

Project Proposal On Goods Exchange Hinge



Computing Project
L5DC Diploma in Computing
Softwarica College of IT and E-Commerce
Mahakabi Marg, Dillibazer, Kathmandu, Nepal
April 9, 2019

Submitted by: -

Mohammad Hussain Sheikh

College ID: 160440

NCC ID: 00174409

Submitted to: -

Sudeep Bajimaya

Keywords

- Dynamic System Development Method (DSDM)
- MVC Design Pattern
- Three tier architecture
- Integrated Development Environment (IDE)
- Work Breakdown Structure (WBS)
- Risk Management Matrix
- Reverse Engineering

Table of Figure

Figure 1 DSDM model.....	8
Figure 2 Agile Development Process	9
Figure 3 Comparison between old and agile method process	10
Figure 4 MVC design pattern	11
Figure 5 Architecture.....	12
Figure 6 Development Tools	12
Figure 7 Project work breakdown structure	13
Figure 8 Project WBS with time estimation.....	14
Figure 9 Project Milestone.....	16
Figure 10 Project Schedule	16
Figure 11 Gantt Chart	17
Figure 12 Risk Matrix	18
Figure 13 Risk Management	19
Figure 14 Project Directory Structure	20
Figure 15 Forward and Reverse Engineering.....	20

Contents

Table of Figure.....	2
1 Introduction	4
1.1 Project Introduction	4
1.2 Justification of project	4
1.2.1 Background of Project	4
1.2.2 Problem Statement.....	5
Solution	5
1.3 Description of the project	6
1.3.1 Features: -	6

1.4	Overview: -.....	6
2	The scope of the project.....	7
2.1	Scope	7
2.2	Limitation	7
2.3	Aims.....	7
2.4	Objectives	7
2.5	Overview of the Scope	8
3.	Development methodology	8
3.1	Description of the methodology	8
3.2	Design Pattern.....	11
3.3	Architecture	11
3.4	Development Tools.....	12
4.	Work Breakdown Structure(WBS) / Project Planning	13
4.1	About WBS.....	13
	Project WBS.....	13
4.2	Milestone	14
4.3	Project Schedule	16
4.4	Gantt Chart.....	17
5	Risk Management	17
5.1	Lifecycle	17
5.2	Steps	18
6	Configuration Management	19
6.1	Reverse Engineering	20
6.2	Version Control System (VCS)	21
7	Conclusion	21
8	Reference	22

1 Introduction

1.1 Project Introduction

A goods exchange commonly known as barter is an old method of exchanging goods and services. It has been used for centuries and long before money was invented where peoples exchanged goods and services for other goods and services in return. In ancient times people exchanged things in the same area. Today's a goods exchange has made global using technologies that are more sophisticated to aid in trading.

Today's technology has made goods exchange method more manageable and easy. Thus I propose to develop a goods exchange system for people who are interested in exchanging their stuff but no longer want or need. So I've choose to call this system a Goods Exchanges Hinge.

1.2 Justification of project

Items that are used as money often have a little value in and of themselves. For example, the paper used to print money is not particularly valuable. Money has value because it is an exchange medium that people understand and accept as such. Similar to this a goods exchange is the process of trading services or goods between two parties without using the money in the transaction. When peoples barter, both parities benefit because they receive items or service they need or want and a medium to meet up with new people. I want to build this system, not because of my academic project but I really want to provide a platform where people can easily find the items they want and exchange with little effort in a most convenient way.

1.2.1 Background of Project

Back into the time around 6000 BC introduced by Mesopotamia tribes, bartering was adopted by Phoenicians. Goods were exchanged for food, tea, weapons, and spices. At those times, human skulls were used as well. People use to exchange goods inconveniently. For example, a human skull was exchanged with the sheep which is inconvenient and miss-match. Civilization takes place, things are understood and managed in a proper way.

Keeping the above thing in mind I want to implement this idea in a more appropriate way where peoples can find the list of available goods in several categories. There is a difference in the application rather than swapping goods randomly I want to make items value comparable with one another this can be done in two way either the owner of the item can clearly state which type of products or goods they are open for a swap or pay a little amount of money instead.

The application is going to provide a description of goods, a list of recently added products in the home page, searching facility, a login system with security for both user and admin where both user and admin have a different perspective of viewing the application.

1.2.2 Problem Statement

It can be difficult to build a system that works perfectly no matter how much research and effort is applied there exist some constraints and issue which should be addressed with a relevant solution for future improvement. Some possible issue and their solution are listed below:

- User has to meet in order to swap things which could be problematic because it is difficult to determine how trustworthy the person is.
- There is no consumer warranties involved this means the goods and services both parties are exchanging may be exchanged for poor or defective items.
- A most common issue is security, where both user and admin have to provide a different level of access to the application.

