PROPOSAL PROJECT ON CAFÉ MANAGEMENT SYSTEM

PRABESH NIROULA

00175909

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT and E-commerce

Kathmandu, Nepal

1st July, 2019

Submitted to: SUDEEP LAL BAJIMAYA

**Chapter 1: Introduction**

* 1. **Introduction to the system**

The topic of this proposal is “Café Management System”. Café Management System is a general software developed particularly for handling the transactions of various cafés. This software is useful for all types of users in the café so that their transactions and details are up to date. The aim of this software is to provide a complete solution for café sales system. The front- end part is done with Microsoft Visual Studio and the database is maintained using Microsoft SQL Server Management.

* 1. **Background of the system**

There are a few management systems built for cafes. But they lack the functionality to handle the daily complications that occur during daily sales. They are often overcomplicated, needs trained professional to use and cost a fortune to deploy. Our system works on these sectors and makes it efficient in all possible ways. So, we can’t say it is totally a new concept but an improvisation of the old worn out system.

* + 1. **Problem Statement**

As for the problem statement part, some of the things should be addressed and improved on. The existing systems have complex user interface. Customer dealings should be done swiftly and correctly in case of large number of customers. So, people often find in hard to work quickly with such tedious interfaces. Another problem is these kinds of system cost a lot of money. Café owners usually buy a cheap copy of unauthorized product with horrible functionality. I would also like to minimize the unnecessary functions and make it usable for people of all age groups.

* 1. **Justification of the system**

Justification of a particular system is done with the help of analysis. It generally deals with why an organization needs to implement the proposed system and how can it be implemented. Our system fits the role here because it has an easy to use interface. Café owners are not necessarily familiar with the complex computer functionalities. So, we have made our system simple and straightforward just like and ATM machine. It provides every function that a café needs to operate on a daily basis kind of like a complete package. It saves time that is wasted on placing orders manually and bill calculation. This system can be implemented quite easily. The program does not take that much space and the requirements needed for this system are minimum.

* 1. **Overview of the proposed system**

So, in a nutshell, the proposed system helps in managing the café. It does this task by effectively and effortlessly generating bills after the order. It does not require large number of manpower or technical skills to use this software. This way the place remains organized and the financial dealings are more transparent.

**Chapter 2: Scope**

**2.1 Aims of the project**

The aim of this proposal is to provide an interactive interface to the cafés for handling the monetary transactions better. It also aims to reduce time consumption and know the customer payment details fast.

**2.2 Objectives of the project**

**a)** A computer-based system is introduced to handle all the primary information to calculate total expenditure of the customer.

**b)** This project intends to introduce more user-friendliness in the various activities involved in using the interface.

**c)** Maintain a proper database of the customer so that it can be retrieved as per necessary.

**d)** Setting up a functional and safe MIS for the customers.

**2.3 Features to be included**

**a)** Aesthetic and interactive user interface which is easy to use.

**b)** Bill calculation is a click away with added service tax and number of items.

**c)** Printing of the bills, saving them and retrieving them when necessary.

**d)** Shows the current date and time.  
 **e)** Copy, cut and paste options of the bill items to increase efficiency.

**2.4 Overview of the scope**

calculations for a middle-ranged café. The menu interface is small and the list of items that can be added into the system is limited. Adding too much items can make the interface less aesthetic and tedious to use.

**Chapter 3: Development Methodology**

**3.1 Methodology to be used**

For the methodology I have decided to use the waterfall model. The reason behind using it is the project itself is not huge. Had it been bigger I would have considered any form of Agile development.

Waterfall model is basically known for having a rigid work process. The processes are sequentially done. When one part is completed then we move onto the next one. It literally follows the principle of a real-life waterfall which moves from up to down plane and does not return back. There is no iteration involved as it increases the cost involved in the project. Therefore, it is the best suitable methodology for our project.

1. Figure showing the waterfall model



1. **Feasibility Study**: The first phase in waterfall model is requirements gathering, here the end user requirements are captured and feasibility study is done. After this software requirements document (SRS) is prepared.
2. **Requirement Analysis and Specification:** In this phase the requirements are gathered by the business analyst and they are analyzed by the team. Requirements are documented during this phase and clarifications can be sought.

The Business Analysts document the requirement based on their discussion with the customer.

Going through the requirements and analyzing them has revealed that the project team needs answers to the following questions which were not covered in the requirements document –

* Will the new application be used in more than one country?
* Do we have to support multiple languages?
* How many users are expected to use the application? Etc.

