CSE 3200: System Development Project

System Development Project Report

on

Software Application on Blood Donation System and Forecasting Blood Donor

Supervised by

Dr. M.M.A. Hashem
Professor
Department of Computer Science & Engineering
Khulna University of Engineering & Technology, Khulna.

18 June 2021

Signature of Supervisor

Team Members

Towsif Ahamed Labib
1707061

Md. Nazrul Islam
1707086

Department of Computer Science & Engineering
Khulna University of Engineering & Technology, Khulna.

Contents

1 Introduction

This is a documentation for an android application for a blood donation company where the authority of the organization can predict the possible number of blood donor who are willing to donate blood. Also, this app has added features for the normal users where they can able to sign up for blood donation and also, they can request blood in case of any emergency situation.

This application is separated into two sections. There will be a section with an option for the admin or staff of the blood donation organization. The other unit will operate for the general users of the application.

Thorough this application the members of the organization will able to predict the possible blood donor based on the previous records with organization's added feature set and using a decision tree-based model integrated into the application. This operation will be run on the cloud system and the show a prediction value to the application. The admins or staff will also be able to predict the stock on a monthly or weekly, or even daily basis. This has been the primary priority for their requirement. In other admin panel functionality, there are options. To use the feature the member of the organization, have to select a csv file containing the necessary feature set added by the organization to train the model. Then to do have an idea how much donor is going to expect for the blood donation they will give the registration records to forecast the possible number of the donors. This is important for a created for blood transfusion unit (BTU) to organize a successful event and also it decreases a lot of wastage of resources.

This application is relevant to the normal users. In this application the users can able to sign up to the volunteering work of blood donation where the application will keep track of the donor's location using OSM and their blood group and inform them about the emergency blood request via an interactive notification system which has been implemented to the application. The donor will also able to track when s/he have donated blood. If the donor is not applicable for the donation, then they will not be notified if the blood donation request. Also, there will be an option for a blood donation request where user can create an account and then able to request for blood if emergency case happen. The application is able to detect if any user tried to make a spamming by multiple requests.

In a nutshell, the application is created for blood transfusion unit (BTU) in mind. The feature has been added to the application according to their needs.

1.1 Purpose and Scope

The purpose of this very document is to give an overview of the application. In this document, the different aspects of the report have been written. The functionality and the structure of the application have been described in this document. The application is run on the android platform and a cloud-based system is created for doing heavy calculations. The scope of the application is

- A predictor model which runs on cloud for the organization member to forecast the possible number of the donors whom are expected to donate blood. This feature is only able to access by the organization members only.
- The application is also able to use by the users to sign up for voluntary blood donations and can keep track of his blood donation history.
- The application can also be used by the user to request for blood donation if any emergency case. To do so the user has to create an account.
- The application is implemented with OSM which is an open-source map service to fetch and show location information.
- This application is implemented with an interactive notification system for the donors where they will find out the emergency case that happens near his/her location if he can donate blood.
- If one user accepts any emergency blood request then the notification will be automatically removed from the other users.
- A spam detection system is been added to the application if any users want to spam the application by requesting blood again and again.

1.2 Product Overview

The created software interface is an application that is run on the android platform. The application is given the name “X”. This application will be used by the “Y” company. This application will be used by the employee and customer of the organization.

- The application needs minimum SDK of 28 to run.
- Most of the coding is done using JAVA
- The cloud system is implemented using python and DJANGO web framework
- A cloud-based system is built for the doing heavy calculations of forecasting the donor number for the organization members
- OSM (open sources-based map service) has been used to track of the location in the application
- An interactive notification system is been created for the donor to respond to the emergency blood request.
- For storing the user data real-time database Firebase has been used.
-

1.3 Structure of the Document

The documentation is written in the following sections.

- Project Management Plan
- Requirement Specifications
- Architecture of the Development
- Design of the Application
- Test Plan of the Application

The purpose of this very document is to give an overview of the application. In this document, the different aspects of the report have been written. The functionality and the structure of the application have been described in this document.

This document can be used for the blood donation organization to understand the functionality and the structure of the application. This document is not intended to be used for the user of the application. Here, in this case, the user is the donor, as this document contains a detailed description of the structure and functionality of the application from scratch.

2 Project Management Plan

2.1 Project Organization

Our respected supervisor Dr M.M.A. Hashem has the overall authority and supervised us for managing and executing this project according to this Project Plan and its Subsidiary Management Plans. The project team will consist of two personnel Towsif Ahamed Labib (1707061) and Md. Nazrul Islam (1707086). We both has divided the work in the coding part, quality control/assurance part, technical writing part, and testing part. All project and subsidiary management plans have been reviewed and approved by the supervisor.

2.2 Lifecycle Model Used

Basic stages of a software development lifecycle consist of planning and requirement analysis, designing project architecture, development and programming, testing and deployment. We have followed a water flow model. Waterfall model is the basic software development life cycle model. It is very simple but idealistic. Earlier this model was very popular but nowadays it is not used. But it is very important because all the other software development life cycle models are based on the classical waterfall model. Classical waterfall model divides the life cycle into a set of phases. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus, the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other.

The sequential phases in Waterfall model are

- Requirement Gathering and analysis
- System Design
- Implementation
- Integration and Testing
- Deployment of system
- Maintenance

2.3 Risk Analysis

As a two-member team and during the time of quarantine it's been a troublesome for us to maintain a sustainable system for accomplish the project. But to reduce the risk for the system we have gone with the version control system for our work flow. The design procedure from the scratch has been documented well if any kind of inconvenience occurs during our problem.

We have spilt the work-low. Mostly all the work has been done with collaborative way. But specially one of the members has been given task to oversight the whole coding portion and another member has been tasked to oversight the planning, documentation and testing phase for the application. Though both members mostly have done the work in a collaborative way but this split of responsibilities has accelerated the workflow and helped maintaining a discipline to the project.

2.4 Hardware and Software Resource Requirements

The application is built for the android platform and it has been created on the SDK 28 version.

2.4.1 Hardware Requirements:

- Must have a screen at least 2.5 inches in physical diagonal size. Have a physical diagonal screen size in the range of 2.5 to 8 inches.
- Are strongly recommended to provide users an affordance to change the display size.
- Have a power source that provides mobility, such as a battery.
- Must have touch screen input
- Must have network connectivity
- Must have a GPS sensor
- Have a system RAM of 4GB
- Have a minimum storage of 100 MB
- Must have a 64bit architecture processor

2.4.2 Software Requirements:

- The android device must run android version 9 or later
- Must give the permission to access the storage and location to run the application.

Deliverables and Schedule

2.5 Monitoring, Reporting, and Controlling Mechanisms

For the monitoring reporting purpose, a notion application has been used to have a record of task, bugs and actions that has been needed to perform for the application.

2.6 Professional Standards

Our job has been making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession.

- During the work all have acted consistently with the public interest.
- All have acted in a manner that is in the best interests of their client and employer consistent with the public interest.
- All have ensured that their products and related modifications meet the highest professional standards possible.
- All have maintained integrity and independence in their professional judgment.
- All have subscribed to and promote an ethical approach to the management of software development and maintenance.
- All have advanced the integrity and reputation of the profession consistent with the public interest.
- All of have been supportive of one another work.

2.7 Impact of the project on individuals and organizations

This application will be a help for the BTUs in general. The forecasting of possible donor from the registration data will surely help them to BTU to organize a blood donation event with ease and it will also help them to estimate the possible need to the resources to successfully complete an event. This is much helpful to the authority of BTU.

Also, the other added features for the general public to sign up for donating blood and request for the blood donation will have a good impact to the society. Also, the way donor can keep track of their regular donation and the interactive notification system for the donation will really encourage them to support the people near his/her area.

3 Requirement Specifications

3.1 Stakeholders for the system

3.1.1 Direct users

The direct user of the application is the BTU. The primary goal for the application was to make an application where the members or stuffs of the BTU can easily can able to forecast the possible blood donor before an event of blood organization based on the registration value and the previous records. And we have implemented a cloud-based system to the implement of the forecasting model. We have used a decision tree-based model to run through the application and predict the possible stock and arrange resources according to that.

3.1.2 Secondary users

The secondary users are the donors. They can able to observe the nearby emergency cases. They will get notified of emergency cases with a interactive notification system. This will help them to encourage them to donate blood. And also, they can able to maintain a record of regular blood donations. They will not be notified if s/he are not eligible to donate blood. As soon as the donor respond to the interactive notification s/he will be shown with details address and the notification from the other user will be removed.

3.1.3 Beneficiaries

The beneficiaries of the application are the in need of blood. They user of this group can able to request for the blood in case of emergency and the possibility of getting a help get increased. If any blood donor responds to their request, they can able to contact them.

3.2 Use case model

3.2.1 Use case: Forecasting of Future Blood Donors

Primary actor: Blood Transfusion Unit “X” organization authority.

Goal in context: To monitor the donor information and forecast the possible future stocks of the BTU.

Preconditions: System has been programmed for a password and for the forecasting necessary model has been implemented in the model.

Trigger: The authority decides to select a CSV file with donor information to forecast the possible future blood donor.

3.2.2 Use case: Respond to Interactive Emergency Donation Opportunities

Primary actor: Donor.

Goal in context: To set the system to monitor the donor's blood donation schedule and get notification for the emergency cases near his area and if possible, respond to the emergency cases. As soon as, the donor reacts to the notification he will be shown the details address of the place of action.

Preconditions: System has been programmed for a password and to recognize each donor.

Trigger: The donor logs into his own account to perform his choice as a donor.

3.2.3 Use case: Request for Emergency Blood and See Donor Update

Primary actor: In Need.

