****

Blood Bank Management System

Project Proposal

Name: Vivek Tamang

College ID: 00175893

College: Softwarica college of it and e-commerce

batch: 23

date: 01/07/2019

2019

**Contents**

[**1. Introduction** 3](#_Toc12924867)

[**1.1 Introduction of the System** 3](#_Toc12924868)

[**1.2 Background of the System** 3](#_Toc12924869)

[**1.3 Justification of the Project** 3](#_Toc12924870)

[**1.4 Overview of the System** 3](#_Toc12924871)

[**2. Scope** 3](#_Toc12924872)

[**2.1 Aims of the project** 3](#_Toc12924873)

[**2.2 Objectives of the project** 3](#_Toc12924874)

[**2.3 Features to be included** 4](#_Toc12924875)

[**2.4 Overview of the scope** 4](#_Toc12924876)

[**3. Developmental Methodology** 4](#_Toc12924877)

[**3.1 Methodology to be used** 4](#_Toc12924878)

[**3.2 Design Pattern** 6](#_Toc12924879)

[**3.3 System Architecture** 7](#_Toc12924880)

[**4. Scheduling** 7](#_Toc12924881)

[**4.1 WBS (Work Breakdown Structure)** 7](#_Toc12924882)

[**4.2 Milestones** 8](#_Toc12924883)

[**4.3 Gantt chart** 10](#_Toc12924884)

[**5. Risk Management** 11](#_Toc12924885)

[**6. Configuration Management** 12](#_Toc12924886)

[**7. Conclusion** 13](#_Toc12924887)

**Table of Figures**

[Figure 1: Waterfall model 5](#_Toc12925151)

[Figure 2: MVC Pattern 6](#_Toc12925152)

[Figure 3: 3 Tier Architecture 7](#_Toc12925153)

[Figure 4: Scheduling 10](#_Toc12925154)

[Figure 5: Gannt Chart 10](#_Toc12925155)

[Figure 6: Github 12](#_Toc12925156)

# **1. Introduction**

## **1.1 Introduction of the System**

Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

## **1.2 Background of the System**

Proper Blood Bank Management System was not introduced before. Data are stored in excel only which is not proper and efficient. There were very high chance of corruption.

## **1.3 Justification of the Project**

The system which we are going to build will overcome all the problem in current system. Our system will be user friendly and provide better HCI.It will be very efficient and secure which will provide very good platform for the needy to get blood from the donor. It will help on managing data and information in a proper way.

## **1.4 Overview of the System**

As there are no proper blood bank management system, we are going to build a proper blood bank management system with better HCI, and which will provide secure platform for the blood donor and receivers.

# **2. Scope**

## **2.1 Aims of the project**

The aim of the project is to build a blood bank management system that manage the data and information of the blood bank so that the users could store, edit, retrieve and view data and information easily.

We will provide the platform to needy people to collect the blood needed from donors.

Our mission is to Define Quality Policy for the IT era, set new span for Services to customers.

## **2.2 Objectives of the project**

Objectives of the project are:

1. Better HCI
2. Efficiency
3. Secure user login and communication
4. Reduce corruption
5. The system prevents blood issue if cross-match is not done or fails

## **2.3 Features to be included**

1. CRUD functions
2. Create and login account by user, donor.
3. Send request for blood
4. All donor related reports are pdf downloadable.

## **2.4 Overview of the scope**

The system that stores the data and information of the blood bank so that users could store, edit, retrieve and view data and information easily and efficiently. It will also provide the secure user login and communication and provides various features like sending requests, reports downloading etc.

# **3. Developmental Methodology**

## **3.1 Methodology to be used**

I have chosen Waterfall Method for the development of this software. It is traditional System Development Life Cycle (SDLC). This method involves a complete set of steps that a team follows. The fundamental idea is to divide the development process into a series of phases or stages, each of which finishes before next one starts.

