# Chapter 1: Introduction

# 1.1 Introduction to the system

The society we live in is a chaotic mess. Most of the people resist getting better organised. It is difficult to remember all the tasks that are to be done in our busy schedule. Meetings, To-do list, medicine etc. are curtail things that should be reminded. My project aims to remind user in a smart and simple way so that they never miss out on a thing.

# 1.2 Background to the system

People are so occupied in different things that they barely remember things they have to do. Everyone is afraid about forgetting important things like meeting, medicine, picking up children from school etc. My project will be able to solve the problem up to some extent.

# 1.3 Justification

I have proposed this system because my system is going to remind people to do their task in smart and interactive way so they never miss out on a thing. The task to be reminded and time will be stored in the database that can easily be extracted and displayed at the desired time by the user.

# Problem statement

Every people should remember their basic duties or things they have to remember. People have a lot ot issues remembering things as they wish for. My application is going to let people set reminder for things they want and store to-do lists as well. This application is going to be efficient and practical in field of use.

# 1.4 Overview of the system

My system will help user to store any reminder, they want the system to remind them about. The application will be very simple to understand but loaded with powerful features. Also a list of to-do can be stored before going to shopping or anything that has list of items to be stored as a handy note. The system will be standalone that does not require internet connection however frequent updates will be rolled out that clears all the bugs and better user interface with time.

# Chapter 2: Scope

# 2.1 Aims of the project

The aim of the project will be to provide user with a application that lets user set reminder for specific day with time. The system is going to be simple and efficient. Also, user can store to-do list if they must store list of groceries, household utensils etc. A lot of tasks given by user will be stored inside the database securely. The system will be light in size that does not take huge storage space. All the stored tasks will be reminded in a smart way to the user for more comprehensive way of getting reminded smartly. The stored data can be edited, deleted easily anytime according to the user’s desire.

# 2.2 Objective of the project

1. Creating a user-friendly design
2. Making HCI as simple as possible
3. Storing user’s data securely
4. Reminding user as per their data entry
5. Creating responsive buttons with good colour combination
6. Making to-do list simple yet strong
7. Letting user easily access the application

# 2.3 Features to be included

1. Icon to choose whether to add reminder
2. Icon to add to-do list
3. Icon to add new alarm
4. Let user choose what colour of UI they want
5. Whether or not to play sound for reminder
6. Swipe left to delete task
7. Swipe right to edit task

# 2.3 Overview of scope

My project is going to fulfil all the features and provide user with ease of use and simplified system. The UI is going to be attractive for user to interact with. There are a lot of exciting features inside the system but there are some limitations as well.

# Chapter 3: Development methodology

# 3.1 Methodology to be used

Waterfall model is the first Software Development Life Cycle that was used widely in order to ensure the success rate of the project. It is also referred to as a linear-sequential cycle model. It is very simple and easy to understand and use. The development process is linear meaning that any development phase can only start if the previous development phase is complete. In this type of development, the phases do not overlap one another.

In my project I am going to be using Waterfall model as the methodology to develop the system. Since it is simple to understand and easy to use for small scale projects. Also, there is no ambiguous requirements, all the features and budget has already been fixed which are not going to change in near future. When using this model at the first stage the requirement is gathered as per the system to be developed. After that the system is designed as per the requirement. And then coding is done to implement all the features that are to be included inside the system. Once the system is developed it is then taken to testing phase where the system is tested for bugs and errors. After testing the system is deployed in use to the client. At last after the system is deployed it will be maintained in maintenance phase rolling out new updates and fixing bugs.



Figure: Waterfall model

# 3.2 Design pattern

MVC stands for Model-View-Controller pattern. It consists of presentation information, control information and data model. The model contains pure application of data, but it does not contain any logic on how to present data to user. The view is only concerned about how to present data to the user. It has no idea about what the data means or how can it be manipulated. The controller acts as bridge between view and model. The events triggered by user are accessed from model and showed in the view that the user understands.

Using MVC pattern has a lot of advantages in it. Multiple developers can work on the model, view and controller at a time. View can be modified in user presentable way. This design pattern is really suitable for the development of my project.



