Project Proposal

On

**Expenses Management System**

Suveksha Khanal

00172884

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT and E-Commerce

Kathmandu, Nepal

04/01/2019



Submitted to: Kiran Rana

Contents

[1.1 Project Introduction 2](#_Toc534464779)

[1.2 Justification for project 2](#_Toc534464780)

[1.2.1 Background of the project 2](#_Toc534464781)

[1.2.2 Problem Statement 2](#_Toc534464782)

[1.3 Description of the project 2](#_Toc534464783)

[1.3.1 Features 2](#_Toc534464784)

[2. Project Scope 2](#_Toc534464785)

[2.1 Scope and Limitation of project 2](#_Toc534464786)

[2.2 Aims and Objectives 2](#_Toc534464787)

[3. Development Methodology 2](#_Toc534464788)

[3.1 Methodology used 2](#_Toc534464789)

[3.2 Design Pattern 2](#_Toc534464790)

[3.3 System Architecture 2](#_Toc534464791)

[4. Work Breakdown Structure (WBS) / Scheduling 2](#_Toc534464792)

[4.1 Work Breakdown Structure 2](#_Toc534464793)

[4.2 Milestones 2](#_Toc534464794)

[4.3 Scheduling / Gantt Chart 2](#_Toc534464795)

[5. Risk Management 2](#_Toc534464796)

[6. Configuration Management 2](#_Toc534464797)

[7. Conclusion of the project 2](#_Toc534464798)

[8. References 2](#_Toc534464799)

**List of figures**

[Figure 1: Waterfall model 7](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531972)

[Figure 2: MVC pattern 8](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531973)

[Figure 3: tier- 3 architecture 9](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531974)

[Figure 4: WBS 10](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531975)

[Figure 5: Schedule 12](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531976)

[Figure 6: Gantt chart 13](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531977)

[Figure 7: tree structure 15](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531978)

[Figure 8:github folders 15](file:///C:\Users\Rohit\Desktop\CP-Proposal-suvee.docx#_Toc534531979)

**1. Introduction**

## **1.1 Project Introduction**

The name of my project is Expenses management system. It is used by a business or an individual to audit their expenses. This system allows user to input their income and expenses. It tracks their spending which allows them to have an idea of their expenses. Expenses management system analyses overall expenses, identifies cost-saving opportunities and controls excessive spending.

## **1.2 Justification for project**

### **1.2.1 Background of the project**

Expenses Management system allows user to register and login here they can add their income as well as their daily expenses. This allows them to control their expenses and think of their savings. Their income and expenses will be saved into their account so they can come back and edit their details. This system allows user to track their money and have control over their spending. Time should be managed properly for this project in order to finish it on time. This expense management system provides an integrated set of features to help you to manage your expenses and cash flow. It provides the ability to group your income/expenses into categories and lets you set a budget and track expenses in the category.

The tools I will be using for this project are Laravel which will be used to create design and code for the project. I will also be using Project Libre to manage and list my schedule. Project libre will help me do my work according to the schedule.

### **1.2.2 Problem Statement**

This project is focused on any individual or organization who wants to keep track of their income and expenses. However, this project is mainly focused on users who work as their income is fixed and trackable. This system will help them track their expenses as well as control their spending. This system can be used anywhere and at any time such as work place or at home. It will be user- friendly as it will be easy to use and understand.

The design of the system should be eye-catching and targeted at any age group.

## **1.3 Description of the project**

### **1.3.1 Features**

The features of the project are as follows:

* User can register on the page
* User can register using their email and password
* Any new information will be sent to the email
* Login
* Username will be set
* User can login using their username and password
* Edit profile
* User can edit their profile such as username/ password
* User can add/ edit expenses and income
* Change password
* Dashboard for user
* Dashboard will help user to navigate through the page easily
* They can go from expenses page to income page
* Log out
* Add income
* User will need to add income to calculate their salary and expenses
* Edit income
* Add expenses
* User need to add their expenses by remembering their spending
* Their expenses will be saved.
* Edit and view expenses
* Add expense category such as “food” “fuel”
* View details of expense category
* Calculate their spending according to their income
* Their spending will be calculated by subtracting it from their income
* This will help user to track their income
* Report of the expense
* Report will be generated by calculating their expenses
* Report of the expense category

# **2. Project Scope**

## **2.1 Scope and Limitation of project**

Our expense management system designed to help individual or business budget, track and possibly control your expenses. It supports tracking of both your expenses and income. This expense management system provides an integrated set of features to help you to manage your expenses and cash flow. It provides the ability to group your income/expenses into categories and lets you set a budget and track expenses in the category.

Individual might not know or remember where they spent their income hence the tracking of the expense will be unreliable. Income of individual might not be regular hence their calculation might be wrong. Their spending might be more than that of their income.

