alert dialog Box as used in our project - (DonorsListScreenActivity.java)

```
new SweetAlertDialog(DonorsListScreenActivity.this, SweetAlertDialog.CUSTOM_IMAGE_TYPE)
            .setTitleText("Contact donor?")
            .setCustomImage(R.mipmap.phone)
            .setContentText("Your contact details will be shared with the Donor. You will be
                    contacted at the " +
                "discretion of the Donor. Do you want to proceed?")
            .setConfirmText(" Send SMS ")
            .setCancelText("  Cancel  ")
            .showCancelButton(true)
            .setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {
              @Override
              public void onClick(SweetAlertDialog sDialog) {
                sendSMSMessage(user_phone);
                sDialog.cancel();

              }
            })
            .setCancelClickListener(new SweetAlertDialog.OnSweetClickListener() {
              @Override
              public void onClick(SweetAlertDialog sDialog) {
                sDialog.cancel();
              }
            })
            .show();
```

## 5. AT&T FEEDBACK & FEEDBACK IMPLEMENTATIONS

**Suggestions & Problems addressed by AT&T People:**

- Encourage blood donors to donate blood by giving incentives and rewards depending on the blood donor’s donation history.
- Contact Hospitals to be as an end-user in this process such that even hospitals can request blood and send notifications to nearby blood donors.
- Extend Blood request notifications functionality by targeting specific users based on location.
- Extend functionality by adding a new tab to list nearby hotels depending on the user’s location to eat after donating blood. Encourage user to take high protein food after donating blood by using the above listed feature.
- List reference names at the bottom of each page from which data has been taken in that specific page.
- Increase donor/receiver privacy setting by hiding contact details until donor have accepted the user’s blood request.

**Feedbacks as were Implemented -**

- A new page “Post-Donation Snacking!” was introduced by integrating it with Yelp API and getting list of nearby eat-outs depending on the user’s location so that the user can go and re-energize himself after blood donation.
- Push notification functionality was extended such that blood requests are notified only to the targeted users depending on location of the request.
- Contact details were kept hidden from all parties unless and until they communicate with each other on their own discretion. Hence, privacy setting was implemented to keep user details secret.
- Source of information has been listed as was addressed in the AT&T feedback.

## 6. FUTURE WORK

- Give incentives and rewards to donors for periodical blood donation based on donation history.
- Encourage hospitals to participate in the application program as an end-user to post blood requests.

# 7. PROJECT MEMBERS ROLES & RESPONSIBILITIES

## 7.1 Taranjeet Singh: Android Developer & Database Developer

- Android UI Development for following screens:
  - Welcome Screen
  - Registration Screen
  - Donor Registration Screen
  - Registration Authentication Screen
  - Login Screen
  - Home Screen
- Android Activity development for the following activities:
  - WelcomeActivity
  - RegistrationActivity
  - DonorRegistrationActivity
  - RegistrationAuthenticationActivity
  - LoginActivity
  - HomeActivity
- Helping team members in UI development of other screens.
- Designed the database model objects (User and Blood Request) for B-Life's MVC architecture and helped the team in implementing them using Parse Cloud Platform.
- Designed and implemented the "Targeted" (based on User's City) push notifications for the app using "Advanced Targeting" method of sending notifications prescribed in Parse Push Notifications service.
- Implemented a customized ParsePushBroadCastReceiver for the app to achieve the goal of invoking a specific target Activity when a push notification is opened.
- Development of Model Classes.
- Research & Development on using Yelp API for searching hospitals & blood banks.
- Unit Testing, UI Testing, System Testing, bug fixes and feedback.
- Review & update design document as per the review comments from team lead.