Figure: MVC pattern

# 3.3 System architecture

In my project I will be using 3-tier architecture. It is a client-server architecture in which the data access, computer data storage, user interface, functional process are developed and maintained as independent modules on separate platform. In this architecture any one of tier can be upgraded or replaced independently. It is very flexible and the system we develop will be faster. The tiers in 3-tier architecture includes Presentation tier, Application tier and Data tier.

1. Presentation Tier: It is the front-end layer that consists of user interface. The UI is usually graphical one. This tier is built with the help of JavaScript, CSS, HTML etc.
2. Application Tier: This tier contains the functional business logic which drives an applications capability. It is usually written in C++, C#, Java.
3. Data Tier: This tier consists of data storage/database of the system. Example includes MySQL, MongoDB etc.

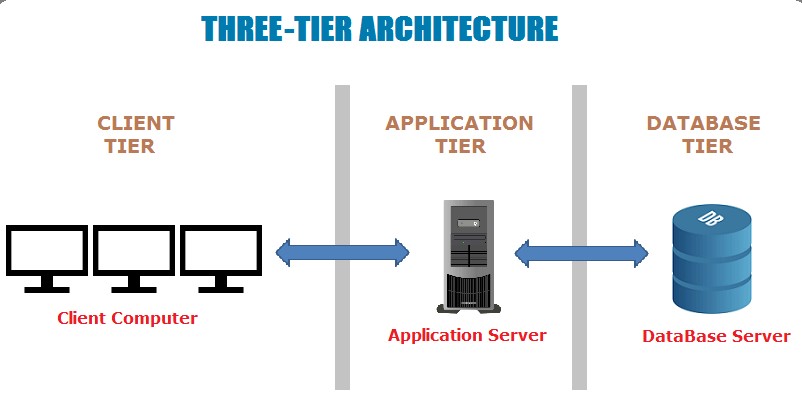


Figure: Three-Tier architecture

# Chapter 4: Work Breakdown Structure

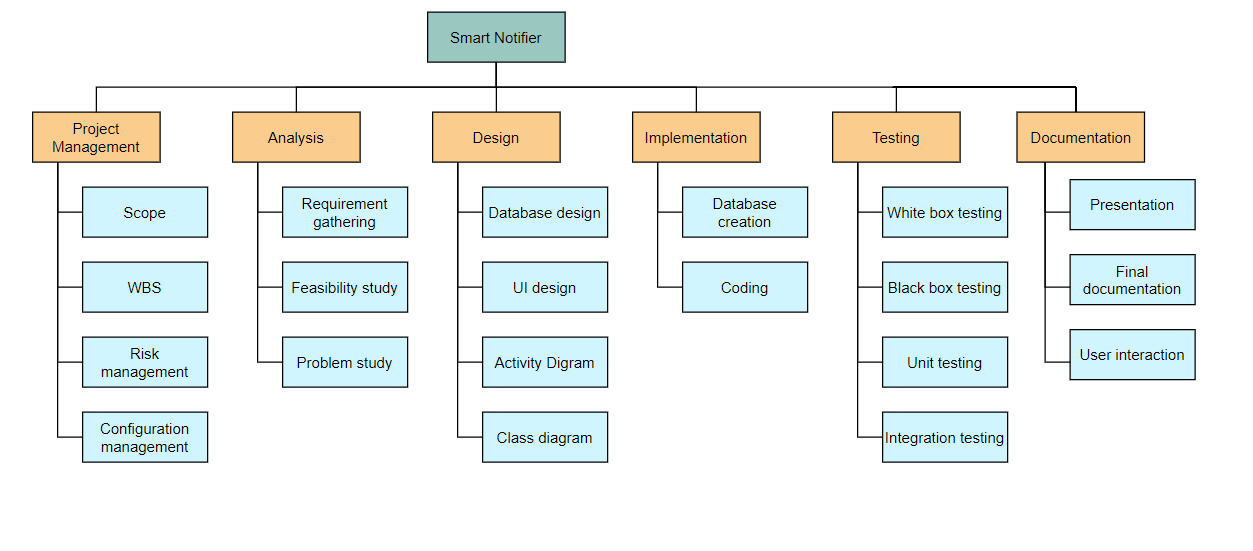


Figure: Work Breakdown Structure

# 4.2 Milestone

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Milestone** | **No. of days** | **Date (2019)** |