## **2.2 Aims and Objectives**

**Aims:**

* To build a system where user can add, update and track their expenses
* To help user control their spending.
* To generate expenses report at the end.
* To calculate spending

**Objectives:**

* To perform better design
* To perform better code
* To manage time
* To work according to the requirements
* To perform different testing to make sure there are no flaws.
* To perform step by step process to finish the project
* To use Development methodology
* To use Design pattern
* To use structural architecture

# **3. Development Methodology**

## **3.1 Methodology used**

**Waterfall Model**

It is a development methodology where software is developed step by step. It is very easy to use and understand. In waterfall model each phase must be completed before moving to the next one, there is no overlapping of the phases. Waterfall model is also suitable for small projects. Waterfall model is divided into separate phases which are as follows:

* Requirement
* System Design
* Implementation
* Testing
* Deployment
* Maintenance

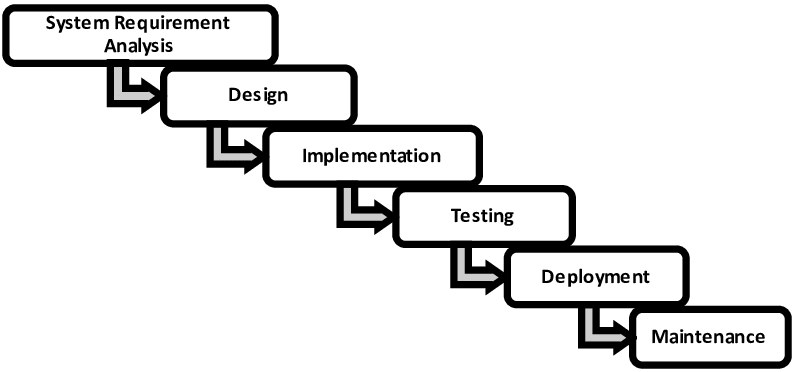


Figure 1: Waterfall model

I will choose this project because:

* It is easy to use
* It is very easy to understand
* It is used for small projects

Advantage of waterfall model:

* Easy to use and understand
* Effective cost
* Each phase is completed one at a time
* It is used for smaller projects

Disadvantage of Waterfall model:

* There is high amount of risk and uncertainty
* Changes cannot be changed
* Phases cannot be jumped or go backwards.

## **3.2 Design Pattern**

**MVC**

I will be using MVC design pattern.

**Model: I**t represents an object. It also has logic to update controller if the data changes.

**View:** It represents the visualization of the data that model contains.

**Controller:** It acts on both model and view. It controls data flow into model object.

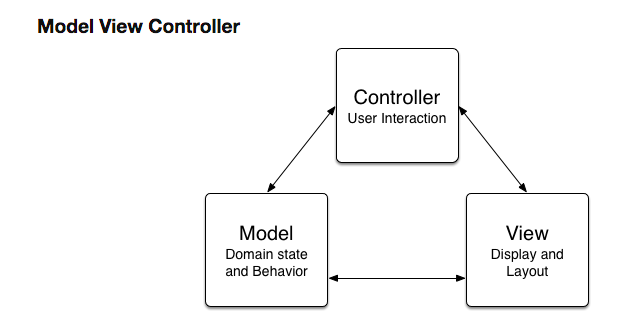


Figure 2: MVC pattern

Advantage of MVC pattern:

* Helps develop application faster
* Multiple developer can work at one project
* Modification does not affect the entire model because model part does not depend on the views part.
* MVC pattern returns data without applying any formatting so the same components can be used and called for use with any interface.

Disadvantage of MVC pattern:

* Complex
* Need multiple programmers
* Inefficiency of data access in view

## **3.3 System Architecture**

I am going to use tier-3. A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture. Three-tier architecture allows any one of the three tiers to be upgraded or replaced independently. The user interface is implemented on a desktop PC and uses a standard graphical user interface with different modules running on the application server.



Figure 3: tier- 3 architecture

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

## **4.1 Work Breakdown Structure**

Work break down structure is a breakdown of a project into smaller parts to remove complexity. It organizes the team’s work into manageable section. WBS is a hierarchical and incremental decomposition of the project into phases, deliverables and work packages. It is a [tree structure](https://en.wikipedia.org/wiki/Tree_structure), which shows a subdivision of effort required to achieve an objective; for example a [progra](https://en.wikipedia.org/wiki/Program_management)m, [project](https://en.wikipedia.org/wiki/Project), and [contract](https://en.wikipedia.org/wiki/Contract).

