

GOVERNMENT POLYTECHNIC, PUNE-16

(An Autonomous Institute of Government of Maharashtra)

DEPARTMENT OF COMPUTER ENGINEERING

Government Polytechnic, Pune-16



PROJECT REPORT

ON

CLINIVEX – A Doctor's and Patient's Application

Submitted By

AshwiniTanpure ,TanviSutar , PoojaKaulge , ShilpaDhongdi

Under the guidance of

Prof. T.P Sharma

Submitted To



CONTENTS

1 ABSTRACT.....	2
2 INTRODUCTION.....	3
2.1 Main Purpose	
2.2 Aim of the Project	
2.3 Project Scope	
2.3.1 doctor certificate verification via utilization of the blockchain technology	
3 LITERATURE REVIEW.....	7
3.1 Existing System	
3.2 Determination of Problem	
3.3 Proposed System	
4 APPLICATION TECHNOLOGIES.....	14
4.1 Android Studio	
4.2 Firebase Database	
4.3 Google Map API	
5 SYSTEM ARCHITECTURE & SYSTEM DESIGN.....	16
5.1 Use Cases	
5.1.1 Patient Use Case	
5.1.2 Doctor's Use Case	
5.1.3 Blood Donor's Use Case	
5.1.4 Blood Seeker's Use Case	
5.1.5 Organ Donor's Use Case	
5.1.6 Organ Seeker's Use Case	
5.2 System Design	
5.2.1 Class Diagram	
5.2.2 ER Diagram	
6 SOFTWARE TESTING.....	27
6.1 System Testing	
7 CONCLUSION.....	29
8 REFERENCES.....	29

1 ABSTRACT

In today's world if someone wants to book a Doctor's Appointment we need to call in clinic or personally go to that place and book the appointment. This consumes precious time of the patient. Also if the doctor cancels his/her schedule, the patient does not come to know about it unless he/she goes to the clinic.

The objective of this project is to build a system that will ease the process of booking appointment of the doctor. The patient will book the appointment through his/her mobile phone. The doctor will come to know the number of patients he has to attend whole day. The system will save patient's as well as doctor's time. It will save the receptionist's paper work. The system will prove to be useful for doctor as he can check his appointments whenever and from wherever he wants from his mobile phone.

This application shows the Doctors the previous history of a particular patient. The main objective of this research is connecting doctor and patient very quickly and easily from any location without any involvement of any third party. Any doctor and patient can easily use this application by registering themselves.

Blood is an important aspect for all living things. It proves to be a lifesaving component in case of emergency requirement. In this users can view all the information provided. The main aim of developing this application is to reduce the time to a great extent that is spent in searching for the right donor and the availability of blood and organ required. As well as User can register himself for his/her organ donation after his/her death to the needed ones.

2 INTRODUCTION

Health care is one of the fastest growing industry all over the world. Before the last few years, medical appointments were usually taken on the phone calls or by visiting the hospitals in person. This process needed the involvement of individuals so, the ability to take appointment was restricted to the availability of schedulers, phone lines or the physical presence of a person.

With the growth of time, everybody demanded timeless and efficient medical care delivery because manual appointments (that requires the physical presence of both individuals) and long waiting lines have formed an irritating situation for the healthcare institutions. So, it created a need for such an integrated health care system that could deliver seamless care to both outpatients as well as inpatients. The emergence of online appointment system offered timeless and efficient access to health care services.

Booking appointment online has become a new trend in the past few years and is considered as one of the key processes in the healthcare industry. A well-designed appointment system supposed to improve patients' satisfaction by reducing cost and time of clinics and hospitals especially in the busy lives we are leading today. Patient/user can also provide additional medical history in advance, giving adequate time to the doctor to prepare the necessary information for consultation. In this way, online appointment scheduling systems are helping doctors and the patients and making the healthcare delivery efficient.

The need for blood and organ is great as it is life, as there is no replacement for human blood and organ. Every day blood and organ is required in hospitals and emergency treatment facilities for patients with Cancer, Thalassemia and other diseases, for organ transplant recipients, and to help save the lives of accident/trauma victims. With a growing population and advances in medical treatments and procedures requiring blood transfusions, the demand for blood and organ continue to increase. In India many people are losing their lives every day in emergency situations because we are suffering from lack of blood and organ in blood and Organ Banks, and they do not receive the blood and organ timely.

Their relatives and friends start searching for a donor to help, but there is no guarantee whether he will come 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 have become increasingly tried to activate the blood and organ donation process. However, this is still inefficient up to day. 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, available time, and same blood type, facilitate the search process for needy people and make it easier than before. Increase number of donors by increasing the facilities provided to them and to increase the awareness of the society about the importance of blood donation. Our system facilitates the donation process in our country.

1 Main Purpose

The main purpose of this project is to link and bring all major private medical clinics of the city (of our country) to a single platform. So that patients can easily get access to the doctor's profile and make online appointments.

The second purpose is to create an online medical history database so that doctors and patients can freely exchange patients' medical history information much easier, faster, and safer.

The third purpose is to simplify and automate the process of searching the blood in case of emergency and maintaining the records of blood donors, recipients, blood donation programs and blood stocks in the bank.

2.2 Aim of the Project

The aim of this project is to create a platform where patients and doctors can access /interact efficiently with each other and provide ease and comfort to the patients. It also aims to resolve the problems that patients have to face while taking appointments and keeping medical files. Patients can choose a medical practitioner based on their professional profile and other patient's reviews. While doctors can access and update a patient's medical record after every checkup.

Following features will be added in the future:

2.3 Project Scope

This system is implemented for all the individuals who want to get treated by the city practitioners. The users can participate only if they have created an account through the registration form and have provided their medical history. Once they get registered themselves further they would not need to update their record as it would be done automatically after each doctor's visit.

“DOCTOR CERTIFICATE VERIFICATION VIA UTILIZATION OF THE BLOCKCHAIN TECHNOLOGY”

The medical certificates issued by universities and other educational institutions are among the most important documents for graduates. A certificate is a proof of a graduate's qualification and can be used to apply for a job or other related matters. The advance of information technology and the availability of low-cost and high-quality office equipment in the market have enabled forgery of important documents such as certificates, identity cards, and passports. However, verification of certificates using traditional methods is costly and very time-consuming. The blockchain technology contains several functions including hash, public/private key cryptography, digital signatures, peer-to-peer networks and proof of work. The model uses various elements to formulate the block which is divided into two main processes, namely issuing a digitally signed academic certificate and verifying the academic certificate.

Institutions issue certificates to students who have completed graduation requirements. A graduation certificate is mostly in the form of a paper-based document, as an electronic document cannot effectively replace a physical certificate. However, due to the presence of advanced and cheap scanning and printing technologies, the forgery of certificates has

increased. This threatens the integrity of the certificate holder and the university that issued the certificate. Therefore, document validation and verification have become important tasks. It is necessary to validate that the medical certificate presented by the graduate is genuine and the holder is the rightful owner. Moreover, a medical certificate has to be verified to ensure that its content is correct and also to ensure that the certificate comes from an authentic source.

During the process of manual checking of the documents and certificates, much time is spent in either reaching out to the university to verify a certificate or in awaiting a reply from the university to confirm that the certificate is valid, and the information is accurate. This process can be extremely laborious and expensive .

THE PROBLEM OF FAKE CERTIFICATES

There are five different sources of fake academic certificates. These include:

1. ‘Degree Mills’ where bogus qualifications are generated and sold to clients.
2. ‘Fabricated Documents’ that represent a fictitious degree or institute.
3. ‘Modified Documents’ that are alterations in legitimate documents such as changes in enrollment or graduation dates, grades, course content, date of birth, specialization, etc.
4. ‘In-House Produced’ which are fake documents fabricated by the employees of legitimate institutions and printed on authentic paper and bearing the seals, stamps, and signatures of the institution.
5. ‘Translations’ or documents inaccurately translated to match requirements in a receiving country.