| 1 | **Project management**  Scope  WBS  Risk management  Configuration management | 16 | **June 16 - July 1**  June 16 - June 19  June 20 – June 22  June 23 – June 27  June 28 – July 1 |
| 2 | **Analysis**  Requirement gathering  Feasibility study  Problem study | 28 | **July 2 – July 29**  July 2 – July 12  July 13 – July 20  July 21 – July 29 |
| 3 | **Design**  Database design  UI design  Activity diagram  Class diagram | 31 | **July 30 – Aug 29**  July 30 – Aug 9  Aug 10 – Aug 15  Aug 16 – Aug 22  Aug 23 – Aug 29 |
| 4 | **Implementation**  Database creation  Coding | 22 | **Aug 30 – Sept 20**  Aug 30 – Sept 5  Sept 6 – Sept 20 |
| 5 | **Testing**  Whitebox testing  Blackbox testing  Unit testing  Integrated testing | 10 | **Sept 21 – Sept 30**  Sept 21 – Sept 23  Sept 24 – Sept 26  Sept 27 – Sept 28  Sept 29 – Sept 30 |
| 6 | **Documentation**  Presentation  Final documentation  User Interaction | 23 | **Sept 30 – Oct 12**  Sept 30 – Oct 3  Oct 4 – Oct 9  Oct 10 – Oct 12 |

Figure: Time estimation for the whole project

# 4.3 Gantt Chart

It is a type of bar chart that illustrates a schedule of a project. The vertical axis represents the tasks to be performed and horizontal axis represents the time interval for the project. It is helpful when doing project work because it helps to estimate time for each part of the work.

