Table of Contents

[Introduction 1](#_Toc12910946)

[1.1 Introduction to the system 1](#_Toc12910947)

[1.2 Background of the system 1](#_Toc12910948)

[1.2.1 Problem statement 1](#_Toc12910949)

[1.3 Justification 1](#_Toc12910950)

[1.4 Overview of the purpose of the system 1](#_Toc12910951)

[Scope 1](#_Toc12910952)

[2.1 Aims of the project 1](#_Toc12910953)

[2.2 Objectives of the system 1](#_Toc12910954)

[2.3 Features of the system 1](#_Toc12910955)

[2.4 Overview of the scope 2](#_Toc12910956)

[Development methodology 2](#_Toc12910957)

[3.1 Methodology to be used 2](#_Toc12910958)

[3.2 Design pattern 3](#_Toc12910959)

[3.3 System architecture 4](#_Toc12910960)

[Scheduling 5](#_Toc12910961)

[4.1 Works breakdown structure 5](#_Toc12910962)

[4.1.1 Analysis 5](#_Toc12910963)

[4.1.2 Design 5](#_Toc12910964)

[4.1.3 Testing 6](#_Toc12910965)

[4.1.4 Codding 6](#_Toc12910966)

[4.1.5 Deployment 6](#_Toc12910967)

[4.1.6 Maintenance 6](#_Toc12910968)

[4.2 Milestones 8](#_Toc12910969)

[4.3 Gantt chart 9](#_Toc12910970)

[Risk management 10](#_Toc12910971)

[Configuration management 11](#_Toc12910972)

[6.1 Version control 12](#_Toc12910973)

[6.2 Release management 14](#_Toc12910974)

[6.3 Change management 15](#_Toc12910975)

[Conclusion 15](#_Toc12910976)

[References 15](#_Toc12910977)

# Introduction

## Introduction to the system

**“Doctor Information system”** provides information of the doctors available in our locality. It is web-based information system. I would like to use Laravel framework for the development of this project. It will add information of doctor in near area and update and delete information if needed by admin. It will provide information of doctor if user search the information. User could able to provide feedback after services provided by doctor.

## Background of the system

Reason behind choosing the Laravel framework for the development of the project is it provide more flexibility, structure, efficient, easier for the development. The main reason for choosing this project is many of people faces a lot of problems because of lack of information of doctors. They travel more in search of good doctor for the treatment of disease.

### Problem statement

People faces lots of problem due to lack of information of doctor available in the locality.

## Justification

It will reduce the cost of travelling and search of good doctor by seeing the people chooses of given rating. It will save time too. It will increase the dependency on technology. Since it has easier interface so will provide efficiency. Since it is cost, time effective project it will be appropriate to do.

## Overview of the purpose of the system

It will be simple and informative information system. It will be cost reducing and time reducing product. It will motivate people to use technology and increase technology dependency.

# Scope

## 2.1 Aims of the project

* To provide proper doctors information about their availability and their specialty

## 2.2 Objectives of the system

* To make the Effective system
* To make easier interface
* To Increase efficiency
* To increase the Availability
* To provide Reliable information
* To reduce the cost
* To save the time

## 2.3 Features of the system

* Provide easier interface for login users
* Give the feedback
* Contact doctor with email
* Take the appointment
* Get information of the doctor’s specific degree
* Able to provide rating to the doctor after they provide facility

## 2.4 Overview of the scope

It will have simple and easy to use interface. It will be cost and time reducing, it increases the efficiency. It has easier goal.

# Development methodology

## 3.1 Methodology to be used

For the development of this product it will appropriate to use Agile methodology. Since, it will be the new project to develop and all the requirement are not conformed well or not known properly. It helps in rapid development of application. It even accepts the late requirement change. Most used agile methodology are mention below: (Kelly, 2019)

* Agile Scrum methodology
* FDD (Features Driven Development)
* XP (Extreme Programming)
* Lean Software Development
* Kanban
* Crystal
* DSDM (Dynamic System Development Method)
* ASD (Adaptive System Development)

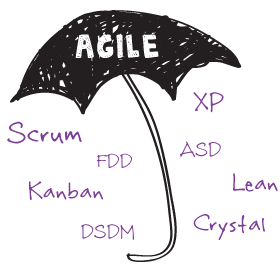


Figure agile methodologies

Among them I will like to use DSDM (Dynamic System Development Methodology). DSDM primary criteria for delivery and acceptance of the product, focuses on 80% useful of the product that can be deploy in 20% of time.

