Organ Donation Android Application

A project submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
INFORMATION TECHNOLOGY

Submitted by:
Yash Mudgal
Roll No.: 11912080
Group No.: 220

Supervised by:
Dr. Mukesh Mann
Assistant Professor

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
SONEPAT -131201, HARYANA, INDIA

ACKNOWLEDGEMENTS

Doing a project is like a bridge between theoretical knowledge and practical knowledge. With immense pleasure I would like to express my special thanks of appreciation to my teacher **Dr. Mukesh Mann** who gave me this golden opportunity to work on this great project "**Organ Donation Android Application**".

I also take this opportunity to express a deep sense of gratitude to **Dr. M.N. Dohja** Director, IIIT Sonepat for his cordial support, valuable information and guidance.

Working on this project helped me in doing a lot of Research and I come to know about so many things.

Lastly, I thank almighty, my parents and friends for their constant encouragement without which this project would not be possible.

Yash Mudgal

SELF DECLARATION

I hereby declare that work contained in the project titled **“Organ Donation Android Application”** is original. I have followed the standards of project ethics to the best of my abilities. I have acknowledged all source of information which I have used in the project.

Yash Mudgal

l1912080
Department of Information Technology Indian
Institute of Information Technology,
Sonepat-131201, Haryana, India

CERTIFICATE

This is to certify that **Mr. Yash Mudgal** has worked on the project entitled **“Organ Donation Android Application”** under my supervision and guidance.

The content of the project, being submitted to the **Department of Information Technology, IIIT Sonepat,** for the award of the degree of **Bachelor of technology** in **Information Technology,** are original and have been carried out by the candidate himself.

This project is not submitted in the full or part for the award of any other degree or diploma to this or any university.

Dr. Mukesh Mann
Supervisor

Department of Information Technology Indian
Institute of Information Technology,
Sonepat - 131201, Haryana, India

ABSTRACT

Name of the student **Yash Mudgal**, Roll No. **11912080** Degree for which submitted **Bachelor of Technology** Department of **Information Technology**, IIIT, Sonepat.

Project Title: **Organ donation android application**

Name of the thesis supervisor: **Dr. Mukesh Mann**
Month and year of the thesis submission: **December 2020**

Organ Donation – A Noble Act That Can Save Lives!

Organ Donation is an extremely noble and honorable act that a person can do to contribute from his/her share of duty to benefits the society and make this world a beautiful place to live.

Our android-based application is a very small step from our side to facilitate this noble cause of organ donation and to make the whole process more efficient and transparent.

If someone wants us to abstract our app in one line, it could possibly be:

CELEBRATING THE GIFT OF LIFE

Organ donation is the donation of biological tissue or an organ of the human body, from a living or dead person to a living recipient in need of a transplantation. This android app will help both donors and recipient of organs. This app will bring organ donors, organ recipient, doctors and hospitals all at a same place and create a healthy environment for them to help each other. It will provide a medium through which an organ seeker can directly connect to the organ provider and can simultaneously take consult of doctors and hospitals. This system will have two main entities, namely users and admin. A user entity will contain organ donor, organ recipient, doctors and hospitals. An admin can login using his/her login credentials. It can view the list of all users, verify them and can add new doctors and hospitals to the app. It can view & verify the organ donation, organ requirement requests from the users. It can also access the database of the app.

A user can register and login using their login credentials. They can update their profiles and the updates will be displayed once verified. A user can create organ donation or organ requirement request along with appointment with doctor or hospitals. A user can cancel or update its request. A user can directly chat with another user through the app and can also ask for personal details.

The app will conduct the whole process according to the rules and regulation of Government of India. Organ donation process will be carried out under the guidance of registered doctors or hospitals. The app is whole heartedly developed for the purpose of connecting organ seekers to organ donors.

Users will get regular notifications and message alerts for their respective requests created on the app.

The app will be supportable in all android mobile phones having android 4 or above. We have used android studio IDE, java language and API`s for the implementation of the app.

LIST OF ABBREVIATIONS

App	Application
SDLC	Software Development Life Cycle
OTP	One-time password

LIST OF FIGURES

List of Tables

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1.1 Introduction

Background

Organ Donation system can be trace from 1954, the first organ transplant was done by Dr. Joseph Murray who transplants kidney from Ronald Lee Herrick to his twin brother Richard Harrick. After 36 years, Dr. Joseph Murray honored with the Nobel Prize in Physiology or Medicine for his pioneering work in organ transplantation.[1] The National Organ and Tissue Transplant Organization (NOTTO) functions as the apex body for activities of relating to procurement, allotment and distribution of organs in the country [2]. Transplantation of Human Organs Act (THOA) 1994 was enacted to provide a system of removal, storage and transplantation of human organs for therapeutic purposes and for the prevention of commercial dealings in human organs [3]. But in big countries like India, it is very difficult to handle whole process by a single agency or department. And if you are live in a small cities or villages it is very difficult to register in NOTTO and follow whole process.

Sometimes it is more difficult to find out the organ donors; we hardly know one or two. Either we found there are fewer possibilities of availability. What happens if when someone needs organ& does not understand what to do? For lessen their efforts in search of donors, we are studying a mobile app using Android and java. The Android app helps to find out organ donors who are available in that particular area where the user is looking for with their exact locations. For using this app user has to only do register.

"Give yourself and those in need an elixir of life by pledging your organs" by Mohith Agadi.

1.2 Problem outline

The propose project is to solve the following problem:

"Create an efficient solution for smooth and convenient way to find and connect organ donor and organ recipient in the specified time frame".

An organ transplantation can save a precious life, as there is no substitute of human organs. Every day many organs are required in hospitals for transplantation in case of accident and organ failure. With a growing population and advances in medical treatments and procedures requiring organ transfusions, the demand of organ continue to increase. In India many people are losing their lives every day in emergency situations because we are suffering from lack of organ in organ Banks, and they do not receive the organ timely. Their relatives and friends start searching for a donor to help, but there is no guarantee whether they will find a donor or not. On the other hand, there are a lot of people who are willing to help and donate.

There are numbers of existing systems that facilitate donation process but still we are far behind to match the organ recipient and organ donor numbers. Besides, we propose to use the latest technologies and the available tools to find a modern system which fills the gap and provides an organized solution. Our system has a quick mean to find the donors easily by their nearest location and in available time frame. This app will search organ donor for needy people and make it easier than ever before it used to be.

This app will also aware people about the importance of organ donation and encourage them to apply for organ donation at least after death. It will also help to bust the myths related to organ donation.

1.3 Problem Objectives

- To decrease the communication gap between organ donors and organ recipients
- To make the search of organs easy, fast & secure
- To make a common platform where organ donor & organ recipients can take consultation of a doctor simultaneously
- To encourage the noble cause of organ donation

1.4 Development methodology

The system would be developed using the **Sashimi Model.** The main idea behind the model is that we allow to overlap the different phases of software development lifecycle. For example, while we are working on the requirements, instead of waiting for the requirement phase to complete, we can start with design while the requirements are being created i.e., we don't have to wait for the previous phase to finish to start the next phase. We can overlap our phase so much so that sometimes we can have even more than one phase be overlapping.

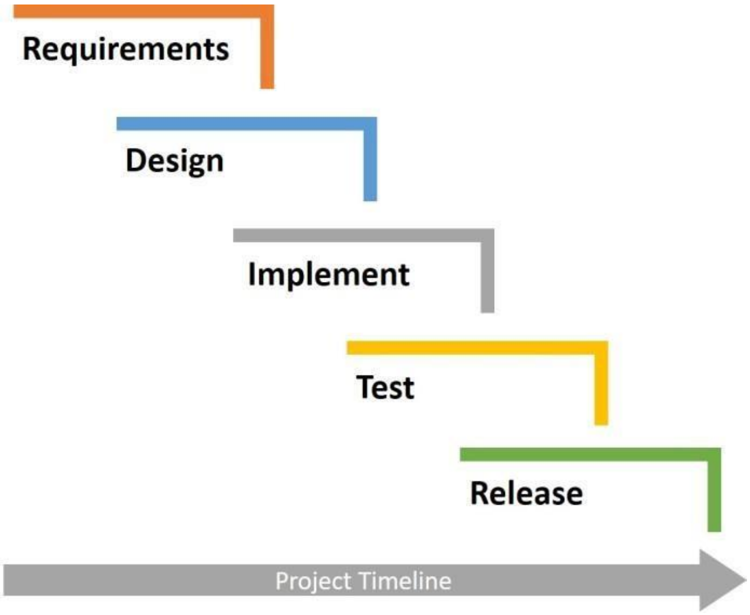


Fig 1.1 Sashimi Model

The sashimi model process is a way of organizing a waterfall with feedback. In this model we can take feedback from one phase to enhance the other phase. This model works just like waterfall method the only difference is that sashimi model also has a feedback line to take feedback from one phase to implement it in the next phase. The feedback is necessary for us as we are developing an application for the first time & we don't have a concrete idea how to make our application so we can improve our application at each phase using feedback from previous phases of development. We decided to go with sashimi model to develop our project because it decreases the time of development & people with different skills can start working without waiting for a specific phase to be completed.

One of the main drawbacks of this process is that it may result in rework, and waste of time which would be a precious asset to waste unnecessary.

1.5 Scope

- The application will help recipients in getting donor information easily at the time of emergency.
- Recipients and donors can also contact directly to doctors and hospitals.
- The app can also help various hospitals and doctors to get organs at the time of emergency.

1.6 Limitation

- This app only works when there is active internet connection on smartphone.
- The system requires a large database, which can increase initial costs.
- This app will not run-on iOS.
- We don’t provide any guarantee to organ seeker or organ donor that their requests will definitely completed, the whole process depends on the availability of organ donors and organ seekers respectively.

1.7 Organization of Project

The total chapters included in this project are:

1. **Introduction**

   In the first chapter we discussed about basic introduction of application. Firstly, we discussed about background related to organ donation then we discussed about why we choose this topic. Further, we discussed about project analysis and then we also discussed about our project objectives and this chapter also includes our methodology that we follow to complete our objective, we also discussed scope and limitations of our project in this chapter.

2. **Design**

   In the design chapter, we discuss about structure and designing of the proposed system. In this chapter we discussed about "How" we going to implement our code. It also contains various technologies that are used in implementing our application. This chapter also contains various diagrams such as UML diagrams, class diagrams, sequential diagrams that provides path to our code.

3. **Implementation**

   It includes implementation. It includes implementation steps included in making project. It includes necessary codes related to project.