LIMITATIONS OF TRADITIONAL VERIFICATION SYSTEM

1. Ownership – Certificates are awarded to and are owned by individuals. However, the issuing authority still needs to reissue or authenticate certificates. The certificate ownership does not automatically appertain to the individuals.
2. Availability – Physical documents may be lost or damaged. Hence, individuals who lose them cannot readily obtain duplicates. Moreover, records cannot be retrieved if the issuing authority stops its operation.
3. Dependency on third-party agencies – Many organizations depend on third-party verification agencies to contact issuing authorities and verify document authenticity.
4. Time consumption – The process is time-consuming. The speed of verification depends on the response time of issuing authorities and their location.
5. Cost – Verification and notarization are costly. A fee is charged for each verified document.

The blockchain is a composite technology containing several functions including hash, public and private key cryptography, digital signatures, peer-to-peer networks, and proof of work. Each of these is explained as follows.

• Hash - A hash is a shortcode of fixed length. Data input from a document into hash-generator results in a hash output containing a certain number of digits. This hash then forms a unique ID. Inputting the same data into the hash generator results in the same hash value. However, even minor differences (such as changing a single letter of text) in data input results in a completely different hash . Figure 1 shows the mechanism of generating a hash.

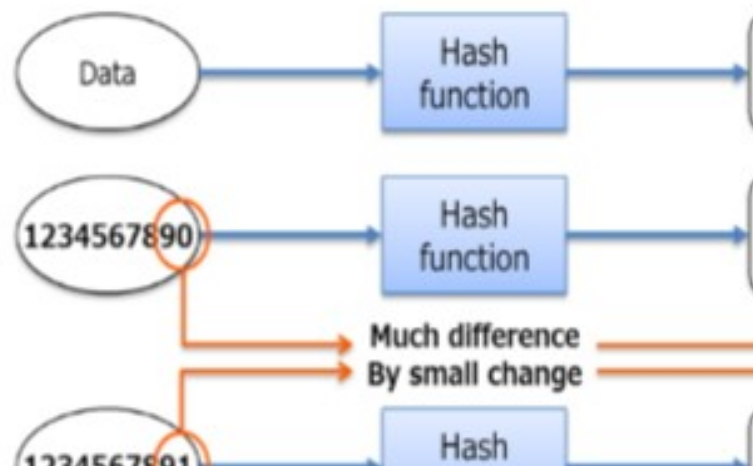


Figure 1

In Figure 1 it is observed that even a small change in the data input will result in a very different hash being generated. This characteristic of a hash may be used for the detection of any falsification of certificate that is been uploaded and is used in a blockchain mechanism for authentication purposes.

Potential Blockchain Uses in CLINIVEX

Now that we've defined blockchain and summarized its technical dimensions, let's review its potential applicability in five areas of health care that industry leaders indicate as most promising.

1. **Longitudinal health care records** – using blockchain to securely link across various health care provider organizations and, over time, a patient's health care records.
2. **Interoperability** – using blockchain to facilitate the gathering of massive amounts of patient data to aid population health initiatives.
3. **Online patient access** – using blockchain to allow patients to securely access their longitudinal health care records.

3 LITERATURE REVIEW

3.1 Existing System

- ❖ OployeeLabs, Developed an apps “Doctorola”, October- 2015, Updated September, 2016. In this project they designed an android application which is developed to find a doctor or a dentist easily and take an appointment with the doctor instantly.
- ❖ Pharma and Medical Concepts PvtLtd Developed "Book Doctor Appointment." System.They used “Appy Pie” to develop the app and also available in Google Play store which is purchasable .User can manage online appointment of listed doctor. There is an easy and quick search option for trusted Doctors across all cities based on their specialty, clinic or doctor’s name.
- ❖ In “The Optimization of Blood Donor Information and Management System by Technopedia” This System is a Web based Android Application. The System consist of two Devices - Android Smartphone and A server (Usually a pc). Donor/Acceptor creates an account and a unique user id and password is given to them. Acceptor can be either the patient or any relative to patient. Every detail is stored in Database (MySQL).

Reference Gap:

1. No Blood Banks are involved.
 2. Does Not include organ donation.
- ❖ In “A Survey Paper on E-Blood Bank and an Idea to use on Smartphone” This Blood Bank system is applicable for a Single Blood Bank. All details such as Blood Group, Total units of Blood available, donors details etc are stored in a Database.

Reference Gap:

1. Applicable for single Blood Bank only.
2. Only Doctor can check for availability of blood.
3. Time Consuming and Does not include organ donation.

3.2 Determination of Problem

- ❖ Starting a private clinic can be a challenge for new practitioners. For example, if a new doctor wants to open his clinic there is no platform exist, by which or from where people can get to know about his clinic.
- ❖ The problem for a new individual in a city In the same way, the same problem goes for any person new to the city as he is unknown, who is the best or closest doctor/clinic to go for a medical checkup.
- ❖ Long waiting lines To get doctor's consultation, patients come twice to the clinic once for taking the appointment and second time for the checkup. Some patients visit the

doctor without any prior appointment resulting in a long waiting time. There is also a possibility that they don't consultation by the doctor even after a long waiting line.

- ❖ Managing paper-based medical files Patients must carry their medical related report or files every time they visit a doctor. And sometimes they lost their files too.
- ❖ The problem of emergency requirement of blood or organ in the hospital during surgery.

3.3 Proposed System

The proposed system consists of two panels: Doctor and Patient. The users will first have to download the application and install it in their mobile devices. Once installed, this application will remain into the device permanently until the user deletes it or uninstalls it.

1.Interface design for patient

The user will have to register into the application for the first time. The user can login using phone number and password for logging into the app each time he uses it. After logging in ,he/she can either choose to book appointment ,request/register for blood donation or request/register himself/herself for organ donation.

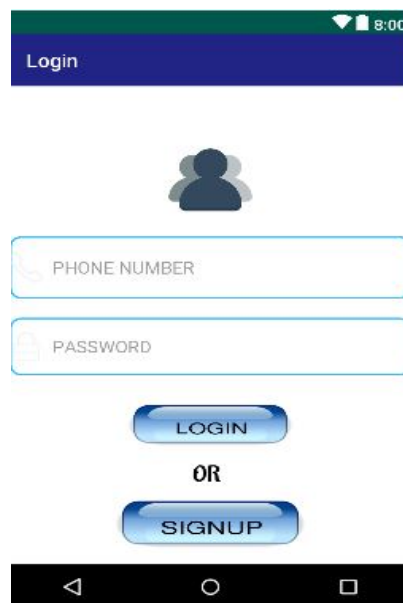
If the user selects book appointment module , the patient will have to select a filtration type. The filtration is done on two bases: Area wise and Specialty wise. After selecting the filtration type, the doctors list will be displayed. The patient can select any particular doctor and view his profile. Also the patient can view the doctor's schedule and look for an appointment according to his convenience. The patient will then send a request for appointment. The doctor can either accept the appointment or reject it.



The purpose of the system is to simplify and automate the process of searching the blood in case of emergency and maintaining the records of blood donors, recipients, blood donation programs and blood stocks in the bank. When the user selects blood donation or organ donation module , seeker can search for blood donors and can call or message the donors through this app on android. This application can also be used by organ donor and seeker where person can register for organ donation.