DSDM eight key principles:

* Focuses on business need
* Delivery on time
* Collaborative
* Never compromise with quality
* Develop iteratively
* Build incremental
* Communicates continuously and clearly
* Demonstrate control



Figure : DSDM Diagram

Reason for choosing this approach of agile for the development of the product:

* System can be delivery at time
* Basic functionality can be quickly developed or delivered and functionality can be developed at regular interval of time.
* More user involvement.

## 3.2 Design pattern

I will like to use MVC (Model View Controller) design pattern for this project. (Chakraborty, 2018) In MVC design pattern whole project will be divided in three main logical component Model, View and Controller. Model component is related to all the data-related to logic that user works with. View component is related to all the UI logic. Controller works intermediate between model and view. It handles the user request by processing data logic (model) and interact with view.



Figure : MVC structure

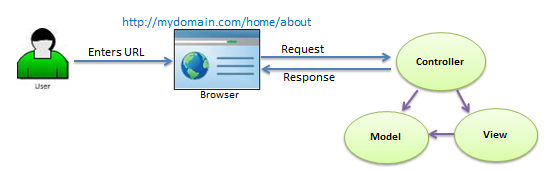


Figure : detailed of MVC

I will be using this design pattern because of these reasons which are given below;

* It decreases the complexity of program.
* It increases the maintainability of the program.
* We can easily find out where is the problem while something went wrong.

## 3.3 System architecture

For the system architecture I will like to use **3-tier** architecture because a tier can be changed or relocate without affecting other tier and makes easier to continually evolve the application as new need or requirement and opportunities arises. 3-tier architecture is modular client server architecture that consists of a presentation tier, data tire, application tier. (Gilbert & Rouse, 2019)

**Presentation tier:** this tier is built with HTML5, CSS and JavaScript are deployed to computing device with web browser. It communicates with other tiers with the help of API (Application Program Interface) call.

**Data tier:** Data tier is referred as storage tier. It consists of database and program for performing data read and write operation.

**Application tier:** it is also known as logical tier which is written in programming language such C#, java and contains business logic that can hold up application’s core function.

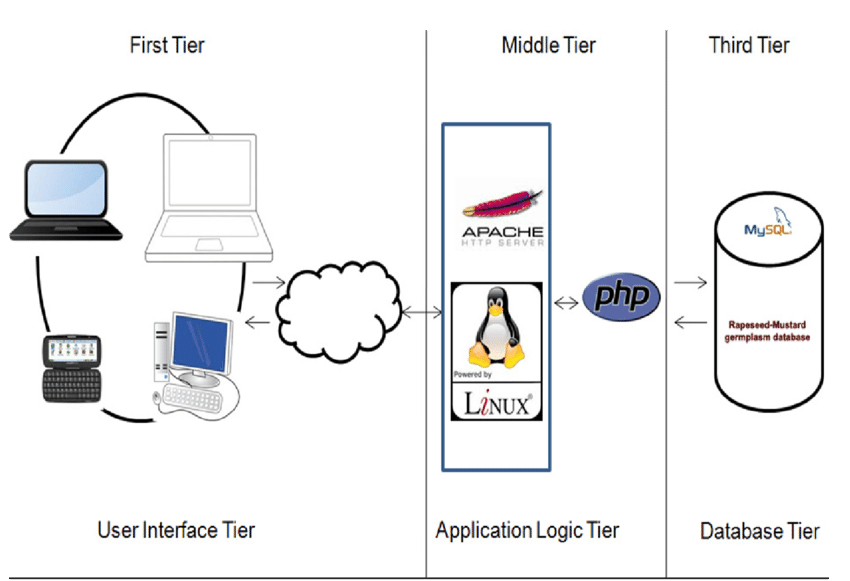


Figure : 3-tier architecture

# Scheduling

## Works breakdown structure

It is the process of dividing the work in different stages. So, that it is easier to study and maintain and easily document the whole project.

### Analysis

In this we analyze about feasibility of the product and the requirement specification, prioritize requirement. We study cost, time feasibility. In this portion we also gather all the requirement.

### Design

Considering the requirement specification from first stage (Analysis), system design is produced. It helps to define all system architecture and even helps to find out system requirement.