Solution

A common solution to all the issue identified above: -

- Proper implementation of the security level in an application should be improved.
- Terms of use should be clearly stated such that the visitor can clearly identify how to use the system.
- Privacy policy and legal problem should be clearly stated to avoid illegal consequences.

1.3 Description of the project

The goods exchange system have a broad level of functionality which requires a high level of research, experience in the related field and expert team. Some of its functionality includes: -

- Generates sales/profits with faster inventory and service-hour turnover
- Creates new customers by bringing parties together who may refer others
- Eliminates additional advertising and deep discounts

Since this is an academic project it only includes a limited functionality of such a vast domain. Some of the features of my project are: -

1.3.1 Features: -

- A login system, user and goods information manipulation, keep goods record.
- Meet awesome people and exchange goods to the nearest locality.
- Specify the market value of goods by the user if the user wants to exchange goods with some amount of money while exchanging goods.
- Exchange goods with absolutely free of cost if there is no market value specified by the user.
- Search items, report user and get help from the community chatbot.

1.4 Overview: -

A goods exchange hinge provides a platform where people can meet with people in their locality and swap or exchange the goods of their demands.

2 The scope of the project

2.1 Scope

The goods exchange aka barter system exists with a broader perspective in itself. It consists of a wide variety of functionality, category and it is practically being used in business globally as it is one of the oldest forms of doing business. Today's companies are using barter for various reason, from getting rid of excess inventory to finding the new customers to conserving cash and supporting the local economy. Since this is an academic project it is going to cover only a few portions of such a large project that will demonstrate how the barter goods exchange system is work.

2.2 Limitation

- The system will only act as a prototype of the real goods exchange system.
- The system will not be going to identify if the user is legitimate or not.
- The system cannot detect whether the goods or items condition is good or defect.

2.3 Aims

- To provide a platform where people can meet people in their community and exchange their goods that are no longer in use or want but maybe the goods or items can help other peoples.
- To analyze how people, react to continue practicing such an old method of exchanging goods using technology.

2.4 Objectives

- To bring stability and smoothness in the economic function as this system demand does not depend on price.
- To provide a direct method of exchanging goods and services.
- To help small business grow and move products while meeting their own needs.

2.5 Overview of the Scope

The scope of goods exchange aka barter system can be relevant and thriving in today's technology worlds. With a technology intelligent platform, combined with the community aspect, goods exchange system can be relevant.

3. Development methodology

3.1 Description of the methodology

There are so many development methodologies available in the market each of them has their own pros and cons which should be understood clearly that helps to pick right solution to develop the project.

For my project, I have a clear mindset to use Dynamic System Development Model (DSDM) method as it is an iterative and incremental approach which provide software development system within the specific time frame. Though my project only covers a small portion of such a huge project I may find this project fascinating later and want to continue developing the system even though my academic project finished.

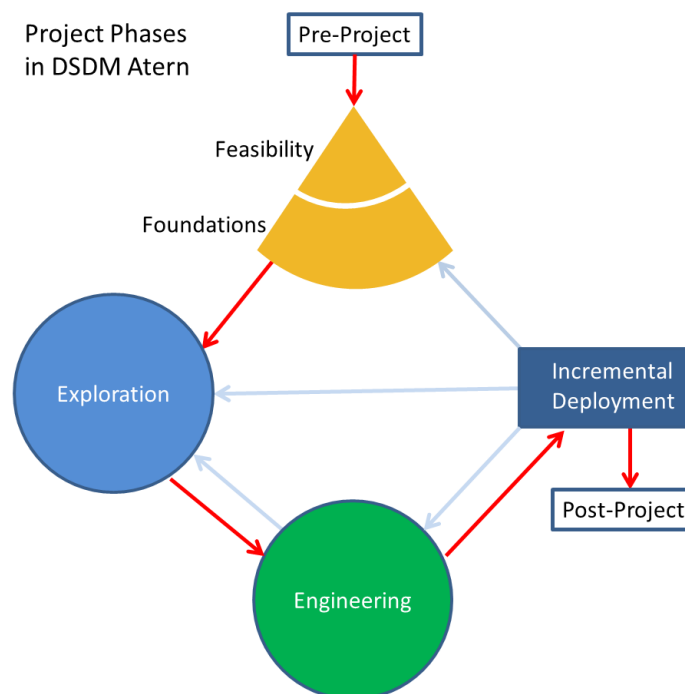


Figure 1 DSDM model

Some features of DSDM model are as follow: -

- It offers easy access to end users by the developers.
- Functionality deliverable is quick.
- Reversible change during development
- Integrate testing throughout the life cycle.

The step by step process of agile development model is shown in figure below.