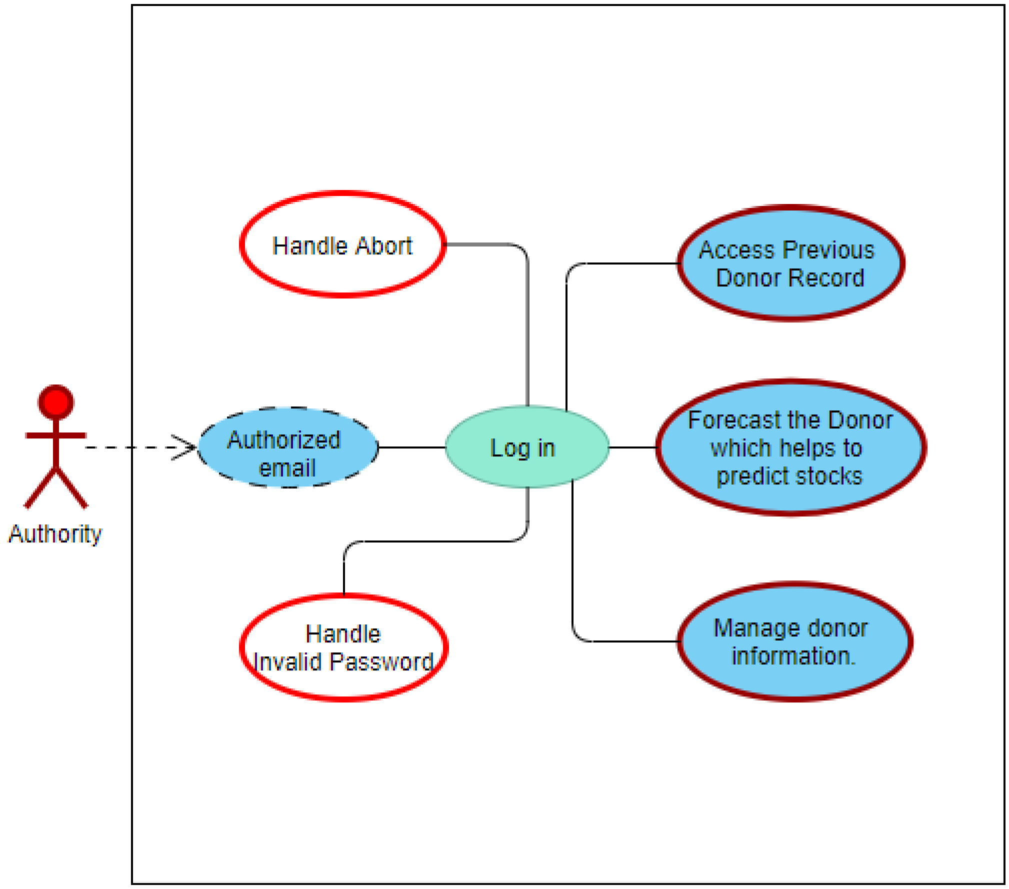
Goal in context: To set the system to monitor the nearby donor's location and ask for the emergency cases near his area.

Preconditions: System has been programmed for a password and to recognize if one needs blood for validity purpose.

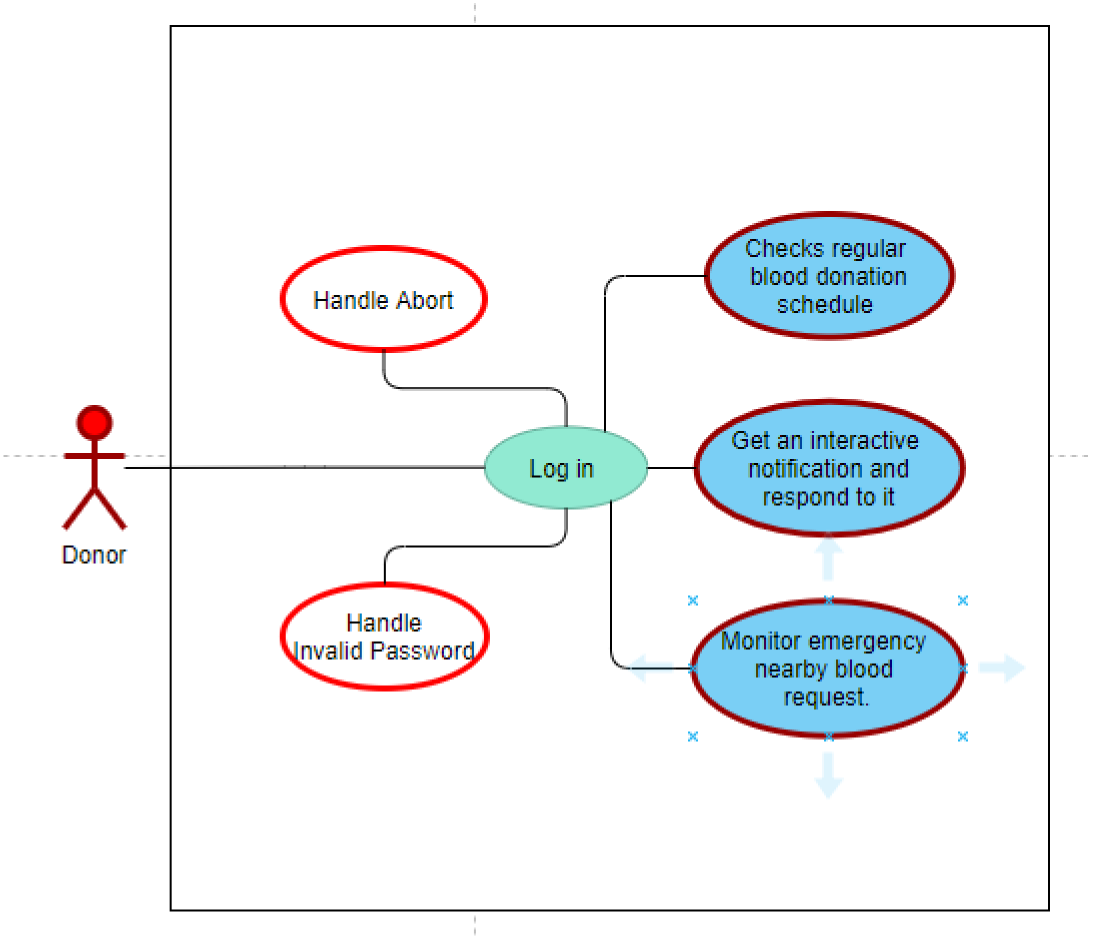
Trigger: The donor logs into his own account to perform his choice as a donor.

3.3 Graphic use case model

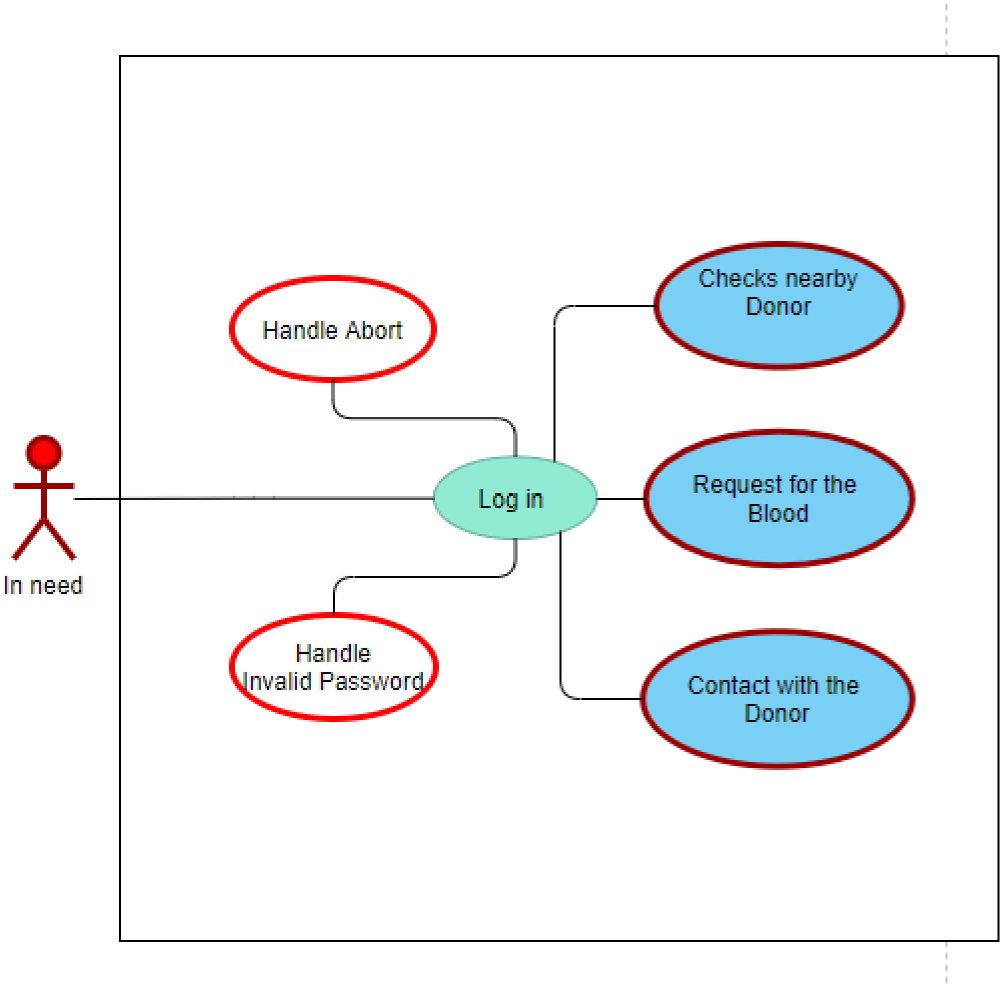
3.3.1 Actor: Organization member or Authority


Handle Abort
Access Previous Donor Record
Authority
Authorized email
Log in
Forecast the Donor which helps to predict stocks
Handle Invalid Password
Manage donor information.

3.3.2 Actor: Donor


Handle Abort
Checks regular blood donation schedule
Donor
Log in
Get an interactive notification and respond to it
Handle Invalid Password
Monitor emergency nearby blood request.

3.3.3 Actor: In Need



3.4 Textual Description for each use case:

3.4.1 Actor: Organization member or Authority

Scenario:

- Donor: observes control panel
- Donor: enters password
- Donor: observe the maps and the notification for the donation of blood
- Donor: monitor the blood donation schedule.

Exceptions:

- Control panel is not ready: the donor checks if the necessary permission for the app is provided to run the application and has a stable internet connection.
- Password is incorrect: donor has to re enter correct password.
- Maps are not displayed if there is no location stored in the device memory buffer.

Priority: Essential, must be implemented
When available: First increment
Frequency of use: Many times, per day
Channel to actor: Via control panel interface
Channels to secondary actors:
Authority: associate members from the group
In Need: in need of blood for emergency situations.