1. **Design:** The architect and senior members of the team work on the software architecture, high level and low-level design for the project.

It is decided that the application needs to have redundant backup and failover capabilities such that system is accessible at all times.

The architect creates the Architecture diagrams and high level / low level design documents.

1. **Coding and Unit Testing:** The development team works on coding the project.

They take the design documents / artifacts and ensure that their solution follows the design finalized by the architect.

Since the application is a café related application and security was a medium priority in the application requirements, they implement intermediate level security checks, audit logging features in the application.

They also perform several other activities like a senior developer reviewing the other developers code for any issues. Some developers perform static analysis of the code.

1. **Integration and System Testing:**  After developers are done with coding and provide final build to testers, testing starts in this phase. The testing team tests the complete application and identifies any defects in the application.

These defects are fixed by the developers and the testing team tests the fixes to ensure that the defect is fixed.

They also perform regressiontesting of the application to see if any new defects were introduced.

1. **Maintenance:** During the maintenance phase, the team ensures that the application is running smoothly on the servers without any downtime.

Issues that are reported after going live are fixed by the team and tested by the testing team.

## **Advantages of waterfall model**

* This model is simple and easy to understand and use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model phases are processed and completed one at a time. Phases do not overlap.
* Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

## **Disadvantages of waterfall model**

* Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

**3.2 Design Pattern**

I will be using MVC design pattern for the project. MVC stands for Module, View and Controller. The pattern requires that each of these be separated into different objects. MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI or the interaction layer of the application.

2. uml diagram of mvc pattern



* The **Model** contains only the pure application data, it contains no logic describing how to present the data to a user.
* The **View** presents the model’s data to the user. The view knows how to access the model’s data, but it does not know what this data means or what the user can do to manipulate it.
* The **Controller** exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.

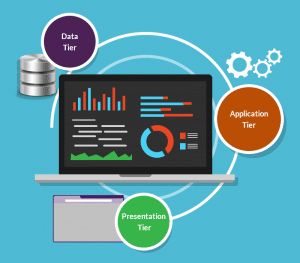
I intend to use MVC design pattern because of the following features it possesses:

1. Faster development process
2. Ability to provide multiple views
3. Support for asynchronous technique
4. Modification does not affect the entire model
5. MVC model returns the data without formatting

**3.3 System Architecture**

We will be using 3-tier architecture in our project. A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system.

3. 3-tier system architecture



* **Presentation Tier-**The presentation tier is the front-end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls.
* **Application Tier-**The application tier contains the functional business logic which drives an application’s core capabilities. It’s often written in Java, .NET, C#, Python, C++, etc.
* **Data Tier-**The data tier comprises of the database/data storage system and data access layer. Examples of such systems are MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Data is accessed by the application layer via API calls.

There are many benefits to using a 3-layer architecture including speed of development, scalability, performance, and availability.  As mentioned, modularizing different tiers of an application gives development teams the ability to develop and enhance a product with greater speed than developing a singular code base because a specific layer can be upgraded with minimal impact on the other layers.  It can also help improve development efficiency by allowing teams to focus on their core competencies. Many development teams have separate developers who specialize in front- end, server back-end, and data back-end development, by modularizing these parts of an application you no longer have to rely on full stack developers and can better utilize the specialties of each team.

Scalability is another great advantage of a 3-layer architecture. By separating out the different layers you can scale each independently depending on the need at any given time. By having disparate layers, you can also increase reliability and availability by hosting different parts of your application on different servers and utilizing cached results. With a full stack system, you have to worry about a server going down and greatly affecting performance throughout your entire system, but with a 3-layer application, the increased independence created when physically separating different parts of an application minimizes performance issues when a server goes down.