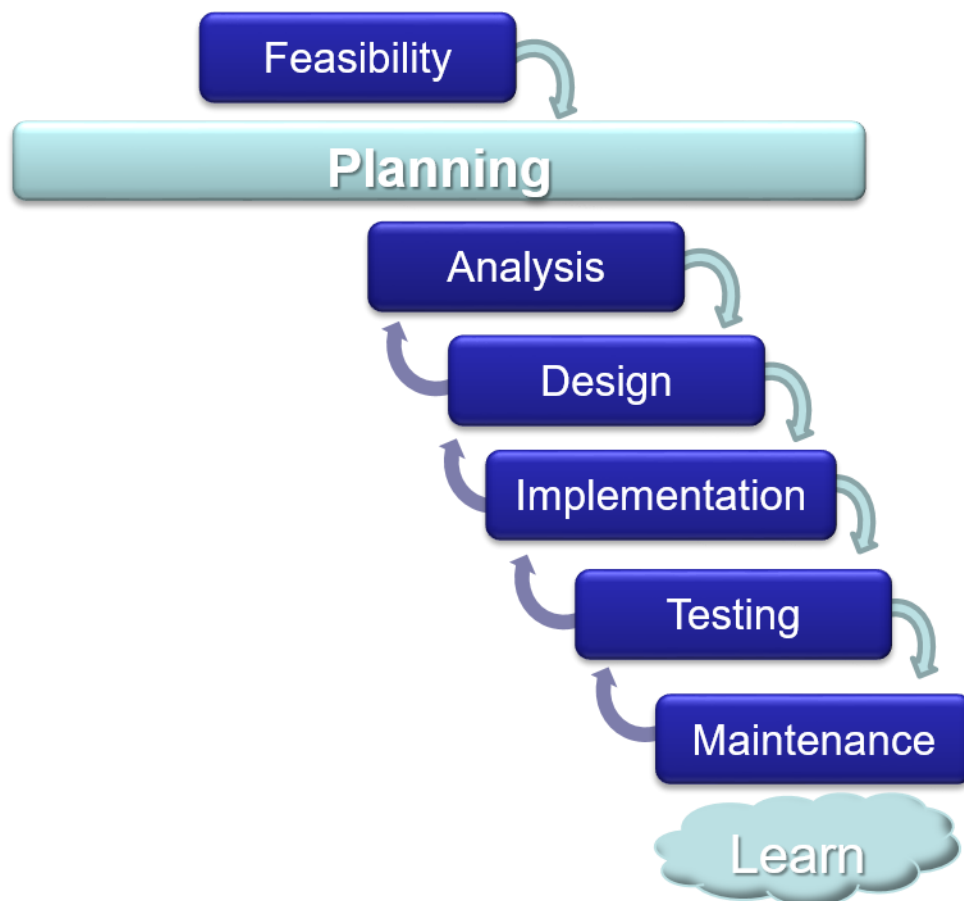


Figure 2 Agile Development Process

The difference between the old method and agile method with justification are given below: -

Agile vs Iterative vs Waterfall – {Process}

	Waterfall	Iterative (hybrid)	Agile
Quality	Quality focus changes from Analysis > Design > Code > Test	Quality focus shifts between Analysis/Design phase to Coding/Testing phase	Quality focus on all aspects of SDLC at any given time.
Quality Control	Detection & fixing during system and regression testing at the last phase of project.	Early detection & fixing in each iteration for new features. Followed by regression testing.	Early detection & fixing in each sprint followed by stabilization.
Continual Improvement (CA & PA)	Lessons learned from previous release implemented in next release	Lessons learned from previous iteration implemented in next iteration.	Lessons learned from previous sprint implemented in next sprint
Risk	No Risk Identification. Firefighting during testing phase.	Risk identification & mitigation in dev & test phase of each iteration.	Early identification & mitigation in every sprint.
Postmortem/ Retrospection	After every release	After every iteration/ milestone	After every sprint in retrospection meeting
Customer Feed back	At the end of the project.	At the end of every iteration	At the end of every sprint

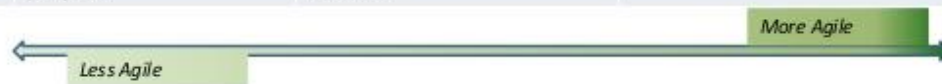


Figure 3 Comparison between old and agile method process

3.2 Design Pattern

MVC design pattern is considered one of the oldest design patterns for web application and is much more flexible to use. The acronyms of MVC is Model View and Controller and is used to separate the application's concern.