Open issues:

- Should the control panel display additional text messages?
- Is there a way to enter as in need from the control panel?

3.4.2 Actor: Donor

Scenario:

- Donor: observes control panel
- Donor: enters password
- Donor: observe the maps and the notification for the donation of blood
- Donor: monitor the blood donation schedule.
- Donor: respond to the interactive notification.

Exceptions:

- Control panel is not ready: the person in need checks if the necessary permission for the app is provided to run the application and has a stable internet connection.
- Password is incorrect: person in need has to reenter the correct password.
- Maps are not displayed if there is no location stored in the device memory buffer.

Priority: Essential, must be implemented
When available: First increment
Frequency of use: Many times, per day
Channel to actor: Via control panel interface
Channels to secondary actors:
Authority: associate members from the group
Donor: donate blood.

Open issues:

- What happen if donor’s current location is different from the included location?
- Should the control panel display additional text messages?

3.4.3 Actor: In Need

Scenario:

- Requester: observes control panel
- Requester: enters password

- Requester: observe the maps and the see possible blood donor
- Requester: contact the blood donor

Exceptions:

- Control panel is not ready: the person in need checks if the necessary permission for the app is provided to run the application and has a stable internet connection.
- Password is incorrect: person in need has to reenter the correct password.
- Maps are not displayed if there is no location stored in the device memory buffer.

Priority: Essential, must be implemented
When available: First increment
Frequency of use: Many times, per day
Channel to actor: Via control panel interface
Channels to secondary actors:
Authority: associate members from the group
Donor: donate blood.

Open issues:

- Should the control panel display additional text messages?

3.5 Non-functional requirements

- The donor data has been kept confidential and no one except the authority are allowed to use the data.
- It is better to turn on the location and if the if the value in the buffer for the GPS is null the app tends to crush. As In this application the open-source map OSM is used which does not have a good integration with the android. Thus, it is necessary to turn on the location and open google maps before opening the application. The problem is caused as OSM has not still updated their support for the SDK 29.

4 Architecture

4.1 Architectural style

A data-centered architecture is followed to build the android application. A data-centered architecture has two distinct components: a central data structure or data store (central repository) and a collection of client software. The datastore (for example, a database or a file) represents the current state of the data and the client software performs several operations like add, delete, update, etc., on the data stored in the data store. In some cases, the data store allows the client software to access the data independent of any changes or the actions of other client software.

In case this application a firebase database system has been used and the client software here in this case is the android application. But for the forecasting the future donor number the used train and test set of data has to be stored in the local device.

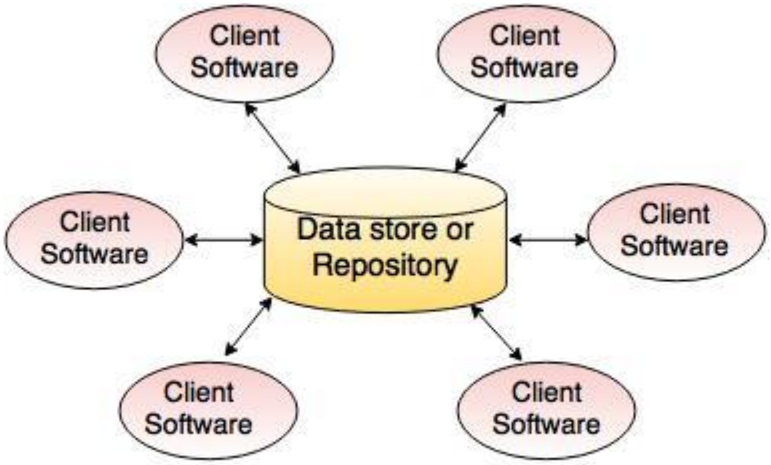


Fig.- Data centered architecture

4.2 Architectural pattern

Client-server pattern has been used in the buildup of the application. This pattern consists of two parties; a server and multiple clients. The server component will provide services to multiple client components. Clients request services from the server and the server provides relevant services to those clients. Furthermore, the server continues to listen to client requests.

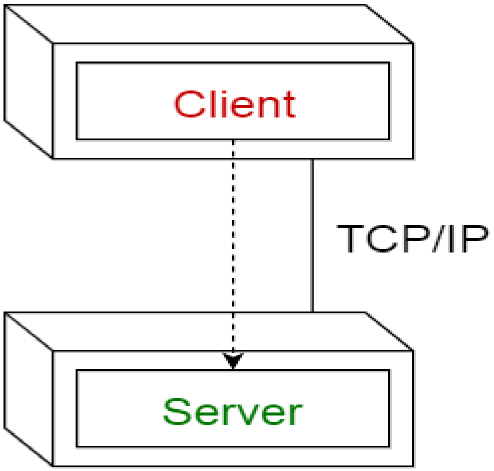


Fig: - Client-Server Architectural Pattern

4.3 Technology, software, and hardware used

To build the application android platform has been used. The application is built on the android SDK 28. The reason behind choosing SDK 28 has been as there was no support of OpenStreetMap (OSM) for the later version of the SDK when the development had been started.

The application is mostly built on using java. For the cloud integration python Django framework has been used. For forecasting donor application. the decision tree-based model has been introduced for fitting the authority data perfectly.

4.3.1 Forecasting the Possible Donor from the Registration to Blood donation event

The forecasting is achieved by a thorough examination of donor-based blood collection data extracted from the database of the OneBlood organization in the year 2020. OneBlood is a blood donation center that serves Tampa, South, Southeast, and Central Florida.

OneBlood employs its self-made propensity model to generate probability scores that indicate whether or not a donor will donate.

In the dataset the donors are represented by UserIDs in the dataset, and they have probability scores assigned to them. A single user can have multiple rows, just as a single donor can make multiple donations. Other features derived from probability scores are present, and the target feature has a binary value based on whether or not the donor donated. It has a value of 1 if the donor donated and 0 if they did not. We gathered the data over the course of a year. Blood donation is scarce, as only less than 5 percent of the donors donated blood.

With the dataset of OneBlood a regression analysis has been implemented using decision tree-based model has been used to forecast the possible donor presence before an eve

4.3.2 Use of OpenStreetMap (OSM)

This application uses a map to show nearby incident. It also responsible for finding the donors' location information. As google map service is not a free service and the data is handled by the Google thus, we have used OpenStreetMap (OSM), an open-source map service.

OpenStreetMap is a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license.

The OpenStreetMap License allows free (or almost free) access to our map images and all of our underlying map data. The project aims to promote new and interesting uses of this data.

OpenStreetMap is more than just a map of the world, place search and navigation. It gives free and unlimited access to the entire dataset, with a complete history of changes, without limiting who and how can use it. OpenStreetMap is a perfect foundation for emerging innovative and creative solutions and targeted products.

The data from OSM can be used in various ways including production of paper maps and electronic maps (similar to Google Maps, for example), geocoding of address and place names, and route planning. Prominent users include Facebook, Wikimedia Maps, Apple, Microsoft, Amazon Logistics, Uber, Craigslist, Snapchat, OSM And, Maps.me, Geocaching, MapQuest Open, JMP statistical software, and Foursquare. Many users of GPS devices use OSM data to replace the built-in map data on their devices. OpenStreetMap data has been favorably compared with proprietary data sources, although as of 2009 data quality varied across the world.

4.3.2.1 Why not choosing the Google Maps?

Customization choices are limited. Despite the fact that the new custom styles are still in beta, the Google Maps API presently only offers a few choices for customizing the look of your integrated maps.

Not an open-source API. We may speculate if this is because Google needs no assistance from the developer community or because the company is retaining the authority to deploy some features without external approval, but that isn't the case.

Price fluctuations that are unpredictable. The 2018 Google Maps API pricing increase has caused a stir among long-time users. An outstanding service like this should be monetized and profitable, but client trust is also a valuable commodity.

4.3.2.2 Why choosing the OSM:

The OpenStreetMap API is free. No added expenditures for your business here.

An open-source map. A large number of contributors passionate about mapping ensures steady growth of the database.

Customization. Full history of changes of a particular place is offered for free.

4.3.3 Implementation of an Interactive Notification Systems

In this application an interactive notification system has been introduced to send the emergency news to the application user so that they can respond to notification.

Interactive Notifications are push notifications featuring supplementary, supplementary action buttons that allow users to engage with notifications outside of the app. Instead of needing to launch the app, app users may reply to notifications directly from the notification. They frequently include buttons that allow users to take instant action, such as purchasing tickets, using a discount coupon, or checking in for a flight. Interactive notifications, unlike normal push notifications, allow for two-way communication. They can drive the user to:

- Take immediate, specific action
- Make decisions or choices

- Express preferences

The fact that push notifications communicate at the client rather than with them has always been a limiting element. Users' reaction rates have decreased as the number of push alerts has increased. Users are just getting immune to push alerts, and many of them are turned off to begin with.

One-way push notifications, which basically served as billboard ads from within apps, have been replaced with interactive notifications. Their ability to engage the user immediately is a victory for the customer experience. However, the consumer retains the ability to ignore or disable them.

In-app messaging is a more effective technique to increase client engagement. In-app messaging is a type of native mobile texting that takes place within the app itself. It's a really conversational and engaging way to communicate directly with consumers.

Push notifications and interactive notifications both have a place in the world. Within a customer experience organization, in-app communications are also quite useful. If the user is still interested in using the app, interactive notifications might be a wonderful way to capture their attention if they haven't opened it in a while. However, once clients are within an app, in-app messaging is usually a more productive and engaging manner of connecting with them.

When any one request for the blood a notification will be send to the all the nearby donor. If someone responds to some notification then the notification will be get deleted from the other application user. Thus, the notification of the application will not create any bloat.

4.3.4 Use of Java:

For the developing the android application a java language has been used.

4.3.5 Use of Python Django Framework:

For implementing the cloud integration for the application python Django framework has been used.

4.4 Rationale for your architectural style and model

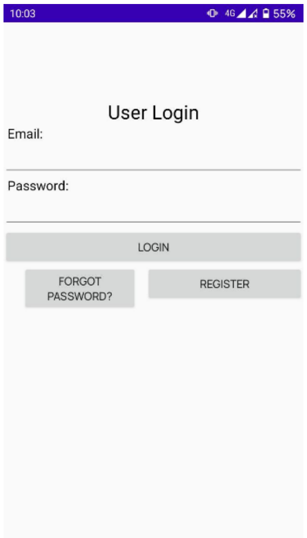
The data-centered architecture style and cloud-server architectural pattern have been used in our application.