### Testing

It should be done before the implementation of code in TDD (Test Driven Development). It is the way of writing test cases first and until and unless test cases pass, we should not go to implementing the code.

### Codding

It is simply process of implementing codes. System are developed in part wise some portion and functionality and gave the user for feedback, if positive feedback then increases the product otherwise improve first as user want and then increase the system.

### Deployment

It is the way deploying the system in user desktop.

### 4.1.6 Maintenance

If any codes/requirement need to be change in future then it comes in this portion. It is simply process of improving the codes and adding the new functionality which makes the system to adapt the new changes and able to survive on market.

Doctor information system

Testing

Others

Implementation

Design

Analysis

Proposal

Unit Testing

introduction and scope

Use Case

Admin section

Others project issue

UI Design

Requirement gathering

Integration testing

Database Design

Development methodology and scheduling

User section

Documentation

Structural model

Acceptance testing

Interface Analysis

Risk and configuration management

Architecture

Finalizing the proposal

Figure : WBS diagram

## Milestones

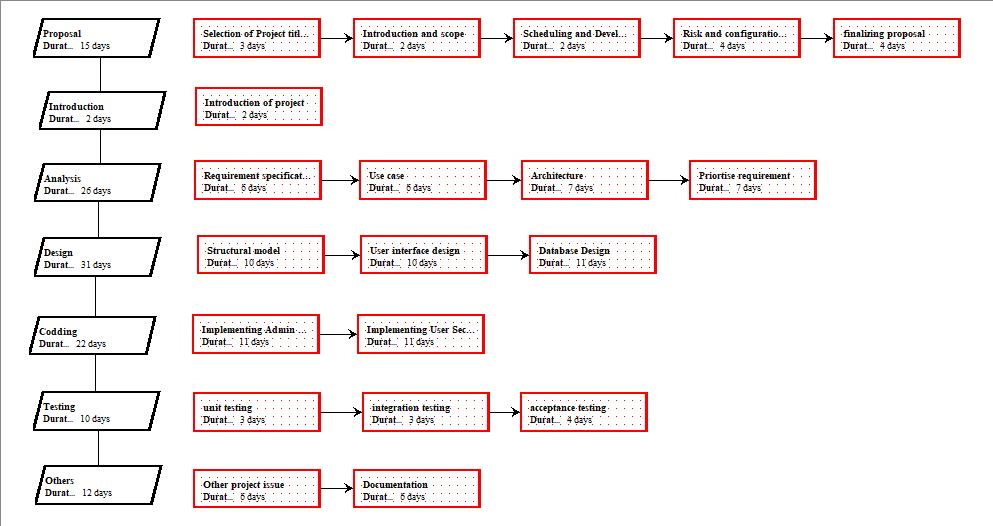


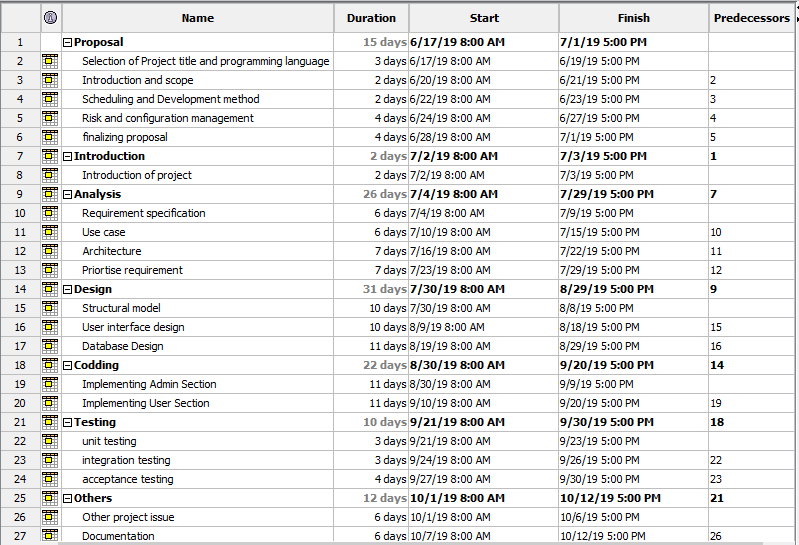
Figure : Milestone extracted from libre

|  |  |  |
| --- | --- | --- |
| **Milestones** | **No. of Days** | **Date (2019)** |
| **Proposal**  Selection of title and language  Introduction and scope  Development methodology  Risk and configuration mgmt.  Finalizing proposal | **15** 3 2 2 4  4 | **June 17th – July 1st,**  June 17th – June 19th  June 20th – June 21st  June 22nd – June 23rd June 24th – June 27th June 28th – July 1st |
| **Introduction**  Introduction of the project | **2**  2 | July 2nd – July 3rd  July 2nd – July 3rd |
| **Analysis**  Requirement specification Use case  Architecture  requirement prioritization | **26** 6  6  7  7 | **July 4th – July 29th** July 4th– July 9th July 10th– July 15th July 16th– July 22nd July 23rd– July 29th |
| **Design**  Structure Design UI Design Database Design | **31** 10  10  11 | **July 30th – August 29th** July 30th –August 8th August 9th – August 18th August 8th – August 29th |
