

Insaniyat

FINAL YEAR PROJECT REPORT (CS-492)

BS(CS) Spring 2020

Anas Bin Faisal - 16K-4064

Syed Kumail Ali - 16K-4047

Abeer Zehra - 16K- 4068



Supervisor: Engr. Abdul Rahman

Co-supervisor: Mr. Murtaza Fazal

Department of Computer Science

**FAST-National University of Computer
& Emerging Sciences, Karachi.**

Sunday 7th June, 2020

Abstract

Karachi is a metropolitan city, with a large population. Majority is living below the line of poverty. Many people want to help the needy but due to their hectic schedules they are unable to do so and have a lack of donation system. On the other hand, everything is transforming towards automation. The services provided by the welfare organization are not easily accessible to this large audience. There is no such platform in Karachi where people could help the poor. About 3M tons of food is being wasted here every year. One of our aims is to prevent this wastage of food by serving this to the poor. We are developing an application which would serve to a large scale of users who could play their part in serving humanity.

Table of Contents

1	Introduction	6
1.1	Need For Product	6
1.2	Benefits to Users	6
1.3	Gap Analysis with existing solution	6
2	Project Description	7
3	Functional Requirements	9
3.1	Functional Hierarchy	9
3.2	Use Cases	9
3.2.1	Ambulance Service	10
3.2.2	Blood Donation	11
3.2.3	Morgue	12
3.2.4	Monetary Donation	12
3.2.5	Meal Donation	13
4	Non-functional Requirements	14
4.1	Performance Requirements	14
4.2	Safety Requirements	14
4.3	Security Requirements	14
4.4	User Documentation	15
5	Design Details	15
5.1	Database Design	15
5.1.1	ER Diagram	15
5.1.2	Data Dictionary	16
6	Application Design	23
6.1	Sequence Diagram	23

6.1.1	Ambulance Service	24
6.1.2	Blood Service	24
6.1.3	Morgue	25
6.1.4	Monetary Donation	26
6.1.5	Meal Donation	27
6.2	State Diagram	27
6.2.1	Ambulance Service	28
6.2.2	Blood Service	28
6.2.3	Morgue	29
6.2.4	Meal Donation	29
7	Implementation Details	30
7.1	System Constraints	30
7.2	System Level Architecture	31
7.3	Operating Environment	32
7.4	Assumption & Dependencies	32
7.5	External Interface Requirements	33
7.6	Hardware Interfaces	33
7.7	Software Interfaces	33
7.8	Communications Interfaces Document Convention	33
8	Screens	34
8.1	User	34
8.2	Volunteer	42
8.3	Ambulance Driver	44
9	Test Plan	47
9.1	Introduction	47
9.1.1	Purpose of The Test Plan Document	47
9.1.2	Test Approach	47

9.1.3	Test Pass/Fail Criteria	47
9.1.4	ENVIRONMENTAL NEEDS	47
9.1.5	Validation Testing	48
9.1.6	FUNCTIONAL TESTING	50

1 Introduction

1.1 Need For Product

In Pakistan there is not a single application where the people who are willing to help the needy can make their contributions just by using their smartphones. In this busy lifestyle people who have good will for the society fail to contribute because of this and the needy missout on such valuable contributions. In this era of technology where people prefer most of their time consuming tasks to be completed through their smartphones there is a desperate need for an application where people can take advantage of their smartphones to help the needy and contribute to the society. We ourselves as residents of Karachi feel the need of a platform through which we can make a welfare accessible and play our role in contributing to the society.

A mobile application was created to manage the workload of students and a wide majority of students reported their grades pick up because of the application.

1.2 Benefits to Users

This project would provide a platform for users to register and would be given an option if they are willing to work as a volunteer as well. The user would be able to connect with the welfare and other volunteers for the distribution of their donations. The users can use the services like food donation, general donation, ambulance service, and blood donation services etc.

1.3 Gap Analysis with existing solution

There exist different applications for different services but no single application where users can avail many services at a time. In our application users can avail many services at a time. Moreover, this single application also contains volunteer

Existing Applications	Ambulance	Blood Donation	Morgue	Monetary Donation	Meal Donation
Aahar	-	-	-	-	Yes
Daan Patra	-	-	-	-	Yes
Ambus	Yes	-	-	-	-
Naba Prabhat Orphanage	-	-	-	-	-
Blood Donor	-	Yes	-	-	-
Qatar Charity App	-	-	-	Yes	-
Sindh Ambulance 1036	Yes	-	-	-	-

service, where users can register them as a volunteer and can take help from registered volunteers. Given below is a chart of some existing applications and the services they are providing.

2 Project Description

Insaniyat app's modular description is as follows:-

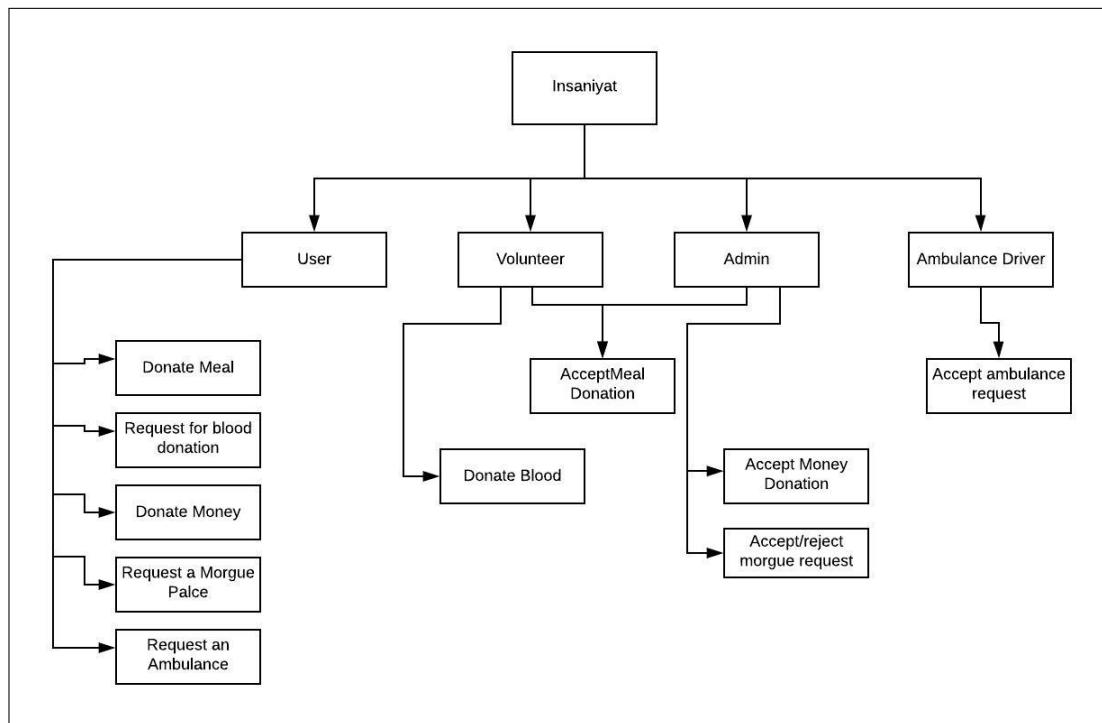
- Ambulance Service:** In this the user will be able to book an ambulance from the above stated welfares which provide ambulance service by sharing his/her location and the ambulance driver would be able to accept the request.
- Blood Donation:** In this feature the users who want blood donations for a specific blood group will be connected to the registered volunteers with the same blood group.
- Monetary Donation:** This feature will enable welfare organizations to receive money donation through the app. The current location where user will be standing while donating the money will be used
- Morgue:** This feature will enable the users that are registered to book the

morgue for their beloved deceased souls. User needs to enter the necessary details.

5. **Meal Donation:** In this feature the users can locate the dastarkhwans organized by the welfare and donate food which will be directly collected by the welfare's volunteer. The other option is to donate food to the volunteers registered on this application as per the donor's preference and volunteer would collect the meal from the user.
6. **General Donation:** In addition to money and meal, our app also consists of an option of general donation. There are times where people want to donate household things or things like that, so in this case we have general donation's option. User can enter the item type and the quantity of donation. Furthermore, he/she also have facility to opt between a volunteer and the organization. In the case of volunteer, all the registered volunteers will get the request and a text message will be send from the person that will accept the request to the donor and the same goes with the welfare organization. Current location where the donor will be standing while donating will be used to collect the donation.

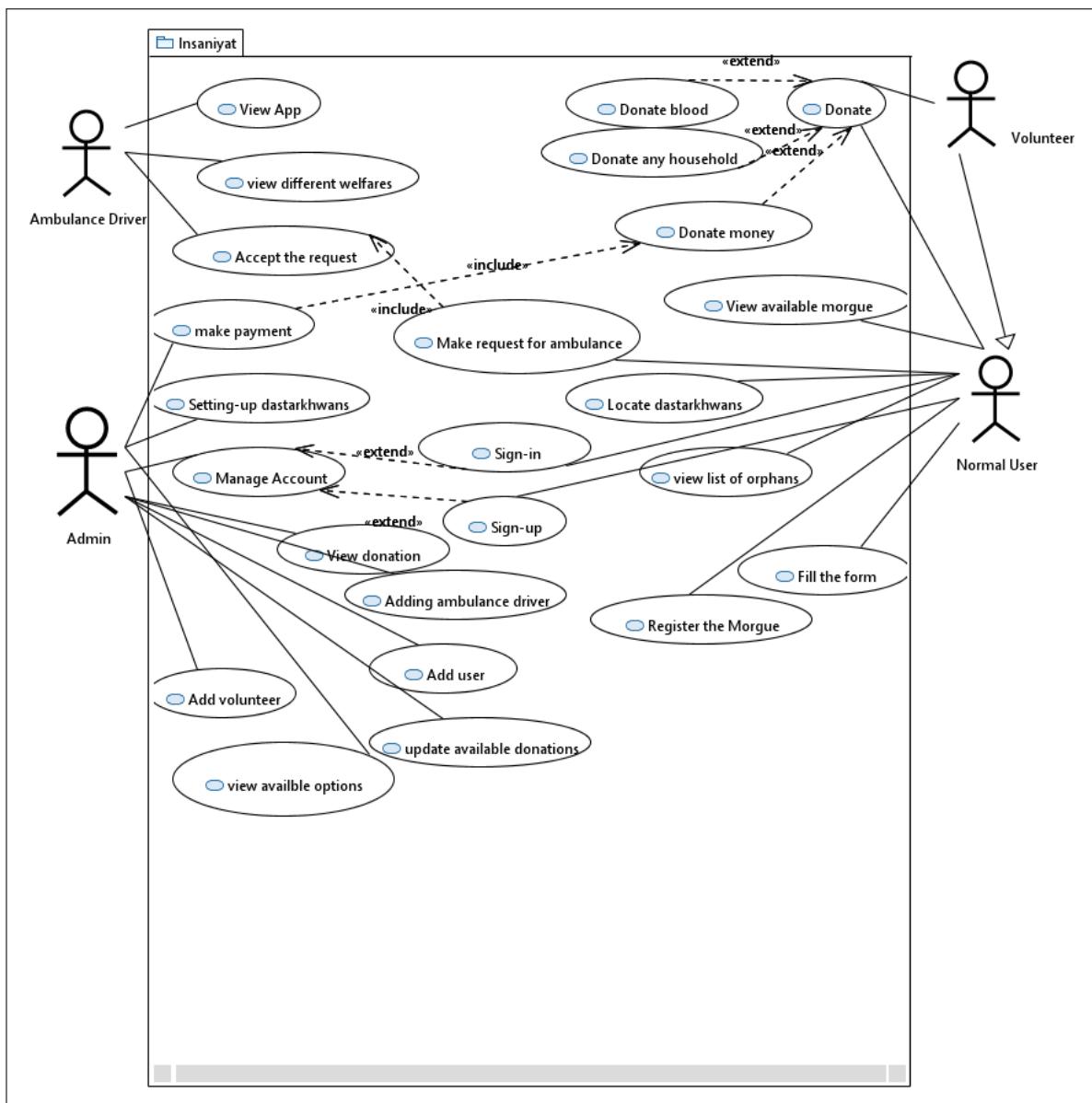
3 Functional Requirements

3.1 Functional Hierarchy



3.2 Use Cases

Use case diagram of insaniyat is as follows:



3.2.1 Ambulance Service

<Ambulance Services>		
Use case Id:	1	
Actors:	Ambulance driver, user	
Feature:	In this feature user can book the ambulance through our app and can track the ambulance.	
Pre-condition:	<ol style="list-style-type: none"> 1. Ambulance driver is registered. 2. User is registered. 3. Both have valid internet connection. 4. Both have Insaniyat a 	
Scenarios	In this the user will be able to book an ambulance from the above stated welfares which provide ambulance service by sharing his/her location and the ambulance driver would be able to accept the request and a fixed amount is payable on completion which the user can pay through our application as per their choice.	
Step#	Action	Software Reaction
1.	Registered user books an ambulance.	Ambulance driver's details will be shown
2.	Driver accepts the request	Notification pops-up
3.	User tracks the ambulance through the maps shown.	Maps will be shown

3.2.2 Blood Donation

<Blood donation>		
Use case Id:	2	
Actors:	User , Donar , admin	
Feature:	User who wants blood donation will upload the request. Same blood group registered users will be notified.	
Pre-condition:	1. Users should be registered.	
Scenarios	In this feature the users who want blood donations for a specific blood group will be connected to the registered volunteers with the same blood group.	
Step#	Action	Software Reaction
1.	Registered user will ask for donation.	
2.	User with same blood group will be notified.	Notification will pop-up
3.	Donar and the user will be connected.	

3.2.3 Morgue

<Morgue>		
Use case Id:	4	
Actors:	user,admin and welfare org. person	
Feature:	Registered user can book a morgue	
Pre-condition:	1. Valid internet connection 2. User is registered	
Scenarios	In this feature the users can register at the morgue for their beloved deceased souls	
Step#	Action	Software Reaction
1.	User will login	
2.	Search for nearest available morgue	
3.	Book the morgue	

3.2.4 Monetary Donation

<Monetary Donation>		
Use case Id:	5	
Actors:	User, Admin	
Feature:	In this feature the users can donate online to the welfare according to their preferences.	
Pre-condition:	User should have means of online transaction.	
Scenarios	The user will login through our website. He can donate through website.	
Step#	Action	Software Reaction
1.	User will open the app.	Homepage will be shown
2.	He will login his ID.	app will ask for username and password.
3.	Now he will donate by adding necessary details.	

3.2.5 Meal Donation

<Meal Donation>		
Use case Id:	6	
Actors:	Admin,meal donor,volunteer	
Feature:	In this feature the users can locate the dastarkhwans organized by the welfare and donate food which will be received directly by the welfare's volunteer or donate food to the volunteers registered on this application as per the donor's preference.	
Pre-condition:	1. There should be a dastarkhwan arranged nearby. 2. Person should be already registered if he/she wants to do volunteer work.	
Scenarios The person who wants to donate meal will sign-in through our app.He will search the near-by dastarkhwan and will check if he/she can go easily or not to donate food. If not then he will search for volunteer so that he can come and pick up the food.		
Step#	Action	Software Reaction
1.	Donar will sign-in through app.	Sign-in page will be shown
2.	He will check the near dastarkhwans.	Maps will be shown
3.	He will donate food by visiting that place.	
4.	If person cannot visit the place then he will ask a volunteer to get the food and donate at dastarkhwan.	

4 Non-functional Requirements

4.1 Performance Requirements

The performance factor of the application is immensely critical, like location identification through maps which need pin point accuracy and precision for the user to have a good experience of the application. The other point is the assignment of the ambulance driver to the user which needs to be closest among the other drivers, if performance lacks in algorithms for selecting the driver, then it could affect the safety or health of the patient. The reliability of this application depends upon how effective it is for the user in accomplishing their tasks.

4.2 Safety Requirements

Our project have user personal info such as id ,password and email address.We have added email verification, until or unless the email is not verified the account could not be logged in. Hence, email could not be missed used.

4.3 Security Requirements

The security factor is a great concern to the user of the application. In this application the data will be kept secure and the personal information of any actor would not be misused in any possible way. No user could sign up with an already existing login Id and during the time of user authentication, login Id and password would be verified through existing database records.

4.4 User Documentation

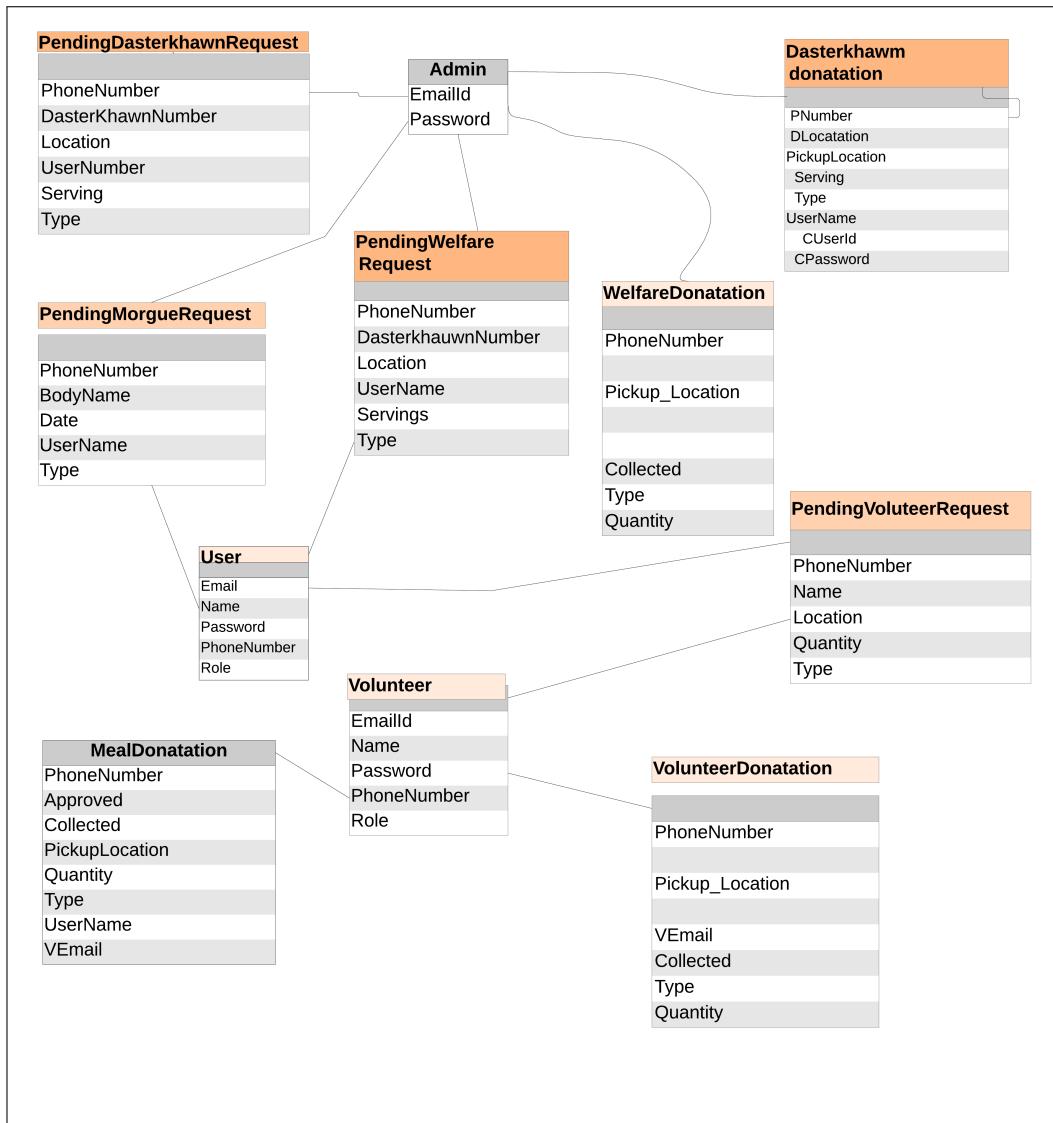
Following documentation would be given along with the software:

- User Manual
- System Requirement Specification (SRS)
- System Design Specification (SDS)

5 Design Details

5.1 Database Design

5.1.1 ER Diagram



5.1.2 Data Dictionary

< Data 1 >						
Name	User					
Alias	Write other names used for the first entry.					
Where-used/how-used	This data is used when Donor is logging in or making donations.					
Content description	User data records					
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Email_ID	Id of Donor	String	20	Not Null		Primary Key
Name	Name of Donor	String	25			
Password	Password for Sign In	String	25	Not Null		
Phone_Num	Phone String of Donor	String	11			
Role	Type of user	String	15			

Figure 1: User

< Data 2 >						
Name	Volunteer					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use the data when Volunteer is assigned a task to perform.					
Content description	Volunteer data records.					
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Email_ID	Id of Volunteer	String	20	Not Null		Primary Key
Name	Name of Volunteer	String	25			
Password	Password for Sign In	String	25	Not Null		
Phone_Nu m	Phone String of Volunteer	String	11			
Role	Is he a Volunteer	String	15			

Figure 2: Volunteer

< Data 3>	
Name	Admin
Alias	Write other names used for the first entry.
Where-used/how-used	Data is used when Admin when to access the application
Content description	Admin data records.

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Id_Login	Login Id for Sign In	String	25	Not Null		Primary key
Password	Password for Sign In	String	25	Not Null		

Figure 3: Admin

< Data 4>	
Name	Pending Morgue Requests
Alias	Write other names used for the first entry.
Where-used/how-used	Data is used when booking a morgue.
Content description	Morgue data records.

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
PhoneString	Phone String of user	String	11	Not Null		Primary Key
BodyName	Name of dead body	String	15			
Date	Date of death	Date	10			
UserName	Who has booked the morgue	String	15			
Type	Details of dead	String	10			

Figure 4: Pending Morgue Request

< Data 5>						
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Phone_Num	User's phone String	String	11	Not Null		Primary Key
D_Location	Name of Dastarkhwan	String	20	Not Null		
Pickup_Location	Location of user (long,lat)	Geopoint	20	Not Null		
Servings	String of servings for the people	String	4			
Type	Type of meal	String	20			
Username	User name	String	20			
Collected	Task completion status	boolean	5			
Approved	Pending request	boolean	5			

Figure 5:Dasterkhwan donation

< Data 6 >						
Name	Pending Dastarkhwan Requests					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use data when donor wants to donate something or for admin activities.					
Content description	Donation detailed data records..					

