Rakt

AN ONLNE BLOOD MANAGEMENT SYSTEM

AGILE SYSTEMS

05-March-2020

*By :- Life Cycle Thugs*

|  |  |  |
| --- | --- | --- |
| GROUP MEMBERS | ROLL NO. | SAP ID |
| Prashant Ruhela | R174218036 | 500071255 |
| Samarth Panwar | R174218047 | 500069716 |
| Utsav Chaudhary | R174218085 | 500071649 |

TABLE OF CONTENTS

[DECLARATION 5](#_Toc37788009)

[MANIFESTO 6](#_Toc37788010)

[PRINCIPLES 7](#_Toc37788011)

[ABSTRACT 8](#_Toc37788012)

[INTRODUCTION 9](#_Toc37788013)

[SCOPE 10](#_Toc37788014)

[ENTITY RELATIONSHIP DIAGRAM 11](#_Toc37788015)

[DATA FLOW DIAGRAM 12](#_Toc37788016)

[ZERO LEVEL DFD 12](#_Toc37788017)

[LEVEL FIRST DFD 13](#_Toc37788018)

[SECOND LEVEL DFD 13](#_Toc37788019)

[PROCESS FLOW DIAGRAM 14](#_Toc37788020)

[PERT CHART 15](#_Toc37788021)

[Pert chart network diagram 16](#_Toc37788022)

[16](#_Toc37788023)

[SELECTED METHODOLOGY 18](#_Toc37788024)

[WATERFALL MODEL 18](#_Toc37788025)

[PROCEDURE 19](#_Toc37788026)

[STATE DIAGRAM 20](#_Toc37788027)

[SEQUENCE DIAGRAM 21](#_Toc37788028)

[CLASS DIAGRAM 22](#_Toc37788029)

# DECLARATION

I thus announce that the project work entitled by under the direction and Supervision of Dr Durgansh Sharma at University of Petroleum & Energy Studies, Dehradun.

The Started Time of the Project,

* 16-Jan-2020

The Ending Time of the Project,

* End of Semester

# MANIFESTO

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

* **Individuals & interactions** over processes and tools.
* **Working software** over comprehensive documentation.
* **Customer collaboration** over contract negotiation.
* **Responding to change** over following a plan.

That is while there is value in the items on the right, we value the items on the left more.

Kent Beck James Grenning Robert C. Martin

Mike Beedle Jim Highsmith Steve Mellor

Arie van Bennekum Andrew Hunt Ken Schwaber

Alistair Cockburn Ron Jeffries Jeff Sutherland

Ward Cunninghum Jon Kern Dave Thomas

Martin Fowler Brain Marick

# PRINCIPLES

We follow these principles:

* Our highest priority is to satisfy the customer through early & continuous delivery of valuable software.
* Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
* Deliver working software frequently, from a couple of weeks to couple of months, with preference to the shorter timescale.
* Business people and developers must work together daily throughout the project.
* Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
* The most efficient & effective method of conveying information to & within a development team is face-to-face conversation.
* Working software is the primary measure of progress.
* Agile processes promote sustainable development. The sponsors, developers & users should be able to maintain a constant pace indefinitely.
* Continuous attention to technical excellence & good design enhances agility.
* Simplicity-the art of maximizing the amount of work not done is essential.
* The best architectures, requirements, and designs emerge from self-organizing teams.
* At regular intervals, the team reflects on how to become more effective then tunes and adjusts its behavior accordingly.

# ABSTRACT

Blood Donor Information System is to make a Computerized Information about the giver and Hospitals that are identified with giving the blood. Through this System any individual who is keen on giving the blood can enroll himself similarly, if any clinics needs to enlist itself with this System that can likewise enlist. What's more, the motivation behind my System is enrolling blood givers, and keep up their subtleties. Not just had those things, utilizing my framework effectively contact the contributors in a basic or crisis circumstance. Since this framework giving more highlights to the customers or the clinics or the blood camp gatherings.