4. **Conclusion and future scope**

   It includes the future scope related to topic or project and also conclusion of the system working and its need. This chapter also contains what further technologies we will include in our project to make more user interactive.

1.8 Summary

Organ donation android application is an app created to eliminate the communication gap between organ donor and organ seeker. It will bring organ donor, organ seeker, doctor and hospital at a same virtual place where they can interact freely and help each other. The app is created to make organ donation process efficient and transparent. It will help to control and minimize the human organ's trafficking to some extent. As it directly connects the organ donor and organ seeker without any intervein of a middleman. The app also provides convenient way for doctors and hospitals to get organs at the time of emergency.

The app has two entities, namely user and admin. An admin can login to the app using its login credentials. It can view and manage the list of users and add new doctors and hospitals to the application. It can also access the database of the app. An admin can also view and verify the requests made by users and can chat with a user if required. All the information on the app will be stored in a database.

The other entity is user. A user can be an organ donor, organ seeker, hospital and doctor. Each user will have different access level and functionality according to their request. An organ donor can make a request for organ donation. An organ seeker can make a request for organ requirement. An organ donor or seeker can also consult with the doctors and hospital associated with the app. A user can also delete its request and can update profile which will be done after OTP verification.

The application will send regular notification and message alerts to the users about the status of the requests. The application will also store personal details of a user which will be secured by the app. The app will also aware people about the importance of organ donation and encourage them to apply for organ donation at least after death. It will also help to bust the myths related to organ donation. The app will follow all the rules and regulation formulated for organ donation process by the specified by the concerned authority.

The app would help recipient, donor, hospital and doctors to get organs at the right time. This would help in saving many precious lives. The project can be supportable in all android devices. The only requirement is an active internet connection.

This is also one of the limitations of the project. Not everyone has a device with active internet connection in our country. Another limitation is the requirement of a large database because of lot of information that would need to be stored. In the coming chapters we would discuss about the further phases of the project.

The app doesn’t provide any guarantee to organ seeker or organ donor that their requests will definitely completed, the whole process depends on the availability of organ donors and organ seekers respectively.

CHAPTER - 2
DESIGN

In the design chapter, we discuss about structure and designing of the proposed system. In this chapter we discussed about "How" we going to implement our code. It also contains various technologies that are used in implementing our 20 application. This chapter also contains various diagrams such as UML diagrams, class diagrams, sequential diagrams that provides path to our code.

The main focus of the analysis phase of Software development is on "What needs to be done". The objects discovered during the analysis can serve as the framework or Design. The class's attributes, methods and association identified during analysis must be designed for implementation language.[4]

Emphasis shifts from the application domain of implementation and computer such as user interfaces or view layer and access layer. During analysis, we look at the physical entities or, that is which players and how they cooperate with each other to do the work of the application.

Here his goal is to design the class that we need to implement the system the difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way of interact with a user or present information.

Design Process:
During the design phase the classes identifies in object-oriented analysis Must be revisited with a shift focus to their implementation. New classes or attributes and Methods must be an added for implementation purposes and user interfaces.

The following are some of the vies of software design life cycle. They are:

- UML Diagrams.
- Sequential Diagrams.
- Activity Diagrams.
- State Chart Diagrams.
- Data Flow Diagrams.

Functional Requirements:

1. Build an android application for a user (i.e., organ donors, patients, doctors & hospitals) to make a **request** of donation or **requirement** of an organ.
2. **Secure** user information using id and **password.**
3. User can **view** and **update** their profiles.
4. Can view & update the list of all users associated with the application.
5. User can use **filters** to classify the organ in a better way.
6. Application must be **synchronized** with **data base** to store user's information and requests.

Non-Functional Requirements:

1. The whole process must be transparent & government's law abiding.
2. Application should support android 10 and later versions.
3. Provide link of how organ donation process works page.
4. Provide link to contact support team.
5. Provide link for government's rule and regulation about the whole process.

Based on these functional requirements, we can broadly identify six major topics as:

1. User’s Request
2. Privacy – Protecting user’s data
3. View & update profile
4. View, update & list of users
5. Filters
6. Database synchronization.

User Stories:

Epic#1: User's Request:

1 As a user(donor), I want to request for organ donation.

2 As a user(recipient), I want to request for organ requirement.

3 As a user (i.e., organ donor, patient), I want to make appointment with doctor or hospitals associated with the application.

4 As a user, I want to track the status of my request.

Epic#2: Privacy – Protecting user data:

1. As a user, I want to make sure that my information must be safe and secure.
2. As a user, I want to make user that my information must not be shared with anyone without my consult.
3. As a user, I want to make sure that each user profile must be verified.

Epic#3: View & update profile:

1. As a user, I want to login and view my profile using user id and password.
2. As a user, I can update my profile using one-time password (otp) verification completes.
3. As a user, I want updates should be published only after verification.

User Story#4: View, update & list of users:

1. As an administrator, I can review & updated users list.

Epic#5: Filters in application:

1. As a user, I want filters for easy and fast search of an organ.
2. As a user, I want filters for searching donors and their location.
3. As a user, I want filters for searching doctors.
4. As a user, I want filters for searching hospitals.

Epic#6: Database synchronization:

1. As a user, I want database must be secured.
2. As an administrator, I can access the database.

- **To visualize the functional requirements of a system we use Use-Case diagram (simplest version of UML diagram)**

2.1 Use Case Diagram for Organ Donation Android Application:

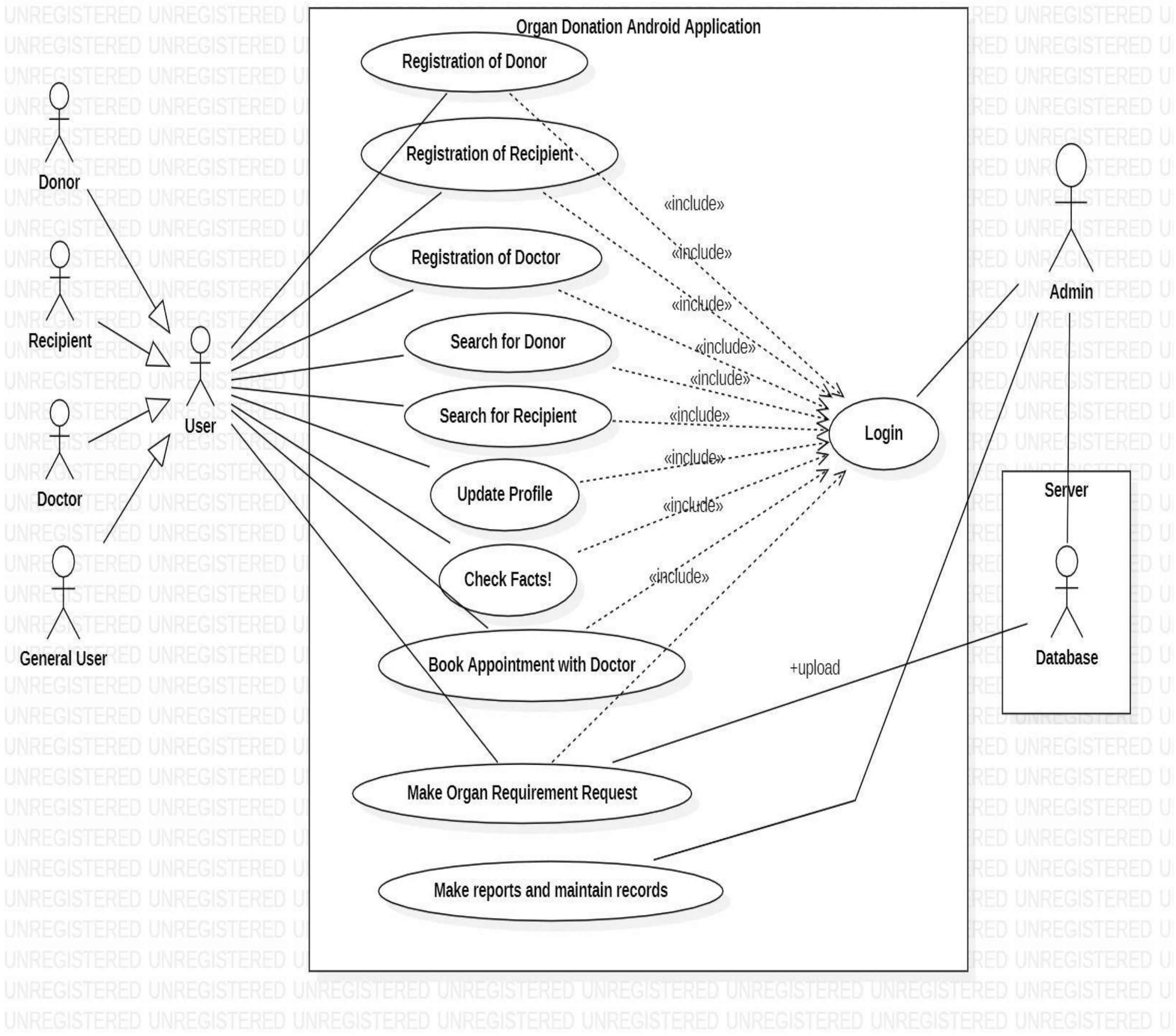


Fig 2.1

- These are some main classes of Organ Donation system:

1. User( )
2. Login( )
3. Admin( )
4. Request( )
5. Organ( )
6. UserList( )

2.2 Interaction Between Classes:

Diagram showing integration between different classes of Organ Donation system:

