Project Proposal

On

**Hotel Management System**

**Bibek Dahal**

**00172931**

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT and E-Commerce

Kathmandu, Nepal

**Date: 2019/01/04**

**Submitted to: Kiran Rana**

Contents

[**1)** **Introduction** 3](#_Toc534544424)

[1.1) Project Introduction 3](#_Toc534544425)

[1.2) Justification for Project 3](#_Toc534544426)

[1.2.1) Background of the project 3](#_Toc534544427)

[1.2.2) Problem Statement 3](#_Toc534544428)

[1.3) Description of the project 4](#_Toc534544429)

[1.3.1) Feature 4](#_Toc534544430)

[**2) Project Scope** 5](#_Toc534544431)

[2.1) Scope and Limitation of Project 5](#_Toc534544432)

[2.2) Aims and Objectives 5](#_Toc534544433)

[**3) Development Methodology** 6](#_Toc534544434)

[3.1) Methodology used 6](#_Toc534544435)

[3.2) Design Pattern 7](#_Toc534544436)

[3.3) System Architecture 8](#_Toc534544437)

[**4)** **Work Breakdown Structure (WBS)/Scheduling** 9](#_Toc534544438)

[4.1) Work Breakdown Structure 9](#_Toc534544439)

[4.2) Milestones 10](#_Toc534544440)

[4.3) Scheduling/Gantt Chart 12](#_Toc534544441)

[**5)** **Risk Management** 14](#_Toc534544442)

[**6)** **Configuration Management** 16](#_Toc534544443)

[**7)** **Conclusion of the project** 17](#_Toc534544444)

[**8)** **References** 17](#_Toc534544445)

# List of figures

Figure 1 Waterfall Model…………………………………………………………6 Figure 2 MVC design pattern………………………………………………....…7 Figure 3 Three tier architecture………………………………………………....8 Figure 4 Work Break-down Structure……………………………………….….9 Figure 5 Allocating days for project…………………………………………...12 Figure 6 Gantt Chart…………………………………………………………….13 Figure 7 Repository in GitHub…………………………………………………16 Figure 8 Listing of folder path………………………………………………….18

**Introduction**

## Project Introduction

This project is based on the Hotel Management System. I will be developing an interactive web-based application that can be used for the daily transactions of hotel. I will develop this project by using language such as HTML, CSS, Bootstrap and Php. To make it robust I will be doing Php in Laravel framework.

## Justification for Project

### 

### 1.2.1) Background of the project

As technology is growing fast and people want their daily life to more productive, I have bought up concept of Hotel Management System which provides them good solution while dealing with the Hotel related activity. Primarily, Hotel Management System is focused to deliver a web-based application that can easily address the day to day activity in more efficient and effective ways.

### 1.2.2) Problem Statement

In traditional system the business process is managed with the paper type documents that are legered in book. The paper based system requires a lot of time for managing and searching the data. Also in traditional system people should go on spot to check for available rooms and details, if the room is packed then they may face wastage of time.

Therefore to overcome this problem, this project will provide the customer to enquiry about the hotel facilities remotely. Employees can manage the data quickly and securely which also helps owner to extract the business intelligence. And also with the web-based application more customer can be attracted.

## Description of the project

## 1.3.1) Feature

**The feature of this project are given bellow:**

1. **Hotel overview:** It can provide overview of the hotel through the information provided on the application.
2. **Login feature:** User can login to application for gaining additional services.
3. **Create account:** Customers can easily create their account.
4. **Update user profile:** After creating the account user can update their personal details.
5. **Reserve room:** Each user can reserve their rooms easily.
6. **Generate the bill:** The final bills can be generated after customer is ready to pay bills.
7. **Forum:** A forum is created for addressing any query about hotel.

# **2) Project Scope**

# 2.1) Scope and Limitation of Project

This project caries a huge potential for attracting customers and provides information regarding hotel. All the required task that are carried out by front-desk can be well managed with this application.

Having good potential this project also has some limitation such as:

**Payment system:** The payment should be done in cash due to lack of online payment features.

**Stock management:** This application doesn’t track any stock of the hotel.

# 2.2) Aims and Objectives

**Aim**

* Automate the daily process of the organization.
* Users should be well managed.
* With the minimum use of resources the application should work effectively.
* The application should be user-friendly and attract the customer.

