

PROJECT
ON
**“ANDROID BLOOD BANK SYSTEM FOR THE HEALTH CARE
PROVIDERS OF BANGLADESH”**

Submitted to:

Ahmed Imran Kabir

Lecturer

School of Business and Economics

Submitted by:

Ferdous Wahid Raz

ID: 111 193 105

Major: Management Information Systems

Spring 2023



School of Business and Economics
United International University

Date of submission: 4th May 2023

Letter of Transmittal

Date: 04, May 2023

Ahmed Imran Kabir

Lecturer

School of Business & Economics

United International University

Subject: Report submission regarding the "Android Blood Bank for the Healthcare Providers of Bangladesh" project

Dear Sir,

This is to let you know that I did my best to thoroughly research the assigned topic of the project, " Android Blood Bank for the Healthcare Providers of Bangladesh."

As part of the university's requirements for my bachelor's degree, I completed this assignment. I have done all possible to complete the project's goals and live up to everyone's expectations. I'm hoping you'll be kind enough to accept my project work and offer me helpful critique regarding my effort. I'm sure the final report's findings and presentation will meet up to your expectations for it.

Sincerely Yours,

Ferdous Wahid Raz

ID: 111 193 105

Signature:

Date: 04th May, 2023

Certification of Similarity Index

DECLARATION of the STUDENT

I, hereby, certify that the work, which is being presented in the project report entitled “Android Blood Bank for the Healthcare Providers of Bangladesh”, brings into light my work completed under Sir Ahmed Imran Kabir’s guidance. I verify that the submissions and conjectures put together result from my own work. Additionally, I authenticate that -

I have not submitted the matter embodied in this dissertation for the award of any other degree at this or any other University. The report was written as per the university’s guidelines. Wherever I had incorporated resources, I credited it and due rights were given and appropriately cited.

Ferdous Wahid Raz

ID: 111 193 105

ACKNOWLEDGEMENT

"Task completed" makes everyone happy. But happiness will be gold without glitter if we do not acknowledge those who have helped us make it a success. Only those who made it a reality will be acknowledged with success, but those whose persistent counsel and support made it possible will be acknowledged first on the eve of success.

At the outset, I express my heartfelt gratitude to Almighty Allah for providing me with the ability and guidance not only during the project period, but throughout my life. Second, I'd want to thank my Project Supervisor, Ahmed Imran Kabir, for his assistance. His assistance and suggestions have been invaluable in finishing the project work at various stages.

Finally, I want to thank my family for their unwavering support throughout my life, notably during my undergraduate years, and for assisting me in finishing my coursework in general and this project in particular.

ABSTRACT:

Blood donation is one of the most significant contributions that a person can make towards the society. The **World Blood Donor Day** is observed on 14th June every year.

An ailing body needs blood for various reasons. It may be due to being diagnosed with anemia, loss of blood during an operation or even an accident. However, the requirement for blood may not always be satisfied and can lead to dire consequences such as complications or in extreme cases even death. The recipient in question requires blood group of a specific type. This is the most major factor in blood donation. Storage of blood is a tedious and strenuous process. It requires proper inventory management and equipment.

In recent times a steady increase has been noticed in the number of blood donors thanks to the various organizations setting up blood donation camps on various occasions. At present, the blood banks are not receiving a sufficient inflow of blood. This is where such camps and initiatives come into play by helping and increasing the inflow of blood. At the same time, it is imperative to ensure that the donor is disease free and the blood being donated is checked to ensure normality.

A very nice and efficient way to ensure proper functioning of a blood bank would be to link it to the internet via a mobile application which in turn is connected to a database which is managed by the blood bank. The application can be geo tagged to show the location of donor and recipients to ensure speedy service as per the requirements. Blood donation doesn't cause any short comings in the functioning of our body as perceived by some sects of our society. In fact, the body can refill the loss within a few days.

[KEYWORDS: Blood groups, donation camps, mobile application, database, GPS.]

Contents

CHAPTER 1:- INTRODUCTION:	9
1.1 PROBLEM IDENTIFICATION	10
1.2 PURPOSE	11
1.3 OBJECTIVE	11
1.4 LIMITATIONS	12
1.5 SIGNIFICANCE OF THE STUDY	12
CHAPTER 2:- LITERATURE REVIEW:	13
2.1.1 Influential Factors to donate Blood	14
Several factors were found to influence the willingness to donate blood amongst people namely:	14
2.1.2 Resources for blood donation	15
2.2 Objectives	16
2.2.1 Objective of the App:	16
2.2.2 Main Reason for Study:	16
2.2.3 Justification for Study:	17
2.3 Significance of Study:	18
2.4 Expected Results:	19
CHAPTER 3: METHODOLOGY	20
3.1 Methodology	20
3.2. Design stage	22
3.3. Implementation stage :	23
3.4. Testing Phase:	24
3.5 RESULT	24
Chapter 4:- DESIGN DOCUMENT	25
4.1 Software Architectural Design	25
Spiral model	25
History	26
Six invariants	27
Define Artifacts Sequentially	27
Perform Four Basic Activities in Every Cycle	27
Risk determines level of effort	27
Risk Influences The Level of Details	27
Use Milestone Anchor Points	28
Emphasize on the Framework and its Life Cycle	28

4.2 UI DESIGN	28
4.3 Programme Design	28
Chapter 5:- CONSTRUCTION OF CODE	30
5.1 Login Page	30
5.2 Login Data Base Adapter	31
5.3 Registration	33
5.4 Received Blood	35
Chapter 6:- Verification AND Validation	37
6.1 Login Test Case	37
6.2 Validation Test Case	37
Bibliography	39

CHAPTER 1: INTRODUCTION

Internet in today's world is the centre point of all technological and technical innovations. The ability to accurately detect a persons need and requirement and the closest point of availability of the required blood is the main goal of our project. Any person can download the application, register themselves and go on to locate blood according to their requirement.

In this fast paced world, many situations arise where there may be a requirement of a certain type of blood. It may not always be feasible to go to different locations to find availability according to requirement. This gives rise to the requirement of a central repository which is easily accessible for everybody. This is where our mobile application comes into play by putting into people's phones the power to locate a place where the required blood is available. This in turn would save those precious minutes which could mean the difference between life and death. To help counter such problems an "app" is being developed using GPS and Google Maps.

Various problems that are faced:

- To find a blood bank with availability as per requirement
- Wastage of time in physically going to locations and checking availability
- Lack of service in remote locations

The app would have the following features embedded in it:

- Blood Bank Catalogue: List of all the blood banks and their details, info
- Mapping: It will map all the blood banks
- Service Request: User request for services provided by the blood banks

Location based service (LBS) is playing a pivotal role in today's technological world. It is basically the ability of the service to provide time information depending on the user's location.

A LBS have two major functions that include:

1. Obtaining the location of user
2. Utilizing this information to provide a service

These 2 actions are used to answer the questions mentioned below for a mobile user in a fast and efficient manner:

- Current location
- Location of nearest blood bank
- Availability
- Directions

The GPS satellite is a navigational satellite that provides almost accurate location information, and combined with LBS, is used to track a mobile device's location. GPS calculates position as coordinates of latitude and longitude values through the GPS receiver. The GPS device is a tiny device that is embedded in various devices like mobile phones, watches, cars etc. **(International Engineering Consortium, n.d.)**

Steps involved for Google maps are:

- Firstly, location information is sent to remote parties. This information is then used today, e.g., in location tracking applications
- Secondly, the location information is used to make communication decisions, e.g., a user agent may automatically disable instant messaging when driving

1.1 PROBLEM IDENTIFICATION

The application will get widespread acceptance only after it overcomes certain constraints:

As it holds information regarding the various blood banks therefore, it is highly recommended to sign up & register all the blood banks so that when the user's location is tracked the server is able to show all the nearest blood banks and contact the relevant blood bank and ask for assistance.