Appointment Booking

The patient will have to register in the application on first use. After registration, the patient will receive a username and password. For sign up, the user has to fill the given fields that are username, email, password and confirm password and then the user clicks on the signup button to register itself and then all the information provided by the user is saved in the database located on the server. If the user registers successfully then a notification message “successfully registered” is displayed as shown below



1.Login Page

After logging in, the menu screen is displayed containing different options like Book Appointment, Blood Donation and Organ Donation as shown below

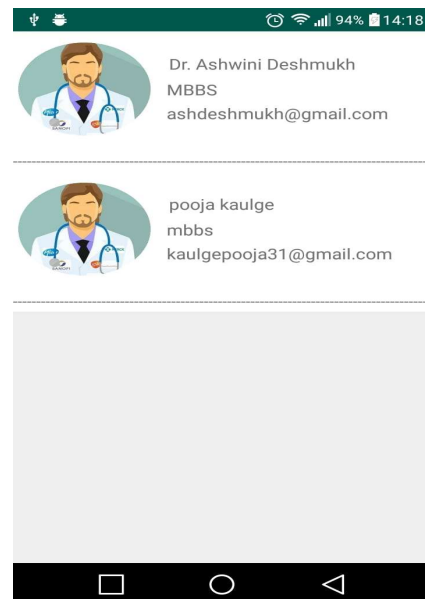


2.Home Page

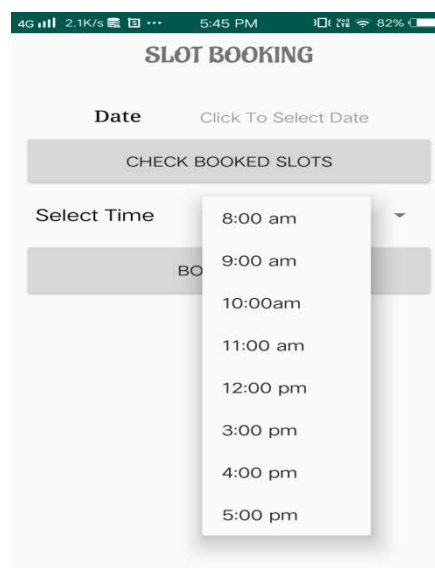
If the patient selects the Book Appointment option then he/she can view a list of Doctors in their specialized field. Then the patient selects the particular Doctor and then he can view the details of doctor and confirm appointment as shown below



3. Doctor's category Page field



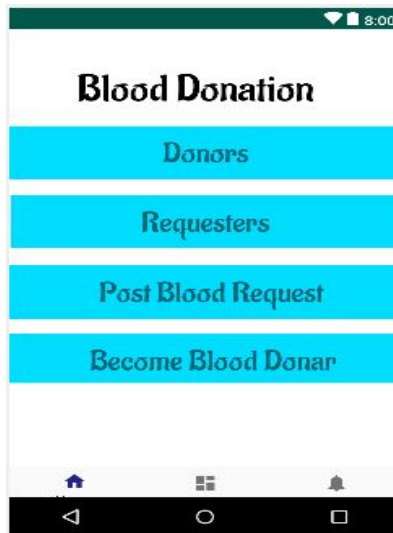
4. List of doctors in a specific field



5. Slot booking

Blood Donation

When the user selects the blood donation option of the home page ,the below depicted screen opens.

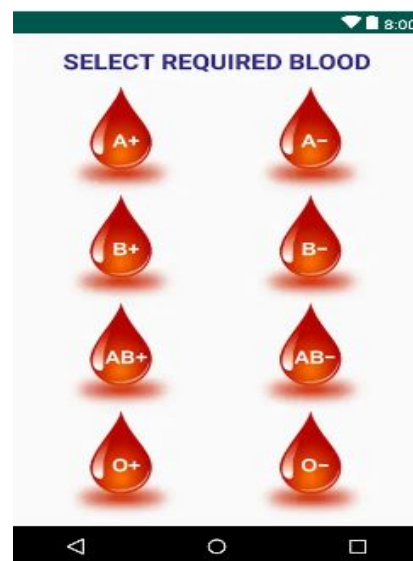


6. Blood Donation Page

7.

The donor or the requester willing to register for donating blood has to fill the registration form so that the seeker can also register himself/herself and see the list of donors and request for blood in emergency times.

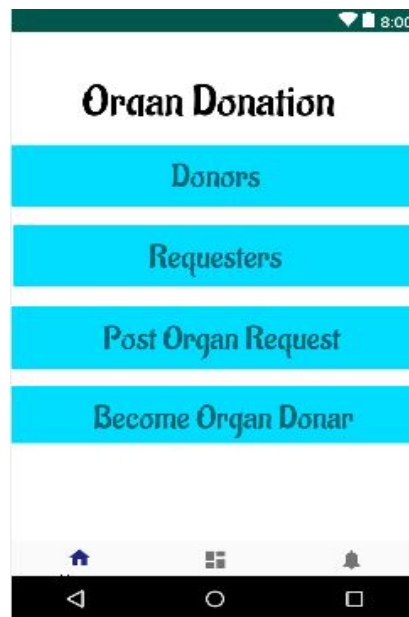
8.Donor's /Seeker's Registration Form



9.Blood Group Selection

Organ Donation

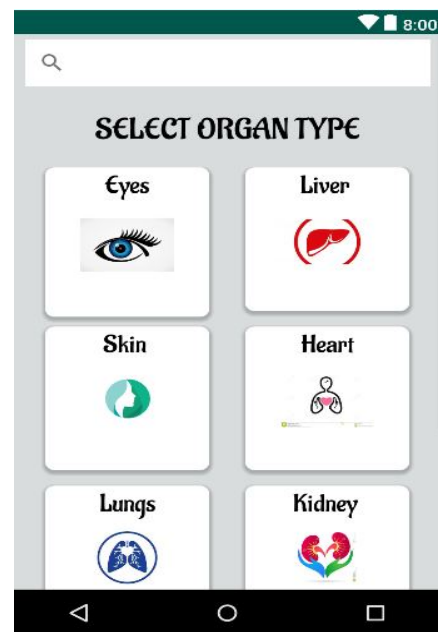
When the user selects the Organ donation option of the home page ,the below depicted screen opens.



10. Organ Donation Page

The donor or the requester willing to register for donating organ has to fill the registration form so that the seeker can also register himself/herself and see the list of donors and request for organs in emergency times.

11. Donor's /Seeker's Registration Form



12 .Blood Group Selection

2.Interface Design for doctor

Any new doctor who wants themselves to get registered to this application ,needs to fill the online registration form. After completing registration, at first the doctor has to login providing his/her user id and password, if the supplied information is correct he/she will be able to access the system.

If the doctor login on the system will get notification regarding patient appointment request as well as the doctor will be able to see total patients list in the app, who can meet him/her for their health checkup. After completing checkup to patient, the doctor can entry the observation details and can provide prescription for a the patient and completing doctor's task the doctor has to save the prescription into the database then the prescription easily can be viewed by patient from his/her apps.

4 APPLICATION TECHNOLOGIES

The purpose of this project is to build an Online appointment and healthcare System. It is important for the user to understand how this application works and knowing the technologies that are used to implement this project. For a better understanding, all steps are described in detail to give a full overview of the system.

4.1 Android Studio



Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Features

The following features are provided in the current stable version:

1. Gradle-based build support
2. Android-specific refactoring and quick fixes
3. ProGuard integration and app-signing capabilities
4. Template-based wizards to create common Android designs and components
5. A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
6. Support for building Android Wear apps
7. Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
8. Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation

mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

4.2 Firebase



Realtime database

Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the realtime database can secure their data by using the company's server-side-enforced security rules. Cloud Firestore which is Firebase's next generation of the Realtime Database was released for beta use.

Firebase Storage

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage.

