Table of Contents

[List of Diagram 3](#_Toc14170709)

[Abstract 6](#_Toc14170710)

[Acknowledgement 7](#_Toc14170711)

[Chapter 1: Introduction 8](#_Toc14170712)

[1.1 Background of System 8](#_Toc14170713)

[1.2 Overview of Project 8](#_Toc14170714)

[1.3 Aims and Objectives 9](#_Toc14170715)

[Chapter 2: Analysis 10](#_Toc14170716)

[2.1) Introduction to Analysis 10](#_Toc14170717)

[2.2) Analysis Methodology 11](#_Toc14170718)

[2.3) Requirement Analysis 14](#_Toc14170719)

[2.3.1) Functional Requirement 14](#_Toc14170720)

[2.3.2) Non-Functional Requirements 17](#_Toc14170721)

[2.3.3) MoSCoW Prioritization 19](#_Toc14170722)

[2.4.4) Software Requirements Specification (SRS) 21](#_Toc14170723)

[2.4) Use Case Diagram 23](#_Toc14170724)

[2.5) NLA (Natural Language Analysis) & Initial Class Diagram 28](#_Toc14170725)

[Chapter 3: Design 31](#_Toc14170726)

[3.1) Structural Modeling 32](#_Toc14170727)

[3.1.1) Final Class Diagram 32](#_Toc14170728)

[3.1.2) Flow Chart 32](#_Toc14170729)

[3.2) Behavioral Modeling 34](#_Toc14170730)

[3.2.1) Activity Diagram 34](#_Toc14170731)

[3.2.2) Sequence Diagram 38](#_Toc14170732)

[3.3) Database Modeling 45](#_Toc14170733)

[3.3.1) Data Dictionary 45](#_Toc14170734)

[3.3.2) ER Diagram 48](#_Toc14170735)

[3.4) Architectural Design 49](#_Toc14170736)

[3.5) Paper Prototyping 50](#_Toc14170737)

[Chapter 4: Implementation and Coding 60](#_Toc14170738)

[4.1 Introduction 60](#_Toc14170739)

[4.2 Programming Language 60](#_Toc14170740)

[4.3 Tools Used 60](#_Toc14170741)

[4.4 Code and Screenshots 61](#_Toc14170742)

[Chapter 5: Testing 98](#_Toc14170743)

[5.1 Introduction 98](#_Toc14170744)

[5.2 Types of Testing 98](#_Toc14170745)

[5.2.1 Unit Testing 98](#_Toc14170746)

[5.2.2. Black Box Testing 98](#_Toc14170747)

[Chapter 6: Other Project Issues 107](#_Toc14170748)

[6.1 Project Limitation 107](#_Toc14170749)

[6.2 Risk Management 107](#_Toc14170750)

[6.3 Configuration Management 109](#_Toc14170751)

[Chapter 7: Conclusion 111](#_Toc14170752)

[Chapter 8: References 111](#_Toc14170753)

[Chapter 9: Appendix 111](#_Toc14170754)

[References 112](#_Toc14170755)

# List of Diagram

[Figure 1: Rich Picture 10](#_Toc14170756)

[Figure 2: Conceptual Diagram 12](#_Toc14170757)

[Figure 3: MoSCoW Prioritization 18](#_Toc14170758)

[Figure 4: Use Case Diagram of Admin 23](#_Toc14170759)

[Figure 5: Use Case diagram of Unregistered User 24](#_Toc14170760)

[Figure 6: Use Case diagram of Registered User 25](#_Toc14170761)

[Figure 7: NLA Candidate Lists 28](#_Toc14170762)

[Figure 8: Initial Class Diagram 29](#_Toc14170763)

[Figure 9: Flow Chart of a System 32](#_Toc14170764)

[Figure 10: Notation used for Activity Diagram 33](#_Toc14170765)

[Figure 11: Login and Registration Activity Diagram 34](#_Toc14170766)

[Figure 12: Activity Diagram of admin 35](#_Toc14170767)

[Figure 13: Activity Diagram of User 36](#_Toc14170768)

[Figure 14: Notation used for making Sequence Diagram 37](#_Toc14170769)

[Figure 15: Registration Sequence Diagram 38](#_Toc14170770)

[Figure 16: Login Sequence Diagram 39](#_Toc14170771)

[Figure 17: Admin Privilege for user 40](#_Toc14170772)

[Figure 18: Sequence Diagram of Products 41](#_Toc14170773)

[Figure 19: User Sequence Diagram 43](#_Toc14170774)

[Figure 20: Data Dictionary 46](#_Toc14170775)

[Figure 21: ER Diagram of a System 47](#_Toc14170776)

[Figure 22: Client Server Architecture 48](#_Toc14170777)

[Figure 23: Admin Login 49](#_Toc14170778)

[Figure 24: View Available Products 49](#_Toc14170779)

[Figure 25: View Registered Users 50](#_Toc14170780)

[Figure 26: Add Product 51](#_Toc14170781)

[Figure 27: Update Product 52](#_Toc14170782)

[Figure 28: Admin edit user 53](#_Toc14170783)

[Figure 29: Register New user 54](#_Toc14170784)

[Figure 30: Update User’s Details 55](#_Toc14170785)

[Figure 31: Change Password 56](#_Toc14170786)

[Figure 32: Forum Ask Question 56](#_Toc14170787)

[Figure 33: Forum Comment 57](#_Toc14170788)

[Figure 34: Review for testimonial 57](#_Toc14170789)

[Figure 35: Order Required Product 58](#_Toc14170790)

[Figure 36: User Registration Form 60](#_Toc14170791)

[Figure 37: Database Connection 61](#_Toc14170792)

[Figure 38: User registration Action 62](#_Toc14170793)

[Figure 39:Code for User registration 65](#_Toc14170794)

[Figure 40: User Login Form 66](#_Toc14170795)

[Figure 41: User Login Code 66](#_Toc14170796)

[Figure 42: Admin Login Form 67](#_Toc14170797)

[Figure 43:Admin Login Code 69](#_Toc14170798)

[Figure 44:Contact Form 70](#_Toc14170799)

[Figure 45: Contact Form Code 72](#_Toc14170800)

[Figure 46: Subscribe Form 73](#_Toc14170801)

[Figure 47: Subscribe Code 73](#_Toc14170802)

[Figure 48: User dashboard and User Update Form 74](#_Toc14170803)

[Figure 49: User Update Code 75](#_Toc14170804)

[Figure 50: User Password Change Form 76](#_Toc14170805)

[Figure 51: User Password Change Code 77](#_Toc14170806)

[Figure 52: Users Forum Form 78](#_Toc14170807)

[Figure 53: Forum Question code and display forum question and comments 79](#_Toc14170808)

[Figure 54: Forum Post and See Question and Comment Code 82](#_Toc14170809)

[Figure 55: Forum Question Code 83](#_Toc14170810)

[Figure 56: Forum Comment code 83](#_Toc14170811)

[Figure 57: Admin Dashboard 84](#_Toc14170812)

[Figure 58: Update Admin Profile Code 84](#_Toc14170813)

[Figure 59: Add New Admin form 85](#_Toc14170814)

[Figure 60: Add Admin Code 85](#_Toc14170815)

[Figure 61: Delete Admin Form 86](#_Toc14170816)

[Figure 62: Select and Delete admin code 86](#_Toc14170817)

[Figure 63: Select registered users 87](#_Toc14170818)

[Figure 64: Edit selected user by admin 87](#_Toc14170819)

[Figure 65: Update and Delete selected user 88](#_Toc14170820)

[Figure 66: Add Product Form 89](#_Toc14170821)

[Figure 67: Add Update and Delete Product 90](#_Toc14170822)

[Figure 68: Add Update and Delete Product Code 93](#_Toc14170823)

[Figure 69: View Queries 93](#_Toc14170824)

[Figure 70: View and Delete Contact Code 93](#_Toc14170825)

[Figure 71: View Subscribed Users 94](#_Toc14170826)

[Figure 72: View Subscriber Code 95](#_Toc14170827)

[Figure 73: Select and Delete Subscriber Code 95](#_Toc14170828)

[Figure 74: Search Form and Result 96](#_Toc14170829)

[Figure 75: Search Form Code 96](#_Toc14170830)

[Figure 76: Ajax and Fetch Search Data Code 96](#_Toc14170831)

# Abstract

For the project, I have use Object Oriented PHP for developing the application. PHP MySQL for database and local server for running my application.

The project aims to provide automatic recording of customer/ information replace the paper work which had negative effect on cost and time. Customer may feel lazy to visit shop which can effect in the sales. The project helps the customer to order the goods they want with their satisfying cost. The project helps to reduce time consumption of both customer and the shop, deals with faster sale. The product detail can be seen but can’t see it physically. For the paper work, more staff is needed to the job, the detail may have lost but this project reduces cost and backup. The project helps the shop to get profit as less staff is needed to do job.

Also, project aims to provide service to people who don’t have an online store so that the people can add their product in our store as per the request. In this way, the people product can be sale through online.

I research about the people regarding this project, they find it problem to visit many store to find their choice product. People feels tired after visiting all the shop and feels disappoint if not found. So, this project make the people feel little flexible and easy.

User/admin can create an account and login to the system. Admin add, view, update and delete the product. Admin can view the messages send by customer. User can view the product added by admin and can order the specific product they want. User can search their needy product.

# Acknowledgement

This project has been developed in order to fulfill the requirement buying of Nepali Handmade products online. Although this is the individual project topic, I should remain thankful to many persons for the successful completion of the project.

First of all, I am thankful to my respected Module Leader **Mr. Sudeep Lal Bajimaya** for his persistence help and guidance throughout my academic study. His suggestion and guidance in every stage is one of the major reasons of the successful completion of my project. Without his guidance my project would not have been accomplished in time.

I am also thankful to my beloved friends for their kind support and encourage to me for the completion of the project. I would also like to add my heartfelt appreciation to my parents for their infinite kindness and support throughout my academic career.

At last, I am very thankful to all the respected teachers of our College for helping and encouraging me in every aspect of my academic study during college. And many thanks to **Softwarica College of IT and E-Commerce** for providing me the opportunity to pursue my education in such wonderful academic environment.

# Chapter 1: Introduction

Nepali Handicraft Information Management System is a web-based application that is developed to provide good and better facility to the customer with its best services. It is a non-profit organization of private sector business and artistic community. It helps to introduce the best and good quality handicraft items to the world. This system also equally manages product more systematically and scientifically by providing all the necessary details of the product.

This web-based application is developed using Core PHP: Hypertext Preprocessor (PHP) and MySQL. For the development of this application I chose Object Oriented PHP.

## Background of System

Until now, all the transactions which is taking place Ordering Products, Creating Bills, Recording Customers and their information, are manually hand held. This led shop backwards in every factors like time, cost, working environment and so on. Increasing in daily customer and growing business made this very difficult to do all the day to day transactions.

The main purpose of this project is to develop an application that will help to reduce all the problem that are mentioned above. This application helps customer to involve in buying and selling their handcrafted product using their smartphones and other means of communication. Use of proper managed storing tools will help to overcome all those difficulties in no time.

## Overview of Project

Collecting handmade product is an interest for some individuals. Its values are significantly more compared to other normal products. People are less mindful of the varieties of products that the store has and many individuals are not mindfully aware of the old and traditional values that hand craved carries along with it. It is within the aim of the store to publicize their items, yet principally maintaining and managing their day by day transactions held with their clients and wholesalers.

This project is supposed to help to fulfil this aim. It helps to a web application that gives the clients to place order of their product, also order receipts through internet. It additionally enables client to get information instantly if necessary, which can contribute in production and advancement of handcrafted items. It likewise informs the clients or customers about ongoing activities. The project guarantees to build up an application that is helpful to utilize and such that customers are motivated to continue engaging with the store.

## Aims and Objectives

The aims of the project are listed below:

* To mechanize and simplify every day transaction of the store
* Keeping business relations unblemished
* Keep clients or customer connected with the store
* Make good negotiation between any two parties while trading items

The objectives of the project to obtain above given goals are listed below:

* Perform examination and gather requirements
* Create a user-friendly GUI
* Designing different types of diagram for the flow of the application
* Database Design
* Perform well documented testing
* Perform upkeep/maintenance
* Provide online shopping facilities
* Produce a well-recorded report or document
* Perform Maintenance time and often

# Chapter 2: Analysis

## 2.1) Introduction to Analysis

Simply, analysis is about analyzing and gathering the information of the software at various stages to know the detail of the project. It is a systematic examination and evaluation of data or information which is done to determine the needs or conditions of the project to meet the requirements of the project.

It is about understanding the problem in depth which may include quantification of capacities, security, robustness and even the uncertainties of the project, which will have factors that need to be considered carefully for the betterment of the project.

Importance of analysis for my project are explained below:

* It helps to gather the important information affecting different aspects of the costumers.
* It helps to understand the problems of local people who are unable to keep in touch with physical market place.
* It helps to meet the quality design of the project which exceeds the requirement for the problem.
* It helps on quality implementation of the project which puts all the work together so it functions as a firm entirely.
* It also helps to understand the project which are involved in different areas.

## 2.2) Analysis Methodology

The methodology that is used to undertake an analysis technique for the project development is said to be analysis methodology. Among various approaches for software development such as: Soft System Approach, Hard System Approach, Combined Approach, etc., I have chosen to take Soft System Approach for my project.

**Soft System Methodology (SSM)**, is a tool (not system design tool) for system requirement investigation which allow the designer to model how the system should operate and what operations the system should perform and more importantly why system should operate.

Simply, it is an approach in which project is analyzed in more people focused way before taking it to the hard approach.

Examples: PEST, SWOT, CATWOE, ETHICS, Rich Picture, Root Definition, etc.

**Advantages:**

* It recognizes that the user interaction is also equally important as technical considerations
* It clarifies the problem area of a system as more people interaction is involved
* It helps to bring different sectors of a project together.

Following are the steps while applying soft approach:

* Analyze and produce **Rich picture**

Rich picture is a drawing of a scenario that will help our project to illustrate the main elements and relationship that needs to be considered. It is called rich picture because it helps to illustrate the richness and the complexity of our project. It helps to explore, acknowledge and define a complex mechanism for learning about the situation. It also helps to express the situation through diagram to create a preliminary mental model.