Therefore, user input data is a must for the repository to be compiled over time and handled in a safe, secure and private manner. Privacy could be an issue, hence only limited data should be collected, which is essential for the service to be provided.

1.2 PURPOSE

The main purpose of this work is to design and implement an application which will help the user to locate their nearest blood bank through their mobile because previously people could not locate the banks conveniently in case of an emergency. This in turn will reduce the time spent in physically going to the desired blood bank and checking the availability. During an emergency every second is valuable. Using the data stored in the app, such as contact number and location would make it easier and faster to track a donor or hospital or a blood bank where the particular type of blood would be available nearby.

1.3 OBJECTIVE

- The objective of the project is tracking user location and providing the nearest blood bank with all relevant information such as blood availability.
- To provide a speedy solution to the people who face emergencies and require blood urgently.
- To provide a proper medium between doctors, hospitals, patients and donors to communicate and keep a tab on the availability of blood required.

1.4 LIMITATIONS

The survey regarding this application includes information gathering from various sources. These sources include some of the blood banks: some big, with access to elaborate paraphernalia and others small, which don't have such access.

During an emergency, if there is no signal range then the customer won't have access to server, as this app is location-based and requires the mobile signal to help triangulate their position via GPS.

1.5 SIGNIFICANCE OF THE STUDY

The group of people who will be benefitted via this app are those who have a smartphone. The people involved here can immediately log on to the app and request the nearest blood bank for assistance in the event of an emergency. Instant information availability of the donors for patients and their families or the doctors and hospitals for urgent blood supply as per requirement.

CHAPTER 2: LITERATURE REVIEW

The scope of the study is to develop a crowd-sourced Blood Bank app so as to create and manage donors' records and of the patients and their families who require blood. This will create a database from where people can get and share the information regarding their requirements of blood impromptu.

Since there are several people, at the very least a few million people worldwide, who need blood transfusions, it is of utmost importance to have a concise database of where and how to avail the blood at the time of need. Blood transfusions are needed so as to replace the lost blood during surgery, accidents or illnesses such as anemia, cancers.

Millions of people need blood transfusions each year, some of them need blood transfusions to replace the blood lost during surgery or after having serious accidents that cause them to lose some of their blood. Others may need blood because of illnesses such as anemia, cancer, hemophilia, and immune disorders or other health issues. (*Global Blood Fund, n.d.*)

The patient may be prescribed the complete blood or part of it, that depends on the patient's conditions. Certain patients require just the hemoglobin while others may need plasma or platelets. (**"Abstract," 2021**)

The project's intent is to develop a GPS and web-based Blood Bank System that would provide the option not only to enter the data but also, manage and make it a user-friendly database to retrieve the required data whenever needed.

In spite of the potential blood donors being available, barely 1% of the total population of Bangladesh donates blood as per the 2016 World Health Organization report. (*Bangladesh Is Still to Meet the Demand of Safe Blood Supply, 2017*)

With the advancement of medical science, more hospital infrastructure being constructed and healthcare being increasingly improved, the need for blood to save lives has also increased. It has therefore become imperative for us to take a more methodical approach in creating a new blood donation system.

This new system ought to eliminate the issues regarding the current blood donation system in a well-organized manner. Most importantly, timely updates would address the issue of supply and demand during an emergency and hasten the current process of blood donation. Blood donor's recruitment in Bangladesh is very less in comparison to other countries, with overall less annual blood donation percentage. Not only this, but the donor screening and the management system to assimilate their data is also not well-maintained. The details of donors are given for the usage of the users so as to contact them whenever blood is required, especially in times of emergency.

A study was performed on 284 students in the Netherlands and found that when subjects had less fear of needles and/or did not perceive the process as painful and time consuming they had more positive views of the whole process of blood donation.

2.1.1 Influential Factors to donate Blood

As per the study several factors were found to influence the willingness to donate blood amongst people namely:

- People who had donated before were more willing to donate again.
- Shorter length of time elapsed from last donation.
- Higher aged people (50-70) years.
- Having a higher education which inculcated awareness.
- People involved in social circles where there were donors present.

(Md. Rezaul Karim et al., 2012)

There are several underlying problems which exist in the medical field but one of the most serious is the immediate requirement of blood for the injured or for patients who recently underwent a major surgery. Presently, not enough blood is available despite several blood banks being there.

Certain websites try to mitigate the issue by providing phone numbers of donors but considering they aren't updated in a periodic manner, phone numbers are wrong and then it becomes difficult to contact the donors. Not only this, but there is barely anyone to help alleviate the concerns of the people who could be potential donors. Their concerns could be about the after care of the donor, or regarding the medical history not correctly shared with the doctor.

If the donor has or had any medical issues or the blood is infected, the patient could be in a threat and it could be a life-or-death situation. As per the Bangladeshi Health Ministry guidelines

certain laws have to be adhered to, such as:

1. Anemic or underweight people should not donate blood.
2. People with transmissible diseases, especially via blood like AIDS, Hepatitis and various other viral infections are disqualified from donating.
3. Pregnant women and women who recently gave birth to a child are ruled out to ensure the safety of the donor during and after pregnancy as there may be medical risk to the mother and child.

(Rummana Ferdous Fagun, 2022)

There can be certain short comings in many blood donation systems such as:

1. Inability to handle emergencies.
2. Data security and privacy of the individual's personal details.
3. Misuse by third parties.
4. No timely and proper updates.
5. Requires a middleman to operate and update the information.
6. Errors in the data leading to erroneous results.

2.1.2 Resources for blood donation

The following section talks about the various blood bank services present in Bangladesh & abroad, it highlights their best features and provides relevant advice also:

Bangladesh Red Crescent Society: Donors in Bangladesh can register at BDRCS after checking out the basic requirements of blood donation. BDRCS requests the donors to register at their centres, located throughout Bangladesh, and then proceed to donate blood.

(Bangladesh Red Crescent Society, n.d.)

Blood Donation Bangladesh Welfare Foundation: Although donors could register themselves to come and donate blood in this foundation, it has been observed that the website has not been updated or maintained since before the COVID-19 pandemic hit the world.

(Blood Donation Bangladesh Welfare Foundation, n.d.)

Blood Bank Management System in Bangladesh: Bangladesh rehires about 9 lakh units of blood annually, of which only 34% of it is collected through blood donation. In this website, donors can easily register themselves and input their personal information as required.

(Sera Doctor, n.d.)

2.2 Objectives

2.2.1 Objective of the App:

- Develop an MIS to maintain record of blood donors.
- Ensure smooth distribution of blood amongst various patients.
- Voluntary blood donation is encouraged.
- Reduce human prone error in system.

2.2.2 Main Reason for Study:

With the ever increasing number of people requiring blood, availability remains a huge obstacle in ensuring people get the required amount when and where they need blood. Despite many people showing their eagerness to donate blood, the lack of information regarding the location of blood banks, the benefits of blood donation and how they can help save lives etc. can be a detrimental factor in coming forward to donating blood.

With the convenience of an app that provides them information about the requirement, location and supply shortages of blood, many people would come forward in helping and showing their eagerness in donating blood.

At the same time, with the increase in blood donations, the upkeep of donor records could become an issue subject to human error. With proper management of demand and supply of the Blood Bank within the app itself, it will be better managed and handled and the records would be there for easy retrieval.

A person who needs blood may follow one of these ways:

- Ask friends & family for a suitable donor.
- Contact hospitals checking for availability regarding suitable donor.
- Research the World Wide Web looking for a donor.

As you can see the above mentioned ways although not totally useless are very time consuming and could mean the difference between life and death for a patient.

Under such circumstances the best thing to do would be:

To log onto a blood bank application and get complete donor details according to our specifications, needs & location. Via this method useful information can be retrieved within minutes and can help save those precious minutes which can make a difference.

2.2.3 Justification for Study:

Blood donation is a self-directed volunteer service. The outcome of this research will help in developing a model which will be very efficient and will also tackle the issues currently being faced by the present system. It will also help educate the masses regarding the benefits of blood donation.

The reasons for non-donation were mainly related to the fear of catching diseases especially the HIV, the lack of information, the religious beliefs and the fear of knowing the result of one's HIV test. Though interest in the blood donation among the population new strategies must be elaborated to inform the population and motivate blood donors.

The main outcome of this study was to:

1. Handle donors, hospitals, and recipient's records.
2. Bring down human error during the upkeep of the records.
3. Blood banks and hospitals can register and create an account on the app then update it with information regarding the hospital, blood types' requirement and availability.
4. Persuade people to voluntarily do blood donations.
5. Ease the distribution of blood in various hospitals in Bangladesh, based on the hospital demands.
6. To facilitate easier donor-recipient contacts for quick and easy search as per the requirements, especially in times of emergency.
7. Create a blood MIS to manage blood donors and recipient's records.
8. Each registered individual can add their own comments in the feedback section within the app.

2.3 Significance of Study:

Around the world, inadequate blood supplies at hospitals and medical facilities has always been an issue. Creating awareness regarding the importance of donating blood among the general public, means that donors are available at all times, play a significant role.

With this project, the organization that would avail the most benefits from are the hospitals. Due to the rise in awareness of the positives of blood donation amongst the general population, events advertising blood donation could further bolster the efforts.

Furthermore, the system will enable the central administrator to facilitate a convenient medium to co-ordinate the blood supply among blood banks, hospitals, donors and patients alike.

Furthermore, the top-level management staff at blood banks and medical facilities can efficiently manage the blood as well as the donor records with better security and added automation function in alerting the administrator when the blood quantity is below par level. In addition, the administrator can trace the person responsible in case of human error such as fault blood transfusion and blood screening result.

The top-level management staff can provide each staff different levels of access for added security and privacy. The blood bank medical lab technicians gain the advantage of having an automated system in assisting their work.

This also provides an environmental friendly approach for the hospitals as the application necessitates the use of less paper by the doctors, nurses and management as the records of previous donors are all kept in a centralized manner. The Object-Oriented approach used to develop this system is the best as the system will be easily maintained.

2.4 Expected Results:

The anticipated outcome of this project is to have an Online Blood Donation Reservation and Management system with two main portals, one for the public and another one for the hospital administrator.

The portal for the public displays the blood donation events to be held and allowing the public to make online reservation. Whereas, the admin portal functions and manages appointments, manage system users, donors and blood stocks and generate reports. Also, the automation function of giving alert to the admin when the blood quantity is below par level is also provided.

The kinds of reports that will be generated by the system are the blood quantity in the blood bank and workflow for each blood donation process.

CHAPTER 3: METHODOLOGY

3.1 Methodology

The waterfall methodology, which is the most common method of system development life cycle, is used for the creation and implementation of the website/application.

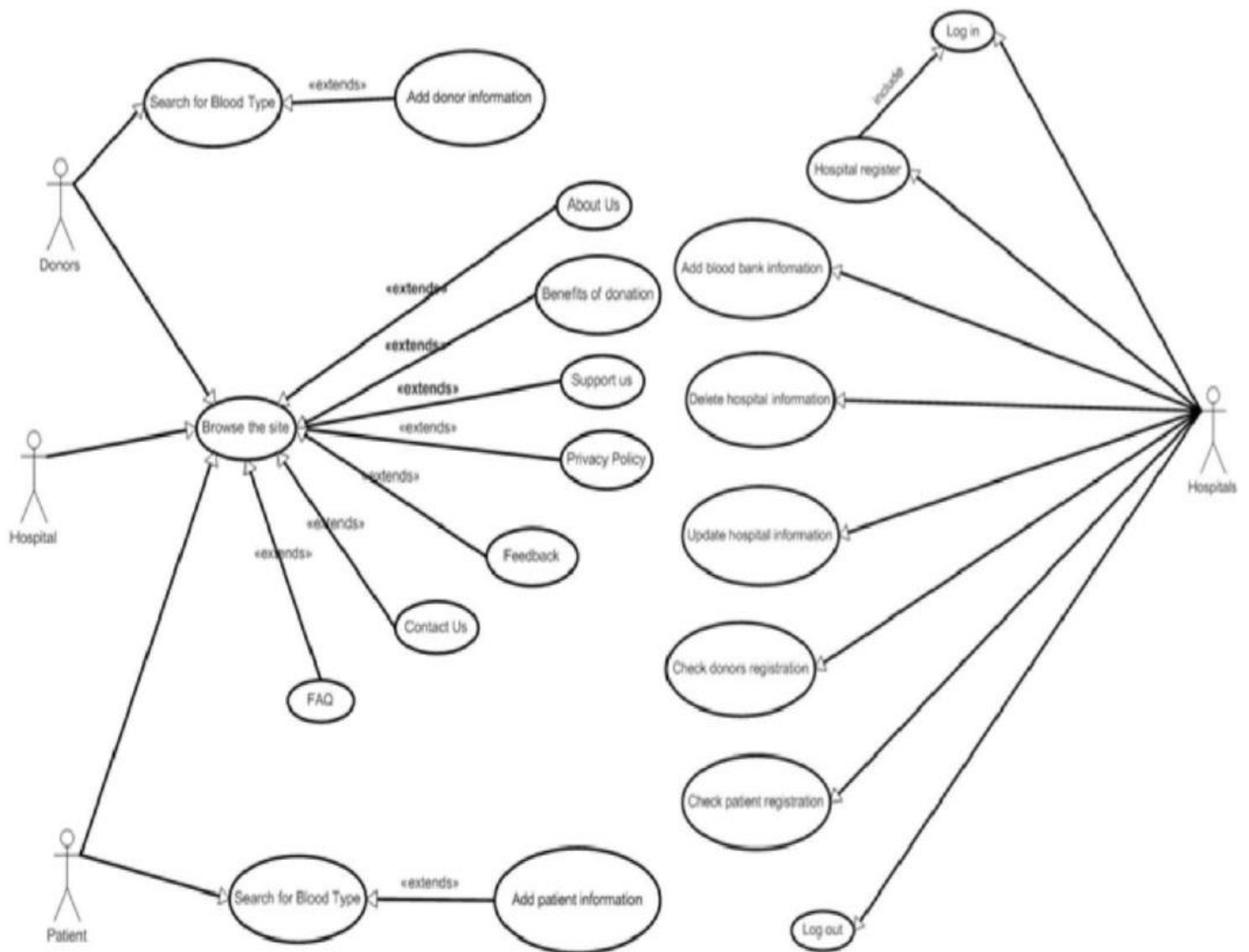
The sequential and linear development method is described. This methodology is used for its clear objectives and goals during each step of the project.

Some of the steps that have been taken to build the blood bank application are:

1. Initial stage

- Identify the problem.
- Search for similar research, determine the objectives of each system.
- Summarize it in one table.
- Determine overall objective.
- Analyze survey & questionnaire results.
- Identify requirements.
- Make overall project plan.
- Analyze hardware & software requirements.
- Identify end users.

Use Case model:



3.2. Design stage

- The looks of the application are defined and prepared from the required data.

Three designs that must be done during this stage are:

- **Prototype Design:**
 - Under this design type there are a further of two types:
 - High Fidelity prototype: It is close enough to the completed product so that the final product could be examined from the usability point of view.
 - Low Fidelity prototype: They are constructed to depict concepts, design alternatives & screen layouts.
- Database Design
- Physical Database Design
- User Interface Design Principles.

3.3. Implementation stage:

This stage is the actual “doing” phase and its main objective is to help in putting together all planned activities and actions together to get the final product.

For the purpose of designing, coding the application & website the following languages were used:

- HTML:
 - Main use is to create the webpages. Web browser should read documents as webpages.
 - Scripts can also be included as and when required (JavaScript).
- PHP:
 - Provides functionality which native HTML cannot.
 - Used for dynamic web page creation.
 - Collects then processes and utilizes data to give desired output.
- MySQL:
 - Used because easy compatibility with PHP.
 - Can store many kinds of data.
 - Easy accessibility.
- JavaScript:
 - Designing interactive websites.
 - Developing web applications.
 - Effective interaction with HTML enabling dynamic content access.

3.4. Testing Phase:

- Testing the final product with the sole objective of finding bugs & errors.
- It also should ensure that all task & processes in the system are working correctly in proper conjunction with each other.

3.5 RESULT

The website & application developed will aim to:

- Efficient management of donor, hospital & recipient's medical records.
- Reduction of human error.
- Allow for registration of a hospital's blood bank or any new one.
- Encourage blood donation.
- Make it easier for recipients to find a suitable donor in minimum time.
- Education of community on benefits regarding blood donation.
- Mitigate blood distribution amongst various hospitals, blood banks etc.

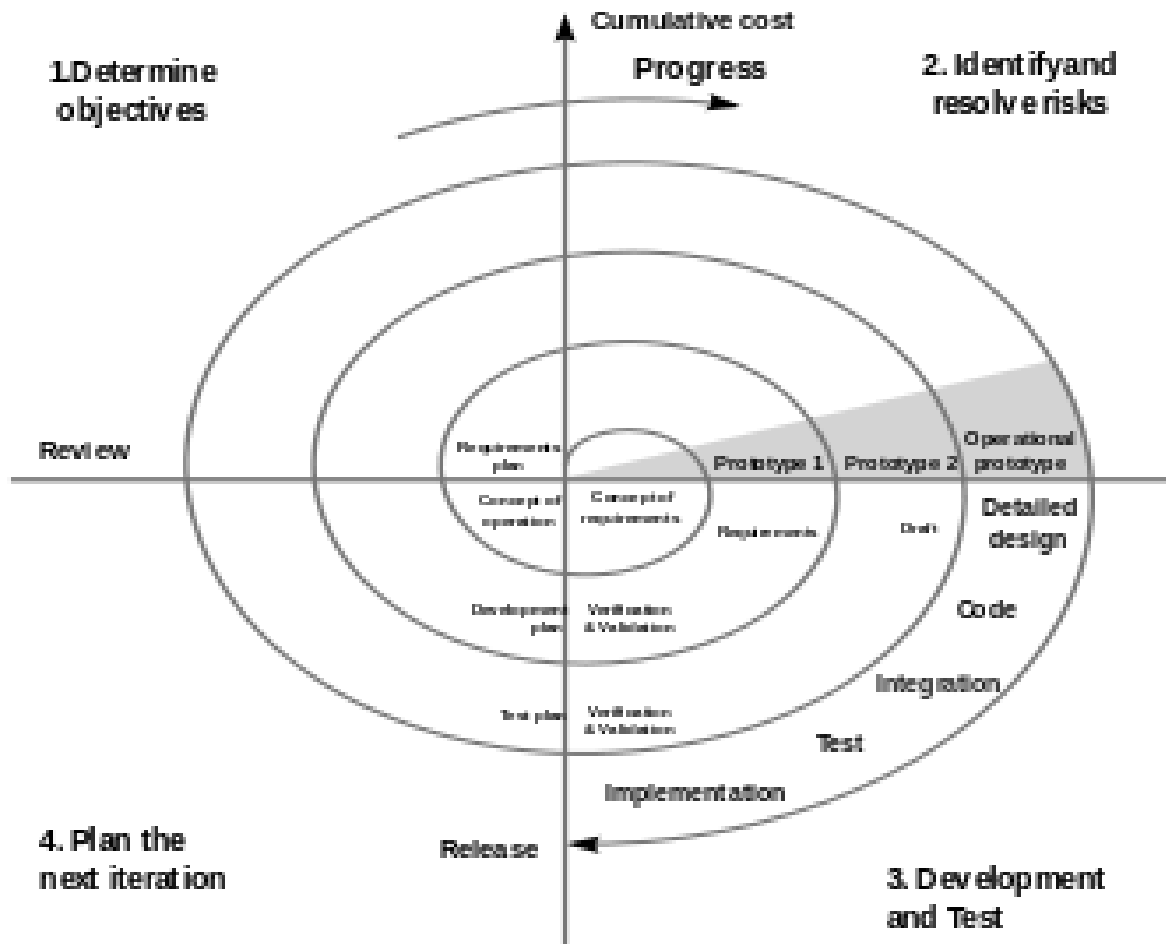
Chapter 4: DESIGN DOCUMENT

A design document serves the main purpose of talking about the design and the scope of a particular software providing the development team with vital information to help the design the software in a better and efficient manner. It basically tells the development team about the architecture of the software project.

4.1 Software Architectural Design

Spiral model

In software development process, the **spiral model** is a risk-driven (*Software Development Process, n.d.*) that provides support for **Risk Handling**. There are several loops of the spiral, whose number is unknown, and depends from project to project. Each loop is called a **Phase** of the software development process. (**Boehm, 1986**)



History

This concept was first introduced by Barry Boehm in his 1986 work "A Spiral Model of Software Development and Enhancement." He then published a similar study to a larger audience in 1988. First presented was a figure that was later used to explain the spiral model in numerous publications.

In these early works, the word "process model" is used to describe a variety of methodologies, including the spiral model, waterfall, incremental, and prototyping.

Boehm lists six traits shared by all genuine uses of the spiral model in order to better separate them from "hazardous spiral look-alikes".

Six invariants

Authentic applications of the spiral model are driven by cycles that always display six characteristics. **(Boehm, 1986)**

Define Artifacts Sequentially

This invariant excludes "hazardous spiral look-alike" processes that use a series of incremental waterfall passes in situations where the waterfall model's underlying assumptions do not apply. The requirements are known prior to implementation.

It is a project risk not to identify the requirements and move sequentially in instances where these assumptions do apply. As a result, the waterfall model is a risk-driven variant of the spiral model.

Perform Four Basic Activities in Every Cycle

This invariant identifies the four activities that must take place in each spiral model cycle:

1. Consider all success-critical stakeholders' win conditions.
2. Identify and assess various approaches to meeting the win conditions.
3. Identify and resolve hazards associated with the chosen approach(es).
4. Obtain approval and commitment to the next cycle from all success-critical stakeholders.

This invariant is violated by some "hazardous spiral look-alike" procedures that exclude important stakeholders from certain sequential phases or cycles. System maintainers and administrators, for example, may not be invited to participate in the system's definition and development. As a result, the system risks failing to meet their win conditions. **(Boehm, 1986)**

Risk determines level of effort

The project team must decide how much effort is sufficient for any project activity (for example, requirements analysis, design, prototyping, and testing). These judgments are made in genuine spiral process cycles by reducing total risk.

"Hazardous spiral look-alikes" that violate this invariant include evolutionary processes that neglect risk owing to scalability concerns, and incremental processes that substantially invest in a technical architecture that must be modified or replaced to support future product increments. **(Boehm, 1986)**

Risk Influences the Level of Details

For each project deliverable (e.g., requirements and project specification, design documentations, test plans, etc.), the project team must decide how much detail is sufficient. In a true spiral process cycle, these decisions are made by decreasing comprehensive risk.

Using the requirements and project specification as an instance, the project should specify exactly what features are mitigated by accurate specification (e.g. interfaces between hardware and

software, interfaces between prime contractor and subcontractor). Conversely, projects should not precisely specify features (e.g., graphical screen layouts, behavior of off-the-shelf components, etc.) where precise specification would escalate the risk. **(Boehm, 1986)**

Use Milestone Anchor Points

In later editions, Boehm's incorporates three anchor point milestones that act as success indicators and points of commitment. Key questions can be used to identify these anchor point milestones.

1. Goals for the Life Cycle. Is there a clear definition of a technological and managerial strategy that meets everyone's win conditions?
2. Architecture for the Life Cycle. Is the preferred approach sufficiently defined to fulfill everyone's win conditions, and have all relevant risks been eliminated or mitigated?
3. Basic operational capability. Is the software, site, users, operators, and maintainers sufficiently prepared to meet everyone's win conditions by launching the system?

Evolutionary and incremental processes that commit large resources to constructing a solution with a poorly specified architecture are examples of "hazardous spiral look-alikes" that violate this invariant. **(Boehm, 1986)**

Emphasize on the Framework and its Life Cycle

This invariant emphasizes the significance of the total system as well as the long-term concerns that span its full life cycle. It excludes "hazardous spiral look-alikes" that are overly focused on initial software code development. **(Boehm, 1986)**

4.2 UI DESIGN

The main features of any software application include:

- **Usability**: This basically encompasses the ease with which the application can be used by different classes of users such as novice & expert users.

A good software application should be well aligned, excessive use of tabs should be avoided so as to avoid over congestion which in turn would make the use of the software confusing and complicated. A good contrast between text and background should be maintained so as to make it easier to read content available on the application.

4.3 Programme Design

The objective of Online Blood Bank is that a person in urgent requirement of blood can quickly log on the application and via the internet connect to all the nearby blood banks and check for the availability of the kind of blood they require. This is achieved by creating an inventory of all blood banks and uploading it on the server which is attached to the application and can be accessed by users from their mobile phones in order to achieve this, the requirement was studied very carefully:

To draw an application, a development platform is to be used which is in Java.

- In this case, Android Studio is used for the development of R & R
- It has embedded SDK tools which gives huge functionality as well as a good graphical UI to user
- After the application is developed, a review is to be conducted for ensuring that the implementation of necessary methods, functions and modules are used
- Each phase is to ensured that, it is functioning as per the scope required
- If found loop holes, it is advisable to re-work on the desired module and check whether it is meeting the scope
- Finally, after the application is ready, several tests such as Black Box testing, Beta testing, Unit testing is carried out to find if there is any bug or error in the code and fix them
- After testing, when the final time comes for releasing it out to the world a sample must be tried and be reviewed by the nearby small group of people of the application. This would give an assurance whether the application is meeting the scope. If any negative reviews come, then it is advisable to stop the release and implement it again and rectify the negative reviews and again do the testing

Chapter 5: CONSTRUCTION OF CODE

5.1 Login Page

```
File Edit Selection View Go Run Terminal Help
Login.java - AndroidBloodBank - Visual Studio Code

J Login.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Login.java
1 package com.bloodbank.androidbloodbank;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.graphics.Paint;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.view.View.OnClickListener;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 public class Login extends Activity implements OnClickListener {
15
16     Button mlogin;
17     Button mcancel, signup;
18
19     EditText musername, password;
20
21     LoginDatabaseAdapter loginDatabaseAdapter;
22
23     /** Called when the activity is first created. */
24     @Override
25     public void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.login_page);
28         // create a instance of SQLite Database
29         loginDatabaseAdapter = new LoginDatabaseAdapter(this);
30         loginDatabaseAdapter = loginDatabaseAdapter.open();
31
32         mlogin = (Button) findViewById(R.id.login);
33         mlogin.setOnClickListener(this);
34
35         musername = (EditText) findViewById(R.id.editText1);
36         password = (EditText) findViewById(R.id.password);
37
38         signup = (Button) findViewById(R.id.register_new_user);
39         signup.setOnClickListener(this);
40         signup.setPaintFlags(signup.getPaintFlags() | Paint.UNDERLINE_TEXT_FLAG);
41
42     }
43
44     public void onClick(View v) {
45         switch (v.getId()) {
46             /*
47              * case R.id.cancel_login:
48              */
49         }
50     }
51 }
```

```
File Edit Selection View Go Run Terminal Help
Login.java - AndroidBloodBank - Visual Studio Code

J Login.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Login.java
49
50     * case R.id.cancel_login:
51     *
52     * #finish();
53     *
54     * break;
55     */
56
57     case R.id.login:
58
59         String value = musername.getText().toString();
60
61         if (value.equals("") || value == null) {
62             Toast.makeText(getApplicationContext(),
63                 "Please enter Username", Toast.LENGTH_SHORT).show();
64         }
65         * } else if (cnfpassword.equals("") || cnfpassword == null) {
66         * Toast.makeText(getApplicationContext(),
67         * "Please enter your Password Again", Toast.LENGTH_SHORT)
68         * .show();
69         */
70     } else {
71         // fetch the Password from database for respective user name
72         String storedPassword = loginDatabaseAdapter
73             .getSingleEntry(musername.getText().toString());
74
75         // check if the Stored password matches with Password
76         // entered by
77         // user
78         if (password.getText().toString().equals(storedPassword)) {
79             Intent int1 = new Intent(getApplicationContext(),
80                 AdminHomePage.class);
81             password.setText("");
82             musername.setText("");
83             Constant.value = "";
84             startActivity(int1);
85             // finish();
86         } else {
87             Toast.makeText(Login.this,
88                 "Username or Password does not match",
89                 Toast.LENGTH_LONG).show();
90         }
91     }
92
93     }
94     break;
95
96     case R.id.register_new_user:
```

```

Login.java - AndroidBloodBank - Visual Studio Code
J Login.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Login.java
70
71 // fetch the Password from database for respective user name
72 String storedPassword = loginDatabaseAdapter
73     .getSignInEntry(username.getText().toString());
74
75 // check if the Stored password matches with Password
76 // user
77
78 if (password.getText().toString().equals(storedPassword)) {
79     Intent int1 = new Intent(getBaseContext(),
80         AdminHomePage.class);
81     password.setText("");
82     username.setText("");
83     Constant.value = "";
84     startActivity(int1);
85     // Finish();
86 } else {
87     Toast.makeText(Login.this,
88         "Username or Password does not match",
89         Toast.LENGTH_LONG).show();
90 }
91
92 break;
93
94
95 case R.id.register_new_user:
96
97     Intent int1 = new Intent(getBaseContext(), Registration.class);
98     startActivity(int1);
99
100 break;
101
102 }
103
104 }
105
106
107 public void onDestroy() {
108     super.onDestroy();
109     // DB.close();
110 }
111
112 }

```

5.2 Login Data Base Adapter

```

LoginDatabaseAdapter.java - AndroidBloodBank - Visual Studio Code
J LoginDatabaseAdapter.java X
app > src > main > java > com > bloodbank > androidbloodbank > J LoginDatabaseAdapter.java
1 package com.bloodbank.androidbloodbank;
2
3 import android.content.ContentValues;
4 import android.content.Context;
5 import android.database.Cursor;
6 import android.database.SQLException;
7 import android.database.sqlite.SQLiteDatabase;
8
9 public class LoginDatabaseAdapter {
10     static final String DATABASE_NAME = "android_bloodBank.db";
11     static final int DATABASE_VERSION = 1;
12     // TODO: Create public field for each column in your table.
13     // SQL Statement to create a new database.
14     static final String DATABASE_CREATE = "create table " + "LOGIN" + "(" +
15         "ID" + " integer primary key autoincrement,"
16         + "USERNAME text,PASSWORD text)";
17
18
19     static final String DATABASE_CREATE_USER = "create table " + "LOGIN_USER" + "(" +
20         "ID" + " integer primary key autoincrement,"
21         + "USERNAME text,PASSWORD text)";
22
23     // Variable to hold the database instance
24     public SQLiteDatabase db;
25     // Context of the application using the database.
26     private final Context context;
27     // Database open/upgrade helper
28     private DatabaseHelper dbHelper;
29
30     public LoginDatabaseAdapter(Context _context) {
31         context = _context;
32         dbHelper = new DatabaseHelper(context, DATABASE_NAME, null,
33             DATABASE_VERSION);
34     }
35
36     public LoginDatabaseAdapter open() throws SQLException {
37         db = dbHelper.getWritableDatabase();
38         return this;
39     }
40
41     public void close() {
42         db.close();
43     }
44
45     public SQLiteDatabase getDatabaseInstance() {
46         return db;
47     }
48
49     public void insertEntry(String username, String password) {
50         ContentValues newValues = new ContentValues();

```

```
File Edit Selection View Go Run Terminal Help
LoginDatabaseAdapter.java - AndroidBloodBank - Visual Studio Code

J LoginDatabaseAdapter.java X
app > src > main > java > com > bloodbank > androidbloodbank > J LoginDatabaseAdapter.java
40 public void insertEntry(String userName, String password) {
41     ContentValues newValues = new ContentValues();
42     // Assign values for each row.
43     newValues.put("USERNAME", userName);
44     newValues.put("PASSWORD", password);
45
46     // Insert the row into your table
47     db.insert("LOGIN", null, newValues);
48     // Toast.makeText(context, "Reminder Is Successfully Saved",
49     // Toast.LENGTH_LONG).show();
50 }
51
52 public void insertEntryUSER(String userName, String password) {
53     ContentValues newValues = new ContentValues();
54     // Assign values for each row.
55     newValues.put("USERNAME", userName);
56     newValues.put("PASSWORD", password);
57
58     // Insert the row into your table
59     db.insert("LOGIN_USER", null, newValues);
60     // Toast.makeText(context, "Reminder Is Successfully Saved",
61     // Toast.LENGTH_LONG).show();
62 }
63
64 public int deleteEntry(String UserName) {
65     // String idString.valueOf(ID);
66     String where = "USERNAME=?";
67     int numberOfEntriesDeleted = db.delete("LOGIN", where,
68     new String[] { UserName });
69     // Toast.makeText(context,
70     // "Number fo Entry Deleted Successfully : " + numberOfEntriesDeleted,
71     // Toast.LENGTH_LONG).show();
72     return numberOfEntriesDeleted;
73 }
74
75 public String getSingleEntry(String userName) {
76     Cursor cursor = db.query("LOGIN", null, " USERNAME=?",
77     new String[] { userName }, null, null, null);
78     if (cursor.getCount() < 1) // Username Not Exist
79     {
80         cursor.close();
81         return "NOT EXIST";
82     }
83     cursor.moveToFirst();
84     String password = cursor.getString(cursor.getColumnIndex("PASSWORD"));
85     cursor.close();
86     return password;
87 }
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2
```

5.3 Registration

```
File Edit Selection View Go Run Terminal Help
Registration.java - AndroidBloodBank - Visual Studio Code

J Registration.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Registration.java
1 package com.bloodbank.androidbloodbank;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.view.View.OnClickListener;
8 import android.widget.AdapterView;
9 import android.widget.AdapterView.OnItemClickListener;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.Spinner;
13 import android.widget.Toast;
14
15 public class Registration extends Activity implements OnClickListener,
16     OnItemSelectedListener {
17
18     // Variable Declaration should be in onCreate()
19     private Button mSubmit;
20     private Button mCancel;
21
22     // private EditText mName;
23     // private EditText mEmail;
24
25     private EditText UserName;
26     public EditText password, cnfpass;
27     private EditText Email;
28
29     Spinner spinner;
30     LoginDatabaseAdapter loginDatabaseAdapter;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.registration);
36         // get Instance of Database Adapter
37         loginDatabaseAdapter = new LoginDatabaseAdapter(this);
38         loginDatabaseAdapter = loginDatabaseAdapter.open();
39
40         spinner = (Spinner) findViewById(R.id.spinner1);
41         Email = (EditText) findViewById(R.id.email);
42
43         // Assignment of UI fields to the variables
44         mSubmit = (Button) findViewById(R.id.register);
45         mSubmit.setOnClickListener(this);
46         mCancel = (Button) findViewById(R.id.tcancel);
47     }
48 }
```

```
File Edit Selection View Go Run Terminal Help
Registration.java - AndroidBloodBank - Visual Studio Code

J Registration.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Registration.java
46 mSubmit = (Button) findViewById(R.id.register);
47 mSubmit.setOnClickListener(this);
48
49 mCancel = (Button) findViewById(R.id.tcancel);
50 mCancel.setOnClickListener(this);
51
52 // mName = (EditText) findViewById(R.id.editText1);
53 // mEmail = (EditText) findViewById(R.id.editText2);
54 UserName = (EditText) findViewById(R.id.rollno);
55 password = (EditText) findViewById(R.id.password);
56 cnfpass = (EditText) findViewById(R.id.cnfpassword);
57
58 spinner.setOnItemSelectedListener(new OnItemSelectedListener() {
59     public void onItemSelected(AdapterView? arg0, View arg1,
60         int arg2, long arg3) {
61         /*
62          * Toast.makeText(getApplicationContext(),
63          * spinner.getSelectedItem().toString(),
64          * Toast.LENGTH_SHORT).show();
65          */
66         // String question = (String) spinner.getSelectedItem();
67     }
68
69     public void onNothingSelected(AdapterView? arg0) {
70     }
71 });
72
73 public void onClick(View v) {
74     switch (v.getId()) {
75     case R.id.tcancel:
76         Intent i = new Intent(getApplicationContext(), Login.class);
77         i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
78         startActivity(i);
79         finish();
80         break;
81     case R.id.register:
82         // String fname = mName.getText().toString();
83         // String email = mEmail.getText().toString();
84         String USERNAME = UserName.getText().toString();
85         String password = password.getText().toString();
86     }
87 }
```

```
File Edit Selection View Go Run Terminal Help
Registration.java - AndroidBloodBank - Visual Studio Code

J Registration.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Registration.java
140 switch (v.getId()) {
141
142
143 case R.id.tcancel:
144     Intent i = new Intent(getApplicationContext(), Login.class);
145     i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
146     startActivity(i);
147     finish();
148     break;
149
150 case R.id.tregister:
151     // String fname = mName.getText().toString();
152     // String email = mEmail.getText().toString();
153     String USERNAME = mName.getText().toString();
154     String passwordd = mPassword.getText().toString();
155     String question = (String) spinner.getSelectedItem();
156     String EMAIL = mEmail.getText().toString();
157
158     if (USERNAME.equals("")) {
159         Toast.makeText(getApplicationContext(),
160             "Please enter your UserName", Toast.LENGTH_SHORT)
161             .show();
162     }
163
164     else if (passwordd.equals("")) {
165         Toast.makeText(getApplicationContext(),
166             "Please Enter Password", Toast.LENGTH_SHORT).show();
167     }
168
169     else if (passwordd.getText().toString()
170         .equalsIgnoreCase(mPassword.getText().toString())) {
171         Toast.makeText(getApplicationContext(), "Password Mismatch",
172             Toast.LENGTH_SHORT).show();
173     }
174
175     else if (!(EMAIL.contains("@") && (EMAIL.contains(".")))) {
176         Toast.makeText(getApplicationContext(),
177             "Please Enter Valid Email Id", Toast.LENGTH_SHORT)
178             .show();
179     }
180
181     else {
182         loginDatabaseAdapter.insertEntry(USERNAME, passwordd);
183         Toast.makeText(getApplicationContext(),
184             "Account Successfully Created ", Toast.LENGTH_LONG)
185             .show();
186     }
187 }
188
189 public void onDestroy() {
190     super.onDestroy();
191     loginDatabaseAdapter.close();
192 }
193
194 public void onNothingSelected(AdapterView<?> arg0) {
195 }
196
197 @Override
198 public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2,
199     long arg3) {
200 }
201 }
```

```
File Edit Selection View Go Run Terminal Help
Registration.java - AndroidBloodBank - Visual Studio Code

J Registration.java X
app > src > main > java > com > bloodbank > androidbloodbank > J Registration.java
140 }
141
142 else if (!(EMAIL.contains("@") && (EMAIL.contains(".")))) {
143     Toast.makeText(getApplicationContext(),
144         "Please Enter Valid Email Id", Toast.LENGTH_SHORT)
145         .show();
146 }
147
148 else {
149     loginDatabaseAdapter.insertEntry(USERNAME, passwordd);
150     Toast.makeText(getApplicationContext(),
151         "Account Successfully Created ", Toast.LENGTH_LONG)
152         .show();
153
154     Intent i_register = new Intent(Registration.this, Login.class);
155     startActivity(i_register);
156     finish();
157 }
158 break;
159 }
160
161 public void onDestroy() {
162     super.onDestroy();
163     loginDatabaseAdapter.close();
164 }
165
166 public void onNothingSelected(AdapterView<?> arg0) {
167 }
168
169 @Override
170 public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2,
171     long arg3) {
172 }
173 }
```