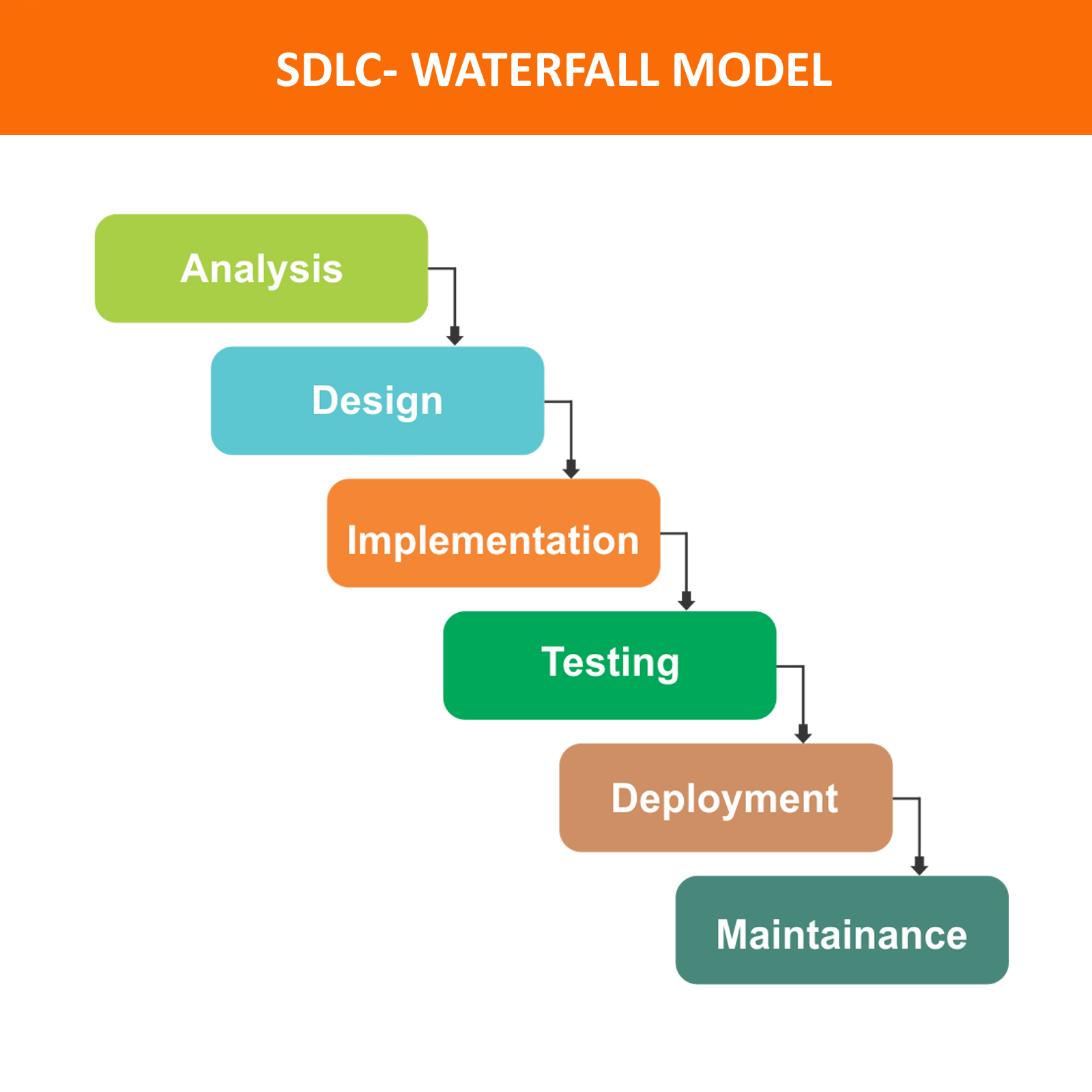


Figure 1: Waterfall model

First all possible requirements of the system to be developed are captured in this phase and documented. And the requirement specification from first phase are studied in this phase and the system design is prepared. With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. All the units developed in the implementation phase are integrated into a system after testing at verification phase.

There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customers’ environment.

## **3.2 Design Pattern**

The design pattern implemented on this project will be **Model View Controller (MVC)** design pattern. Most of the languages use MVC pattern these days. Both web and desktop-based languages uses MVC pattern but not all the language uses this pattern. It allows a front-end developer and a back-end developer to work on the same system without interfering, sharing, or editing files either party is working on.



Figure 2: MVC Pattern

**Model**

This part is responsible for data and datatype which are manipulated on the controller.

**Controller**

Here the data is manipulated using CRUD functions and provided to View for output.

**View**

Here the user receives or provides the data to be manipulated.

## **3.3 System Architecture**

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system.

3-tier architecture has been shown below.

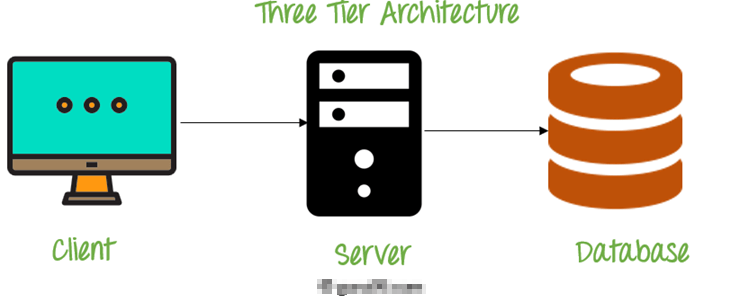


Figure 3: 3 Tier Architecture

# **4. Scheduling**

## **4.1 WBS (Work Breakdown Structure)**

The process of breaking down complex structure into manageable process is WBS.