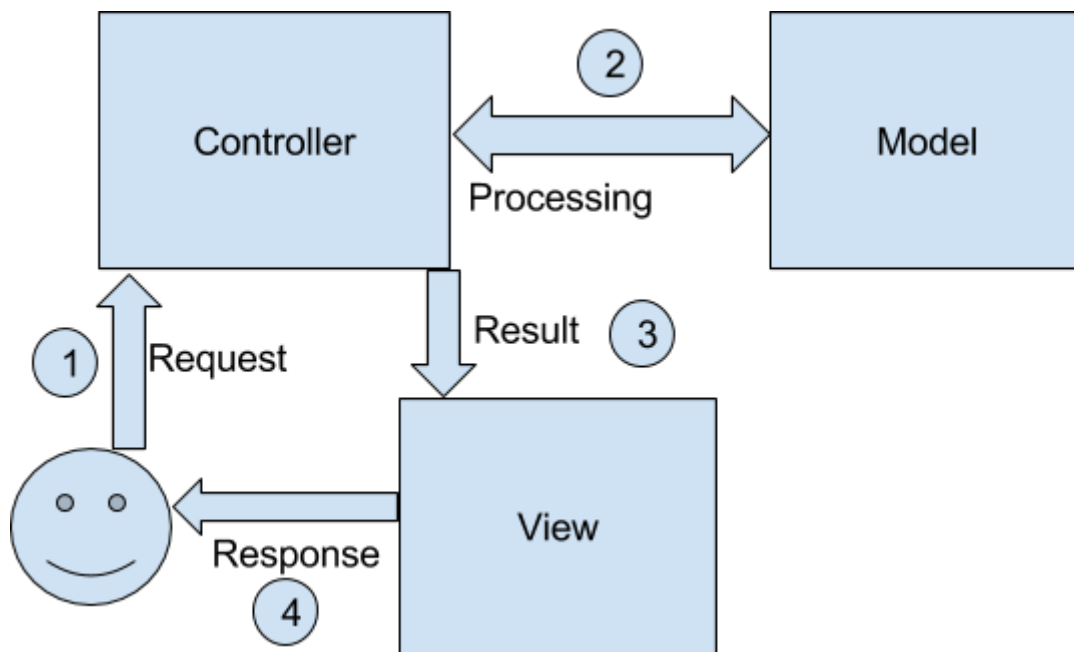


Figure 4 MVC design pattern

Model: it consists of pure data, no logic and no description of how to present the data to the users.

View: It is used to visualize the data that the model contains.

Controller: It is a mediator between model and view. It controls the flow of data between the model object and updates the view whenever data changes.

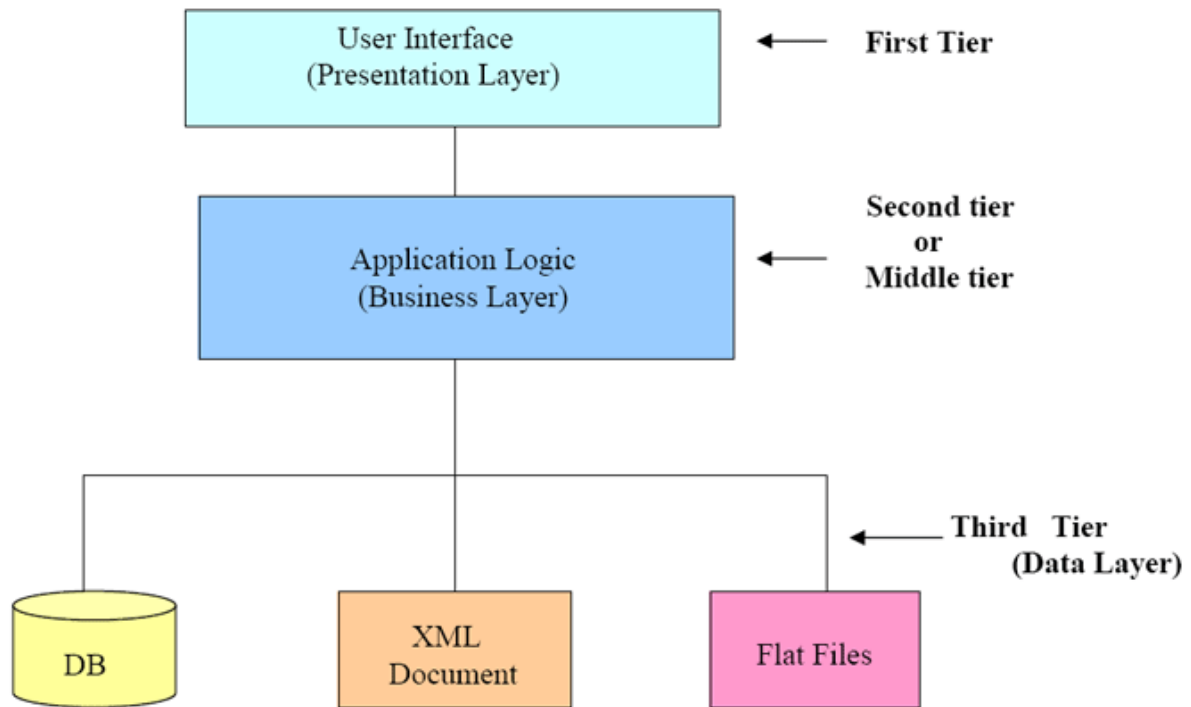
3.3 Architecture

Architecture in term of software development methodology is considered as a client-server architecture. Generally, it is a three-tier architecture. The three layer in the application namely presentation, Business and Data Layer.

