Project Proposal

On

Blood Bank Management System



Pramila Danuwar

170174

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT and E-Commerce

Kathmandu , Nepal

01/04/2019

Submitted to : Kiran Rana

Contents

[1. Introduction 3](#_Toc534565240)

[1.1 Introduction about Blood Bank Management System 3](#_Toc534565241)

[1.2 Justification for project 3](#_Toc534565242)

[1.2.1 Background of the project 3](#_Toc534565243)

[1.2.2 Problem Statement 3](#_Toc534565244)

[1.3 Description of the project 4](#_Toc534565245)

[1.3.1 Features 4](#_Toc534565246)

[2. Project Scope 4](#_Toc534565247)

[2.1 Scope and Limitation of project 4](#_Toc534565248)

[2.2 Aims and Objectives 5](#_Toc534565249)

[3. Development Methodology 6](#_Toc534565250)

[3.1 Methodology used 6](#_Toc534565251)

[3.2 Design Pattern 7](#_Toc534565252)

[3.3 System Architecture 8](#_Toc534565253)

[4. Work Breakdown Structure (WBS) / Scheduling 9](#_Toc534565254)

[4.1 Work Breakdown Structure 9](#_Toc534565255)

[4.2 Milestones 10](#_Toc534565256)

[4.3 Scheduling / Gantt Chart 12](#_Toc534565257)

[5. Risk Management 14](#_Toc534565258)

[6. Configuration Management 15](#_Toc534565259)

[7. Conclusion of the project 17](#_Toc534565260)

[8. References 17](#_Toc534565261)

List of Figure

[Figure 1 waterfall Model 6](#_Toc534564704)

[Figure 2 Model view controller 7](#_Toc534564705)

[Figure 3 3-tier 8](#_Toc534564706)

[Figure 4 Work Breakdown Structure 9](#_Toc534564707)

[Figure 5 Scheduling 11](#_Toc534564708)

[Figure 6 Gantt chart 12](#_Toc534564709)

[Figure 7 github 15](#_Toc534564710)

[Figure 8 command prompt 15](#_Toc534564711)

# 1. Introduction

## 1.1 Introduction about Blood Bank Management System

Blood Bank is a place where blood bags that is collected from blood donation event is stored in one place. The term 'Blood bank' refers to a division of a hospital laboratory where the storage of blood bank product occurs and proper testing is performed to reduce the risk. The system keeps the record of all the donors, recipients, blood donation programs, rejected bloods. This system has an ability to keep track of the donors' donation records and blood stock in the blood bank. This project intends to computerize the blood and donor management system in blood bank in order to improve the record management efficiency due to the grown size of records of data.

## 1.2 Justification for project

### 1.2.1 Background of the project

This project is designed to handle the daily transactions of the blood bank and research the details when required. It also helps to register the details of donors, blood collection details as well as blood issued reports and all the aspects of blood banking is completely managed by the software.

### Problem Statement

To build a solution to the ever growing requirement of blood due to accidents and various health problems the system is developed for accessing the information about various blood banks and hospitals and their blood stock.

* Scarcity of rare blood group.
* Unavailability of blood during emergency.
* Less awareness among people about blood donation and blood transfusion.

## 1.3 Description of the project

### 1.3.1 Features

The features of the project are as follows :

* Provide the searching facilities based on various factors such as blood bank, donors, blood cell, etc.
* It tracks all the information of Blood Group, Blood stock, etc.
* Blood Donation Camp & Camp Organizer Management.
* Donor Management - Donor Registration, Managing donor database, recording their physical and medical statistics.
* Blood requisition and issuance of blood.
* List of Donors who are eligible for donation on a particular date with contact Number.
* Camp Wise Donor List and Printing of Donor Cards

# 2. Project Scope

## 2.1 Scope and Limitation of project

**Scope:**

* The project should be able to give complete info about blood donors, and activities of the blood bank regarding the blood donation.
* Donors are provided with registration process to maintain future donations as well as for the future reference.
* Blood recipients can place order for blood.
* It can be easily understood by the user and the operator.
* It saves time, it is not necessary to create manifest it can be printed directly.