In the application the data for the user is directly send to the firebase real-time database. The user can update the user data on the application which directly changed in the real-time database. The blood request data is also sent to the firebase real-time database. This data is then utilized to implement the interactive notification. The interactive notification works based on the fetching the data from the firebase real-time database. All of these data have been stored in a firebase real-time database which make the implementation easier. Thus, the data-centered architecture style has been used.

In our application the client-server-based pattern is used. The user of the application is client here in case. The user information and the information which is shown in the application is directly from the firebase real-time database. Thus, choosing client-server-based architectural pattern makes it simpler to implement the model.

5 GUI (Graphical User Interface) design

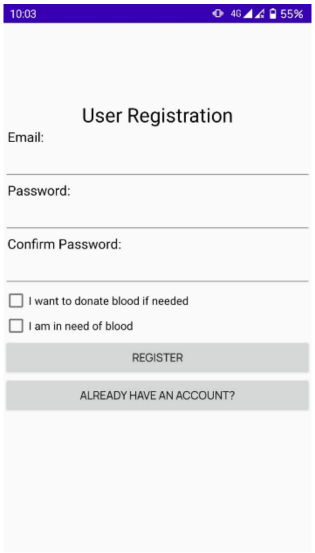
The following picture contain the user login, user registration, homepage, user Profile, company page and notification system respectively.


10:03 55%

User Login

Email:

Password:


10:03 55%

User Registration

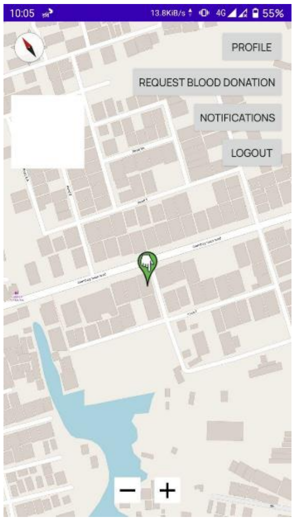
Email:

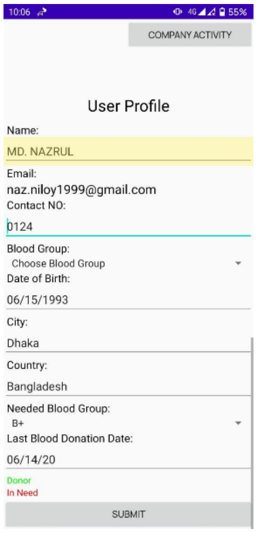
Password:

Confirm Password:

□ I want to donate blood if needed

□ I am in need of blood




10:06 55%

User Profile

Name:
MD. NAZRUL

Email:
naz.niloy1999@gmail.com

Contact NO:
0124

Blood Group:
Choose Blood Group

Date of Birth:
06/15/1993

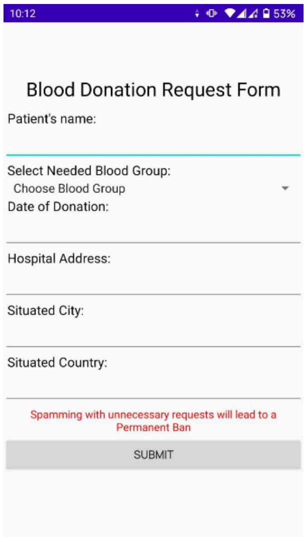
City:
Dhaka

Country:
Bangladesh

Needed Blood Group:
B+

Last Blood Donation Date:
06/14/20

Donor
In Need


10:12 53%

Blood Donation Request Form

Patient's name:

Select Needed Blood Group:
Choose Blood Group

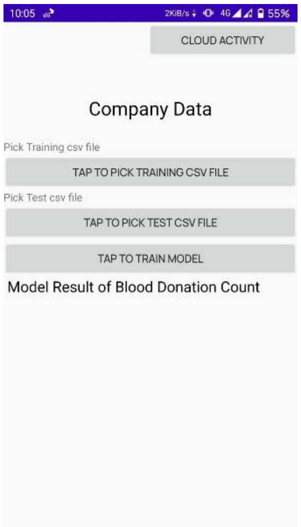
Date of Donation:

Hospital Address:

Situated City:

Situated Country:

Spamming with unnecessary requests will lead to a Permanent Ban

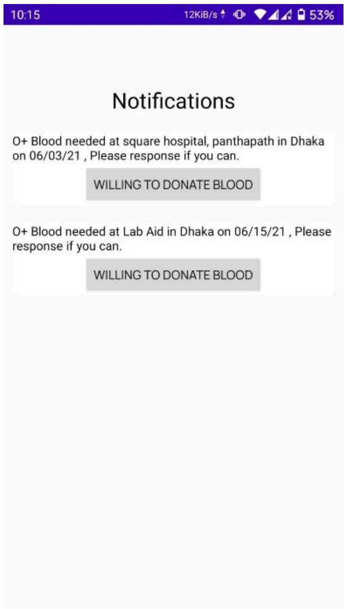

10:05 2KiB/s 55%

Company Data

Pick Training csv file

Pick Test csv file

Model Result of Blood Donation Count



10:15 12KiB/s 53%

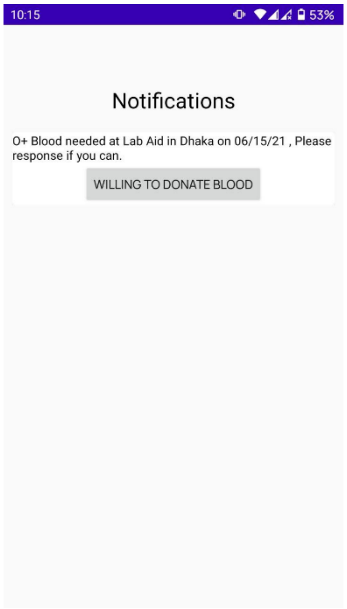
Notifications

O+ Blood needed at square hospital, panthapath in Dhaka on 06/03/21 , Please response if you can.

WILLING TO DONATE BLOOD

O+ Blood needed at Lab Aid in Dhaka on 06/15/21 , Please response if you can.

WILLING TO DONATE BLOOD

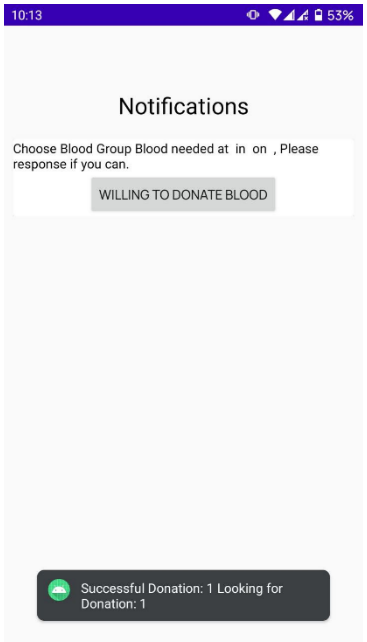


10:15 53%

Notifications

O+ Blood needed at Lab Aid in Dhaka on 06/15/21 , Please response if you can.

WILLING TO DONATE BLOOD



10:13 53%

Notifications

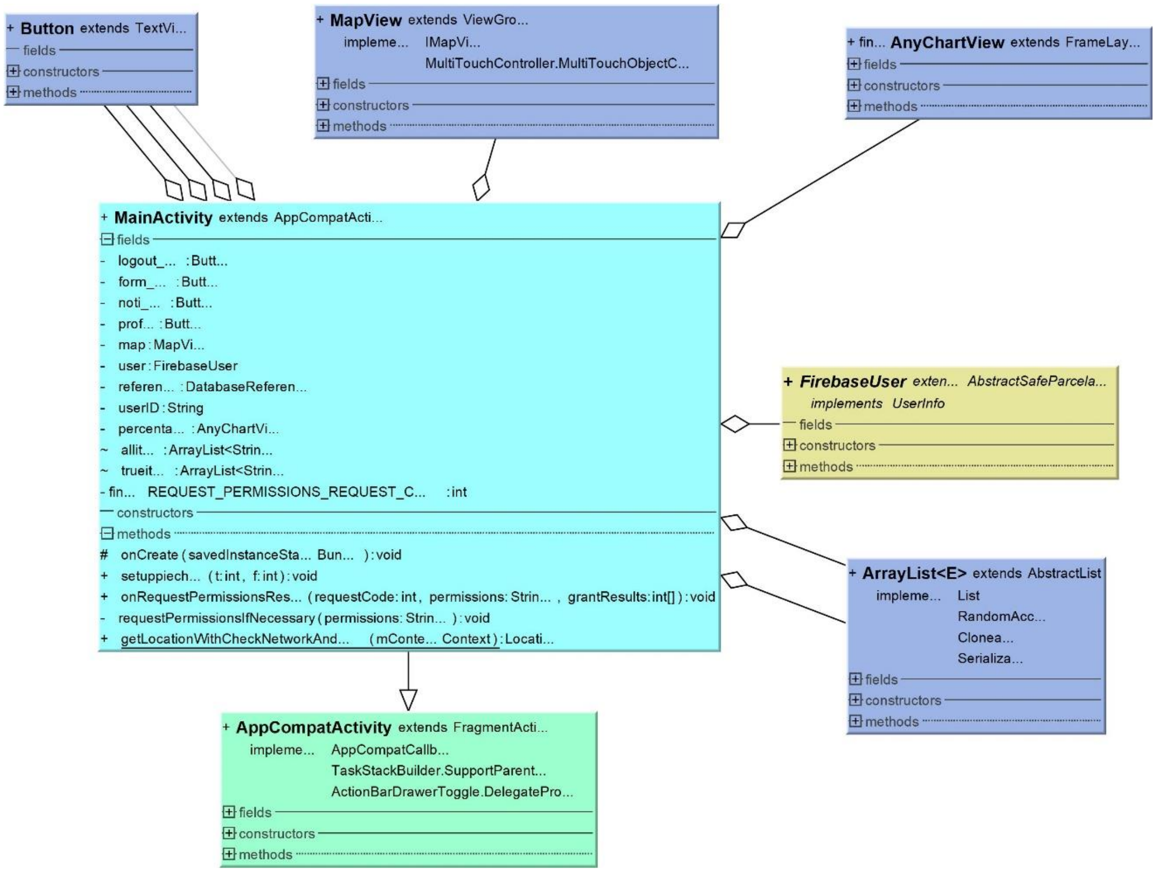
Choose Blood Group Blood needed at in on , Please response if you can.