Automated frameworks when contrasted with Paper record Systems are time consuming, laborious, and expensive. This paper presents the audit of the fundamental highlights, benefits and faults gave by the current Computer-Based Information System for Blood Banks. This study shows the correlation of different existing framework and giving some more thought regarding the automated framework.

# INTRODUCTION

Blood is all around perceived as the most valuable component that supports life. It spares endless lives over the world in an assortment of conditions. A blood donation center is a spot structured particularly for the capacity of blood and blood items. The expression "blood bank "typically alludes to a division of an emergency clinic research center where the capacity of blood item happens and where appropriate testing is performed to decrease the danger of transfusion related occasions. Huge coolers hold these items at a consistent temperature and they are accessible immediately. The blood benefactor data framework offers functionalities to snappy access to enroll the giver, and gathered contributor subtleties from different pieces of the Provinces. It empowers observing of the outcomes and execution of the blood gift action with the end goal that important and quantifiable targets of the association can be checked. In my framework I'm giving the proficient hunt who needs the blood in their own city, name, and blood bunches as quick as would be prudent.

Blood donation center or the Hospital acknowledge the gave blood, just if giver fulfill the entirety of the following conditions :-

* If the donor’s are between age group of 18 – 60 years.
* If the donor’s weight is 45 Kg or more.
* If the donor’s haemoglobin is 12.5 gm% minimum.
* If the donor’s last blood donation was 6 months earlier & etc.

# SCOPE

This framework isn't just for business reason. This can take for social administrations, in such a case that we are utilizing this framework for a medical clinic, it will be make simple to enlist the givers and contact the benefactors in a crisis circumstance.

This framework is independent application, this framework utilizing neighborhood Database to store contributor's subtleties utilizing GUI (Graphical User Interface). This framework interface will have many capacity to control no problem at all