**Limitation:**

Although I have put my best effort to make the software, flexibility, easy to operate but limitations cannot be ruled out even by me.

* Off-line reports of blood bank, blood cell, and donor cannot be generated due to batch mode execution.
* The transactions are executed in off-line mode, hence on-line data for donor, blood stock capture and modification is not possible.

## 2.2 Aims and Objectives

**Aim** :

* The system is used for maintaining all the process and activities of blood bank management system.
* The system can be extended to be used for maintaining records of hospital, organ donation and other similar sectors.
* While developing the system, there shall be space for further modification. There shall be a proper documentation so that further enhancement becomes easy.
* It helps to make all the procedure automated and with the help of computer it becomes fast and accurate.

**Objective** :

* To provide a means for the blood bank to publicize and advertise blood donation programs.
* To allow the probable recipients to make research and match the volunteer donors, and make request for the blood.
* TO improve the efficiency of blood stock management by alerting the blood bank staffs when the blood quantity is below it par level or when the blood stock has expired.
* To provide an efficient donor and blood stock management function to the blood bank by recording the donor and blood details.

# 3. Development Methodology

## 3.1 Methodology used

For this project I have chosen waterfall model for the development part. Waterfall model is a software development process. The waterfall model emphasizes that a logical progression of steps be taken throughout the software development life cycle.

Figure 1 waterfall Model

# 

Firstly all the requirements for the development of the software are collected and then they are documented in this phase. When the requirements are fulfilled from the first stage then they are studied and design is made for the system. A design specification is an outlines how the system covers the whole project. Then all the codes are implemented and are integrated into a system after verification. When the entire project is done there may come some difficulties or some issues that need to be solve out so it is done in maintenance phase.

## 3.2 Design Pattern

A software design pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is not a finished design that can be transformed directly into source or machine code.

For this project I have use model view controller, Model–view–controller is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

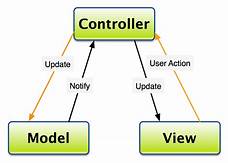


Figure 2 Model view controller

The **Model** in a todo app might define what what a “task” is and that a “list” is a collection of tasks.

The **View** code will define what the todos and lists looks like, visually. The tasks could have large font, or be a certain color.

Finally, the **Controller** could define how a user adds a task, or marks another as complete. The Controller connects the View’s add button to the Model, so that when you click “add task,” the Model adds a new task.

## 

## 3.3 System Architecture

A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers.

For example, the user interface of a web application could be redeveloped or modernized without affecting the underlying functional business and data access logic underneath. This architectural system is often ideal for embedding and integrating 3rd party software into an existing application. This integration flexibility also makes it ideal for embedding analytics software into pre-existing applications and is often used by embedded analytics vendors for this reason.

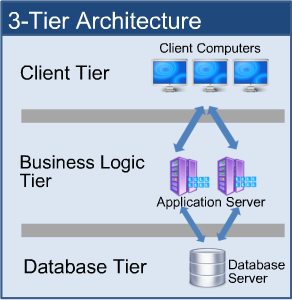


Figure 3 3-tier

* A presentation layer that sends contact to browsers in the form of HTML/JS/CSS. This might leverage frameworks like React, Ember, Aurora,etc.
* An Application Layer that uses an application server and processes the business logic for the application. This might written in C#, java,etc.,
* A Data Layer which is a database management system that provides access to application data. This could be MSSQL, MySQL,etc.

# 4. Work Breakdown Structure (WBS) / Scheduling

## 4.1 Work Breakdown Structure

WBS (Work Breakdown Structure) is a hierarchical and incremental decomposition of the project into phases, deliverables and work packages. It is a tree structure, which shows a subdivision of effort required to achieve an objective. A WBS also provides the necessary framework for detailed cost estimating and control along with providing guidance for schedule development and control.