4.3 Google Map API

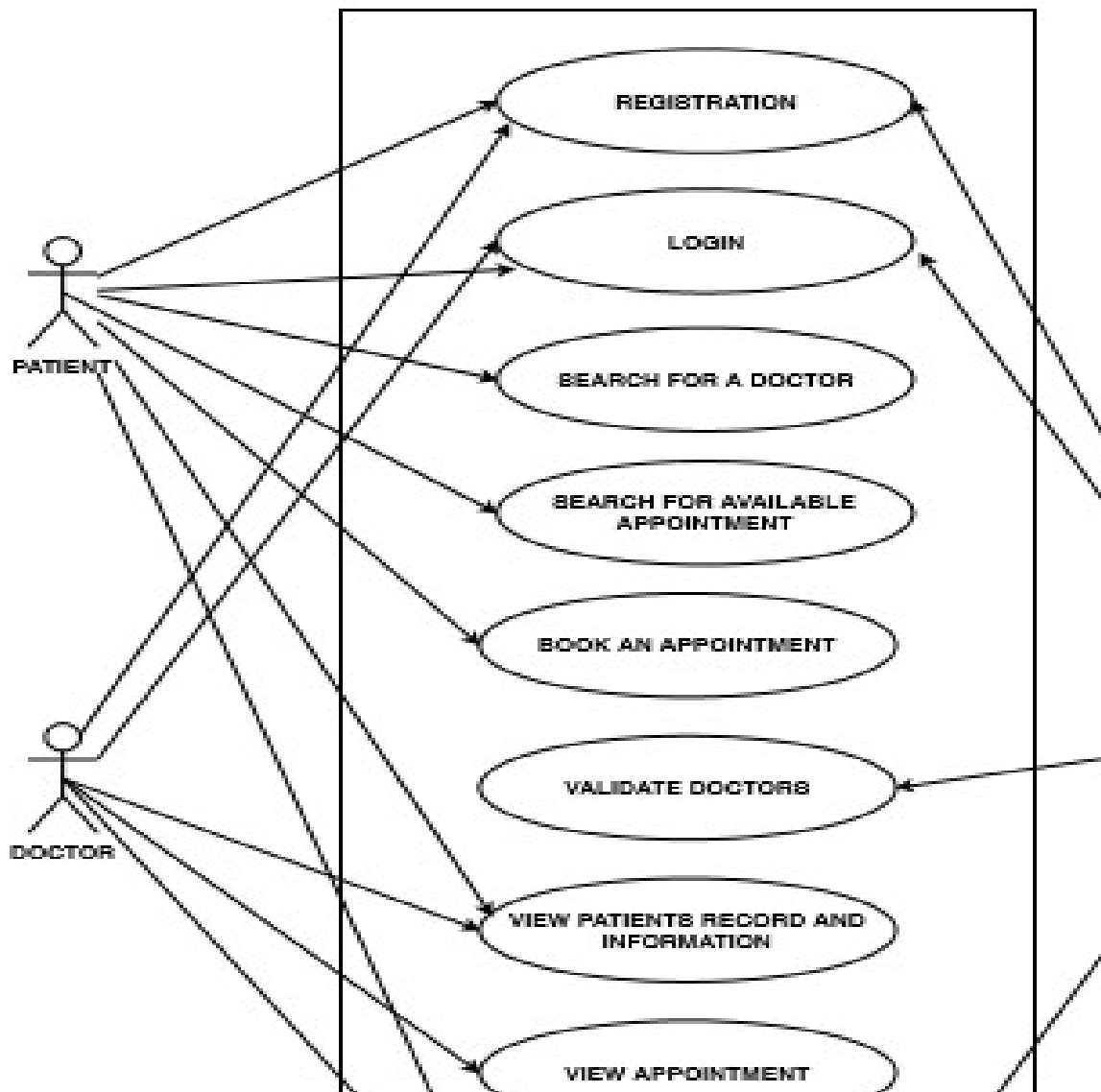


The Google Maps API is free for commercial use, provided that the site on which it is being used is publicly accessible and does not charge for access, and is not generating more than 25,000 map accesses a day. Sites that do not meet these requirements can purchase the Google Maps API for Business.

5 SYSTEM ARCHITECTURE

5.1 Use Cases

To explain the better view and functionalities of the system, use case diagrams are chosen. The use case diagram is important to document the requirement of the system as well as to specify functionalities of the system. Use case diagrams better explain the way the user interacts with the system.



5.1.1 Patient/user's Use Case

Use-Case 1: Registration

Primary-actor: generic user/patient.

Description: To make an appointment, user registration is required.

Precondition: valid email address and phone number.

Basic use-case flow: For the registration process, the user/patient needs to give some information by filling the form. Some of the required information includes the following: First name ,Last name ,Gender ,Email address and phone number etc.

Main scenario: The user/patient will go onto the patient sign-up button either from the main page or from the drop-down menu. After clicking the sign-up button, a registration form will appear, where the user must give his personal information i.e. name, gender, email address, any medical history etc. After giving the required information user will submit the form. If all the fields are filled including the valid email address and phone number, the user will be registered onto the system. In case of any missing entry or invalid format of an email , the error occurs onto the page.

Exception: Invalid email address, phone number.

Use-case 2: Login

Primary-actor: Patient/user.

Description: Before taking any appointment or get access to his medical record, the user must have to provide his username and password.

Precondition: the user must have a valid username and password.

Basic use-case flow: a valid username with a password must be entered by the user.

Main scenario: To be able to get into the system, the user needs to enter his username and password either from the main page or from the drop-down menu from the top of the page. After clicking onto the login button, authentication request will be forwarded to the system.

Exception: wrong/invalid entered username or password.

Use-case 3: search for a doctor

Primary-actor: User patient/generic user

Description: In this use-case, any registered or unregistered user, can look and search for a doctor of any desired specialty.

Precondition: this use-case has no exception.

Basic use-case flow: By going into the category of any specific specialty, the user can view the list of all registered doctors.

Main scenario: The user will go to the down menu at the top of the page. A list of all specialties will appear in a drop-down list. The user will hit on the desired specialty. After that, a page will open with all doctors of that specific specialty. Now the user has the choice to choose any doctor based on qualification, experience, location etc.

Use-case 4: Take Appointment

Primary-actor: User/patient

Description: After choosing a doctor user will go further to send an appointment request from the available timings.

Precondition: the user must be login

Basic use-case flow: The patient/user hits the button for taking an appointment for the doctor. A list of available timings will appear for the chosen date. The user will select the suitable time for him and send the request for approval.

Main scenario: The user will hit the button for "take appointment". List of available timings will appear for a chosen date. The user selects the suitable time. The user will hit the submit button to send the request for approval.

Use-case 5: View history

Primary-actor: User/patient

Description: the user can view his medical history.

Precondition: the user must be signed in.

Basic use-case flow: The user/patient will click on the name of the patient and it opens the patient's profile. By going into it the user can view the history by clicking on the button for the patient's old reports/history.

Main scenario: The user clicks on the name of the patient and then press the button "view history" to view old reports.

Use-case 6: Log out

Primary-actor: User/patient

Description: the user will log out from the system.

Precondition: the user must be logged in

Basic use-case flow: the user can sign out himself from the system.

Main scenario: the user clicks on the log out button. The system will bring the user to the main page for the generic user.

5.1.2 Doctor's Use Case

Use-case 1: Registration

Primary-actor: Generic user/doctor

Description: To get online appointment requests, the doctor must register himself as a user on the application/system.

Precondition: active email address and phone number.

