

BLOOD BANK
A Project Report Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF COMPUTER APPLICATION

By
ADITYA SINGH
University Roll No. 1702914002
Under the Supervision of
Dr. Sangeeta Arora
Assistant Professor
KIET Group of Institutions



Submitted to
DEPARTMENT OF COMPUTER APPLICATION
Affiliated to
DR. A. P. J ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW
JULY,2021

DECLARATION

I hereby declare that the work presented in this report entitled **“Blood Bank”**

Application was carried out by US. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution.

I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

ADITYA SINGH

University Roll No.-17029014002

TRAINING CERTIFICATE



THERE'S ALWAYS A WAY.

TRAINING LETTER

Project completion certificate

Dear Aditya,

Due to pandemic situation, treat it as formal completion project certificate.

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Aditya Singh, MCA Final year student of KIET, Ghaziabad has completed his project work in Virtual Touch Pro Tech Pvt. Ltd, Noida on "R&D and Data Analystist" under the guidance of Mr. Amit Sharma, Technical Consultant towards the fulfillment of the award of the degree "Master Of Computer Applications" during the period of 1st April 2021 to 30th June 2021.

Regards,
Rahul Tiwari

VIRTUAL TOUCH PRO TECH PVT. LTD.

Director/Authorised Signatory

VIRTUAL TOUCH PRO TECH PVT. LTD

Address :
D-340, First Floor, Sector 10
Noida - 201301 (U.P.)
Office :
0120-4338910

Email :
support@virtualtouchprotech.com
Website :
www.virtualtouchprotech.com

CIN : U72300UP2015PTC068572

CERTIFICATE

Certified that **ADITYA SINGH (Univ. Roll No.-1702914002)** have carried out the project work having “**Blood Bank**” for Master of Computer Application from Dr.A.P.J.Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Aditya Singh (Univ. Roll No -1702914002)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Dr. Sangeeta Arora
Assistant Professor
Department of Computer Application
KIET Group of Institutions, Ghaziabad

Signature of External Examiner

Signature of Internal Examiner

Dr. Ajay Kumar Shrivastava
Head, Department of Computer Application
KIET Group of Institutions, Ghaziabad

ABSTRACT

This system mainly reduces the work task and it is easy to maintain the records for a long time than normal hand written records as well give ease. The user can check his record details by just entering his demand no need to search all the record. So, the maintenance and Presidency of institute became very easy.

- Easy accessibility.
- It makes searching records easier and faster.
- User is no longer required to check his register in search of records, as now it can be searched over the software by choosing some options.
- The user need not to type in most of the information.
- On the whole it liberates the user from keeping lengthy manual records.
- Everyone wants his/her work to be done by computer automatically and displaying the result for further manipulations.
- So, this project is about providing convenience.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor **Dr. Sangeeta Arora**, for his guidance, help and encouragement throughout our project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professor and Head, Department of Computer Application, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Aditya Singh

Univ. Roll No. 1702914002

TABLE OF CONTENTS

Declaration	ii
Certificate	iii
Abstract	iv
Acknowledge	v
Table of content	vi
List of tables	viii
List of figures	viii
Chapter 1: Introduction	9-20
1.1 Project description	9
1.2 Project Scope	11
1.3 Identification of need	12
1.4 Problem Statement	13
1.5 Hardware/software used in project	14
1.6 Project schedule	18
1.6.1 Pert Chart	
1.6.2 Gantt Chart	
Chapter 2: Literature Review	21-26
Chapter 3: Feasibility Study	27-32
3.1 Introduction	27
3.2 Main Aspects	29
3.2.1 Technical feasibility	
3.2.2 Economical Feasibility	
3.2.3 Operational Feasibility	
3.3 Benefits	31
3.4 SRS	32

Chapter 4: Design	34-44
4.1 Introduction	34
4.2 Analysis	35
4.3 SDLC	37
4.4 Soft. Engg. Paradigm	39
4.4.1 Prototype model	
4.5 DFD	42
4.7 ER Diagram	44
 Chapter 5: Report	 45-51
5.1 Gist	45
5.2 Some Screenshots	46
 Chapter 6: Coding	 52-59
 Chapter 7: Testing	 60-62
7.1 Introduction	60
7.2 Level of testing	61
7.3 Some Important observations	62
7.4 test case result summary	62
 Chapter 8: Conclusion & Future scope	 64
8.1 Conclusion	64
8.2 Future Scope	64
 Bibliography	 65

LIST OF FIGURES

Figure 1.1 Pert chart.....	18
Figure 1.2: Gannt chart for project.....	19
Figure 2.1: Design.....	23
Figure 2.2: Flow of website	23
Figure 2.3: Some screens	24
Figure 4.1: Above image depicting the planning step.....	377
Figure 4.2: Prototype model	40
Figure 4.3: Apex Work	42
Figure 4.4 Context level.....	42
Figure 4.5 Dfd 1 level.....	43
Figure 4.6: Uml use case diagram.....	43
Figure 4.7:ER diagram of system	444
Figure 5.1 System	455
Figure 7.1: Testing pyramid.....	61

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

The population of the world is multiplying with each coming year and so are the diseases and health issues. With an increase in the population there is an increase in the need of blood. The growing population of the world results in a lot of potential blood donors. But in spite of this not more than 10% of the total world population participates in blood donation.

With the growing population and the advancement in medical science the demand for blood has also increased. Due to the lack of communication between the blood donors and the blood recipients, most of the patients in need of blood do not get blood on time and hence lose their lives. There is a dire need of synchronization between the blood donors and hospitals and the blood banks.

This improper management of blood leads to wastage of the available blood inventory. Improper communication and synchronization between the blood banks and hospitals leads to wastage of the blood available. These problems can be dealt with by automating the existing manual blood bank management system.

IA high-end, efficient, highly available and scalable system has to be developed to bridge the gap between the donors

This feature is the efforts required to search for blood donors. This project is developed to manage the blood stock in the blood bank and the blood prices are maintained in the database. New blood details are entered in to the project to manage blood details.

focused on making personalized contact with a customer for various campaigns designed by the marketing team. It also provides real-time sales alerts based on the level of engagement with a customer.

his feature helps in quickly finding experts who can help in closing a deal based on customer queries and feedback. In short, it helps in bringing in a collaborative effort to engage an entire team in the deal and make the deal happen.

It provides a metric-based goal setting, and also continuous feedback and rewards and recognition for the sales team. This helps in enhancing the performance of the sales team.

This feature initiates and tracks the leads that are in progress. It also helps in continually optimizing campaigns across every channel.

This feature helps in building a community with partners. It also helps in connecting directly with channel partners to share goals, objectives, and activities.

This is the mobile platform to carry out all the above activities on a mobile platform.

It is a visual design to automate the business processes. The interface provides simple drag and drop options to make this design. It helps in creating a flexible approval process with deal discounts and expense Presidency etc.

Salesforce can integrate to an existing email platform. This helps in providing flexibility to the existing team with no additional learning curve.

This feature provides the sales team the power to easily share various files, discuss them and update them as needed. Also receive alerts when something in the file changes.

Dashboards offer a real-time picture of the business at a glance. With this, anyone can create detailed reports which can be accessed from anywhere.

This feature helps in getting a real time view of the forecast of a sales team. It provides multi-currency support and an in-line editing mode to manage the sales forecast well.

This feature is used to create multiple territory models, preview them before rollout, and continually optimize and balance territories throughout the year.

1.2 PROJECT SCOPE

When I get the study, I knew that the existing manual system contains a lot of drawbacks like it take more manual effort and also it is very time consuming etc. So in the proposed system all the drawbacks of the system is get overcome and the work that is going over there is changed to computerized and this make the work more easy like the consent giving details and also stored the minimum requirements etc. can be stored in the data base and the checking is automatically done by the software itself and the details of the consent get filled. This increases the total productivity. The use of paper file is avoided and all the data are efficiently manipulated by the system.

It is basically for Institutes with the new trend of managing the stuffs.

- User view all blood bank information is location wise.
- It is very user friendly.
- Donor easily donate blood near blood bank location
- Accuracy in work.
- Patient easily request for blood near blood bank location.
- In this system also supported inquiry for user.
- Access to any information individually.
- Work becomes very speedy.
- Easy to update information
- Easy availability.

