Contents

[Chapter-1 Introduction 3](#_Toc5713189)

[Project introduction 3](#_Toc5713190)

[Justification to the project 3](#_Toc5713191)

[Background of the project 3](#_Toc5713192)

[Problem statement 3](#_Toc5713193)

[Description of the project 3](#_Toc5713194)

[Features 3](#_Toc5713195)

[Overview of the project 4](#_Toc5713196)

[Chapter-2 Scope of the project 5](#_Toc5713197)

[Limitation 5](#_Toc5713198)

[Aims 5](#_Toc5713199)

[Objectives 5](#_Toc5713200)

[Overview 5](#_Toc5713201)

[Chapter-3 Development methodology 6](#_Toc5713202)

[Description of methodology 6](#_Toc5713203)

[Design pattern 8](#_Toc5713204)

[Architecture 10](#_Toc5713205)

[Chapter-4 Work Breakdown Structure 12](#_Toc5713206)

[Work Breakdown Structure (WBS) 12](#_Toc5713207)

[Milestone 15](#_Toc5713208)

[Risk management 19](#_Toc5713209)

[Configuration Management 20](#_Toc5713210)

[Conclusion 21](#_Toc5713211)

[Chapter 7 – References/ Bibliography 22](#_Toc5713212)

**Table of figure**

[Figure 1 Waterfall model 6](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713229)

[Figure 2 Model view controller design pattern 8](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713230)

[Figure 3Abstract factory design pattern 9](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713231)

[Figure 4Factory method 9](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713232)

[Figure 5Three tier Architecture 10](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713233)

[Figure 6 WBS 13](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713234)

[Figure 7 Milestones 15](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713235)

[Figure 8Project schedule2 17](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713236)

[Figure 9 Project schedule1 17](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713237)

[Figure 10 Gantt chart 1 18](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713238)

[Figure 11 Gantt chart 2 18](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713239)

[Figure 12 Configuration 20](https://d.docs.live.net/633a4214e51ce4b6/CP_proposal_Sanam_shrestha_00174357/CP_proposal_Sanam_shrestha_00174357.docx#_Toc5713240)

# Chapter-1 Introduction

## Project introduction

Note and planning have been hard to manage and organize from very beginning. All scatter notes difficult to find. Particular written note in hardcopy (notebook) take time to search it.

Todolist & Note Manager is a web application in which users will be able to note down notes and plan the activities. This application provides area to users to plan activities and check which are left to be complete. Here users also able to keep note important note and can share it.

## Justification to the project

To solve the problems that every notebook has which is availability. This web application is introduced to solve that problem and to provide extra features to the users. This project also helps to solve for planning activities.

## Background of the project

Before digitization were present, all the notes and planning were had to do manually in hardcopy. Now all notes and planning can be stored in digital form. Now this web application provides this service (save notes and do planning) in digital form which can be access anytime.

## Problem statement

The main goal of this web application is to provide space to users, where they can use that space to take note and prepare plan. Although this web application provides service to save their notes but cannot support audio and video format.

## Description of the project

Todolist & Note Manager is an online notebook and activity checking application that can be access from anywhere. This application provides space to users for keeping users notes and providing access to their note anytime. They can create, update, delete and share their notes. They can also notebook where small notes can be created

## Features

* Easy access of notes anytime
* Change styles easy and quick
* Pre-installed templates of notes
* Share notes
* Search notes
* Group to do list and planning
* Backup notes
* Change background color of note

## Overview of the project

Todolist & Note Manager is a web application where users can create notes, share and make activities plan.

# Chapter-2 Scope of the project

## Limitation

This application has many merits point but also some limitation of application. Some are listed:

1. To have access towards notes one must have internet connection.
2. This application does not supports audio or video for attachment.

## Aims

* To help users to create and save notes.
* To helps users to track completion of their activities.

## Objectives

* Organize all notes in one place.
* Share notes to co-workers and friends.
* User-friendly interface.
* Easy to prepare notes.
* Easy to find notes.

## Overview

This web application is introduced for users to create and manipulate the created notes. This application another main agenda is to make list of activities to be done and what left to be done (to do list).