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
PhoneString	Phone String of user	String	11	Not Null		Primary Key
DastarkhwanName	Name	String	20			
Location	PickupLocation	Geopoint	20			
UserName	name of user	String	20			
Type	Type of donation	String	20			

Figure 6: Pending dasterkhwan request

< Data 7 >						
Name	Ambulance					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use data when user wants to book an ambulance or for admin privileges.					
Content description	Ambulance related data records.					

Column Name	Description	Type	Length	Nullable	Default Value	Key type
Id	Id of each Ambulance	String	4	Not Null	Auto Generate	Primary key
Phone_Nu m	Phone String of ambulance Driver	String	11			
Id_Login	Id for Login	String	25	Not Null		Primary key
Password	Password for Login	String	25	Not Null		
Status	Availability of ambulance	String	15			
Location	Location of Ambulance	Geopoint	50	Not Null		
Gender	Gender of driver	String	6			
Fare	Amount of booking an ambulance	String	4			

Figure 7: Ambulance

< Data 8>						
Name	Pending Volunteer Request					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use data when donor wants to donate food.					
Content description	Meal donation data records.					

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
PhoneString	Phone String of user	String	4	Not Null		Primary Key
Location	Location of user	Geopoint	20			
UserName	Name of user	String	20			
Quantity	Serving String of person	String	4			
Type	Type of donation	String	10			

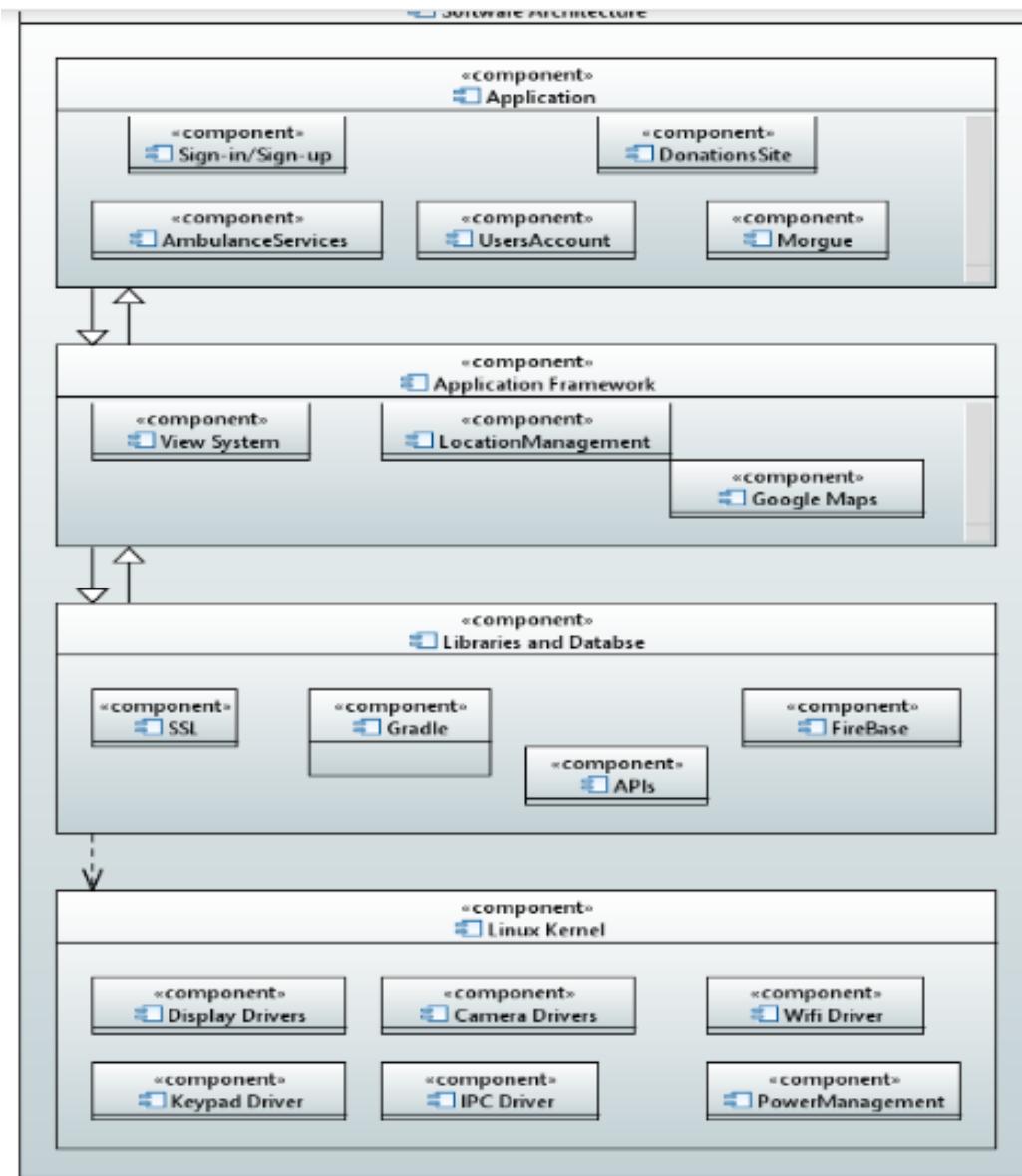
Figure 8: Pending volunteer request

< Data 9>						
Name	Blood Donation					
Alias	Write other names used for the first entry.					
Where-used/how-used	Used when donor wants to donate blood.					
Content description	Blood donation data records.					

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Id	Each Blood transaction has an individual Id	String	4	Not Null	Auto Generate	Primary key
Donar_Id	Id of blood donor	String	4			Foreign Key
Volunteer_Id	Id of volunteer, who will collect the blood	String	4			Foreign Key
Blood_Group	Type of Blood	String	3			
No_Of_Bottle	String of bottles of blood required	String	3			

Figure 9:Blood Donation

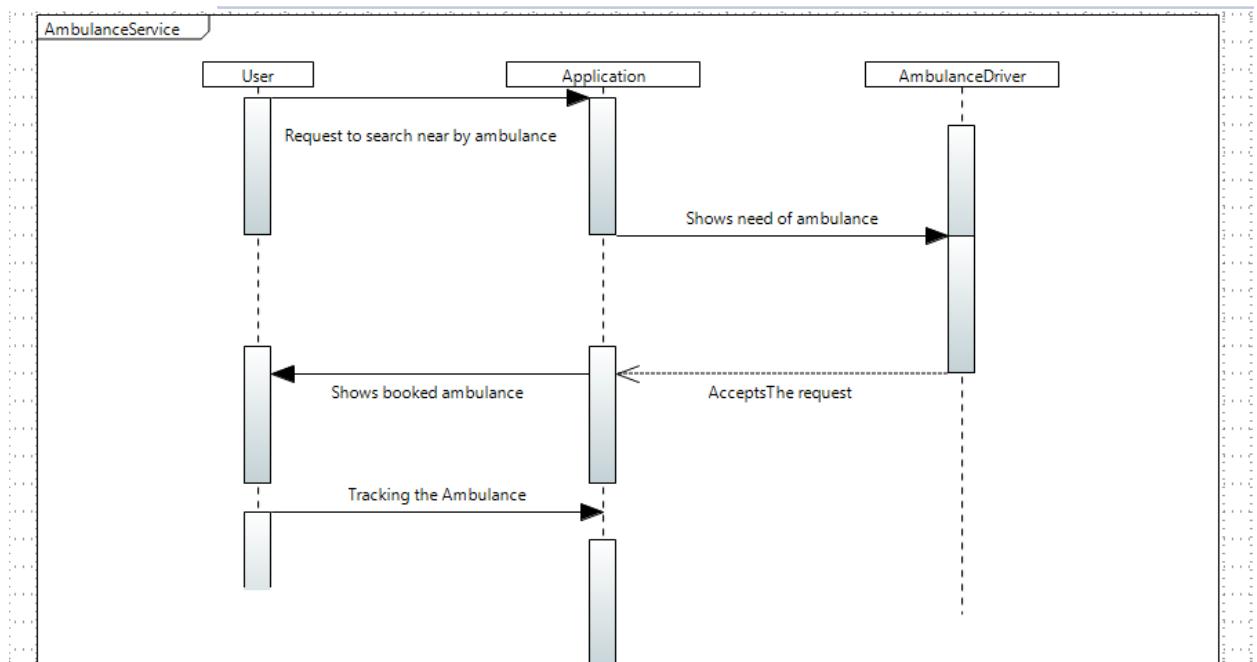
6 Application Design



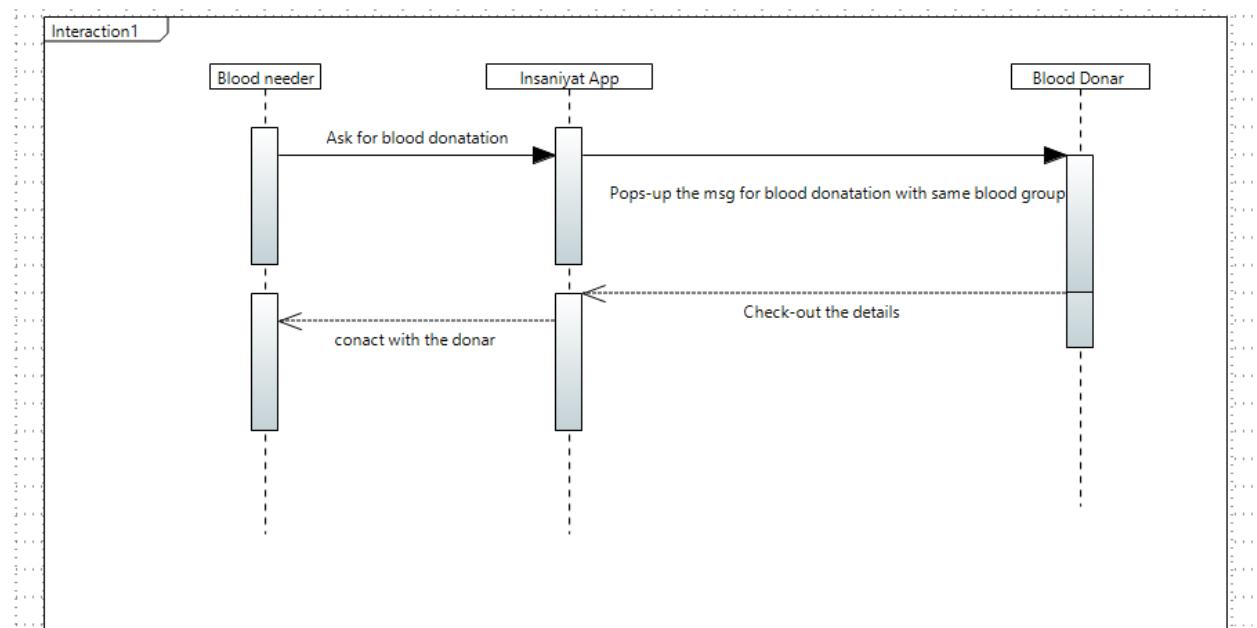
6.1 Sequence Diagram

Following are the sequence diagrams of all the features that we have implemented in our android application.