1.3 IDENTIFICATION OF NEED

User need identification and analysis are concerned with what user needs rather than what he/she wants. Not until the problem has been identified, defined, and evaluated should the analyst think about solutions and whether the problem is worth working. This step intended to help the user and analyst understand the real problem rather than its symptoms. The user or the analyst may identify the need for a candidate system or for enhancement in the existing system.

An analyst is responsible for performing following tasks:

- Studied strength and weakness of the current system.
- Determined “what” must be done to solve the problem.
- Prepared a functional specifications document.

Also provide availability 24*7.

1.4 PROBLEM STATEMENT

In the existing system all the work is done manually. This is chance of committing errors and it will take more time to perform or checkout any information. There are so many limitations in the existing system. So, the existing system should be automized. If the system is carried over manually, for everything it takes more time. So, it is difficult to take immediate decisions.

- In the traditional system, if you wish to analyze any record you have to turn pages many times.
- Existing systems are time consuming as it requires too much planning and so much human involvement.
- As it involves much human involvement, the cost of the system automatically gets increased.
- Existing systems require paper use, which isn't good for the environment.
- With too much human involvement, there are high chances of risk as well.
- There is too much of paper work too, which makes the tasks in the existing system, very tedious.

1.5 HARDWARE / SOFTWARE USED IN PROJECT

1.5.1 HARDWARE REQUIREMENT

Hardware	Configuration
Processor	Above 4.4
Ram	4GB
Phone	Android

1.5.2 SOFTWARE REQUIREMENT

Software	Configuration
Database	Firebase
Language	JAVA

1.5.3 SOME REQUIREMENTS

Performance Requirements:

To achieve good performance the following requirements must be satisfied

- Scalability: The ease with which a system or component can be modified to fit the problem area.
- Portability: The ease with which a system or component can be transferred from one hardware or software environment to another.
- Security: It is the ideal state where all information can be communicated across the internet / company secure from unauthorized persons being able to read it and/or manipulate it.
- Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.
- Reliability: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- Reusability: The degree to which a software module or other work product can be used in more than one computing program or software system.

Safety Requirements:

In case scenarios where data integrity can be compromised, measures should be taken to ensure that all changes are made before system is shutdown. The user must have a registered account to use all facility of the application.

1.5.4 OTHER REQUIREMENTS

1.5.4.1 ANDROID

Android is a software platform and operating system for mobile devices. Android is available as open source. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries.

Android SDK was released by Open Handset Alliance in the month of November of the year 2007. Android was actually developed using the kernel of Linux 2.6 and the highlighting features of Android include the relationship (CRM) suite offering applications for small, midsize and enterprise organizations, with a focus on sales and support. Android ships with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts and others. All applications are built using the Java. Each of the applications aims at performing a specific task that it is actually intended to do

Applications Layer

Android ships with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts and others. All applications are built using the Java. Each of the applications aims at performing a specific task that it is actually intended to do.

Application Framework Layer

The next layer is the application framework. This includes the programs that manage the phone's basic functions like resource allocation, telephone applications, switching between processes or programs and keeping track of the phone's physical location. Application developers have full access to Android's application framework. This allows them to take advantage of Android's processing capabilities and support features when building an Android application. We can think of the application framework as a set of basic tools with which a developer can build much more complex tools.

Libraries Layer

The next layer contains the native libraries of Android. These shared libraries are all written in C or C++, compiled for the particular hardware architecture used by the phone and preinstalled by the phone vendor

Android Runtime Layer

Android Runtime layer includes Dalvik Virtual Machine (DVM) and a set of core java libraries. Every Android app gets its own instance of DVM. Dalvik has been written so that a device can run multiple virtual machines efficiently and it executes files with. dex (Dalvik Executable Format) extension optimized for minimum memory

1.5.4.2 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM), formerly known as Google Cloud Messaging (GCM), is a cross-platform cloud solution for messages and notifications for Android, iOS, and web applications, which as of 2021 can be used at no cost. Firebase Cloud Messaging allows third-party application developers to send notifications or messages from servers hosted by FCM to users of the platform or end users. The service is provided by Firebase, a subsidiary of Google. On October 21, 2014, Firebase announced it had been acquired by Google for an undisclosed amount. The official Google Cloud Messaging website points to Firebase Cloud Messaging (FCM) as the new version of GCM. Firebase is a mobile platform which supports users in developing mobile and web applications. Firebase Cloud Messaging is one of many products which are part of the Firebase platform. On the platform users can integrate and combine different Firebase features in both web and mobile applications. is strongly typed object-oriented, on-demand programming language Integrated

- Easy to use
- Data Focused
- Hosted
- Multitenant aware
- Easy to test
- Versioned
- Object-Oriented

1.6 PROJECT SCHEDULE

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, costs and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempt to define “best case” and “worst case” scenarios so that project outcomes can be bounded.

The first activity in software project planning is the determination of software scope. Function and performance allocated to software during system engineering should be assessed to establish a project scope that is ambiguous and understandable at Presidency and technical levels. Software scope describes function, performance, constraints, interfaces and reliability.

During early stages of project planning, a microscopic schedule is developed. This type of schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into detailed schedule. Here specific software tasks are identified and scheduled.

Scheduling has following principles:

1. Compartmentalization: the project must be compartmentalized into a number of manageable activities and tasks.
2. Interdependency: the interdependencies of each compartmentalized activity or tasks must be determined.
3. Time allocation: each task to be scheduled must be allocated some number of work units.
4. Effort validation: every project has a defined number of staff members.
5. Defined responsibilities: every task that is scheduled should be assigned to a specific team member.
6. Defined outcomes: every task that is scheduled should have a defined outcome.

1.6.1 Pert chart

Program evaluation and review technique (pert) is a project scheduling method that is applied to software development.

Pert provide quantitative tool that allow the software planner to-Determine the critical path-the chain of tasks that determines the duration of the project; Establish “most likely” time estimates for individual tasks by applying statistical models; and

Calculate “boundary times” that defines a time “window” for a particular task.

Pert chart (program evolution review technique) for project-

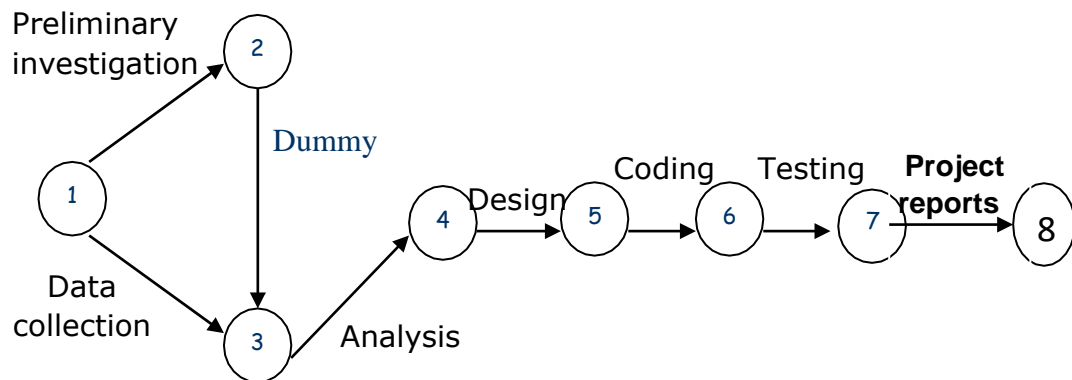


Figure 1.1 Pert chart

1.6.2 Gantt Chart

When creating a project schedule, the planner begins with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network. Effort, duration and start dates are input are each task network. As a consequence of this input, a timeline chart also called a Gantt chart is generated. A timeline chart is developed for entire project.

