**Project proposal**

**on**

**Bike Showroom Management System**

Asmita Adhikari  
 00172902

**Computing Project (CP)**   
 Level 5 Diploma in Computing

Softwarica College of IT and E-Commerce  
Kathmandu, Nepal

2019/01/06

Submitted to: Kiran Rana

# Introduction

## 1.1 Project Introduction

Project entitled “Bike showroom management system” is based on web-application developed for bike showroom maintaining daily activities like customer maintenances, bikes information, sales of bikes, booking of bike, bike companies details, new models of bikes could be seen by customers with their price ranges, Models of bikes from different companies. Admin could add new bikes on stock, delete sold out bikes also update the new stocks within their showroom.

## 1.2 Justification for project

### 1.2.1 Background of project

The system is developed for the users who could view bikes kept in sales and simultaneously book them. The system allows users to search the various models of bikes with their price ranges and functionality. Users could explore various kinds of bikes like sports bike, scooters, motorbikes and so on. When users book their desired bike, and request is sent to admin.

### 1.2.2 Problem Statement

The objective of the project is to create the management system for bike showroom is to show details about bikes in the showroom their functions with brands of bikes and in your budget. The system is web-based application which is new for the organization. Till now the showroom doesn’t have its web-based application they are following the old techniques of   
database storage in files or either in draft. Typically managing database would be difficult job, the new system might be somehow helpful for database storage, but some time system might crash, and work couldn’t be done.

## 1.3 Description of project

### 1.3.1 Features

Following are the features included in the project of bike showroom management system.

* Login system
* Budget for the bikes.
* Types of bikes.
* Search for the bikes
* Different mileage of bikes
* New launches of bikes.
* Brands of bikes from different companies.
* Bike features like most popular, new launches and upcoming models.
* Feedback form users.

# 2. Project Scope

## 2.1 Scope and limitation of project

A part of project that involves planning which involves defining and detailed list of specific goals, features, quality of a project within the untimely cost. In other word, project scope is what needs to be achieved in work done to deliver of project. (totally tech, n.d.)

**Scope of project**

The project of bike showroom management system is made for online customer who could book the desired bike. The project removes old methodology of people going to showroom and see the bike of their want.

**Limitation of project**

* Project doesn’t have online payment system customer could only book bikes.

## 2.2 Aims and objectives

Aims for the project bike showroom management system are as follows:

* Implement a web-based bike showroom management system.
* User convenient.
* Maintain data in web-page so that costumer would be up-to-date.

Objective for project bike showroom management system are as follows:

* To access web application for showroom.
* Admin awareness about system uses, how system works.
* Update all the showroom up-to-date stocks.
* Managing and creating proper documentation for further references.

# 3. Development Methodology

## 3.1 methodology used

Software development methodology used in the project is waterfall model. Waterfall model is linear sequential method of software development which involves continuous process in development. Each process must be completed going to another phase.

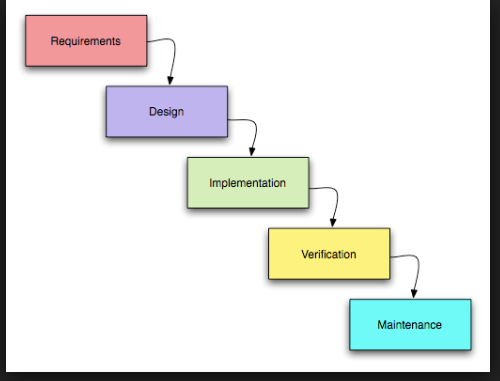


Figure 1: Waterfall Model

Waterfall model used in software development methodology has five steps as shown in above figure. Each step should be completed before going to another step.

Advantages of waterfall model:

1. Each phase has its own distinct goals.
2. Each phase would be complete in definite deadline.
3. Easy to manage.
4. Completed in one time and phases don’t overlap.

## 3.2 Design pattern

Design pattern used in the project is MVC (Model View Controller)

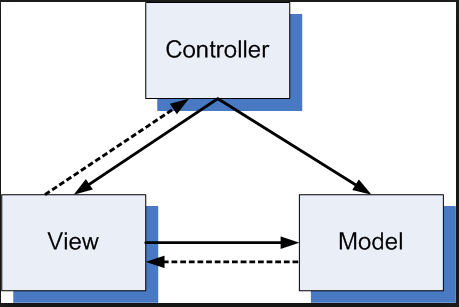


Figure 2: Design pattern using MVC

Architectural pattern used for software development MVC design pattern specifies that application contains data model, presentation model and controls information. MVC requires each separated into different objects. The design patterns mostly relate UI interaction layer of application.