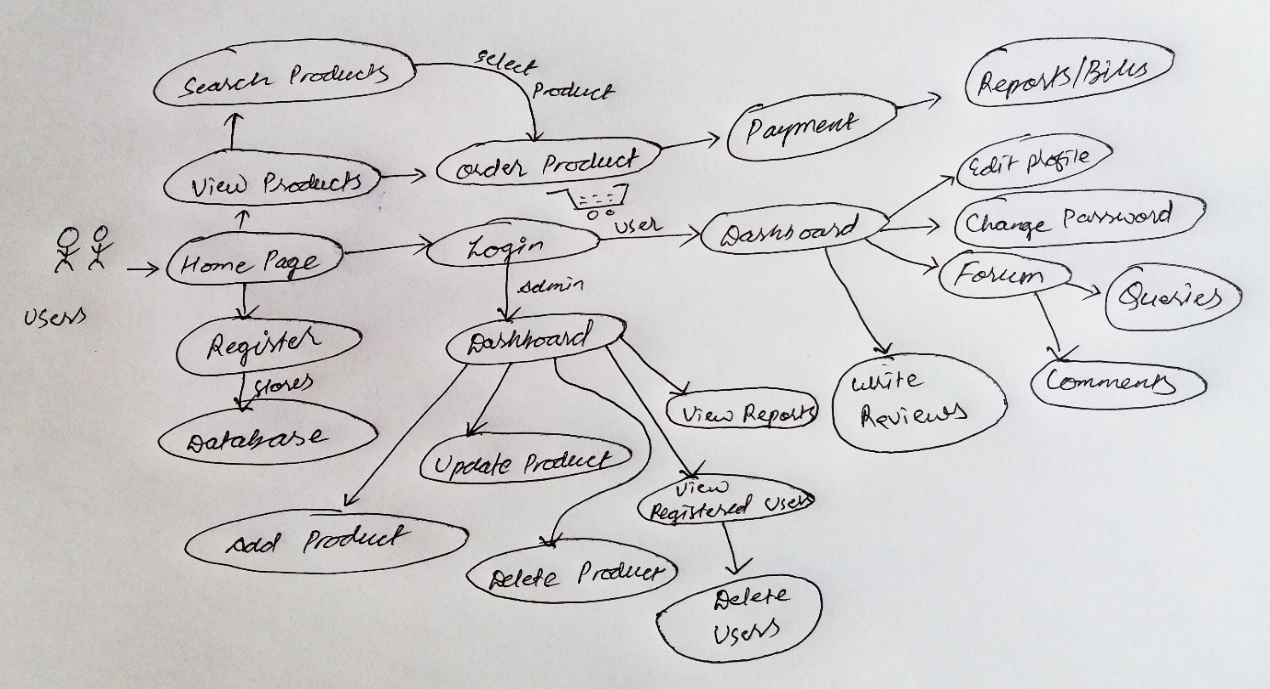
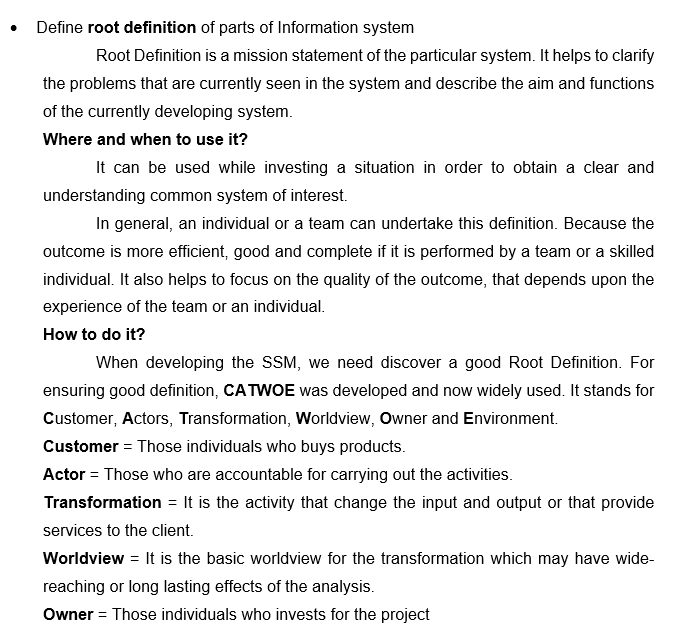


Figure : Rich Picture



**Environment** = The key constraints outside the system boundary that are significant to the system. (Burge, 2015)

* Produce **conceptual models** of system

The model which is constructed by the help of rich picture and root definition to represent a system by easily understanding the different models of the system can be said as **conceptual model**.

Primary advantage of conceptual model includes:

* **Establishes Entities**

By establishing entities and relation, various entities and concepts for the software development life cycle can be ensured.

* **Defines Project Scope**

It can be used to define project scope because it helps in managing and scheduling time.

* **Base Model for Other models**

Many other models need to be generated beyond the roughly sketched concepts, conceptual model acts as a base model for other data models such as logical data models and so on.

* **High-Level Understanding**

Conceptual Model acts as an extraordinary tool by giving high level understanding of a system throughout the software development lifecycle.

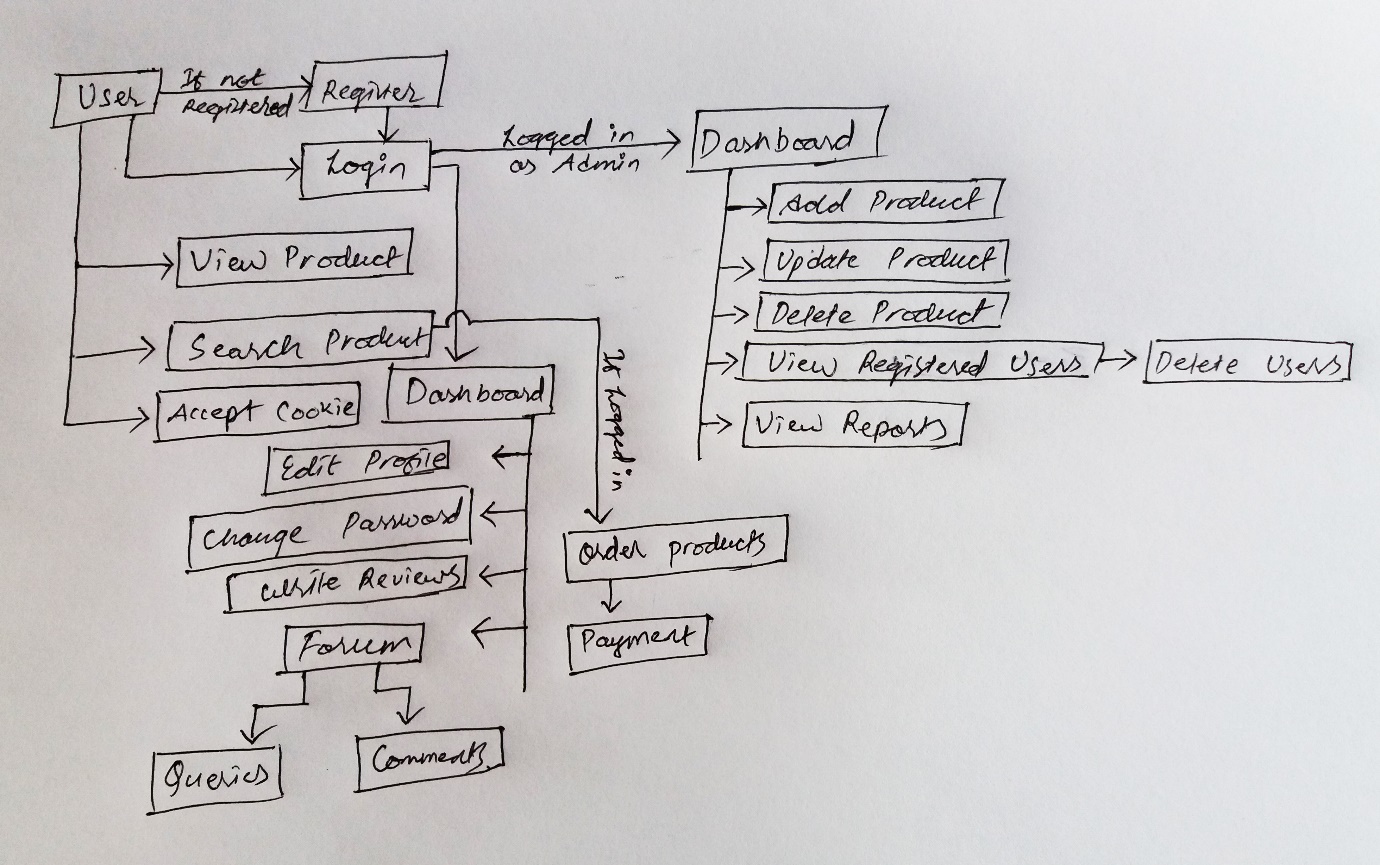
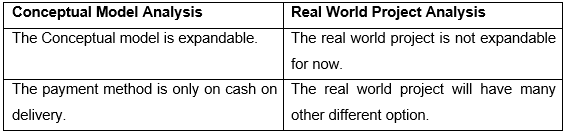


Figure : Conceptual Diagram

* Compare concept of the system with actual system

This step helps to compare two different aspect of conceptual model analysis and real world project analysis. As not everything is perfect in the model.



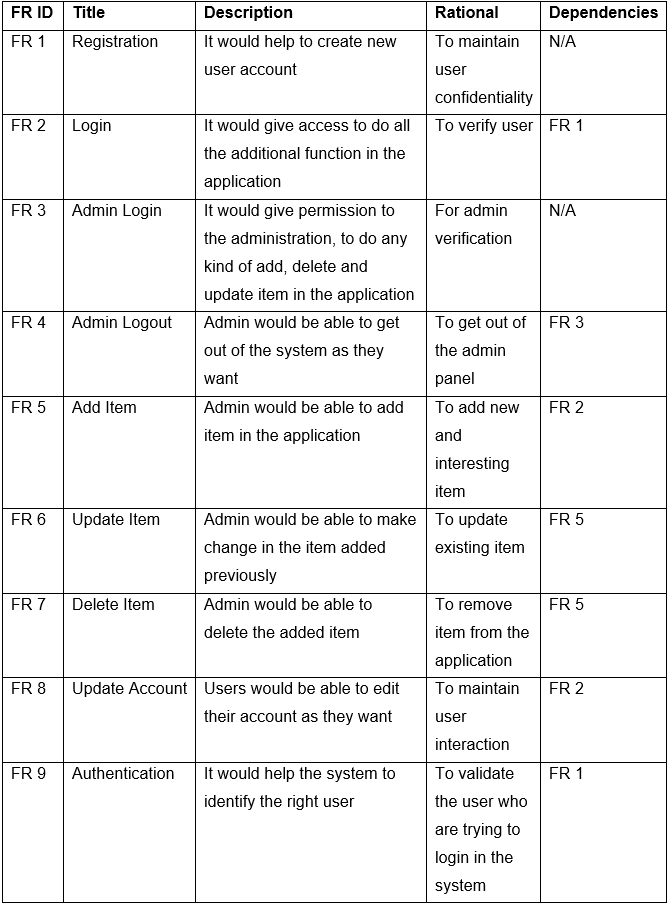
* Define and select feasible options for development
* Implement the system

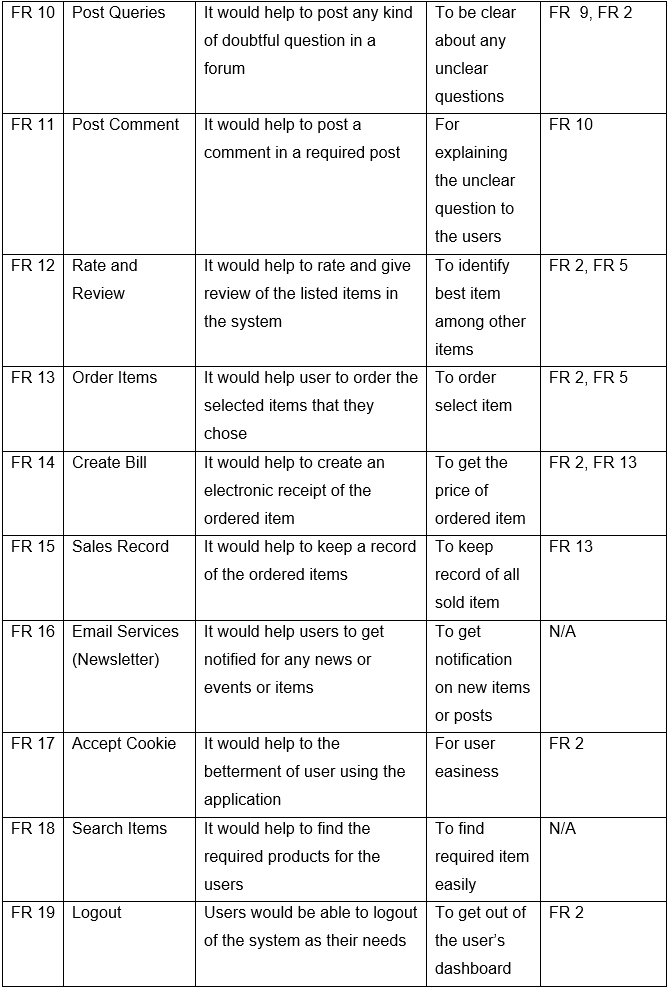
## 2.3) Requirement Analysis

### 2.3.1) Functional Requirement

Simply, Functional Requirement (FR) means, those functions or features which are most important and suitable for a project development. It can include technical, hardware-software and their functionality required to perform the user needs and tasks.