Gantt chart for project:

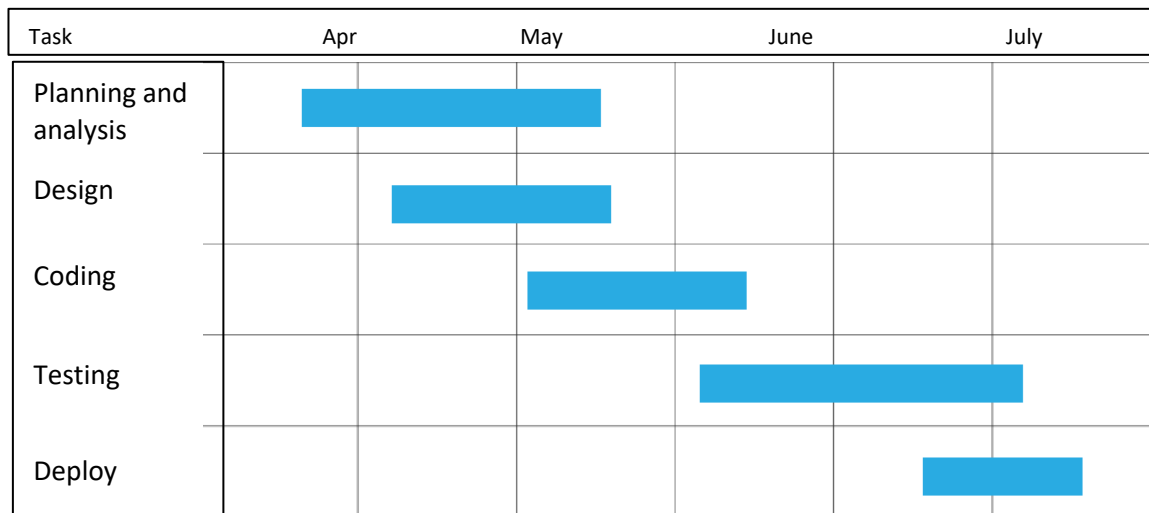


Figure 1.2:Gannt chart for project

Here horizontal bars indicate the duration of each task.

CHAPTER 2

LITERATURE REVIEW

ABSTRACT

Blood is a saver of all existing lives in case of emergency needs. The task of blood bank is to receive blood.

from various donors, to monitor the blood groups database and to send the required blood during the need to the hospital in case of emergencies. The problem is not insufficient number of donors, but finding a willing donor at the right time.

We want to build a network of people who can help each other during an emergency. This application timely updates the information regarding the donors where the administrator accesses the whole information about blood bank management system. Donor will be prompted to enter an individual's details, like name, phone number, and blood

group. In the urgent time of a blood requirement, you can quickly check for blood banks or hospitals matching a particular or related blood group and reach out to them through the App. Blood bank App provides list of blood banks

in your area. A large number of blood donors are attracted using an Android application. Since almost everyone carries.

KEYWORDS: - Blood bank, Android, Blood transfusion, Database, Donors, Acceptors, Administrator,

2.1 INTRODUCTION

The blood is specialized bodily fluid that delivers necessary substances to the body's cells such as nutrients and oxygen. Blood banking is a cache or bank of blood or blood components, gathered as a result of blood donation, stored and preserved for later use in blood transfusions. In addition to this, the blood type of patients also needs to be determined for compatibility's sake for a blood transfusion. It is possible in some situations that the patient is unable to get the required amount of blood at right time due to lack of interrelationship in form of a networked database among the blood banks which leads to the lack of knowledge of updated record of all blood donors. Today mobile and mobile based applications have become a part of our day-to-day life. With the revolution in mobile computing many great features were added to the field and the mobiles got smaller, faster and better as the decade passed. This Android application is developed to easily search for blood in nearby areas for emergency. In this Android app one will get clear access to blood in real time

2.2 RELATED WORK

2.2.1 College Management System

It's a system which manages the information of college, information of student, information of placement, various event going in college. It keeps tracking record of all the data. It has also a notice board which have data about various programmer either its cultural or any sports which is supposed to be held soon or technical.

2.2.2 The Optimization of Blood Donor Information and Management System by Technopedia.

Blood is a saver of all existing lives in case of emergency needs. P. Priya, V. Saranya, S. Shabana, Kavitha Subramani has proposed an extended web application to timely update the information regarding the donors, acceptor and patients where the administrator access the whole information about blood bank management system. Also, the proposed work has security, to protect the contact details of the donors in web application where it can be misused by third parties. It also maintains the amount of each available blood groups, if the stock of a particular blood group is lower than the required amount then the proposed method notifies the donor to donate blood addition to web application, an android mobile application

2.2.3 A Survey Paper on E-Blood Bank and an Idea to use on Smartphone

Institute Management System Blood is an important aspect for all living things. It proves to be a lifesaving component in case of emergency requirement. None of the online blood bank offers the direct contact between donor and blood bank.

This is the major drawback of the existing system. Existing systems are time consuming; require more manpower and it is costly.

2.2.4 Blood Bank Management Information System in India

Vikas Kulshreshtha Research Scholar, Dr. Sharad Maheshwari has introduced the review of the main features, merits and demerits provided by the existing Web-Based Information System for Blood Banks. Blood is universally recognized as the most precious element that sustains life. It saves innumerable lives across the world in a variety of conditions. A blood bank is a place designed especially for the storage of blood and blood products. The term „blood bank” typically refers to a division of a hospital laboratory where the storage of blood product occurs and where proper testing is performed to reduce the risk of transfusion related events.

2.2.5 Blood Connect

Blood cannot be manufactured in factories; it can only come from generous donors. To cater to this demand, Blood Connect was launched on 1st April, 2010 (as a project under NSS IIT Delhi) with an unparalleled objective of solving the problem of blood shortage India. According to WHO data, India faces a shortage of 3million blood units. This shortage can easily be eliminated if only an additional 2% of India's youth donates blood. To make this possible, Blood Connect acts as a channel connecting voluntary blood donors with those who need blood.

2.2.6 E- Blood Bank

This application helps you find people donating blood in your area. You can contact them through phone number or email address. You can see the location of user in map and if register yourself with App, you can get push notification in case your blood group matches with the need of blood. You can find nearby hospitals and access them

2.3.2 PROPOSED SYSTEM

the application. After downloading the application, registration form appears. For registration basic details like name, address, contact, date of birth, blood group, email id, emergency contact etc. are needed. If the person has already registered, then he/she has to login. If he does not remember his password, he can click on „Forgot password“ button which sends SMS containing

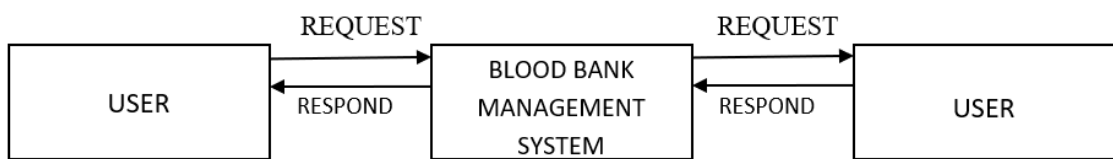
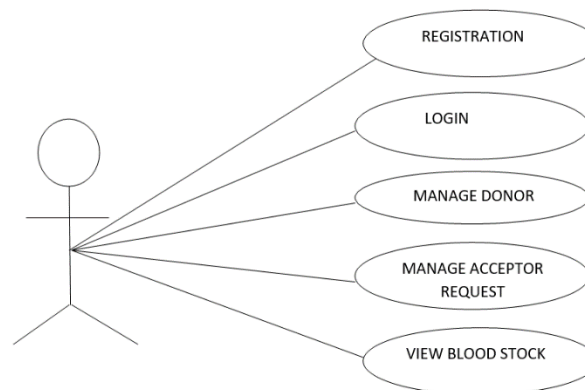


Figure 2.2 flow of website

2.3 DESIGN

2.3.1 Design

Figure 2.3 Design



2.3.2.1 JAVA

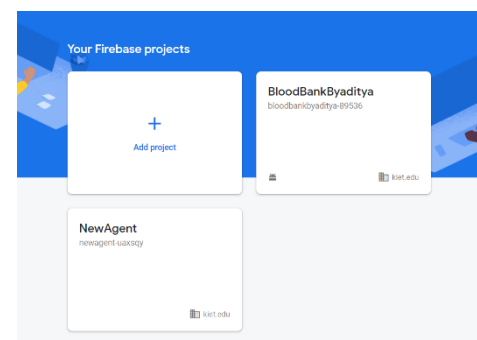
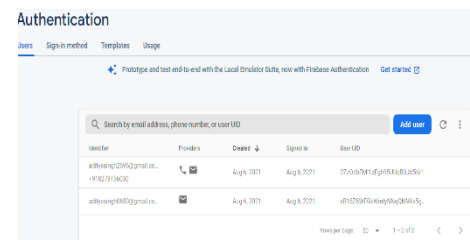
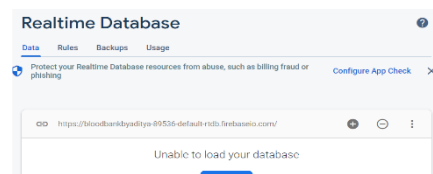
Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Over time new enhanced versions of Java have been released. The current version of Java is Java 1.7 which is also known as Java 7. From the Java programming language, the Java platform evolved. The Java platform allows software developers to write program code in other languages than the Java programming language and still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

2.3.2.2 Android

Android is a software platform and operating system for mobile devices. Android is available as open source. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries. Android SDK was released by Open Handset Alliance in the month of November of the year 2007. Android was actually developed using the kernel of Linux 2.6 and the highlighting features of Android include the No fees for licensing, distribution and release approval. GSM, 3G EDGE networks for telephony. IPC message passing. Background processes and applications. Complete multimedia hardware control. APIs for location-based services such as GPS.