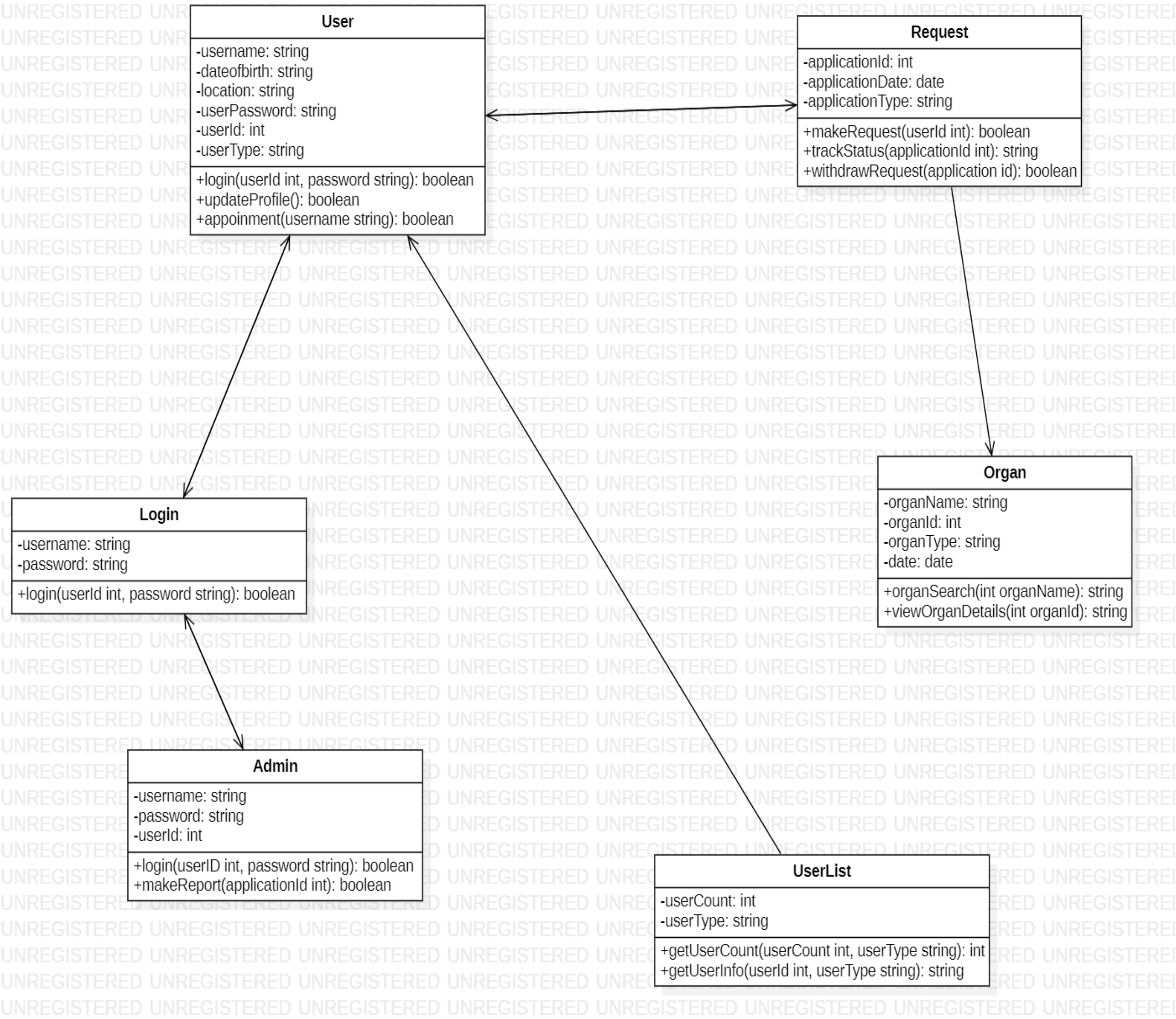

User
-username: string
-dateofbirth: string
-location: string
-userPassword: string
-userId: int
-userType: string
+login(userId int, password string): boolean
+updateProfile(): boolean
+appoinment(username string): boolean
Request
-applicationId: int
-applicationDate: date
-applicationType: string
+makeRequest(userId int): boolean
+trackStatus(applicationId int): string
+withdrawRequest(application id): boolean
Login
-username: string
-password: string
+login(userId int, password string): boolean
Organ
-organName: string
-organId: int
-organType: string
-date: date
+organSearch(int organName): string
+viewOrganDetails(int organId): string
Admin
-username: string
-password: string
-userId: int
+login(userID int, password string): boolean
+makeReport(applicationId int): boolean
UserList
-userCount: int
-userType: string
+getUserCount(userCount int, userType string): int
+getUserInfo(userId int, userType string): string

Fig 2.2

2.3 Sequence Diagram:

Use case and class diagrams are static diagrams. But we if need to show how objects interacts with each other during run time then we need sequence diagrams.

When our objects get created and how long they are around?

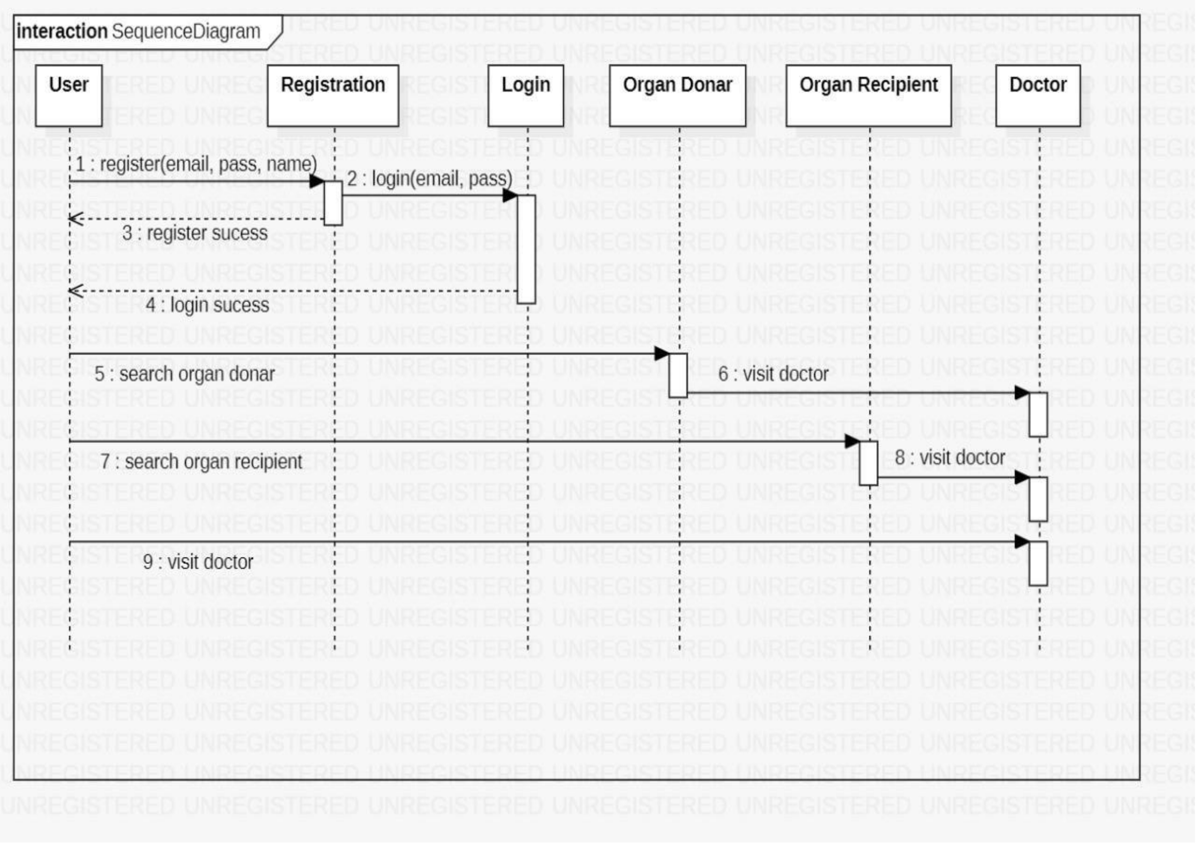


Fig 2.3

2.4 Activity Diagram:

Activity diagrams are behavioral diagrams, which are used to describe the workflows.

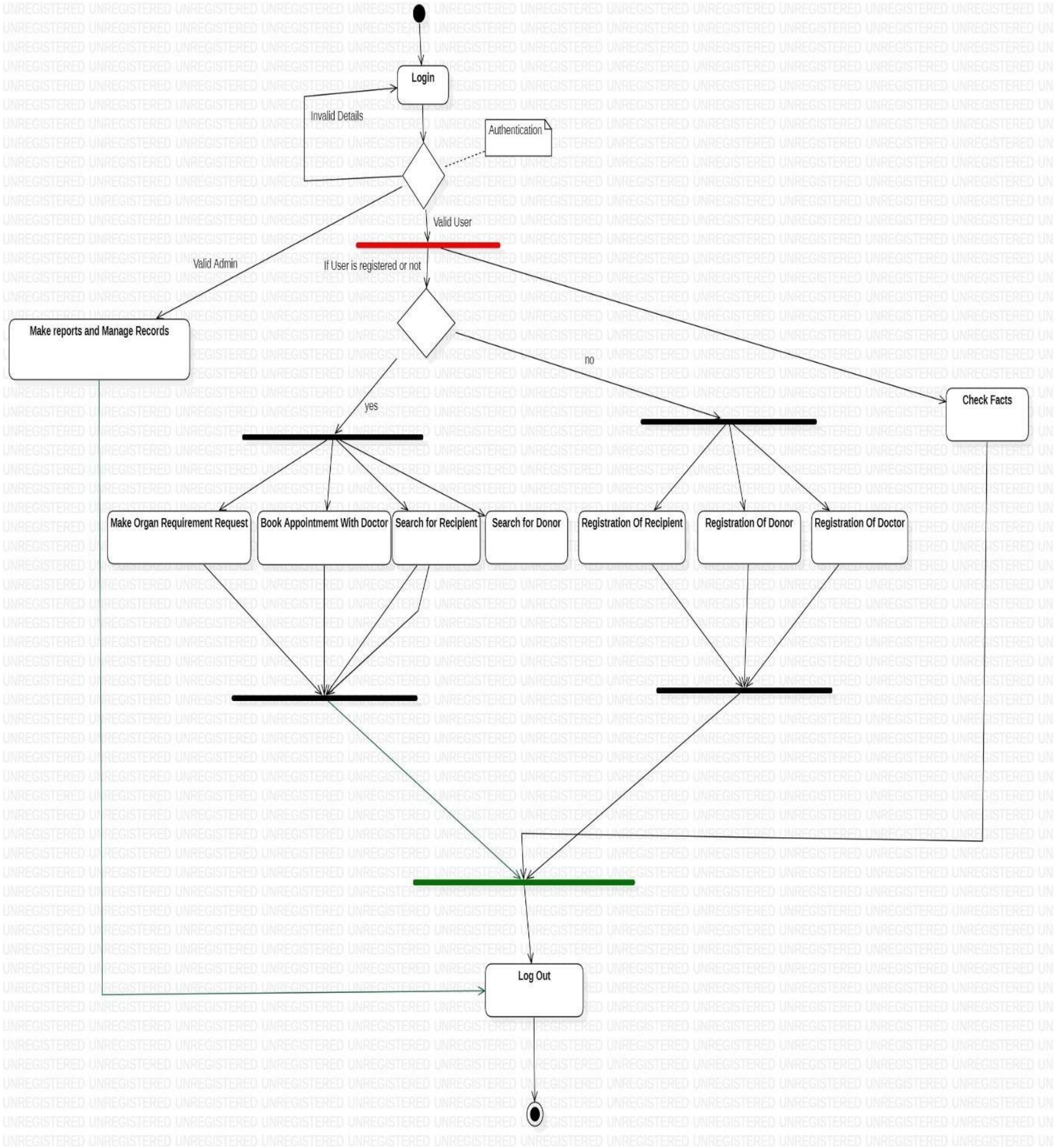


Fig 2.4

2.5 State Chart Diagram:

How an object transition from one state to another over its life time, this behavior can be modeled using State chart diagram.