# ENTITY RELATIONSHIP DIAGRAM

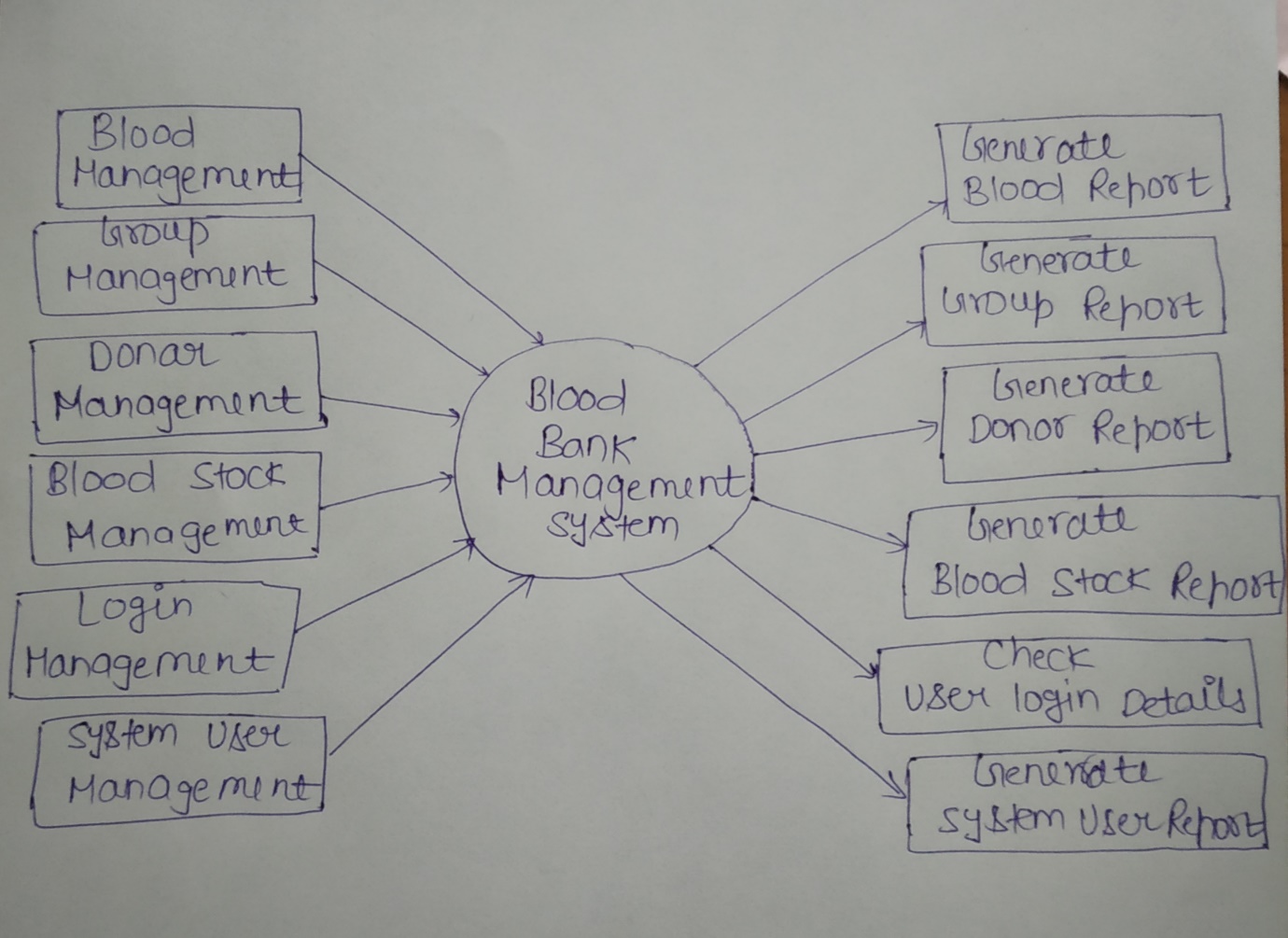


# DATA FLOW DIAGRAM

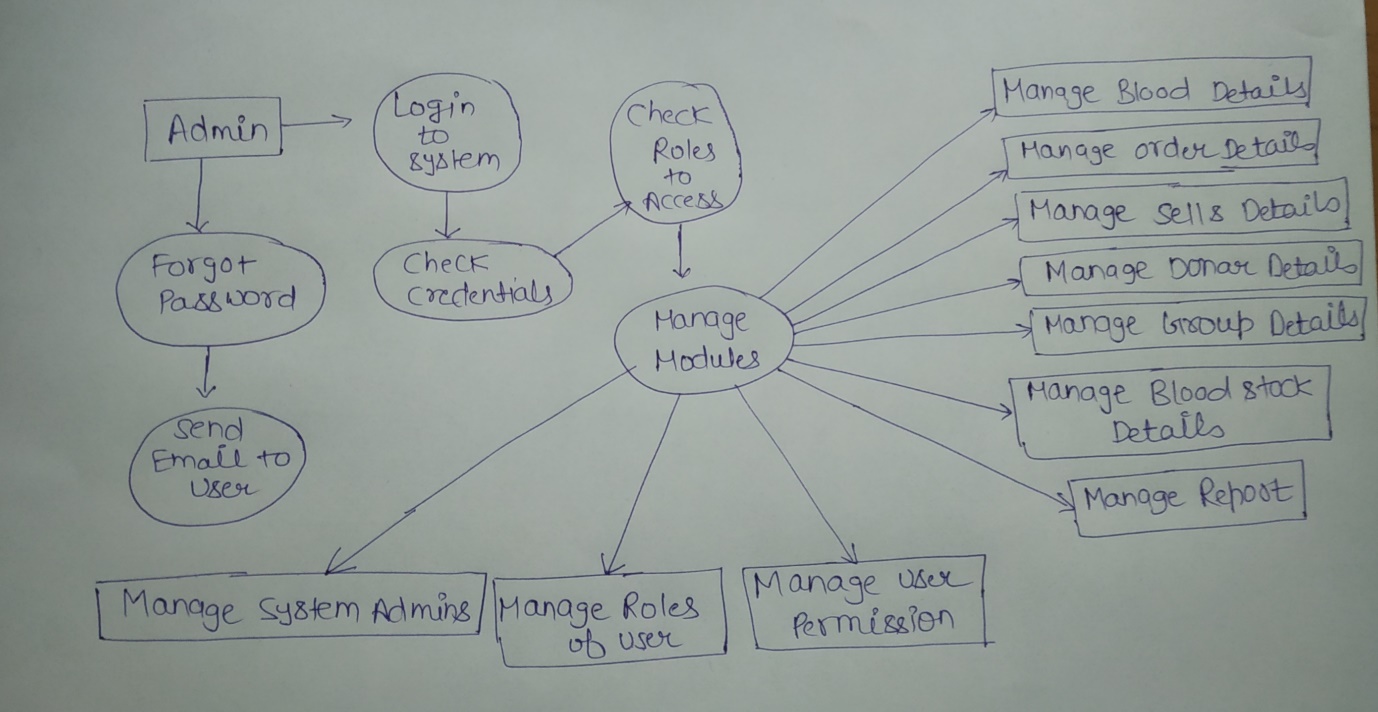
## ZERO LEVEL DFD



## LEVEL FIRST DFD

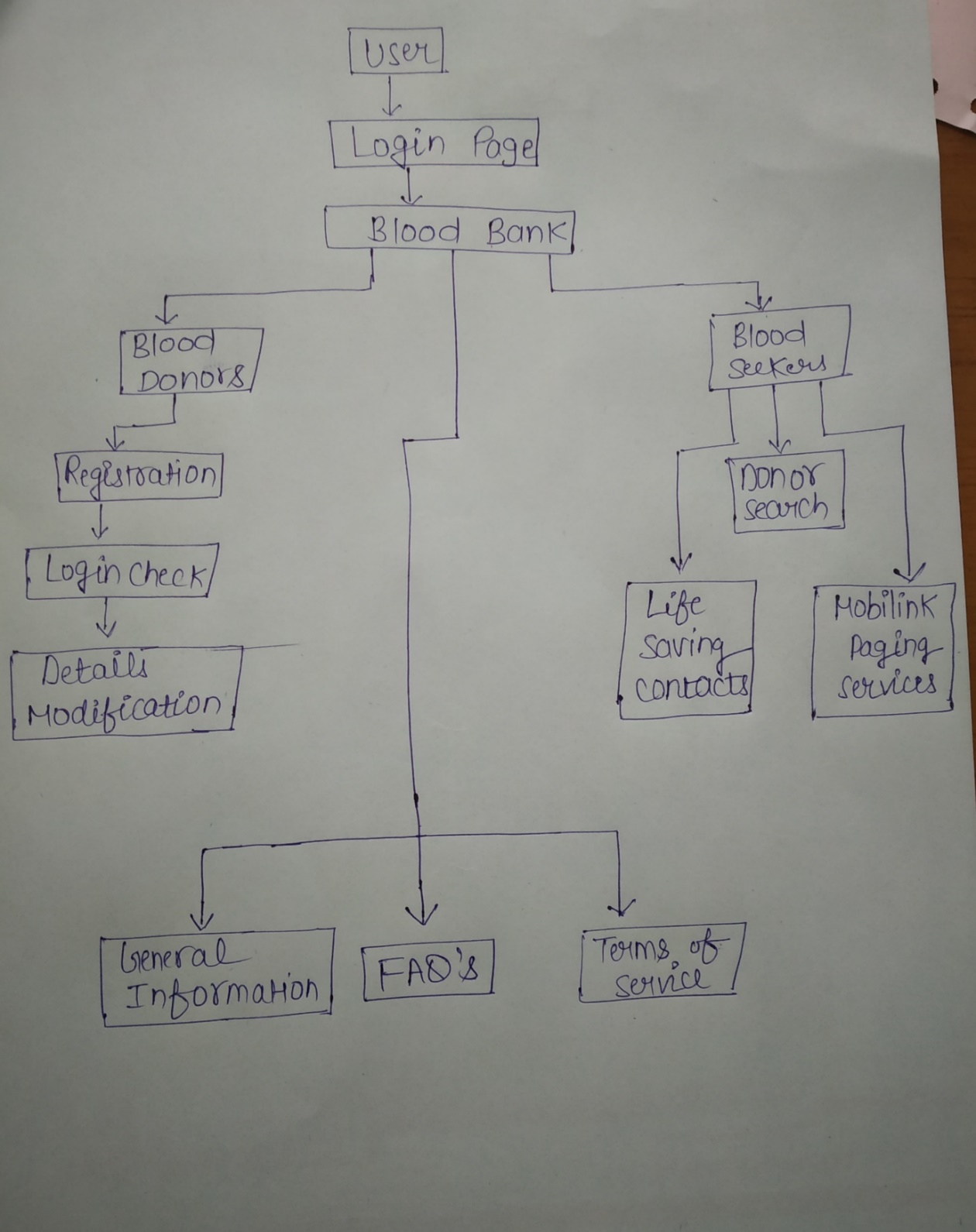