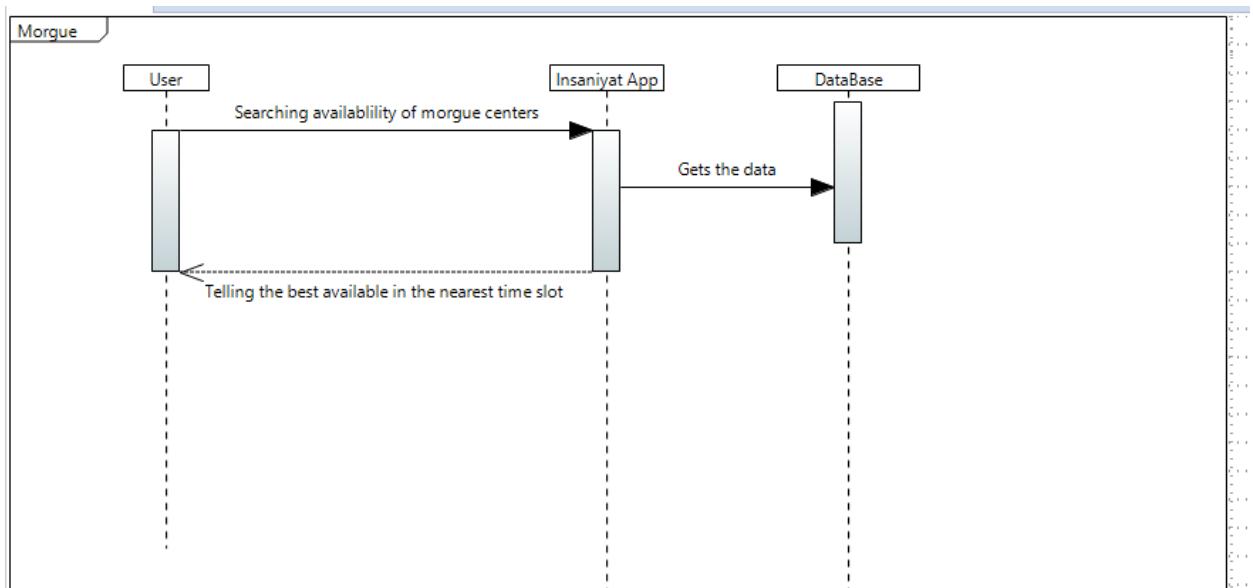
6.1.1 Ambulance Service



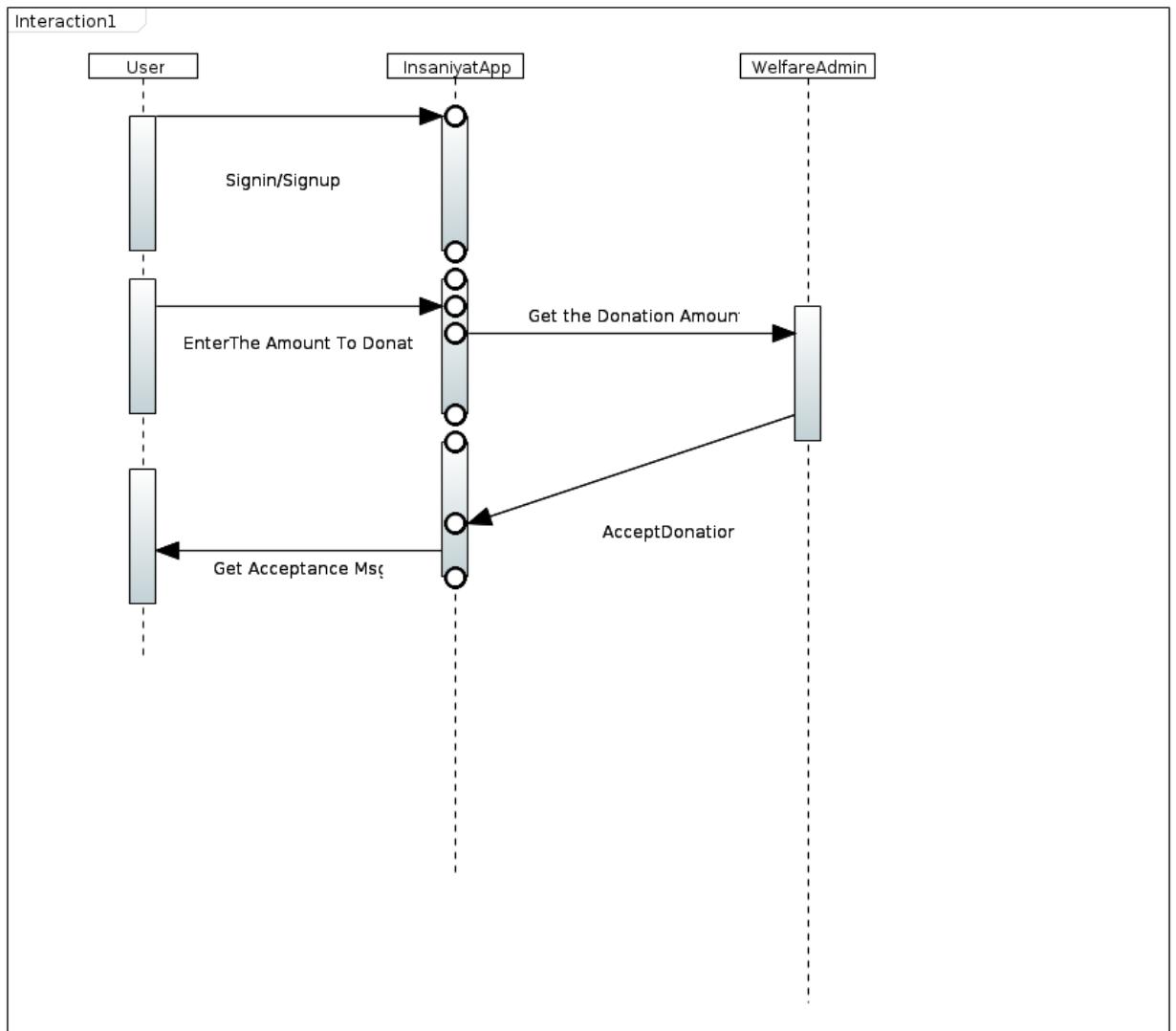
6.1.2 Blood Service



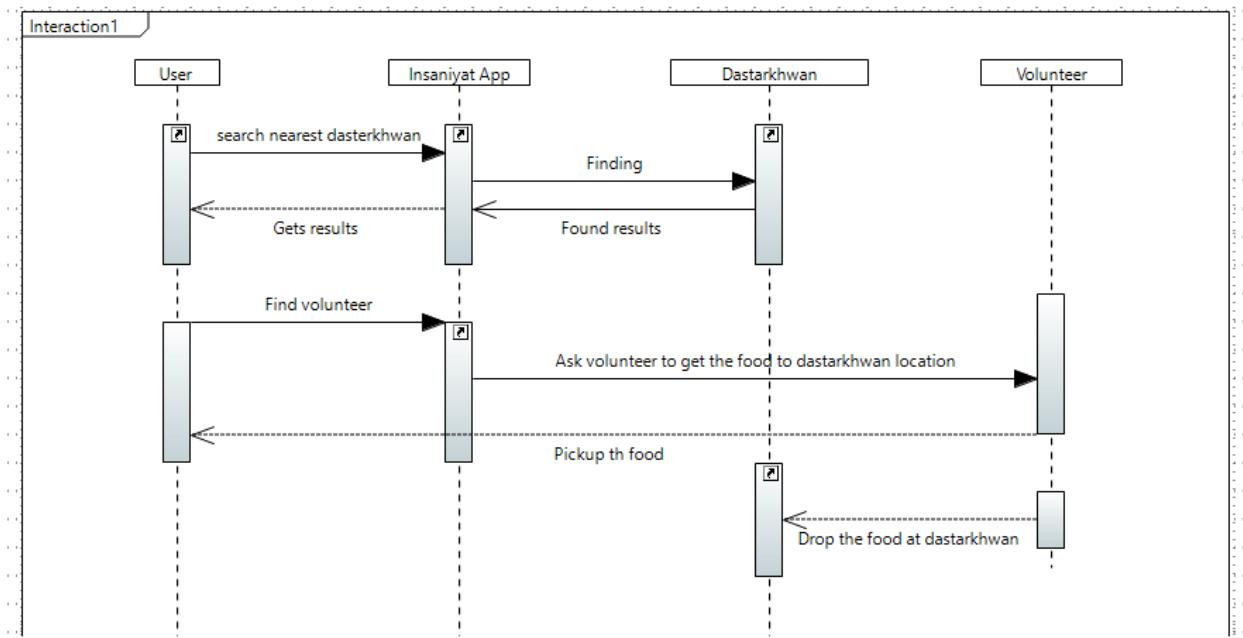
6.1.3 Morgue



6.1.4 Monetary Donation



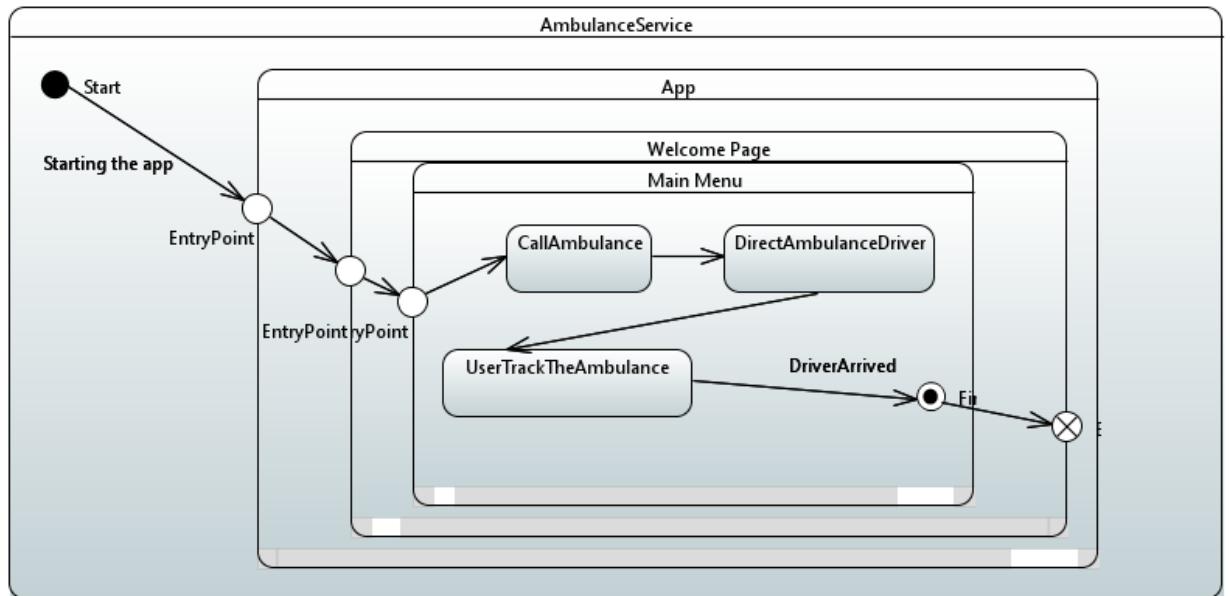
6.1.5 Meal Donation



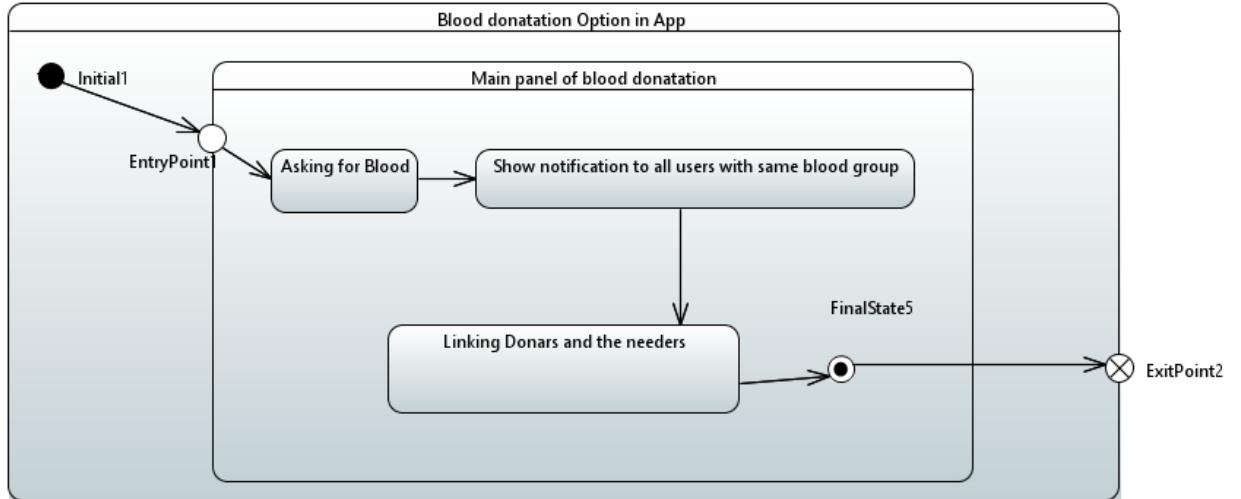
6.2 State Diagram

Here are state diagrams of all the features of our android application

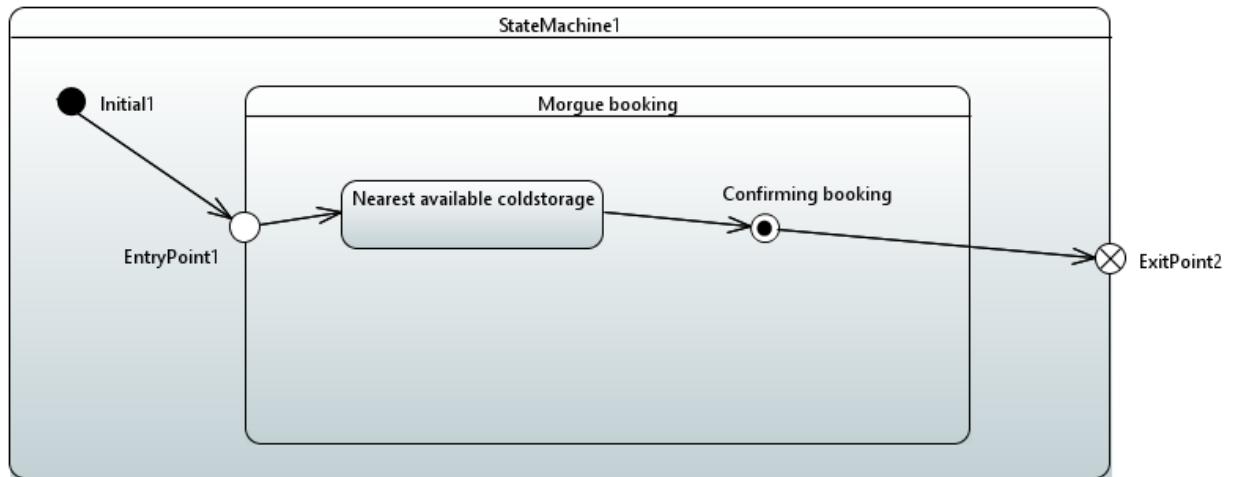
6.2.1 Ambulance Service



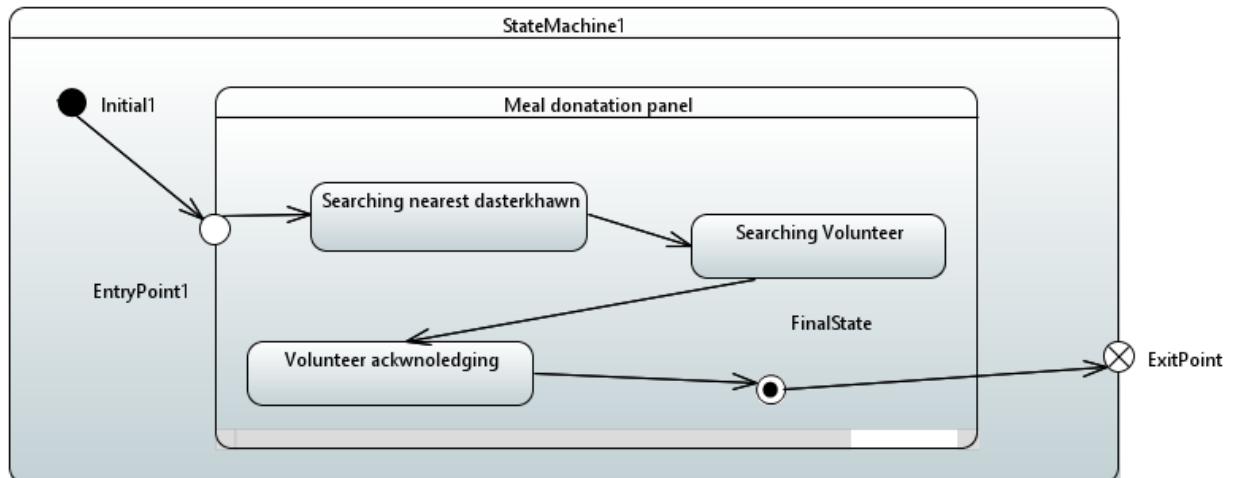
6.2.2 Blood Service



6.2.3 Morgue



6.2.4 Meal Donation



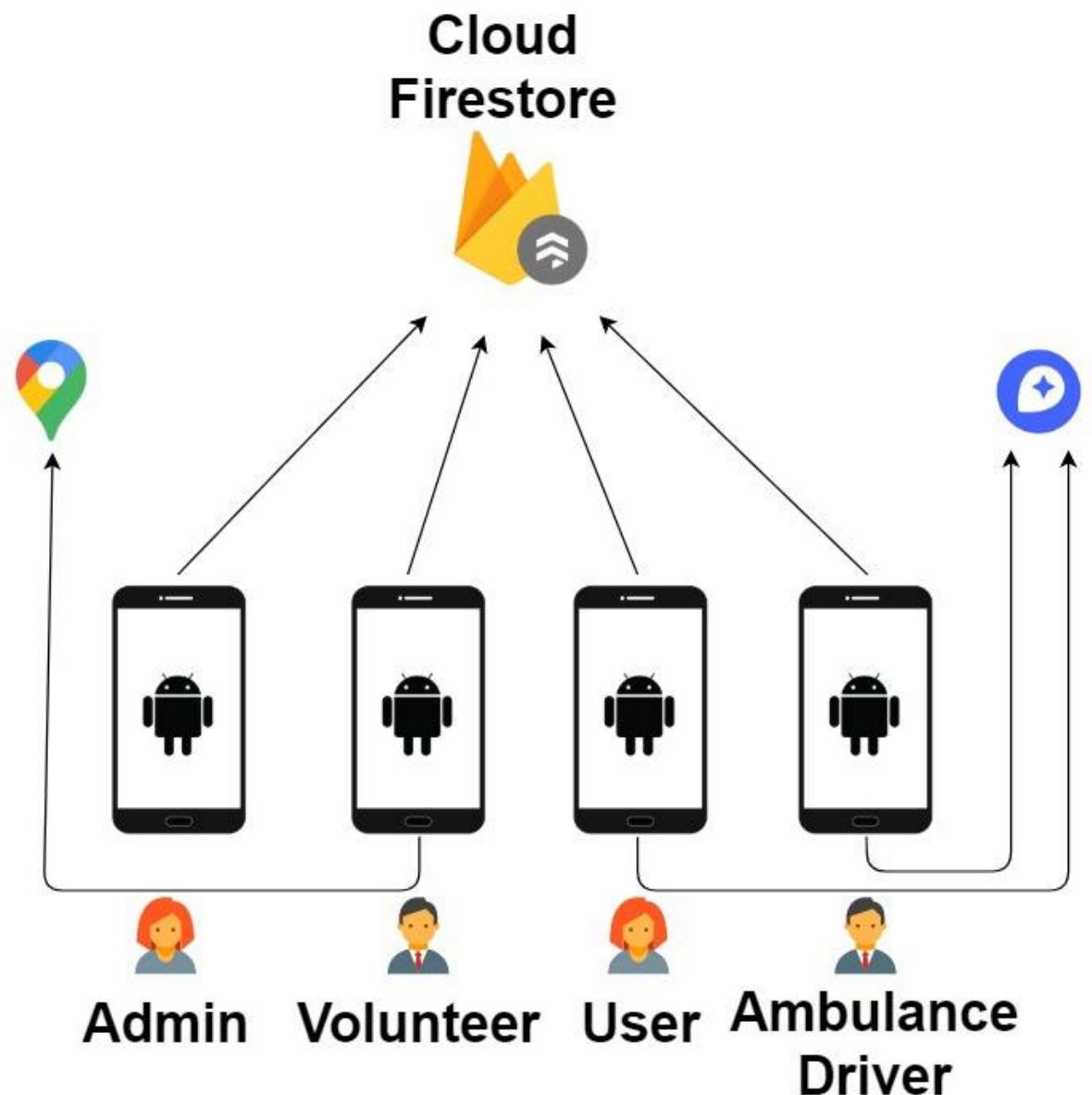
7 Implementation Details

7.1 System Constraints

External environment may be caused by the stakeholders, business conditions, technical issues, academic requirements etc and may include the following:

- Software constraints
 - The efficiency of access decreases as the data set becomes more and more fragmented.
 - The API should be available/functioning at all times or else the most important functionality would be affected.
- Hardware constraints
 - Android phone with API level of minimum 15.
- Cultural constraints
 - The application would only support english language. Urdu language support is not available for this application.
- Environmental constraints
 - The environment should not be bright enough that the screen display is not clear.

7.2 System Level Architecture



7.3 Operating Environment

This application would require an android smartphone with stable Wi-Fi connection or cellular network(GPRS). The GPS should be working as well for services which require location of the user.

7.4 Assumption & Dependencies

- Assumptions
 - Assuming that the user can operate a smartphone.
 - Assuming that the user can understand english to use the application and operate.
 - The user is willing to trust the application with monetary donation.
- Dependencies :
 - The user of the application must have a stable network connection.
 - The user the application must have a working GPRS to use the services.
 - Google Maps API â Utilized as mapping and geocoding engine for the application.
 - Firebase - Non-relational cloud based database would handle the load giving availability at all times.

7.5 External Interface Requirements

7.6 Hardware Interfaces

Need separate android phones for all the four actors which are user, admin, ambulance driver and volunteer. The android phone specifications follow below:

- 2 GB RAM or above
- 50 MB internal storage or above
- GPS enabled
- He/she must have a valid internet connection

7.7 Software Interfaces

To track the location of the users of the mobile application, we need to ask the location coordinates from the user in order to have the precise accuracy. We will also store the encrypted form of passwords to ensure secure data storage.

7.8 Communications Interfaces Document Convention

- User Login for accessing the mobile application.
- Email forwarding or text message forwarding for confirmation of blood donation or any other kind of donation.
- Electronic forms are being used for doing paper work of booking an orphan.
- Critical data transfer is kept secured by encryption.