FR suitable for my project is listed in the table below:

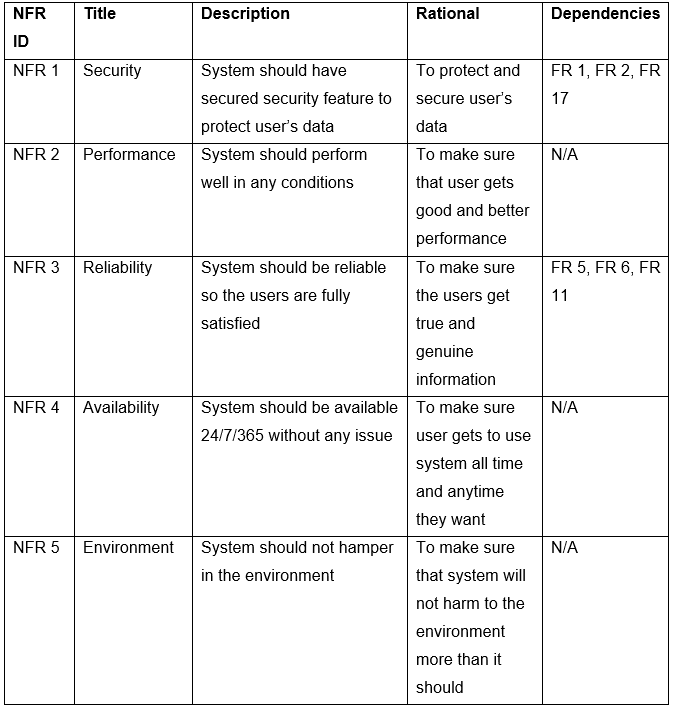


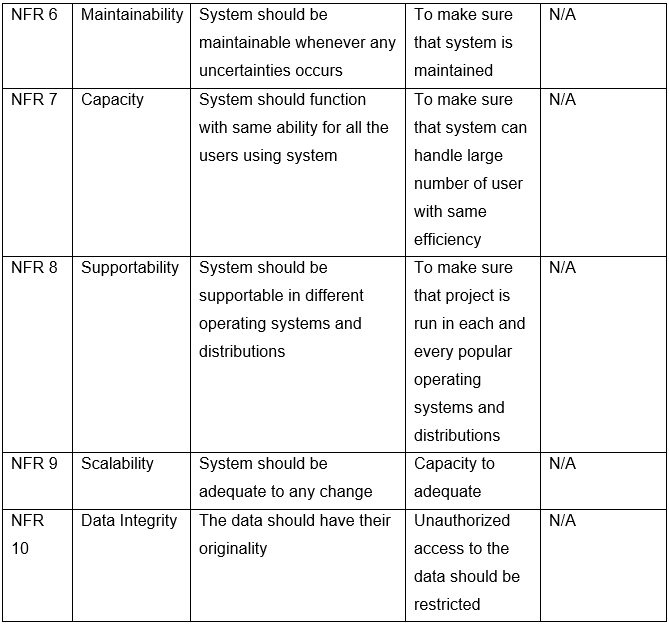


### 2.3.2) Non-Functional Requirements

Simply, Non-Functional Requirements (NFR) describes how the system attributes such as security, performance, reliability, maintainability, scalability and usability works. It is also known as system qualities which ensure the usability and effectiveness of the entire system.

NFR suitable for my project is listed in the table below:





### 2.3.3) MoSCoW Prioritization

Also known as MoSCoW method, it us used to determine which requirement should be included in a certain system delivery. It helps to minimize the risk as importance is given to the most important work of the project. Importance of MoSCoW Prioritization are explained below:

1. For classifying important task and give time, cost and effort accordingly to the projects that are in progress
2. It is simplistic and high level requirement prioritization
3. We don’t need any prior knowledge or training to understand the concept

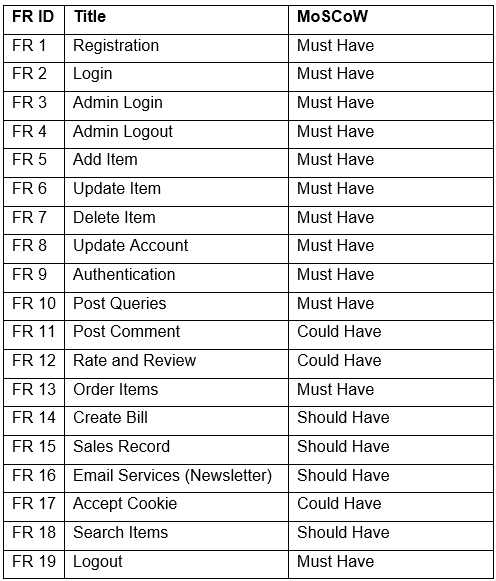


Figure : MoSCoW Prioritization

The four level of the prioritization are explained below:

* **M**ost Have: It is the most top priority items that the project need in order to move forward.
* **S**hould Have: It is not critical to launch, but priority items are considered to be important.
* **C**ould Have: It is a requirement that the priority items are desirable but not necessary.
* **W**ould not Have: It is a requirement that will not be implemented now, but such priority items may be included in a future.

MoSCoW Prioritization of Functional Requirement:



MoSCoW Prioritization of Non Functional Requirement

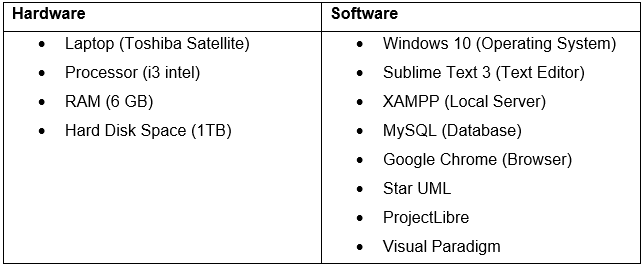


### 2.4.4) Software Requirements Specification (SRS)

A Software Requirement Specification also known as System Requirement Specification, is a set of document that helps to explain the feature and behavior of a software requirement. It also provides various hardware and software description while developing a system. The hardware and software requirements required for my project is given below.

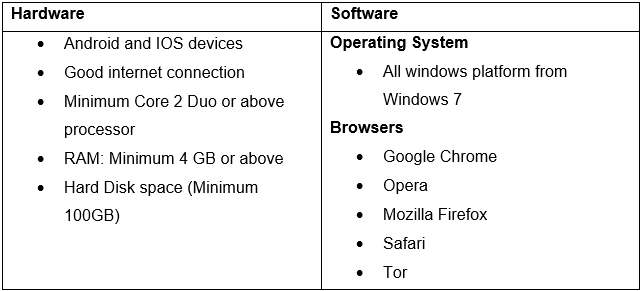
* Pre-project requirements

The hardware and software required for developing the project from the beginning to ending is said to be Pre-project requirements.



* Post project requirements

The hardware and software that is required to run the project smoothly without any bugs or error is said to be post project requirement.



## 2.4) Use Case Diagram

Use Case Diagram is the representation of a user’s interaction with the system which helps to show the relationship between the user and the different use case involved in the system. It helps to distinguish the role of the different users/actors on the function of the system.

The advantages of creating use case diagram for the project can be listed below:

1. It is easy to understand and represent the system’s functionalities
2. It helps to identify the role of different users
3. It helps to explain the system to any non-technical person

The symbol description used for this diagram are explained in short below:

1. Rectangular box helps to represent the system of the project
2. Actors helps to represent the actions in the system
3. Use Case helps to represent the action that is performed in the system
   1. Include Relationship
   2. Extend Relationship
   3. Generalization Relationship

The following use case diagrams for my project is displayed and explained below:

* + 1. Admin Use Case Diagram:

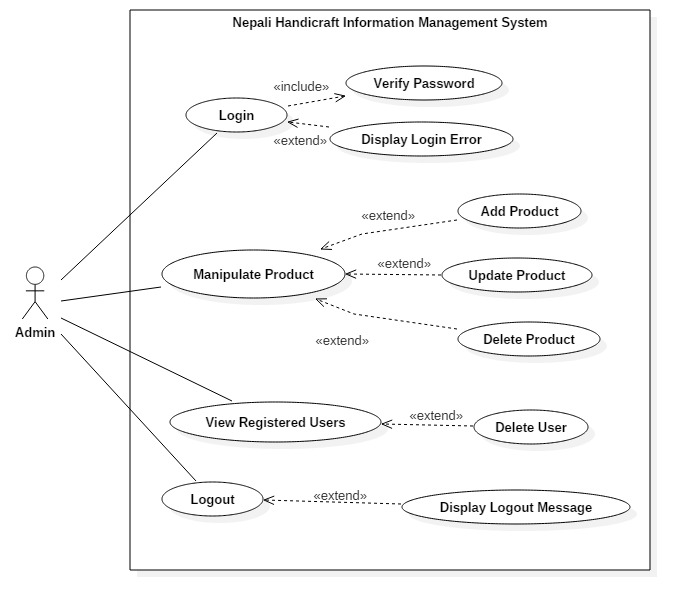


Figure : Use Case Diagram of Admin

**Scenario Description:**

Actor: Admin

The super user of the system who can do each and every activity mentioned below:

* It can Login to the system
* It can manipulate the product data, that is it can perform CRUD operations
* It can view registered users and also delete as per the user’s request
* It can logout of the system so that admin privilege is not accessible by any unauthorized people.
  + 1. Unregistered Users Use Case Diagram:



Figure : Use Case diagram of Unregistered User

**Scenario Description:**

Actor: Unregistered User

The unregistered user is also a part of the system who can do activity mentioned below:

* Unregistered User can register to the system to get login access.
* Unregistered User can login into the system and act as a user.
* Unregistered User can view the listed products which are listed by admin.
* Unregistered User can search for the required products.
  + 1. Registered Users Use Case Diagram:

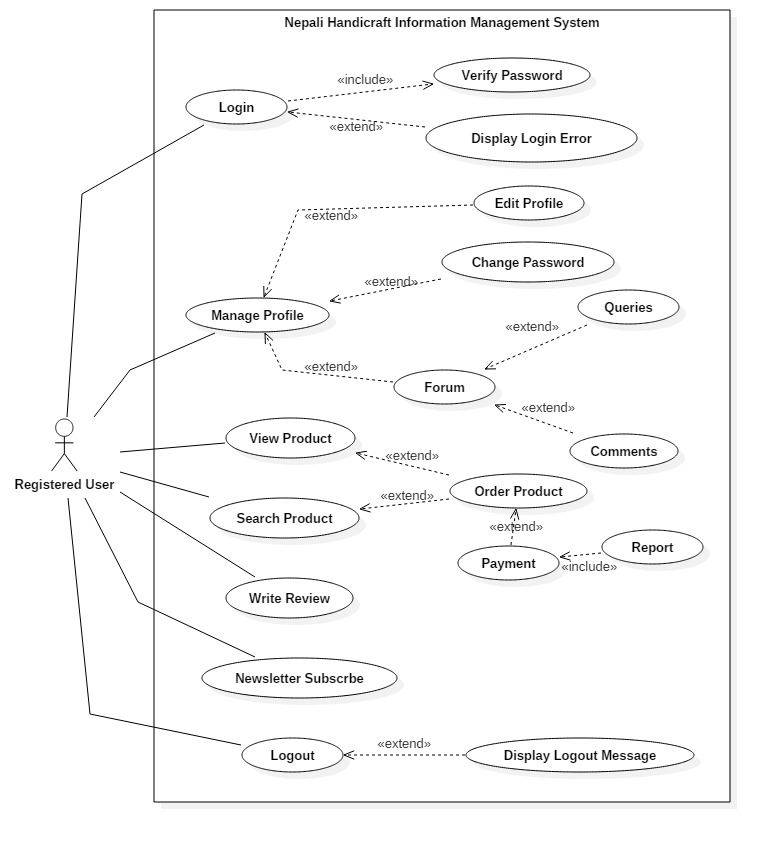


Figure : Use Case diagram of Registered User

**Scenario Description:**

Actor: Registered User

The Registered user is also a part of the system who can do activity mentioned below:

* Registered User can login into the system and act as a user for ordering products.
* Registered User can manage their profile:
  + They can update their profile information,
  + They can change password,
  + They can ask any queries regarding products,
  + They can reply to any asked queries.
* Registered User can write a review for the testimonials.
* Registered Users can enter their email address for subscription.
* Registered Users can logout of the system so that user privilege is not is not accessible by any other unauthorized person.

## 2.5) NLA (Natural Language Analysis) & Initial Class Diagram

The manner of figuring out the distinctive elements of a machine by the use of an unstructured textual content that are absolutely understandable to non-technical humans is called Natural Language Analysis. It helps to extract nouns as potential candidate class, verbs as potential candidate methods and adjectives as potential attributes. By the help of these three elements, we can easily create different diagrams such as: Class Diagram, Activity Diagram and so on.

Nepali Handicraft Information Management System (NHIMS) is a web-based application whose main aim is to improve and introduce Nepali Handicraft item to the world. It also helps to manage handicraft products more systematically and scientifically.

By studying the currently existing framework and technique, the basic procedure is clarified as far as useful technique, substance and modules. The NHIMS is designed to give all the necessary and historical information about the handcrafted products to the customer. A web-based application consists of Admin and User. Admin being a super user he/she should be able to do all the functions such as: update, add, delete products, view registered users, view reports and delete users. Being user he/she should be able to register into the system, login to update their profile, change password, ask queries and write comments. Database will hold user’s (name, email, address, gender, telephone / mobile number). Asking queries and writing comments should be on forum so that all the users can read the queries and comments to that queries. Once user logs in he/she can order the products one at a time. Payment method is done on cash on delivery. After completion of order bill should be generated. Also for unregistered user he/she should be able to view and search products.

The system should include about page, privacy policy page and contact page. It should include proper validation form with proper login access control. The database should be managed properly and also manage user’s privacy properly.

From the scenario given above we can classify class, attribute and method as following ways:

1. Class can be found by the help of nouns.
2. Attributes can be found by the help of adjectives.
3. Methods can be found by the help of verbs

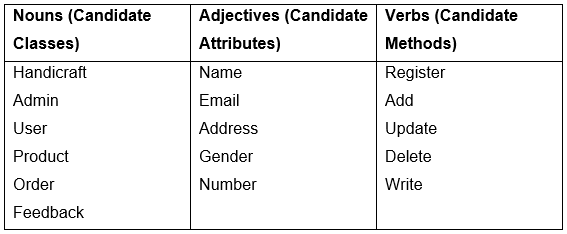


Figure : NLA Candidate Lists

**Initial Class Diagram**

Simply, Class is a blueprint of an object that defines what object can do. Class Diagram is a diagram that gives static view of a system. It helps describe the type of object and its relationship that exist among the system. By the help of this diagram almost all method can run with Object Oriented Models. (TutorialsPoint, 2017)

Purposes of class diagram are listed below:

1. Analysis and design of system in static view.
2. Helps to describe the responsibility of the system.
3. Acts as base component diagram and also helps in forward and reverse engineering.