WILLING TO DONATE BLOOD

Successful Donation: 1 Looking for Donation: 1

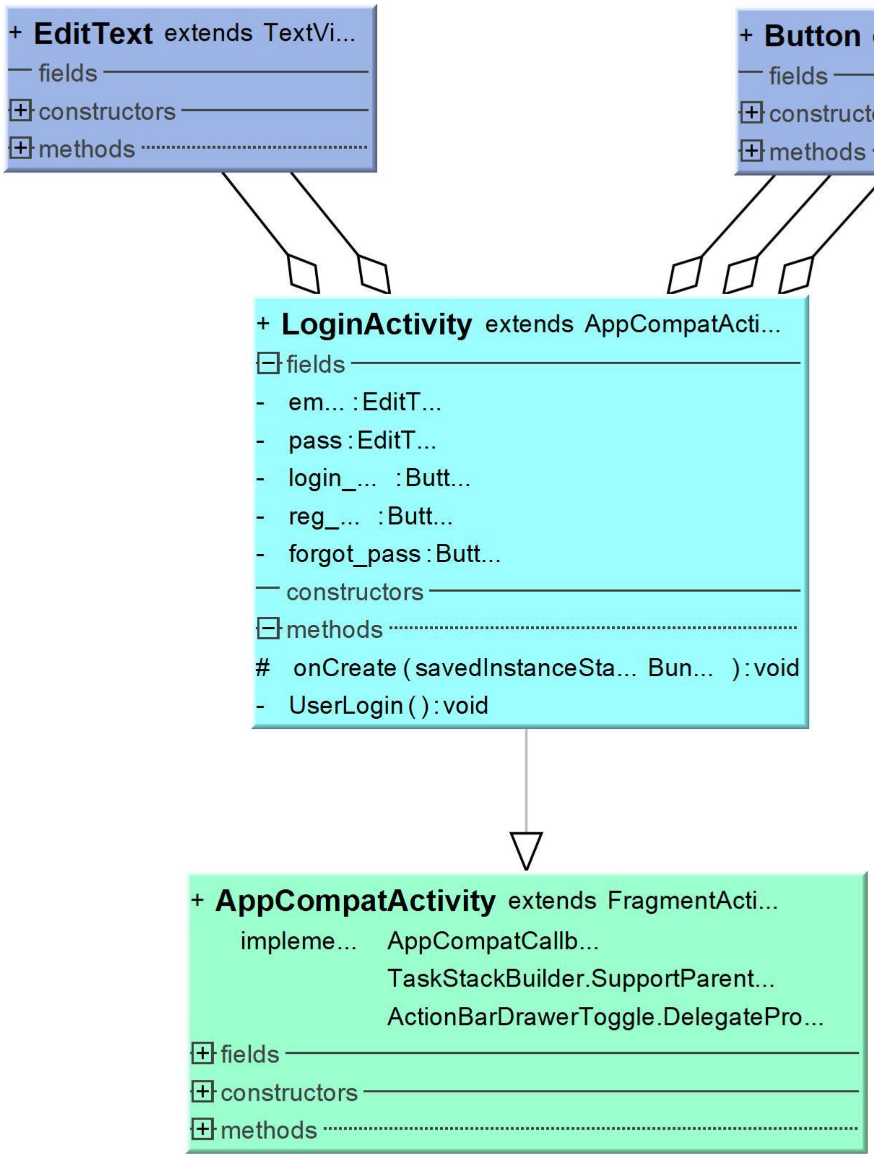
5.1 Static model – UML diagrams

5.1.1 Main Activity

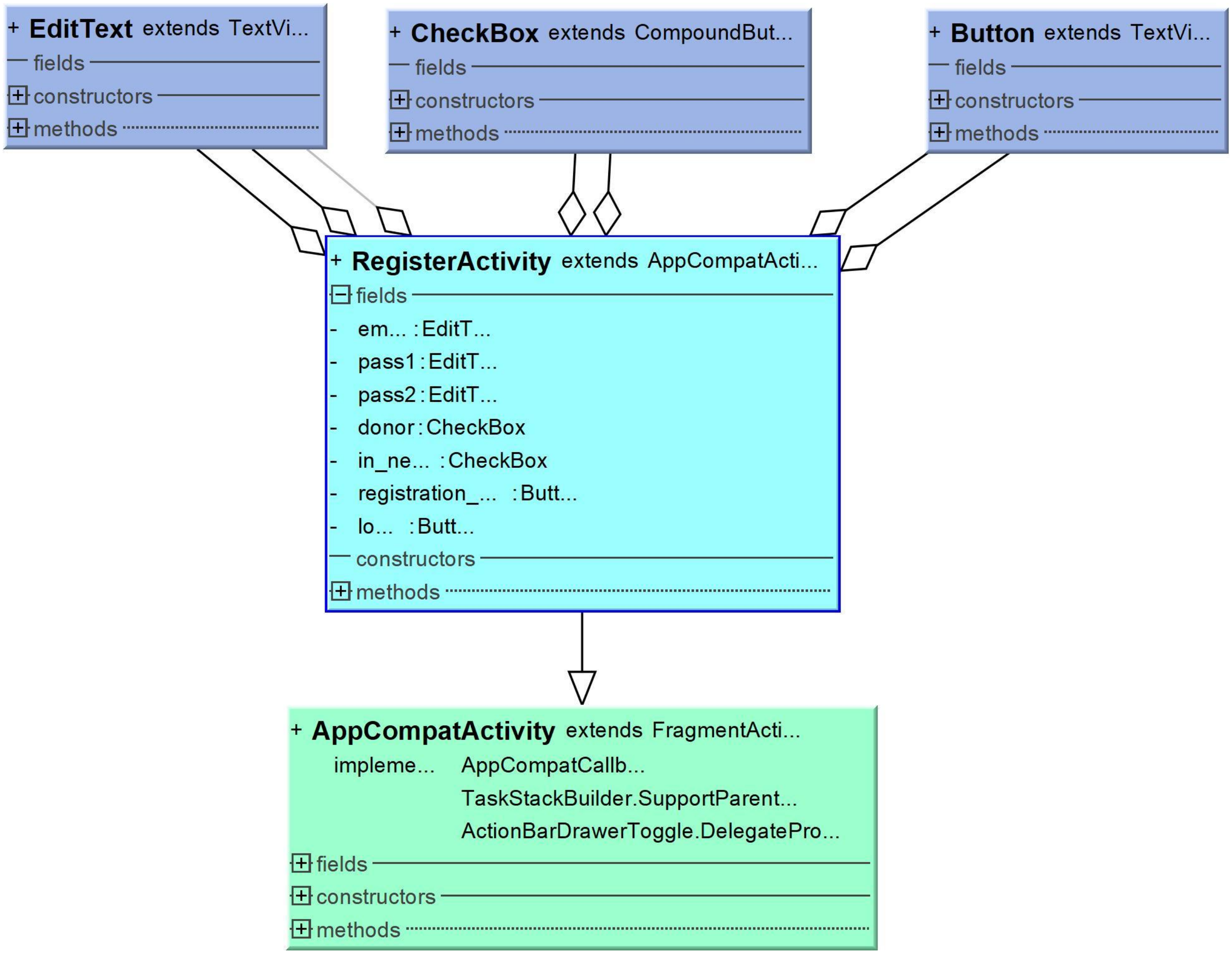


+ Button extends TextVi...
fields
constructors
methods
+ MapView extends ViewGro...
impleme... IMapVi...
MultiTouchController.MultiTouchObjectC...
fields
constructors
methods
+ fin... AnyChartView extends FrameLay...
fields
constructors
methods
+ MainActivity extends AppCompatActi...
fields
- logout_... :Butt...
- form_... :Butt...
- noti_... :Butt...
- prof... :Butt...
- map : MapVi...
- user : FirebaseUser
- referen... : DatabaseReferen...
- userID : String
- percenta... : AnyChartVi...
~ allit... : ArrayList<Strin...
~ trueit... : ArrayList<Strin...
- fin... REQUEST_PERMISSIONS_REQUEST_C... :int
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ setuppiech... (t: int , f: int) : void
+ onRequestPermissionsRes... (requestCode: int , permissions: Strin... , grantResults: int[]) : void
- requestPermissionsIfNecessary (permissions: Strin...) : void
+ getLocationWithCheckNetworkAnd... (mConte... Context) : Locati...
+ FirebaseUser exten... AbstractSafeParcela...
implements UserInfo
fields
constructors
methods
+ ArrayList<E> extends AbstractList
impleme... List
RandomAcc...
Clonea...
Serializa...
fields
constructors
methods
+ AppCompatActivity extends FragmentActi...
impleme... AppCompatCallb...
TaskStackBuilder.SupportParent...
ActionBarDrawerToggle.DelegatePro...
fields
constructors
methods