8 Screens

8.1 User

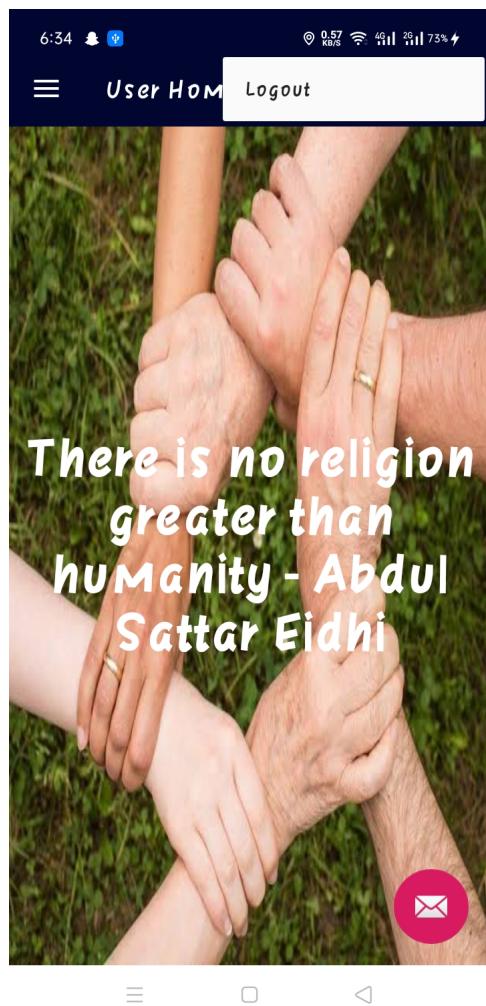


Figure 1: User Home Screen

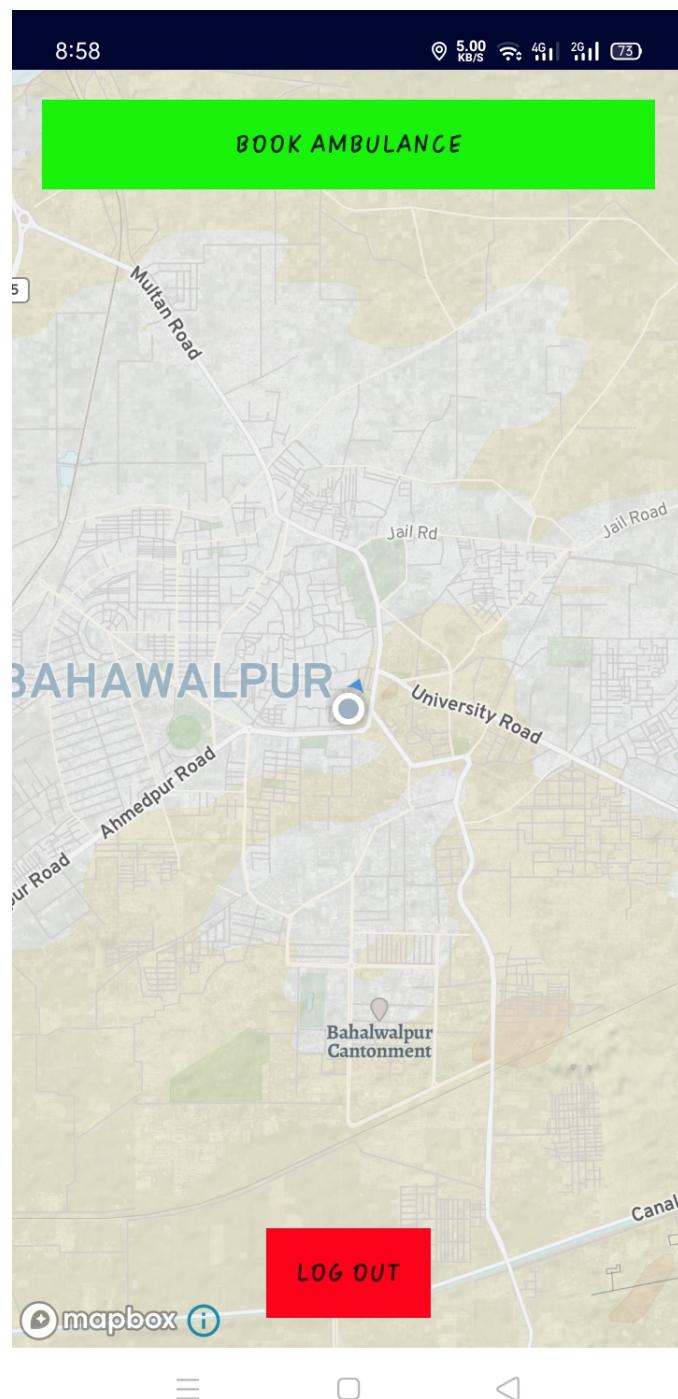


Figure 2: Book Ambulance



Figure 3: Meal Donation

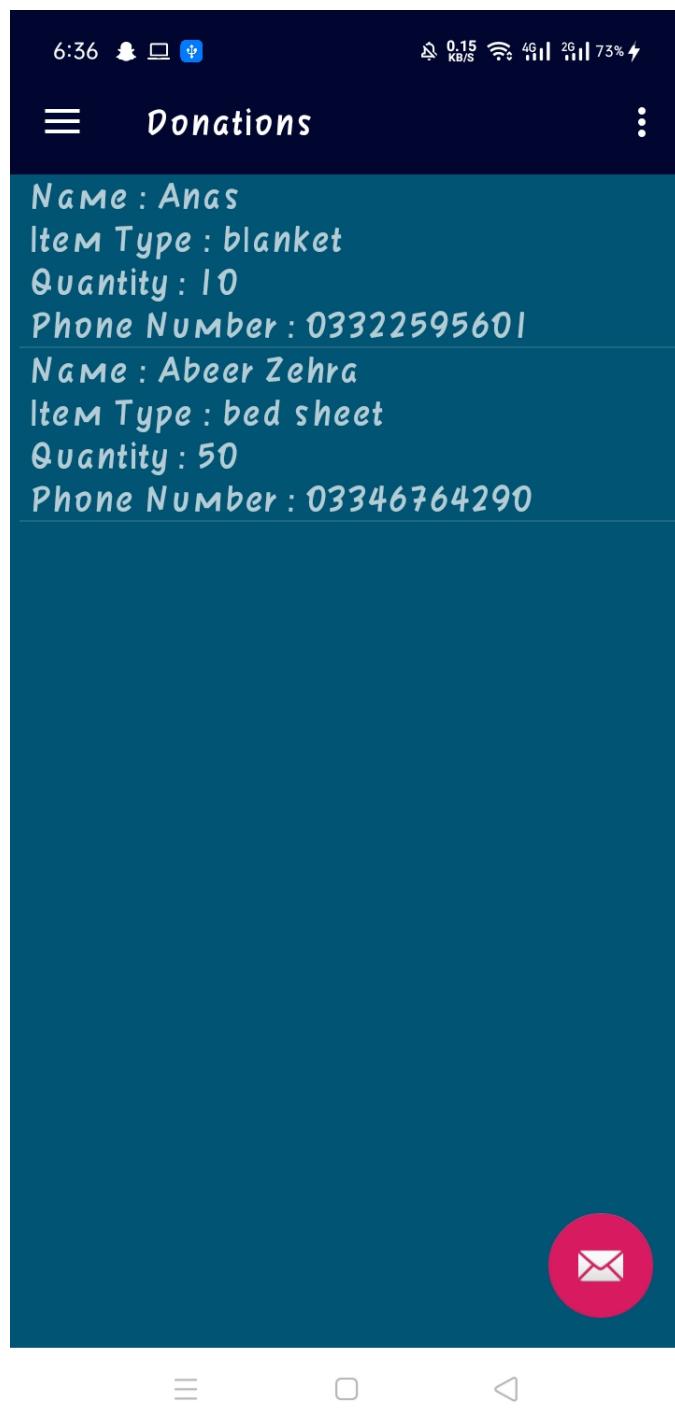


Figure 4: General Donation

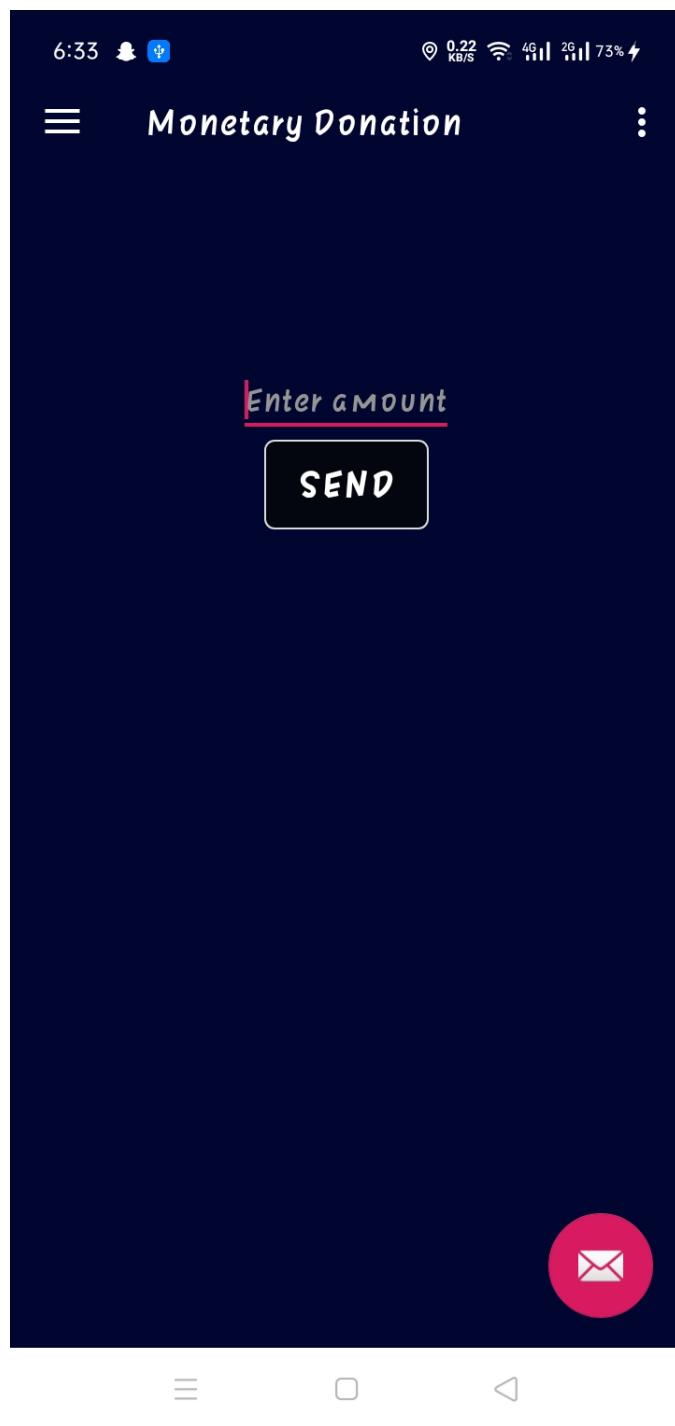


Figure 5: Monetary Donation

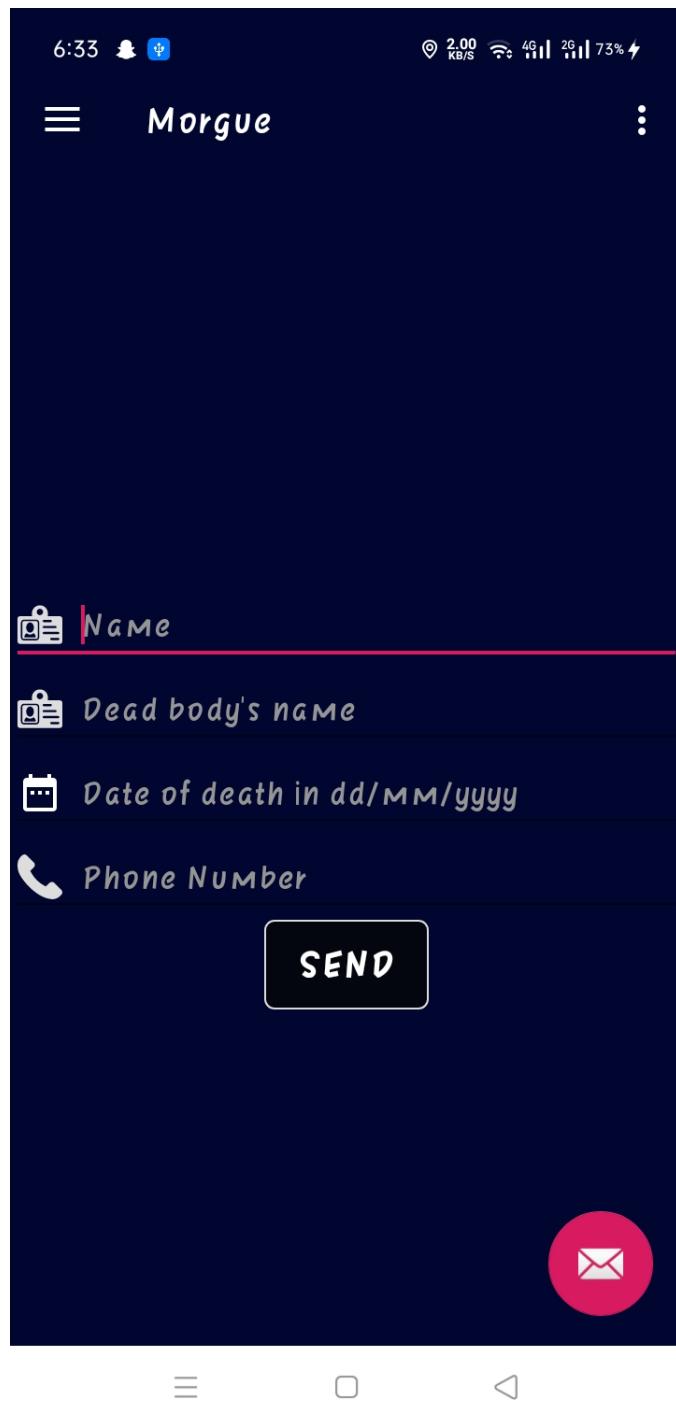


Figure 6: Morgue Request

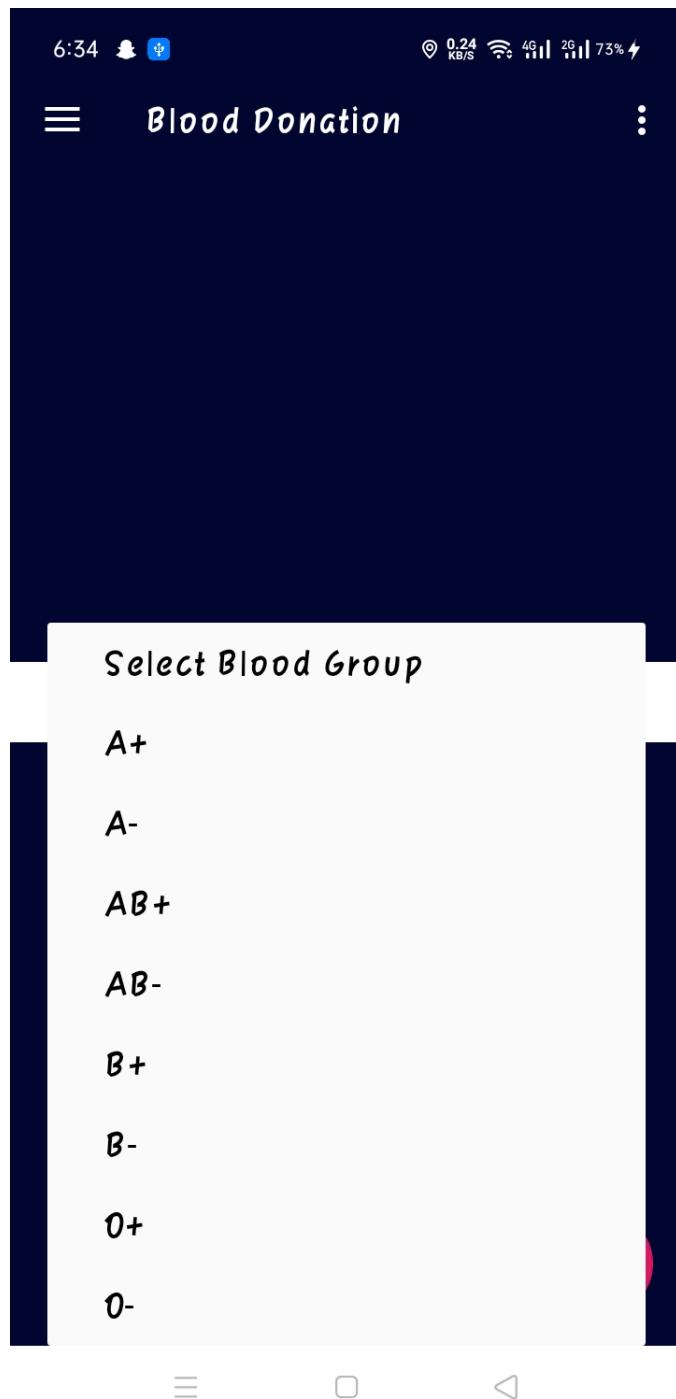


Figure 7: Blood Donation Request

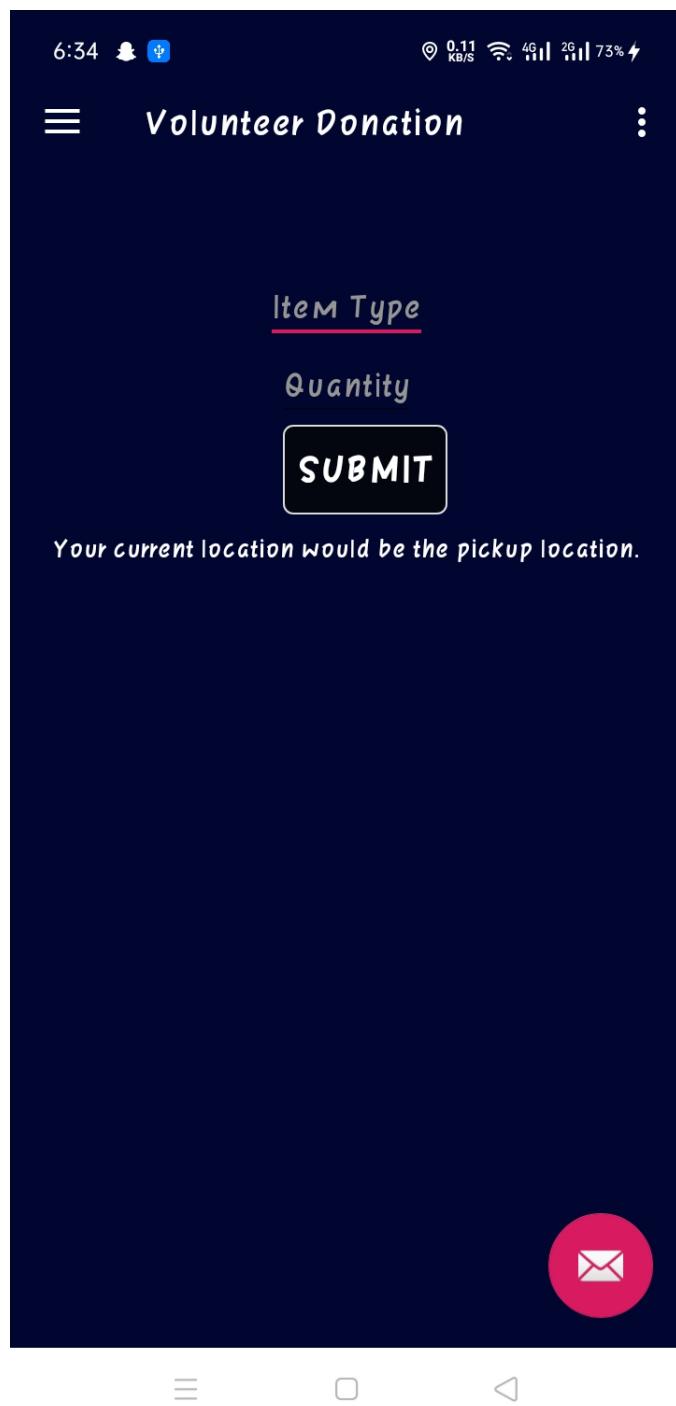


Figure 8: General Donation through volunteer

8.2 Volunteer

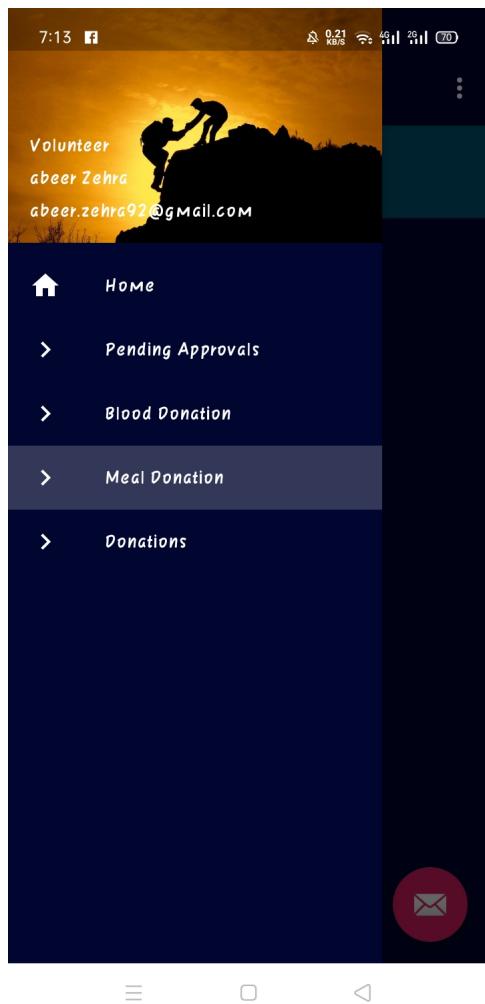
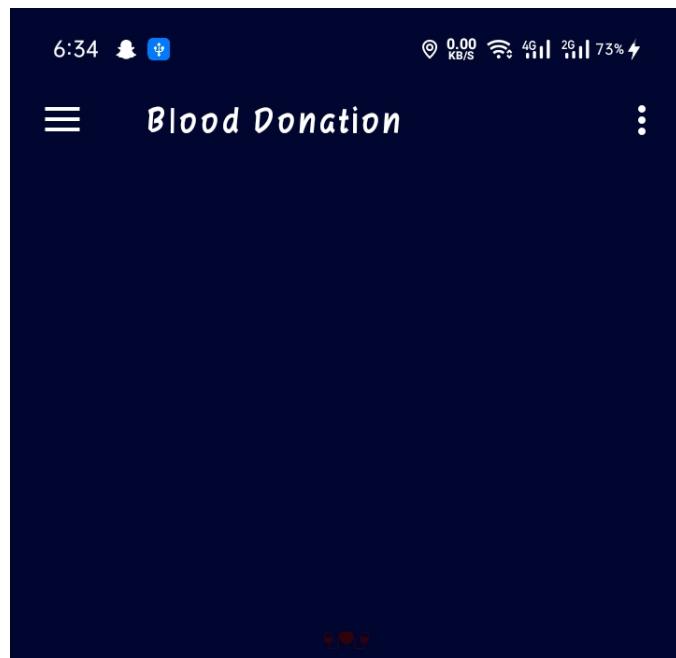


Figure 9: Volunteer



Select Blood Group

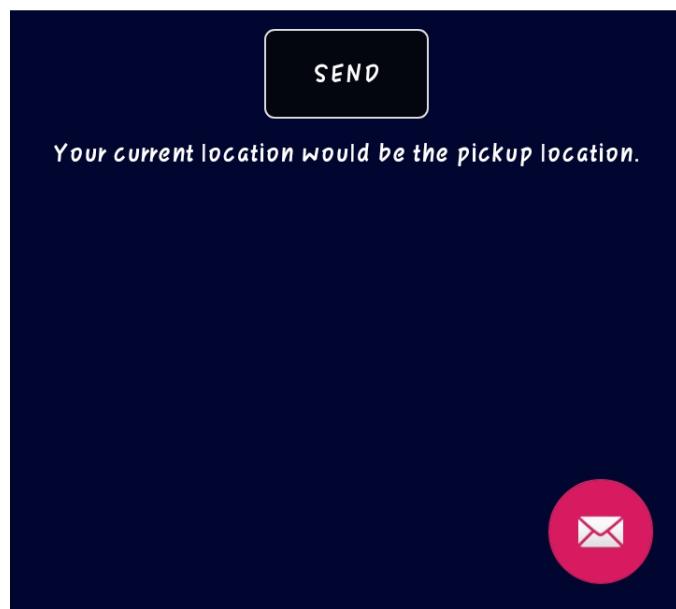


Figure 10: Blood Donation

8.3 Ambulance Driver

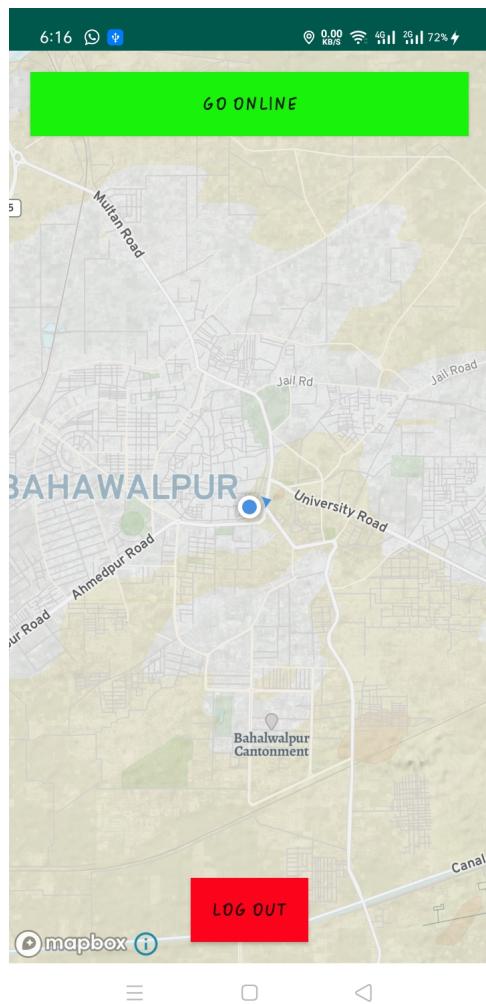


Figure 11: Ambulance Driver



Figure 12: Ambulance Request

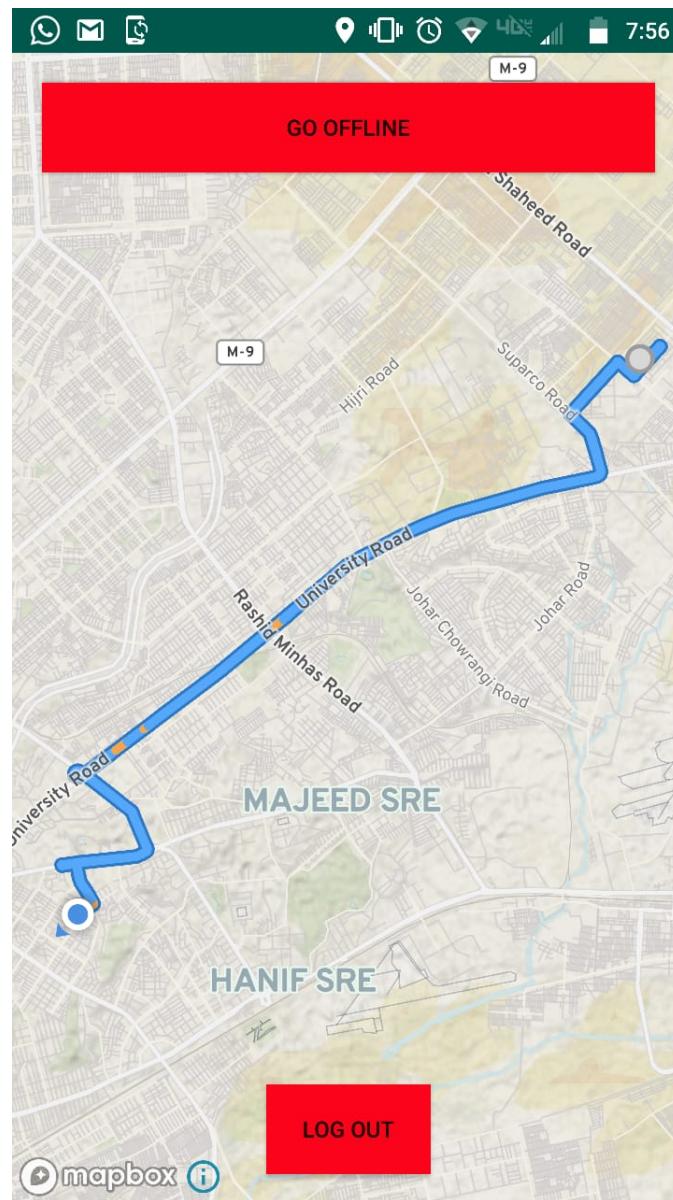


Figure 13: Tracking location

9 Test Plan

9.1 Introduction

9.1.1 Purpose of The Test Plan Document

Insaniyat- is an app that serve as a bridge between welfare organization and general public. Moreover, it also aims to minimize the wastage of food and provides the platform of volunteer where general public,welfare organization and volunteer can work together for the betterment of humanity.

9.1.2 Test Approach

There are two approaches for testing a software system i.e reactive and proactive. Reactive approach is used after designing and coding while proactive can be used before completion of coding of the system. So we are using reactive approach. When our application will be almost complete, we will start testing to identify bugs or issues in our application.

9.1.3 Test Pass/Fail Criteria

If 70 percent of users will be able to use our application smoothly, that means if they could understand application's color schemes and UI then our app will be considered as pass otherwise fail.

9.1.4 ENVIRONMENTAL NEEDS

- Users should have an android mobile phone with api level of atleast 6.
- Users needs to have a valid internet connection.
- GPRS should be in working condition to detect the location of user.

9.1.5 Validation Testing

ITEMS TO BE TESTED / NOT TESTED

We will be testing all the main features of our app. This will not include signin/signup and it's verification feature.

Item to Test	Test Description
DifferentButtonSize	Buttons should be easier to click.
SplashActivity	Activities that remains for few milliseconds are splash activity. These should be understood by user to fulfil our purpose.
TextColor	Text colors used should be visible to every users as our application will be used by almost all age groups.
BackgroundImages	If background images are understood by users and whether they could associate it with their purpose of use.
Icons	We will see if users can identify the meaning of icons used in the application.
NavigationDrawer	We will see if user could direct himself towards the navigation drawer.

TEST CASES

Test Case:-Text Color, Icons Description:- The user will be asked to install our application and check the visibility level of our application.		
No.	Steps	Expected Results
1.	The user will be asked to install the application on his mobile.	He will signin/signup through the application
2.	He will be asked to browse through the application randomly.	He will walk through the application.

Test Case 1

Test Case:-Navigation Drawer, Background Activities, Splash Activity. Description:- The user will be asked to browse through the application.		
No.	Steps	Expected Results
1.	The user will be asked to use the application.	He will signin/signup through the application
2.	The user will be asked to navigate through the navigation drawer	User will click top left most icon.

Test Case 2

9.1.6 FUNCTIONAL TESTING

ITEMS TO BE TESTED / NOT TESTED

We will be testing all the main features of our app. This will not include signin/signup and it's verification feature.

ItemtoTest	Test Description
Morgue	The user will be asked to request a place for morgue by entering the necessary details. This request will be send to admin. he also have to test the admin app at the same time.
GeneralDonation	The user will be asked to enter the item that he wants to donate and the quantity of it.
BloodDonation	The user will be asked to request the blood.This request will be send to the volunteer of matching blood group.If volunteer will accept, he will receive a conformation email.
MealDonation	The user will select the dasterkhuwan place.He will enter the quantity of food. If he sends this to volunteer then there will ba a notification in volunteer app and if he send to admin then notification in admin's app.
AmbulanceService	The user will be able to book nearest ambulance
MonetaryDonation	The user will asked to test this feature by entering the amount of money he wants to donate.

Test Case:- Ambulance service test
Description:- The user will be asked to test the ambulance service feature by using the application.

No.	Steps	Expected Results
1.	The user will be asked to book an ambulance service by click the button.	User will be able to send a request for an ambulance.
2.	Request will be send to the ambulance drivers.	Driver will receive the request.
3.	Driver will accept the request	User will be notified. And can see the drivers location on maps.