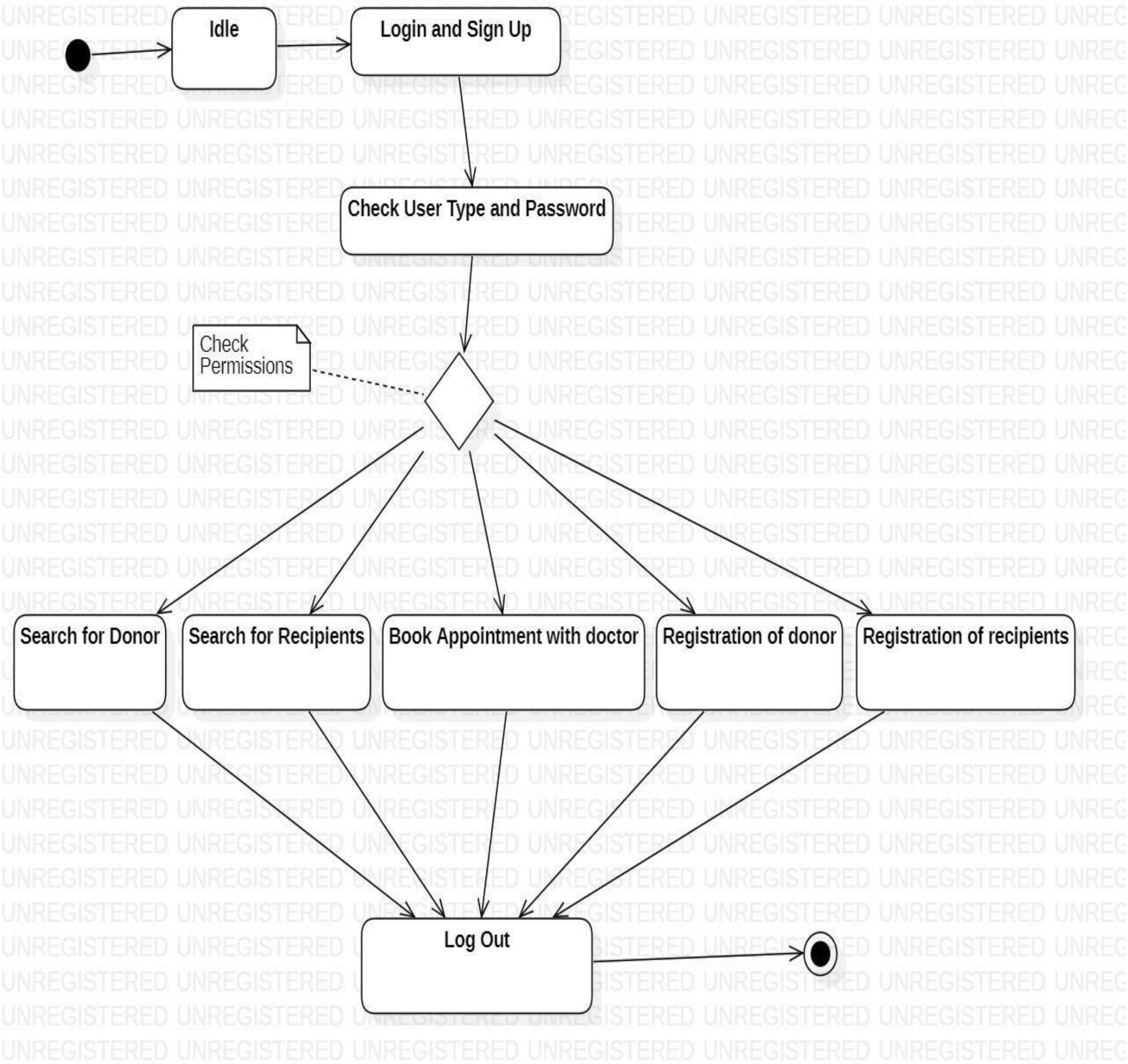


Fig 2.5

2.6 Data Flow Diagram (DFD):

A **data-flow diagram** is a way of representing a **flow** of **data** through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself.

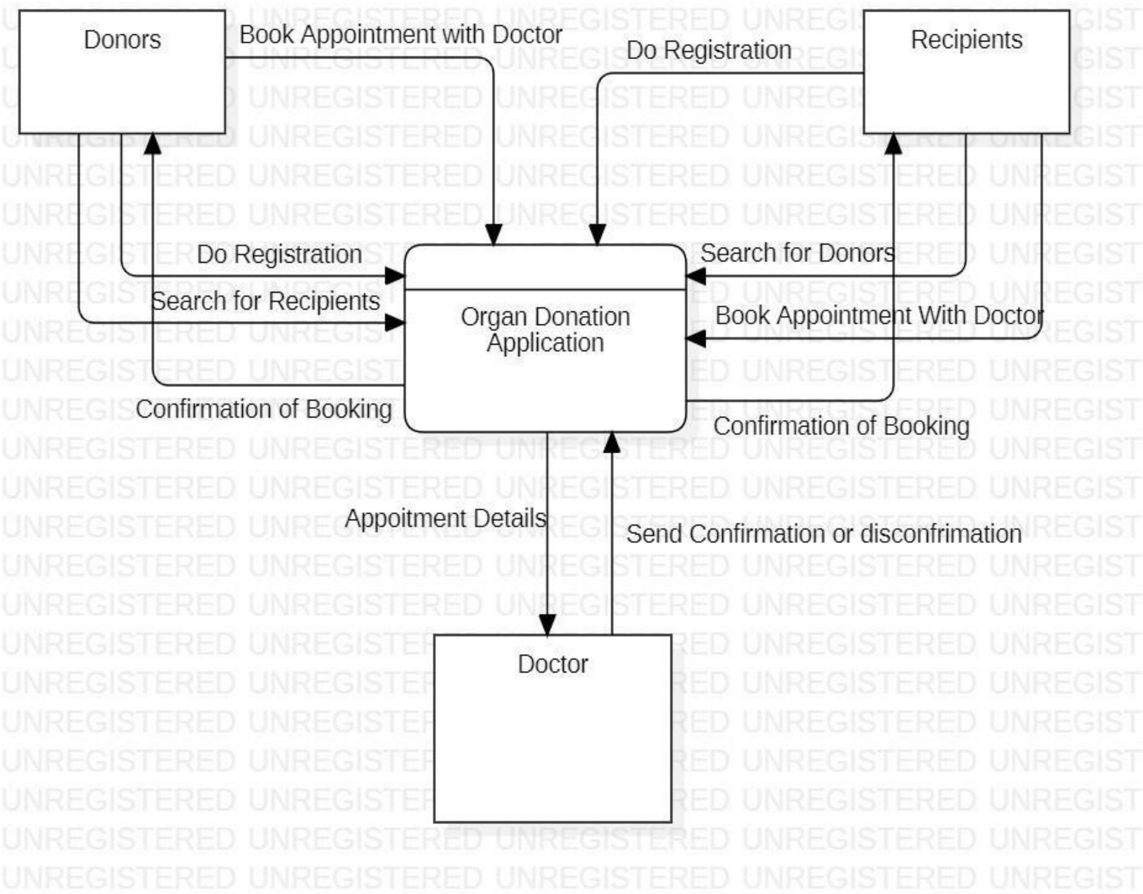


Fig 2.6

CHAPTER 3

IMPLEMENTATION

3.1 Software and Hardware Requirements:

3.1.1 Software Requirement Specifications:

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system which enables the user to interact with several physical hardware devices.

The minimum software requirements specifications for developing this project are as follows:

Designing Front End: Java. Backend: Firebase.
UML: Star UML.
IDE: Android Studio.

3.1.2 Hardware Requirement Specifications:

The collection of internal electronic circuits and external physical devices used in building a computer is called hardware.

The minimum hardware requirement specifications for developing this project are as follows:

Processor: i3 (8th gen)
RAM: 8GB RAM
Hard Disk: 500GB

3.2 Technology Used:

This project is mainly based on Firebase for real-time database an android application to be developed in Android Studio by using Java as a programming language.

3.2.1 Java:

Java is a programming language, created in 1995. It is developed by James Gosling. It is basically owned by Oracle, and more than 3 million devices run java. It has many uses. Some of the uses are as follows:

- Mobile Applications(apps).
- Desktop Applications.
- Web applications.
- Web servers and application servers.
- Games.
- Database connection.

It is a class-based, object-oriented programming language (OOP) that is designed to have as few implementation dependencies as possible. Due to OOP, it is in demand continuously. Java codes are mainly compiled by bytecode that can run on any Java Virtual machine (JVM) regardless of the underlying computer architectures. Java runtime provides dynamic capabilities such as reflection and runtime code modification that are typically not available in traditional compiled languages.[4] It is very much popular in client-server web applications.

We use java as our main programming language in this project. We do full code in java in android studio ide. We develop login page, sign-up page, home page, registration forms and many other pages.

3.2.2 Firebase:

We are using Google’s mobile application development platform **Firebase** for handling backend requirement of our mobile application. It provides a variety of tools and services to develop mobile applications. It seems us a very user-friendly way to store & retrieve data of the application as it does not require any SQL knowledge to store and access data in Firebase. It is super simple to connect our application to the firebase console and google has also provided tons on firebase documentation tohelp us. Some of the main services that firebase provides under a same platform are:

- **Authentication** which helps us to maintain Login & Registration of users in our app
- **Realtime Database** is a cloud-hosted NoSQL database that lets you store and sync data between users in Real-time even we don’t have to refresh the firebase console to see the changes in Real-time database.
- **Storage** allow us to store big size data file such as images, text file etc. & provides a link of the file that we can store in Real-time database to access it from our application.
- **Cloud Messaging** allows admin or developer to send In-app notification to the app users. A developer can send notification to all users at a time or can send notification to some specific users.