Class Diagram consist of 3 sections’ rectangular box. At first section class’s name is kept, at second or middle section attributes are kept and on last section, methods are kept. Classes also contain 3 different access control level.

1. Public, denoted by ‘+’
2. Private, denoted by ‘-’
3. Protected, denoted by ‘#’

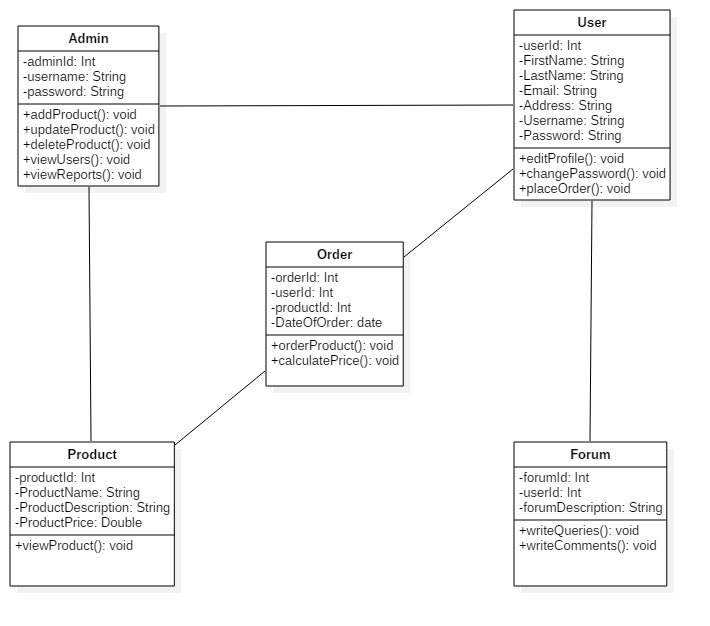


Figure : Initial Class Diagram

# Chapter 3: Design

Design, this is the third and important phase while designing any application or a software which is intended to accomplish certain goals. Simply, Design is the process of transforming user’s requirement into some suitable form, which helps the programmer in software coding and implementation. (Anon., 2019) Designing phase helps to give us the answer of “**HOW**” question. We user various tools and techniques even models to develop the logical solutions which is later implemented in the system.

Some of the importance of design while developing my project are listed below:

1. It helps to understand the requirement in better way.
2. It helps to increase reusability and prevent redundancy.
3. It helps to allow both admin and users to realize how the application is going to function.
4. It helps to create class and objects obtained during analysis, which will eventually help to make the work of programmer easy.

For doing this phase, I have used **Star UML**, a software which is used to make different class, object, use case diagram, component diagram, etc. **Visual Paradigm,** a UML CASE tool for making ER Diagram. **Belsamiq Mockup 3,** a user interface designing tool for developing Paper Prototypes.

## 3.1) Structural Modeling

Models that help to show the different component along with their relationship that are organized in a system is said to be structural modelling. Structural Modeling helps to show the static relationship of different components which are present in a system.

### 3.1.1) Final Class Diagram

Class is a blueprint of an object that defines what object can do. Class Diagram is a diagram that gives static view of a system. It helps describe the type of object and its relationship that exist among the system. By the help of this diagram almost all method can run with Object Oriented Models. (TutorialsPoint, 2017)

Purposes of class diagram are listed below:

1. Analysis and design of system in static view.
2. Helps to describe the responsibility of the system.
3. Acts as base component diagram and also helps in forward and reverse engineering.

### 3.1.2) Flow Chart

Flow chart is a formalized graphic representation of a diagram containing different symbols having its own information about steps or a sequence of works. Each of the works are linked with arrows to show the flow of the process.

Importance of creating flow chart for my project is listed below:

* Communication: It helps to explain the logic of a system to all the users, whether he/she is skilled or not.
* Proper Documentation: It serves as a good program documentation, which helps to make things more effective and efficient.
* Efficient Coding: Flow chart helps as a guide during the development of an application.
* Proper Debugging: As it helps in good communication for explaining the logic of a system, debugging process becomes fast.

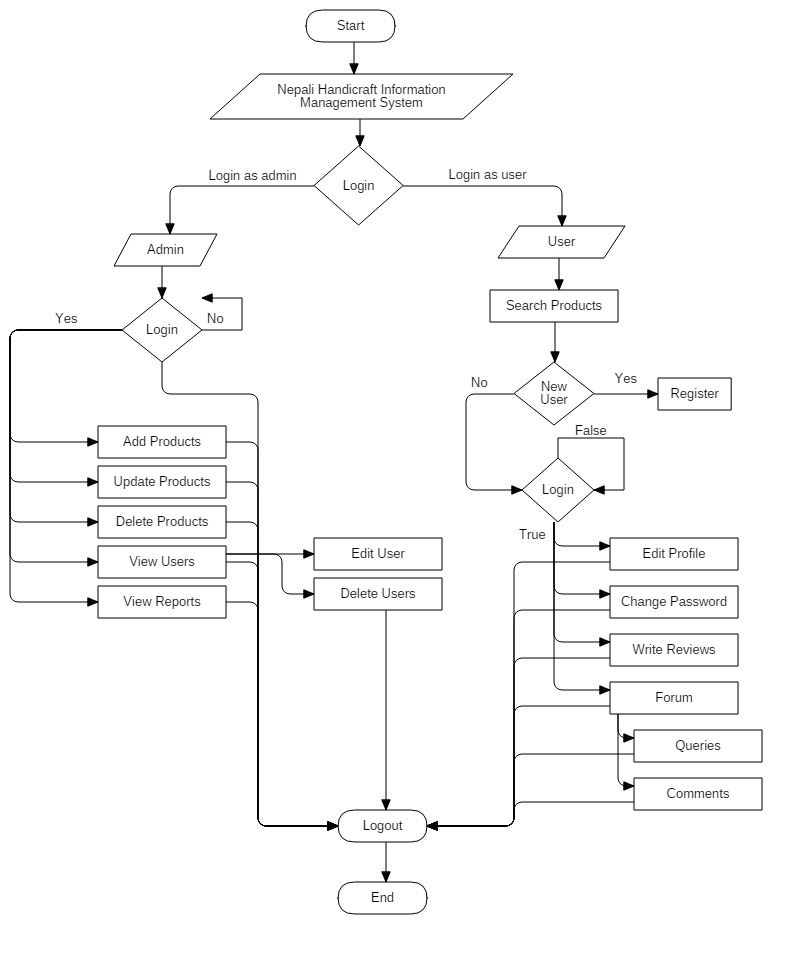


Figure : Flow Chart of a System

In above diagram, Admin can easily login to dashboard. Before going into the dashboard, Admin need to correctly type username and password. Once it is correct then admin is directed to the homepage else login is failed. After redirecting to the homepage of Admin, S/He can manipulate products, edit and delete any registered user from the database. For Users, one can easily search the products after or before login but cannot buy a product once user logs out. After logging into the system, user can edit their profile, change password, write a reviews and even ask any queries and comment in the forum section.

## 3.2) Behavioral Modeling

The type of modeling that models the dynamic behavior of the system to produce a specific behavior often designated through a use case is said to be behavioral modeling. It shows how objects engage with each other to provide a specific behavior.

### 3.2.1) Activity Diagram

Activity Diagram is a representation of the flow of activity from one to another, which helps to describe the dynamic aspects of the system.

Importance of activity diagram for my project is given below:

* It shows different steps involve in the UML use case.
* It helps to represent multiple conditionals activities easily.
* It can be easily understood by any people or the end users.

**Notations used for making Activity Diagram**

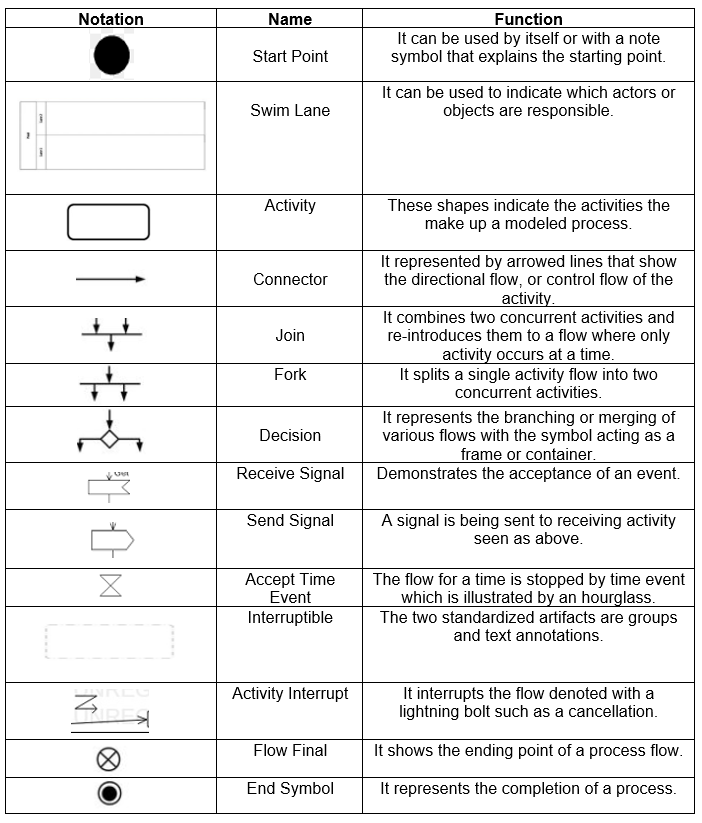


Figure : Notation used for Activity Diagram

**Login and Registration Activity Diagram**

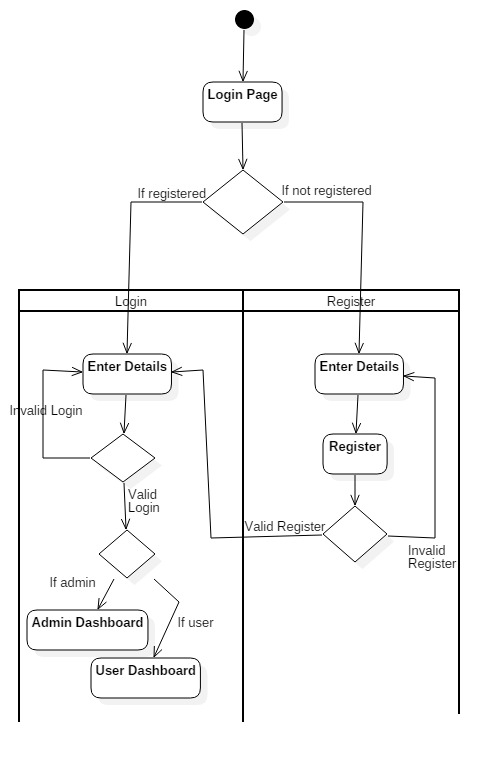


Figure : Login and Registration Activity Diagram

In above diagram, the activity of user starts from Login page. Then decision notation will help to take a decision whether a user is registered or not. If a user is registered, one will open login form else register form. For not registered user, Once register form is opened he/she has to enter their details in the form and proceed their process to register. After that decision is taken, if user provides valid information then user is successfully registered else warnings is shown. After registering, user is redirected to the login form and user will be asked to fill the login form. In case of incorrect email or password, error message is displayed or it will be redirected to the corresponding dashboard i.e., Admin Dashboard or User Dashboard.

**Activity Diagram of Admin**

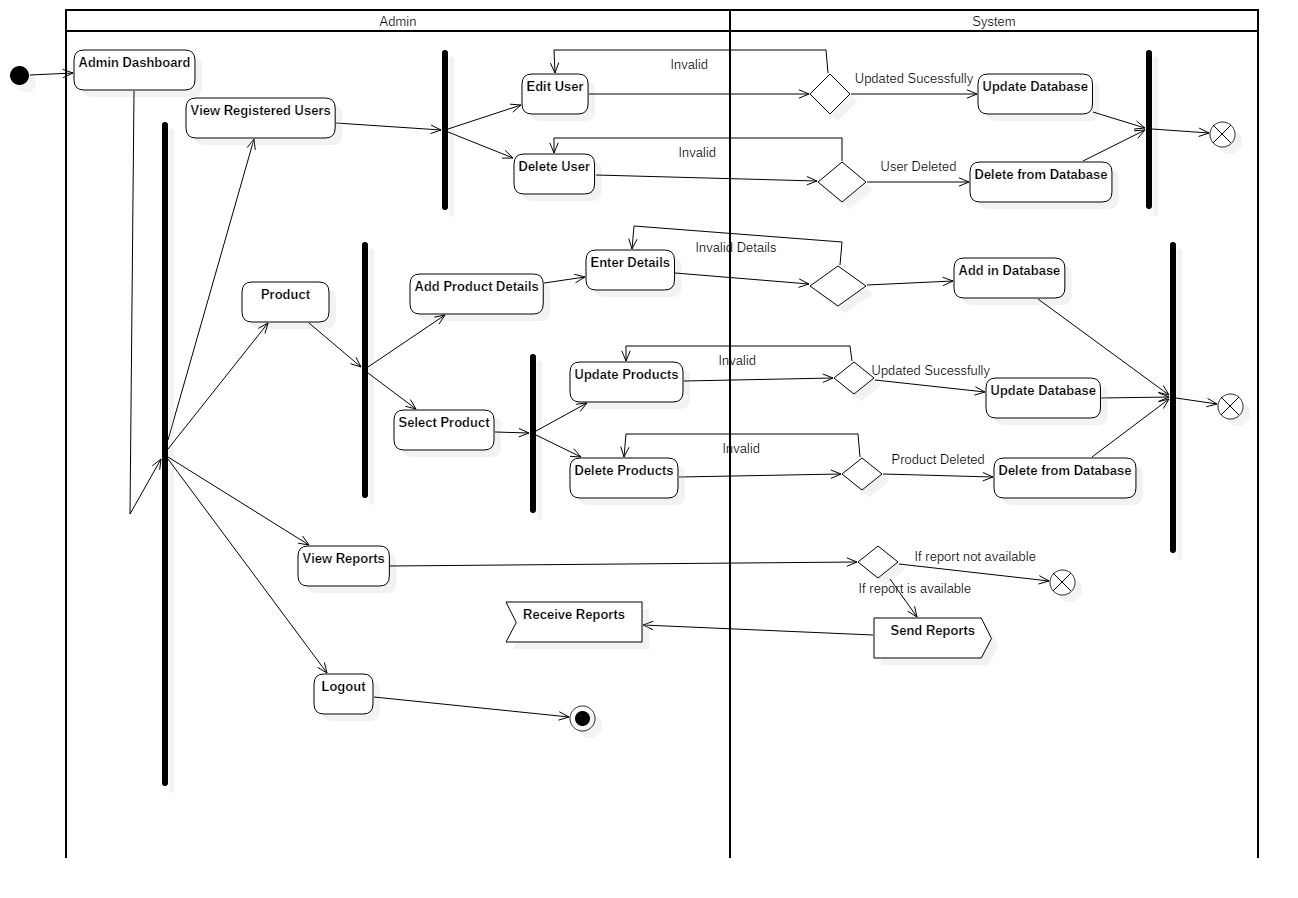


Figure : Activity Diagram of admin

Above given diagram explains all the activities that can be performed by Admin of the system. Once admin successfully logins to the system, admin can perform following tasks as shown by Fork Notation. That is, Admin can view registered users, Manipulate Products, View Reports and Get out of the system. Further, Admin can edit user and delete users in the database with the permission granted from the system. Admin can view reports in database with permission from the system, if reports are not available then process is ended else message is send from system to admin for receiving a message.