Test Case 3

Test Case:- General donation service test
Description:- The user will be asked to test the general donation feature by using the application.

No.	Steps	Expected Results
1.	The user will be asked to navigate through the navigation drawer to the general donation fragment.	He will be able to navigate.
2.	He will be asked to enter the amount .	He will enter the amount.
3.	He will then press the send button.	Verification message will be sent once the admin approves it.

Test Case 4

Test Case:- Meal donation test Description:- The user will be asked to test meal donation feature by using the application.		
No.	Steps	Expected Results
1.	We will ask user to open the app	The user will be able to signin/signup.
2.	He will be asked to open the meal donation module.	The user will be able to navigate through the drawer and select between donating to welfare or volunteer.
3.	Then he will be asked to select the place of dasterkhuwan if he is donating to the welfare and amount else just enter the item.	Amount and location entered will be recorded to collect the donation.
4.	Then he will finish donation by	Welfare/Volunteer will accept the request and confirmation message will be sent.

Test Case 5

Test Case:- Morgue service test Description:- The user will be asked to test the morgue feature by using the application.		
No.	Steps	Expected Results
1.	The User will be asked to select the morgue option from the navigation drawer.	The user will navigate to the morgue feature.
2.	He will be asked to enter all the details.	Details will be entered by the user.
3.	Admin will receive the request and upon situation accept/reject it.	User will receive message of acceptance/rejection.

Test Case 6

Test Case:- Blood donation feature test Description:- The user will be asked to test the blood donation feature by using the application.		
No.	Steps	Expected Results
1.	The user will be asked to test this feature by navigating through the blood donation.	He will be able to browse to the blood donation feature.
2.	He will be asked to select the blood group.	All the registered volunteer will be notified, and user will be notified from the one accepting the request.

Test Case 7

FYP Report

by Abeer Zehra

Submission date: 06-Jun-2020 10:58PM (UTC+0500)

Submission ID: 1316752607

File name: Insaniyat.pdf (4.82M)

Word count: 3003

Character count: 15934

Insaniyat

FINAL YEAR PROJECT REPORT (CS-492)

BS(CS) Spring 2020

Anas Bin Faisal - 16K-4064

Syed Kumail Ali - 16K-4047

Abeer Zehra - 16K- 4068



Supervisor: Engr. Abdul Rahman

Co-supervisor: Mr. Murtaza Fazal

13

Department of Computer Science

**FAST-National University of Computer
& Emerging Sciences, Karachi.**

Saturday 6th June, 2020

Abstract

Karachi is a metropolitan city, with a large population. Majority is living below the line of poverty. Many people want to help the needy but due to their hectic schedules they are unable to do so and have a lack of donation system. On the other hand, everything is transforming towards automation. The services provided by the welfare organization are not easily accessible to this large audience. There is no such platform in Karachi where people could help the poor. About 3M tons of food is being wasted here every year. One of our aims is to prevent this wastage of food by serving this to the poor. We are developing an application which would serve to a large scale of users who could play their part in serving humanity.

Table of Contents

1	Introduction	6
1.1	Need For Product	6
1.2	Benefits to Users	6
1.3	Gap Analysis with existing solution	6
2	Project Description	7
3	Functional Requirements	9
3.1	Functional Hierarchy	9
3.2	Use Cases	9
3.2.1	Ambulance Service	10
3.2.2	Blood Donation	11
3.2.3	Morgue	12
3.2.4	Monetary Donation	12
3.2.5	Meal Donation	13
4	Non-functional Requirements	14
4.1	Performance Requirements	14
4.2	Safety Requirements	14
4.3	Security Requirements	14
4.4	Security Requirements	15
4.5	User Documentation	16
5	Design Details	16
5.1	Database Design	16
5.1.1	ER Diagram	16
5.1.2	Data Dictionary	17
6	Application Design	24

6.1 Sequence Diagram	24
6.1.1 Ambulance Service	25
6.1.2 Blood Service	25
6.1.3 Morgue	26
6.1.4 Monetary Donation	26
6.1.5 Meal Donation	27
6.2 State Diagram	27
6.2.1 Ambulance Service	28
6.2.2 Blood Service	28
6.2.3 Morgue	29
6.2.4 Monetary Donation	29
6.2.5 Meal Donation	30
7 Implementation Details	31
7.1 System Constraints	31
7.2 System Level Architecture	32
7.3 Operating Environment	33
7.4 Assumption & Dependencies	33
7.5 External Interface Requirements	34
7.6 Hardware Interfaces	34
7.7 Software Interfaces	34
7.8 Communications Interfaces Document Convention	34
8 Screens	35
8.1 User	35
8.2 Volunteer	43
8.3 Ambulance Driver	45
9 Test Plan	48
9.1 Introduction	48

17		
9.1.1	Purpose of The Test Plan Document	48
9.1.2	Test Approach	48
8	9.1.3 Test Pass/Fail Criteria	48
	9.1.4 ENVIRONMENTAL NEEDS	48
	9.1.5 Validation Testing	49
	9.1.6 FUNCTIONAL TESTING	51

1 Introduction

1.1 Need For Product

In Pakistan there is not a single application where the people who are willing to help the needy can make their contributions just by using their smartphones. In this busy lifestyle people who have good will for the society fail to contribute because of this and the needy missout on such valuable contributions. In this era of technology where people prefer most of their time consuming tasks to be completed through their smartphones there is a desperate need for an application where people can take advantage of their smartphones to help the needy and contribute to the society. We ourselves as residents of Karachi feel the need of a platform through which we can make a welfare accessible and play our role in contributing to the society.

A mobile application was created to manage the workload of students and a wide majority of students reported their grades pick up because of the application.

1.2 Benefits to Users

This project would provide a platform for users to register and would be given an option if they are willing to work as a volunteer as well. The user would be able to connect with the welfare and other volunteers for the distribution of their donations. The users can use the services like food donation, general donation, ambulance service, and blood donation services etc.

1.3 Gap Analysis with existing solution

There exist different applications for different services but no single application where users can avail many services at a time. In our application users can avail many services at a time. Moreover, this single application also contains volunteer

Existing Applications	Ambulance	Blood Donation	Morgue	Monetary Donation	Meal Donation
Aahar	-	-	-	-	Yes
Daan Patra	-	-	-	-	Yes
Ambus	Yes	-	-	-	-
Naba Prabhat Orphanage	-	-	-	-	-
Blood Donor	-	Yes	-	-	-
Qatar Charity App	-	-	-	Yes	-
Sindh Ambulance 1036	Yes	-	-	-	-

service, where users can register them as a volunteer and can take help from registered volunteers. Given below is a chart of some existing applications and the services they are providing.

2 Project Description

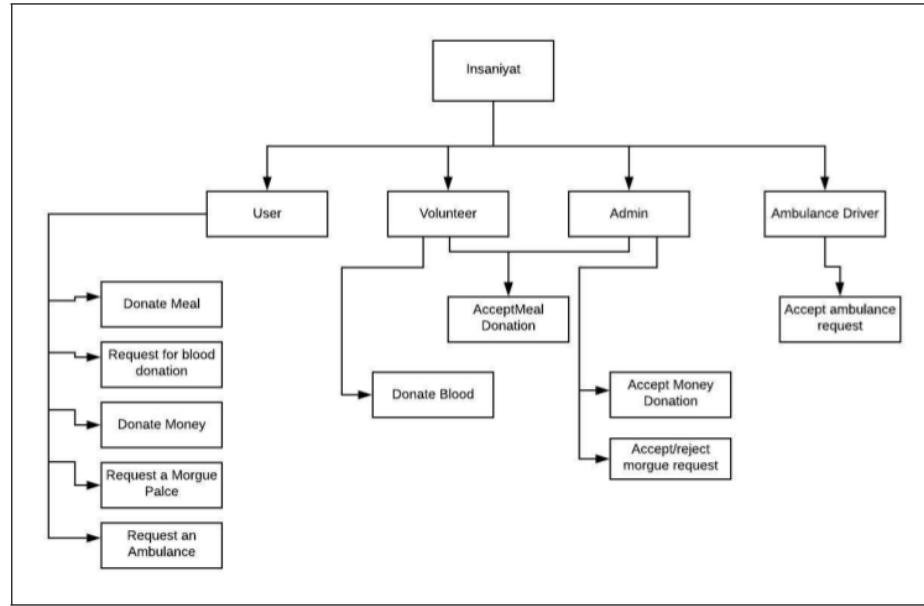
Insaniyat app's modular description is as follows:-

- Ambulance Service:** In this the user will be able to book an ambulance from the above stated welfares which provide ambulance service by sharing his/her location and the ambulance driver would be able to accept the request and a fixed amount is payable on completion which the user can pay through cash.
- Blood Donation:** In this feature the users who want blood donations for a specific blood group will be connected to the registered volunteers with the same blood group.
- Monetary Donation:** This feature will enable welfare organizations to receive money donation through the app. The current location where user will be standing while donating the money will be used

4. **Morgue:** This feature will enable the users that are registered to book the morgue for their beloved deceased souls. User needs to enter the necessary details.
5. **Meal Donation:** In this feature the users can locate the dastarkhwans organized by the welfare and donate food which will be directly collected by the welfare's volunteer. The other option is to donate food to the volunteers registered on this application as per the donor's preference and volunteer would collect the meal from the user.
6. **General Donation:** In addition to money and meal, our app also consists of an option of general donation. There are times where people want to donate household things or things like that, so in this case we have general donation's option. User can enter the item type and the quantity of donation. Furthermore, he/she also have facility to opt between a volunteer and the organization. In the case of volunteer, all the registered volunteers will get the request and a text message will be send from the person that will accept the request to the donor and the same goes with the welfare organization. Current location where the donor will be standing while donating will be used to collect the donation.

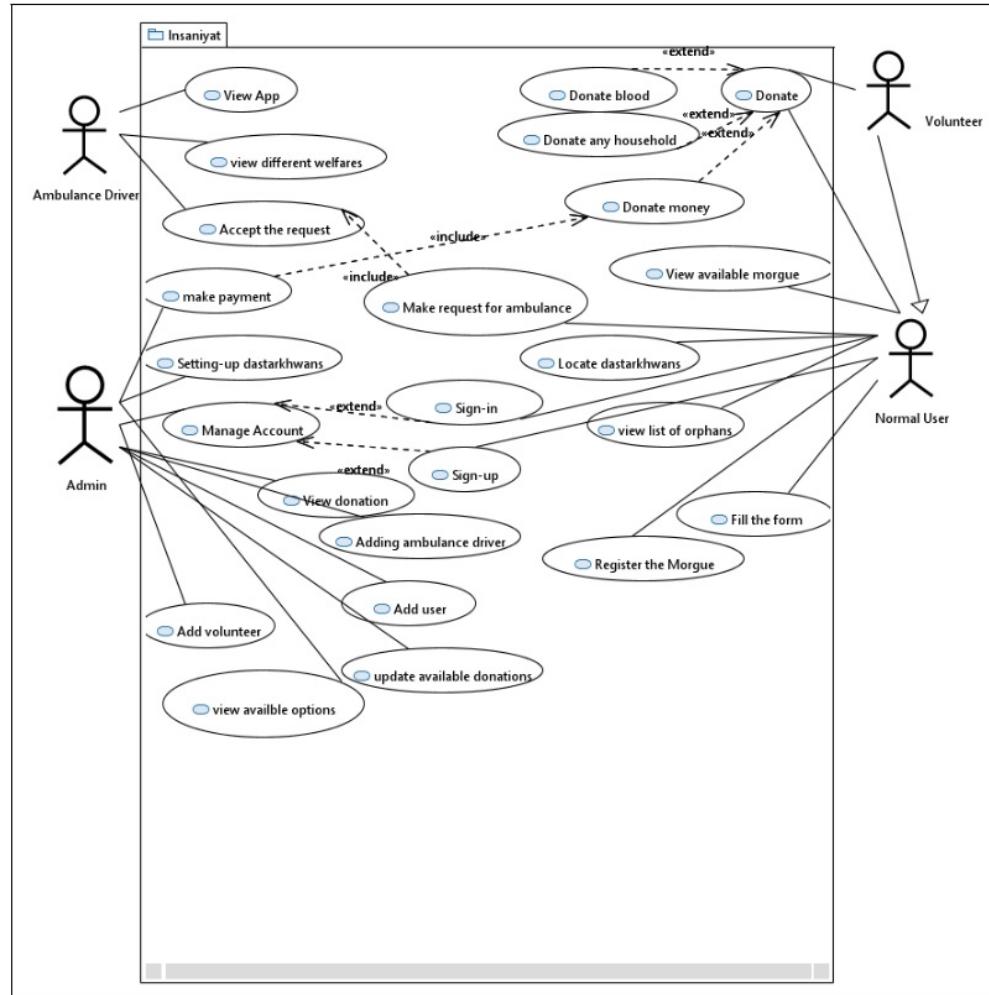
3 Functional Requirements

3.1 Functional Hierarchy



3.2 Use Cases

Use case diagram of insaniyat is as follows:



3.2.1 Ambulance Service

<Ambulance Services>		
Use case Id:	1	
Actors:	Ambulance driver, user	
Feature:	In this feature user can book the ambulance through our app and can track the ambulance.	
Pre-condition:	1. Ambulance driver is registered. 2. User is registered. 3. Both have valid internet connection. 4. Both have Insaniyat a	
Scenarios	In this the user will be able to book an ambulance from the above stated welfares which provide ambulance service by sharing his/her location and the ambulance driver would be able to accept the request and a fixed amount is payable on completion which the user can pay through our application as per their choice.	
Step#	Action	Software Reaction
1.	Registered user books an ambulance.	Ambulance driver's details will be shown
2.	Driver accepts the request	Notification pops-up
3.	User tracks the ambulance through the maps shown.	Maps will be shown

3.2.2 Blood Donation

<Blood donation>		
Use case Id:	2	
Actors:	User , Donar , admin	
Feature:	User who wants blood donation will upload the request. Same blood group registered users will be notified.	
Pre-condition:	1. Users should be registered.	
Scenarios	In this feature the users who want blood donations for a specific blood group will be connected to the registered volunteers with the same blood group.	
Step#	Action	Software Reaction
1.	Registered user will ask for donation.	
2.	User with same blood group will be notified.	Notification will pop-up
3.	Donar and the user will be connected.	

3.2.3 Morgue

<Morgue>		
Use case Id:	4	
Actors:	user,admin and welfare org, person	
Feature:	Registered user can book a morgue	
Pre-condition:	1. Valid internet connection 2. User is registered	
Scenarios	In this feature the users can register at the morgue for their beloved deceased souls	
Step#	Action	Software Reaction
1.	User will login	
2.	Search for nearest available morgue	
3.	Book the morgue	

3.2.4 Monetary Donation

<Monetary Donation>		
Use case Id:	5	
Actors:	User, Admin	
Feature:	In this feature the users can donate online to the welfare according to their preferences.	
Pre-condition:	User should have means of online transaction.	
Scenarios	The user will login through our website. He can donate through website.	
Step#	Action	Software Reaction
1.	User will open the app.	Homepage will be shown
2.	He will login his ID.	app will ask for username and password.
3.	Now he will donate by adding necessary details.	

3.2.5 Meal Donation

<Meal Donation>		
Use case Id:	6	
Actors:	Admin, meal donor, volunteer	
Feature: In this feature the users can locate the <u>dastarkhwans</u> organized by the welfare and donate food which will be received directly by the welfare's volunteer or donate food to the volunteers registered on this application as per the donor's preference.		
Pre-condition:	<ol style="list-style-type: none">1. There should be a dastarkwan arranged nearby.2. Person should be already registered if he/she wants to do volunteer work.	
Scenarios The person who wants to donate meal will sign-in through our app. He will search the near-by dastarkwan and will check if he/she can go easily or not to donate food. If not then he will search for volunteer so that he can come and pick up the food.		
Step#	Action	Software Reaction
1.	Donar will sign-in through app.	Sign-in page will be shown
2.	He will check the near dastarkhwans.	Maps will be shown
3.	He will donate food by visiting that place.	
4.	If person cannot visit the place then he will ask a volunteer to get the food and donate at dastarkwan.	

14

4 Non-functional Requirements

4.1 Performance Requirements

The performance factor of the application is immensely critical, like location identification through maps which need pin point accuracy and precision for the user to have a good experience of the application. The other point is the assignment of the ambulance driver to the user which needs to be closest among the other drivers, if performance lacks in algorithms for selecting the driver, then it could affect the safety or health of the patient. The reliability of this application depends upon how effective it is for the user in accomplishing their tasks.

4.2 Safety Requirements

Our project will have user personal info such as id ,password,email address and courses taken this semester.This data of lost could lead in students not able to track their progress of workload they had do do.We therefore plan to use the local cache of the phone a secondary backup of our the user data so that the in case of any server crash or hack leading to loss of data we still will have a backup inside the users personal phone.

4.3 Security Requirements

The safety factor is very vital for any application to prosper or develop in further dimensions. Taking an example of our application, we are ensuring that the identification of the ambulance driver is being done beforehand, so that all the drivers are verified and no problem occurs for the user of the application.

4.4 Security Requirements

The security factor is a great concern to the user of the application. In this application the data will be kept secure and the personal information of any actor would not be misused in any possible way. This application will also make sure to keep sensitive data like amount donated and credit card details safe as it is the basic right of the user. No user could sign up with an already existing login Id and during the time of user authentication, login Id and password would be verified through existing database records.

4.5 User Documentation

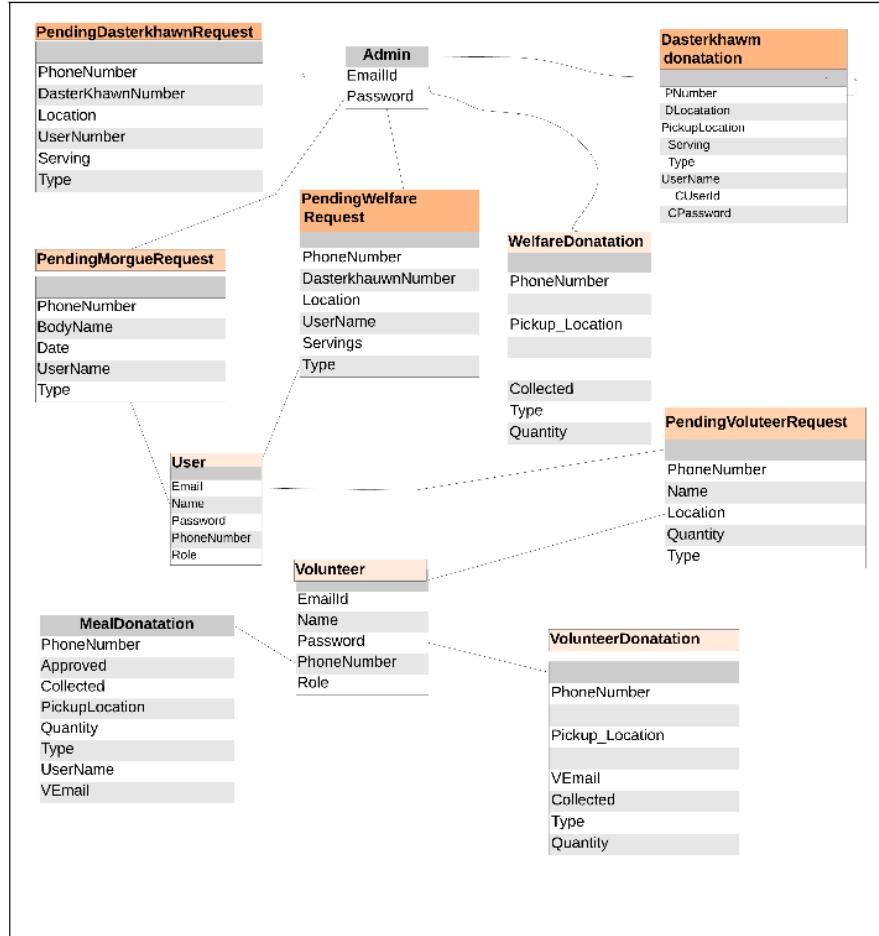
Following documentation would be given along with the software:

- User Manual
- System Requirement Specification (SRS)
- System Design Specification (SDS)

5 Design Details

5.1 Database Design

5.1.1 ER Diagram



5.1.2 Data Dictionary

< Data 1 >						
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Email_ID	Id of Donor	String	20	Not Null		Primary Key
Name	Name of Donor	String	25			
Password	Password for Sign In	String	25	Not Null		
Phone_Num	Phone String of Donor	String	11			
Role	Type of user	String	15			

Figure 1: User

< Data 2 >						
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Email_ID	Id of Volunteer	String	20	Not Null		Primary Key
Name	Name of Volunteer	String	25			
Password	Password for Sign In	String	25	Not Null		
Phone_Nu	Phone String of Volunteer	String	11			
Role	Is he a Volunteer	String	15			

Figure 2: Volunteer

< Data 3>						
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Id_Login	Login Id for Sign In	String	25	Not Null		Primary key
Password	Password for Sign In	String	25	Not Null		

Figure 3: Admin

< Data 4>						
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
PhoneString	Phone String of user	String	11	Not Null		Primary Key
BodyName	Name of dead body	String	15			
Date	Date of death	Date	10			
UserName	Who has booked the morgue	String	15			
Type	Details of dead	String	10			

Figure 4: Pending Morgue Request

< Data 5>						
Name	Dastarkhwan Donation					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use data when a donor wants to know the information about dasterkhwar/langer setup.					
Content description	Dastarkhwan data records.					