These are some main services of firebase that we have used in our application. There are many more services provided by firebase such as In- App Messaging which we try to incorporate in our application later. We have used Real-time database to store information about organ donors, organ recipients and doctors.

Data in Real-time database is stored in a node tree like structure every node have two components name & value, name is used to access the value stored in the node & to access a node we have to go all through the root node to that particular node.[5] The Real-time database can be visualized as a one big JSON object that the developers can manage in real-time. We have used firebase because Real-time syncing makes it easy for our application users to access their data from any device conveniently. Another amazing feature of firebase that incline us toward itself is that when users go offline, the Realtime database SDKs use local cache on the device to serve and store changes.

When the device comes online, the local data is automaticallysynchronized. Firebase not only used for development & testing of the application but can also be used for engaging audience for the application which is a must need of our application to aware and encourage our audience toward noble cause of organ donation.

3.2.3 Android Studio:

We are using Android Studio IDE to develop our appoint is easily available for download on windows, macOS and Linux based operating systems.

It has many good features such as:

1. Gradle- based build support.
2. Android -specific refactoring and quick fixes.
3. Support for Android Wear apps.
4. Built-in support for Google Cloud Platform, enabling integration withFirebase Cloud Messaging and Google App Engine.[6]

- We have built our app in Android Studio IDE with the use of java language.

3.2.4 Git:

We use Git for versional control. It has many good features and have many benefits such as:

1. You have access to all versions of all files in Git repository at any time. It's impossible to lost any part or section of code.
2. Multiple developers can work on one project at the same time without disturbing each other, and without losing any changes made by different members of group.

3.3 Front End:

In this section, we discussed about front end of our project. In this section, we include screenshots of our front end-based pages.

3.3.1 Modules in front end:

The system comprises of 2 major modules with their sub-modules as follows:

3.3.1.1 User

- **Register:** User can register and obtain credentials.
- **Login:** User can login using credentials.
- **Forget Password:** send an email with OTP, and reset Password.
- **Home:** There will be three slots Recipient, Doctors and Donor
- **Profile:** User can View and update their profile
- **Change Password:** User can change their password in case of security.
- **Recipient:** User can manage request by adding and updating request. User can also check the assigned doctor and donor etc. and can Chat with the Admin**.**
- **Doctors:** select and view doctors list and also their details
- **Donor:** User can manage donors. they can view and manage all previous donations.
- **Contact us:** User can write an email

3.3.1.2 Admin

- **Login:** Admin can login using credentials.
- **Manage Doctors:** Admin can view all Doctors and they can add, update and delete Doctors
- **View Organs Request:** Admin can view and assign Doctor, Donor and note of the meeting details etc. they can also Chat with User
- **View Donations List:** Admin can view list of donations.
- **View Users List:** Admin can view user list.

3.3.2 Front Screen or Starting Screen:

This is front screen or starting page of our app.

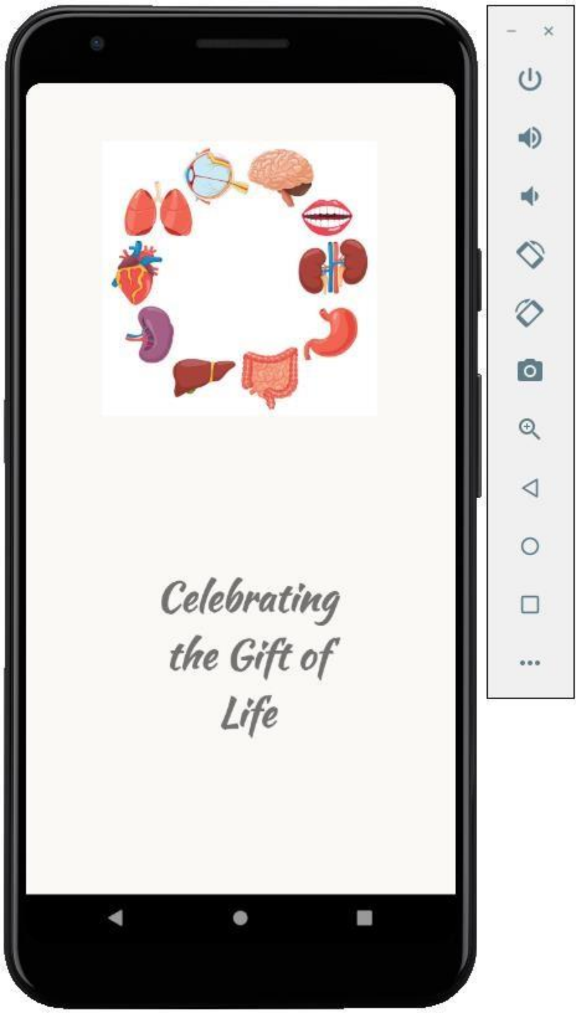


Fig 3.1

- We have used top & bottom animations to make this UI

3.3.3 User Login:

This is a java-based login page. The screenshot of User Login of our app is given below:

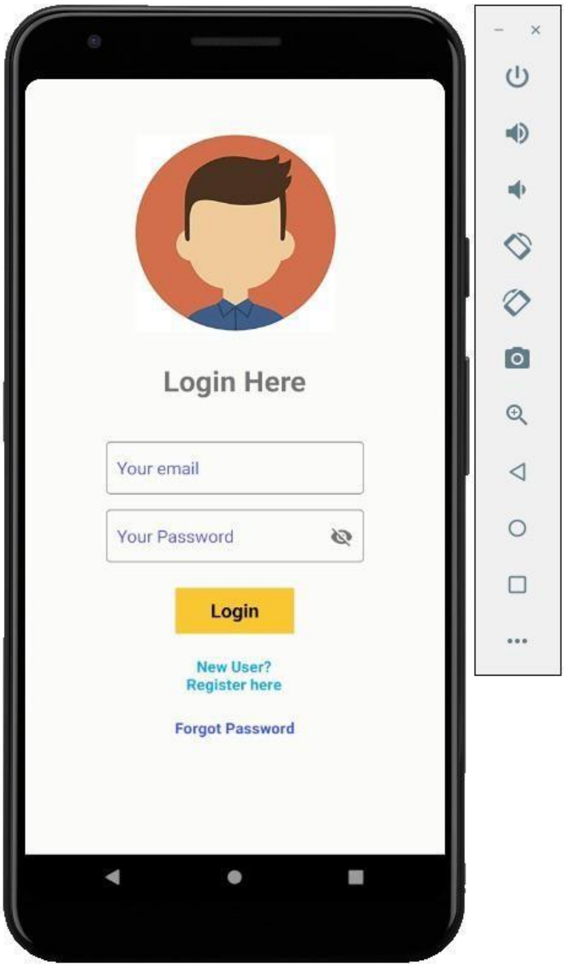
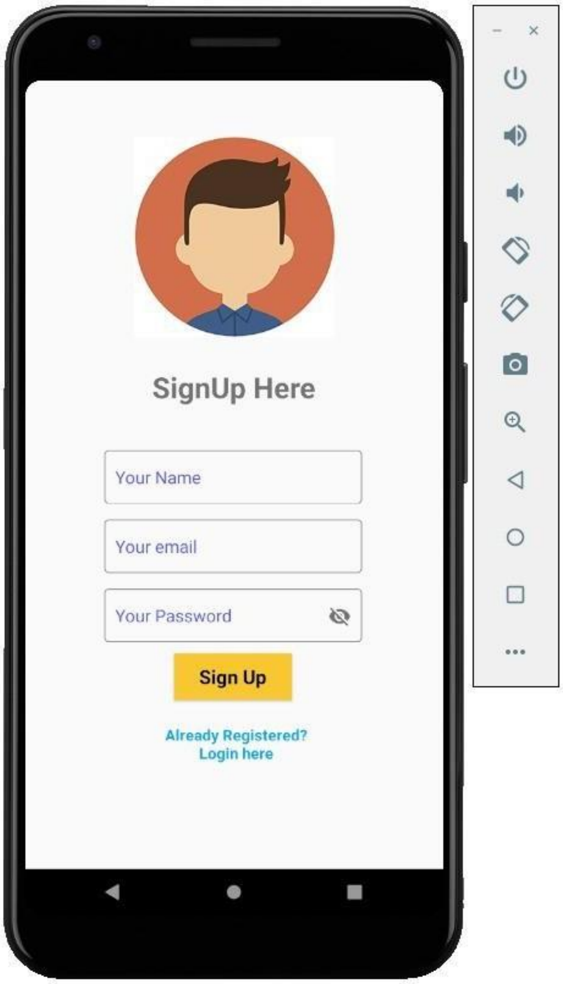


Fig 3.2

- We have used firebase authentication to enable Login functionality
- We have used PlainText, TextView with onClick enabled & Button to make this UI

3.3.4 User Sign-Up:

This is a java-based sign-up page. In this, we code in java for user Sign-Up. The screenshot of User Sign-Up of our app is given below:



SignUp Here

Your Name

Your email

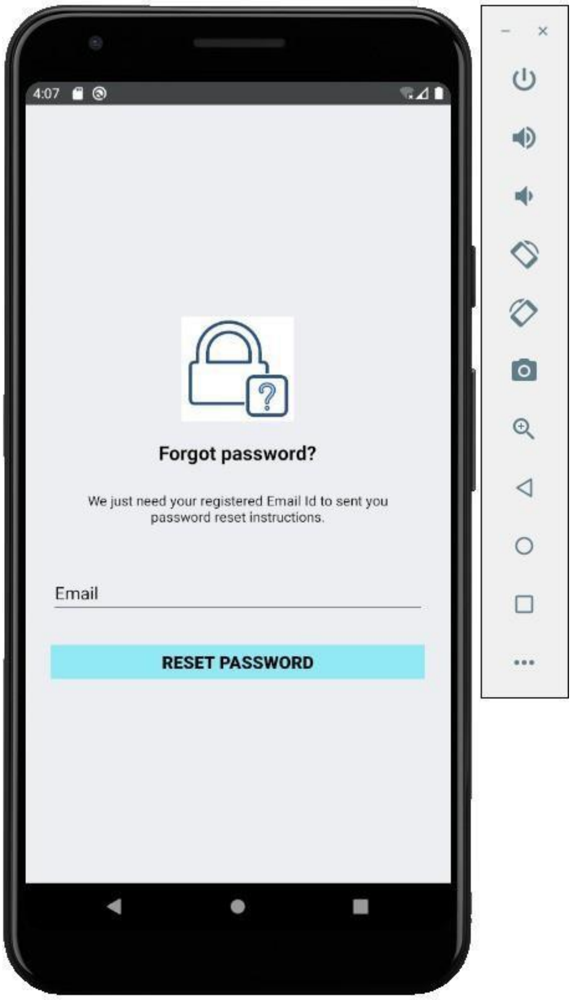
Your Password

Fig 3.3

- We have used firebase authentication to enable SingUp functionality
- We have used firebase Realtime database to store name of the user
- We have used PlainText, TextView with onClick enabled & Button to make this UI

3.3.5 Forgot Password:

This is a java based forgot password page. In this we code in java for user so that if he/she is forgot password then what he should have to do. The Screenshot is given below:



Forgot password?