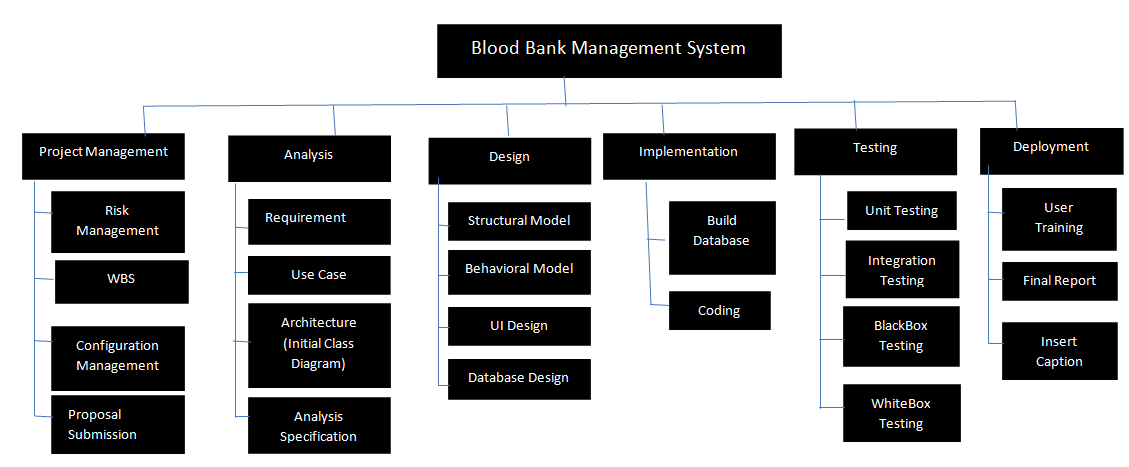


Figure 4 Work Breakdown Structure

## 

## 4.2 Milestones

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  WBS  Configuration Management  Proposal Submission | From 21-dec-2018 to 3-jan-2019  From 21-de-2018 to 25-de-2018  From 26-de-2018 to 29-de-2018  From 30-de-2018 to 1-jan-2019  From 2-jan-2019 to 3-jan-2019 |
| **Analysis**  Requirement analysis  Planning  Use Case  Architecture ( Initial Class Diagram) | From 4-jan-2019 to 28-jan-2019  From 4-jan-2019 to 10-jan-2019  From 11-jan-2019 to 15-jan-2019  From 16-jan-2019 to 22-jan-2019  From 23-jan-2019 to 28-jan-2019 |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER , Data Dictionary) | From 29-jan-2019 to 27-feb-2019  From 29-jan-2019 to 5-feb-2019  From 6-feb-2019 to 11-jfeb-2019  From 12-feb-2019 to 20-feb-2019  From 21-feb-2019 to 27-feb-2019 |
| **Implementation**  Building Database  Coding | From 28-feb-2019 to 31-march-2019  From 28-feb-2019 to 11-march-2019  From 12-march-2019 to 31-march-2019 |
| **Testing**  Unit Testing  Integration Testing  Blackbox Testing  Whitebox Testing | From 1-april-2019 to 10-april-2019  From 1-april-2019 to 2-april-2019  From 3-april-2019 to 4-april-2019  From 5-april-2019 to 7-april-2019  From 8-april-2019 to 10-april-2019 |
| **Deployment**  User Training  Final Report | From 11-april-2019 to 20-april-2019  From 11-april-2019 to 12-april-2019  From 13-april-2019 to 21-april-2019 |

**Description of Milestones :**

* **Project Management (14 days)**

I have allocated 14 days for project management.