**Objectives**

* **The primary focus is to create a computerized process.**
* **Credentials are provided according to their role such as admin and customer.**
* **Customer can send the request for booking.**
* **Admin can accept or reject the request sent by customer.**

# **3) Development Methodology**

## 3.1) Methodology used

## 



Figure 1 Waterfall model

I will be using Waterfall methodology as development methodology, unlike agile methodology it doesn’t require a team to complete a project. A single person can handle all the steps. Here I have described each stage of waterfall model:

**Requirement:** The all requirements of the user are noted for analyzing the project.

**Design:** This phase verifies all the document that satisfy input, output and process.

**Implementation:** In this phase a working physical system is developed including database.

**Verification/ Testing:** In this phase all each and every element of software are thoroughly tested to verify it quality. If the software meet’s up the desired goal then, application is delivered to user.

**Maintenance:** After successful delivering the product software needs to be maintained once in a while.

## 3.2) Design Pattern

Design pattern is a template that helps to provide solutions for any problems. However it doesn’t provide the exact process to be carried out. In this project I have used MVC (Model View Controller) design pattern. MVC design pattern is one of the widely followed pattern around the globe. Here I have described about MVC pattern.



Figure 2 MVC design pattern

* **Model:** The model provides insight about the Object and data.
* **View:** The view represents the object that user can generally get access to.
* **Controller:** Controller lies between the view and model that triggers the event of application which moves data around the system.

## 3.3) System Architecture

**I will be doing my project based on three tier architecture. The three tier architecture provides flexibility during client-server system and modifying them. Client lies on the presentation tier, server lies on application tier and database lies on data tier. There are several beneficial of using three tier architecture such as:**

* **The different teams can do their task related to their tier independently.**
* **It maximizes the reliability of software.**
* **Maintenance can be easier to handle without affecting another layer such as modifying the presentation layer doesn’t affect business logic.**

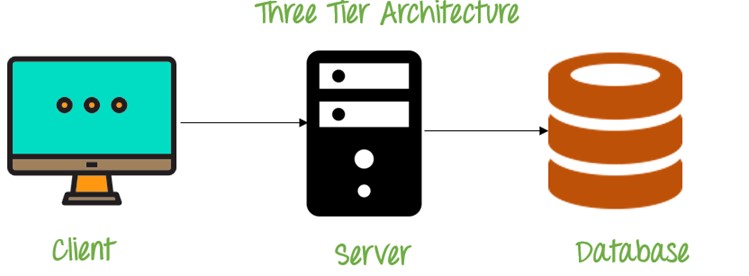


Figure 3 Three Tier Architecture

# **4) Work Breakdown Structure (WBS)/Scheduling**

### 4.1) Work Breakdown Structure

**WBS (Work Breakdown Structure) is one of the must tools needed during the project management. Work Breakdown Structure is a hierarchical structure that is formed after breaking down the large project into small task. The WBS helps to manage the complexity of system as smaller chunk and they are easier to understand. Hence, WBS also saves lot of time that can cause understating the larger project**