**Activity Diagram of User**

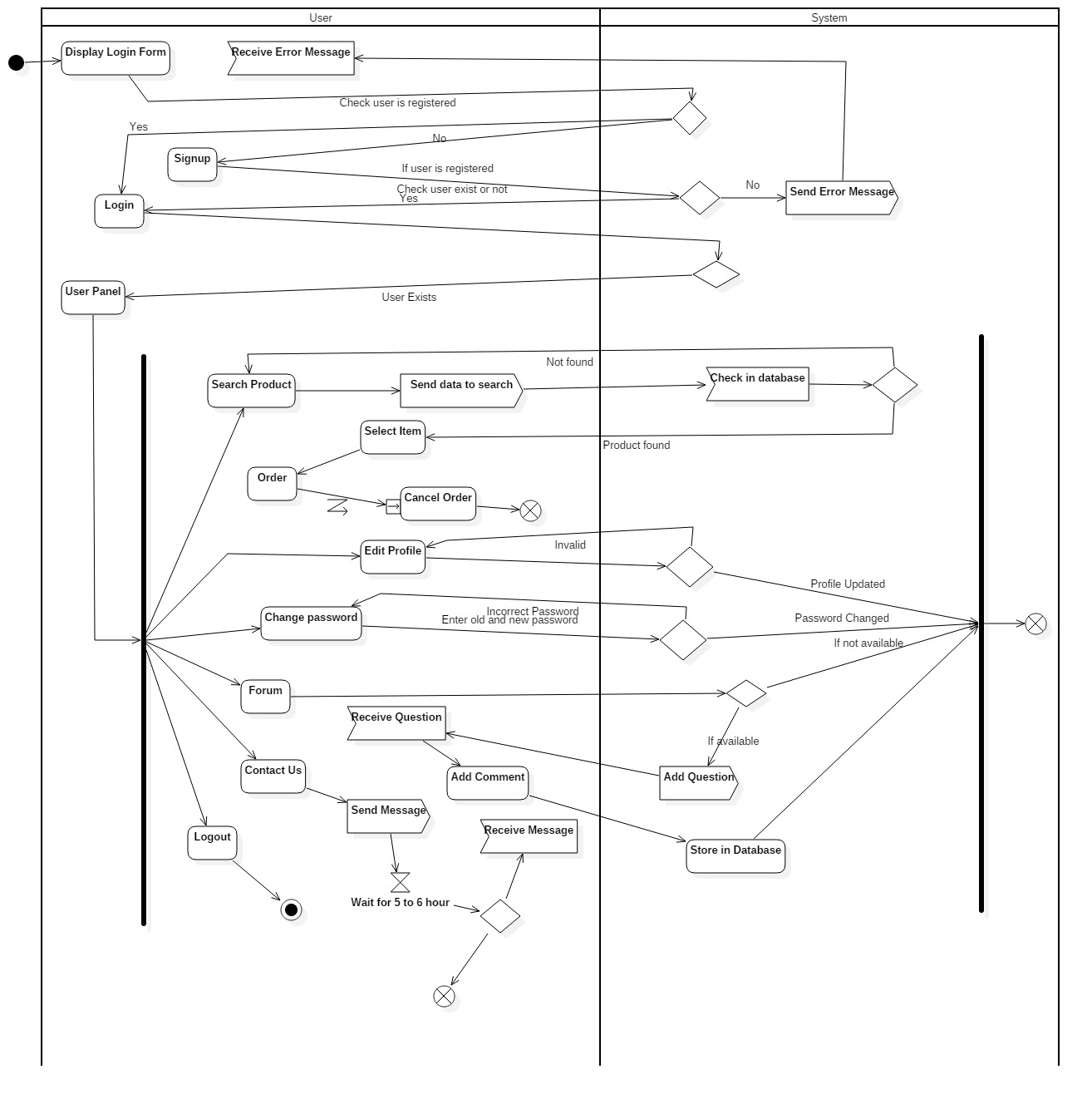


Figure : Activity Diagram of User

In the diagram above, user will be displayed with a login form, through decision notation whether users are registered or not it is checked. If user is registered, then login form is opened and if user is not registered error message is displayed from the system and that message is received by the user. Once user is successfully registered, user will be navigated to their panel. Users can do the following activities denoted by fork notation in the diagram. User can search product, after searching product user can select item they require and even cancel the order. User can change their password, edit profile, contact admin for any queries and get out of the system whenever they like. In forum section, user can ask a question about products and also reply as a comment if they want.

### 3.2.2) Sequence Diagram

Sequence diagram is a diagram that helps to represent the interaction between any objects in a sequential order. It helps to describe how and what order the objects in a system function. It shows the logical flow of the system is a good way.

Importance of this diagram for my project are listed below:

* Helps to understand the detailed function of the system.
* Helps to model and visualize the logic behind advance function or operation.

**Notations used for making Sequence Diagram**

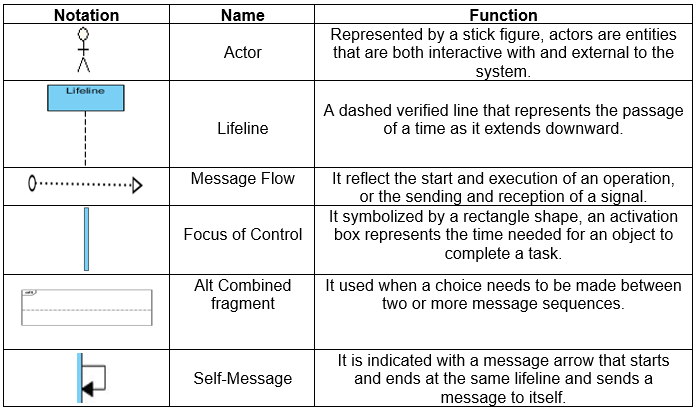


Figure : Notation used for making Sequence Diagram

**Registration Sequence Diagram**

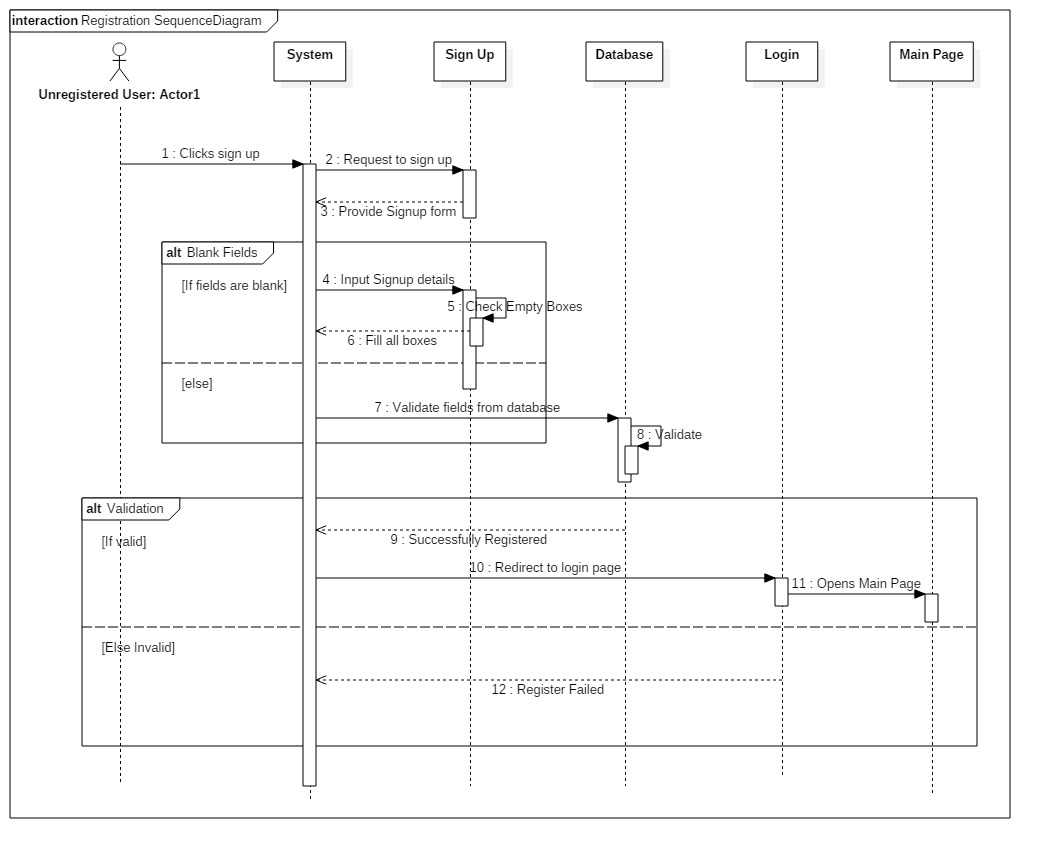


Figure : Registration Sequence Diagram

From the diagram above, at first user will click on sign up button, system will request for a signup form. Now user will be asked to input their details, during this process if user lefts any empty field user will see error message and if all the fields are filled properly satisfying the validation rules then user is created and stored into the database. After successfully signup user is now redirected to the login page for further actions.

**Login Sequence Diagram**

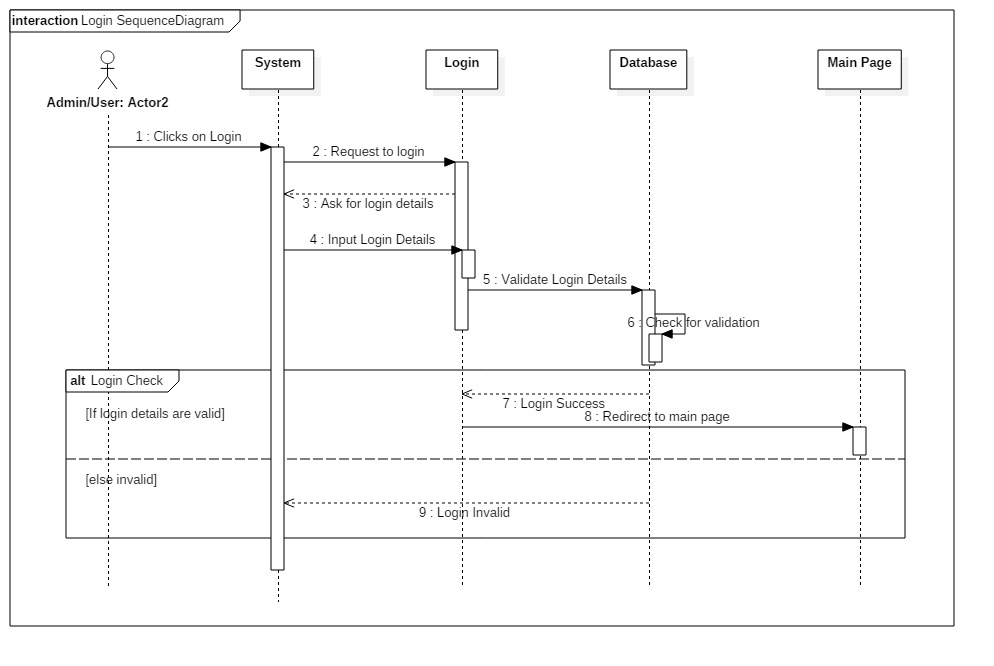


Figure : Login Sequence Diagram

After successfully registration of a user, user can now login. For that user can click on Login and system will request for a login form and then user will enter their required credentials. After entering the credentials, is validate from database. If the required credentials matches, user is redirected to main page or user dashboard else invalid login message is displayed.