# Chapter-3 Development methodology

## Description of methodology

**Waterfall model**

Waterfall model is s also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. This model contains following phases:

1. Requirement gathering and documentation
2. System design
3. Implementation
4. Testing
5. Deployment
6. Maintenance

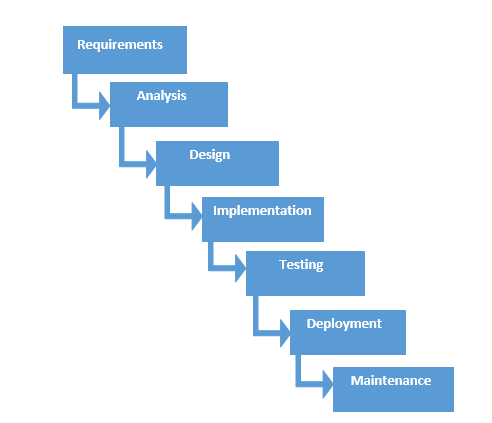


Figure 1 Waterfall model

**Benefits of Waterfall method**

* This method is simple and easy to understand
* The requirements of project will be unchanged which helps to make the products that we want to build
* Every steps must be complete to proceed further
* Steps do not overlap each other
* Start date and end date of every steps can be identified which helps to easy to measure progress

After comparing and analyzing the two different methods, I prefer to use Waterfall method over Agile.

Reasons I choose Waterfall model is, it is easy to understand and implement. Waterfall method is good for small to medium level project.

## Design pattern

**Model View Controller (MVC)**

Model View Controller is a common architectural pattern for developing software. MVC divides application into three main parts that are interconnected. It contains data model, presentation layer and control layer.

Figure 2 Model view controller design pattern

**Abstract factory**

Abstract factory pattern is also known as factory of factories. In this design pattern creates factory which creates other factories. Abstract factory falls under creational pattern. Here an interface creates factory of related objects without explicitly identifying their classes.

Figure 3Abstract factory design pattern

**Factory method**

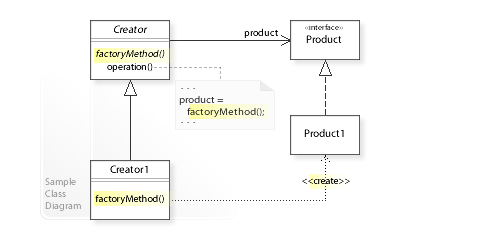
Factory method design pattern also falls under creational pattern. Here we create object without showing the creational logic to the end user and refer to newly created object using common interface.

Figure 4Factory method

After analyzing some different design pattern, I prefer MVC design pattern for this application.

I prefer MVC pattern due to following reasons;

* Any data modification made on any layer would not affect other layers.
* MVC removes code duplication.
* Code can be reuse.

## Architecture

A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture.

**Presentation layer**: this layer provides information to users related to service available in the system.

**Business layer**: this layer takes data from presentation layer provided by user and controls the functions using data access layer.

**Data access layer**: This layer stores all information provided by presentation and business layer.

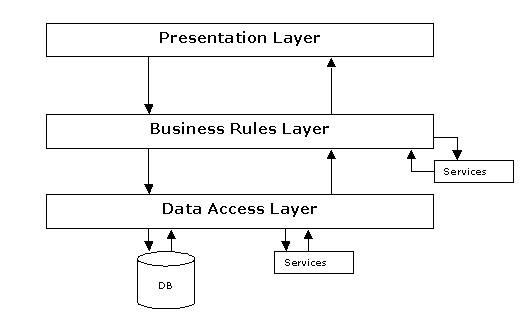
**Tools**

Figure 5Three tier Architecture

|  |  |
| --- | --- |
| **IDE/Editor** | Visual Studio Code |
| **Programming Language** | HTML, CSS, Java Script, PHP |
| **Framework** | - |
| **Programing Structure** | Object oriented |
| **Development Methodology** | Waterfall method |
| **Design Pattern** | MVC Pattern Design |
| **Architecture** | 3-tier Architecture |
| **Database** | MySQL |
| **Server** | XAMMP control panel |
| **Modeling Tool** | Visual Paradigm |

# Chapter-4 Work Breakdown Structure

## Work Breakdown Structure (WBS)

Work Breakdown Structure is breakdown of project into small parts. WBS helps to achieve project objectives. WBS provides framework for detailed cost estimation and schedule development. It is tree structure that display subdivision of that main part.