**Chapter 4: Scheduling**

**4.1 WBDS (Work Breakdown Structure)**

Café Management System

Project

Management

Testing

Implementation

Design

\

Analysis

Maintenance

Unit testing

Structural Diagram

Requirements Specification

Improvement

Database Building

Planning

Scope

Management

Maintainability

White box testing

Diagram

Task management

Coding

Behavioral Diagram

Black box testing

WBS

Architecture

Integration testing

UI design

Team management

Analysis specification

Database design

Risk management

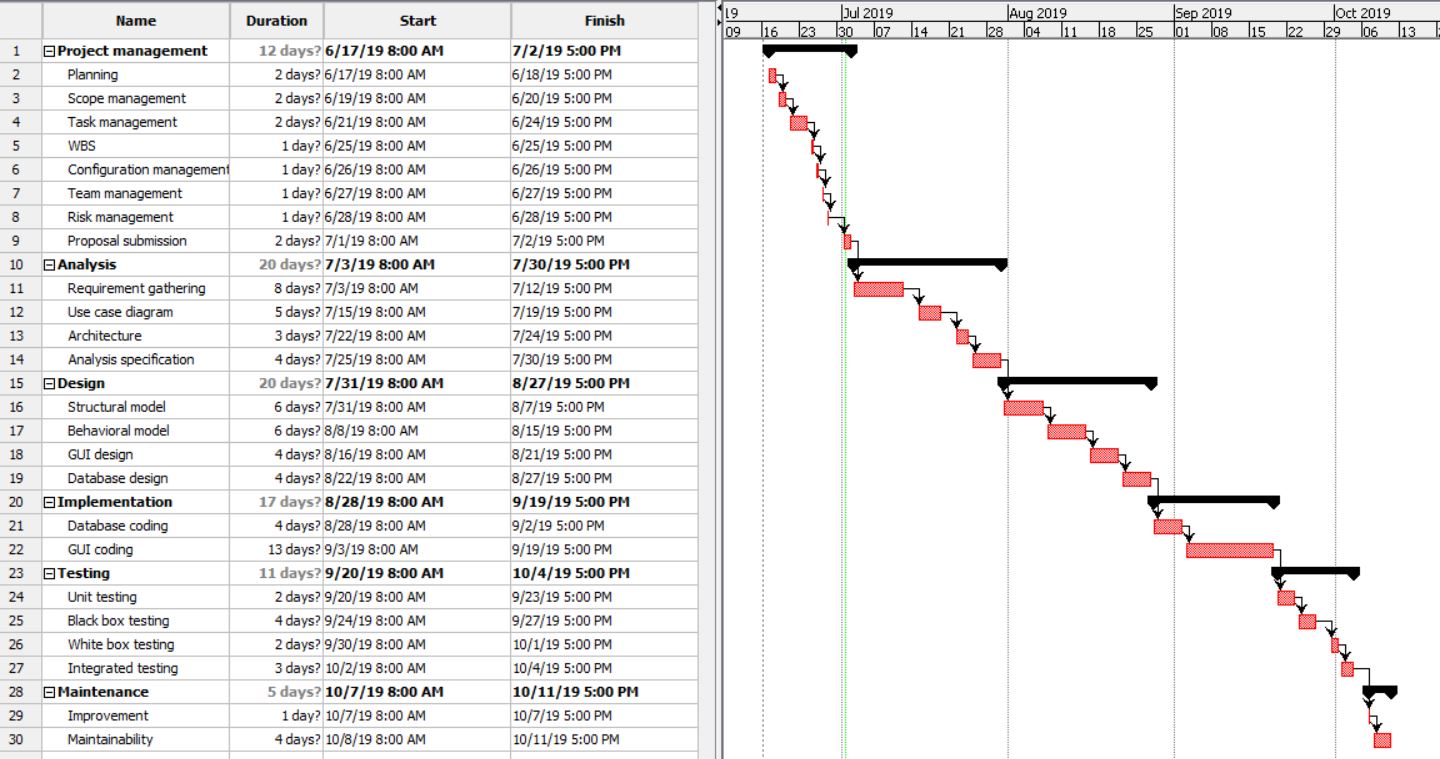
Submission

**4.2 Milestone**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Milestone** | **Date** |
| **1.** | **Project Management**  Planning  Scope management  Task management  WBS  Configuration management  Team management  Risk management  Proposal submission | **6/17/19 8:00 AM to 7/2/19 5:00 PM**  6/17/19 8:00 AM to 6/18/19 5:00 PM  6/19/19 8:00 AM to 6/20/19 5:00 PM  6/21/19 8:00 AM to 6/24/19 5:00 PM  6/25/19 8:00 AM to 6/25/19 5:00 PM  6/26/19 8:00 AM to 6/26/19 5:00 PM  6/27/19 8:00 AM to 6/27/19 5:00 PM  6/28/19 8:00 AM to 6/28/19 5:00 PM  7/1/19 8:00 AM to 7/2/19 5:00 PM |
| **3.** | **Analysis**  Requirement gathering  Use case diagram  Architecture  Analysis specification | **7/3/19 8:00 AM to 7/30/19 5:00 PM**  7/3/19 8:00 AM to 7/12/19 5:00 PM  7/15/19 8:00 AM to 7/19/19 5:00 PM  7/22/19 8:00 AM to 7/24/19 5:00 PM  7/25/19 8:00 AM to 7/30/19 5:00 PM |
| **4.** | **Design**  Structural diagram  Behavioral diagram  UI Design  Database Design | **7/31/19 8:00 AM to 8/27/19 5:00 PM**  7/31/19 8:00 AM to 8/7/19 5:00 PM  8/8/19 8:00 AM to 8/15/19 5:00 PM  8/16/19 8:00 AM to 8/21/19 5:00 PM  8/22/19 8:00 AM to 8/27/19 5:00 PM |
| **5.** | **Implementation**  Database coding  GUI coding | **8/28/19 8:00 AM to 9/19/19 5:00 PM**  8/28/19 8:00 AM to 9/2/19 5:00 PM  9/3/19 8:00 AM to 9/19/19 5:00 PM |
| **6.** | **Testing**  Unit testing  Black box testing  White box testing  Integrated testing | **9/20/19 8:00 AM to 10/4/19 5:00 PM**  9/20/19 8:00 AM to 9/23/19 5:00 PM  9/24/19 8:00 AM to 9/27/19 5:00 PM  9/30/19 8:00 AM to 10/1/19 5:00 PM  10/2/19 8:00 AM to 10/4/19 5:00 PM |
| **7.** | **Maintenance**  Improvement  Maintainability | **10/7/19 8:00 AM to 10/11/19 5:00 PM**  10/7/19 8:00 AM to 10/7/19 5:00 PM  10/8/19 8:00 AM to 10/11/19 5:00 PM |

**4.3 Gan Chart**

4. figure showing Gan-chart



**Chapter 5: Risk Management**

Project risk management is the process of identifying, analyzing and then responding to any risk that arises over the life cycle of a project to help the project remain on track and meet its goal. Risk management isn’t reactive only; it should be part of the planning process to figure out risk that might happen in the project and how to control that risk if it in fact occurs.