Basic use-case flow: To register as a doctor, the user must fill the form. Required fields include the following information: Full name Email address National identity number Qualification Experience Valid PMDC certificate (Practicing license issued by the PMDC authority) Clinic address Clinic timings

Main scenario: The user/ doctor has to go onto the doctor's sign-up button either from the drop-down menu at the top or from the main page. After clicking the sign-up button, a registration form will appear, where doctor/user has to give his personal as well as professional information i.e. name, clinic address, qualification, working experience etc. The user must give valid PMDC certificate for the registration process. After giving the required information user/ doctor will submit the form. If all the fields are filled, request for registration will be sent to the admin. As soon as admin receives a request for a doctor's registration he will verify his documents and only then the user will be accepted /registered. In case of any missing entry or invalid format of email , an error occurs onto the page. PMDC certificate is a license or a proof that an individual is allowed by the authority to practice medicine. Any user fails to give valid PMDC certificate issued by PMDC authority will not be allowed to register as a doctor. PMDC certificate is important to avoid any scam.

Exception: False documents, expired or invalid PMDC certificate, missing fields, expired national ID card or inactive email.

Use-case 2: log in

Primary-actor: User/doctor

Description: For further functions, the user must have to provide his email address and password.

Precondition: the user must enter the username and password.

Basic use-case flow: user/doctor should provide his username and password to log in.

Main scenario: To be able to get into the system, the user needs to enter his username and password either from the main page or from the drop-down menu from the top of the page. After clicking onto the login button, an authentication request is forwarded to the system.

Exception: occurs if fails to provide username and password.

Use-case 3: View calendar

Primary-actor: User/doctor

Description: User would be able to view his calendar.

Precondition: The user must be signed in.

Basic use-case flow: After logging in, the user/patient selects the date from the calendar to filter out the appointment.

Main scenario: After signing in, the user/doctor can view the page of his calendar. The user will select the month and date to see the appointments of that period or date. The user can view all the patient's appointments of any date.

Use-case 4: Accept or reject a request

Primary-actor: User/doctor

Description: User/doctor can accept or reject any patient's request.

Precondition: User/doctor must be signed in.

Basic use-case flow: User/doctor selects any date and decides to accept or reject any appointment request.

Main scenario: After logging in, all the patient's appointment requests will be appearing on the calendar according to the dates. User /Doctor will select the date from the calendar. User /Doctor can view all the appointment requests. User /Doctor can select the option to accept or reject the appointment request from the drop-down menu.

Use-case 5: View patient

Primary-actor: User/doctor

Description: User/doctor would be able to view the patient's detailed information including his/ her medical history.

Precondition: User/doctor must be signed in

Basic use-case flow: After selecting the date, user/doctor selects the patient. User/doctor can view the patient's detail and medical history.

Main scenario: After selecting the date, list of patients of that particular date will appear. User /Doctor will select the patient. User (doctor) can view his all information including the medical history by clicking on the "view patient" button.

Use-case 6: log out

Primary-actor: User/doctor

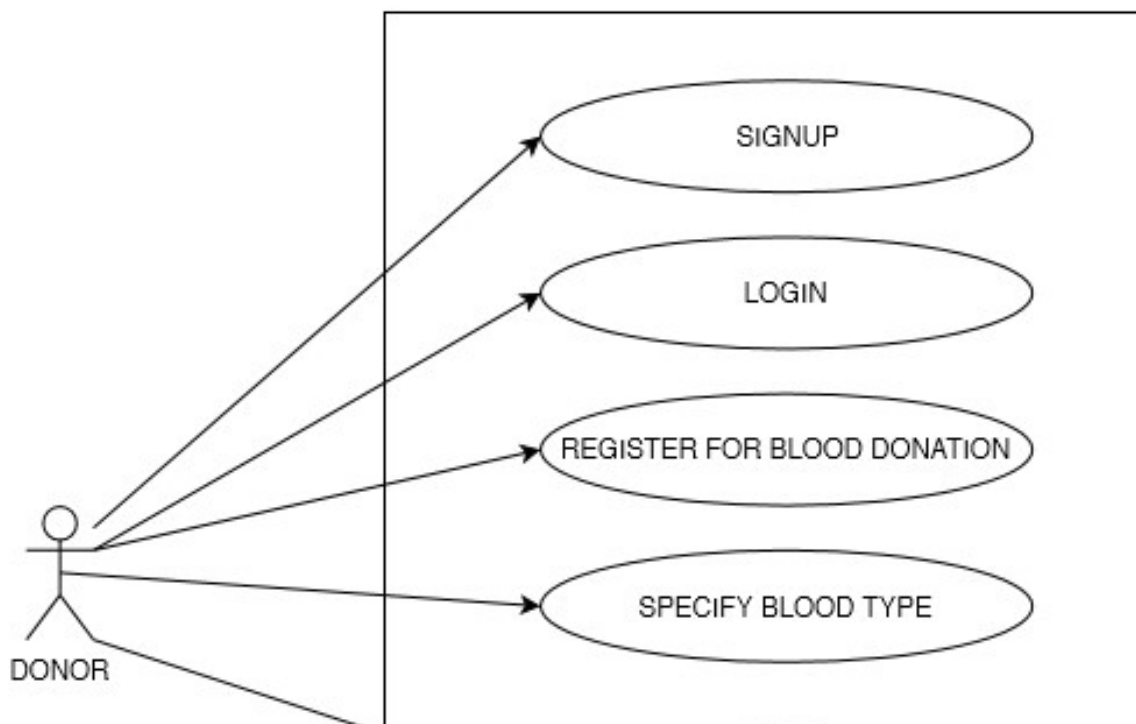
Description: the user/doctor will log out from the system.

Precondition: the user/doctor must be signed in

Basic use-case flow: the user/doctor can sign out himself from the system.

Main scenario: the user/doctor clicks on the log out button. The system will bring the user to the main page for the generic user.

5.1.3 Donor's Use Case



Use-Case 1: Sign Up

Primary-actor: generic user/donor.

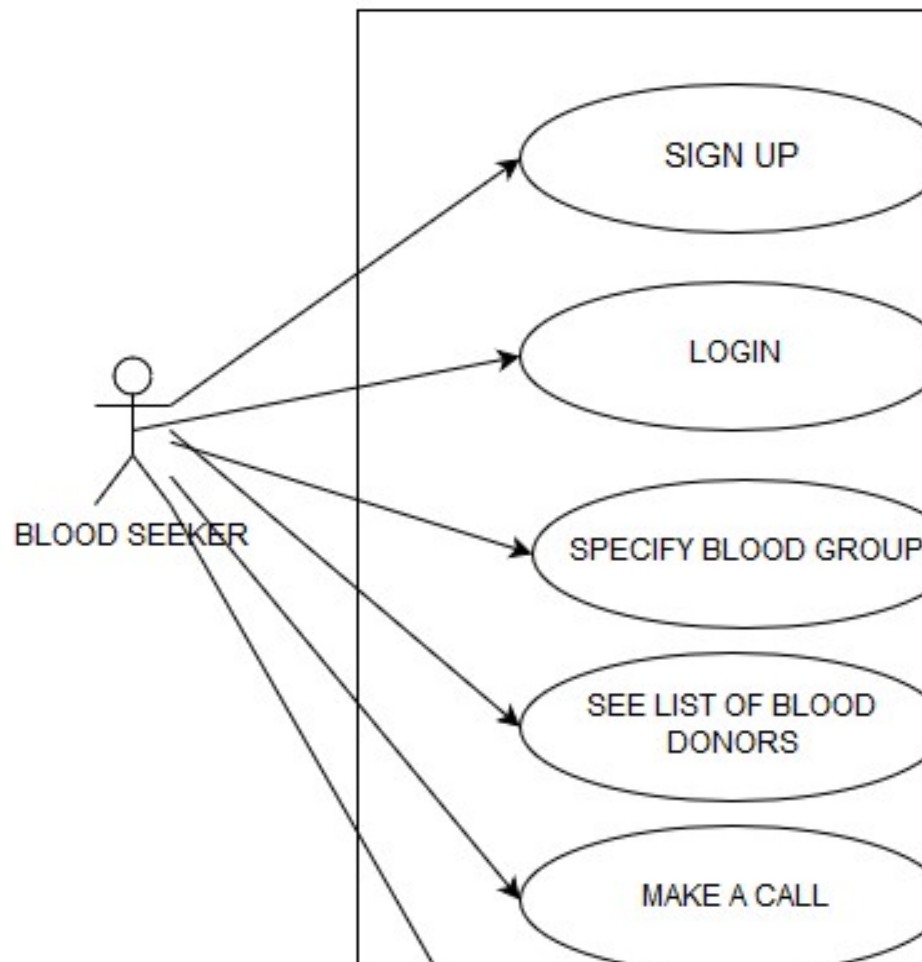
Description: To Register himself/herself for donation blood .