2.3.3 Firebase

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development. user interface is one important factor by which the user caneasily access the website to understandwhat he needs Some of the GUI are as follow:



2.4 RESULT

We have come on result that our system working fine in each case. It is user friendly as well as efficient to use. we have done different things to verify the performance.

TEST CASE RESULT

TestCase#	Description	Result
TC#1	Loading the	Passed
TC#2	Login	Passed
TC#3	Validating	Passed
TC#4	Content	Passed
TC#5	Course page	Passed
TC#6	Reports page	Passed
TC#7	Logout	Passed

2.5 DISCUSSION

Blood bank management system is to maintained all the information o blood donors, different blood groups available in each and help them manage in best way. Blood bank management system can help to collect blood from many donors from various sources and distribute the blood to needy people who require blood. This system have many facilities like online transfer blood from one blood bank to another. This project can manage list of donors who are eligible for donation on a particular date with contact number. This system is an blood bank management system that helps in managing various blood bank operations effectively The purpose of the operational feasibility is to determine whether the new system will be used if it is developed and implemented and whether there the possible application benefits.

2.6 CONCLUSION & FUTURE SCOPE

2.6.1 CONCLUSION

We hereby conclude by proposing an efficient and reliable android blood bank application. The donor will get himself registered through this improved system

2.6.2 FUTURE SCOPE

We can give more advance software for Blood Bank Management System including more facilities. We can add printer in future.

2.7 REFERENCES

- [1] Vikas Kulshreshtha and Sharad Maheshwari, "Benefits of Management Information System in Blood Bank", International Journal of Engineering and Science, Vol. 1, Issue 12, PP 05-07, 2012.
- [2] Hayes, Helen and Onkar Sharma, "A decade of experience with a common first year program for computer science, information systems and information technology majors". Journal of Computing Sciences in Colleges, Vol. 18, No. 3 pp. 217–227, 2003,
- [3] Polack, Jennifer, "Planning a CIS Education Within a CS Framework". Journal of Computing Sciences in Colleges, Vol. 25, No. 2, pp. 100–106, 2009.
- [4] J. Scott Armstrong, "The Value of Formal Planning for Strategic Decisions: A Reply". Strategic Management Journal, Vol. 7, pp. 183–185, 1986. L. Long, college management System, Prentice Hall, 1989, pp.116-117.

- [5] Sayable Dhond, Pandya Randhawa, Bhagya Hali Munde, Raj Nandini Patil, and Vikas Patil, "Android Based Health Application in Cloud Computing For Blood Bank", International Engineering Research Journal (IERJ) Volume 1 Issue, 2015. Institute Management System (Ims) Ijarce, Mustafiz Sharique pp 1-6
- [6] A Research Paper On College Management System Lalit Mohan Joshi M.Tech Scholar.
- [7] IJCSMC, Vol. 7, Issue. 7, July 2018, pg.20 – 31 Design and Implement a Novel Student Information Management System – Case Study Ihsan Ali Hassan
- [8] Shimozono K., Itsukiy M., Harasaka Y. and Furukawa Z., (2010), "User Management in an Educational Computer System: Personal Information Management," Fukuoka, Japan, pp. 1-6.
- [9] Rakesh kumar, sonu, Agarwal, pragya -Extremely Effective Crm Solution Using Salesforce, JETIR,
- [10] Norasiah M. A. and Norhayati A., (2003), "Intelligent Student Information System," in 4th National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia, pp. 212-215
- [11] Premuda, B.G., "Application of Spatial Decision Support System to Blood Bank Information Systems", International Journal of Geoinformatics, Vol.6, No. 2,
- [12] The optimization of Blood Donor Information and Management System by Technopedia," International Journal of Innovative Research in Science, Engineering and Technology, Volume 3, Special Issue 1, 2014.

CHAPTER 3

FEASIBILITY STUDY

3.1 INTRODUCTION

Feasibility of the system in an important aspect, which is to be considered. The system needs to satisfy the law of economic, which states that the maximum output should be yielded in minimum available resources.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

4. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility

studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

5. Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, et

3.2 MAIN ASPECTS

There are three aspects of feasibility to be considered namely.

1. Technical
2. Operational
3. Economical

TECHNICAL:

In the technical aspects one may consider the hardware equipment for the installation of the software. The system being centralized will required very little hardware appliances. Hence this helps the system to work smoothly with limited number of working capitals.

OPERATIONAL:

In the operational aspects may think of the benefits of the workload that many a personal may have to share. This is eased out and the required output may be retrieved in a very short time. Thus there is accuracy in the work on time is also saved there will be very little work that needs to be performed.

ECONOMICAL:

Economical system is definitely feasible because the hardware requirement is less and the operational working for the system requires a smaller number of recruits. This help introduction over-staffing and wastage funds.

We studied on the position to evaluate solution. Most important factors in this study were tending to overlook the confusion inherent in system Development the constraints and the assumed studies. It can be started that it the feasibility study is to serve as a decision document it must answer three key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

On these questions it can be explained that feasibility study of the system includes following different angles.

3.2.1 Technical feasibility:

This centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed additional equipment .in this stage of study, we have collected information about technical tools available by which I could decide my system design as the technical requirements.

3.2.2 Operational Feasibility:

In this stage of study we have checked the staff availability. I concentrate on knowledge of end users that are going to use the system. This is also called as behavioral feasibility in which I have studied on following aspects; people are inherently resistant to change, and computers have been known to facilitate change .An estimate has been made to how strong a reaction the user staff is having toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover. I had explained that there is need to educate and train the staff on new ways of conducting business.

3.2.3 Economical feasibility:

Economical analysis is the most frequently used method for evaluating the effectiveness of candidate system. More commonly known as cost\benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs. The decision was to design and implement system because it is for having chanced to be approved. This is an ongoing effort that improves the accuracy at each phase of the system life cycle.

In developing cost estimates for a system, I need to consider several cost elements. Among these is hardware personal facility. Operating and supply costs.

3.3 BENEFITS

Benefits of conducting a feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a “go/no-go” decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project
- Identifies reasons not to proceed

3.4 SYSTEM REQUIREMENT SPECIFICATION

Any system can be designed after specifies the requirement of the user about that system. For this first of all gathered information from user by the preliminary investigation which is starting investigation about user requirement..

The data that the analysts collect during preliminary investigation are gathered through the various preliminary methods.

Documents Reviewing Organization

The analysts conducting the investigation first learn the organization involved in, or affected by the project. Analysts can get some details by examining organization charts and studying written operating procedures.

Collected data is usually of the current operating procedure:

- The information relating to clients, projects and students and the relationship between them was held manually.
- Managing of follow-ups was through manual forms.
- Complaints require another tedious work to maintain and solve.
- Payments details had to be maintained differently.

Gathering Information By Asking Questions

Interviewing is the most commonly used techniques in analysis. It is always necessary first to approach someone and ask them what their problems are, and later to discuss with them the result of your analysis.

Questionnaires

Questionnaires provide an alternative to interviews for finding out information about a system. Questionnaires are made up of questions about information sought by analyst. The questionnaire is then sent to the user, and the analyst analyzes the replies.

Electronic Data Gathering

Electronic communication systems are increasingly being used to gather information. Thus it is possible to use electronic mail to broadcast a question to a number of users in an organization to obtain their viewpoint on a particular issue.