**Planning**: Planning for project will be covered in this section.

**Analysis**: Requirement, use cases and architecture will be done in analysis part

**Design**: Structural model, behavior model, UI design and DB design will be done in design part.

**Implementation**: Building database and coding will be done in implementation part.

**Testing**: Unit testing, integration testing, functional and non-functional testing will be done in testing part.

**Documentation**: User manual and final report will be done in this part.

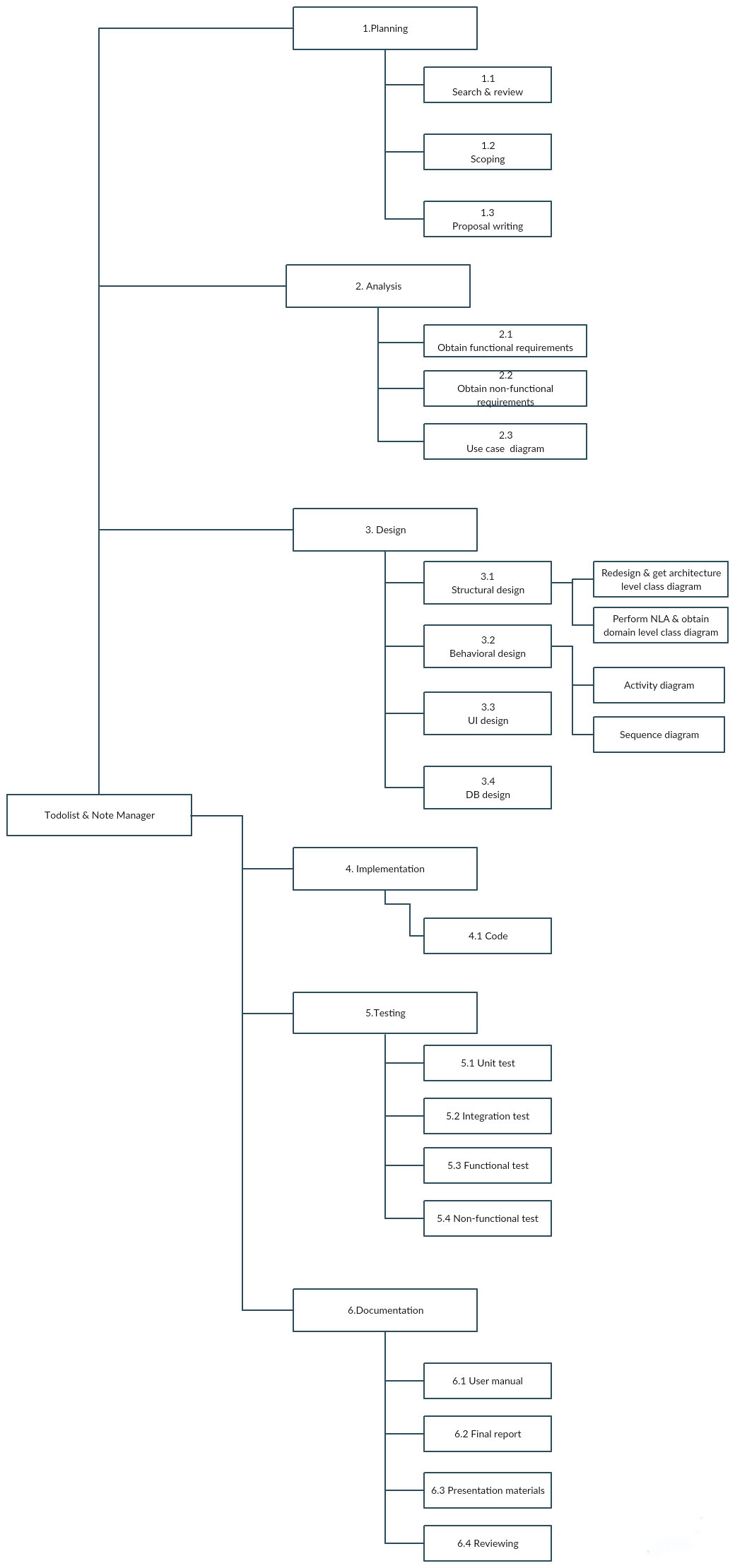


Figure 6 WBS

|  |  |  |
| --- | --- | --- |
|  | **Task name** | **Time estimated(days)** |
| 0 | **Todolist & Note manager** | **110 (Total)** |
| 1 | **Planning** | **16** |
| 1.1 | Search & review | 4 |
| 1.2 | Scoping | 2 |
| 1.3 | Proposal writing | 8 |
| 2 | **Analysis** | **29** |
| 2.1 | Obtain functional requirements | 10 |
| 2.2 | Obtain non-functional requirements | 10 |
| 2.3 | Use case diagram | 9 |
| 3 | **Design** | **25** |
| 3.1 | Structural design |  |
| 3.1.1 | Redesign & get architecture level class diagram | 5 |
| 3.1.2 | Perform NLA & obtain domain level class diagram | 5 |
| 3.2 | Behavioral design |  |
| 3.2.1 | Activity diagram | 4 |
| 3.2.2 | Sequence diagram | 2 |
| 3.3 | UI design | 5 |
| 3.4 | DB design | 4 |
| 4 | **Implementation** | **22** |
| 4.1 | Code | 22 |
| 5 | **Testing** | **7** |
| 5.1 | Unit test | 2 |
| 5.2 | Integration test | 2 |
| 5.3 | Functional test | 2 |
| 5.4 | Non-functional test | 1 |
| 6 | **Documentation** | **11** |
| 6.1 | User manual | 2 |
| 6.2 | Final report | 2 |
| 6.3 | Presentation material | 2 |
| 6.4 | Reviewing | 2 |
|  |  | 3 |
|  |  |  |

## Milestone

Milestones are used to mark specific progress points on the development timeline. It focuses on major progress points that must be reached to achieve success

Figure 7 Milestones