## 7.2 Pranav Bhoraskar: Team Leader & Android Developer

- Android UI Development for following screens:
  - Search Donors Screen
  - Donors List Screen
  - Send Blood Request Screen
  - View Blood Requests Screen
  - Blood Required Screen - SingleItemView - Dialog Activity

- Android Activity development for the following activities:
  - SearchDonorsActivity
  - DonorsListActivity
  - SendBloodRequestActivity
  - ViewBloodRequestsActivity
  - BloodRequiredActivity - SingleItemView - Dialog Activity

- Also assisted teammates in the following activities:
    - Manage Requests Screen
    - Registration Activity
    - Donor Registration Activity
    - Registration Authentication Activity
- Assigning & tracking tasks assigned to all team members.
- Helping team members in development of other activities.
- Worked along with the teammates for a proper database connectivity with Parse.com and ensured a correct data flow.
- Performed client side validations for the smooth responsive and interactive UI.
- Helped the team with Google Maps API feature for better and proper user interaction with the application.
- Performed Android Lollipop 5.0 implementations along with advanced UI features for smoother and interactive UI.
- Unit Testing, Peer review others code, bug fixes, and feedback.
- Peer review initial and final design documents & provide feedback.

## 7.3 Yeshwant Dasari: Android Developer & Database Developer

- Android UI Development for following screens:
    - Health Tips Screen
    - About B-Life Screen
    - Settings Screen
    - Edit Profile Screen
    - Edit Notifications Screen
- Android Activity development for the following activities:
    - HealthTipsActivity
    - AboutBLifeActivity
    - SettingsActivity
    - EditProfileActivity
    - EditNotificationsActivity
- JSON creation and parsing while sending and receiving data to and from the android app.
- Gathered all the requirements of the application and designed the initial prototype in photoshop which were later used as reference to built the application.
- Researched and Analyzed the best data modeling tool for the application and went ahead with Parse cloud platform which serves as PAAS (Platform as a service) for database hosting.
- Implemented data models in parse platform and wrote all the queries for a proper flow of data between application and the Parse platform.
- Customized the action bar in such a way that, donor related menu will be shown when logged in as donor or else receiver related menu will be shown.
- Maintained the user sessions so that user doesn’t have to login each time.
- Unit Testing, Peer review others code, bug fixes, and feedback.

## 7.4 Chaitanya Manyam: Android Developer

- Android UI Development for following screens:
  - Manage Requests Screen
  - Search Hospitals Screen
  - Search Blood Banks Screen
  - Helpline Numbers Screen
  - FAQs Screen
- Android Activity development for the following activities:
  - Manage Requests Screen
  - Search Hospitals Screen
  - Search Blood Banks Screen
  - Helpline Numbers Screen
  - FAQs Screen

- Researched and implemented the Yelp API for listing nearby hospitals, blood banks and eating joints nearer to your location.
- Researched content online for posting them in Frequently Asked Questions and Helpline Numbers.
- Implemented the SMS feature by using SMS Manager and generating a 4 digit number as a verification code.
- Implemented the Google Maps API for listing location of the selected hospital/blood bank/eat-out joint with navigation directions on the fragment.
- Interacted with Parse API for updating user's profile information and deleting blood requests.
- Implemented the GPS tracker functionality to find the user's device location for listing nearby locations.
- Implemented the Internet Availability function to check whether the device has GPS connectivity or connected to the wifi.
- Implemented data models in parse platform and wrote all the queries for a proper flow of data between application and the Parse platform.
- Unit Testing, Peer review others code, bug fixes, and feedback.

## 8. PROJECT MILESTONES

| **Deliverables** | **Date** |
|---|---|
| Project Proposal | February 9th 2015 |
| Initial Design Document | March 18th 2015 |
| Initial UI Design<br>Web Server Development environment set up<br>Server side Database design & implementation | March 30th 2015 |
| Initial Prototype presentation | April 6th 2015 |
| Refining & Testing of UI<br>Major database facing Activities development | April 13th 2015 |
| Web Service to handle JSON objects<br>Integration of Android app with the Web Server & database | April 20th 2015 |
| Presentation at AT&T | April 27th 2015 |
| Implementation of feedback from AT&T<br>Complete Application Development<br>Final testing of application | May 4th 2015 |
| Final Project Demo | May 11th 2015 |