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Phone_Num	User's phone String	String	11	Not Null		Primary Key
D_Location	Name of Dastarkhwan	String	20	Not Null		
Pickup_Location	Location of user (long,lat)	Geopoint	20	Not Null		
Servings	String of servings for the people	String	4			
Type	Type of meal	String	20			
Username	User name	String	20			
Collected	Task completion status	boolean	5			
Approved	Pending request	boolean	5			

Figure 5:Dasterkhuwan donation

< Data 6 >						
Name	Pending Dastarkhwan Requests					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use data when donor wants to donate something or for admin activities.					
Content description	Donation detailed data records.					
Column Name	Description	Type	Length	Nullable	Default Value	Key Type
PhoneString	Phone String of user	String	11	Not Null		Primary Key
DastarkhwanName	Name	String	20			
Location	PickupLocation	Geopoint	20			
UserName	name of user	String	20			
Type	Type of donation	String	20			

Figure 6: Pending dasterkhuwan request

< Data 7 >						
Name	Ambulance					
Alias	Write other names used for the first entry.					
Where-used/how-used	Use data when user wants to book an ambulance or for admin privileges.					
Content description	Ambulance related data records.					
Column Name	Description	Type	Length	Nullable	Default Value	Key type
Id	Id of each Ambulance	String	4	Not Null	Auto Generate	Primary key
Phone_Nu m	Phone String of ambulance Driver	String	11			
Id_Login	Id for Login	String	25	Not Null		Primary key
Password	Password for Login	String	25	Not Null		
Status	Availability of ambulance	String	15			
Location	Location of Ambulance	Geopoint	50	Not Null		
Gender	Gender of driver	String	6			
Fare	Amount of booking an ambulance	String	4			

Figure 7: Ambulance

< Data 8>	
Name	Pending Volunteer Request
Alias	Write other names used for the first entry.
Where-used/how-used	Use data when donor wants to donate food.
Content description	Meal donation data records.

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
PhoneString	Phone String of user	String	4	Not Null		Primary Key
Location	Location of user	Geopoint	20			
UserName	Name of user	String	20			
Quantity	Serving String of person	String	4			
Type	Type of donation	String	10			

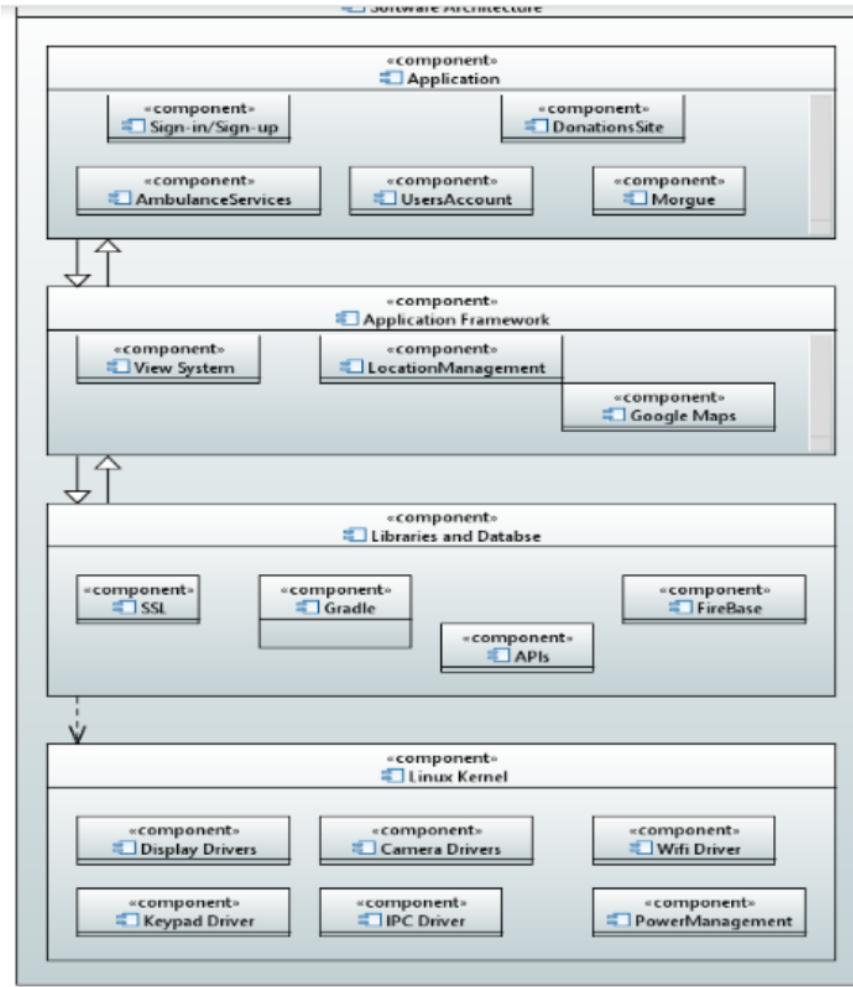
Figure 8: Pending volunteer request

< Data 9>						
Name	Blood Donation					
Alias	Write other names used for the first entry.					
Where-used/how-used	Used when donor wants to donate blood.					
Content description	Blood donation data records.					

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
Id	Each Blood transaction has an individual Id	String	4	Not Null	Auto Generate	Primary key
Donor_Id	Id of blood donor	String	4			Foreign Key
Volunteer_Id	Id of volunteer, who will collect the blood	String	4			Foreign Key
Blood_Group	Type of Blood	String	3			
No_Of_Bottle	String of bottles of blood required	String	3			

Figure 9:Blood Donation

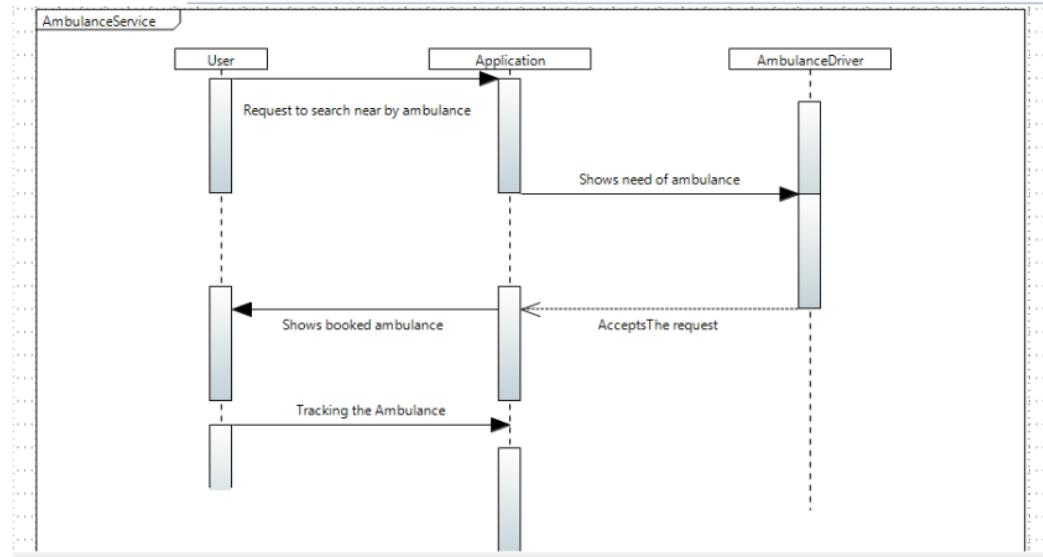
6 Application Design



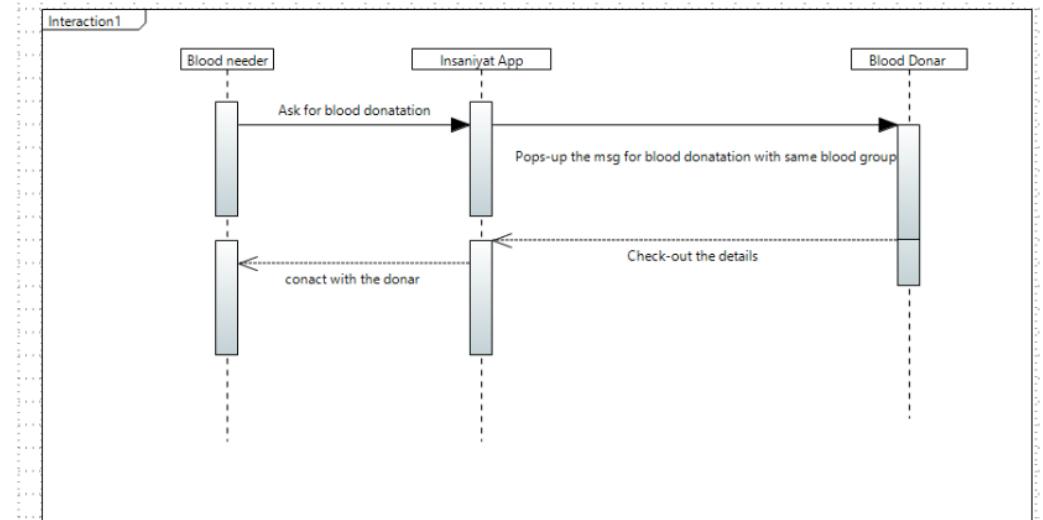
6.1 Sequence Diagram

Following are the sequence diagrams of all the features that we have implemented in our android application.

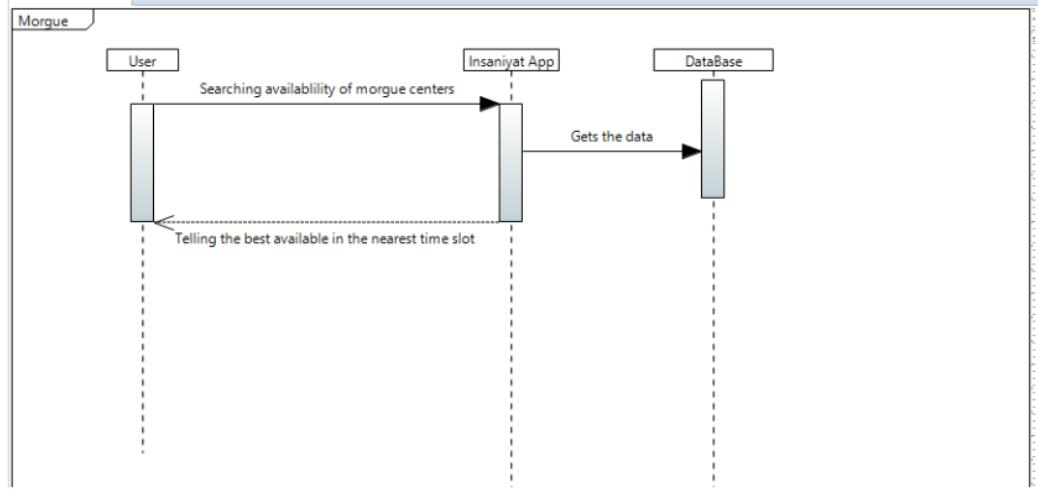
6.1.1 Ambulance Service



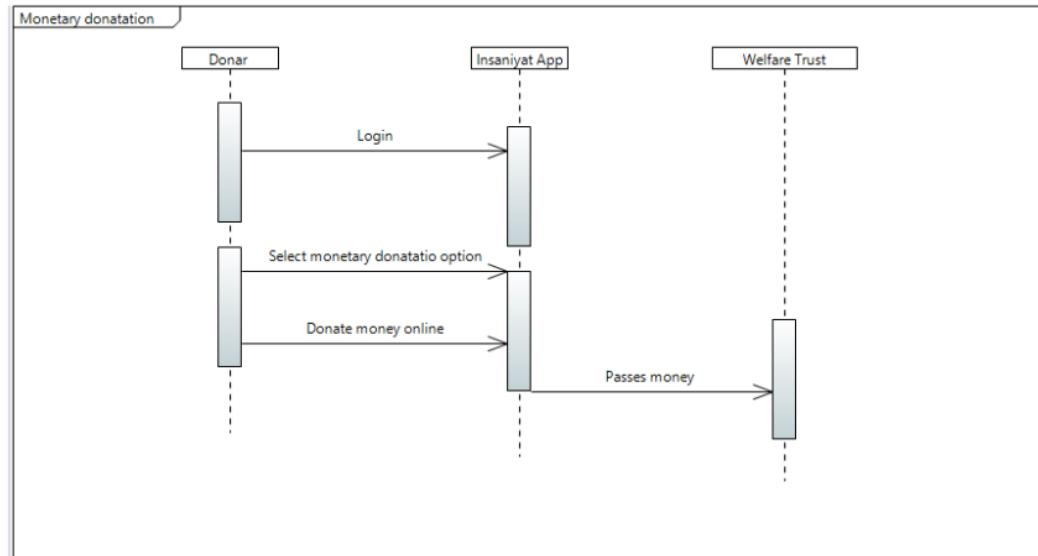
6.1.2 Blood Service



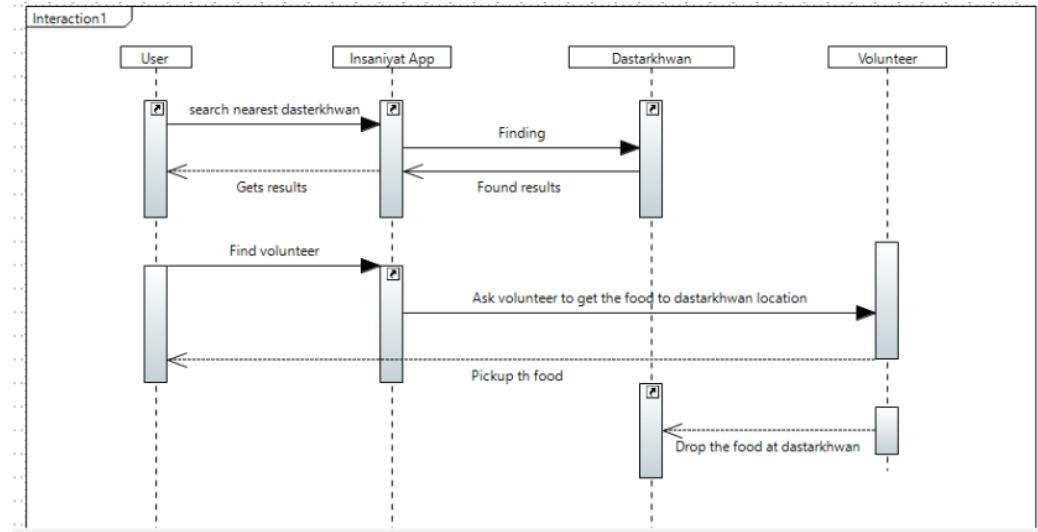
6.1.3 Morgue



6.1.4 Monetary Donation



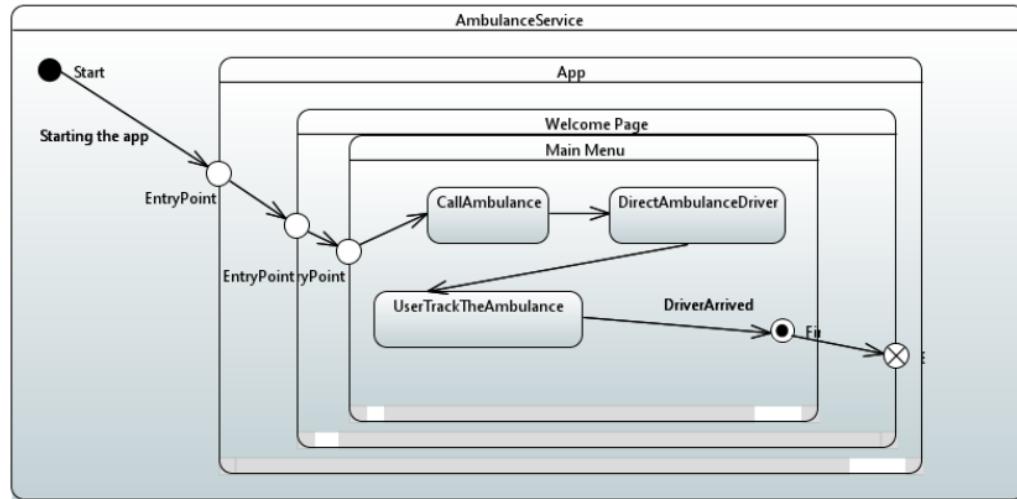
6.1.5 Meal Donation



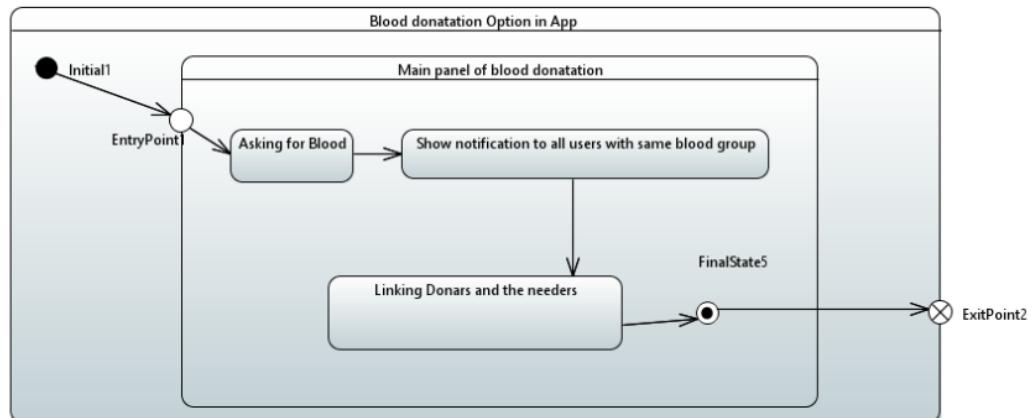
6.2 State Diagram

Here are state diagrams of all the features of our android application

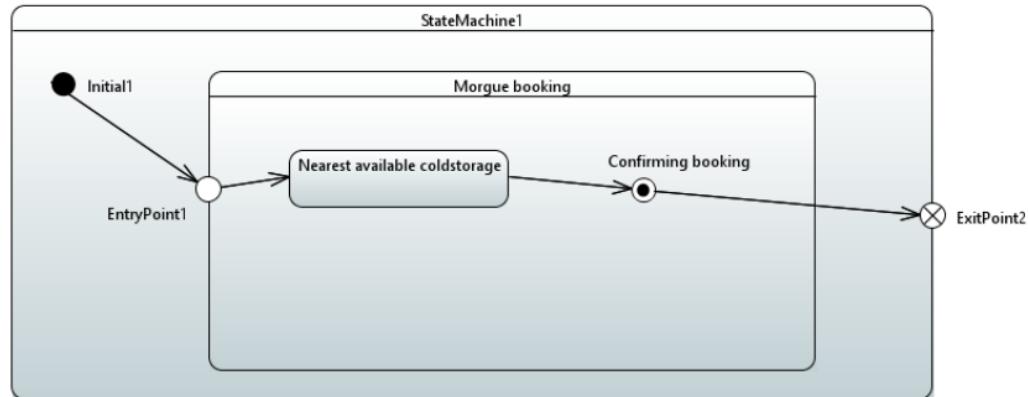
6.2.1 Ambulance Service



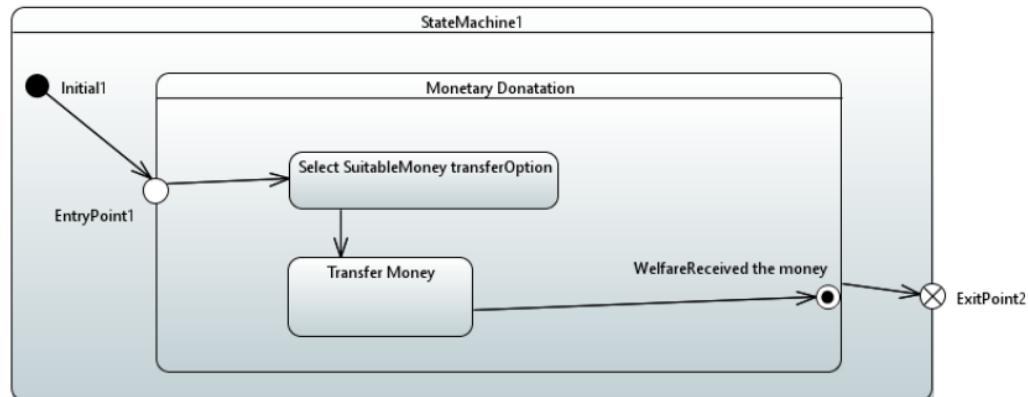
6.2.2 Blood Service



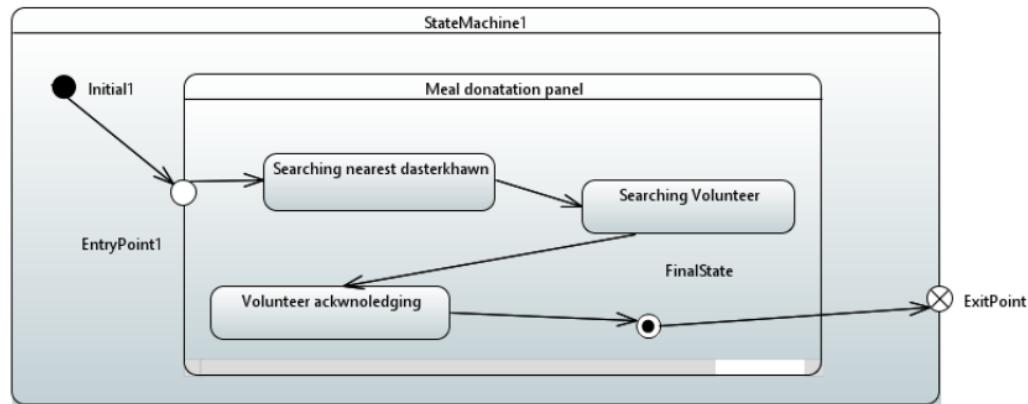
6.2.3 Morgue



6.2.4 Monetary Donation



6.2.5 Meal Donation



7 Implementation Details

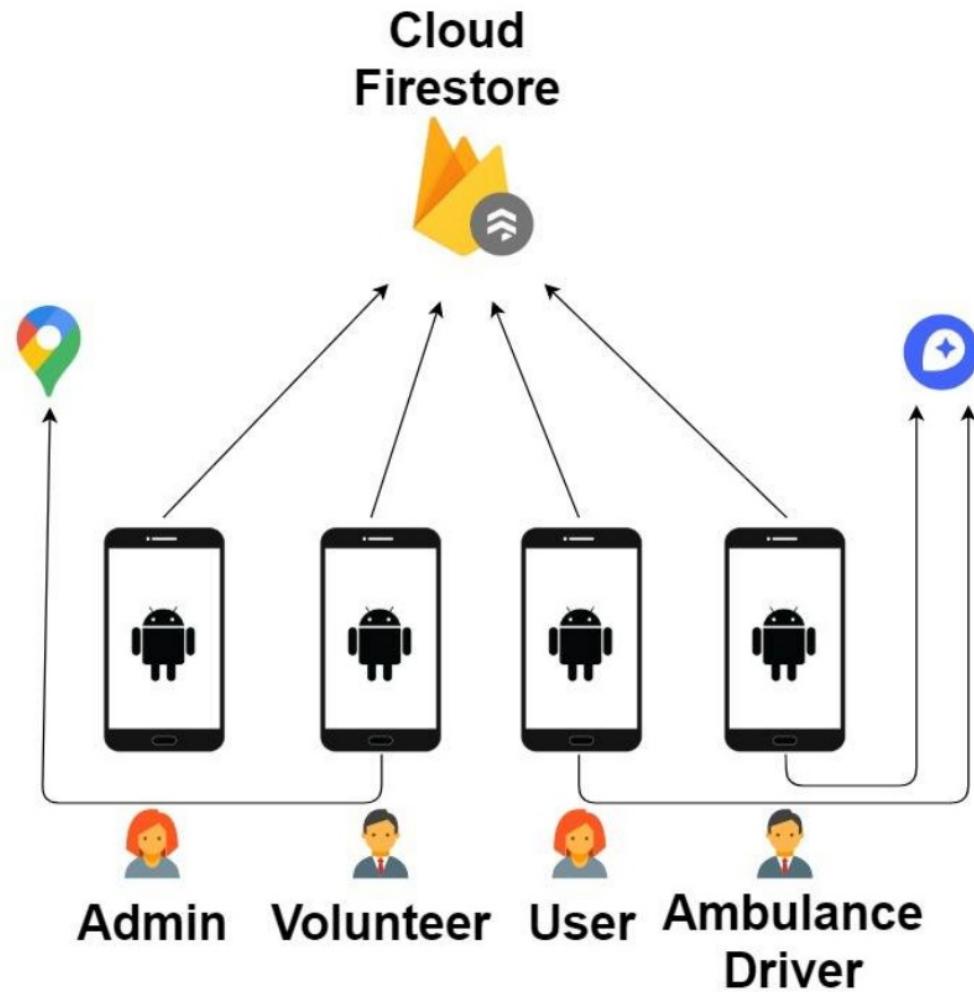
7.1 System Constraints

5

External environment may be caused by the stakeholders, business conditions, technical issues, academic requirements etc and may include the following:

- Software constraints
 - The efficiency of access decreases as the data set becomes more and more fragmented.
 - The API should be available/functioning at all times or else the most important functionality would be affected.
- Hardware constraints
 - Android phone with API level of minimum 15.
- Cultural constraints
 - The application would only support english language. Urdu language support is not available for this application.
- Environmental constraints
 - The environment should not be bright enough that the screen display is not clear.