For this project the work is divided into:

* Project Management
* Analysis
* Design
* Implementation
* Testing
* Deployment

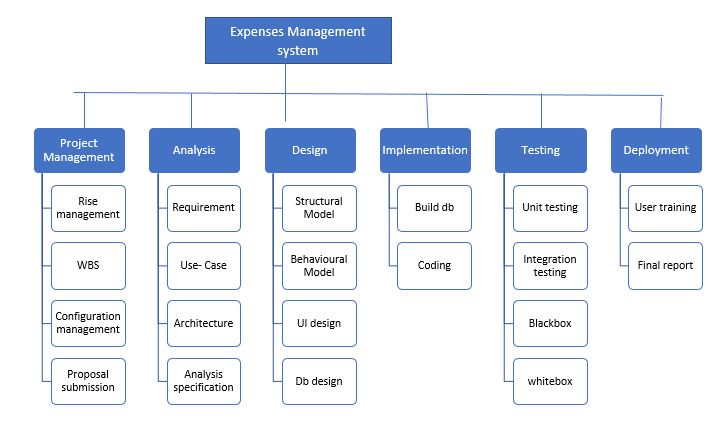
There are subsections where the project is divided into further parts.

Figure 4: WBS

## **4.2 Milestones**

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  WBS  Configuration Management  Proposal Submission | 21/12/2019- 3/01/2019  24/12/2019  26/12/2019  31/12/2019  03/01/2019 |
| **Analysis**  Feasibility Study  Requirement analysis  Planning  Use Case  Architecture (Initial Class Diagram) | 4/01/2019- 28/01/2019  13/01/2019  17/01/2019  24/01/2019 |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER, Data Dictionary) | 29/01/2019- 27/02/2019  04/02/2019  10/02/2019  21/02/2019 |
| **Implementation**  Building Database  Coding | 28/02/2019- 31/03/2019  05/03/2019  31/03/2019 |
| **Testing**  Unit Testing  Integration Testing  Blackbox Testing  Whitebox Testing | 1/04/20191- 10/04/2019  03/04/2019  10/04/2019  07/04/2019  05/04/2019 |
| **Deployment**  User Training  Final Report | 11/04/2019- 20/03/2019  17/04/2019  20/04/2019 |

**Description of Milestones:**

These are the estimated days that will take to complete each phase. The project needs to be completed in limited amount of time hence appropriate days are given to each steps in order to finish the project effectively.

* **Project Management (14 days)**
* **Analysis (25 days)**
* **Design (30 days)**
* **Implementation (32 days)**
* **Testing (10 days)**
* **Deployment (10 days)**

## **4.3 Scheduling / Gantt Chart**

Schedule is a plan for carrying out certain activity in intended amount of time. It is an arrangement to take place at certain time. Here is the schedule and Gantt Chart for this project.