5.4 Received Blood

```
File Edit Selection View Go Run Terminal Help
ReceiveBlood.java - AndroidBloodBank - Visual Studio Code

J ReceiveBlood.java X
app > src > main > java > com > bloodbank > androidbloodbank > J ReceiveBlood.java
1 package com.bloodbank.androidbloodbank;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import android.os.Bundle;
7 import android.support.v4.app.Fragment;
8 import android.text.Editable;
9 import android.text.TextWatcher;
10 import android.view.LayoutInflater;
11 import android.view.View;
12 import android.view.ViewGroup;
13 import android.widget.EditText;
14 import android.widget.ListView;
15
16 public class ReceiveBlood extends Fragment {
17     ArrayList<Contact> ImageArray = new ArrayList<Contact>();
18     Adapter ImageAdapter;
19     ListView dataList;
20     DatabaseHandler db;
21     ArrayList<Contact> ar;
22     EditText search_box;
23     View rootView;
24
25     @Override
26     public View onCreateView(LayoutInflater inflater, ViewGroup container,
27         Bundle savedInstanceState) {
28
29         rootView = inflater.inflate(R.layout.receive_blood, container, false);
30
31         dataList = (ListView) rootView.findViewById(R.id.listView);
32         search_box = (EditText) rootView.findViewById(R.id.search_box);
33         // showImage = (ImageView) findViewById(R.id.showImage);
34         /**
35          * create DatabaseHandler object
36          */
37         db = new DatabaseHandler(getActivity());
38         /**
39          * Reading and getting all records from database
40          */
41         List<Contact> contacts = db.getAllContacts();
42         for (Contact cn : contacts) {
43
44             // add contacts data in arraylist
45             ImageArray.add(cn);
46         }
47         /**
48          * Set Data base Item into listView
49          */
50     }
51 }
```

```
File Edit Selection View Go Run Terminal Help
ReceiveBlood.java - AndroidBloodBank - Visual Studio Code

J ReceiveBlood.java X
app > src > main > java > com > bloodbank > androidbloodbank > J ReceiveBlood.java
49
50     * Set Data base Item into listView
51     */
52     ImageAdapter = new Adapter(getActivity(), R.layout.row_design_,
53         ImageArray);
54     dataList.setAdapter(ImageAdapter);
55
56     search_box.addTextChangedListener(new TextWatcher() {
57
58         @Override
59         public void onTextChanged(CharSequence cs, int arg1, int arg2,
60             int arg3) {
61             ar = new ArrayList<Contact>();
62             System.out.println("cs" + cs.toString() + "CS Value" + cs);
63             for (int i = 0; i < ImageArray.size(); i++) {
64
65                 if (ImageArray.get(i).getAddress().contains(cs.toString())
66                     || ImageArray.get(i).getBloodGroup().
67                         contains(cs.toString())
68                     || ImageArray.get(i).getBloodGroup().toLowerCase()
69                         .contains(cs.toString())
70                     || ImageArray.get(i).getBloodGroup().toUpperCase()
71                         .contains(cs.toString())
72                     || ImageArray.get(i).getAddress().toLowerCase()
73                         .contains(cs.toString())
74                     || ImageArray.get(i).getAddress().toUpperCase()
75                         .contains(cs.toString())) {
76                     ar.add(ImageArray.get(i));
77                 }
78                 if (cs.equals("")) {
79                     ar = ImageArray;
80                 }
81             }
82             ImageAdapter = new Adapter(getActivity(), R.layout.row_design_,
83                 ar);
84             dataList.setAdapter(ImageAdapter);
85             // When user changed the Text
86             // MainActivity.this.adapter.getFilter().filter(cs);
87
88         }
89
90         @Override
91         public void beforeTextChanged(CharSequence arg0, int arg1,
92             int arg2, int arg3) {
93             // TODO Auto-generated method stub
94
95         }
96     })
97 }
```