| **Implementation** Admin section User section | **22**  11  11 | **August 30th - Sept 20th**  August 30th – Sept 9th Sept 12th –Sept 20th |
| **Testing** Unit Testing Integration Testing Acceptance Testing | **10** 3 3 4 | **Sept 21st – Sept 30th** Sept 21st - Sept 23rd Sept 24th – Sept 26th Sept 27th – Sept 30th |
| **Others** Others project issue Documentation | **12** 6 6 | **Oct 1st – Oct 12th** Oct 1st –Oct 6th, 2019 Oct 7th –Oct 12th, 2019 |
| **Total Days** | **118** |  |

Table : milestones table

It will take **118 days** to complete the whole project.

## Gantt chart

1. 
2. Figure : Milestone

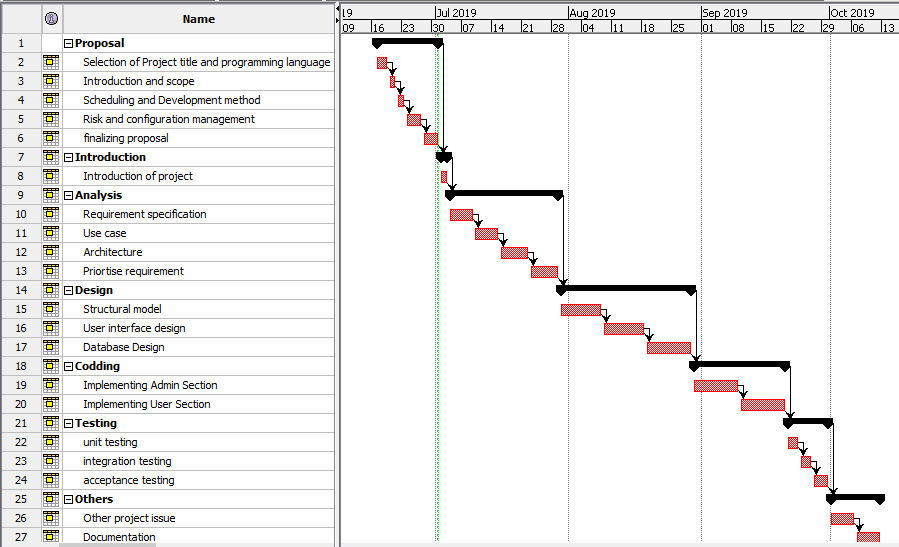


Figure : Gantt chart

# Risk management

The process of managing all the possible risk is known as risk management. It also considers as process of identification, evaluation and prioritization risks in such a way that its impact of risk can be minimized or controlled. (Cole, 2016)

Likelihood and its values are given below:

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

Table : Likelihood and its value table

Consequences and its values are shown:

|  |  |
| --- | --- |
| Consequences | Values |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

Table : Consequences and its value

Impact =likelihood \* Consequences

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| R\_id | Risk description | Likelihood | Consequences | Impact | Action |
| 001 | Hard drive crash | 1 | 5 | 5 |  |
| 002 | IDE update | 2 | 4 | 8 |  |
| 003 | SQL injection | 3 | 5 | 15 |  |
| 004 | Error occur during implementation | 3 | 2 | 6 |  |
| 005 | Project resources not found | 1 | 3 | 3 |  |
| 006 | Mis direction of the project | 1 | 4 | 4 |  |
| 007 | Timeline risk | 2 | 4 | 8 |  |
| 008 | Maintain security | 1 | 5 | 5 |  |

Table : Risk management table

# Configuration management

Configuration management is the process of establishment and preserving consistency of the system’s performance, function and physical attributes with its design, requirement and operational information through out its life. It includes provisions for tracking, storing, and updating all the system information on the system basis. (Sharma, 2017)

Benefits of configuration management are given below

* Disaster recovery
* Uptime and site reliability
* Easier scaling
* It increases the efficiency, stability and control by improving tracking and visibility.