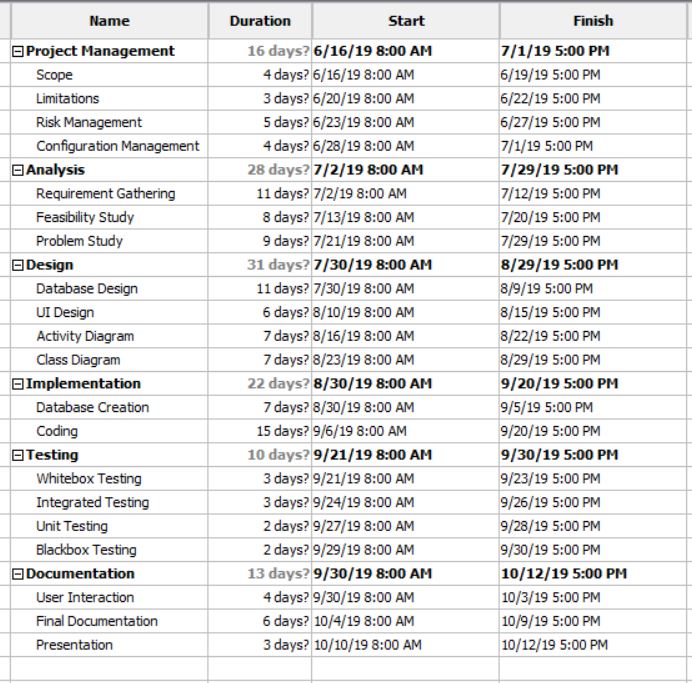


Figure: Time estimation for Gantt chart

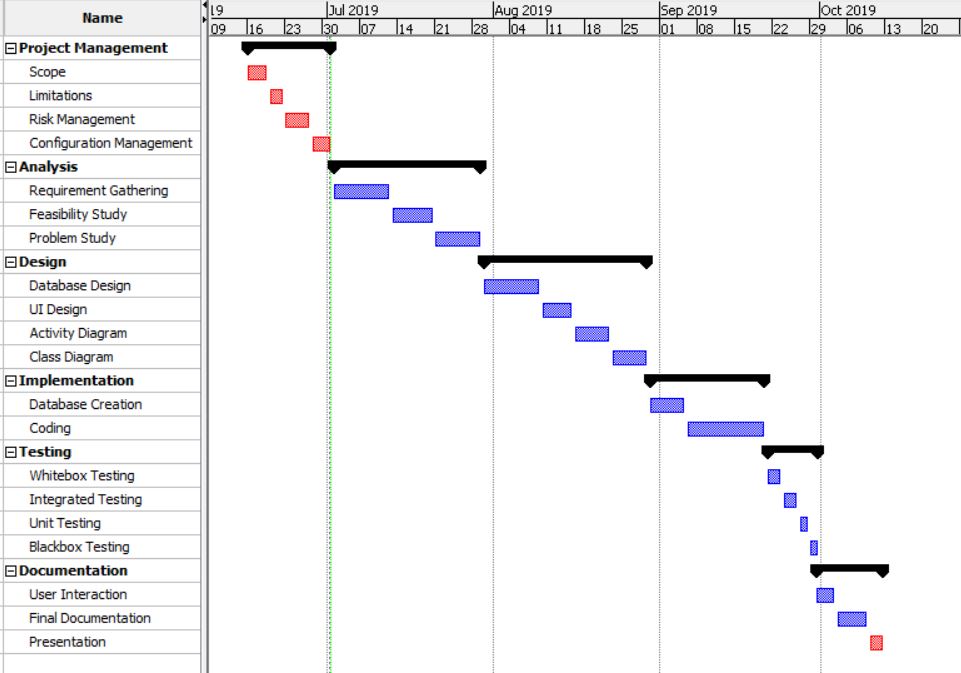


Figure: Gantt chart

# Chapter 5: Risk Management

Risk management is the technique of analysing, identifying and overcoming the risks throughout the development of the process. Risk can be highly minimized by this technique and the success rate of project becomes high.

Following are the basic tasks that risk management technique includes: -

1. To identify the risk at early stage
2. To analyse the consequence of the risk
3. To evaluate the consequence caused by the risk
4. Categorize risk impact from highest to lowest
5. Develop ways to minimize risk as far as possible

The value of likelihood is listed below: -

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

Figure: Table of likelihood

The value of consequences is listed in the table below: -

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

Figure: Table of censequences

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SN** | **Risk** | **Likelihood** | **Consequence** | **Impact** | **Action** |
| 1 | Budget and time limitation | 3 | 5 | 15 | Time and budget should be allocated and managed properly |
| 2 | Server failure | 2 | 5 | 10 | Backup server and power supply must be consistent |
| 3 | Hardware failure | 3 | 5 | 15 | Data must be backed up and maintained accordingly |
| 4 | Change in requirement | 1 | 3 | 3 | Requirements must be gathered before starting the project |
| 5 | Complexity using technology | 2 | 3 | 6 | New technology should be introduced and train the users accordingly |
| 6 | Natural disaster | 1 | 5 | 5 | Backup of services must be done and backup power supply must be active all times |

Risk = Likelihood \* Consequence

Figure: Risk Management

# Chapter 6: Configuration Management

Configuration management helps companies to systematically manage, control and organize the changes in the documents, codes and other entities during SDLC. It aims to control cost and work effort involved in making changes to the software system. The main goal of configuration management is to increase productivity with less possible mistakes. Whenever there is change in project requirements the organisation is expected to publish versions of each log.

Following are the reasons for using configuration management in the system: -

1. Rules and regulations might be updated
2. Helps to identify bad configuration changes
3. Redundancy of the project is reduced
4. Helps developer in debugging