Precondition: valid email address and phone number.

Basic use-case flow: For the registration process, the user/patient needs to give some information by filling the form. Some of the required information includes the following: blood group ,Email address and phone number etc.

Main scenario: The user/patient will go onto the patient sign-up button either from the main page or from the drop-down menu. After clicking the sign-up button, a registration form will appear, where the user must give his personal information i.e. blood group, email address, phone number etc. After giving the required information user will submit the form. If all the fields are filled including the valid email address and phone number, the user will be registered onto the system and the user can register for donating blood .

5.1.4 Seeker's Use Case



Use-Case 1: Sign Up

Primary-actor: generic user/blood seeker.

Description: To Register himself/herself requesting for blood .

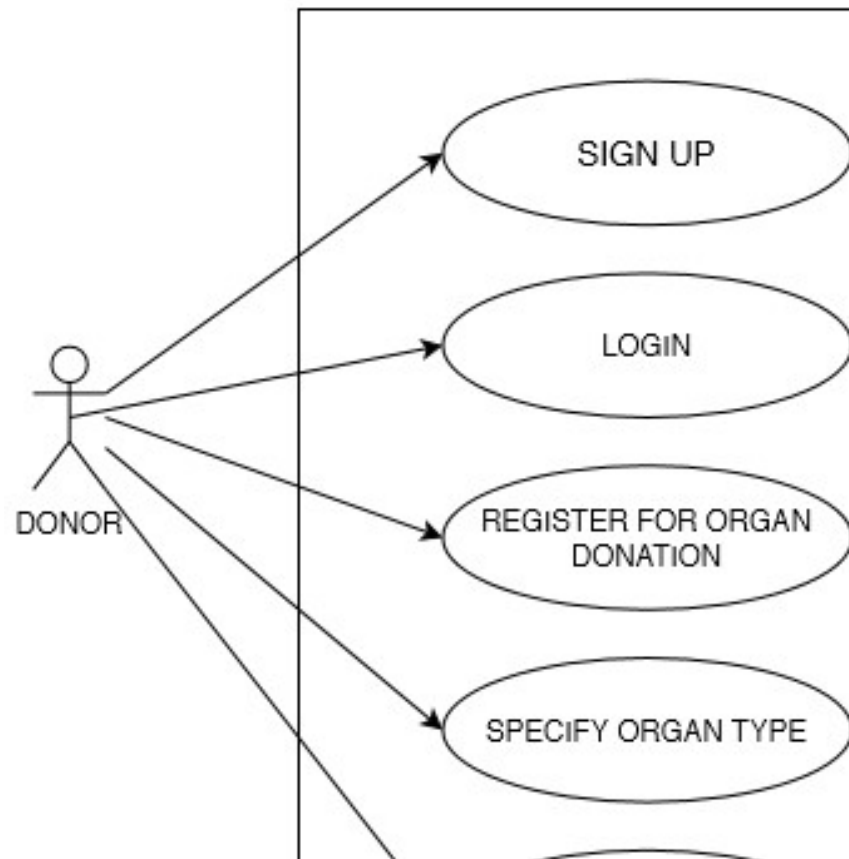
Precondition: valid email address and phone number.

Basic use-case flow: For the registration process, the user/patient needs to give some information by filling the form. Some of the required information includes the following: blood group ,Email address and phone number etc.

Main scenario: The user/seeker will go onto the sign-up button either from the main page or from the drop-down menu. After clicking the sign-up button, a registration form will appear, where the user must give his personal information i.e. blood group, email address, phone number etc. After giving the required information user will submit the form. If all the fields are filled including the valid email address and phone number, the user will be see the list of

blood donors and post a request or can call directly to the donors of matching blood group of his/her.

5.1.5 Organ Donor's Use Case



Use-Case 1: Sign Up

Primary-actor: generic user/donor.

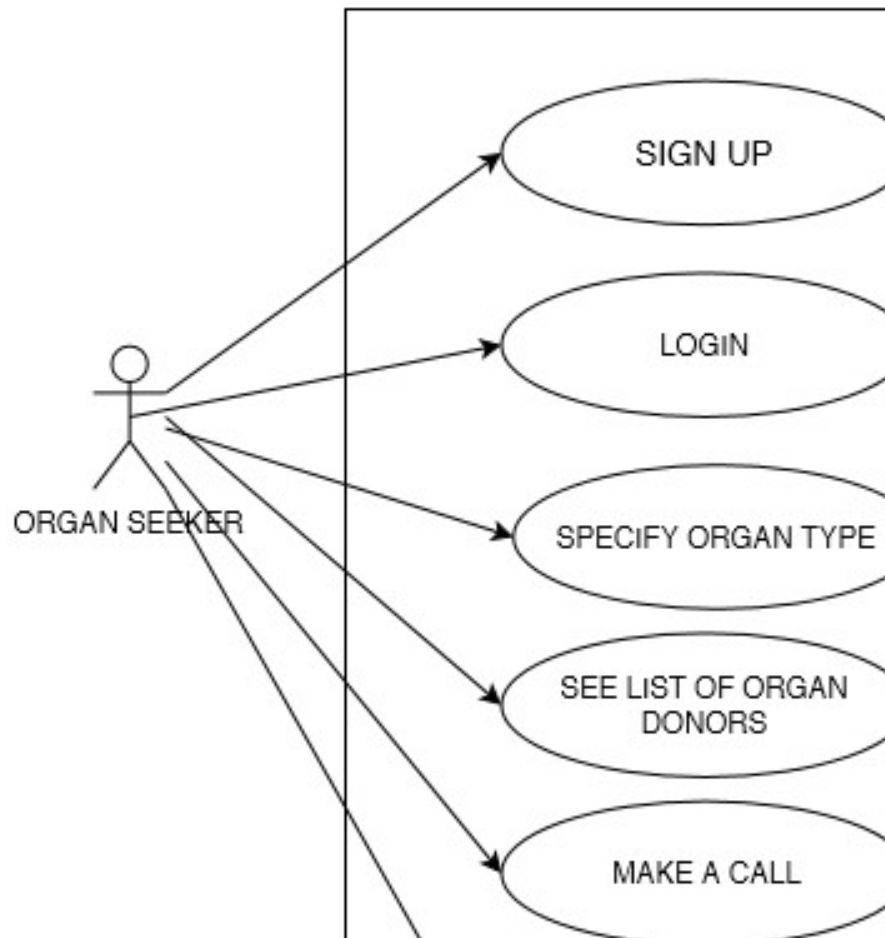
Description: To Register himself/herself for donating organ .

Precondition: valid email address and phone number.

Basic use-case flow: For the registration process, the user/patient needs to give some information by filling the form. Some of the required information includes the following: organ type ,Email address and phone number etc.