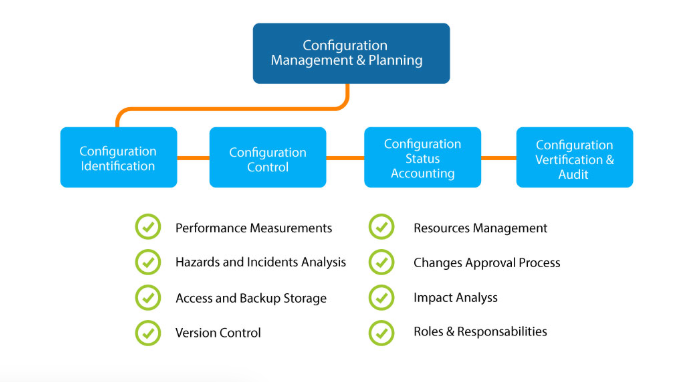


Figure : configuration management process

## Version control

For the availability of the project I have put my project in my git hub by creating the repository in my account named “yaklesh0007” and repository named “cp”. Directory link of this project is: <https://github.com/yaklesh0007/CP> .

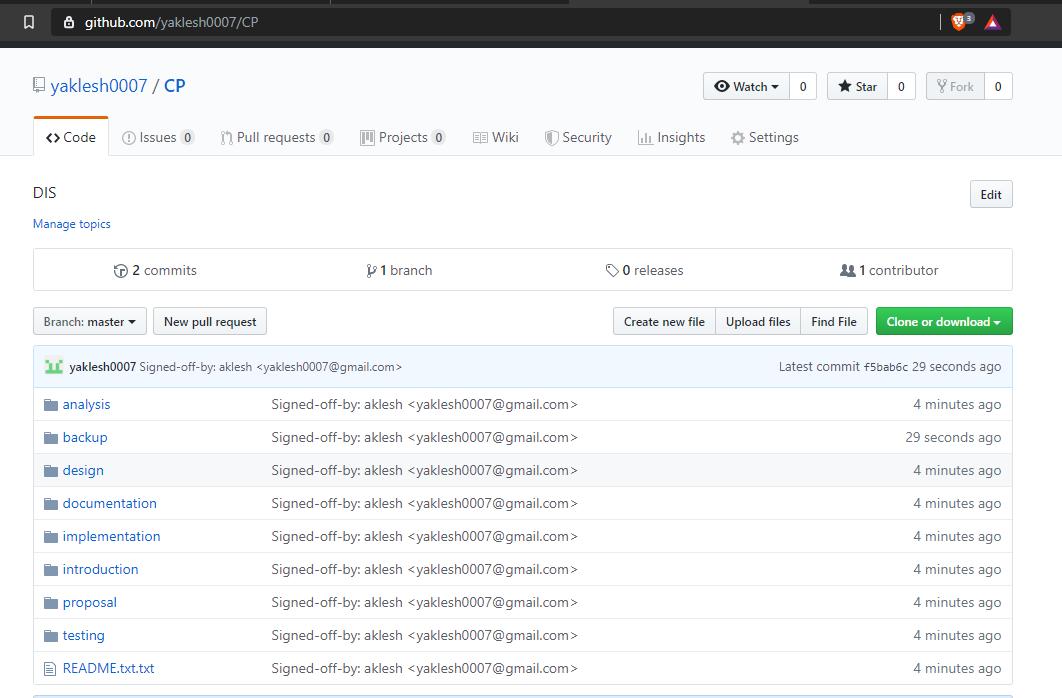


Figure : Version control GitHub directory

I have put all this folder on desktop and I have created backup folder which store other folder’s file like analysis, testing, codding, etc. for the backup purpose. Tree diagram for the directory is mention.

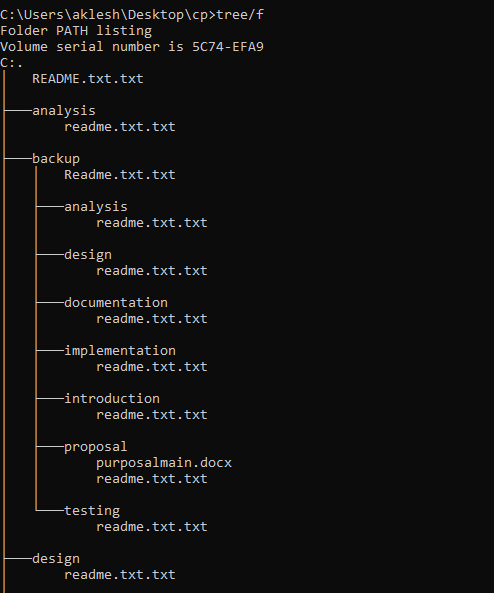




Figure : tree directory

## Release management

Release Management is the process responsible for planning, scheduling, and controlling the build, in addition to testing and deploying Releases.

There different process involves in release management that are mention below:

* Release management support
* Release planning
* Release built
* Release deployment
* Early life (post release) support
* Release closure

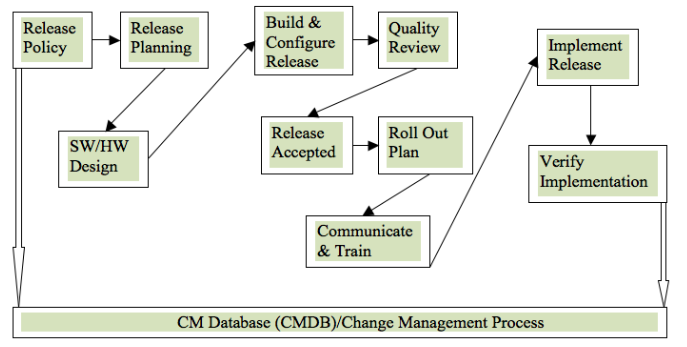


Figure : release management process

## Change management

Change management is the way that guide how to prepare, equip and support individuals to successfully adopt changes in order to drive organizational success. There is different level of change management:

* Individual change management
* Organization/initiative change management
* Enterprise change management capability

# Conclusion

It will be simple informative system that will provides information of doctors in our locality. It will help to solve the people problem that people are facing due lack of information of doctor’s details. Since it will be reducing the time of people for the search of doctors. Since, time is money, so in other way we can say cost will be also reduce/ save. Since, it is easy to manage the project, time relevant, cost effective, etc so, I will like to do this project.

# References

Chakraborty, A., 2018. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/mvc\_framework/mvc\_framework\_introduction.htm  
[Accessed 29 6 2019].

Cole, B., 2016. *TechTarget.* [Online]   
Available at: https://searchcompliance.techtarget.com/definition/risk-management  
[Accessed 29 6 2019].

Gilbert, C. & Rouse, M., 2019. *Techtarget.* [Online]   
Available at: https://searchsoftwarequality.techtarget.com/definition/3-tier-application  
[Accessed 29 6 2019].

Kelly, C., 2019. *blueprintsys.* [Online]   
Available at: https://www.blueprintsys.com/agile-development-101/agile-methodologies  
[Accessed 29 6 2019].

Sharma, l., 2017. *ToolSQA.* [Online]   
Available at: https://www.toolsqa.com/software-testing/configuration-management/  
[Accessed 1 7 2017].

[Figure 1 agile methodologies 3](#_Toc12911028)

[Figure 2: DSDM Diagram 4](#_Toc12911029)

[Figure 3: MVC structure 5](#_Toc12911030)

[Figure 4: detailed of MVC 5](#_Toc12911031)

[Figure 5: 3-tier architecture 6](#_Toc12911032)

[Figure 6: WBS diagram 8](#_Toc12911033)

[Figure 7: Milestone extracted from libre 9](#_Toc12911034)

[6 Figure 8: Milestone 10](#_Toc12911035)

[Figure 9: Gantt chart 11](#_Toc12911036)

[Figure 10: configuration management process 12](#_Toc12911037)

[Figure 11: Version control GitHub directory 13](#_Toc12911038)

[Figure 12: tree directory 15](#_Toc12911039)

[Figure 13: release management process 16](#_Toc12911040)