We just need your registered Email Id to sent you password reset instructions.

Email

Fig 3.4

- We have used firebase authentication to enable forgot password functionality
- We have used ImageView, Plain Text & Button to make this UI

3.3.6 Home Screen:

This is home screen of our app. This page comes after login screen. The Screenshot is given below:

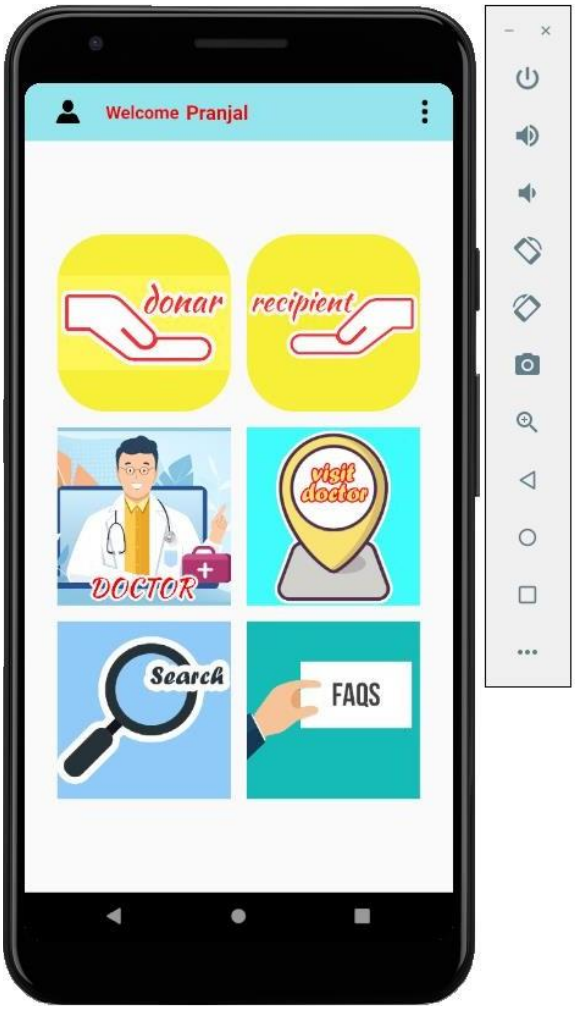


Fig 3.5

- The name of the current user will be dynamically displayed at run time
- We have used ImageButton to make this UI

3.3.7 Search Donor:

This page is under ‘Search Donor’ section of home page. The screenshot of this is given below:

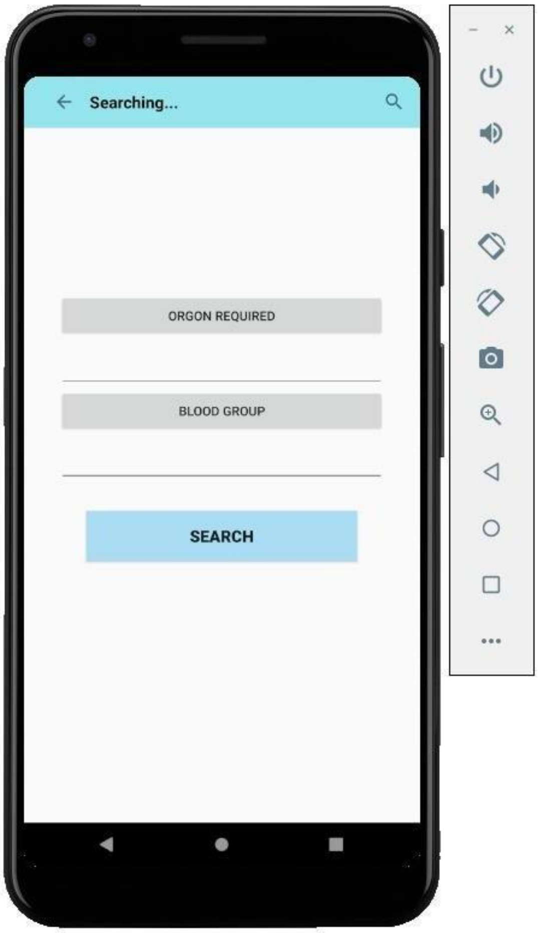


Fig 3.6

- We have used firebase Realtime database to search donor with the help of organ name keyword
- We have used PlainText & Buttons to make this UI

3.3.8 Registration of donor:

This page is under ‘Are You A Donor’ section of home page. The screenshot of this is given below:

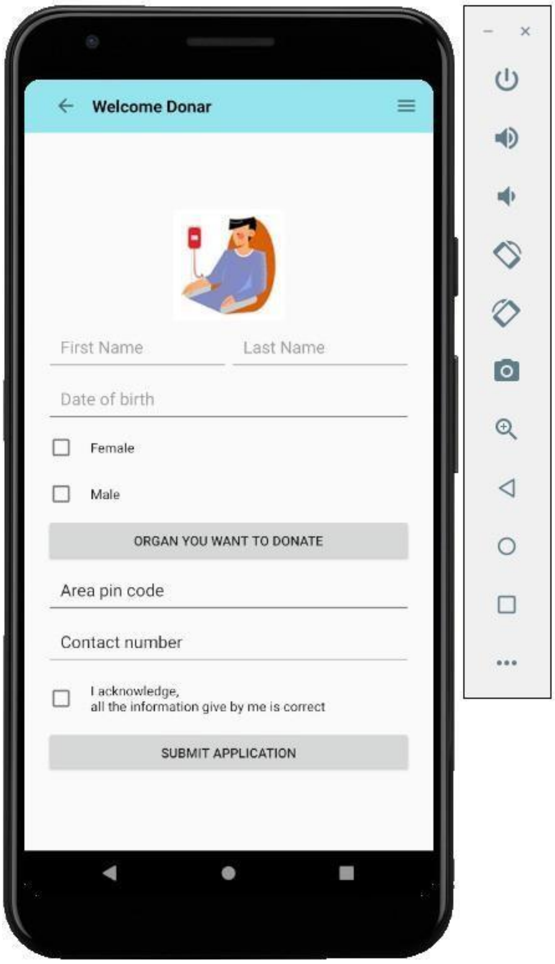


Fig 3.7

- The details of the Donor will be stored in firebase Realtime database
- We have used PlainText with hints, Buttons & CheckBox to make this UI
- We have used glide dependency to handle image of the user

3.3.9 Registration of Recipient:

This page is under ‘Are You A Recipient’ section of home page. The screenshot of this is given below:

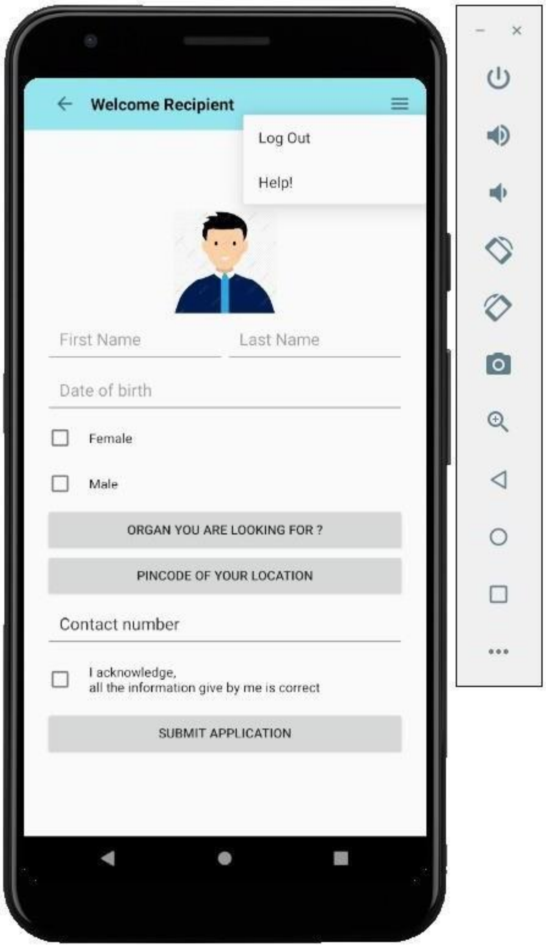
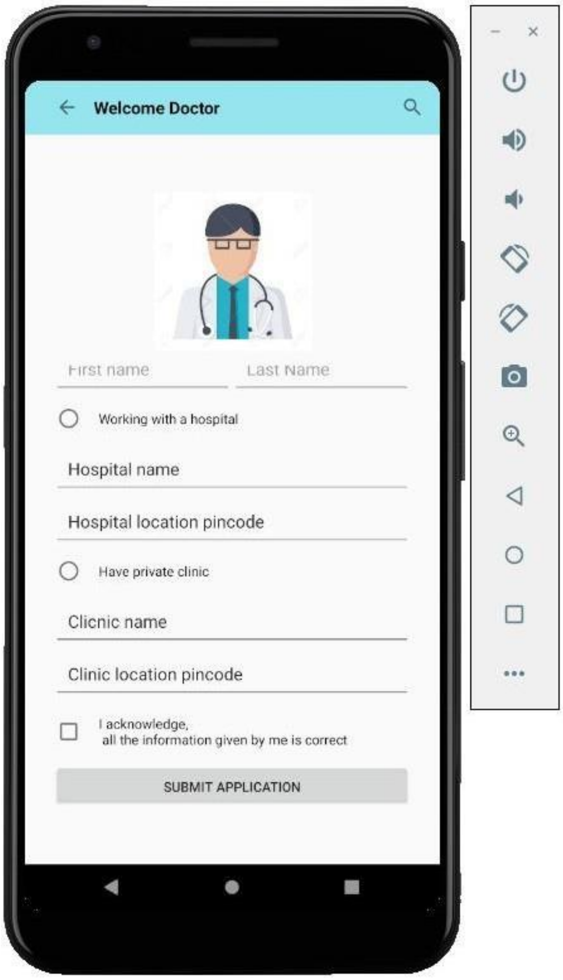