## SECOND LEVEL DFD



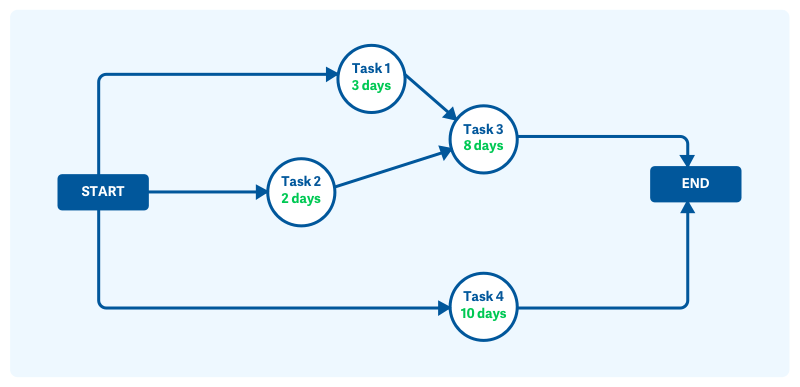
# PROCESS FLOW DIAGRAM

A **process flow diagram** (PFD), also known as a flowsheet, is a type of **flowchart** used by chemical and **process** engineers to illustrate high-level **processes**. You should create your **process flow diagram** so that it focuses on major plant **processes** and not show minor details.