Figure 4: WBS

## **4.2 Milestones**

|  |  |  |
| --- | --- | --- |
| **Milestones** | **No. of Days** | **Date** |
| **Proposal**  Introduction  Scope and objectives  Development Methodology  Scheduling  Risk Management  Configuration Management | **14** 2 3 4 1  3  2 | **June 17th - June 30th, 2019** June 17th - June 19th, 2019  June 20th- June 22nd, 2019  June 23rd – June 26th ,2019  June 24th – June 24th ,2019  June 25th – June 27th ,2019  June 27th – June 29th ,2019 |
| **Analysis**  SRS Use case Requirement Analysis | **28** 9  9  11 | **1st July– July 29th, 2019**  July 1st – July 9th, 2019 July 10th – July 18th, 2019 July 18th – July 29rd, 2019 |
| **Design**  Database Design Interface Design Structure Design Response Design | **31** 7  8  9  7 | **July 30th – August 29th, 2019** July 30th – August 5th, 2019 August 6th – August 12th, 2019 August 13th – August 22th, 2019 August 23th – August 29th, 2019 |
| **Coding** Database Creation  System Creation | **22**  11  11 | **August 30th – September 20th, 2019** August 30th – September 9th, 2019 September 10th – September 21nd, 2019 |
| **Testing** Input/ Output Testing White Box Testing Black Box Testing Validation Test | **10** 3 3 3 1 | **September 21st – September 30th, 2019** September 21st - September 23rd, 2019 Septemner 24th – September 26th, 2019 September 27th – September 29th, 2019 September 30th, 2019 |
| **Deployment** User Interaction Final Report | **12** 8 4 | **October 1st – October 12th, 2019** October 1st – October 8th, 2019 October 9th – October 12th, 2019 |
| **Total Days** | **117** |  |

Allocation Description

* **Proposal**

It has been allocated with 14 days. 14 days are further divided into 4 tasks..

* **Analysis**

In this phase, I will gather and analyze the requirements of the system.

* **Design**

In this phase, I shall design the system which includes UI design, database design, structural modelling and Response design.

* **Coding**

In this phase, I need to develop the system which include its overall coding and database design/building.

* **Testing**

After the system has been completed, I will test it for its efficiency and reliability. Testing of the system includes Black Box testing, White Box testing, input/output testing and Validation Testing.

* **Deployment**

In this phase, I need to finalize the product developed. It includes, training the user and the system’s documentation.