# 6.1 Version Control

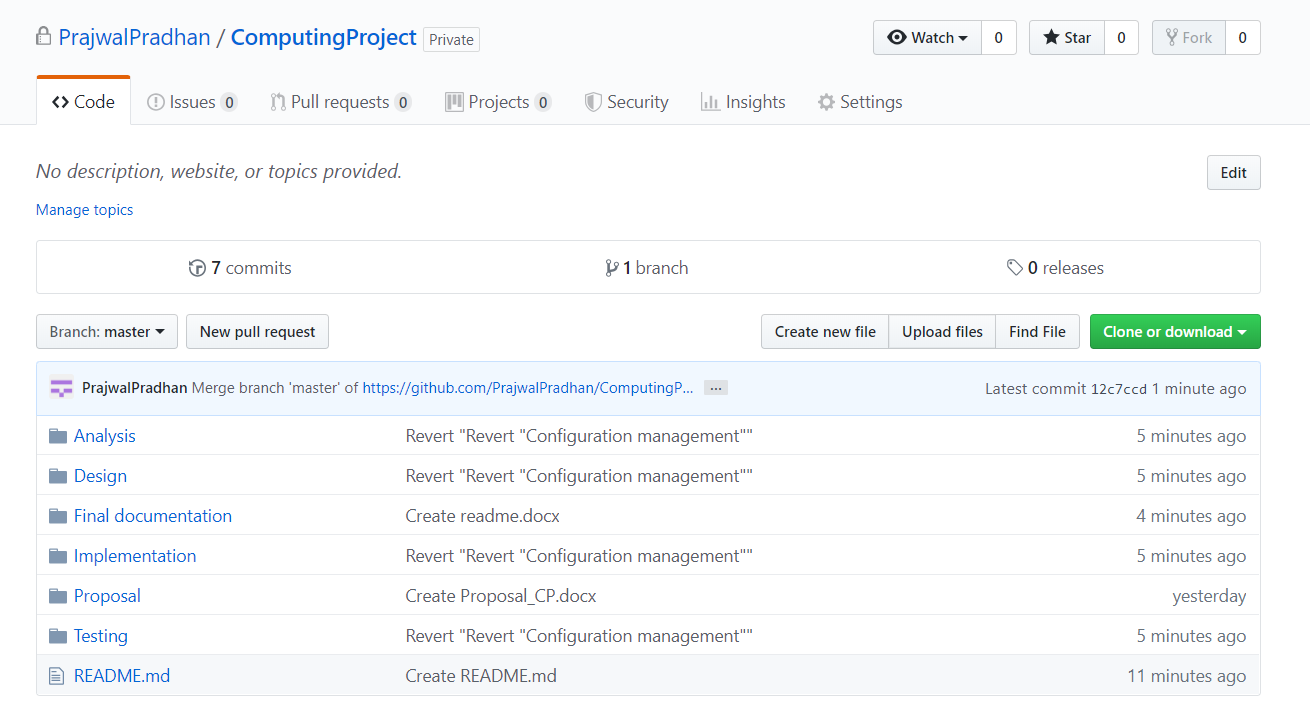


Figure: My project pushed on GitHub

Github is a online platform that helps to store software development version using Git. I have created my id named ‘PrajwalPradhan’ and pushed all my files there.

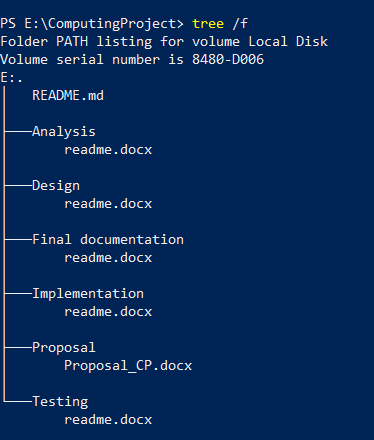


Figure: Tree of my backup files in my laptop

# Chapter 7: Conclusion

Therefore, overview of my project ‘Smart Notifier’ has been shown in this proposal. The project is well planned and organised that is going to meet all the requirements. Work breakdown structure helped me to break down the project into different parts. With the help of milestone, I have specified specific time for each structure. Gantt chart helped to show overview of the project and the time and total days required to finish my project.

The potential risks that may occur have been minimized with the help of Risk management. Configuration management has been implemented to track the version control and change logs of the project. In order to finish project efficiently, different pattern and cycle have been followed. The project will meet all the requirement and will be deployed as per discussed.

# References and Bibliography

1. www.tutorialspoint.com. (2019). *SDLC Waterfall Model*. [online] Available at: https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm [Accessed 1 Jul. 2019].
2. GeeksforGeeks. (2019). *MVC Design Pattern - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/mvc-design-pattern/ [Accessed 1 Jul. 2019].
3. Workbreakdownstructure.com. (2019). *What is a Work Breakdown Structure (WBS) - Workbreakdownstructure.com*. [online] Available at: https://www.workbreakdownstructure.com [Accessed 1 Jul. 2019].
4. GitHub. (2019). *Build software better, together*. [online] Available at: https://github.com [Accessed 1 Jul. 2019].
5. Izenda. (2019). *5 Benefits of a 3-Tier Architecture - Izenda*. [online] Available at: https://www.izenda.com/5-benefits-3-tier-architecture/ [Accessed 1 Jul. 2019].