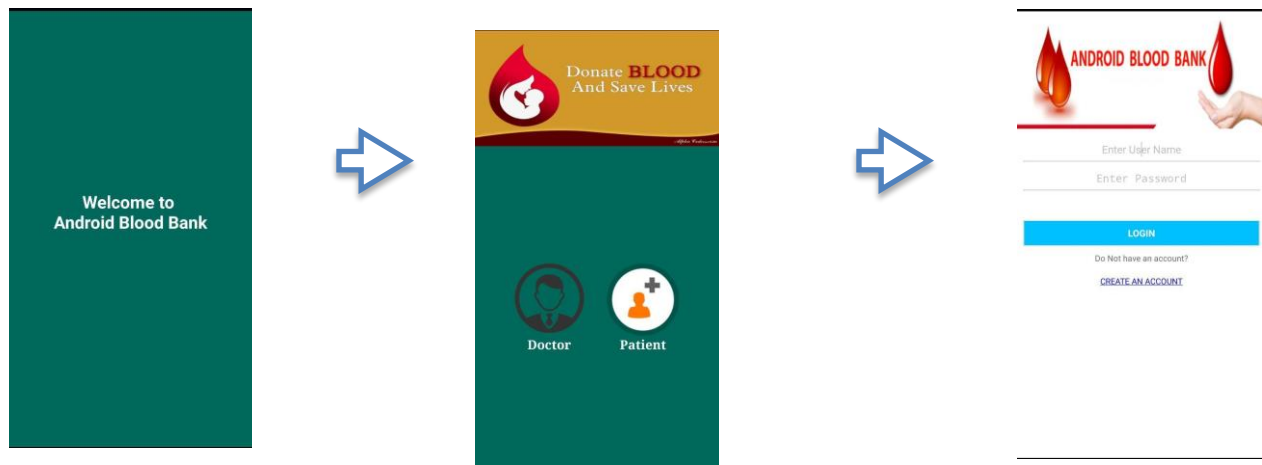
```
File Edit Selection View Go Run Terminal Help
ReceiveBlood.java - AndroidBloodBank - Visual Studio Code

J ReceiveBlood.java X
app > src > main > java > com > bloodbank > androidbloodbank > J ReceiveBlood.java
//
78     }
79     if (cs.equals("")) {
80         an = imageArray;
81     }
82 }
83 imageAdapter = new Adapter(getActivity(), R.layout.row_design_,
84     an);
85 dataList.setAdapter(imageAdapter);
86 // When user changed the text
87 // MainActivity.this.adapter.getFilter().filter(cs);
88
89 }
90
91 @Override
92 public void beforeTextChanged(CharSequence arg0, int arg1,
93     int arg2, int arg3) {
94     // TODO Auto-generated method stub
95 }
96
97 @Override
98 public void afterTextChanged(Editable arg0) {
99     // TODO Auto-generated method stub
100 }
101
102 });
103
104 return rootView;
105 }
106
107 @Override
108 public void setUserVisibleHint(boolean isVisibleToUser) {
109     // TODO Auto-generated method stub
110     if (rootView != null) {
111         imageAdapter = new Adapter(getActivity(), R.layout.row_design_,
112             imageArray);
113         dataList.setAdapter(imageAdapter);
114         System.out.println("Called");
115     }
116     System.out.println("Called else");
117     super.setUserVisibleHint(isVisibleToUser);
118 }
119
120 }
121
122 }
```

Chapter 6: Verification AND Validation

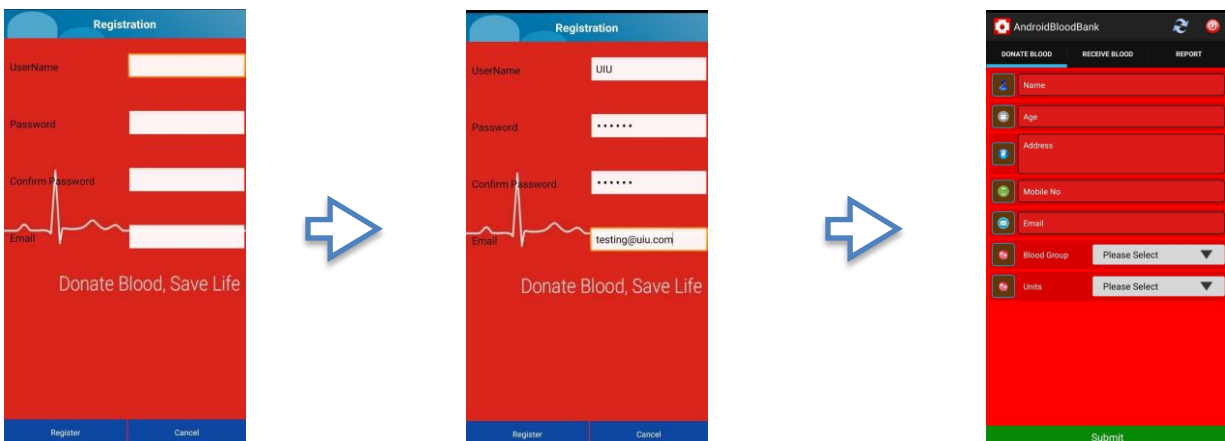
6.1 Login Test Case

Once the page for the Login is set up, we can test the running of the app on the Android Studio software on our computer. After the Login page is tested, then we can proceed to check the running of the rest of the application.



6.2 Validation Test Case

After the Login page is tested, we can proceed to further validate the data inputs and the correct display of the data entered in the database. Once that is done, we can proceed ahead with the final stages of the application and then publish it as a stand-alone app.



DRAWBACKS

Currently, there are not many apps available in Bangladesh that will help mitigate the demand and supply constraints of blood donors and blood requirements of hospitals. There are many situations where there is a need to get information fast and reliably so as to ensure safety of person in need. Due to lack of such services many times the person in need does not get blood on time and may succumb to his/her injuries. Certain services that are available are reachable by only big and well to do organizations due to high costs to maintain an application and a server and to set up relevant equipment.

This application aims to bring down the overall cost by providing a user friendly and cost efficient application without the need for huge investments and elaborate paraphernalia.

The issue that may arise could be due to hesitation of hospitals and big organizations to implement a simple and easy-to-use application due to red tape and self-profiting interests. It could pose a challenge to make them come around and see the greater good for the benefit of the public in general.

Bibliography

1. Abstract. (2021). *Vox Sanguinis*, 116(S1), 5–188. <https://doi.org/10.1111/vox.13117>
2. *Bangladesh is still to meet the demand of safe blood supply*. (2017). [Health]. <https://www.who.int/bangladesh/news/detail/14-06-2017-bangladesh-is-still-to-meet-the-demand-of-safe-blood-supply>
3. *Bangladesh Red Crescent Society*. (n.d.). [Health]. Bangladesh Red Crescent Society. <https://bdracs.org/>
4. *Blood Donation Bangladesh Welfare Foundation*. (n.d.). [Health]. Blood Donation Bangladesh Welfare Foundation. Retrieved October 5, 2023, from <https://www.bdbwf.com/>
5. Boehm, B. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), 14–24. <https://doi.org/10.1145/12944.12948>
6. *Global Blood Fund*. (n.d.). GBF. Retrieved October 5, 2023, from https://globalbloodfund.org/global-need/work/?gclid=CjwKCAjwge2iBhBBEiwAfXDDBR3cMYsIIDh892pJGUwxgWY5ZPYwUP1AI4v1LBqf7VuoqWxJx0RK5gxOCvOoQAvD_BwE
7. International Engineering Consortium. (n.d.). *IP Communications*. International Engineering Consortium. Retrieved October 5, 2023, from https://books.google.co.in/books/about/IP_Communications.html?id=APIEIAAACAAJ&redir_esc=y
8. Md. Rezaul Karim, Morshed Alam, & Md. Manzur Rahman Farazi. (2012). *Factors Influencing Blood Donation Behavior of University Level Students in Bangladesh*. 35–40.
9. Rummana Ferdous Fagun. (2022, September 8). Things to keep in mind while seeking blood donation. *The Business Standard*. <https://www.tbsnews.net/bangladesh/health/things-keep-mind-while-seeking-blood-donation-474218>
10. *Sera Doctor*. (n.d.). Sera Doctor. <https://seradoctor.com/service/blood-donors-club>
11. *Software development process*. (n.d.). https://en.wikipedia.org/wiki/Software_development_process