Following explains MVC:

**Model**: It contains pure application data which doesn’t concern UI or presentation.

**View**: It present model data to user in a proper interface and allows user to manipulate data.

**Controller**: controller is between model and view. View is updated when model changes and model is updated when user manipulates the view.

## 3.3 System Architecture

3 tier-architecture is used to make the application. Software architecture containing 3 layers of logical computing. 3 tier architecture contains 3 layers.

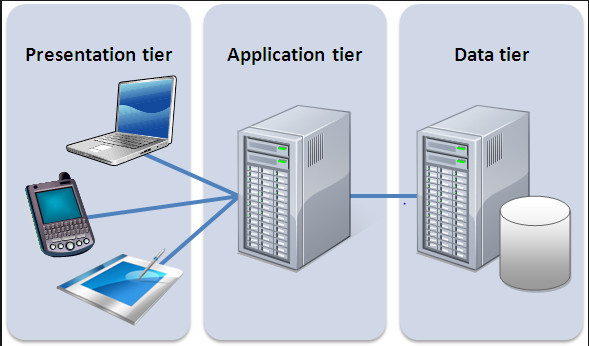


Figure 3: 3 tier-architecture

1. Presentation layer: Font-end layer in three tyre-architecture consist of user interface. This layer interface is graphical accessible through web browser application.
2. Application layer: Application tier contains logical functions of an application.
3. Data layer: data layer contains data storage system in database.

Advantages of MVC.

1. Faster development process.
2. Friendly development platform.
3. Modification doesn’t affect entire model.

# Work Break Down Structure (WBS)

## 4.1 Work breakdown structure

WBS structure is a hierarchical decomposition structure of a project which divides work into small parts. WBS provides foundation to project which helps to plan, manage and evaluation. It concentrates on desirable milestone rather than focusing on individual action.

Figure 4: Work Break down structure

## Milestone

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  WBS  Configuration Management  Proposal Submission | 12/21/2018 to 1/3/2019  12/21/2018 to 12/25/2018  12/26/2018 to 12/28/2018  12/29/2018 to 12/30/2018  12/31/2018 to 1/3/2019 |
| **Analysis**  Use case  Requirement analysis  Architecture (Initial Class Diagram)  Analysis specification | 1/4/2019 to 1/28/2019  1/4/2019 to 1/5/2019  1/6/2019 to 1/9/2019  1/10/2019 to 1/23/2019  1/24/2019 to 1/28/2019 |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER, Data Dictionary) | 1/29/2019 to 2/27/2019  1/29/2019 to 1/31/2019  2/1/2019 to 2/5/2019  2/6/2019 to 2/18/2019  2/19/2019 to 2/27/2019 |
| **Implementation**  Building Database  Coding | 2/28/2019 to 3/31/2019  2/28/2019 to 3/9/2019  3/10/2019 to 3/31/2019 |
| **Testing**  Unit Testing  Black box Testing  White box Testing  Integration Testing | 4/1/2019 to 4/10/2019  4/1/2019 to 4/2/2019  4/3/2019 to 4/5/2019  4/6/2019 to 4/8/2019  4/9/2019 to 4/10/2019 |
| **Deployment**  User Training  Final Report | 4/11/2019 to 4/20/2019  4/11/2019 to 4/15/2019  4/16/2019 to 4/20/2019 |

Table 1: Milestone

## 4.2 Description of milestone

**1. Project Management**

|  |  |
| --- | --- |
| Subtasks | Days/month allocation |
| 1. Risk Management | **5 days** |
| 1. WBS | **3 days** |
| 1. Configuration Management | **2 days** |
| 1. Proposal submission | **4 days** |
| **Total** | **14 days** |

Table 2: Project management

**2. Analysis**

|  |  |
| --- | --- |
| Subtasks | Days/Months allocation |
| 1. Use task | **2 days** |
| 1. Requirement analysis | **4 days** |
| 1. Architecture | **14 days** |
| 1. Analysis specification | **5 days** |
| **Total** | **25 days** |

Table 3: Analysis

**3. Design**

|  |  |
| --- | --- |
| Subtasks | Days/Months allocation |
| 1. Structural module | **3 days** |
| 1. Behavioural module | **5 days** |
| 1. UI design | **13 days** |
| 1. Database design | **9 days** |
| **Total** | **25 days** |

Table 4: Design

**4. Implementation**

|  |  |
| --- | --- |
| Subtask | Days/Months allocation |
| 1. Build database | **10 days** |
| 1. coding | **22 days** |
| **Total** | **32 days** |

Table 5: Implementation

1. **Testing**

|  |  |
| --- | --- |
| Subtask | Days/Months allocation |
| 1. Unit testing | **2 days** |
| 1. Black box testing | **3 days** |
| 1. White box testing | **3 days** |
| 1. Integration | **2 days** |
| **Total** | **10 days** |

Table 5: Testing

**6. Deployment**

|  |  |
| --- | --- |
| Subtask | Days/Months allocation |
| 1. User training | 5 days |
| 1. Final report | 5 days |
| **Total** | **10 days** |

Table 6: Deployment

## Scheduling /Gantt chart

I have broken down work break down structure and divided task into smaller parts assigning dates to each tasks and subtask I started my project in 21st December 2018. The project is divided into various parts to make task easier.