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

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

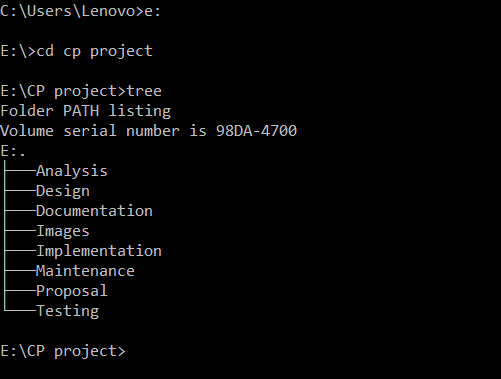
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Risk | Likelihood | Consequence | Impact | Action |
| 1 | Hardware failure | 2 | 5 | 10 | Frequent analysis and repair of hardware |
| 2 | Crashing of software | 3 | 4 | 12 | Proper backup should be maintained |
| 3 | Time and budget shortage | 3 | 3 | 9 | Proper allocation of time and budget |
| 4 | Resources not met | 1 | 3 | 3 | Requirements should be taken carefully |
| 5 | Natural calamities | 1 | 5 | 5 | Backup in cloud or google drives |

**Chapter 6: Configuration Management**

Project configuration management (PCM) is the collective body of processes, activities, tools and methods project practitioners can use to manage items during the project life cycle. PCM addresses the composition of a project, the documentation defining it, and other data supporting it.

The Importance of Configuration Management. Configuration management (CM) focuses on establishing and maintaining consistency of a product's performance, and its functional and physical attributes with its requirements, design, and operational information throughout its life.

I have maintained a local backup in my pc containing a main folder and sub-folders.

****

For online backup, I have created a GitHub account with username “prabeshmetalz” where the folders are managed in the same way.

**Chapter 7: Conclusion**

Therefore, the proposal includes everything from analysis, design, implementation and other plans required to make this project a success. The aims and features have been clearly listed. Scheduling has been done properly. Milestone tracking and risk management issues have been examined. This project is going to be built using Waterfall model and MVC design pattern.

**References:**

* 1. www.tutorialspoint.com. (2019). *MVC Framework Introduction*. [online] Available at: https://www.tutorialspoint.com/mvc\_framework/mvc\_framework\_introduction.htm [Accessed 1 Jul. 2019].
  2. JReport. (2019). *3-Tier Architecture: A Complete Overview - JReport*. [online] Available at: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/ [Accessed 1 Jul. 2019].
  3. Managementstudyonline.blogspot.com. (2019). *What is the Waterfall Model in SDLC?* [online] Available at: http://managementstudyonline.blogspot.com/2014/04/what-is-waterfall-model-in-sdlc.html [Accessed 1 Jul. 2019].