Fig 3.8

- The details of the Recipient will be stored in firebase Realtime database
- We have used PlainText with hints, Buttons & CheckBox to make this UI
- We have used glide dependency to handle image of the user

3.3.10 Registration for Doctor:

This page is under ‘Are You A Doctor’ section of home page. The screenshot of this is given below:



Welcome Doctor

First name | Last Name

○ Working with a hospital

Hospital name

Hospital location pincode

○ Have private clinic

Clicnic name

Clinic location pincode

□ I acknowledge,
all the information given by me is correct

SUBMIT APPLICATION

Fig 3.9

- The details of the Doctor will be stored in firebase Realtime database
- We have used PlainText with hints, Buttons & CheckBox to make this UI
- We have used glide dependency to handle image of the user

3.4 Back-End:

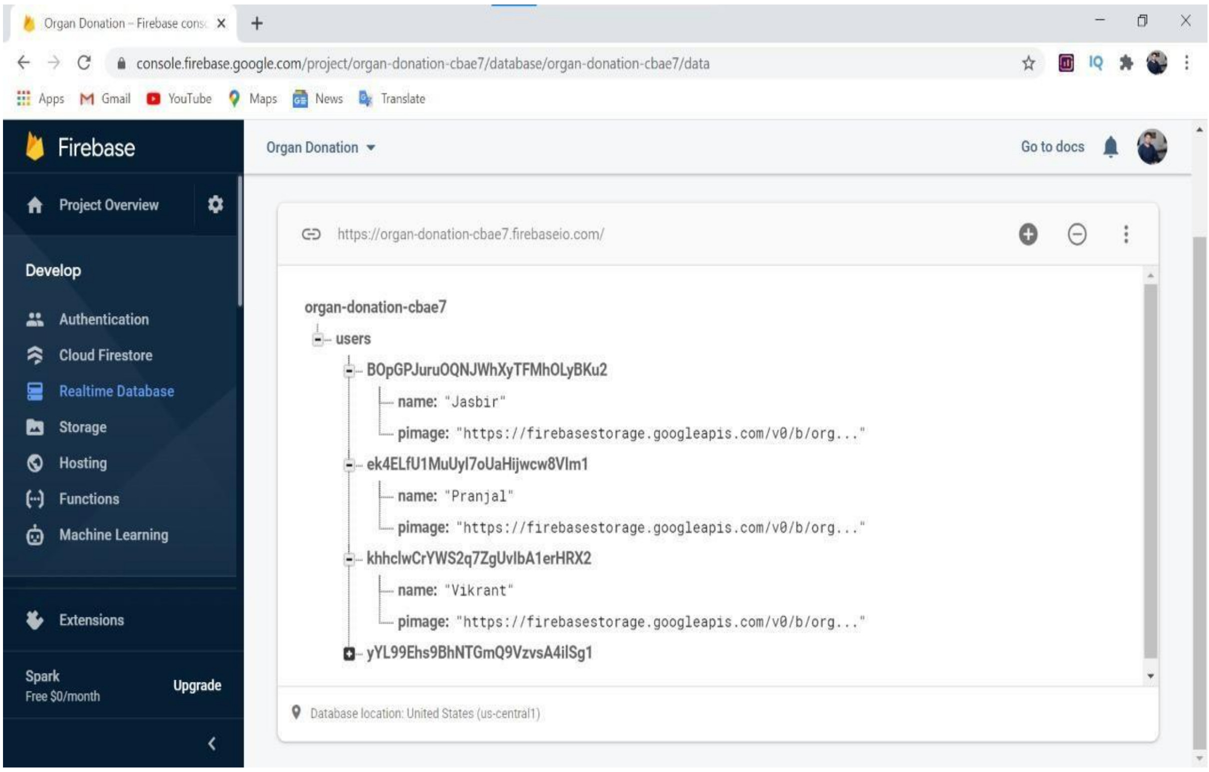
We are using Google’s mobile application development platform **Firebase** for handling backend requirement of our mobile application. It provides a variety of tools and services to develop mobile applications. It seems us a very user-friendly way to store & retrieve data of the application as it does not require any SQL knowledge to store and access data in Firebase. It is super simple to connect our application to the firebase console and google has also provided tons on firebase documentation to help us. Some of the main services that firebase provides under a same platform are:

- **Authentication** which helps us to maintain Login & Registration of users in our app
- **Realtime Database** is a cloud-hosted NoSQL database that lets you store and sync data between users in Real-time even we don’t have to refresh the firebase console to see the changes in Real-time database.
- **Storage** allow us to store big size data file such as images, text file etc. & provides a link of the file that we can store in Real-time database to access it from our application.
- **Cloud Messaging** allows admin or developer to send In-app notification to the app users. A developer can send notification to all users at a time or can send notification to some specific users.

These are some main services of firebase that we have used in our application. There are many more services provided by firebase such as In- App Messaging which we try to incorporate in our application later. We have used Real-time database to store information about organ donors, organ recipients and doctors.

Data in Real-time database is stored in a node tree like structure every node have two components name & value, name is used to access the value stored in the node & to access a node we have to go all through the root node to that particular node. The Real-time database can be visualized as a one big JSON object that the developers can manage in real-time. We have used firebase because Real-time syncing makes it easy for our application users to access their data from any device conveniently.

Another amazing feature of firebase that incline us toward itself is that when users go offline, the Realtime database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. Firebase not only used for development & testing of the application but can also be used for engaging audience for the application which is a must need of our application to aware and encourage our audience toward noble cause of organ donation. We have given below screenshots of firebase which we are using in our project.



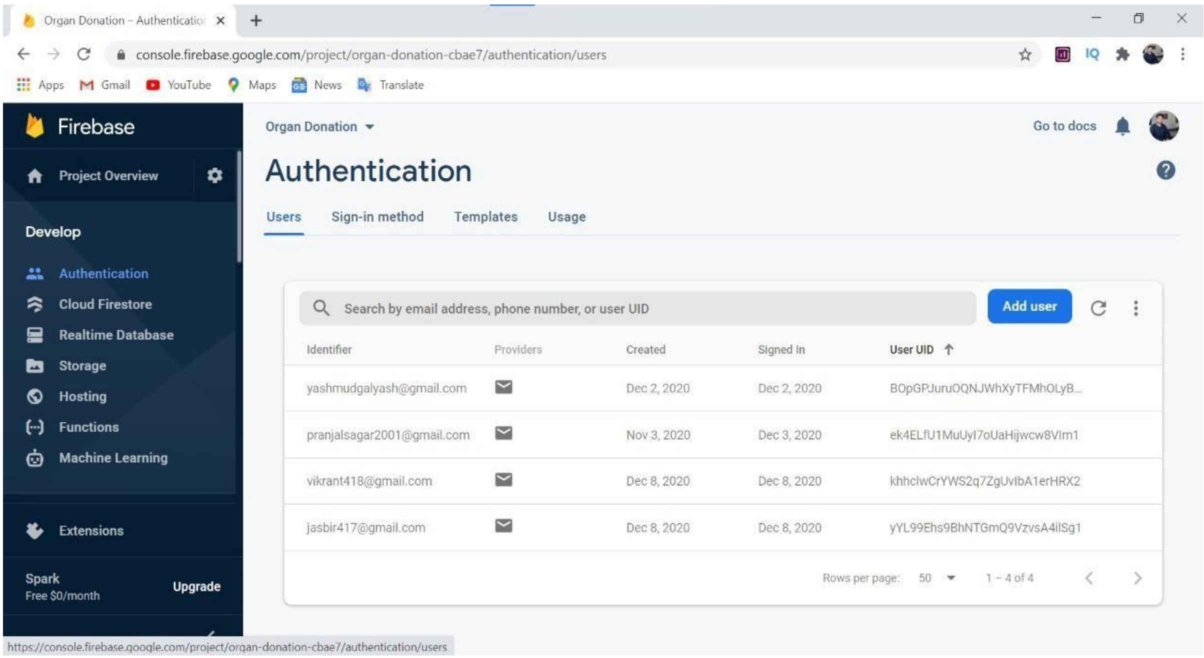
https://organ-donation-cbae7.firebaseio.com/

```
organ-donation-cbae7
  users
    BOpGPJuruOQNJWhXyTFMhOLyBKu2
      name: "Jasbir"
      pimage: "https://firebasestorage.googleapis.com/v0/b/org..."
    ek4ELfU1MuUyl7oUaHijwcw8Vlm1
      name: "Pranjal"
      pimage: "https://firebasestorage.googleapis.com/v0/b/org..."
    khhclwCrYWS2q7ZgUvlbA1erHRX2
      name: "Vikrant"
      pimage: "https://firebasestorage.googleapis.com/v0/b/org..."
    yYL99Ehs9BhNTGmQ9VzvsA4ilSg1
```

Database location: United States (us-central1)

This is the screen shot of firebase Realtime database showing:

- how a new user’s credentials are stored using unique UserId
- user name & image URL from firebase storage is also stored under user’s unique UserId



Identifier	Providers	Created	Signed In	User UID ↑
yashmudgalyash@gmail.com	✉	Dec 2, 2020	Dec 2, 2020	BOpGPJuruOQNJWhXyTFMhOLyB...
pranjalsagar2001@gmail.com	✉	Nov 3, 2020	Dec 3, 2020	ek4ELfU1MuUyI7oUaHijwcw8VIm1
vikrant418@gmail.com	✉	Dec 8, 2020	Dec 8, 2020	khhclwCrYWS2q7ZgUvIbA1erHRX2
jasbir417@gmail.com	✉	Dec 8, 2020	Dec 8, 2020	yYL99Ehs9BhNTGmQ9VzvsA4ilSg1

Fig 3.10

- Screen shot of current user registered with the application having unique UserId (i.e. UID)

CHAPTER 4

TESTING & VERIFICATION

4.1 Introduction:

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.

The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing as a White Box and Black Box Testing. In simple terms, Software Testing means the Verification of Application Under Test (AUT).

It is vital in a software that it behaves as expected. To ensure that the system and its components will behave as required a number of tests was taken.