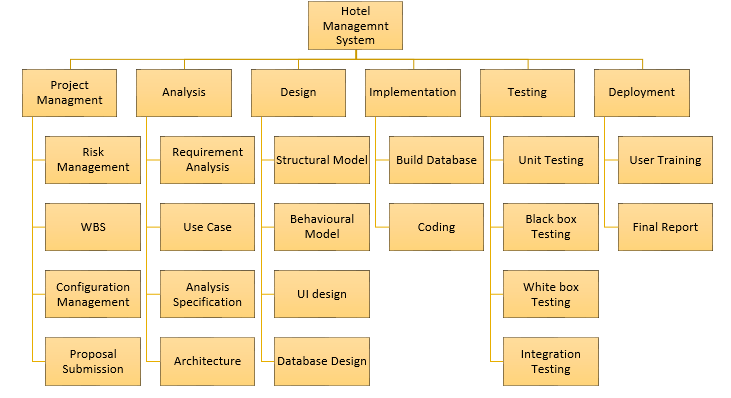


Figure 4 Work Break-down Structure

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  Work Breakdown Structure  Configuration Management  Proposal Submission | **12/21/2018 -1/3/2019**  12/21/2018 - 12/23/2018  12/24/2018 - 12/26/2018  12/27/2018 - 12/30/2018  12/31/2018 - 1/3/2019 |
| **Analysis**  Requirement Analysis  Use Case  Architecture (Initial Class Diagram)  Analysis Specification | **1/4/2019 - 1/28/2019**  1/4/2019 - 1/10/2019  1/11/2019 - 1/15/2019  1/16/2019 - 1/21/2019  1/22/2019 - 1/28/2019 |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER, Data Dictionary) | **1/29/2019 - 2/24/2019**  1/29/2019 - 2/4/2019  2/05/2019 - 2/9/2019  2/10/2019 - 2/19/2019  2/20/2019 - 2/27/2019 |
| **Implementation**  Building Database  Coding | **2/28/2019 - 3/28/2019**  2/28/2019 - 3/11/2019  3/12/2019 - 3/31/2019 |
| **Testing**  Unit Testing  Integration Testing  Blackbox Testing  Whitebox Testing | **4/01/2019 - 4/10/2019**  4/01/2019 - 4/03/2019  4/4/2019 - 4/05/2019  4/6/2019 - 4/7/2019  4/8/2019 - 4/10/2019 |
| **Deployment**  User Training  Final Report | 4/11/2019 - 4/20/2019  4/11/2019 - 4/13/2019  4/14/2019 - 4/20/2019 |

## 4.2) Milestones

**Description of Milestones:**

* **Project Management(14 days)**
  + - Risk Management (3 days)
    - Work Breakdown Structure (3 days)
    - Configuration Management (4 days)
    - Proposal Submission (4 days)
* **Analysis(25 days)**
  + - Requirement (7 days)
    - Use Case (5 days)
    - Architecture(Initial class diagram) (6 days)
    - Analysis Specification (7 days)
* **Design(30 days)**
  + - Structural Module (7 days)
    - Behavioral Module (5 days)
    - UI design (10 days)
    - Database design (8 days)
* **Implementation(32 days)**
  + - Build Database (12 days)
    - Coding (20 days)
* **Testing(10 days)**
  + - Unit Testing (3 days)
    - Integration Testing (2 days)
    - Black box Testing (2 days)
    - White box Testing (3 days)
* **Deployment(10 days)**
  + - User Training (3 days)
    - Final Report (7 days)

## 4.3) Scheduling/Gantt Chart

For the on-time delivery we need to breakdown and allocate the days. With proper scheduling the time can be managed in more precise way. The Gantt chart includes the start and end date for each of the task. Each task should be carried out with in the bound of same allocated time.

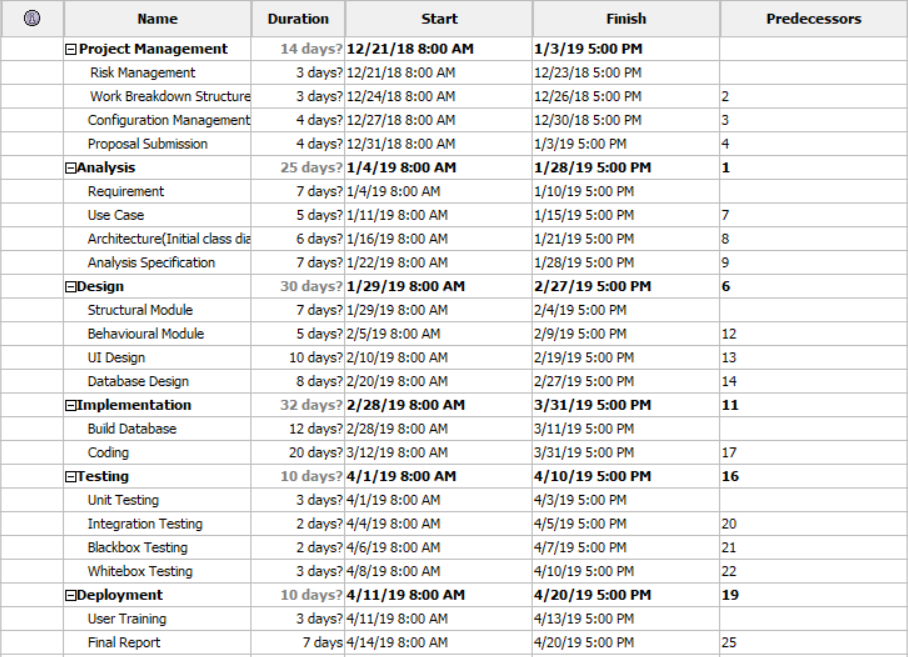


Figure 5 Allocating days for project.