Here, this project has six major milestone. These major milestones represents phases of Waterfall method, which are, discusses below:

**Planning – End date 9th April 019**

This milestone is for planning search and review, scope of project and writing proposal. After completion of final proposal official, the planning phase is complete. For this I have allocated 16 days for proper planning and reviewing.

**Analysis – End date 8th May 019**

This phase starts after completion of planning phase and after submitting final proposal, which must be approved by upper level management. This milestone covers the analysis part of project. Analysis part like functional requirements, non-functional requirements and use case diagram will be covered in this section. Since this phase needs to be looked carefully so I have separated 29 days.

**Design – End date 3rd June 019**

After completing analysis phase of this project, design part will be enrolled. This part covers design part of system. Database design, user interface design, development of activity diagram and sequence diagram will be created. Design needs to be user friendly and easy to use so this phase need to have 25 days.

**Implementation – End date 24th June 019**

The implementing code for the project will be done in this part. This part comes after design and entire coding part is done here. For design pattern MVC is used. For implementing I have separated 22 days out of total days since it take lots of time.

**Testing – End date 1st July 019**

After whole implementation phase is finished, project is passes to QC where all code is tested. Here unit, integration, functional, non-functional testing will be carried out. For this phase, 7 days have been allocated because many testing have to be carried out.

**Documentation – End date 12th July 019**

Here all documentation related to this project will be prepared along with user manual and final report. For documentation I have separated 11 days for creating user manual and final report.