I have separated Gantt chart and schedule of task.

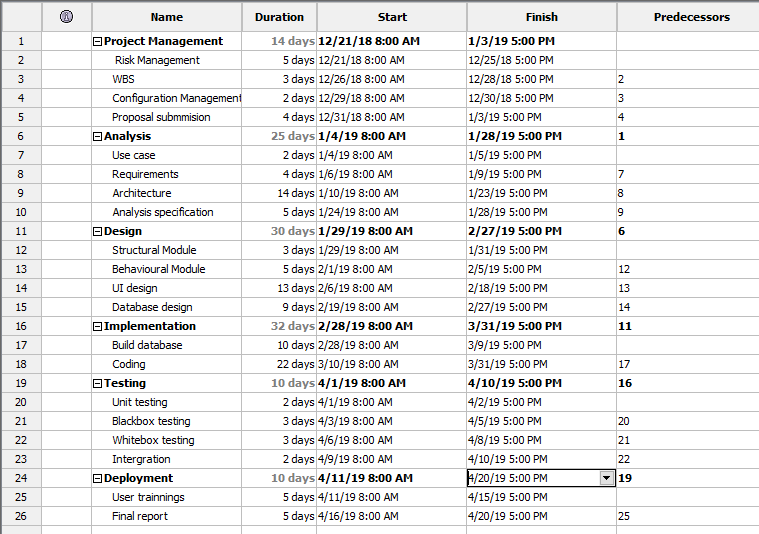


Figure 5: Scheduling

## Gantt chart

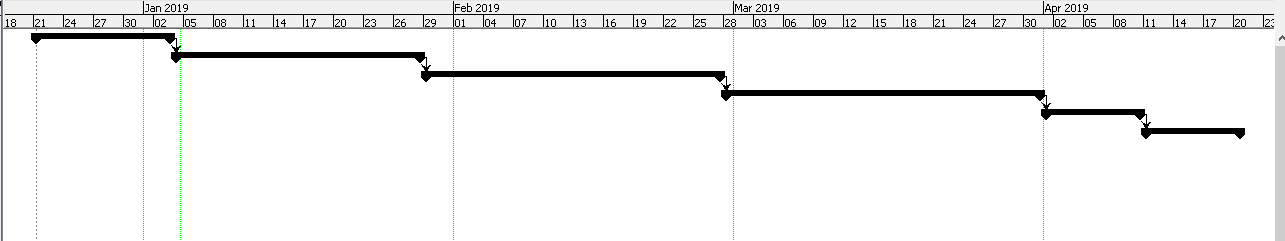


Figure 6: Gantt Chart

Graph diagram illustration of a project schedule. Above Gantt chart shows schedule of project starting from 21st December. Scheduling of projects are divided into various breakdowns. Project is divided into 6 parts where each part has assigned certain time to complete task.

# Risk Management

Process of identifying, analysing the risk of the certain task and identifying solution to overcome the problem is known as risk management. A risk is anything which impact directly on the project’s performance, usability or budget.

Risk could be managed by following methods.

* Identifying risk
* Analysing the risk
* Evaluating the risk
* Overcoming the risk
* Monitoring and reviewing risk

*Impact = Likelihood \* Consequence*

I am going to use following risk alleviation approach to manage risks and provide proper actions.

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

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

Table 7: Risk Likelihood and Consequence Values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Consequence** | **Impact** | **Action** |
| Server failure | 1 | 5 | 5 | Monitor server regularly. |
| Database Failure | 1 | 4 | 4 | Back-up of data in database. |
| Power fluctuation | 1 | 2 | 2 | Reduction of amplitude voltage. |
| Server overload | 2 | 5 | 10 | Maintenance on monthly basic, cleaning the servers and databases. |
| Employee theft | 2 | 5 | 10 | Marinating trustful environment with employee. Employee dissatisfaction leads to employee theft. |

Table 8: Risk Management Table

# Configuration Management

Configuration management is the process of ensuring consistency among physical and logical assets in an operational environment

# Conclusion of project

The project bike showroom management system is a user-friendly web application made focusing to the customer of showroom. This application removes old methodology where customer goes to showroom and book a bike. This application adopts new methodology but has limitation where customer couldn’t make online payment for the bike.