**Admin Privilege to user**

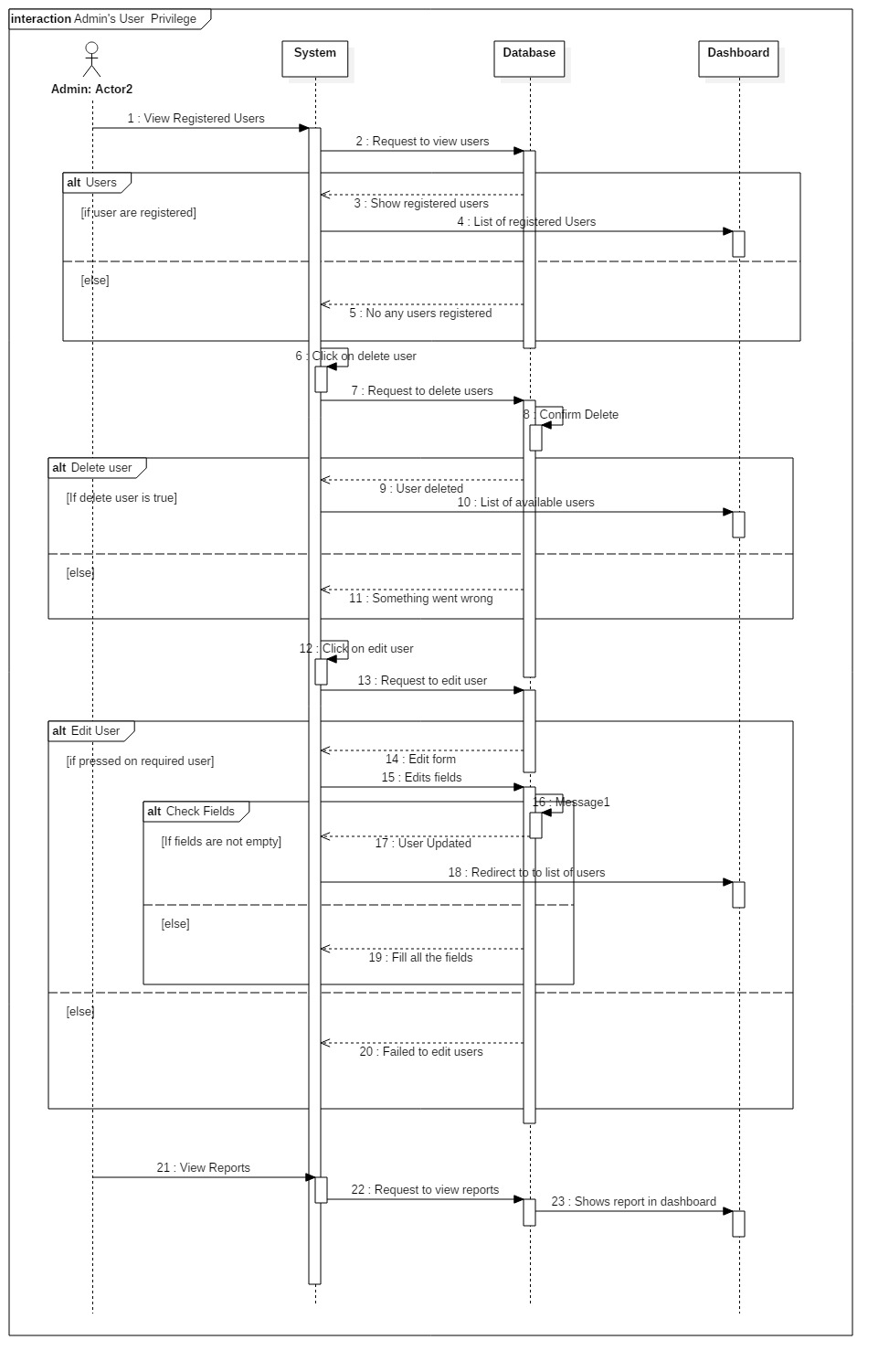


Figure : Admin Privilege for user

**Sequence Diagram of Products**

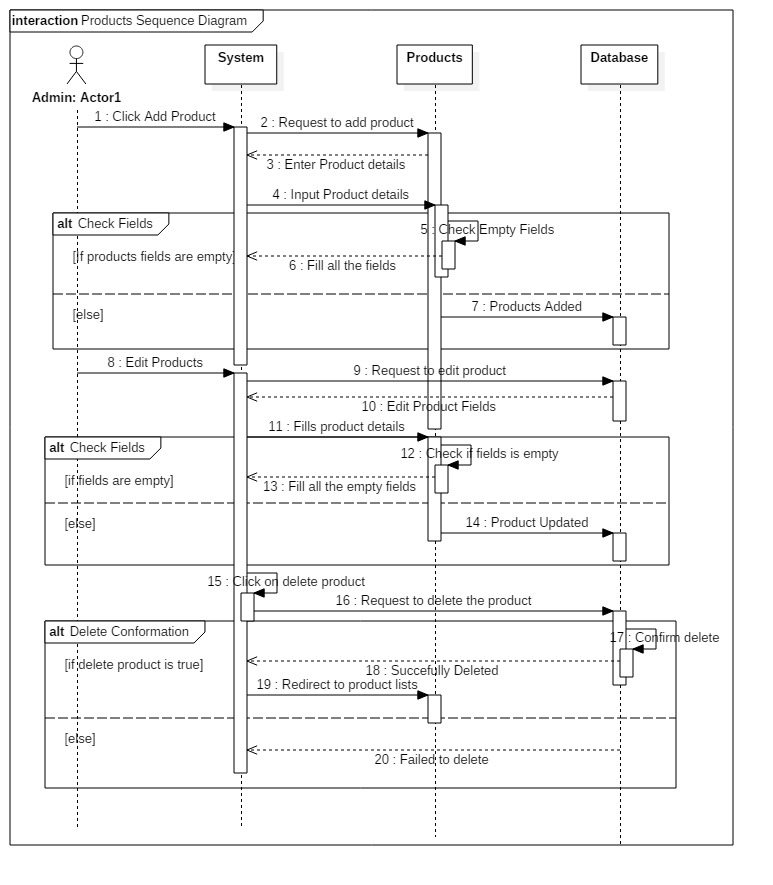


Figure : Sequence Diagram of Products

From Figure 9 and 10, These two diagram represent all the actions or the functions which is performed by admin. Admin request to view registered users from the system, system request from database if user are registered in the database then lists of all the registered users are seen else no any users are seen. If admin wish to delete any registered user from the system then, S/He will click on delete user, after clicking system will request its permission and if permissions and conditions are okay then selected user is deleted from the database else error message is displayed.

Admin willing to add product, can click on add product and system requests for a product to add. Then admin will receive an add product form, after filling all the products details empty fields are checked. If found admin will be displayed error else product is successfully added in the database. Also for deleting product admin has to select the required product from the product list and if the confirmation goes as planned then product is deleted else error message is displayed to the system. For updating product, system request to select product from database, product edit form is opened for admin to make changes to the product. If any fields are left empty, error message is displayed else product is successfully updated and stored into the database.

**User Sequence Diagram**

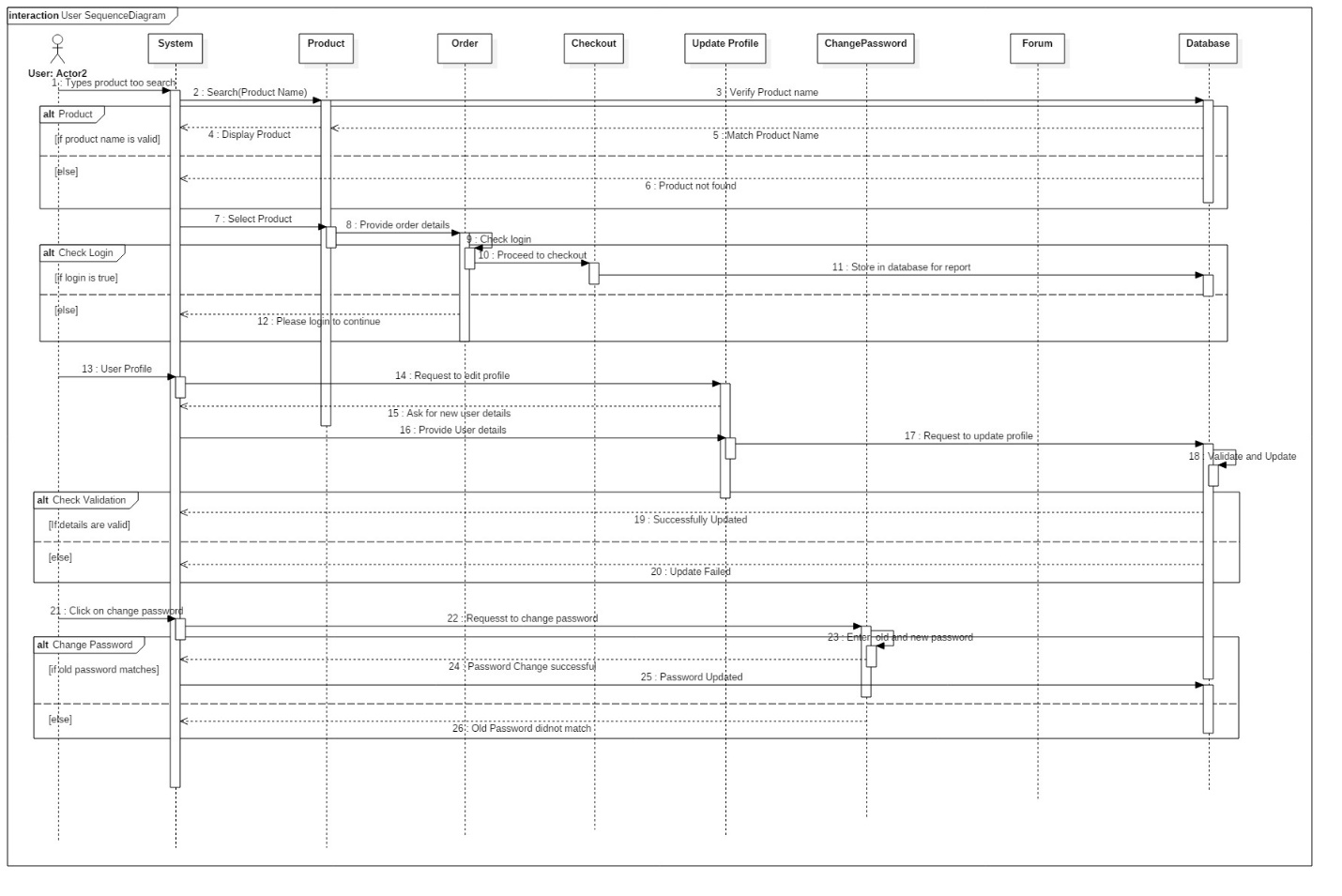


Figure : User Sequence Diagram

From the diagram above, User can easily search product, system requests the typed keywords of user from the database. If the keyword matches the product which is in database, then it is displayed else not. If displayed is selected user login is checked. User can only proceed to checkout only if user has logged in to the system else user should first login to the system and only continue the process. For editing the user’s profile, one need to login into the system and system will request to edit profile. Then update profile form is displayed from the system, user need to enter the boxes without leaving any fields. If user fails to fill all the boxes, error message is displayed else profile is successfully updated. User can change their password too, for doing that user need to click on change password. After doing that system will request to change a password, system is replied with change password form having current password, new password and re-password in it. Here, only if old password matches then password is successfully changed or the error message is displayed.

## 3.3) Database Modeling

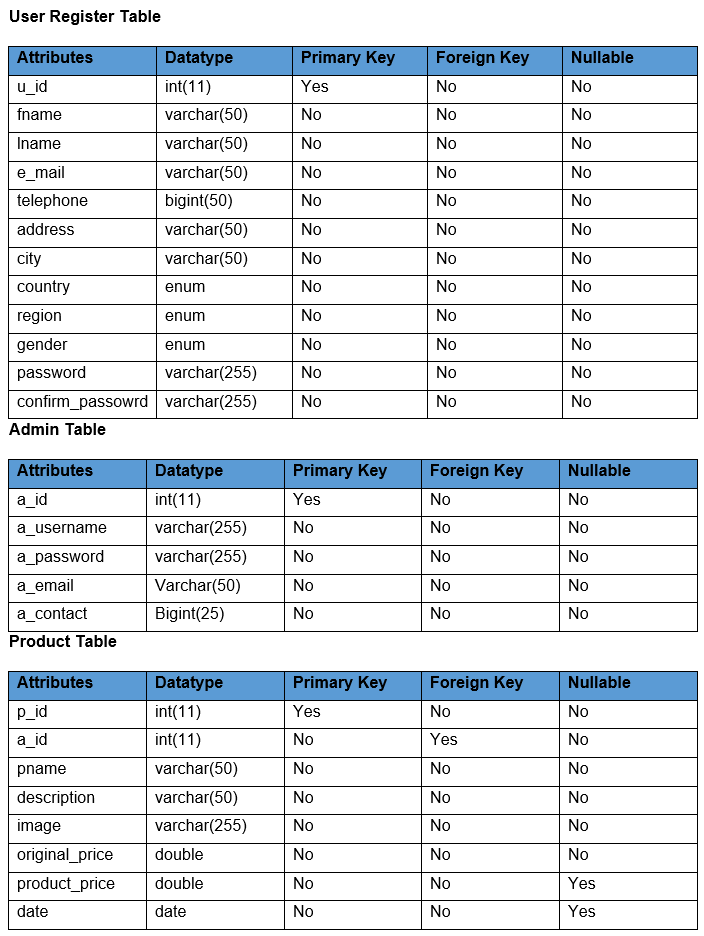
It is the type of data model that helps to determine all the necessary and important logical structure of a database. It fundamentally determines in which way a data should be stored, organized and manipulated.

### 3.3.1) Data Dictionary

Simply, Data Dictionary means data of a data. It is the collection of information describing about the contents, format and structure of a database.

Importance of Data Dictionary in my project are listed below:

* It holds ownership of a data.
* It provides detail information about data elements.
* It helps to figure out the basic data structures required for the system.
* It helps to prevent data duplication; each entry field is identified for faster performance.