7.2 System Level Architecture



7.3 Operating Environment

This application would require an android smartphone with stable Wi-Fi connection or cellular network(GPRS). The GPS should be working as well for services which require location of the user.

7.4 Assumption & Dependencies

- Assumptions
 - Assuming that the user can operate a smartphone.
 - Assuming that the user can understand english to use the application and operate.
 - The user is willing to trust the application with monetary donation.
- Dependencies :
 - The user of the application must have a stable network connection.
 - The user the application must have a working GPRS to use the services.
 - Google Maps API â Utilized as mapping and geocoding engine for the application.
 - Firebase - Non-relational cloud based database would handle the load giving availability at all times.

7.5 External Interface Requirements

7.6 Hardware Interfaces

Need separate android phones for all the four actors which are user, admin, ambulance driver and volunteer. The android phone specifications follow below:

- 2 GB RAM or above
- 50 MB internal storage or above
- GPS enabled
- He/she must have a valid internet connection

7.7 Software Interfaces

To track the location of the users of the mobile application, we need to ask the location coordinates from the user in order to have the precise accuracy. We will also store the encrypted form of passwords to ensure secure data storage.

7.8 Communications Interfaces Document Convention

- User Login for accessing the mobile application.
- Email forwarding or text message forwarding for confirmation of blood donation or any other kind of donation.
- Electronic forms are being used for doing paper work of booking an orphan.
- Critical data transfer is kept secured by encryption.

8 Screens

8.1 User

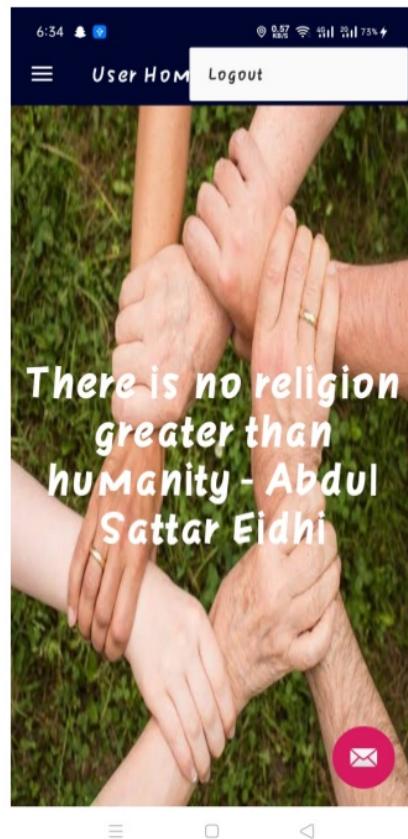


Figure 1: User Home Screen

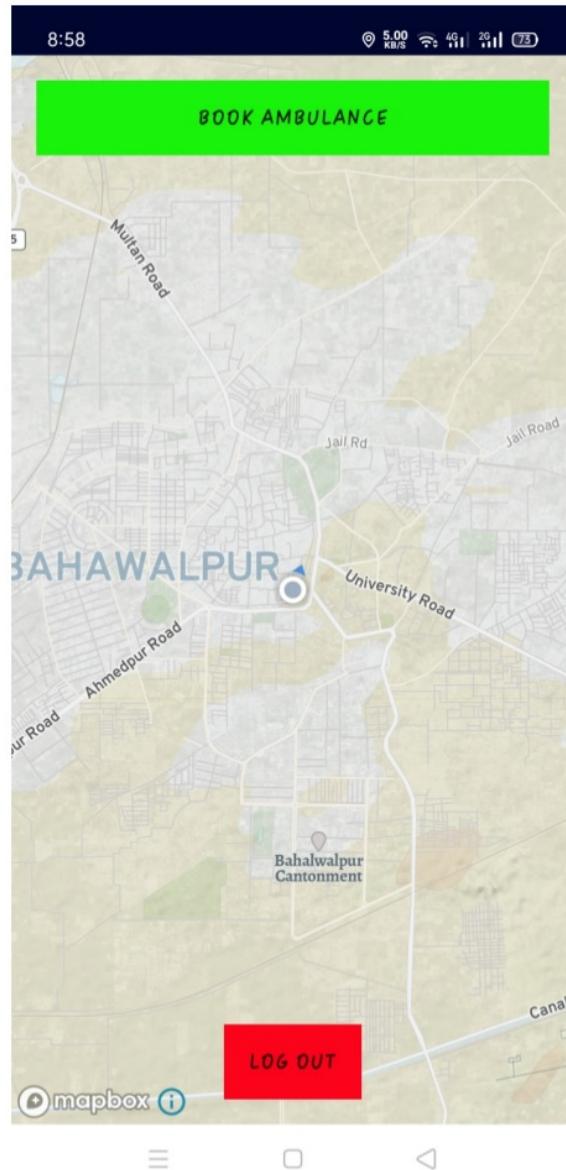


Figure 2: Book Ambulance



Figure 3: Meal Donation

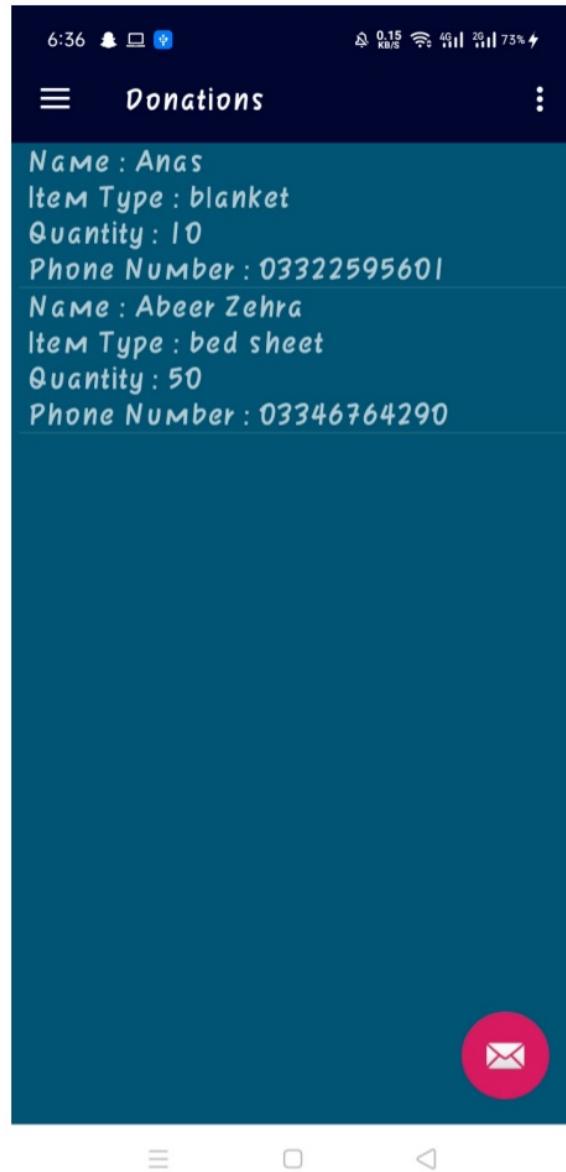


Figure 4: General Donation

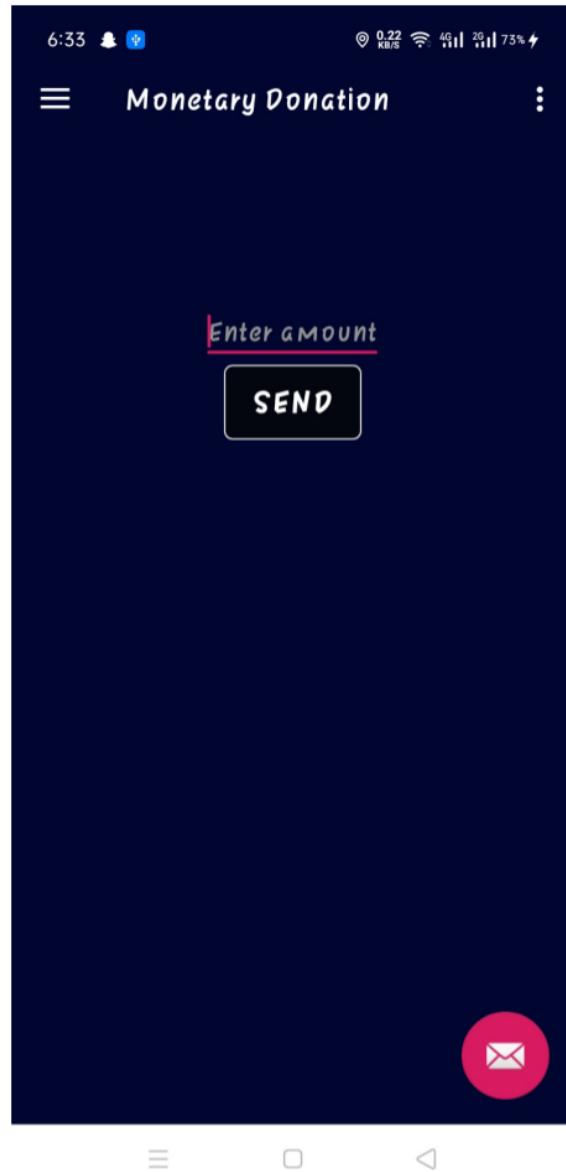


Figure 5: Monetary Donation

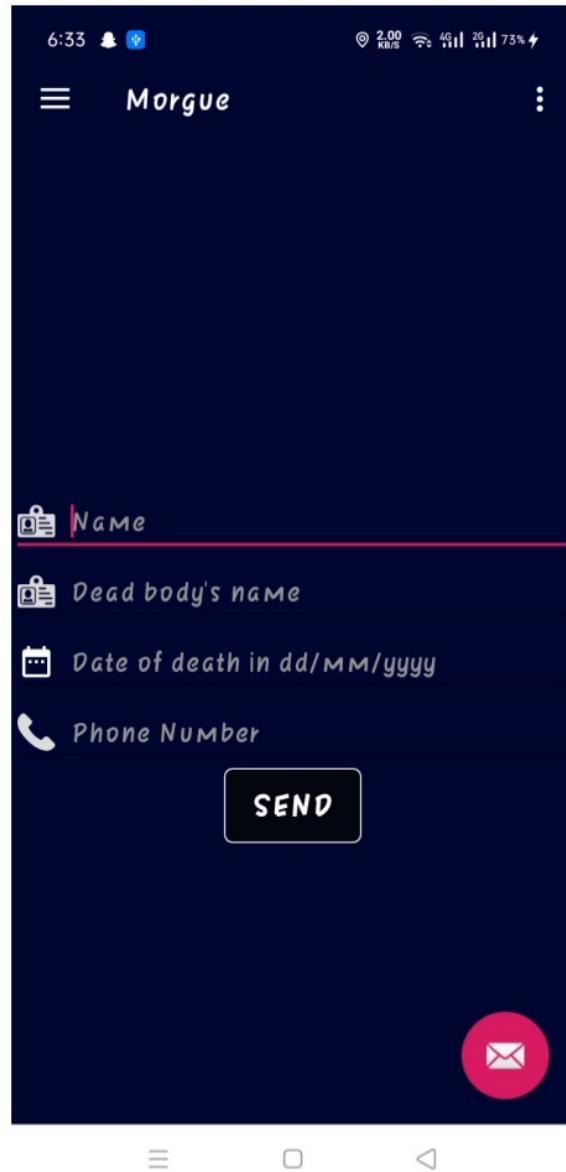


Figure 6: Morgue Request

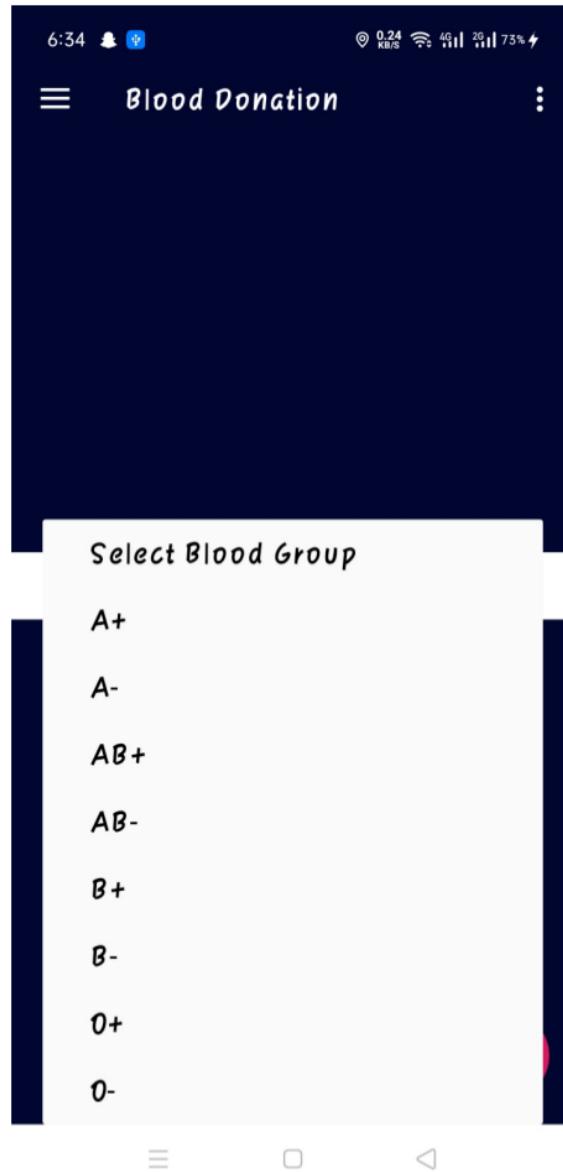


Figure 7: Blood Donation Request

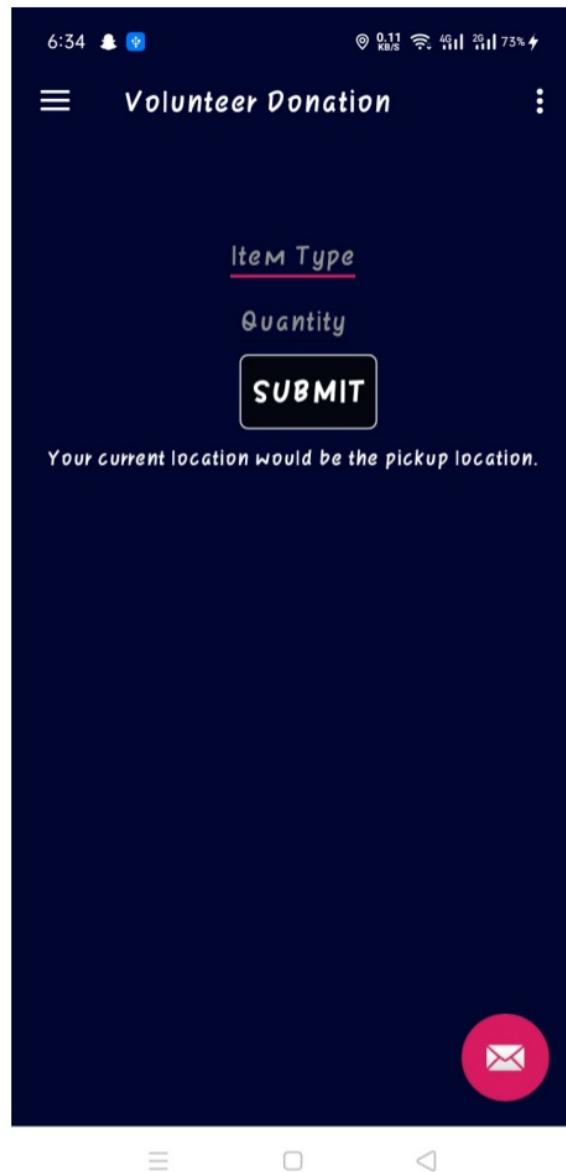


Figure 8: General Donation through volunteer

8.2 Volunteer



Figure 9: Volunteer

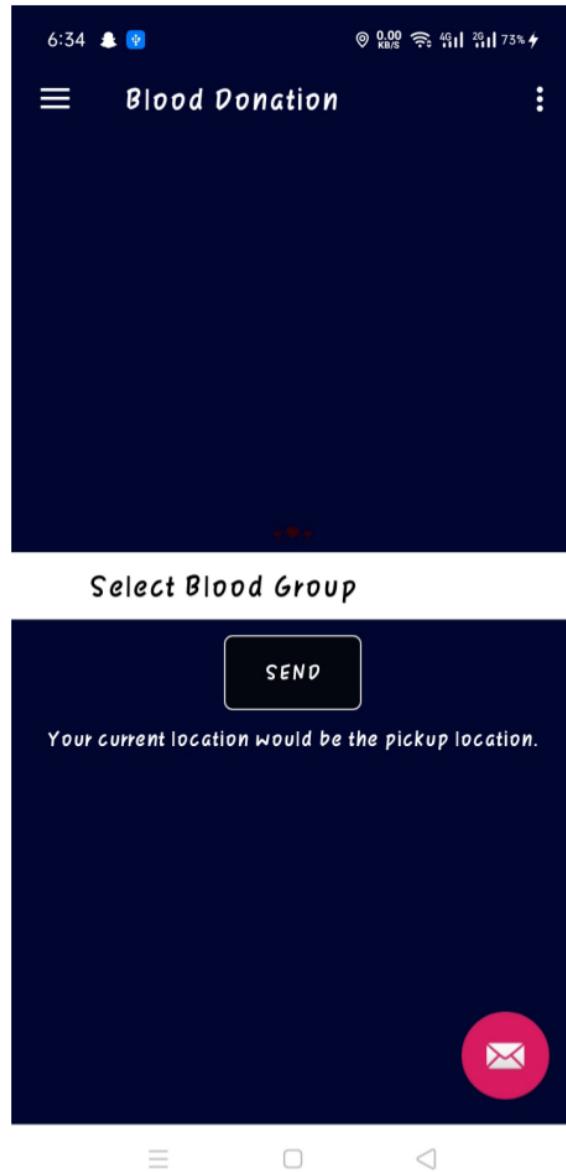


Figure 10: Blood Donation

8.3 Ambulance Driver

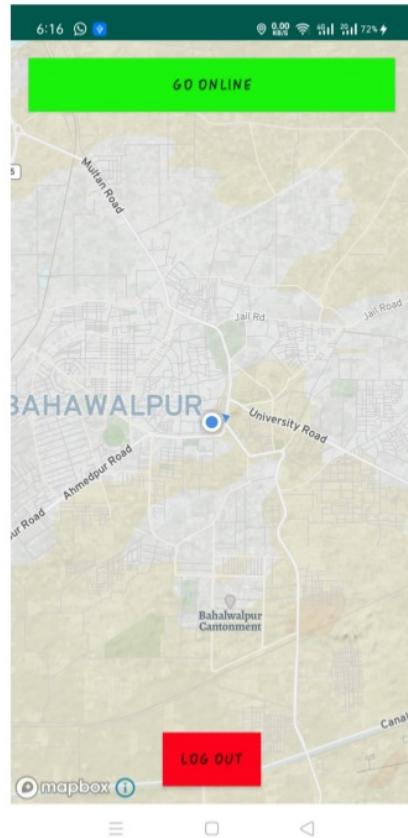


Figure 11: Ambulance Driver

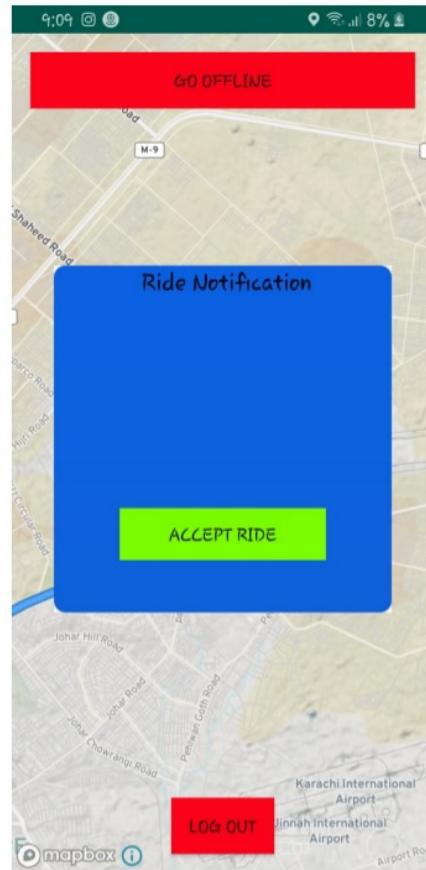


Figure 12: Ambulance Request

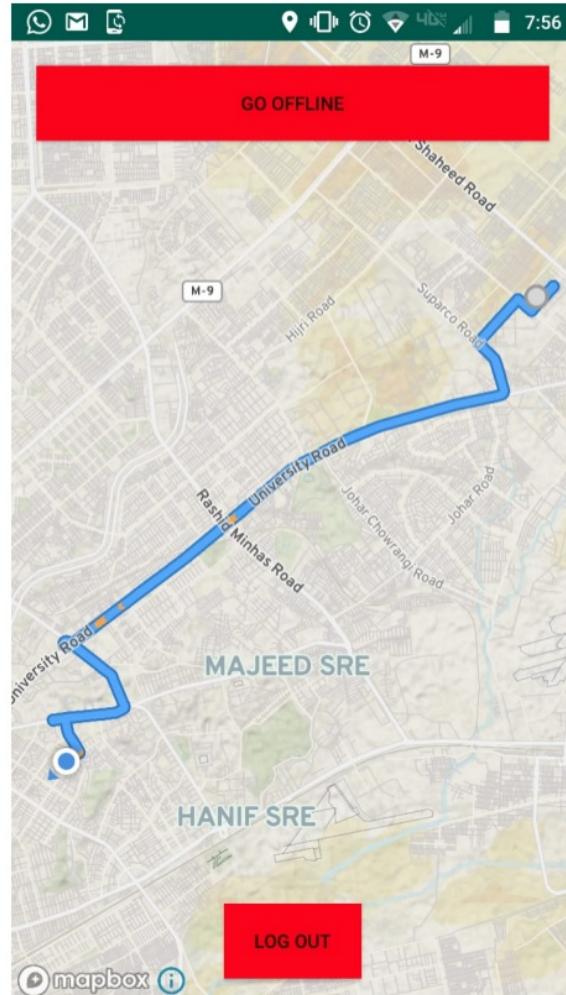


Figure 13: Tracking location

9 Test Plan

9.1 Introduction

9.1.1 Purpose of The Test Plan Document

Insaniyat- is an app that serve as a bridge between welfare organization and general public. Moreover, it also aims to minimize the wastage of food and provides the platform of volunteer where general public,welfare organization and volunteer can work together for the betterment of humanity.