## **4.3 Gantt chart**

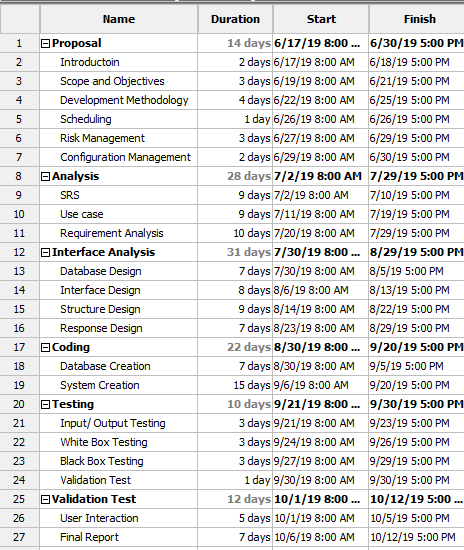


Figure 5: Scheduling

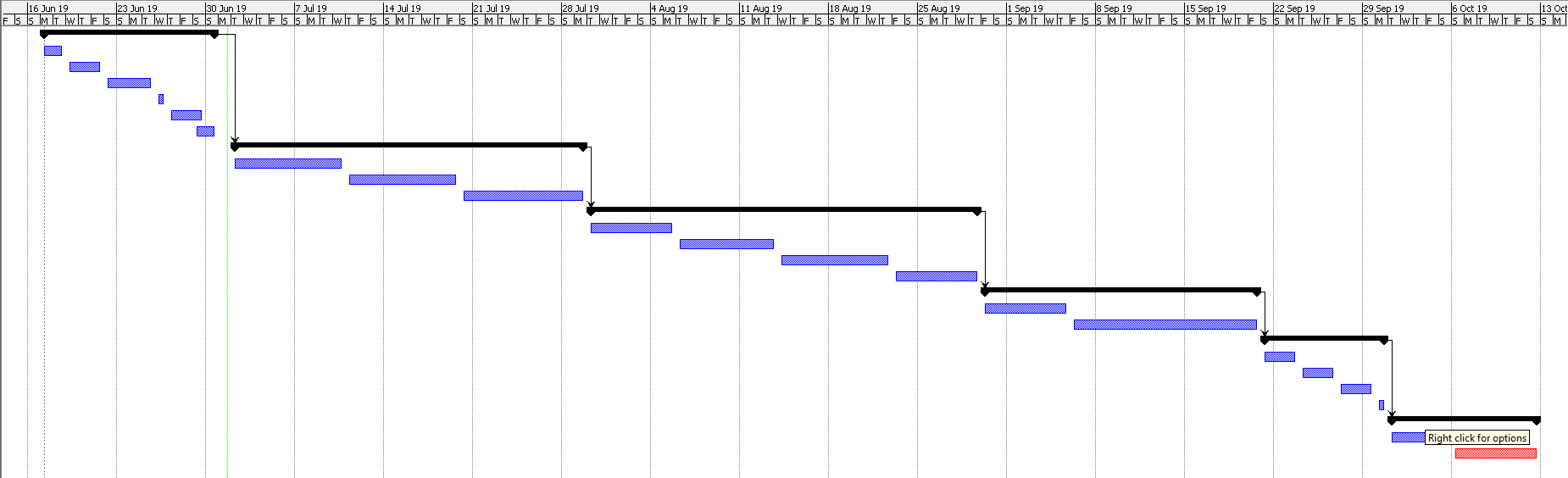


Figure 6: Gannt Chart

# **5. Risk Management**

There are many risks to a project and it’s our job to identify them and solve them so that problems doesn’t arise in future.

Some of the methods to handle those risks are:

1. Avoid

Some risks can avoidable, so the best way for it is to avoid to complete project in time.

1. Reduce

We can reduce the risks, if we cannot prevent it.

1. Transfer

We can transfer the risk to others so that we don’t have to face the risk.

1. Accept

If the risks cannot be avoided, reduced or transferred then only the option is to accept it.

Table for Risk likelihood.

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Table for Risk Consequence

|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very Low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

We are going to determine Impact by referring to how much likelihood an event might occur and its consequence. Using a table below, Impact is determined.   
**Impact = Likelihood \* Consequence**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Event** | **Likelihood** | **Consequence** | **Impact** | **Solution** |
| 1 | System Crash | 1 | 4 | 4 | Back up should be created |
| 2 | User requirement not fulfilled | 1 | 5 | 5 | Requirement analysis should be done properly. |
| 3 | Lack of user friendliness | 2 | 3 | 6 | Design should be done with user in mind. |
| 4 | Malware/ Virus | 2 | 4 | 8 | Antivirus and other related software should be used. |

# **6. Configuration Management**

Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.

It helps to classify documenting functional capabilities, individual configuration items and interdependencies.

Link: https://github.com/vivekyonjan04/CP

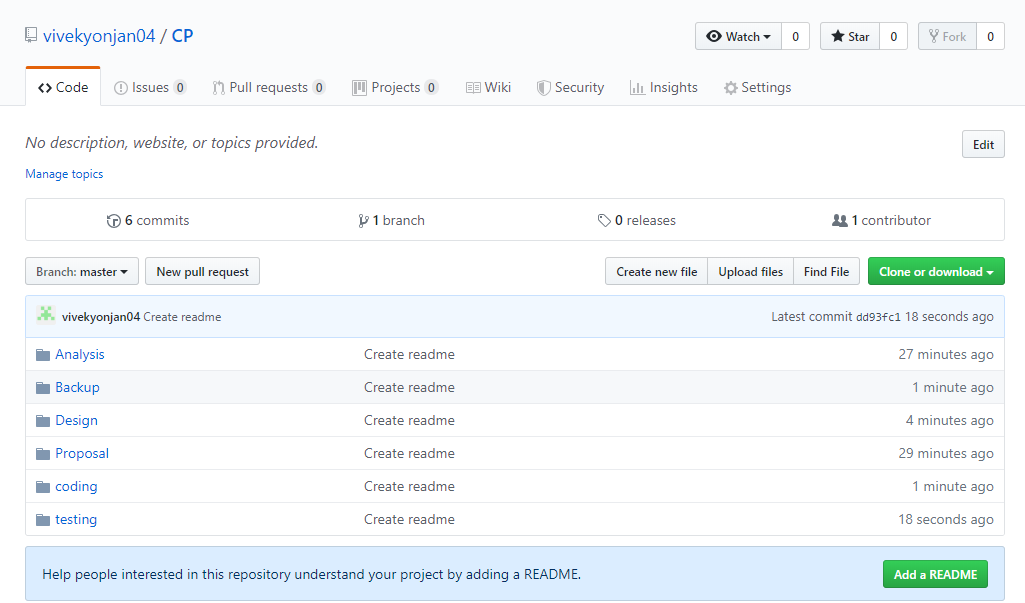


Figure 7: Github

# **7. Conclusion**

Our project will focus on providing platform for the needy people to get donor easily. It will offer user with a friendly environment. Data management will be proper and efficient with the help of our system.