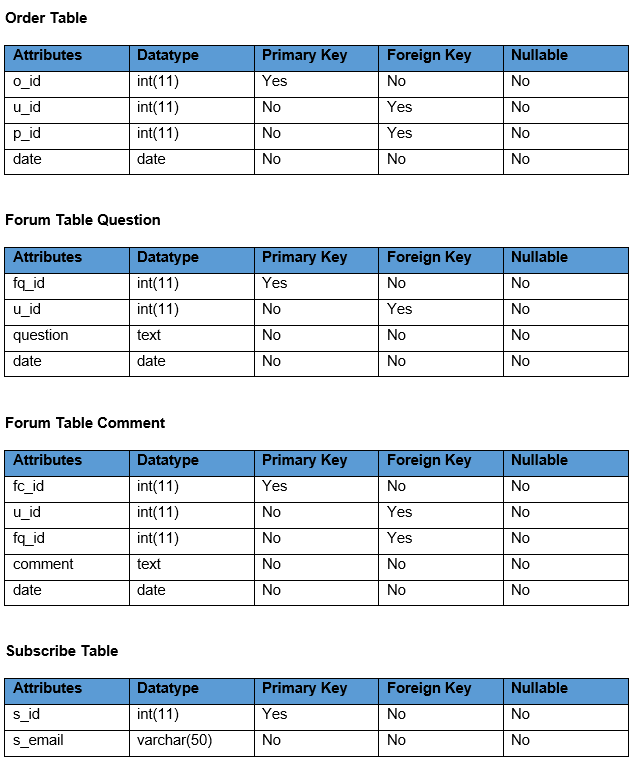


Figure : Data Dictionary

### 3.3.2) ER Diagram

Entity Relationship Diagram (ERD) is a data modeling technique that graphically represents a system’s entities and relationship. It is a backbone for developing a database design.

Importance of making ERD in my project are listed below:

* Base for designing a database design.
* Database will be well documented.
* Helps to know the relationship including their cardinality.



Figure : ER Diagram of a System

## 3.4) Architectural Design

**Client-Server Architecture:**

It is an architecture of a computer network, in which the client requests a service from a centralized server and the centralized server gives service to a remote processor or a client-server. Advantages of client server architecture are:

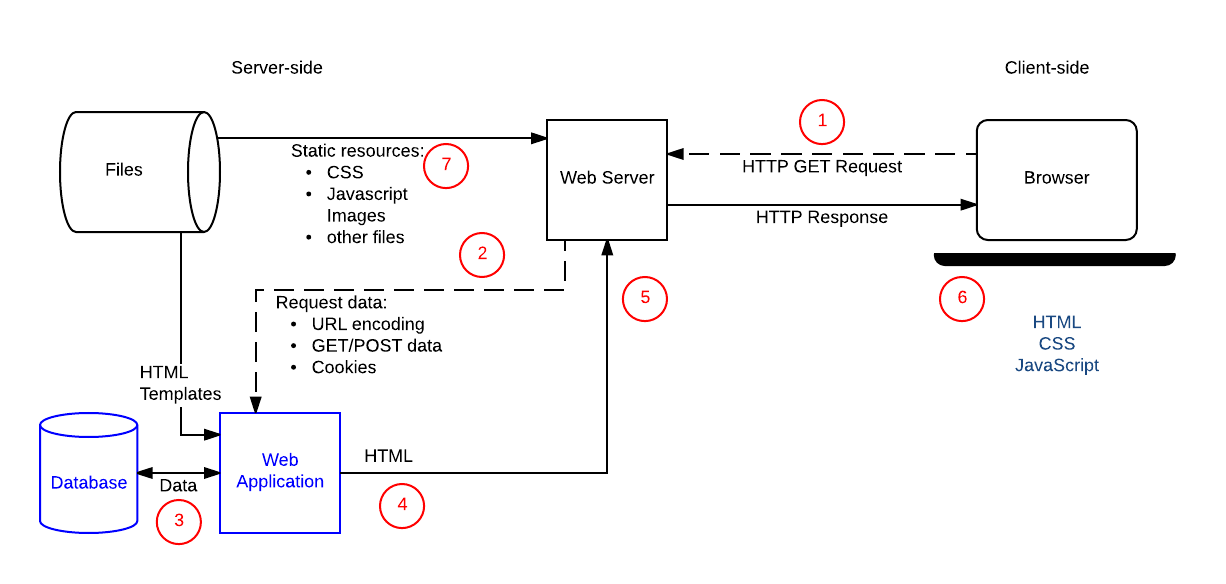
* It helps to share resources among client and servers easily.
* It reduces data of client by storing data on server instead.

Figure : Client Server Architecture

Comparing above figure with my current project, first of all user will send HTTP GET request to the server side through web server. Travelling through web server it requests URL encoding, GET/POST data or cookies from web application and that web application will ask data from database. If it is image files or other files it will be pulled from another part of server side, which is file in above diagram. After requesting files from database by web application, our requested data is transferred to web server again and Web Server will send the HTTP Response back to the browser or client. And Client will be able to see its requested response in a fraction of a time.

## 3.5) Paper Prototyping

Paper Prototyping is one type of prototyping method in which paper models are used to test concept of system. In this project I am using Mockup Prototyping, which is useful for visually conception of the system. Following are the mockups of my project.