In my project, with the help of Marg software solutions, I have send questionnaire through electronic mail to twenty employees of the company and retrieved the information regarding the problem faced by existing system.

Interviews

Interview allows the analysts to learn more about the nature of the project request and reason of submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally or technically.

One of the most important points about interviewing is that what question you need to ask.

It is often convenient to make a distinction between three kinds of question that is

- Open questions
- Closed question
- Probes

Open questions are general question that establish a person's view point on a particular subject.

Closed questions are specific and usually require a specific answer.

Probes are question that follow up an earlier answer.

CHAPTER 4

DESIGN

4.1 INTRODUCTION

System is created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities.

Since a new system is to be developed, the one most important phases of software development life cycle are system requirement gathering and analysis. Analysis is a detailed study of various operations performed by a system and their relationship within and outside the system. Using the following steps, it becomes easy to draw the exact boundary of the new system under consideration.

All procedures, requirements must be analyses and documented in the form of detailed DFDs, logical data structure and miniature specifications.

System analyses also include sub-dividing of complex process involving the entire system, identification of data store and manual processes.

4.2 SYSTEM DESIGN

System design is the process of planning a new system or to replace the existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product.

System design phase follows system analysis phase. Design is concerned with identifying functions, data streams among those functions, maintaining a record of the design decisions and providing a blueprint the implementation phase.

Design is the bridge between system analysis and system implementation. Some of the essential fundamental concepts involved in the design of application software are:

- Abstraction
- Modularity
- Verification

Abstraction is used to construct solutions to problem without having to take account of the intricate details of the various component sub problems. Abstraction allows system designer to make step-wise refinement, which at each stage of the design may hide, unnecessary details associated with representation or implementation from the surrounding environment.

Modularity is concerned with decomposing of main module into well-defined manageable units with well-defined interfaces among the units. This enhances design clarity, which in turn eases implementation, Debugging, Testing, Documenting and Maintenance of the software product. Modularity viewed in this sense is a vital tool in the construction of large software projects.

Verification is fundamental concept in software design. A design is verifiable if it can be demonstrated that the design will result in implementation that satisfies the customer's requirements. Verification is of two types namely.

- Verification that the software requirements analysis satisfies the customer's needs.
- Verification that the design satisfies the requirement analysis.

Some of the important factors of quality that are to be considered in the design of application software are:

Reliability:

The software should behave strictly according to the original specification and should function smoothly under normal conditions.

Extensibility:

The software should be capable of adapting easily to changes in the specification.

Reusability:

The software should be developed using a modular approach, which permits modules to be reused by other application, if possible.

The System Design briefly describes the concept of system design and it contains four sections. The first section briefly describes the features that the system is going to provide to the user and the outputs that the proposed system is going to offer.

The second section namely Logical Design describes the Data Flow Diagrams, which show clearly the data movements, the processes and the data sources, and sinks, E-R diagrams which represent the overall logical design of the database, and high-level process structure of the system.

Preliminary Design:

Preliminary design is basically concerned with deriving an overall picture of the system. Deriving entire system into modules and sub-modules while keeping Cohesion and Coupling factors in mind. Tools, which assist in preliminary design process, are Data Flow Diagrams.

Code design:

The purpose of code is to facilitate the identification and retrieval for items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or attribute. To achieve unique identification there must be only one place where the identified entity or the attribute can be entered in the code; conversely there must be a place in the code for every thing that is to be identified. This mutually exclusive feature must be built into any coding system.

The codes for this system are designed with two features in mind. Optimum human oriented use and machine efficiency They are also operable i.e., they are adequate for present and anticipate data processing both for machine and human use.

Input /Output design:

is a part of overall system design, which requires very careful attention The main objectives of input design are:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable to and understood by the user staff.

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also to provide a permanent hard copy of these results for later consultation.

The various types of outputs are required by this system are given below:

- External outputs, whose destination is outside the concern and which require special attention because they, project the image of the concern.
- Internal outputs, whose destination is within the concern and which require careful design because they are the user's main interface within the computer.
- Operation outputs, whose use is purely within the computer department, E.g., program listings, usage statistics etc,

4.3 SDLC

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.



Figure 4.1: Above image depicting the planning step

SDLC Phases

Given below are the various phases:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only. Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Deployment

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

4.4 SOFTWARE ENGG. PARADIGM APPLIED

Software engineering is a layered technology. The foundation for software engineering is the process layer. Software engineering processes the glue that holds the technology layers together and enables ratios and timely development of computer software. Process defines a framework for a set of key process areas that must be established for effective delivery of software engineering technology.

Software engineering methods provide the technical how-to's for building software. Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing and support. Software engineering tools provide automated or semi-automated support for the process and the methods. When tools are integrated so that information created by one tool can be used by another tool, a system for the support of software development, called computer-aided software engineering is established.

The following paradigms are available:

1. The Waterfall Model
 2. The Prototyping Model
 3. The Spiral model
- Etc.

4.4.1 The Prototype models

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possibly exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

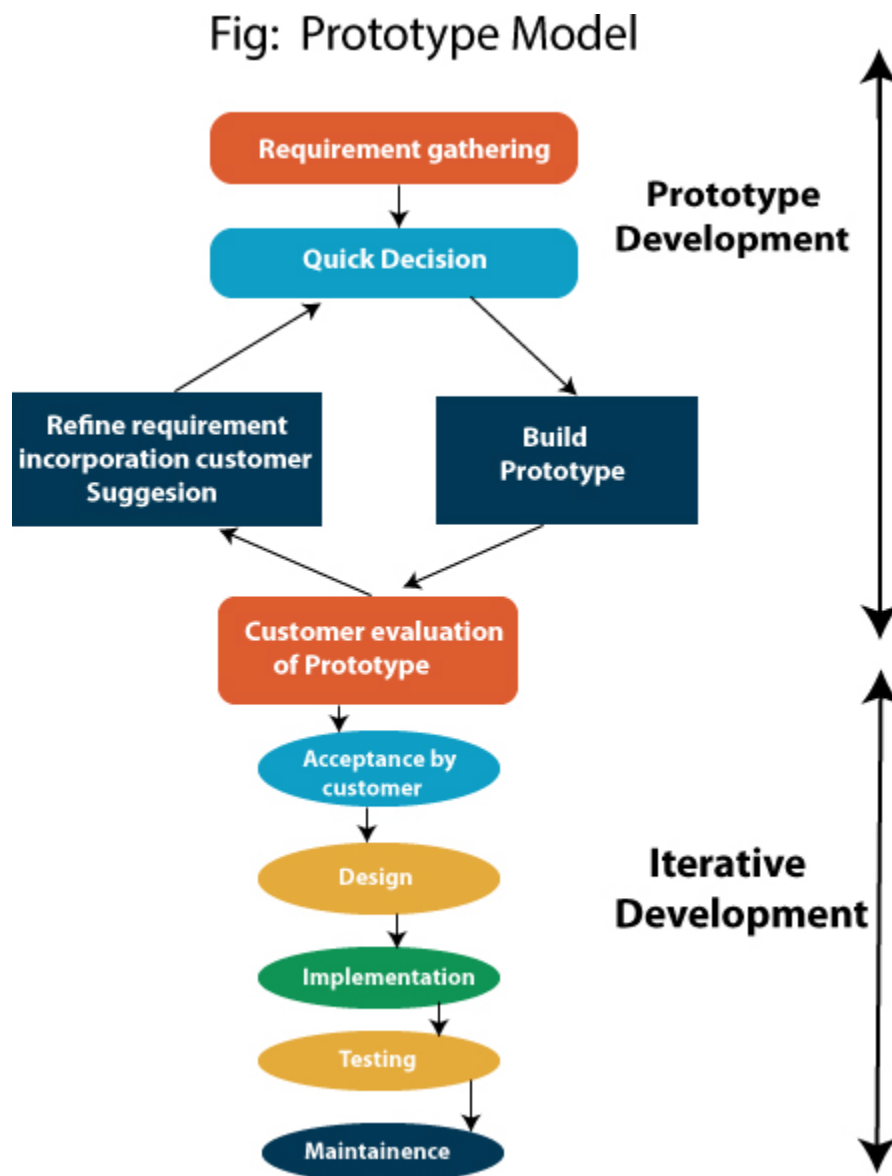


Figure 4.2: Prototype model

4.4.1.1 Advantage of Prototype Model

1. Reduce the risk of incorrect user requirement
2. Good where requirement are changing/uncommitted
3. Regular visible process aids Presidency
4. Support early product marketing
5. Reduce Maintenance cost.
6. Errors can be detected much earlier as the system is made side by side.