Main scenario: The user/patient will go onto the patient sign-up button either from the main page or from the drop-down menu. After clicking the sign-up button, a registration form will appear, where the user must give his personal information i.e. organ type, email address, phone number etc. After giving the required information user will submit the form. If all the fields are filled including the valid email address and phone number, the user will be registered onto the system and the user can register for donating organ .

5.1.6 Organ Seeker's Use Case



Use-Case 1: Sign Up

Primary-actor: generic user/organ seeker.

Description: To Register himself/herself requesting for blood .

Precondition: valid email address and phone number.

Basic use-case flow: For the registration process, the user/patient needs to give some information by filling the form. Some of the required information includes the following: organ type ,Email address and phone number etc.

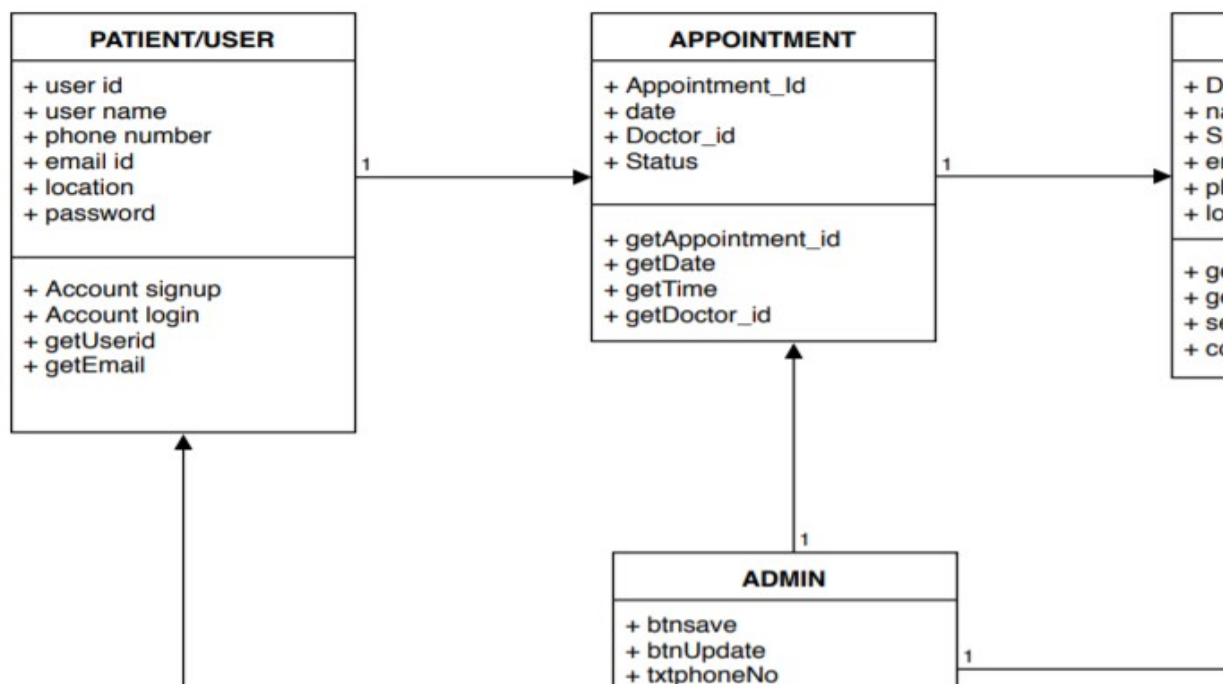
Main scenario: The user/seeker will go onto the sign-up button either from the main page or from the drop-down menu. After clicking the sign-up button, a registration form will appear, where the user must give his personal information i.e. blood group, email address, phone number etc. After giving the required information user will submit the form. If all the fields are filled including the valid email address and phone number, the user will be see the list of organ donors and post a request or can call directly to the donors of matching organ group of his/her.

5.2 System Design

5.2.1 Class Diagram for Appointment Booking

The class diagram is chosen to explain the design phase of the system. A class diagram describes classes of the system, attributes, and operations and relationships of the classes in a better way. We can also say that class diagrams are used to justify the structure or behavior of use cases of the system. Class diagrams best explain the conceptual model of the system in terms of entities and their relationships. The class diagram looks like a shape of a rectangle, comprising three compartments stacked vertically. The first top box comprises the class name, the second middle box contains the attributes of the class and third the last box contains the methods or functions performed by that class. The first compartment /box of the name is compulsory while rest of the two can be omitted to simplify the diagram. So, in any class diagram first compartment must be drawn while the second two compartments are optional.

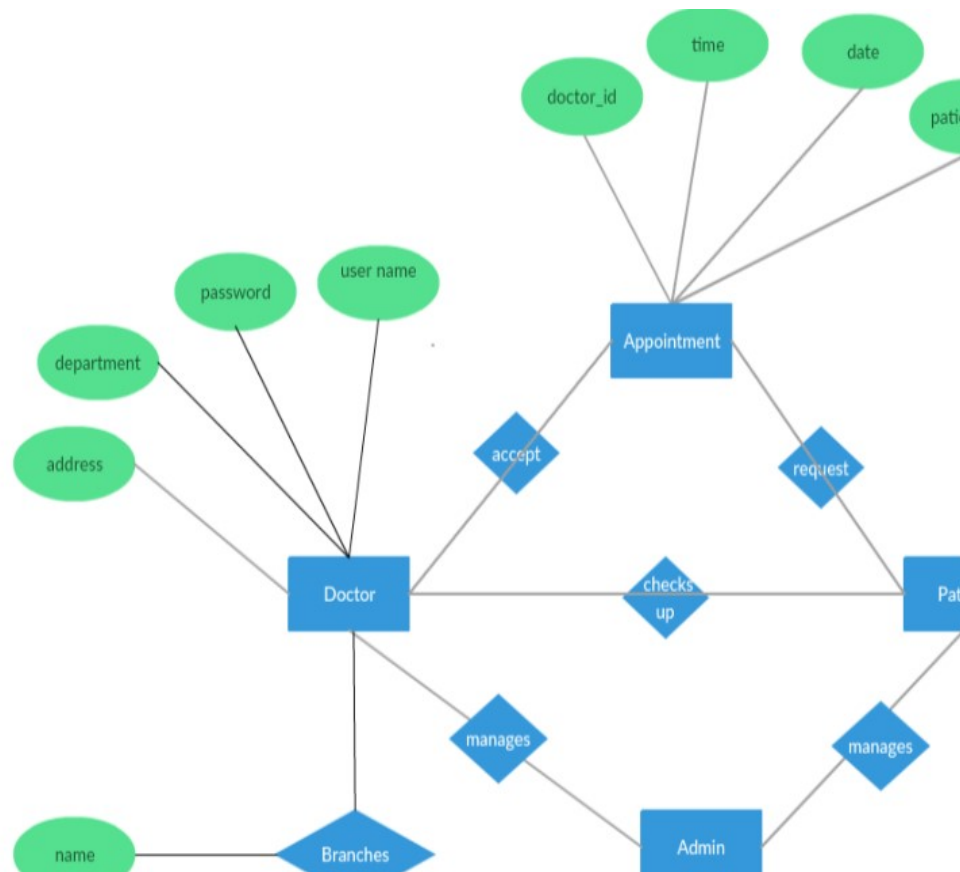
The class "patient" contains multiple parameters (such as id, name, age, address), which depict the information of all the registered patients. The user class also contains the methods performed by these users such as get appointment, view/ create own medical record etc. In the same way, the class "doctor" has the parameters id, name, department, address possessing all the required information of the users registered as a doctor on to the system. Methods include accept/reject the appointment, check the patient, view a medical record of any patient etc. These methods are the functions performed by the users registered as a doctor on the system. The class "appointment" has the parameters of date and time, explaining what time or day patient user has requested for the appointment to the doctor. The class "department" has the parameters id and name and methods include add/ delete doctor and add or delete department. Every doctor user must belong to any department class. The "report" is another class containing methods like write report or prescribes medicine.