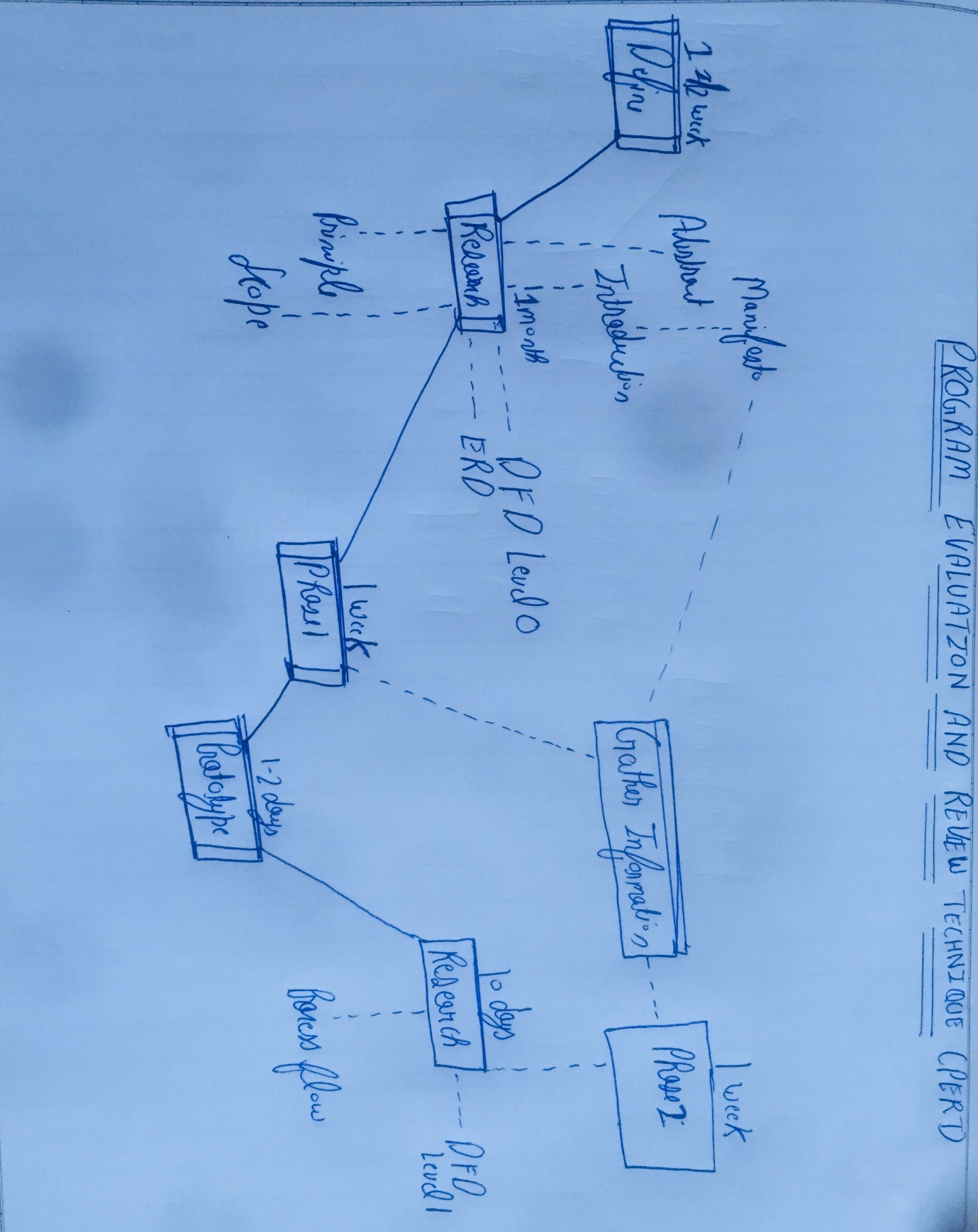
# PERT CHART

Pert chart is a project management tool used to plan, sort out, and arrange assignments inside a task. It is essentially a technique to analyze the charts associated with finishing a given venture, particularly the time expected to finish each undertaking, and to distinguish the base time expected to finish the all out task.



## Pert chart network diagram

## 



# SELECTED METHODOLOGY

While picking an advancement life cycle, we don't simply confide in our emotions. Choose based on factors that truly matter. Which life cycle will work best for our venture? This is a significant key question because settling on an inappropriate decision could prompt unfortunate consequences of catastrophic proportions. Consider deferred conveyances, miserable customers, venture invades, and cancelled projects. During the 80s and mid 90s, the cascade model was the accepted in venture conveyance. With the quick pace in programming advancement and well known utilization of the Internet, numerous companies started moving to increasingly adaptable life cycles, for example, the iterative, gradual, winding, and agile. These new life cycle techniques give greater adaptability and bolster quick paced development, giving organizations the edge in conveying "the first" in the business. Until now, there are dozens of life cycle techniques accessible to browse, each having its own favorable circumstances and disadvantages. I have chosen the Waterfall Method to build up my framework called blood giver information system. Since the water fall technique is ideal to full fill every phase inside the time allotment.

## WATERFALL MODEL

The cascade model is a consecutive structure process, regularly utilized in programming development processes, in which progress is viewed as streaming consistently downwards (like a cascade) through the periods of arranging, examining, planning, coding, testing, executing, and maintaining. The cascade improvement model starts in the assembling and construction industries. Exceptionally organized physical situations in which sometime later changes are prohibitively expensive, if certainly feasible. Since no proper programming advancement methodologies existed at that point, this equipment arranged model was just adjusted for software development. When done well the cascade technique is astounding for huge ventures and there are no surprises when the application is at long last conveyed as all highlights and even the appearance of the application has been completely determined and comprehended by future clients of the system. If the prerequisites stage is done gravely the cascade strategy conveys disappointment as the end result will just ever be in the same class as the determinations.