4.4.1.2 Disadvantage of Prototype Model

1. An unstable/badly implemented prototype often becomes the final product.
2. Require extensive customer collaboration
 - Costs customer money
 - Needs committed customer
 - Difficult to finish if customer withdraw
 - May be too customer specific, no broad market
3. Difficult to know how long the project will last.
4. Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.
5. Prototyping tools are expensive.
6. Special tools & techniques are required to build a prototype.
7. It is a time-consuming process.

4.5 DFD

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways.

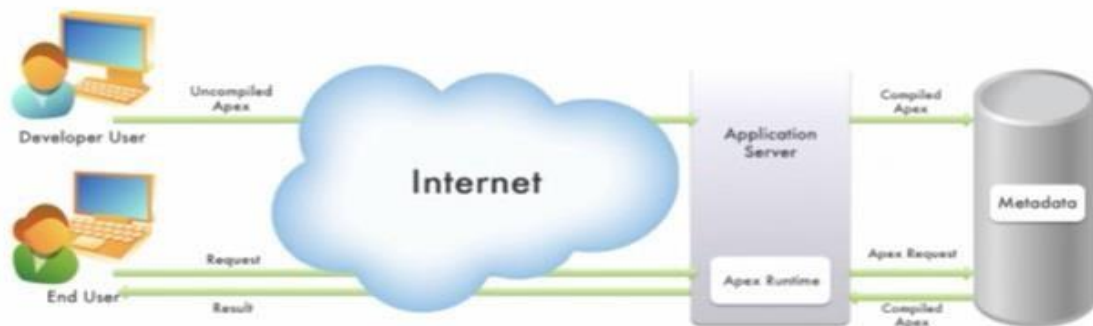


Figure 4.3 Apexworking

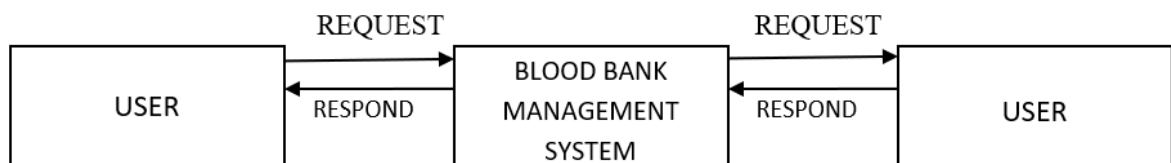


Figure 4.4 Context level

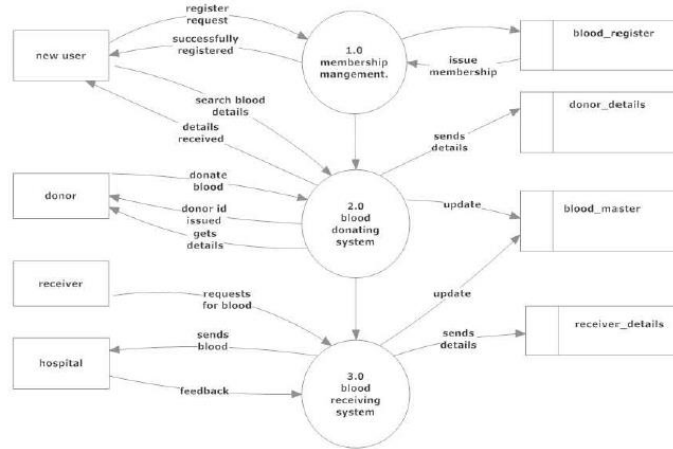


Figure 4.5 Dfd 1 level

4.6 UML use case diagram

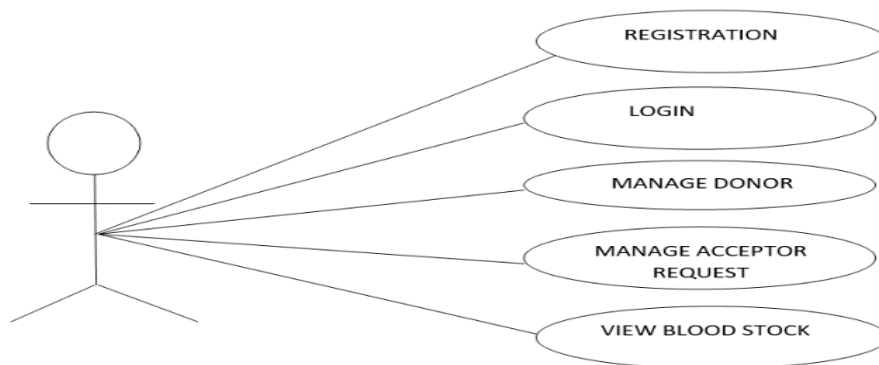


Figure 4.6 Uml use case

4.7 ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship

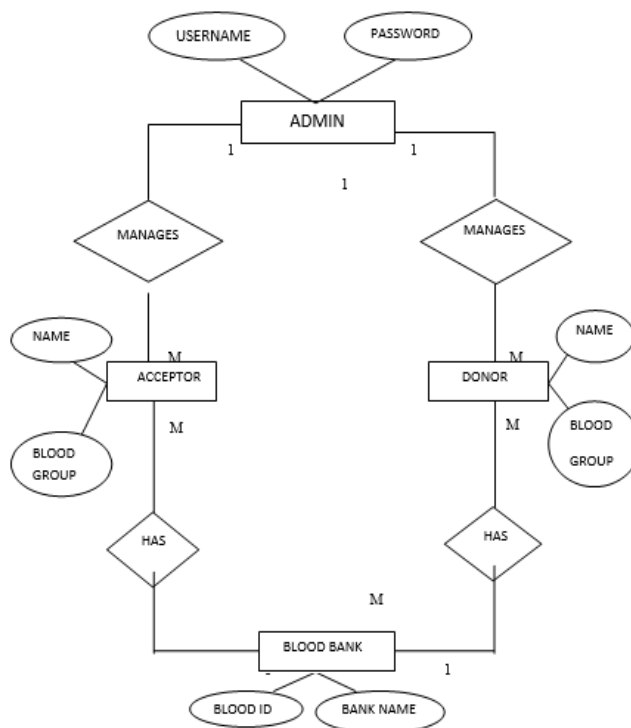


Figure 4.7:ER diagram of system

4.7.1 ER- Diagram Notations

ER- Diagram is a visual representation of data that describe how data is related to each other.

- **Rectangles:** This symbol represent entity types
- **Ellipses:** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes

CHAPTER 5

REPORT

5.1 GIST

The diagram **figure 5.1**, depicting our system.
We have designed and developed an easy, Useful, reliable system.

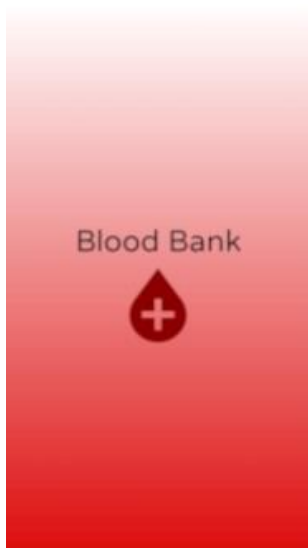


Figure 5.1 System

This gives a high level view of the system with the main components and the services they provide and how they communicate. It consists of the general graphical user interface facilities.

5.2 SOME SCREENSHOTS

BLOOD BANK



Main Activity



Register Screen



Address Details

Address Details

State

District

Tahsil

Village

NEXT

State & village, donata blood.