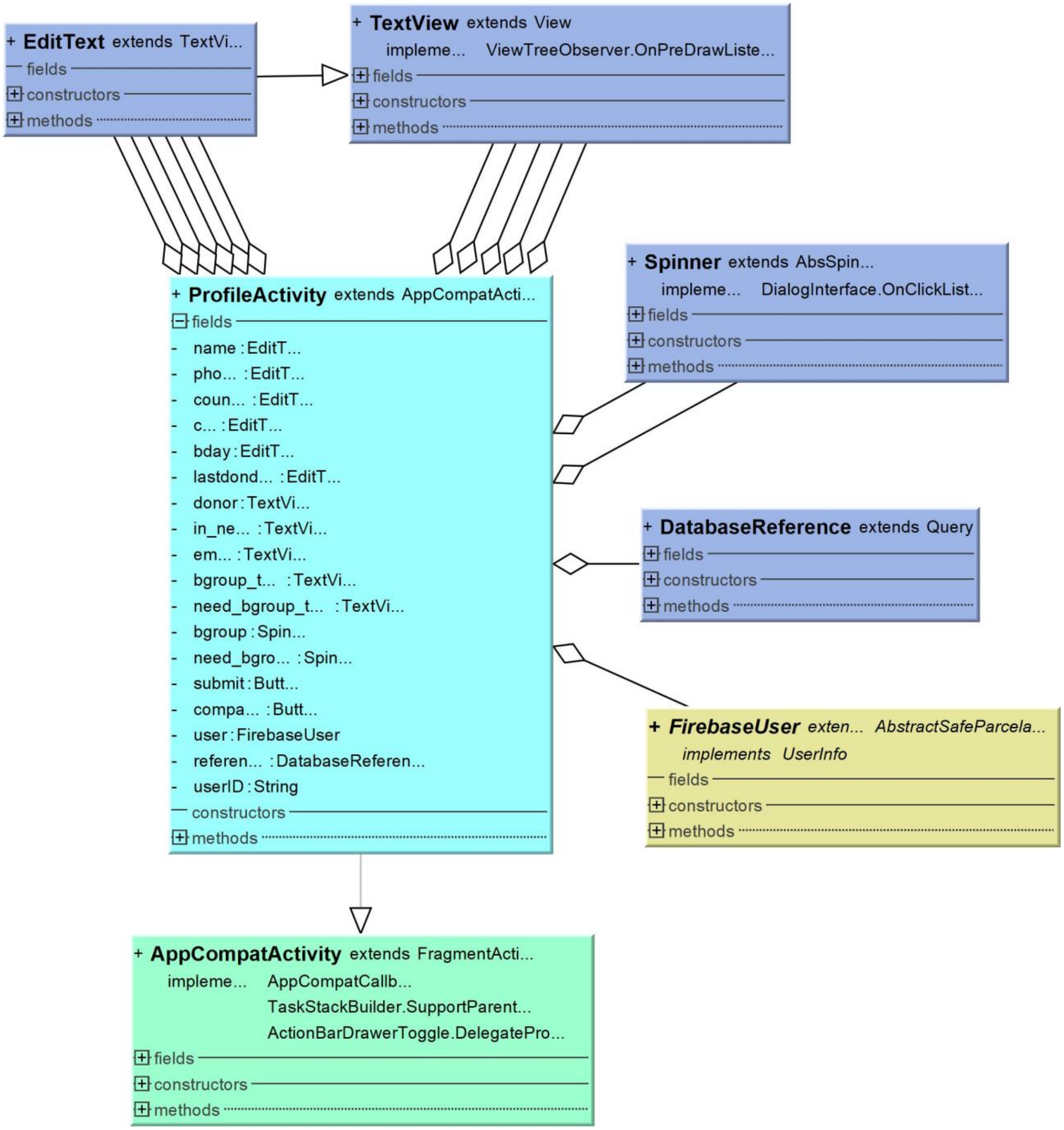
5.1.2 Login Activity


+ EditText extends TextVi...
fields
constructors
methods
+ Button extends TextVi...
fields
constructors
methods
+ LoginActivity extends AppCompatActi...
fields
- em... : EditT...
- pass : EditT...
- login_... : Butt...
- reg_... : Butt...
- forgot_pass : Butt...
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
- UserLogin ( ) : void
+ AppCompatActivity extends FragmentActi...
impleme... AppCompatCallb...
TaskStackBuilder.SupportParent...
ActionBarDrawerToggle.DelegatePro...
fields
constructors
methods

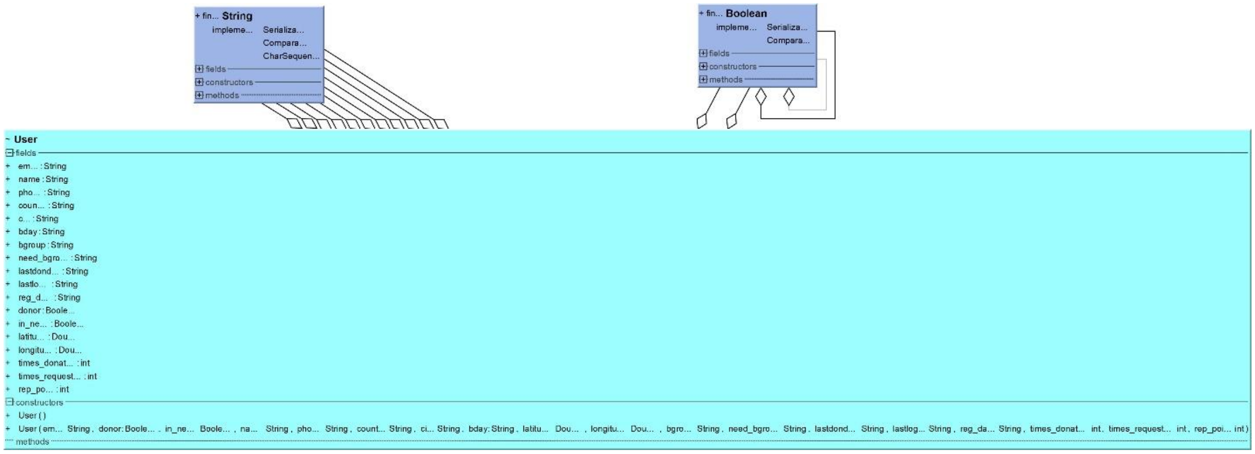
5.1.3 Register Activity


+ EditText extends TextVi...
fields
constructors
methods
+ CheckBox extends CompoundBut...
fields
constructors
methods
+ Button extends TextVi...
fields
constructors
methods
+ RegisterActivity extends AppCompatActi...
fields
- em... : EditT...
- pass1 : EditT...
- pass2 : EditT...
- donor : CheckBox
- in_ne... : CheckBox
- registration_... : Butt...
- lo... : Butt...
constructors
methods
+ AppCompatActivity extends FragmentActi...
impleme... AppCompatCallb...
TaskStackBuilder.SupportParent...
ActionBarDrawerToggle.DelegatePro...
fields
constructors
methods