* My initial step is to make the practical detail. This regularly starts life as a veryabstract prerequisites particular gave by the customer.
* At the point when the application is finished a beta discharge is distributed and given to thebusiness to testing. Any bugs discovered are quickly fixed.

going before stage is finished and culminated. Notwithstanding, there are different adjusted cascade models that may incorporate slight or significant varieties upon this process. Phases of the cascade life cycle model.

# PROCEDURE

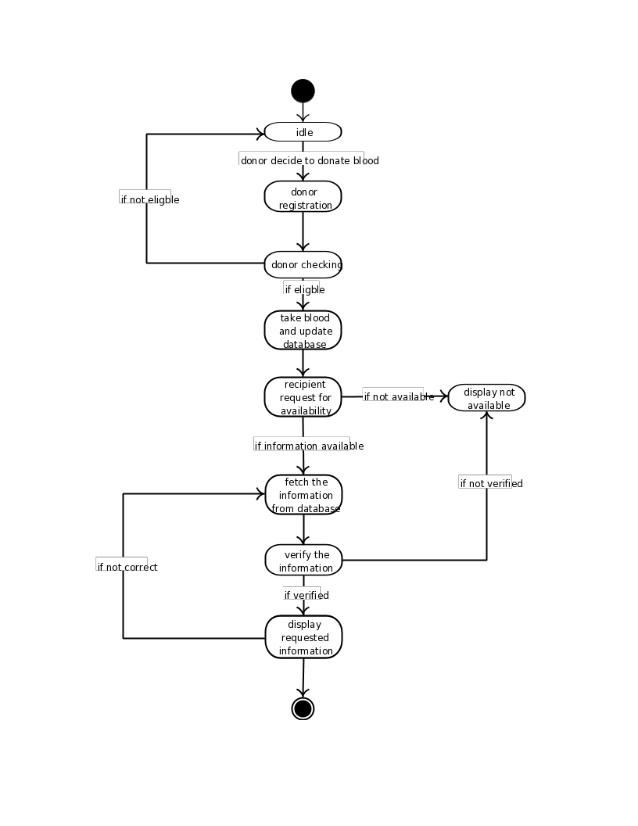
The cascade model continues starting with one stage then onto the next in a consecutive way. For instance, one initially finishes prerequisites particular, which after close down are viewed as unchangeable. At the point when prerequisites are finished, one continues to plan. The product being referred to is structured and a diagram is drawn for implementers (coders) to follow this plan ought to be an arrangement for actualizing the necessities given. At the point when the plan is finished, an execution of that structure is made by coders. Towards the later phases of this usage stage, separate programming segments delivered are consolidated to present new usefulness and diminished hazard through the expulsion of errors. Thus the cascade model keeps up that one should move to a stage just when it's

going before stage is finished and idealized. Notwithstanding, there are different altered waterfall models that may incorporate slight or significant varieties upon this procedure.