Cell & Blood Details

Cell & Blood Details

Mobile Number

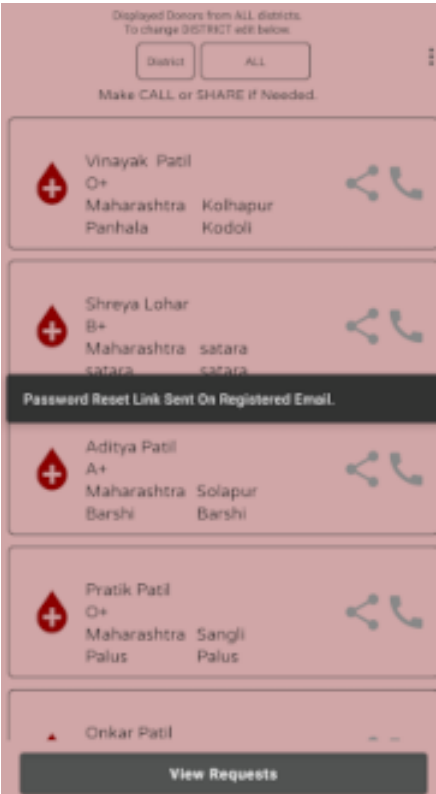
Blood Group

OTP

Verify

Blood is Life, Please do (D)

Password Reset



Request Blood



View Doeors




Firestore



Authentication

Authentication


[Users](#)[Sign-in method](#)[Templates](#)[Usage](#)


 Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication





[Get started](#)



Search by email address, phone number, or user UID

Add user






Identifier	Providers	Created 	Signed In	User UID
adityasingh2595@gmail.co... +918273456030	 	Aug 6, 2021	Aug 6, 2021	2Zz0dhFMI1aEgh95UUqB0Jtt5Kr1
adityasingh0680@gmail.co...		Aug 6, 2021	Aug 6, 2021	xB16Z6WFGkWmlyNNqQNMIlv5g...

Rows per page: 50 1 - 2 of 2  


Realtime Database


Realtime Database

[Data](#)[Rules](#)[Backups](#)[Usage](#)




 Protect your Realtime Database resources from abuse, such as billing fraud or phishing

[Configure App Check](#)





https://bloodbankbyaditya-89536-default-rtdb.firebaseio.com/

Unable to load your database

[Try again](#)

CHAPTER 6

CODING

This chapter contains some codes of the project. The goal of the coding is to translate the design of the system into code in a given programming language. For a given design, the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly.

Some Codes are as Written below:

```
Public class Contacts Today Controller {

public class DispalyRequestsActivity extends AppCompatActivity {

RecyclerView list;
ArrayList<User>requests,temp;
UserAdapter adapter;
EditText districtFilter;
User self;
String uid = FirebaseAuth.getInstance().getUid();
PopupMenu popupMenu;

com.google.android.material.button.MaterialButton requestCancelBtn;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_dispaly_requests);
}
```

```

initializeComponents();

getRequests();

private void updateList(String toString) {
    temp.clear();
    for( User v : requests){
        if(v.getDistrict().toUpperCase().contains(toString)||toString.equalsIgnoreCase("ALL")) {
            System.out.println(v.getDistrict());
            temp.add(v);
        }
    }
}

private void initializeComponents() {
    requestCancelBtn = findViewById(R.id.btnAddRequest);
    popupMenu = new PopupMenu(this, findViewById(R.id.more));
    popupMenu.getMenuInflater().inflate(R.menu.context_menu, popupMenu.getMenu());
    popupMenu.setOnMenuItemClickListener(item -> {
        if(item.getItemId() == R.id.changePass){
            FirebaseAuth.getInstance().sendPasswordResetEmail(self.getEmail());
            Snackbar snack = Snackbar.make(findViewById(android.R.id.content), "Password Reset
Link Sent On Registered Email.", Snackbar.LENGTH_LONG);
            View view1 = snack.getView();
            FrameLayout.LayoutParams params
            =(FrameLayout.LayoutParams)view1.getLayoutParams();
            params.gravity = Gravity.CENTER_VERTICAL;
            view1.setLayoutParams(params);
            snack.show();
        }else if(item.getItemId() == R.id.logout){
            FirebaseAuth.getInstance().signOut();
            startActivity(new Intent(this, SplashScreen.class));
            DispayRequestsActivity.this.finish();
        }
    });
    return true;
}

```

```

for(Tasktsk:my_tasks){
contactIds.add(tsk.WhoId);
}
for(Eventevt:my_events){
contactIds.add(evt.WhoId);
}
for(Casecse:my_cases){
contactIds.add(cse.ContactId);
}
List<Contact>contacts=[SELECTId,Name,Phone,DescriptionFROM
ContactWHERE
IdIN:contactIds];
for(Contactc:contacts){
c.Description="";
for(Tasktsk:my_tasks){
if(tsk.WhoId==c.Id){
c.Description+='BecauseofTask'+tsk.Subject+'\n';
}
}
for(Eventevt:my_events){
if(evt.WhoId==c.Id){
c.Description+='BecauseofEvent'+evt.Subject+'\n';
}
}
for(Casecse:my_cases){
if(cse.ContactId==c.Id){
c.Description+='BecauseofCase'+cse.Subject+'\n';
}
}
}
}

```

```

returncontacts;
}
}

Public class Contacts Today ControllerTest{
Public static void testGetContacts ForToday(){
Accountacct=newAccount(
Name='TestAccount'
);
insertacct;
Contactc=newContact(
AccountId=acct.Id,
FirstName='Test',
LastName='Contact'
);
inserttsk;
Eventevt=newEvent(
Subject='TestEvent',
WhoId=c.Id,
StartDateTime=Date.today().addDays(-6),
EndDateTime=Date.today().addDays(-5)
);
insertevt;
Casecse=newCase(
Subject='TestCase',
ContactId=c.Id,
Status='Closed'
);
insertcse;

```



```

List<Contact>contacts=ContactsTodayController.getContactsForToday();System
m.assertEquals(0,contacts.size());
}
}

```

```

insertc;

Tasktsk=newTask(
Subject='TestTask',
WhoId=c.Id,
Status='NotStarted'
);

```

```

inserttsk;

Eventevt=newEvent(
Subject='TestEvent',
WhoId=c.Id,

```

```

StartDateTime=Date.today().addDays(5),
EndDateTime=Date.today().addDays(6)
);
insertevt;

```

```

Casecse=newCase(
Subject='TestCase',
ContactId=c.Id
);
insertcse;

```

```

List<Contact>contacts=ContactsTodayController.getContactsForToday();
System.assertEquals(1,contacts.size());

```

```

System.assert(contacts[0].Description.containsIgnoreCase(tsk.Subject));System.ass
ert(contacts[0].Description.containsIgnoreCase(evt.Subject));
System.assert(contacts[0].Description.containsIgnoreCase(cse.Subject));

```

```

    }

    public static void testGetNoContactsForToday(){
        Account acct = new Account(
            Name = 'TestAccount'
        );
        insert acct;

        Contact c = new Contact(
            AccountId = acct.Id,
            FirstName = 'Test',
            LastName = 'Contact'

        );
        insert c;

        Task tsk = new Task(
            Subject = 'TestTask',
            WhoId = c.Id,
            Status = 'Completed'
        );
        insert tsk;

        Event evt = new Event(
            Subject = 'TestEvent',
            WhoId = c.Id,
            StartDateTime = Date.today().addDays(-6),
            EndDateTime = Date.today().addDays(-5)
        );
        insert evt;

        Case cse = new Case(
            Subject = 'TestCase',
            ContactId = c.Id,
            Status = 'Closed'
        );
    }
}

```

```

);
insertcse;

List<Contact>contacts=ContactsTodayController.getContactsForToday();System
m.assertEquals(0,contacts.size());

}

}

```

```

Public class Opportunity Alert Controller{
Publicstatic
List<Opportunity>
getOpportunities(DecimaldaysSinceLastModified,String
oppStage,BooleanhasOpen){
DateTimelastModifiedDateFilter=
DateTime.now().addDays((Integer)daysSinceLastModified*-1);
List<Opportunity>opportunities=[
SELECTId,Name,StageName,LastModifiedDate,CloseDate
FROMOpportunity
WHEREStageName=:oppStageANDLastModifiedDate<=:lastModifiedDateFilter
];
Map<Id,Opportunity>oppMap=new Map<Id,Opportunity>(opportunities);if(hasOp
pen==true){
List<Task>tasks=[SELECTID,WhatIdFROM
TASKWHEREIsClosed=falseAND
WhatIdIN:oppMap.keySet()];
List<Opportunity>opps_with_tasks=new List<Opportunity>();
for(Taskta:tasks){
if(oppMap.containsKey(ta.WhatId)){
opps_with_tasks.add(oppMap.get(ta.WhatId));
}
}
}