Presentation Layer: The uppermost layer also known as the client layer. Using this layer user can access the webpage. The main functionality of this layer is to communicate with the application layer.

Business Layer: This layer also known as an application or logic layer. It contains the functional logic which is used in the application to drives the core capabilities.

Data Layer: In this layer, data is stored through which application communicate with the database to retrieve data.



Typical Three Tier Architecture

Figure 5 Architecture

3.4 Development Tools

Development Environment	Spring Tool Suit version 3.9.0
Programming Language used	Java 8 jre/jdk1.8.0_131 and JavaScript ES5
Framework used	Spring framework, Hibernate ORM 5 or above
Web Server used	XAMPP Control Panel V 3.2.2
Database used	MySQL
Modeling Tool used	Visual Paradigm CE 10.1
UI Design	CSS, Bootstrap 4 and JavaScript

Figure 6 Development Tools

4. Work Breakdown Structure(WBS) / Project Planning

4.1 About WBS

WBS is a project management method which helps to decompose a complex, multi-step project into simple manageable chunks. It provides definition and details at each level of the work. It also structures the project in hierarchical order in which folder and sub-folder can be created and even go further to divide the individual task into subtask.

Project WBS

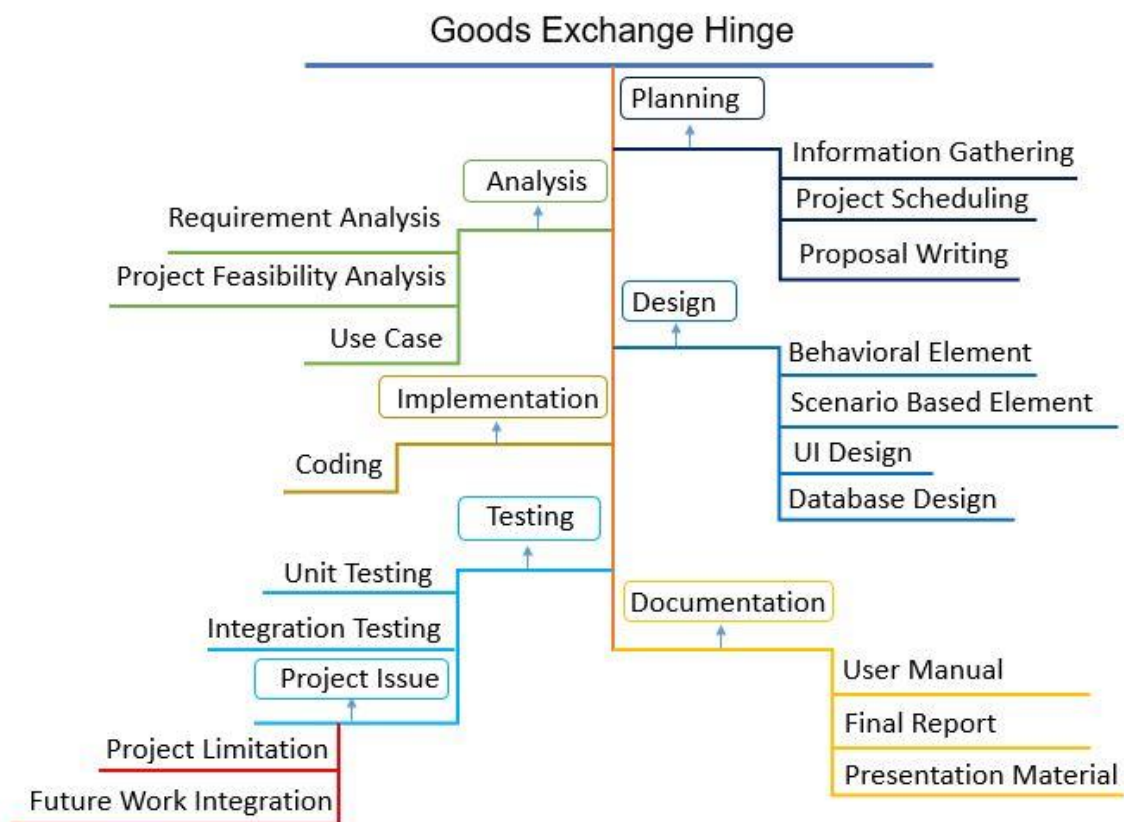


Figure 7 Project work breakdown structure

WBS	Task Name	Time Estimation in days
0.0	Goods Exchange Hinge	Total 109 Days
1	Planning	15
1.1	Information Gathering	6
1.2	Project Schedule	4
1.3	Proposal Writing	5
2	Analysis	29
2.1	Requirement Analysis	11
2.2	Project Feasibility Analysis	9
2.3	Use Case	9
3	Design	26
3.1	Behavioral Element	3
3.2	Scenario Based Element	4
3.3	UI Design	14
3.4	Database Design	5
4	Implementation	21
4.1	Coding	21
5	Testing	7
5.1	Unit Testing	2
5.2	Integration Testing	2
6	Project Issue	3
6.2	Project Limitation	2
6.3	Future Work Integration	1
7	Documentation	11
7.1	User Manual	4
7.2	Final Report	5
7.3	Presentation Material	2

Figure 8 Project WBS with time estimation

4.2 Milestone

This project has various milestones which assures that the specific development step is completed successfully. The major milestones are enlisted below as:

- Planning
- Analysis
- Design
- Implementation
- Testing
- Documentation

These milestones are directly related to the major steps of SDLC (Software Development Lifecycle). All the milestones are briefly discussed below:

- Planning

Throughout this phase, the project will have concluded Information gathering, project scheduling and proposal writing as per the days' approximation mentioned above.

- Analysis

In this phase, we conclude Requirement Analysis, Project Feasibility Analysis and Use case diagram.

- Design

During this phase, within the approximated days, we will have concluded Behavioral element, Scenario Based element, UI and database design.

- Implementation

The coding tasks are concluded in this particular

- Testing

The entire testing tasks which includes Unit testing, Integration testing, and related project issues are addressed.

- Documentation

Documentation of user manual, Final report and other presentation material are concluded in this particular phase within the approximated days.

The project milestone has been concluded following the appropriate steps which is show in figure 8.



Figure 9 Project Milestone

4.3 Project Schedule

		Name	Duration	Start	Finish	Predecessors
1		Planning	15 days	3/26/19 8:00 AM	4/9/19 5:00 PM	
2		Information Gathering	6 days	3/26/19 8:00 AM	3/31/19 5:00 PM	
3		Project Schedule	4 days	4/1/19 8:00 AM	4/4/19 5:00 PM	2
4		Proposal Writing	5 days	4/5/19 8:00 AM	4/9/19 5:00 PM	3
5		Analysis	29 days	4/5/19 8:00 AM	5/3/19 5:00 PM	
6		Requirment Analysis	11 days	4/5/19 8:00 AM	4/15/19 5:00 PM	3
7		Project Feasibility Analysis	9 days	4/16/19 8:00 AM	4/24/19 5:00 PM	6
8		Use Case	9 days	4/25/19 8:00 AM	5/3/19 5:00 PM	7
9		Design	26 days	4/25/19 8:00 AM	5/20/19 5:00 PM	
10		Behavioral Element	3 days	4/25/19 8:00 AM	4/27/19 5:00 PM	7
11		Scenario Based Element	4 days	4/28/19 8:00 AM	5/1/19 5:00 PM	10
12		UI Design	14 days	5/2/19 8:00 AM	5/15/19 5:00 PM	11
13		Database Design	5 days	5/16/19 8:00 AM	5/20/19 5:00 PM	12
14		Implementation	21 days	5/21/19 8:00 AM	6/10/19 5:00 PM	
15		Coding	21 days	5/21/19 8:00 AM	6/10/19 5:00 PM	13
16		Testing	7 days	6/11/19 8:00 AM	6/17/19 5:00 PM	
17		Unit Testing	2 days	6/11/19 8:00 AM	6/12/19 5:00 PM	15
18		Integration Testing	2 days	6/13/19 8:00 AM	6/14/19 5:00 PM	17
19		Project Issue	3 days	6/15/19 8:00 AM	6/17/19 5:00 PM	18
20		Project Limitation	2 days	6/15/19 8:00 AM	6/16/19 5:00 PM	
21		Future Work Integration	1 day	6/17/19 8:00 AM	6/17/19 5:00 PM	20
22		Documentation	11 days	6/17/19 8:00 AM	6/27/19 5:00 PM	
23		User Manual	4 days	6/17/19 8:00 AM	6/20/19 5:00 PM	20
24		Final Report	5 days	6/21/19 8:00 AM	6/25/19 5:00 PM	23
25		Presentation Material	2 days	6/26/19 8:00 AM	6/27/19 5:00 PM	24