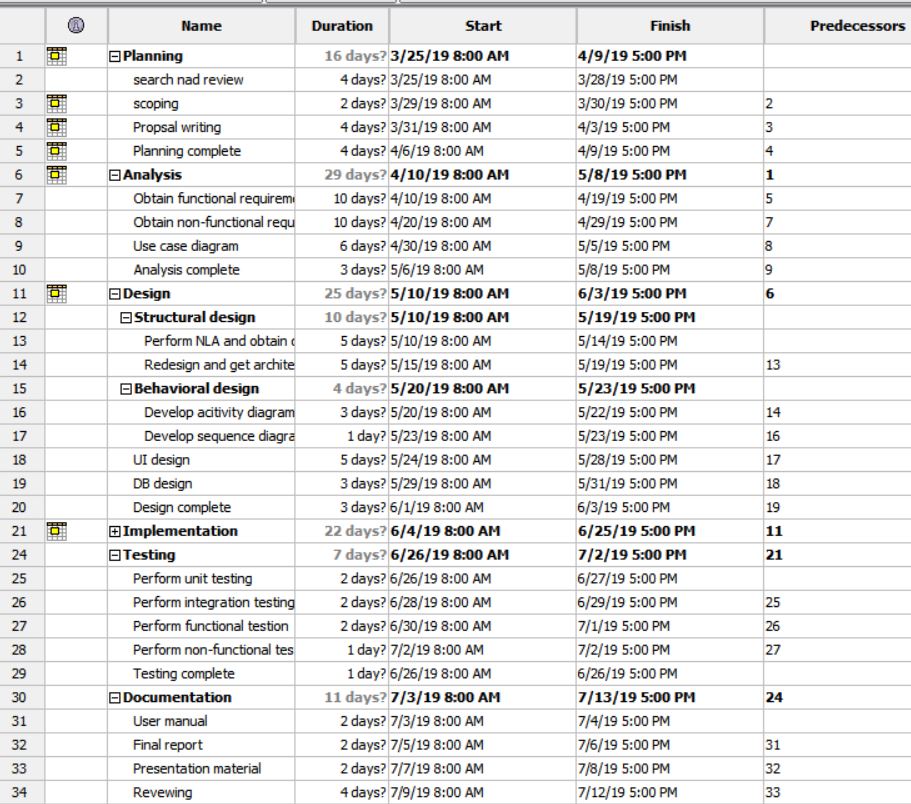
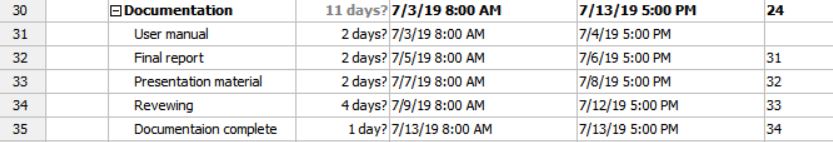