```

```

    }
    opportunities=opps_with_tasks;
    }

```

```

}}

```

```

publicclassOpportunityAlertControllerTest{
    publicstaticvoidtestGetOpptyWithoutOpenTasks(){
        Opportunityoppty=newOpportunity(
            Name='TestOppty',
            CloseDate=Date.today(),
            StageName='Prospecting'
        );
        insertoppty;
        Tasktsk=newTask(
            Subject='TestTask',
            WhatId=oppty.Id,
            Status='Completed'
        );
    }
}

```

```

publicstaticvoidtestGetOpptyWithOpenTasks(){
    Opportunityoppty=newOpportunity(
        Name='TestOppty',
        CloseDate=Date.today(),
        StageName='Prospecting'
    );
}

```

```
);  
insertoppty;  
Tasktsk=newTask(  
Subject='TestTask',  
WhatId=oppty.Id,  
Status='NotStarted'  
);  
inserttsk;  
List<Opportunity>opps;  
opps=OpportunityAlertController.getOpportunities(0,'Prospecting',false);  
System.assertEquals(1,opps.size());  
opps=OpportunityAlertController.getOpportunities(0,'Prospecting',true);  
System.assertEquals(1,opps.size());  
}
```

CHAPTER 7

TESTING

7.1 INTRODUCTION

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The increasing visibility of software as a system element and the attendant “costs” associated with a software failure are motivating forces for well planned, thorough testing.

7.1.1 Testing Objectives

The following are the testing objectives:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error
- successful test is one that uncovers an as yet undiscovered error.

7.1.2 Testing Principles

The basic principles that guide software testing are as follows:

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The separate principle applies to software testing.

Pareto principle states that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components.

Testing should begin “in the small “and progress toward testing “in the large.”

Exhaustive testing is not possible.

7.2 LEVEL OF TESTING

There are different levels of testing

->Unit Testing

->Integration Testing

->System Testing

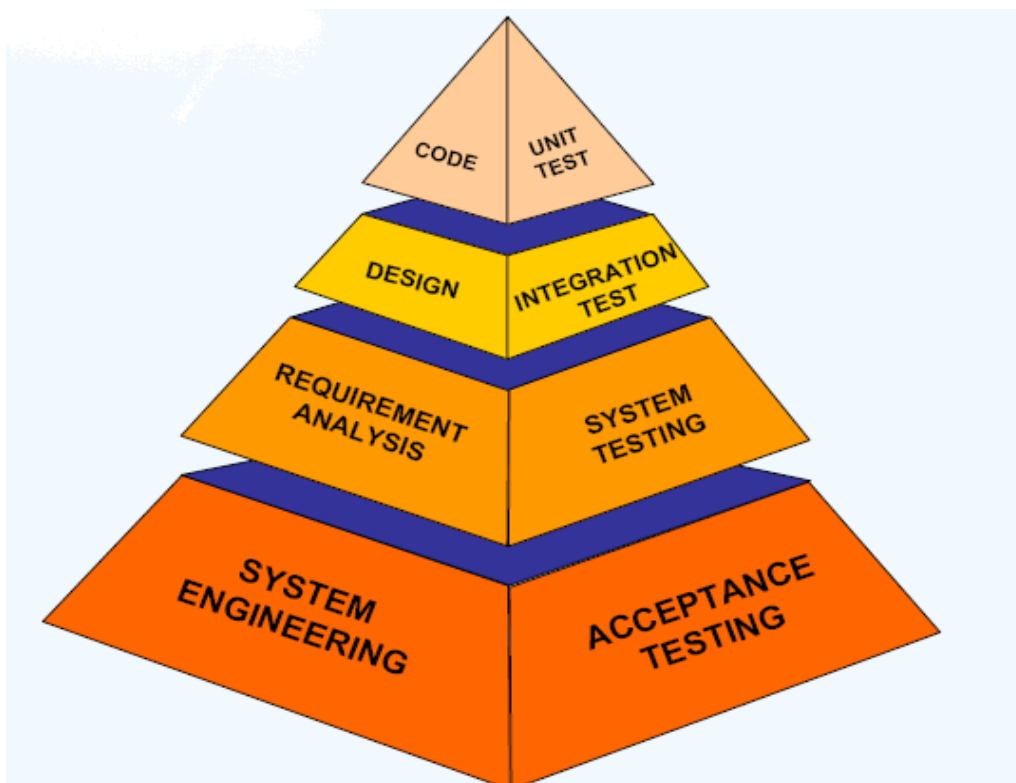


Figure 7.1: Testing pyramid

7.2.1 Unit testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The important control parts are tested to uncover with in the boundary of the module. The module interface is tested to ensure that the information properly flows into and out of the program unit and boundary conditions are tested to ensure that the modules operate properly at boundaries established to limit or restrict processing. Test data is provided through testing screens.

7.2.2 Integration testing

Integrating testing is a systematic technique for constructing Program structure while conducting tests to uncover error associates with interfacing. The objective is to take unit modules and built a program structure that has been directed by design.

- Integration Testing will test whether the modules work well together.
- This will check whether the design is correct.
- Integration can be done in 4 different ways:

7.2.3 System testing

System testing is the process of testing the completed software as a part of the environment it was created for. It is done to ensure that all the requirements specified by the customer are met. System testing involves functional testing and performance testing.

- System Testing will contain the following testing :
 - Functional Testing.
 - Performance Testing.
- Function Testing will test the implementation of the business needs.
- Performance Testing will test the non-functional requirements of the system like the speed, load etc

7.3 SOME IMPORTANT OBSERVATIONS

7.3.1 System Testing and Validation Results.

System testing was done after the system was duly coded. Individual modules of the system were checked to ensure they are fully functional units before the integrating them. This was done by examining each unit; each script was checked to ensure that it functions as required and that it performed exactly as intended. The success of each individual unit gave us the go ahead to carryout integration testing.

The system was validated using a short questionnaire that was filled by representatives of the users who were let to interact with the system using test data and provided feedback about the system features. This was done to assess if the system met their needs and requirements as regards. It was found out that the system performed in conformance to the then defined user needs and requirements. Results of the validation are shown as percentages of respondents against each requirement.

7.3.2 Testing Test Scenarios

1. Check if the page load time is within the acceptable range.
2. Check the page load on slow connections.
3. Check the response time for any action under a light, normal, moderate, and heavy load conditions.
4. Check the performance of database stored procedures and triggers.
5. Check the database execution time.
6. Check for load testing of the application.
7. Check for the Stress testing of the application.
8. Check CPU and memory usage under peak load conditions.

We have checked for scenarios and find that our system performing well in the circumstances.

7.4 TEST CASE RESULT SUMMARY

TestCase#	Description	Result
TC#1	Loading the home page	Passed
TC#2	Login	Passed
TC#3	Validating	Passed
TC#4	Content	Passed
TC#5	Course page loading	Passed
TC#6	Reports page loading	Passed
TC#7	Logout	Passed

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

A software project means a lot of experience. I learned a lot through this project. This project has sharpened our concept android application. It provides easy methods to manage the load of work easily for the users. It is much fast and more efficient as the data once entered can be used and accessed easily. This project has given me an ample opportunity to design, code, test and implements an application. This has helped in putting into practice of various Software Engineering principles concepts like maintaining integrity and consistency of data.

8.2 FUTURE SCOPE

- The Future scope is to make the system more user friendly and enhanced.
- And we will make web page for our system.
- We can give more advance software for Blood Bank Management System including more facilities.
- We will host the platform on online servers to make it accessible worldwide.
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

BIBLIOGRAPHY

The books, which are referred and which really helped me in building this system in time, are as follows: -

Books:

ANDROID	GOOGLE
JAVA	James Gosling
SOFTWARE ENGINEERING	Roger S. Pressman
LEARNING FIREBASE PROGRAMMING	Google

Web Sites:

<https://developer.android.com/studio>
<https://developer.android.com/training/basics/fir>
[tapp](https://developer.android.com/training/basics/fir)
[https://www.tutorialspoint.com/android/android_](https://www.tutorialspoint.com/android/android_studio.htm)
[studio.htm](https://www.tutorialspoint.com/android/android_studio.htm)
<https://www.javatpoint.com/android-tutorial>