9.1.2 Test Approach

There are two approaches for testing a software system i.e reactive and proactive. Reactive approach is used after designing and coding while proactive can be used before completion of coding of the system. So we are using reactive approach. When our application will be almost complete, we will start testing to identify bugs or issues in our application.

9.1.3 Test Pass/Fail Criteria

If 70 percent of users will be able to use our application smoothly, that means if they could understand application's color schemes and UI then our app will be considered as pass otherwise fail.

9.1.4 ENVIRONMENTAL NEEDS

- Users should have an android mobile phone with api level of atleast 6.
- Users needs to have a valid internet connection.
- GPRS should be in working condition to detect the location of user.

9.1.5 Validation Testing**8
ITEMS TO BE TESTED / NOT TESTED**

We will be testing all the main features of our app. This will not include signin/signup and it's verification feature.

Item to Test	Test Description
DifferentButtonSize	Buttons should be easier to click.
SplashActivity	Activities that remains for few milliseconds are splash activity. These should be understood by user to fulfil our purpose.
TextColor	Text colors used should be visible to every users as our application will be used by almost all age groups.
BackgroundImages	If background images are understood by users and whether they could associate it with their purpose of use.
Icons	We will see if users can identify the meaning of icons used in the application.
NavigationDrawer	We will see if user could direct himself towards the navigation drawer.

18

TEST CASES

Test Case:-Text Color, Icons
Description:- The user will be asked to install our application and check the visibility level of our application.

No.	Steps	Expected Results
1.	The user will be asked to install the application on his mobile.	He will signin/signup through the application
2.	He will be asked to browse through the application randomly.	He will walk through the application.

Test Case 1

Test Case:-Navigation Drawer, Background Activities, Splash Activity.
Description:- The user will be asked to browse through the application.

No.	Steps	Expected Results
1.	The user will be asked to use the application.	He will signin/signup through the application
2.	The user will be asked to navigate through the navigation drawer	User will click top left most icon.

Test Case 2

9.1.6 FUNCTIONAL TESTING**8
ITEMS TO BE TESTED / NOT TESTED**

We will be testing all the main features of our app. This will not include signin/signup and it's verification feature.

Item to Test	Test Description
Morgue	The user will be asked to request a place for morgue by entering the necessary details. This request will be send to admin. he also have to test the admin app at the same time.
GeneralDonation	The user will be asked to enter the item that he wants to donate and the quantity of it.
BloodDonation	The user will be asked to request the blood. This request will be send to the volunteer of matching blood group. If volunteer will accept, he will receive a conformation email.
MealDonation	The user will select the dasterkhuwan place. He will enter the quantity of food. If he sends this to volunteer then there will ba a notification in volunteer app and if he send to admin then notification in admin's app.
AmbulanceService	The user will be able to book nearest ambulance
MonetaryDonation	The user will asked to test this feature by entering the amount of money he wants to donate.

Test Case:- Ambulance service test
Description:- The user will be asked to test the ambulance service feature by using the application.

No.	Steps	Expected Results
1.	The user will be asked to book an ambulance service by click the button.	User will be able to send a request for an ambulance.
2.	Request will be send to the ambulance drivers.	Driver will receive the request.
3.	Driver will accept the request	User will be notified. And can see the drivers location on maps.

Test Case 3

Test Case:- General donation service test
Description:- The user will be asked to test the general donation feature by using the application.

No.	Steps	Expected Results
1.	The user will be asked to navigate through the navigation drawer to the general donation fragment.	He will be able to navigate.
2.	He will be asked to enter the amount .	He will enter the amount.
3.	He will then press the send button.	Verification message will be sent once the admin approves it.

Test Case 4

Test Case:- Meal donation test
Description:- The user will be asked to test meal donation feature by using the application.

No.	Steps	Expected Results
1.	We will ask user to open the app	The user will be able to signin/signup.
2.	He will be asked to open the meal donation module.	The user will be able to navigate through the drawer and select between donating to welfare or volunteer.
3.	Then he will be asked to select the place of dasterkhuwan if he is donating to the welfare and amount else just enter the item.	Amount and location entered will be recorded to collect the donation.
4.	Then he will finish donation by	Welfare/Volunteer will accept the request and confirmation message will be sent.

Test Case 5

Test Case:- Morgue service test
Description:- The user will be asked to test the morgue feature by using the application.

No.	Steps	Expected Results
1.	The User will be asked to select the morgue option from the navigation drawer.	The user will navigate to the morgue feature.
2.	He will be asked to enter all the details.	Details will be entered by the user.
3.	Admin will receive the request and upon situation accept/reject it.	User will receive message of acceptance/rejection.

Test Case 6

Test Case:- Blood donation feature test
Description:- The user will be asked to test the blood donation feature by using the application.

No.	Steps	Expected Results
1.	The user will be asked to test this feature by navigating through the blood donation.	He will be able to browse to the blood donation feature.
2.	He will be asked to select the blood group.	All the registered volunteer will be notified, and user will be notified from the one accepting the request.

Test Case 7

FYP Report

ORIGINALITY REPORT



PRIMARY SOURCES

- | | | |
|---|---|----|
| 1 | Submitted to University of Hertfordshire
Student Paper | 3% |
| 2 | Submitted to Informatics Education Limited
Student Paper | 2% |
| 3 | Submitted to Etisalat University College
Student Paper | 2% |
| 4 | Submitted to University of Surrey Roehampton
Student Paper | 1% |
| 5 | Submitted to Higher Education Commission
Pakistan
Student Paper | 1% |
| 6 | www.studymode.com
Internet Source | 1% |
| 7 | Submitted to The Hong Kong Polytechnic
University
Student Paper | 1% |
| 8 | Submitted to Trident University International
Student Paper | 1% |

9	Submitted to Hong Kong Baptist University Student Paper	1 %
10	Submitted to Swinburne University of Technology Student Paper	1 %
11	www.londonbrt.ca Internet Source	<1 %
12	Submitted to Icon College of Technology and Management Student Paper	<1 %
13	www.archive.org Internet Source	<1 %
14	Submitted to Royal Melbourne Institute of Technology Student Paper	<1 %
15	Submitted to Flinders University Student Paper	<1 %
16	elearning.nic.in Internet Source	<1 %
17	Submitted to Kennesaw State University Student Paper	<1 %
18	export.arxiv.org Internet Source	<1 %
19	ijmcs.dmcsp.pl	

Internet Source

<1 %

20

documents.mx

Internet Source

<1 %

21

Submitted to University of Nottingham

Student Paper

<1 %

22

Submitted to City University of Hong Kong

Student Paper

<1 %

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off

Source code

by Abeer Zehra

Submission date: 07-Jun-2020 02:13PM (UTC+0500)

Submission ID: 1339009740

File name: sourceCode.docx (48.59K)

Word count: 931

Character count: 8745

```
public
class
User
{
    public String email,password,name,phoneNumber,role,status;
    public Double latitude,longitude;
    User()
    {
    }
    User(String name, phone, email, password,Double latitude,Double longitude)
    {
        this.name=name;
        phoneNumber=phone;
        email = email;
        password=password;
        role="Ambulance Driver";
        latitude = latitude;
        longitude = longitude;
        status="Unvailable";
    }
    getEmail()
    {
        email;
    }
    getPassword()
    {
        password;
    }
    getName()
    {
        name;
    }
    getPhoneNumber()
```

```

        {
            phoneNumber;
        }
        getRole()
        {
            role;
        }
        Double getLatitude() {
            latitude;
        }
        Double getLongitude() {
            longitude;
        }
    }

Log.i("*****", "Location
changed"+loc.getLatitude()+"+"+loc.
getLongitude());



Toast(TheService.this,
"("++loc.getLatitude()+"+"+loc.getLongitude()+"", .show();

Log.d("Email: ",""+email);

DocumentReference Ref = mFireStore.collection("Ambulance
Drivers").document(""+email);

// update location

Ref.update("latitude",
""+loc.getLatitude(),"longitude",""+loc.getLongitude())

```

```
final User user = new User(naam,phone,email,pwd,latitude,longitutde);

db.collection("Ambulance Drivers").document(""+email).set(user);

emailId.setText("");

password.setText("");

if(location!=null)
{
    longitutde =
location.getLongitude();
    latitude = location.getLatitude();

Log.d("MyLatitude:::::>",""+latitude);

Log.d("MyLongitude:::::>",""+longitutde);

//Toast.makeText(SignUpActivity.this, "Show Location
"+latitude+" "+longitutde, Toast.LENGTH_SHORT).show();
}

final DocumentReference dref = db.collection("Ambulance
Drivers").document(""+email);

Toast.makeText(SignInActivity.this,"Incorrect Email or
Password, Please Login Again "/show());

//Show
details
of
customer

sms();
```

```
Display()
3
width =
dm.widthPixels;

height = dm.heightPixels;

getWindow().setLayout((int)(width
* .8), (int) (height * .4));

collRef = db.collection("AssignedRides");

collRef.whereEqualTo("AmbulanceDriverID",
"+FirebaseAuth.getInstance().getCurrentUser().getEmail()

number = document.getData().get("U_PhoneNumber").toString();
d_name = document.getData().get("D_Name").toString();
d_number = document.getData().get("D_PhoneNumber").toString();

smgr.sendTextMessage(number,null,"Your driver"+d_name+" is on his way.\nYou
can contact him on : "+d_number,null,null);

drefUser = db.collection("Ambulance
Drivers").document(""+mFirebaseUser.getEmail());

if
(_isChecked)
{

    //logic for signout
    indicator = false;
    //FINDLOC();
    Log.d("Before service starts:","1");
    //Start service for update in driver's
    location
        startService(new Intent(getApplicationContext(),
TheService.class));
    //Listening to any update in the Assigned
```

```
        Driver Collection
                attachListener();
        }

else
{
    // The toggle is disabled
    //logic for signout
    indicator = true;
    //Start of (End ride logic)
    if(obj != null ){
        //getting document id for assigned ride
        mFireStore.collection("AssignedRides")
            .whereEqualTo("AmbulanceDriverID",
              ""+mFirebaseUser.getEmail()

if (_task.isSuccessful())
{
    assignedDocID = _document.getId();
    //driverDocID =
    _document.getData().get("AmbulanceDriverID").toString();

    Log.d("Status","Fetched doc Id");

    Log.d("Status",""+an_document.getId() + " => " +
    an_document.getData());
}
//querry 2
```

```
mFireStore.collection("AssignedRides").document(assignedDocID)

Log.d("Status","an_Document deleted "+assignedDocID);

Log.d("Status","Document could not be deleted");

Log.d("Status","Error deleting document"+e);
}

});

//delete end
//querry 2 end

} else {
    Log.d("Status","Could not fetched doc Id");
}
}

});

//end getting
}

else{
Log.d("Ride status","No rides "+assignedDocID);
//querry3
// Set "Status" to "Unavailable" of ambulance driver
when goes offline
```

```
DocumentReference washingtonRef =
mFireStore.collection("Ambulance
Drivers")

    .document(mFirebaseUser.getEmail());
    washingtonRef

        .update("status", "Unavailable")
    {

        An_Log.d("Press Go
Offline:","Updated to
Unavailable");

    }

//end querry 3

restartActivity();
    detachListener();
    //End of (End ride logic)
    //end service
    Log.d("After service stops:","1");
    stopService(new
Intent(getApplicationContext(), TheService.class));
    }
}

});;

DocumentReference washingtonRef = mFireStore.collection("Ambulance
Drivers").document(mFirebaseUser.getEmail());
```

```
washingtonRef.update("status", "Available")

Log.d("Location : ","Lat: "+location.getLatitude());

washingtonR
ef

    .update("latitude",
    ""+location.getLatitude(),"longitude",""+location.getLongitude())

case
ADDED:
        obj=dc;
        //Pop up
notification for Ambulance Driver

openPopUpWindow();
                    //Storing useful
info for map navigation and notification pop-up
                    driverlat= new
Double( dc.getDocument().getData().get("D_Latitude").toString());
                    driverLong= new
Double( dc.getDocument().getData().get("D_Longitude").toString());
                    userLat = new
Double(dc.getDocument().getData().get("U_Latitude").toString());
                    userLong = new
Double(dc.getDocument().getData().get("U_Longitude").toString());
                    Log.d("Data
added:>", "New Driver: " + dc.getDocument().getData());
//Toast.makeText(LocationComponentActivity.this,
""+dc.getDocument().getData(), Toast.LENGTH_LONG).show();
```

```
//Start navigation for driver

StartNavigation();

case an_REMOVED:

restartActivity();
}

}

public void StartNavigation()
{
Point origin = Point.fromLngLat(driverLong,driverlat);
Point destination = Point.fromLngLat(userLong,userLat);
Log.d("Driver coord:",(""+origin));
Log.d("User coord:",(""+destination));
getRoute(origin, destination);
}

//////////////////driver ends here
```

```
BloodRequest bloodRequest = new
BloodRequest(name,bldGrp,phn,"blood",pickupLocation);

bloodRequestCRef.set(bloodRequest);

public class
DastarkhwanRequest
{
    String name,servings, phoneNumber,dastarkhwan,type;
    GeoPoint location;
    boolean approved;
    DastarkhwanRequest()
    {
    }
    DastarkhwanRequest( n, s,d, phn , t, boolean
b,GeoPoint g)
    {
        name = n;
        servings = s;
        phoneNumber = phn;
        location = g;
        dastarkhwan=d;
        approved = b;
        type = t;
    }
    getDastarkhwan()
    {
        dastarkhwan;
    }
    getType()
    {
        type;
    }
    setDastarkhwan(dastarkhwan) {
```

```
        dastarkhwan = dastarkhwan;
    }
    getName()
    {
        name;
    }
    getServings()
    {
        servings;
    }
    getPhoneNumber()
    {
        phoneNumber;
    }
    GeoPoint getLocation() {
        location;
    }
}

public class
DonationRequest
{
    String name,quantity, phoneNumber,type;
    GeoPoint location;
    boolean approved;
    DonationRequest()
    {
    }
    DonationRequest( n,  s,  phn , t,boolean b,GeoPoint g)
    {
        name = n;
        quantity = s;
        phoneNumber = phn;
        location = g;
```

```
        approved = b;
        type = t;
    }
    String getName()
    {
        name;
    }
    getQuantity()
    {
        quantity;
    }
    getPhoneNumber()
    {
        phoneNumber;
    }
    GeoPoint getLocation() {
        location;
    }
    getType() {
        type;
    }
}

1 mAppBarConfiguration = new AppBarConfiguration.Builder
(
    R.id.nav_home,
    R.id.nav_ambulance,R.id.nav_monetary_donation,R.id.nav_morgue,R.id.nav_welfare_meal_donation,R.id.nav_welfare_donation,
    R.id.nav_blood_donation,R.id.nav_volunteer_meal_donation,R.id.nav_volunteer_donation
) .setDrawerLayout(drawer)
```

```
public class
MonetaryDonation
{
    String name,phoneNumber,amount;
    GeoPoint location;
    MonetaryDonation( n, phn, amount,GeoPoint g)
    {
        name = n;
        phoneNumber = phn;
        amount = amount;
        location = g;
    }
    getName() {
        name;
    }
    getPhoneNumber() {
        phoneNumber;
    }
    getAmount() {
        amount;
    }
    GeoPoint getLocation() {
        location;
    }
}
```

```
DocumentReference pendingMorgueRequestCRef =
db.collection("PendingMorgueRequests").document("'" + phone);
```

```
//fetching
phone
numbers
and names
```

```
public String DriverPhoneNumber
UserPhoneNumber
UserName
DriverName;

if
(isChecked)
{
    // The toggle is enabled
    //logic for signout
    indicator = false;
    //uses gps
    FINDLOC();

mFireStore.collection("Ambulance Drivers")
.whereEqualTo("status", "Available")

if(diffBetweenTwoLoc <
minimumDist)
{
    minimumDist = diffBetweenTwoLoc;

    docID = document.getId();

    docLat = lat;

    docLong = lon;

    docName = Name;

    5
//Toast.makeText(MainActivity.this,(""+docID,
Toast.LENGTH_SHORT).show();
}
```

```
if(doc
ID != null)
{
    //get phoneNumbers of user and driver, and other logics
    runMultipleQuerries();
}

//newcod
e4

StartNavigation();
//newcode4 end

//for listener
attachListener();

mFireStore.collection("AssignedRides")
.whereEqualTo("AmbulanceDriverID", ""+docID)

mFireStore.collection("AssignedRides").document(assig
nedDocID) .delete()
onSuccess(Void
aVoid) {

    Log.d("Status","Document deleted
"+assignedDocID);

    Log.d("Status","DocumentSnapshot
successfully deleted!");

}
```

```
        restartActivity();

        detachListener();

    public void attachListener()
    {
        noteListener=
        mFireStore.collection("AssignedRides").whereEqualTo("AmbulanceDriverID",
        docID.toString())

        case REMOVED:
            restartActivity();
            Log.d("Data removed", "Driver
Removed is : " +
            dc.getDocument().getData());

            //ge
            t
            user
            phon
            e
            numb
            er
            and
            name

        mFireStore.collection("U
sers")
        .whereEqualTo("email",
        ""+mFirebaseUser.getEmail())
    }
}
```

```
mFireStore.collection("Ambulance Drivers")
    .whereEqualTo("email", ""+docID)

mFireStore.collection("AssignedRides")
    .add(Ride)

public class
PendingMealsReq
uest
{
    String
    name,quantity,
    phoneNumber,type;
    GeoPoint
    location;
    boolean
    approved;

PendingMealsReque
st()
{
}

PendingMealsReque
st( n, s,
phn,t,boolean
b,GeoPoint g)
{
    name = n;
    quantity =
s;
```

```
phoneNumber =  
phn;  
location = g;  
approved = b;  
type = "meal";  
}  
String  
getName()  
{  
name;  
}  
getQuantity()  
{  
quantity;  
}  
  
getPhoneNumber()  
{  
phoneNumber;  
}  
  
GeoPoint  
getLocation() {  
location;  
}  
getType() {  
type;  
}  
}
```

```
public
class
Admin
{
    String name,welfare,email,password;
    Admin()
    {
    }
    Admin(name, welfare, email, password)
    {
        name=name;
        welfare=welfare;
        email=email;
        password=password;
    }
    getName()
    {
        name;
    }
    getWelfare()
    {
        welfare;
    }
    getEmail()
    {
        email;
    }
    getPassword()
    {
        password;
    }
}
```

```
public class
BloodRequest
{
    String name,bloodGroup,phonenumber,type;
    GeoPoint pickupLocation;
    BloodRequest()
    {
    }
    BloodRequest(n, bloodgrp, phn, t, GeoPoint loc)
    {
        name = n;
        bloodGroup = bloodgrp;
        phonenumber = phn;
        pickupLocation = loc;
        type = t;
    }
    getType()
    {
        type;
    }
    getName()
    {
        name;
    }
    getBloodGroup()
    {
        bloodGroup;
    }
    getPhonenumber()
    {
        phonenumber;
    }
    GeoPoint getPickupLocation() {
        pickupLocation;
```

```
    }  
}
```

Source code

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to University of Northampton Student Paper	2%
2	Submitted to Nanyang Technological University Student Paper	1%
3	Submitted to Southampton Solent University Student Paper	1%
4	Submitted to London School of Commerce Student Paper	1%
5	Submitted to Nottingham Trent University Student Paper	1%

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off

Source code

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21