Every unit and aspect of the system was tested to make sure that every functionality in working and actual result of each operation is same as we expected. Our integration of all the units of the code the system was checked its operation with other system units.

Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

Testing is vital for any system; no system design is perfect. Testing is also carried out in two phases, first phase is during the software engineering that is during the creation of models, second phase of testing takes place after the system is deployed or when software is completed.

Testing is important because bugs can be expensive or even dangerous for the system. Even a simple can lead to major problems.

4.2 Types of testing:

There are many types of system testing some of them are:

Black box testing:

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

White Box Testing:

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

Unit Testing:

Each module is considered independently; it focuses on each unit of software as implemented in the source code.

Integration Testing:

It aims at constructing the program structure while at the same time constructing tests to uncover errors associated with interfacing the modules, modules are integrated using top-down approach.

5.3 System Implementation:

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project- Programming and Testing- This activity encompasses actual development, writing, and testing of program units or modules.

User Training- This activity encompasses writing user manuals for understanding the system, preparation of user training materials, conducting training programs, and testing procedures.

Training of application software- After providing the necessary basic training on the computer awareness, the users will have to be trained upon the new system such as the screen flows and screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the way to correct the data entered. It should then cover information needed by the specific user or group to use the system.

The steps involved in software testing are:

1. Preparation of test cases.
2. Preparation of possible test data.
3. Complete code review of the module.
4. Manual Testing
5. Modifications done for the errors found in the testing phase.
6. Preparing the final system.

5.4 Summary:

In this chapter the testing and implementation of the software is told, that is how a system is tested for bugs, how the bugs are eradicated and how the final software is deployed.

In this Student portal testing of the software is done on the information made on preset available information of students. Testing is continuously done on many information and different use cases and scenarios.

5.5 Some Test Cases:

Table 2.1: Some test cases:

Test Case Id	Test Description	Expected Result	Actual Result	Result
TC001	Different Interface for Admin and User	Admin Screen Displayed	Admin Screen Displayed	PASS
TC002	User makes request of donation	Request made successful	Request made successful	PASS
TC003	User fills registration form of Donor	Successfully form filled	Successfully form filled	PASS
TC004	User Fills registration form of Recipient	Successfully form filled	Successfully form filled	PASS
TC005	User Fills Registration form of Doctor	Successfully form filled	Successfully form filled	PASS
TC006	User Books Appointment with Doctor	Appointment Booked	Successfully form filled	PASS
TC007	User Search for Donor	Shows list of donors	Shows list of donors	PASS
TC008	User Search for Recipient	Shows list of recipients	Shows list of recipients	PASS

Table 2.2 Test case for Signup module

Test Case ID	Test case name	Test case description	Test steps			Test status (P/F)	Test Priority
			Steps	Expected result	Actual Result		
User	Validate First Name and Last Name	To verify that First Name and Last Name must be of minimum 2 characters	Enter First Name and Last Name less than 2 characters	An Error Message "First Name and Last Name cannot be less than 2 characters"	Error Message & icon Displayed	P	Medium
				& error icon must be displayed.			
			Enter First Name and LastName of 3 characters	o Error Icon	No Error Message Displayed	P	Medium

Pwd	Validate Password	To verify that password should be between5to 10 characters	Enter password greater than 10 characters	An Error Message “password cannot be greater than 10 characters” & error icon must be displayed.	Error Message & icon Displayed	P	High
			Enter password of 10 characters	No Error Icon displayed (Valid password)	No Error Icon Displayed	P	High
			Enter password less than 5 characters	An Error Message “password cannot be less than 5 characters” & error icon Must be displayed.	Error Message & icon displayed	P	High

Table 2.3: Test case for Login module

Test ID	Test Cases & User Action	Test Data	Expected Result	Actual Result	Comment
TC001	Blank Username & Valid Password Clicks On ‘Login’ Button	Username =”” & Password=“E-patient”	Display message “Username cannotbekept blank”	Valid message displayed	Working properly
TC002	Valid Username & Blank Password Clicks On ‘Login’ Button	Username =” Admin ” & Password =””	Display message “Password cannotbekept blank”	Valid message displayed	Working properly
TC003	Blank Username &Blank Password Clicks On ‘Login’ Button	Username =”” & Password =””	Display message “Bothfieldsare mandatory”	Valid message displayed	Working properly

TC004	Invalid Username & Valid Password Clicks On ‘Login’ Button	Username= “GHGGF” & Password =	Display message “Invalid Username”	Valid message displayed	Working properly
TC005	Valid Username & Invalid Password Clicks On ‘Login’ Button	Username =” Admin” & Password=”E-patient”	Display message “Invalid Password”	Valid message displayed	Working properly
TC006	Invalid Username & Invalid Password Clicks On ‘Login’ Button	Username =” GHGGF” & Password =” passwd”	Display message “Bothfieldsare invalid”	Valid message displayed	Working properly
TC007	Valid Username & Valid Password Clicks On ‘Login’ Button	Username =” Admin” & Password = ” E-patient”	Go to the next page having users created project and details.	Proper form displayed	Working properly

CHAPTER 5

FUTURE SCOPE

Some old age man is always trying to gain more and more wealth. Man is always to develop more and more new modified techniques with increasing the aesthetic look and economic consideration. Hence there is always more and more scopes in every project. But being the engineer and having the ability to think, plan and execute something. But due to time constraints, we only have thought some new ideas which we are going to execute in our project in near time.

This application can be easily implemented under various situations. We can add new features as well when we require. Modifications is possible as and when require in this application. There is flexibility in all the modules. So, we put in the report the following future modifications: -

1. We will be going to develop or implement chatting part in our android application. In this, doctors, donors and recipients can chat with each other easily and change their thoughts or queries on that.

2. We will be going to develop organ donation service with accurate matching of organs with receivers using Machine Learning. This would help recipients make a better decision about whether to accept the organ that is offered by donor by matching organ which is best suitable for them. To accurately match potential organ by the use of machine learning and some algorithms. We will be going to develop a tool that is able to match organs to transplantation recipients.

3. We will also provide same functionalities for blood also. In near time, we will be going to implement blood donation system also in our app.

4. We will also be going to add new feature in which donor can see recipient's location on a google map and also recipient can see location of donor on google map which will make easy for donor and recipient to safe from frauds.

- Thus, in future there are so many modifications, which we can make to survive the huge global world of competition.

CHAPTER 6

CONCLUSION

In this project,” Organ Donation Android Application” we have tried to bridge the gap between donors and recipients by providing donor, recipient and doctor a single platform. Organ donation android application is an app created to eliminate the communication gap between organ donor and organ seeker. It will bring organ donor, organ seeker, doctor and hospital at a same virtual place where they can interact freely and help each other. The app is created to make organ donation process efficient and transparent. It will help to control and minimize the human organ’s trafficking to some extent. As it directly connects the organ donor and organ seeker without any intervein of a middleman. The app also provides convenient way for doctors and hospitals to get organs at the time of emergency.

Organ Donation Android Application is very flexible android-based app. In this app, donor can easily search for recipients easily in critical situations and can contact to them directly through our app. Organ recipient can also submit organ requirement request through our app. We have also giving the facility of book appointment with doctor direct through our app.

This project gives me more than enough opportunities for an android-based project to design, code, test and execute. This has helped to implement the different software engineering principles. This project also helped me to learn about new platforms like firebase, GitHub, Android Studio, etc. This has also helped me to learn about report writings (paper work) for project.

I am grateful to my guide for his valuable contributions in getting the project forward. I also thank my friends who supported and encouraged me to successfully complete the project.

REFERENCES

1. Marc A. Shampo and Robert A. Kyle – "Joseph E. Murray-Noble Prize Of Organ Transplantation" , Stamp Vigenette of Medical Science - Volume 26 , Issue 3, P240, March 01,2001. https://www.mayoclinicproceedings.org/article/S0025-6196(11)62883-8/fulltext#:~:text=Murray%E2%80%94Nobel%20Prize%20for%20Organ%20Transplantation,-Marc%20A.&text=American%20physician%20Joseph%20Edward%20Murray,1954%2C%20in%20Boston%2C%20Mass

2. NOTTO - "About Us", Contents owned and updated by Ministry of Health & Family Welfare Government of India copyright 2013 NIC. https://www.notto.gov.in/about-us.htm

3. DGHS – "Background", Contents owned and updated by Ministry of Health & Family Welfare Government of India, Last Updated on 16/05/2016.

   https://dghs.gov.in/content/1353_3_NationalOrganTransplantProgramme.aspx?format=Print

4. ACAMEDIA – "Object Oriented Concepts Contents", Unit -1 Lesson - 1 Downloaded on 2 October,2020 from https://www.academia.edu/12326400/UNIT_I_LESSON_1_OBJECT_ORIENTED_CONCEPTS_CONTENTS

5. Firebase- "Documentation", Contents owned and updated by Google , last updated on 10-11-2020 https://firebase.google.com/docs/storage/gcp-integration

6. Android Developers: https://developer.android.com/

7. Rob Percival and Nick Walter - “The Complete Oreo Android Developer Course – Build 23 Apps” ,Downloaded from Udemy on 20-08-2020, last Updated on 10-2020 https://www.udemy.com/course/the-complete-android-oreo-developer-course/learn/lecture/8732006#content

8. HRSA – “Matching Donors and Recipients” - managed by Health Resources & Souces Administration and U.S Department of Health and Human Services https://www.organdonor.gov/about/process/matching.html

9. Sam Judd – “Glide” https://github.com/bumptech/glide

10. Henning Dodenhof – “CircularImageView”, Jan 12 ,2014 https://github.com/hdodenhof/CircleImageView

11. Orhan Obut – “dialogplus” https://github.com/orhanobut/dialogplus

12. Icons8 – “Android” https://icons8.com/icons/set/android

13. Android Studio: Documentation- Guides: “Create Dynamic lists with Recycler View”- Last Updated on 11-11-2020 https://developer.android.com/guide/topics/ui/layout/recyclerview

14. Android Studio: Documentation- Guides: “Dialogs”- Last Updated on 11-11-2020 https://developer.android.com/guide/topics/ui/dialogs

15. Android Studio: Documentation- Guides: “Material Design for Android”- Last Updated on 11-11-2020 https://developer.android.com/guide/topics/ui/dialogs

16. Unsplash – “Editorial”

    https://unsplash.com/

APPENDIX

- **GitHub:**

  https://github.com/yashmudgal05/Organ-Donation-Android-Application

- **Back-End:**

  https://console.firebase.google.com/project/organ-donation-cbae7/overview