Figure 8Project schedule2

Figure 9 Project schedule1

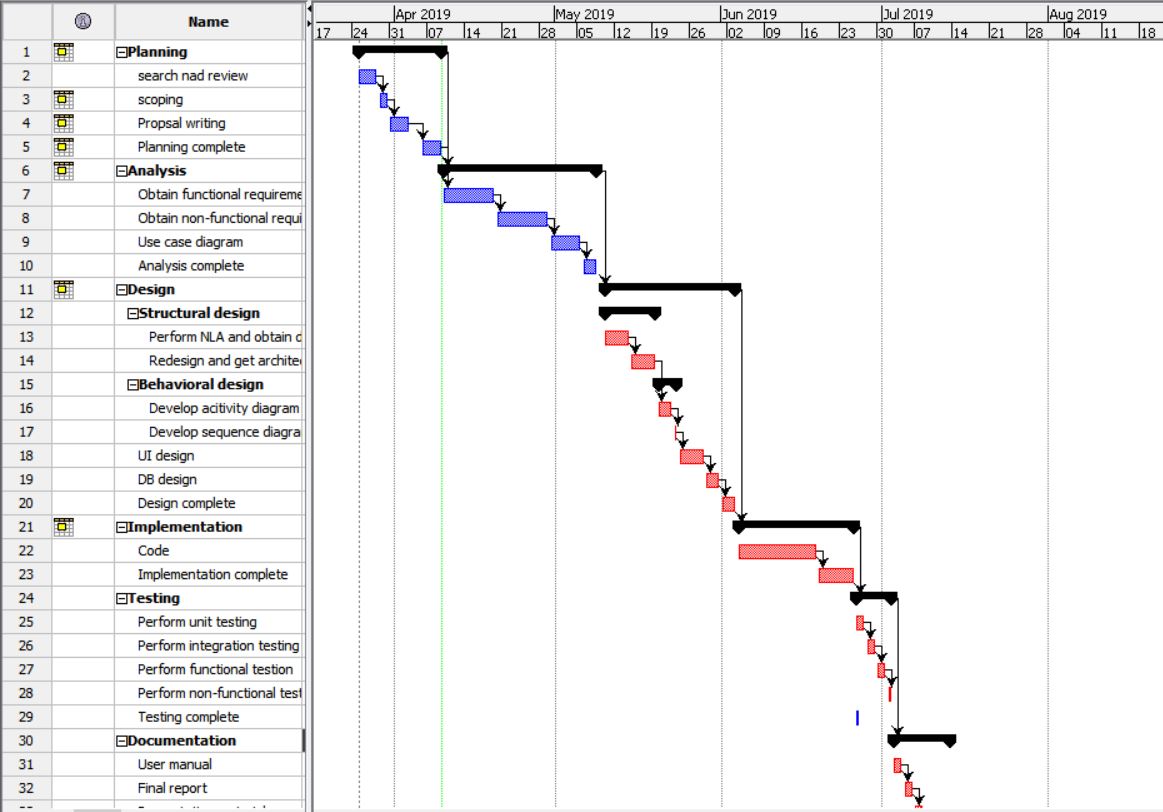


Figure 10 Gantt chart 1

Figure 11 Gantt chart 2

# Risk management

Risk management is a way of identifying and evaluating the risk. Many risk may arise that could interrupt the flow of work. For such kind of instance, a guideline must be prepared for tackling those risks.

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Consequence** | **Impact** | **Action** |
| Performance degrade due to hardware compatibility | 2 | 4 | 8 | Upgrade hardware |
| System bugs | 2 | 4 | 8 | Release of patches and new updates |
| Requirements change | 2 | 5 | 10 | Regular contact with clients and user |
|  |  |  |  |  |
| Natural disasters | 1 | 5 | 5 | Regular backup data and |

# Configuration Management

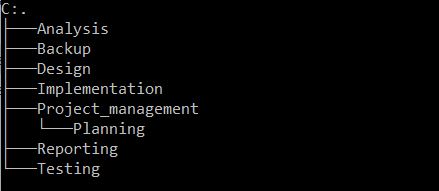
Configuration of tracking management is a process tracking the changes made in software. If any problem arise in software, then it helps to identify the root cause of problem so problem can be brought down easily.

Figure 12 Configuration

I have used GitHub for version control where my user id is <https://github.com/saanam>

All the files and folder of above will be uploaded in GitHub where version control will be applied. Any roll back or change to back can be easily done since all files will be uploaded in backup folder.

# Conclusion

This document have described all the project scope, aim, limitation, objectives, project schedule, development methodology and many other various important topic. Now this project is ready to be prepared and all the necessary resources are ready.

# Chapter 7 – References/ Bibliography

Anon, (n.d.). Programme and project configuration management | Department of Finance. [Online] Available at: https://www.finance-ni.gov.uk/articles/programme-and-projectconfiguration-management.

Anon, (n.d.). The Work Breakdown Structure Project Management. [online] Available at: http://2020projectmanagement.com/2013/10/the-work-breakdown-structure-wbs/.

Anon, (n.d.). What is MVC? Advantages and Disadvantages of MVC - Interserver Tips. [Online] Available at: https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/.

Kienitz, P. (2018). The pros and cons of Waterfall Software Development | DCSL Software Ltd. [online] DCSL Software Ltd. Available at: https://www.dcslsoftware.com/pros-cons-waterfallsoftware-development/ [Accessed 4 Apr. 2019].

Anon, (2018). [Online] Available at: https://www.brokerlink.ca/blog/risk-management/ [Accessed 4 Apr. 2019].