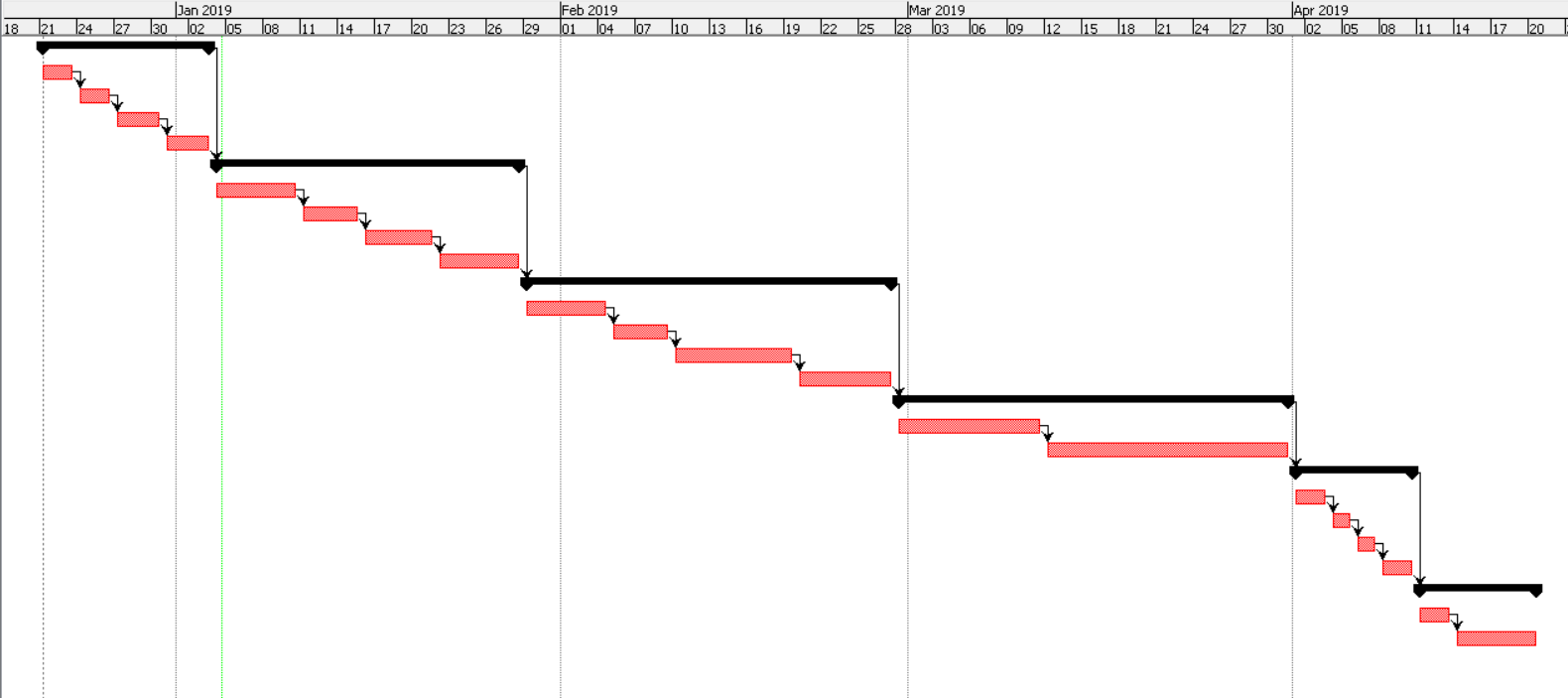
****

Figure 6 Gantt Chart

# **5) Risk Management**

Risk Management is a process of identifying the possible risk and prioritizing them for solution. Without Risk Management company can face serious loss on various aspect such and financial and human resources. In this project I also find the possible risk and provide suitable solution. The risk factor should be also monitored once in a while after the application is delivered so that it would not create problem in near future.

To calculate the impact we will use the simple formula i.e.:

Impact= consequences \* likelihood.

Here are values for both Likelihood and Consequences

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Risks | Likelihood | Consequences | Impact | Solution |
| 1 | Bad Scheduling | 2 | 4 | 6 | To avoid bad scheduling we should properly figure out the days that the project requires. And strictly following the Gantt chart. |
| 2 | Server failure | 2 | 5 | 10 | We should place the server in cold room that is free from dust or vulnerable to any disaster. |
| 3 | Failure to meet requirement | 2 | 5 | 10 | Proper identification of desired product with overall element should be noted before developing the system |
| 4 | Data leakage | 2 | 4 | 8 | The access control should be managed well so that feature of application can be served with their role. |
| 5 | Financial loss | 2 | 3 | 6 | The financial expenses that are to be spent should be analyzed clearly so that the development process can run smoothly. |
| 6 | Lack of human resource | 2 | 3 | 6 | All the employees should be made happy by offering them suitable working time with appropriate work load. |
| 7 | DDoS Attack | 2 | 4 | 8 | To avoid DDoS Attack we should implement high quality firewall. |

# **6) Configuration Management**

Configuration Management is tool for updating, configuring and validating the code during development. Configuration management avoids the misunderstanding and mistakes. It also helps for tracking the defects in the system and gives remote access to the multiple member of a team. With the proper configuration management we can also manage the version or control versions of system.