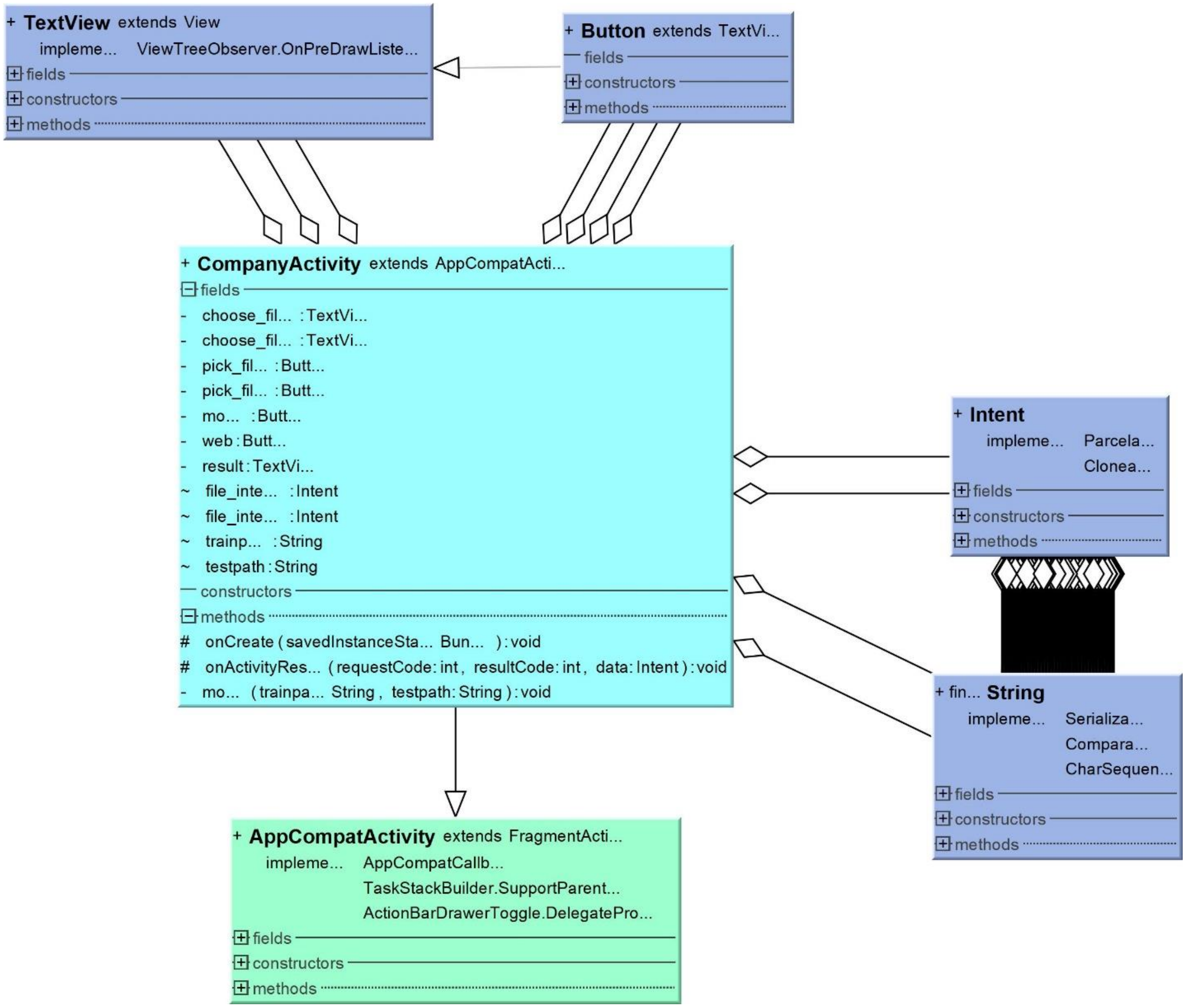
5.1.4 Profile Activity



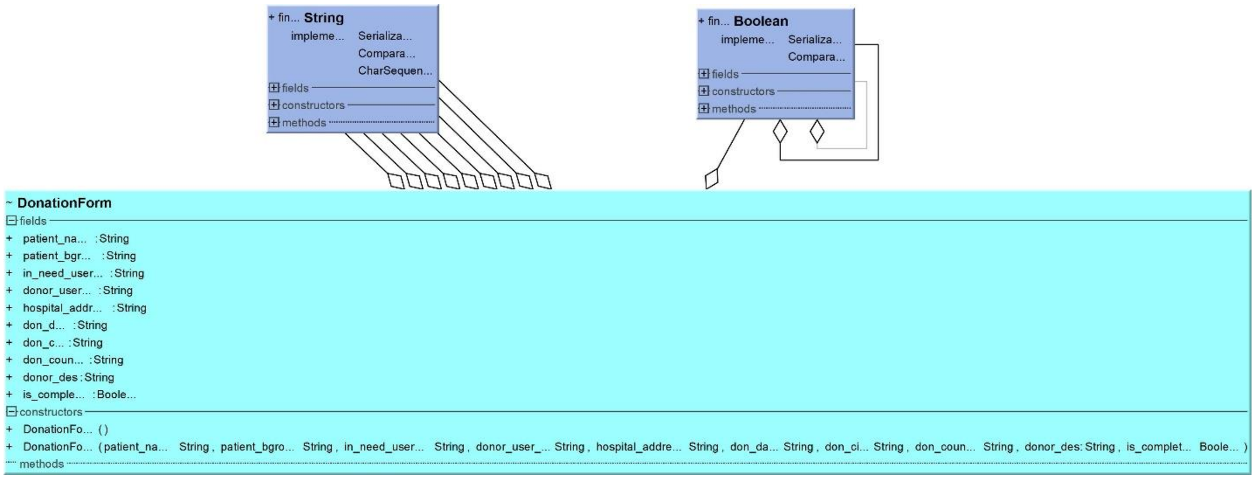
5.1.5 User Data



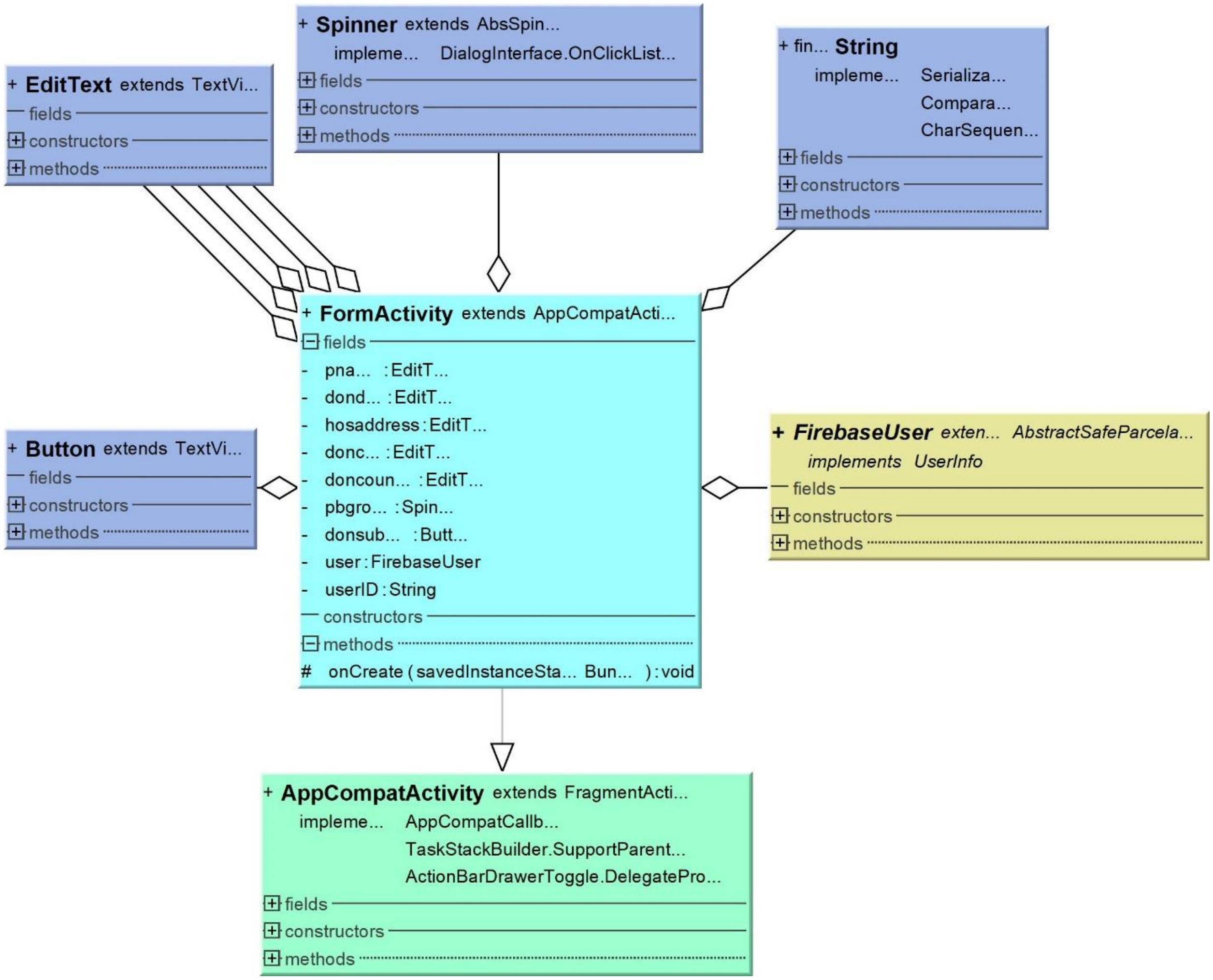
5.1.6 Company Activity


+ TextView extends View
impleme... ViewTreeObserver.OnPreDrawListe...
fields
constructors
methods
+ Button extends TextVi...
fields
constructors
methods
+ CompanyActivity extends AppCompatActi...
fields
- choose_fil... : TextVi...
- choose_fil... : TextVi...
- pick_fil... : Butt...
- pick_fil... : Butt...
- mo... : Butt...
- web : Butt...
- result : TextVi...
~ file_inte... : Intent
~ file_inte... : Intent
~ trainp... : String
~ testpath : String
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
# onActivityRes... (requestCode : int , resultCode : int , data : Intent) : void
- mo... (trainpa... String , testpath : String) : void
+ Intent
impleme... Parcela...
Clonea...
fields
constructors
methods
+ fin... String
impleme... Serializa...
Compara...
CharSequen...
fields
constructors
methods
+ AppCompatActivity extends FragmentActi...
impleme... AppCompatCallb...
TaskStackBuilder.SupportParent...
ActionBarDrawerToggle.DelegatePro...
fields
constructors
methods