Figure 10 Project Schedule

4.4 Gantt Chart

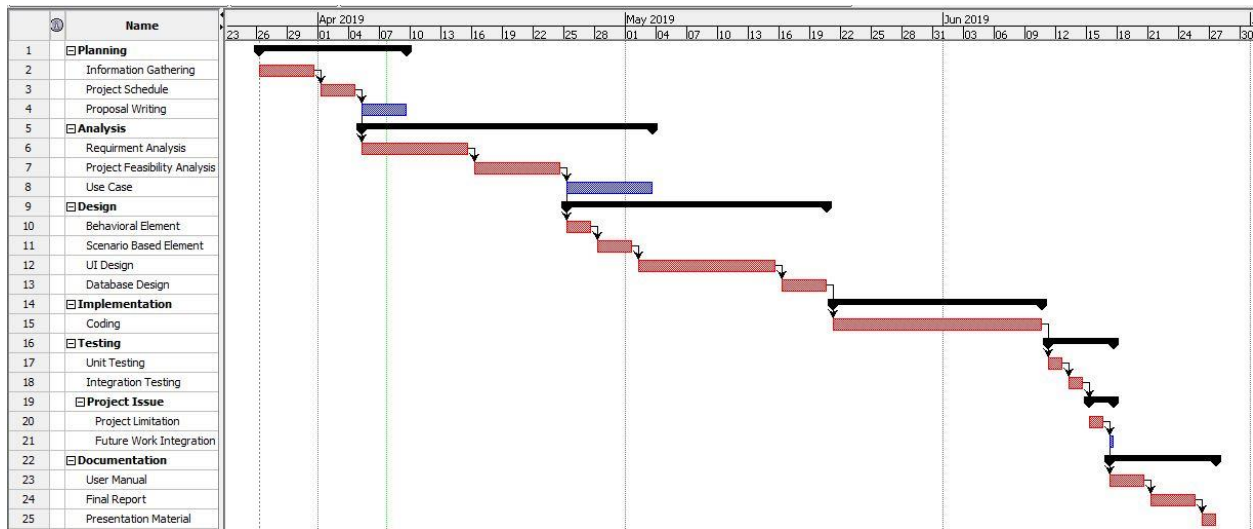


Figure 11 Gantt Chart

5 Risk Management

Risk Management has the involvement of two words **risk** which means an uncertainty event that may have a negative or positive impact on any project similarly, **management** it is the process of identifying the risk and migrating them. Risk management is important in any project because knowingly or unknowingly we are getting some kinds of risk that will be affecting all aspect of the project whether it is cost, scope or project schedule.

5.1 Lifecycle

Below are the basic steps of risk management lifecycle: -

- **Identification:** - The first steps is to discover the risk and define them i.e. identifying the areas where those risk may affect the project.
- **Assessment:** - The second steps is to measure risk likelihood and impact. Once, the risk is identified this will define the risk priorities.
- **Treatment:** - After the risk is prioritized, the risk must be defined and identify which risk require to take action.

- **Monitoring:** - After the above stages completed risk cannot be left. It must require a review process. The point to remember here is a risk is evolutionary and can change that is why a review process is essential.
- **Reporting:** - The above steps are the core stages that helps to take effective decision for risk management, therefore, the reporting process should be defined at an early point in the risk management.

5.2 Steps

The risk management matrix table: -

	Likelihood
Low	Happens very rarely and less impact
Medium	Somewhere between low and high impact
High	Chances are more likely to occurs so more impact

	Impact			
Likelihood		Low	Medium	High
	Low	Very Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	Very High

Figure 12 Risk Matrix

List of Risk	Likelihood	Impact	Impact of Risk	Action
Device Damage	<i>Low</i>	<i>Medium</i>	<i>Low</i>	Backup project into multiple places like online storage and devices
Software Failure	<i>Medium</i>	<i>Low</i>	<i>Low</i>	Do not upgrade or update the development tool until the project finished
Hard Disk Crash	<i>Low</i>	<i>Low</i>	<i>Very Low</i>	Backup project into multiple devices and online storage
Natural Disaster	<i>Low</i>	<i>Low</i>	<i>Very Low</i>	Daily backup required
Sticky bugs and complex algorithm	<i>High</i>	<i>Medium</i>	<i>High</i>	Make the algorithm simple and get help from an expert or module teacher if required
Deadline Overdue	<i>Medium</i>	<i>Medium</i>	<i>Medium</i>	Contingency required
Requirement Changes	<i>Low</i>	<i>Low</i>	<i>Very Low</i>	Use of proper development methodology
Health Issue	<i>Medium</i>	<i>Low</i>	<i>Low</i>	Take a break, do exercise and work