Risk Management for 5days

WBS for 4 days

Configuration Management for 3days

Proposal Submission for 2 days

* **Analysis**

I have allocate 25 days for analysis

Requirement analysis for 7 days

Planning for 5 days

Use Case for 7 days

Architecture ( Initial Class Diagram) for 6 days

* **Design**

I have allocate 30 days for Design

Structural Diagram for 8 days

Behavioral Diagram for 6 days

UI Design for 9 days

Database Design (ER , Data Dictionary) for 7 days

* **Implementation**

I have allocated 32 days for implementation

Building Database for 12 days

Coding for 20 days

* **Testing**

I have allocate 10 days for testing

Unit testing for 2 days

Integration testing for 2 days

Black box testing for 3 days

White box Testing for 3 days

* **Deployment**

I have allocate 10 days for testing

User Training for 2 days

Final Report for 8 days

## 4.3 Scheduling / Gantt Chart

Scheduling is generally one of the critical variables to project success for engineers.  In spite of this, most engineers don’t understand the basics of project scheduling, except for remembering it from their college days.  In this article I will document the two primary scheduling techniques available to the engineering project manager.

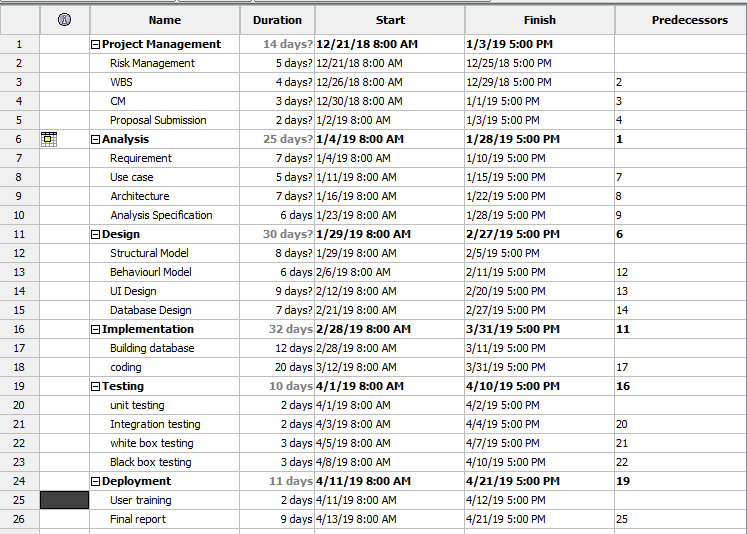


Figure 5 Scheduling

Gantt charts do an excellent job of showing the time required for various tasks and work best for an “overall picture” of where the project is at in relation to all the varied tasks. Today, Gantt charts are most commonly used for tracking project schedules. For this it is useful to be able to show additional information about the various tasks or phases of the project, for example how the tasks relate to each other, how far each task has progressed, what resources are being used for each task and so on.

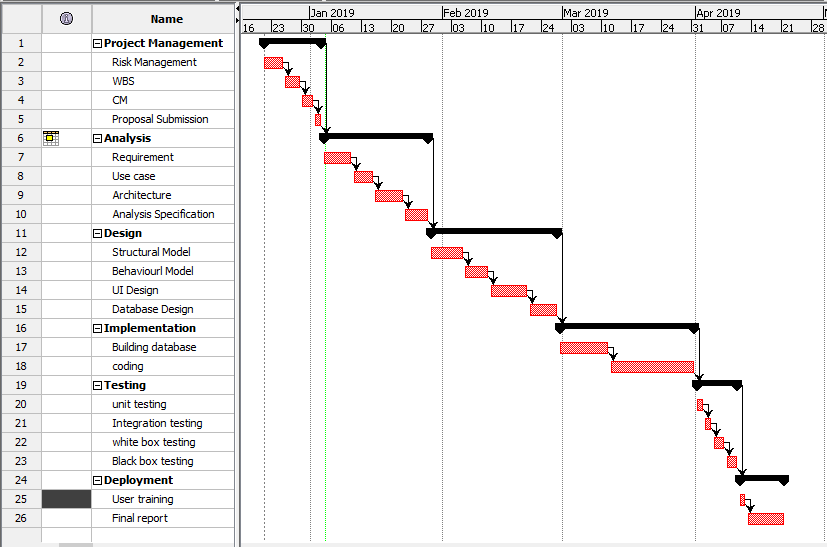


Figure 6 Gantt chart

# 5. Risk Management

Risk Management is the process of identifying, analyzing and responding to risk factors throughout the life of a project and in the best interests of its objectives. Proper risk management implies control of possible future events and is proactive rather than reactive.