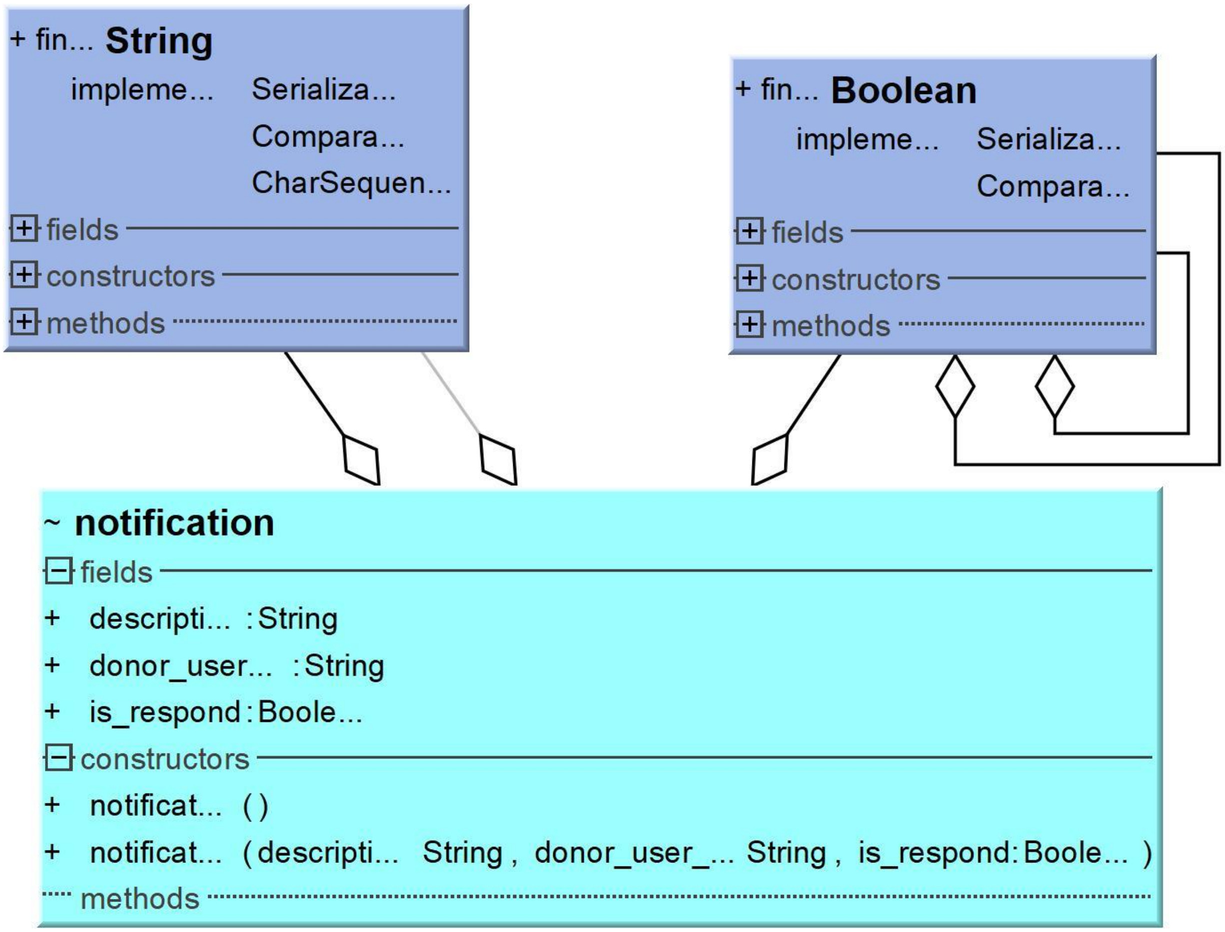
5.1.7 Donation Form



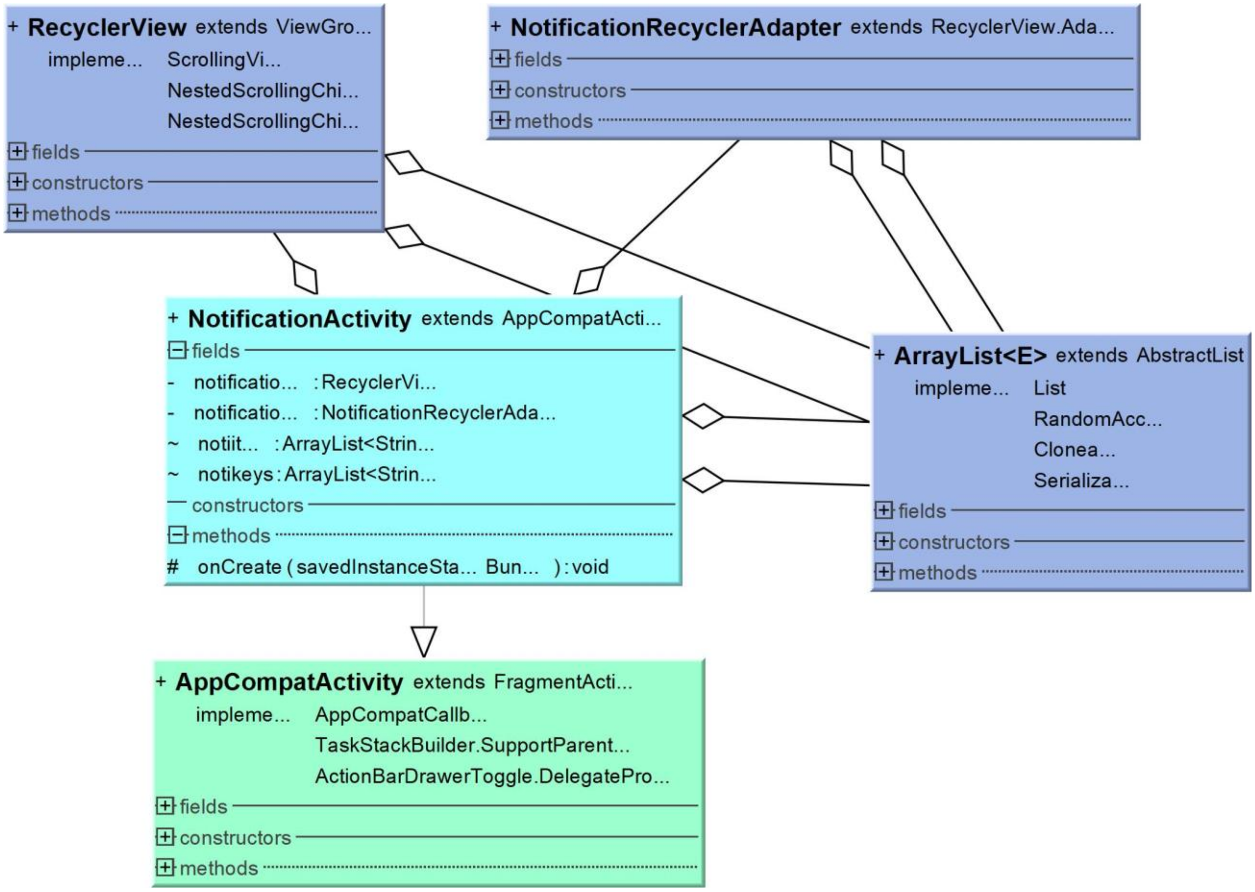
5.1.8 Form Activity



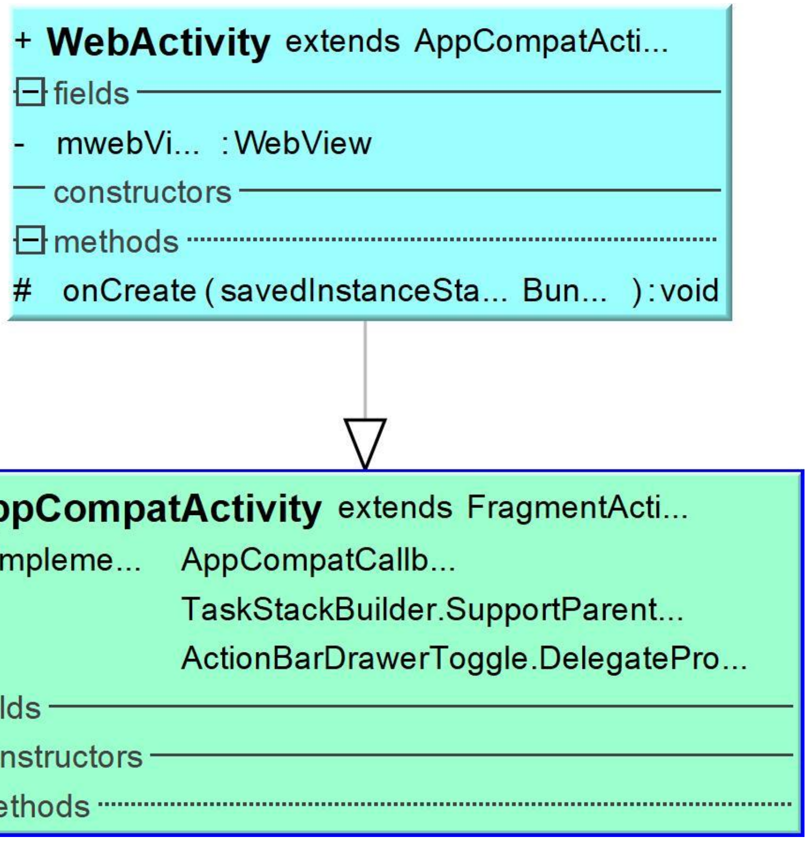
5.1.9 Notification Data:



5.1.10 Notification Activity:


+ RecyclerView extends ViewGro...
impleme... ScrollingVi...
NestedScrollingChi...
NestedScrollingChi...
⊞ fields
⊞ constructors
⊞ methods
+ NotificationRecyclerAdapter extends RecyclerView.Ada...
⊞ fields
⊞ constructors
⊞ methods
+ NotificationActivity extends AppCompatActi...
⊟ fields
- notificatio... : RecyclerVi...
- notificatio... : NotificationRecyclerAda...
~ notiit... : ArrayList<Strin...
~ notikeys : ArrayList<Strin...
constructors
⊟ methods
# onCreate (savedInstanceSta... Bun...) : void
+ ArrayList<E> extends AbstractList
impleme... List
RandomAcc...
Clonea...
Serializa...
⊞ fields
⊞ constructors
⊞ methods
+ AppCompatActivity extends FragmentActi...
impleme... AppCompatCallb...
TaskStackBuilder.SupportParent...
ActionBarDrawerToggle.DelegatePro...
⊞ fields
⊞ constructors
⊞ methods

5.1.11 Web activity


+ WebActivity extends AppCompatActi...
fields
- mwebVi... :WebView
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ AppCompatActivity extends FragmentActi...
impleme... AppCompatCallb...
TaskStackBuilder.SupportParent...
ActionBarDrawerToggle.DelegatePro...
fields
constructors
methods

5.2 Traceability from requirements to detailed design model

5.2.1 General User Requirements

In this application we can see that the user can be log in as a donor or request for blood in the action.

An interactive notification system has been implemented

5.2.2 Company Requirements

In the company activity the environment for the setting the forecasting has been implemented.

The web activity we have implemented the cloud portion of the work

To do the calculation on the cloud Django framework has been used

These were the company requirements.

6 Test Plan

6.1 Requirements/specifications-based system level test cases

6.1.1 Business Requirement

BR#	Module	Application Role	Description
BR1	Login Logout	Members of the BTU and Users	Member: Member will able to login and will have the access to the Company Activity page User: User will able to see own profile.
BR2	Forecast Donor number	Members of the BTU	Member of the BTU will able to forecast the possible donor number before an event based on the registration
BR3	Respond to the Interactive Nonfiction	Users	Users will able to respond the emergency notification
BR4	Blood Request Form	Users	Users will be able to request for the emergency blood request

6.1.2 Function Requirement

FR#	Module	Application Role	Description
FR1	Detect Spam Request	Automatic	If any users tries to spam blood request automatically system will detect
FR2	Show OSM	Users	User’s current location will be auto sync up
FR3	Interactive Notification	Users	Show the nearby emergency blood request

6.1.3 Technical Requirement

TR#	Description
TR1	Email must not be blanked
TR2	Password must not be blanked
TR3	If email and password are valid login
TR4	Notification will not be sent to a donor if s/he cannot able to donate blood
TR5	Automatically the notification will be removed if someone respond to it

6.2 Traceability of test cases to use cases

Testcase#	BR #	FR #	TR #	Testcase	Test Data	Expected	Remark
1	BR 1	---	TR1, TR2, TR3	Verify Log in	Email=naz.niloy1999@gmail.com Password=123456789	Successful log in	Success
2	BR 2	---	---	Forecasting	Train Data=January Test Data=February	47389(with an acc of 99.7)	Success
3	BR 3	---	---	Blood Request Form	User=YY User Group=In need	Submit a blood request Form	Success
4	BR 4	FR 3	TR4, TR5	Interactive Nonfiction	User=XX User Group=Donor User=AA User Group=Donor	See notification of emergency blood request And after responding the notification get removed from everyone	Success
5		FR 2		Show OSM	User=XX User Group=Donor	Fetch current location information	Success
6		FR 1		Spamming detection	User=ZZ User Group=In Need	The account gets suspended	Success

6.3 Techniques used for test generation with the requirement traceability matrix

The testcase of the application has been checked with the requirement traceability matrix

Business Requirement	Functional Requirement	Technical Requirement	Testcase ID
BR1			1
BR2			2
BR3			3
BR4			4
	FR1		6
	FR2		5
	FR3`		4
		TR1, TR2, TR3	1
		TR4, TR5	4

6.4 Assessment of the goodness of your test suite

The application has been tested with different consideration in mind. As the in the testing phase there has been only small number of active users have been considered so further testing is required to ensure a good performance.