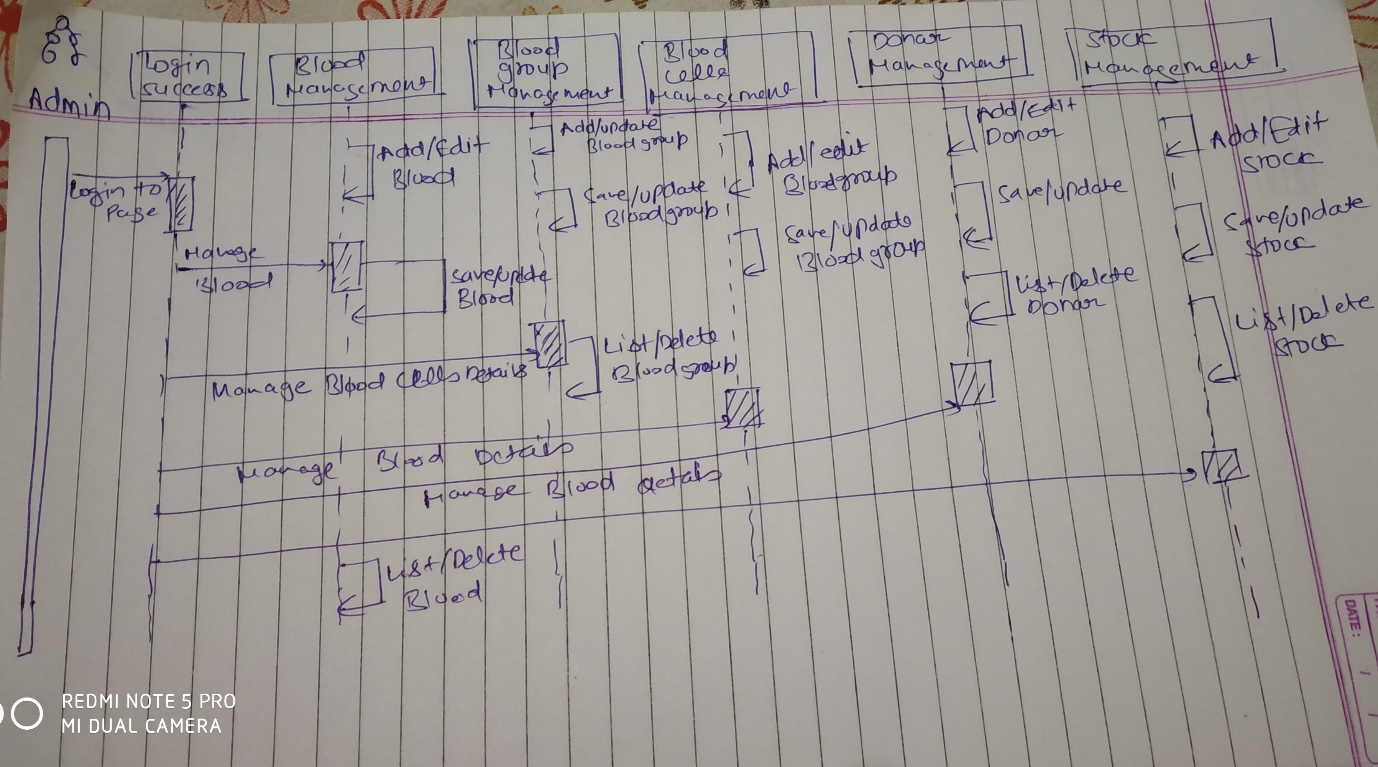
Phases of the Waterfall life cycle model

* **Requirement :-** The principal stage includes understanding what we have to plan and what is its capacity, reason and so forth. Except if we realize what we are going to structure, we cannot approach the issue. Here, the particulars of the info and yield or the final item is examined and checked.
* **Analysis :-** According to the necessities, the product and equipment required for the proper finishing of the venture is broke down right now. Directly from choosing which computer language ought to be utilized for structuring the product, to the database system that can be utilized for the smooth working of the product, such highlights are decided at this stage.
* **Design:-** The calculation (pseudo-code) of the program or the product code to be written in the following stage, is made at this point. This calculation frames the spine for the actual coding process. Appropriate arranging identifying with the plan of client interface, flowcharts is done here.
* **Coding:-** In view of the calculation or flowchart structured, the real coding of the software is done at this stage. The flowcharts/calculations are changed over into instructions written in a programming language.
* **Testing:-** The product planned, requirements to experience steady programming testing and mistake rectification procedures to see whether there are any blemish or blunders. Testing is done with the goal that the customer doesn't confront any issue during the establishment of the product.
* **Maintenance:-** There are a few issues which come up in the customer condition. To fix those issues patches are discharged. Like wise to improve the item some better forms are discharged. Upkeep is done to convey these adjustments in the client condition.

# STATE DIAGRAM



# SEQUENCE DIAGRAM



# CLASS DIAGRAM