In this project I have used GitHub as my configuration management because it is Open source and easy to use.

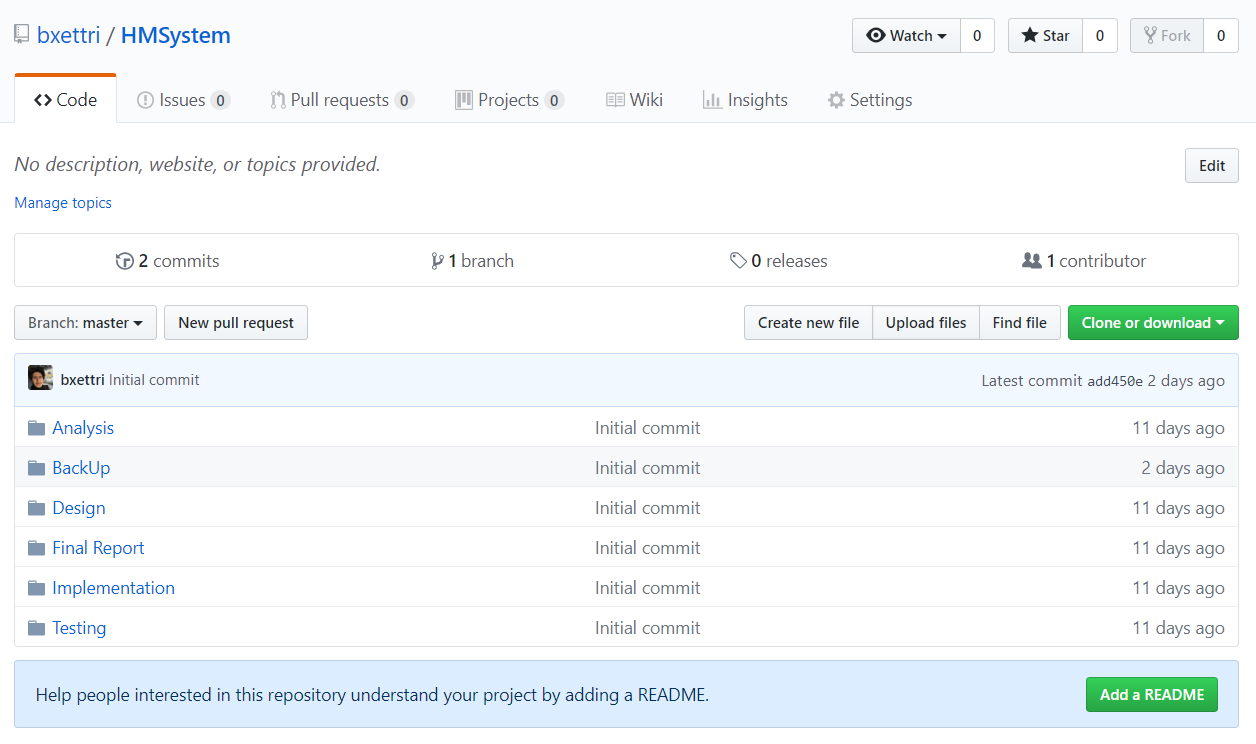


Figure 7 Repository in GitHub

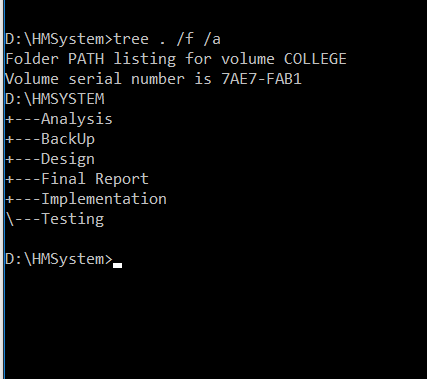


Figure 8 Listing of folder path.

# **7) Conclusion of the project**

Hence, this is my proposal for Hotel Management System. I have covered all the required elements for application. I will also be adding more feature to this project in near future. I hope my proposal would be accepted.

# **8) References**