5.2.2 ER Diagram

A basic ER model consists of objects called entities and specifies relationship among those entities. Purpose of this diagram is not to define any functionality rather show association and dependency among entities. ER diagram is drawn with "rectangular boxes" as entities and the "straight lines" showing the relationship between these boxes. An entity is an object or a thing that has an independent existence and can be easily differentiated from others. Each entity has some attributes like name, age, address, department etc. In the following diagram the doctor, patient, appointment, admin etc, all are different entities.

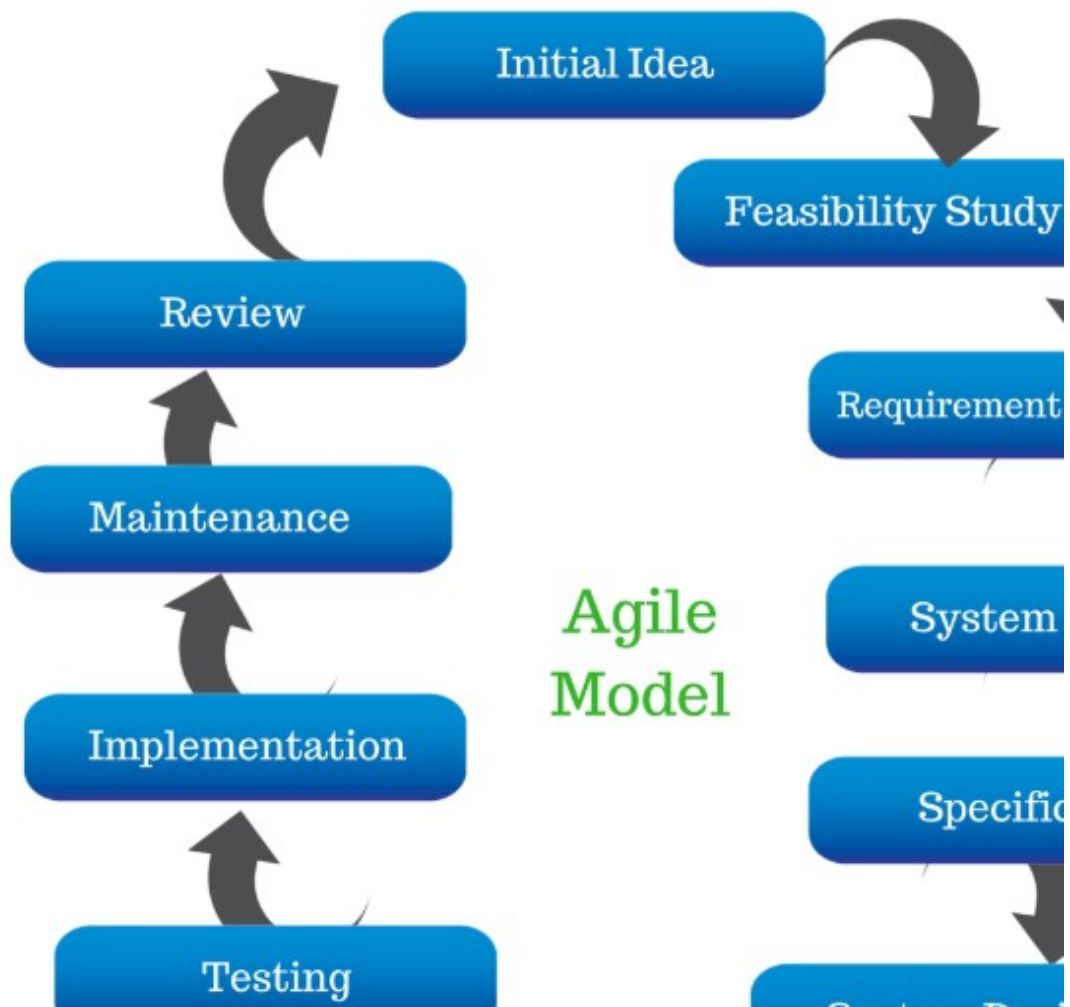
Entities consisting of similar attributes make the entity sets. These entities have some association among each other which make a relationship. These relationships can be "one to one" or "one to many" or "many to many". For example, a doctor and department can have "one to many" relationships, means one department can have many doctors but one doctor is related to only one department.



6. SOFTWARE TESTING

6.1 System testing

For the sake of software quality assurance, system testing is a very essential thing to do. It is a process by which we try to make the system error proof by performing the program to find an error. The goal is to run the program, find errors or bugs and then fix them. Testing is considered a very essential step in software development and any system is not considered to be complete without this process. Different developers follow different approaches for system testing. System testing is followed to make sure that system is working fine and meets the requirements.



TEST CASE ID	TEST CASE NAME	PRECODITION	INPUT TEST DATA	STEPS TO BE EXECUTED	EXPECTED RESULT	STATUS
C01	User_registration	No	Valid phone number	Need to fill the registration form and submit	Registration successful	OK
			Invalid phone number		Error Message	Error
C02	User_login	Must be registered earlier	Valid phone number and password	Enter phone number and password and submit	A successful login to the system	OK
			Invalid phone number or password		Login failed, and the system displays an error message	Error
C03	Doctor_signup		Valid phone number and license certificate	Need to fill the registration form and submit	Registration successful	OK
			Invalid phone number or license certificate		Error Message	Error
C04	Doctor_login	Must be registered earlier	Valid phone number and password	Enter phone number and password and submit	A successful login to the system	OK
			Invalid phone number or password		Login failed, and the system displays an error message	Error
C05	Organ_donor's Registration	No	Valid phone number and address	Need to fill the registration form and submit	Registration successful	OK
			Invalid phone number		Error Message	Error
C06	Blood_donor's Registration	No	Valid phone number and address	Need to fill the registration form and submit	Registration successful	OK
			Invalid phone number		Error Message	Error

7 CONCLUSION

The major goal of this application is to make an online interaction between doctors and patients. Clinivex - A doctor's and Patient's application meets most of the functionality to manage appointment with doctors and patients accordingly. Also the patient can get the doctor of his choice through various filters used in the application. The doctor is also able to view his day to day appointment list which makes it easier for him to plan his schedule.

The purposed system also simplify and automate the process of searching the blood in case of emergency and maintaining the records of blood donors, recipients, blood donation programs and blood stocks in the bank. Using this application ,blood seeker can search for blood donors and can call or message the donors. This application can also be used by organ donor and seeker where person can register for organ donation.

8 REFERENCES

- [1].OplooyeeLabs.“Doctorola.”, Internet:<https://doctorola.com/>, Jan. 3, 2017[June. 10, 2016].
- [2]. Dayspring Limited. “Doctor Apps Pulse. ”, Internet: <http://www.pulse.com.bd/> , Oct. 21,2014 [Feb. 6, 2016]
- [3]. Frank Sposaro and Gary Tyson, “iFall: An android application for fall monitoring and response”, 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1:6119–22, 2009.
- [4]. P. Priya, V. Saranya, S. Shabana, Kavitha Subramani :”The Optimization of Blood Donor Information and Management System by Technopedia” [IJIRSET] - Feb 2014.
- [5]. Tushar Pandit, Satish Niloor, A.S. Shinde :”A Survey Paper on E-Blood Bank and an Idea to use on Smartphone” [IJCA] - March 2015.
- [6]. Prof. Snigdha, Varsha Anabhavane, Pratikshalokhande, Siddhi Kasar, PranitaMore :”Android Blood Bank” [IJARCCE] – Nov 2015.