**Impact = Likelihood \* Consequence**

Risk Likelihood values are shown as follows

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

Risk Consequence values are shown below

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Risks | Likelihood | Consequences | Impact | Solution |
| 1 | Lack of resources and team | 2 | 3 | 6 | Resources and team should be collected at the beginning. |
| 2 | Market risk | 1 | 5 | 5 | Product should give good service . |
| 3 | Lack of planning | 2 | 5 | 10 | Proper planning is to be done in every phase. |
| 4 | Communication | 1 | 3 | 3 | Making team engaged in project |
| 5 | Requirement | 2 | 4 | 8 | Knowing the exact requirement of the project |
| 6 | Feasibility | 2 | 2 | 4 | Asking key member to do their own sanity check. |
| 7 | Hard disk Failure | 2 | 3 | 6 | Data backup should be done. |

# 6. Configuration Management

Configuration management (CM) is a governance and system engineering process for ensuring consistency among physical and logical assets in an operational environment. The configuration management process seeks to identify and track individual configuration items (CIs), documenting functional capabilities and interdependencies.  Administrators, technicians and software developers can use configuration management tools to verify the effect a change to one configuration item has on other systems.

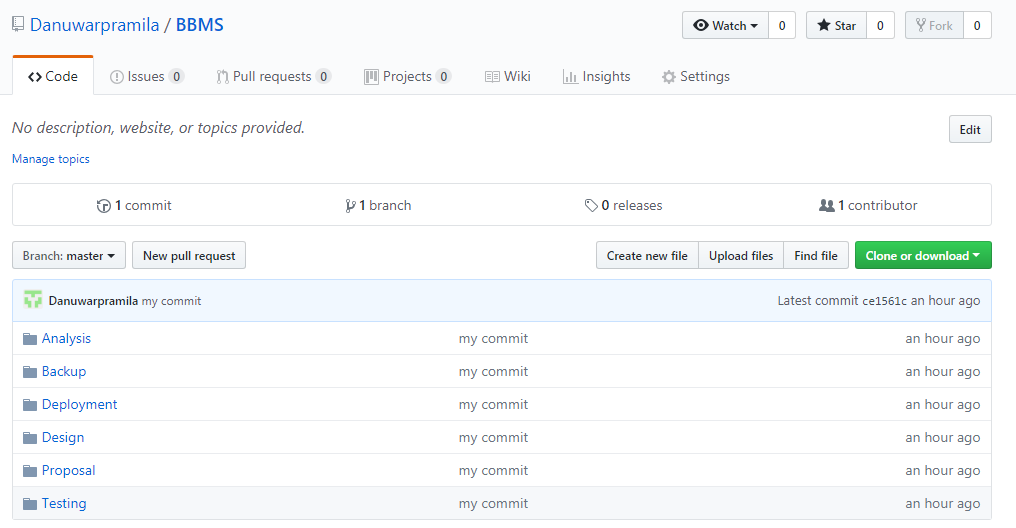


Figure 7 github

# 

Figure 8 command prompt

# 7. Conclusion of the project

The main objective of blood bank is to develop an emergency application for a person who needs blood by using altering system. As the donor's data is saved in the system we can reject those who have any infections. As we have the contact details of the donor we can contact donor whenever necessary comes. And we have a proper storage management of expired packs of blood.

# 8. References

<https://www.scribd.com/document/342098928/Synopsis-of-Blood-Bank-Management-System>

<https://searchitoperations.techtarget.com/definition/configuration-management-CM>