Figure 13 Risk Management

6 Configuration Management

Configuration management is a term mostly used in the software industry to keep control on various version of the program. The data and documentation are managed and are configured during the course of a software development initiative.

```
Windows PowerShell
PS E:\Contemplate\Fourth Semester\CP_Project> tree
Folder PATH listing
Volume serial number is C66C-02A7
E:..
├── Analysis
├── Backup
│   ├── Database Backup
│   └── Project Backup
├── Design
│   ├── Databse Design
│   └── UI Design
├── Documentation
│   ├── Final Doc
│   ├── Presentation Material
│   └── User Manual
├── Implementation
│   └── Coding
├── Planning
│   └── Proposal
├── Testing
│   └── Project Issue
PS E:\Contemplate\Fourth Semester\CP_Project>
```

Figure 14 Project Directory Structure

6.1 Reverse Engineering

Reverse engineering is the process of creating of design document from source code and the specification document from the design document.

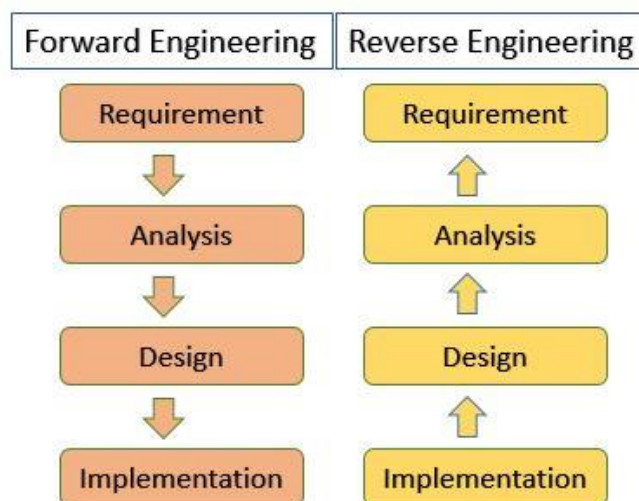


Figure 15 Forward and Reverse Engineering

Basically, reverse engineering requires to understand the program comprehension.

The goal of reverse engineering:

- Cope with complexity
- Recover lost information
- Facilitate reuse

6.2 Version Control System (VCS)

Version control system is used to record changes to a file or set of files over specific time. It keeps track of every modification to the code in a specific kind of database. If any mistake happens then it allows to revert files back to the previous state or revert the entire project back to the previous state. There are so many free VCS systems available but I am going to use GitHub which is free. To help myself from the wise thing and recover from complexity if any mistake happens during the project I have created a GitHub repository for my project is: - [Mohammadhusssain257 / Goods_Exchange_Hinge](#)

7 Conclusion

Finally, the proposal of this Goods Exchange Hinge project have been proposed which consumes a lot of research and time to collect facts and figure related to this project. A goods exchange hinge aka barter system is a method of exchanging goods in the local community. This system is particularly focused to solve a problem related to good exchange by evaluating the value of goods and provide features like whether the goods made available free of cost or take some amount instead. The proper steps like project scope, limitation and methodology have followed. The clear point about why to use DSDM agile methodology has been stated. As this is an academic solo project the whole project schedule, milestone, WBS, risk management, project configuration, and time limitation has clearly stated. The project should be completed within 3 months.

8 Reference

Barter System History

Available at: <https://www.mint.com/barter-system-history-the-past-and-present>

Access Date: 2019-03-28

Barter in business

Available at:

https://www.1stsource.com/advice/business/SBR_template.cfm?DocNumber=PL08_0080.htm,
<https://www.forbes.com/sites/patrickhull/2013/07/05/bartering-is-a-great-strategy-for-success/#59c69ad317b6>, <https://www.entrepreneur.com/article/78334>

Access Date: 2019-03-29

Development methodology

Available at: <https://acodez.in/12-best-software-development-methodologies-pros-cons/>,
<http://dsdmofagilemethodology.wikidot.com/>, <https://www.agilebusiness.org/content/choosing-dsdm-your-agile-approach-0>

Access Date: 2019-04-01

Design Pattern

Available at: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm,
<https://www.geeksforgeeks.org/mvc-design-pattern/>, <https://www.journaldev.com/16974/mvc-design-pattern>

Access Date: 2019-04-01

Architecture

Available at: <https://www.techopedia.com/definition/24649/three-tier-architecture>,
<https://www.softwaretestingmaterial.com/software-architecture/>

Access Date: 2019-04-03

Project configuration management

Available at: <https://www.wrike.com/project-management-guide/faq/what-is-configuration-management-in-project-management/>

Access date: 2019-04-03

Reverse Engineering

Available at: <https://www.techopedia.com/definition/3868/reverse-engineering>,
<http://reversingproject.info/>

Access Date: 2019-04-04

Version control system

Available at: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>,
<https://www.atlassian.com/git/tutorials/what-is-version-control>

Access Date: 2019-04-05