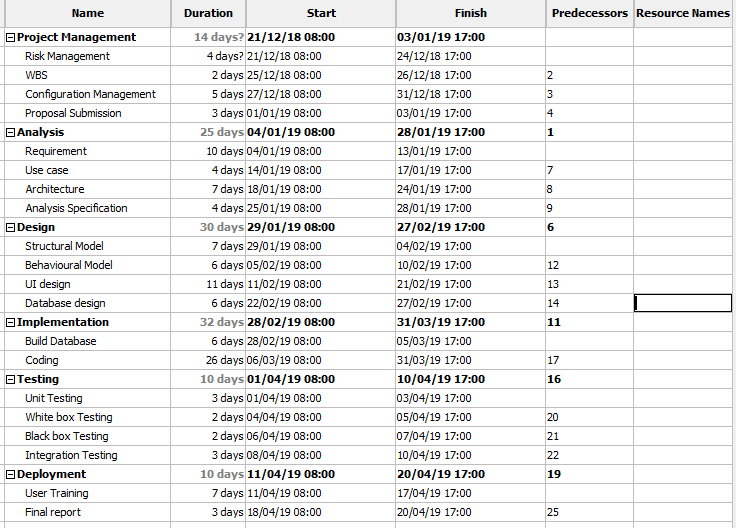


Figure 5: Schedule

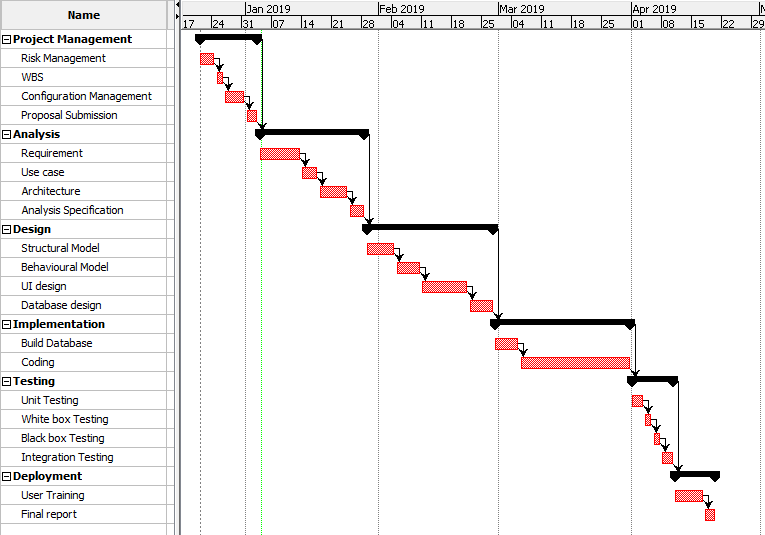


Figure 6: Gantt chart

# **5. Risk Management**

Risk management is the process of identifying, accessing and controlling threats or risks of a system. It is easier to identify risk before they occur so it can be solved faster when they actually occur. Risk management also accounts human factors including potential errors. Risk management helps minimize risks and time before they happen. Extra money and time is saved through risk management.

Impact= likelihood\* consequence

Risk Likelihood values are shown as follows

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

Risk Consequence values are shown below

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Risks | Likelihood | Consequences | Impact | Solution |
| 1 | Lack of resources | 2 | 3 | 6 | Resources needed for the project must be available. |
| 2 | OS failure | 2 | 4 | 8 | Back up of file should be kept. |
| 3 | Electricity cut | 1 | 5 | 5 | UPS must be set up and backup should be done. |
| 4 | Mistakes of human | 3 | 3 | 9 | Testing should be done identify mistakes and solution. |
| 5 | Virus | 2 | 4 | 8 | Antivirus should be installed. |
| 6 | Human illness | 3 | 2 | 6 | Time should be managed properly. Extra hour should be done at the end. |
| 7 | Hard disk failure | 1 | 4 | 4 | Cloud back up should be done. |

# **6. Configuration Management**

Configuration management is a [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering) process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.

Following is my folders in github and tree structure:

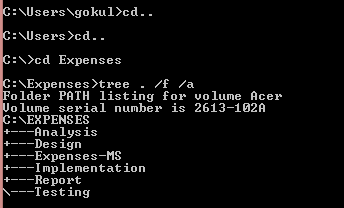


Figure 7: tree structure

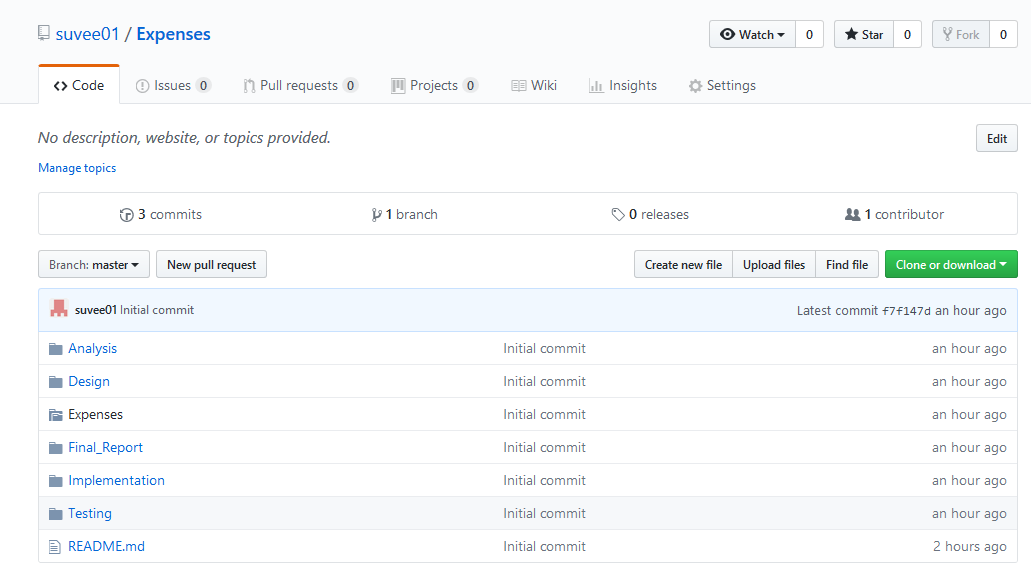


Figure 8:github folders

# **7. Conclusion of the project**

Therefore, this project focuses on expenses calculation to motivate the user to reduce their spending. This project is planned to finish at certain time hence all the effort is given to this project. Expenses Management System helps reduce financial burden off the shoulder. They do not need to worry about spending a lot. User can easily keep their spending record hence manage their expenses.

# **8. References**

[**https://www.webprojectbuilder.com/item/expense-management**](https://www.webprojectbuilder.com/item/expense-management)

[**https://www.freeprojectz.com/paid-projects/php-mysql/expense-management-system**](https://www.freeprojectz.com/paid-projects/php-mysql/expense-management-system)

[**https://www.techopedia.com/definition/24649/three-tier-architecture**](https://www.techopedia.com/definition/24649/three-tier-architecture)

[**https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm**](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)

[**https://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm**](https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm)