# Introduction

## Introduction to the system

**“Doctor Information system”** provides information of the doctors available in our locality. It is web-based information system. I will like to use Laravel framework for the development of this project.

## Background of the system

This will be the new project.

### Problem statement

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

## Justification

## Overview of the purpose of the system

# Scope

## 2.1 Aims of the project

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

## 2.2 Objectives of the system

* Effective
* Easy to use
* Increase Robustness
* Availability
* Reliability

## 2.3 Features of the system

* User login
* Admin login
* Give the feedback
* Contact doctor with email
* Take the appointment
* Get information of the doctor’s specific degree
* Give the rating

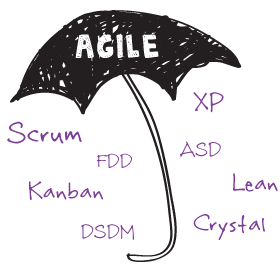
## 2.4 Overview of the scope

# 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)



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:

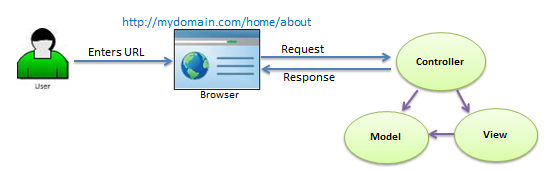
* Focuses on business need
* Delivery on time
* Collaborative
* Never compromise with quality
* Develop iteratively
* Build incremental
* Communicates continuously and clearly
* Demonstrate control



## 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.





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.

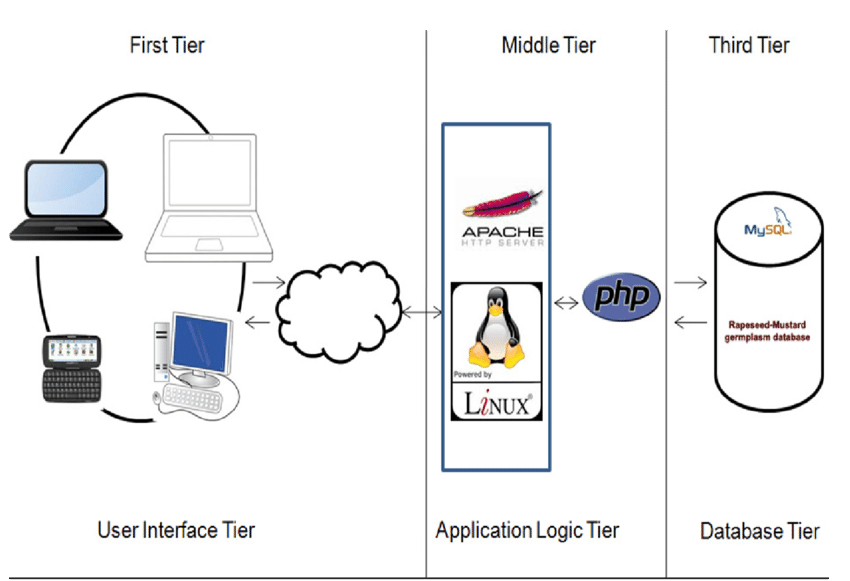
## 3.3 System architecture

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.



# 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.

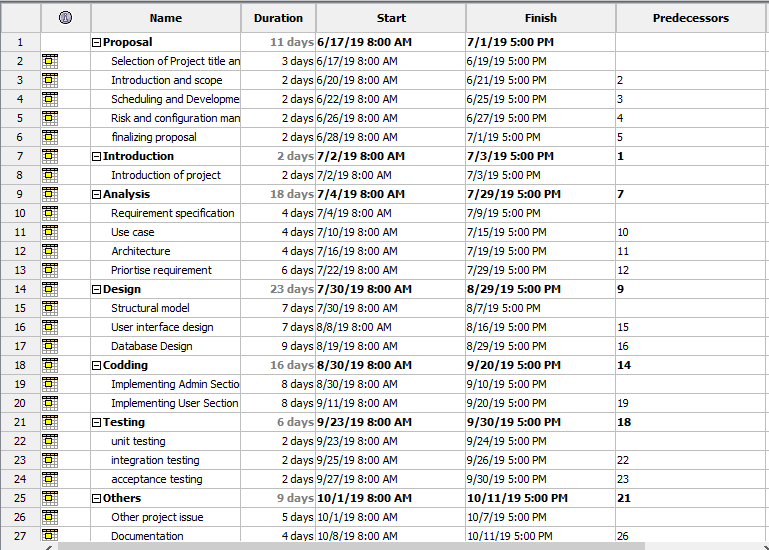
### Deployment

It is the way deploying the system in user desktop.

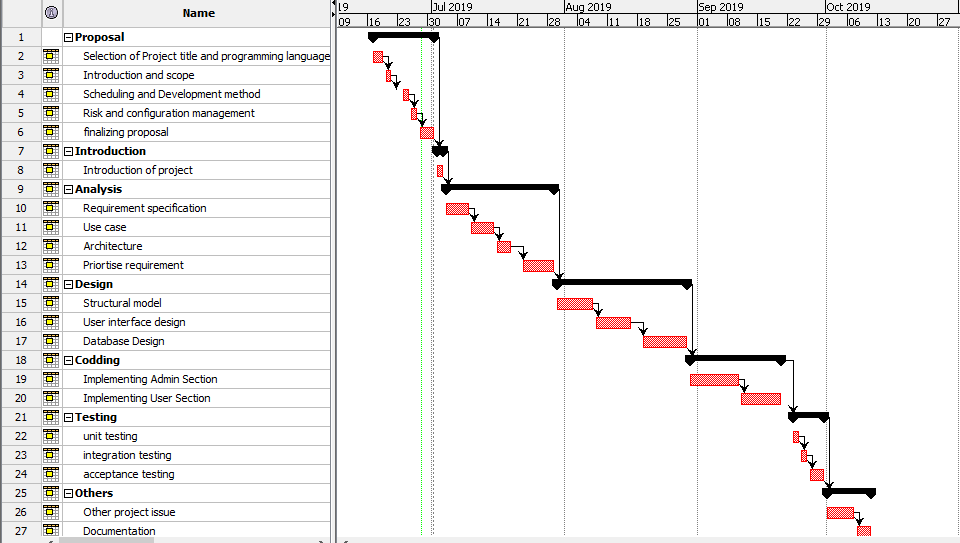
### 4.1.6 Maintenance

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.

## Milestones



## 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 |

Consequences and its values are shown:

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

Impact =likelihood \* Consequences

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| R\_id | Risk description | Likelihood | Consequences | Impact | Action | Remarks |
| 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 | Financial risk | 1 | 5 | 5 |  |  |

# Configuration management

# Conclusion

# Reference and bibliography