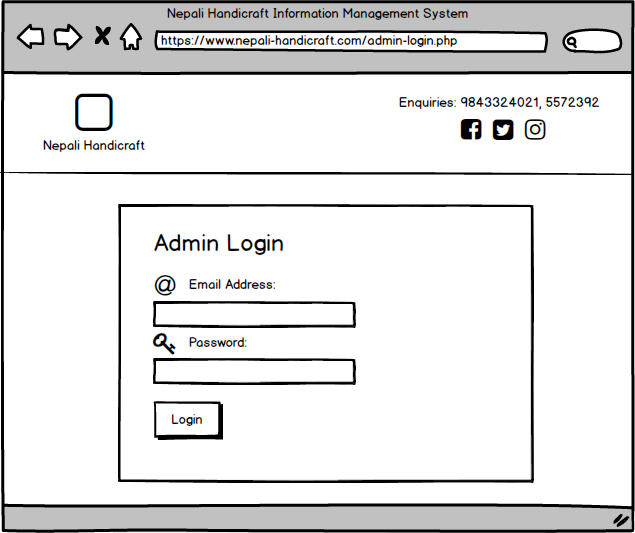


Figure : Admin Login

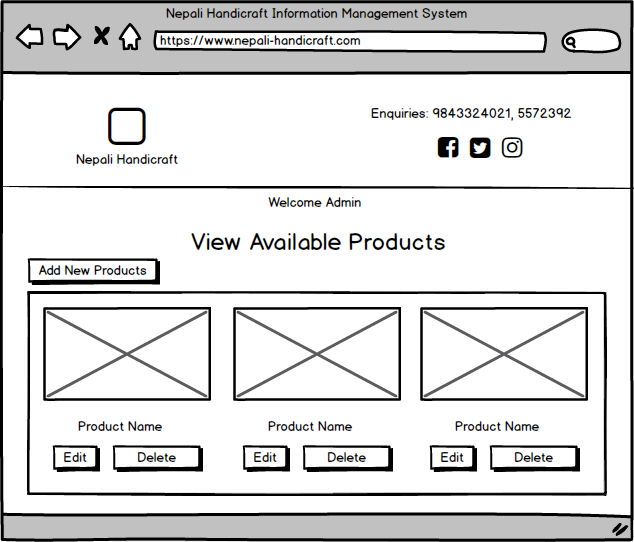


Figure : View Available Products

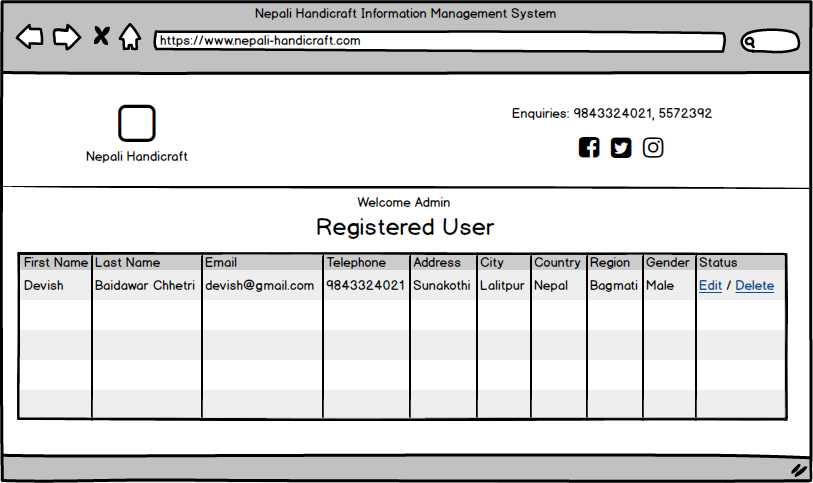


Figure : View Registered Users

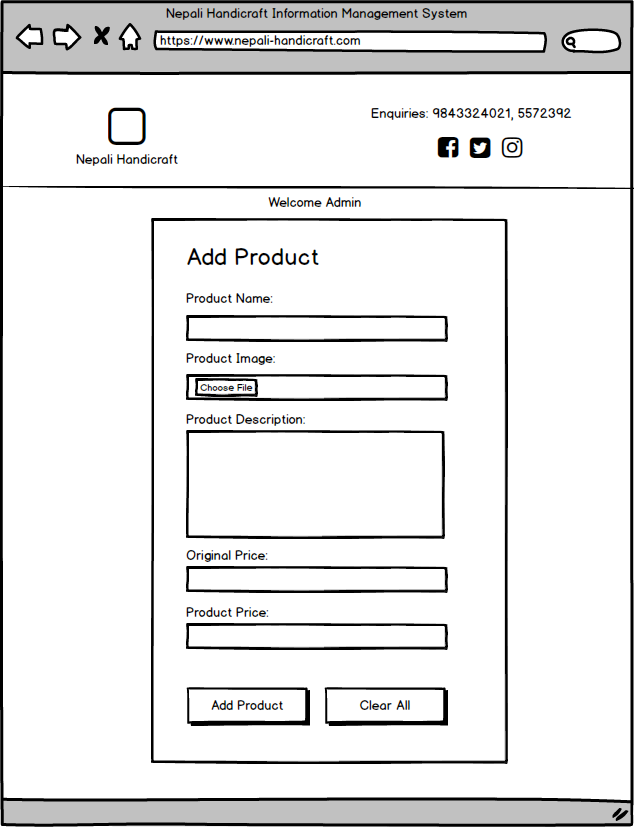


Figure : Add Product

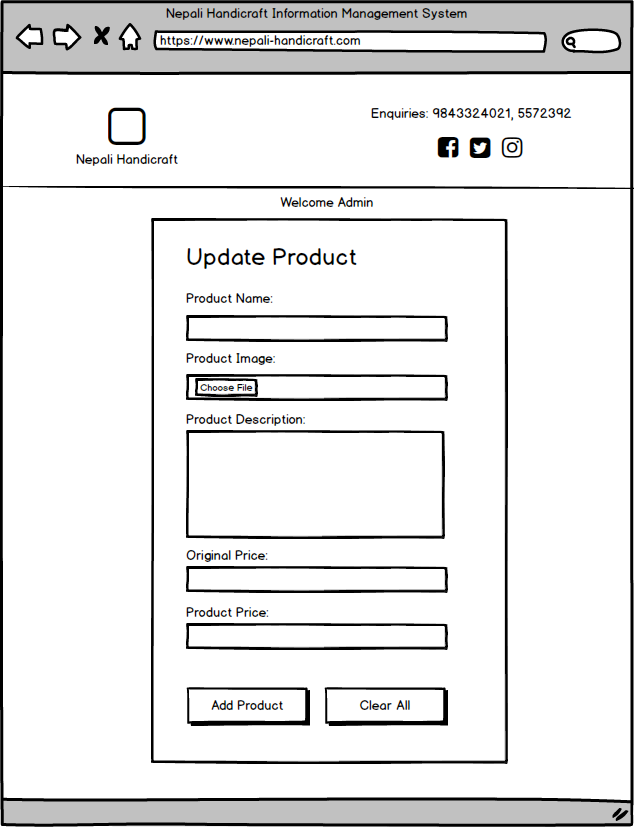


Figure : Update Product



Figure : Admin edit user

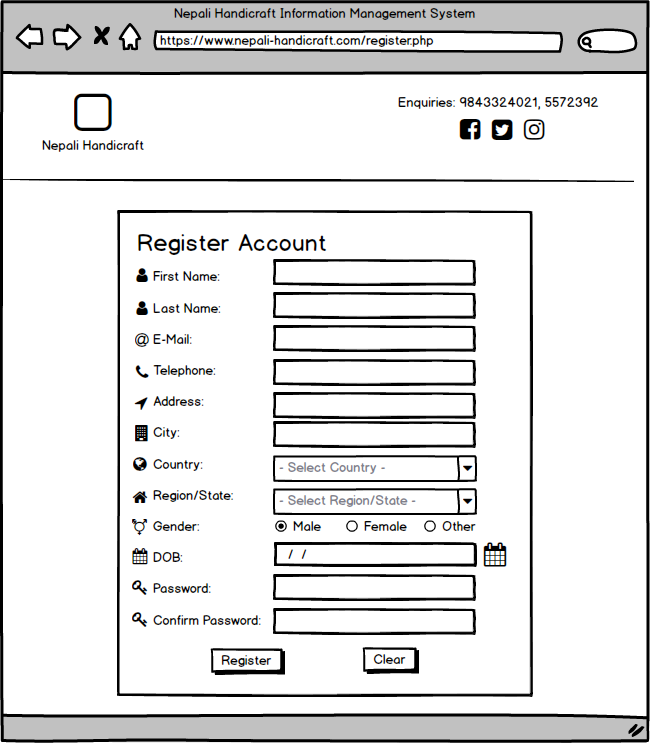


Figure : Register New user



Figure : Update User’s Details

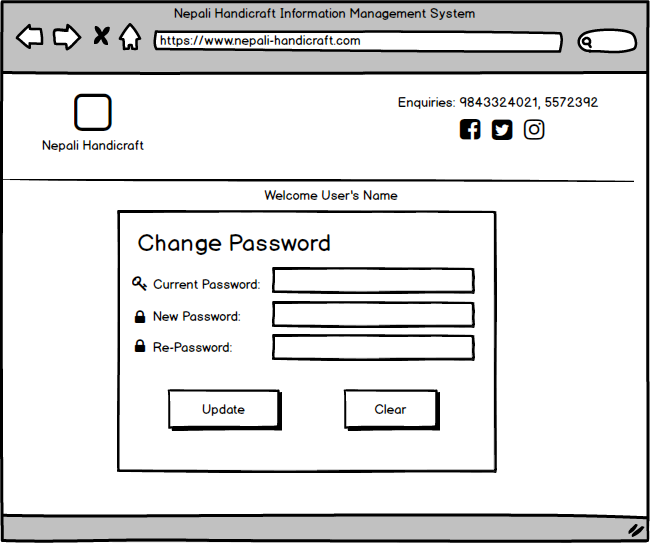


Figure : Change Password



Figure : Forum Ask Question

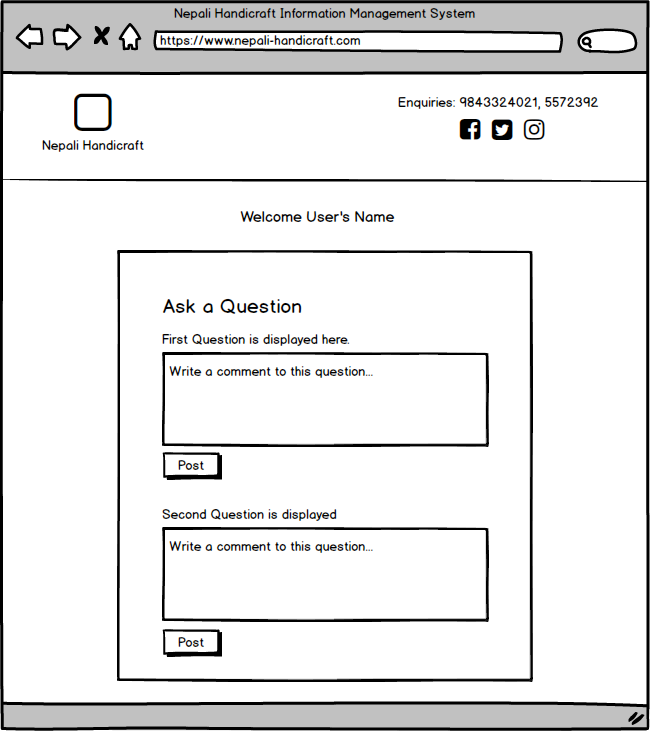


Figure : Forum Comment

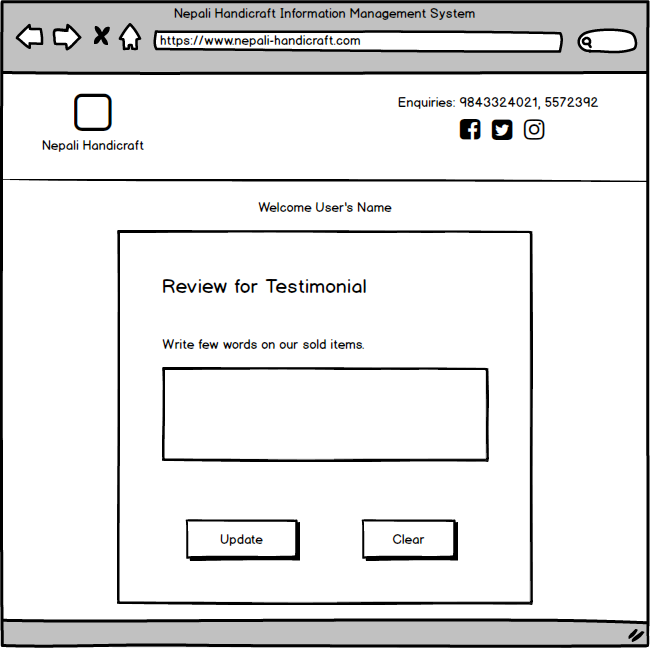


Figure : Review for testimonial

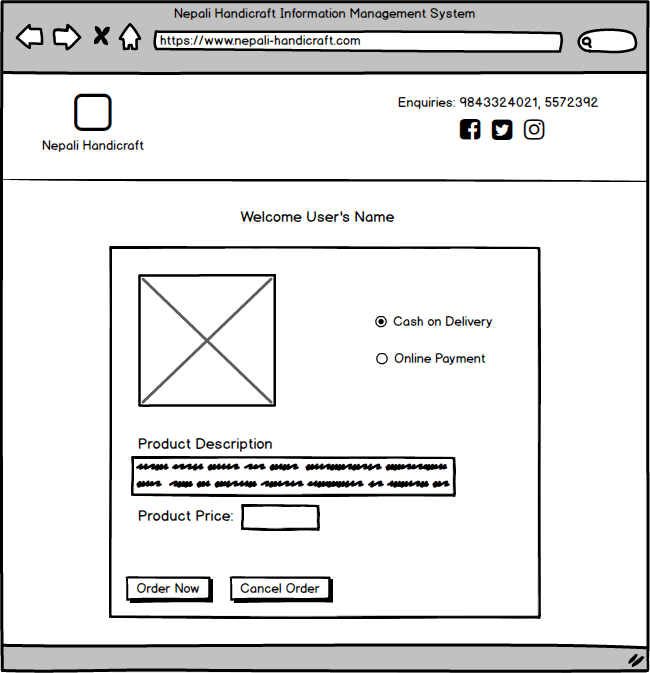


Figure : Order Required Product

# Chapter 4: Implementation and Coding

## 4.1 Introduction

Transforming a design of a system into a code in any high level programing language can be said as coding. Coding gives a uniform appearance to the code written by different developers in different programing languages. After designing phase, coding and implementation phase comes under third phase of System Development Life Cycle.

## 4.2 Programming Language

In this project, I have chosen Hypertext Markup Language (HTML) and Cascading Style Sheet (CSS) for designing the front end. Object Oriented Programming (OOP) PHP: Hypertext Preprocessor (PHP) for working on backend and MySQL for storing data and values into database.

## 4.3 Tools Used

Following tools are used to develop my project.

* **Windows 10** as an operating system
* **Visual Studio Code** as a code editor (Open source code editor application)
* Google’s **Chrome** and **Brave** as browsers
* **XAMPP** as a server and database for my machine

## 4.4 Code and Screenshots

Following are the screenshots and code of the respective frontend and backend:

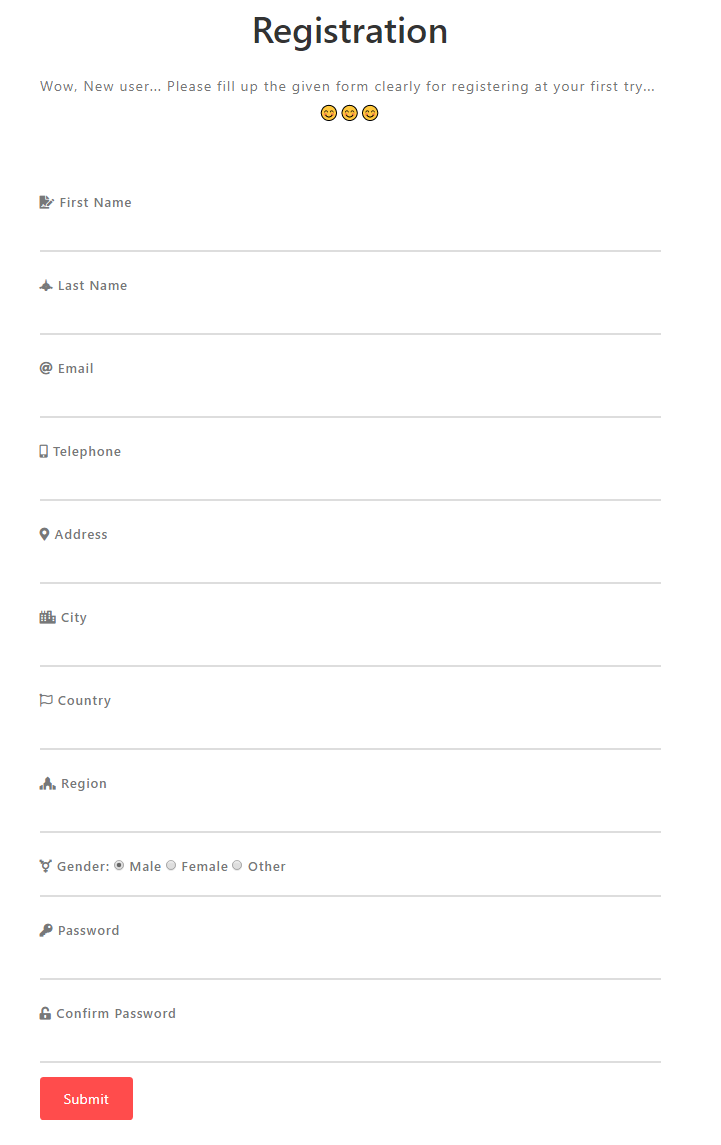


Figure : User Registration Form



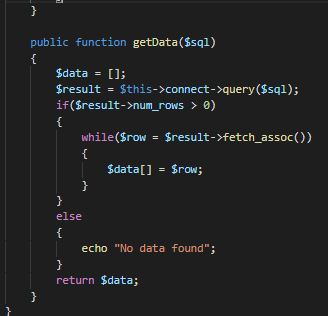
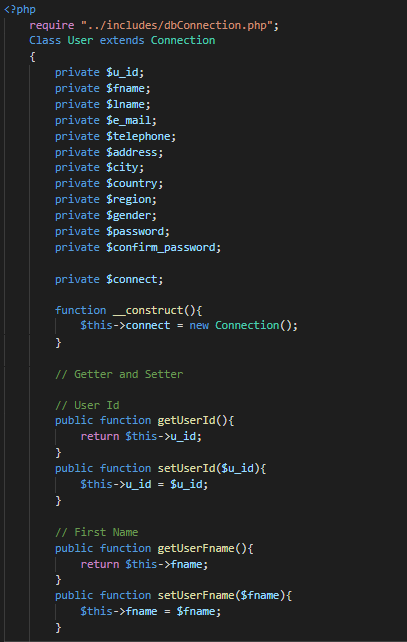


Figure : Database Connection



Figure : User registration Action



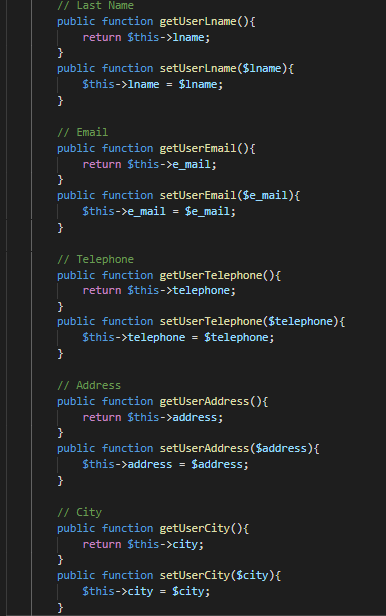






Figure :Code for User registration

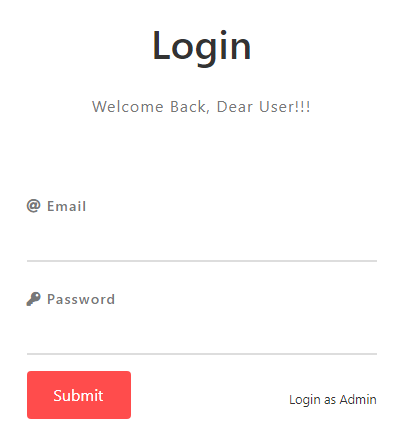


Figure : User Login Form

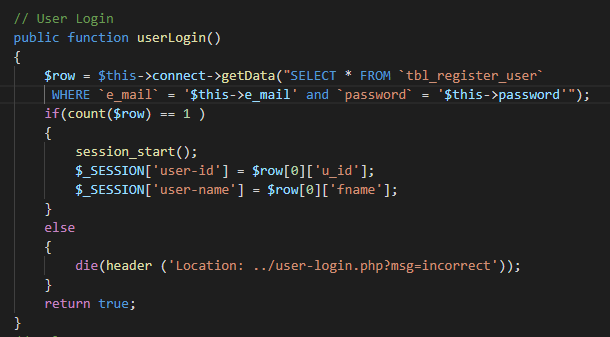


Figure : User Login Code

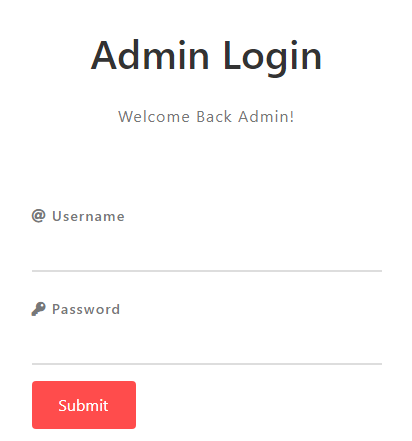
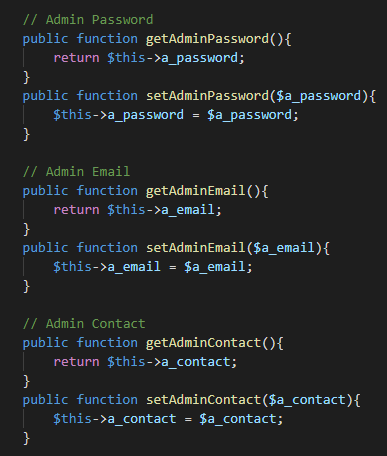


Figure : Admin Login Form





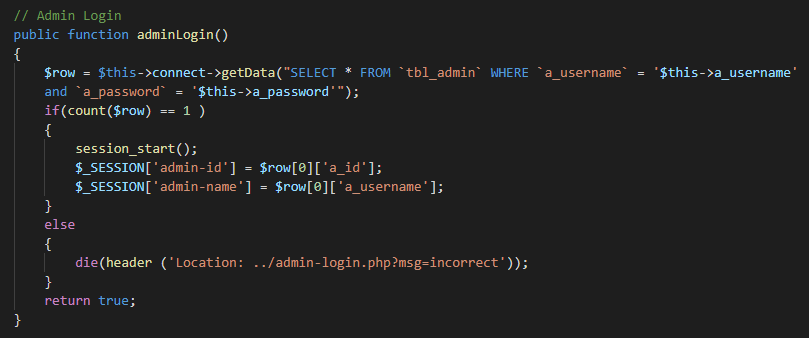


Figure :Admin Login Code

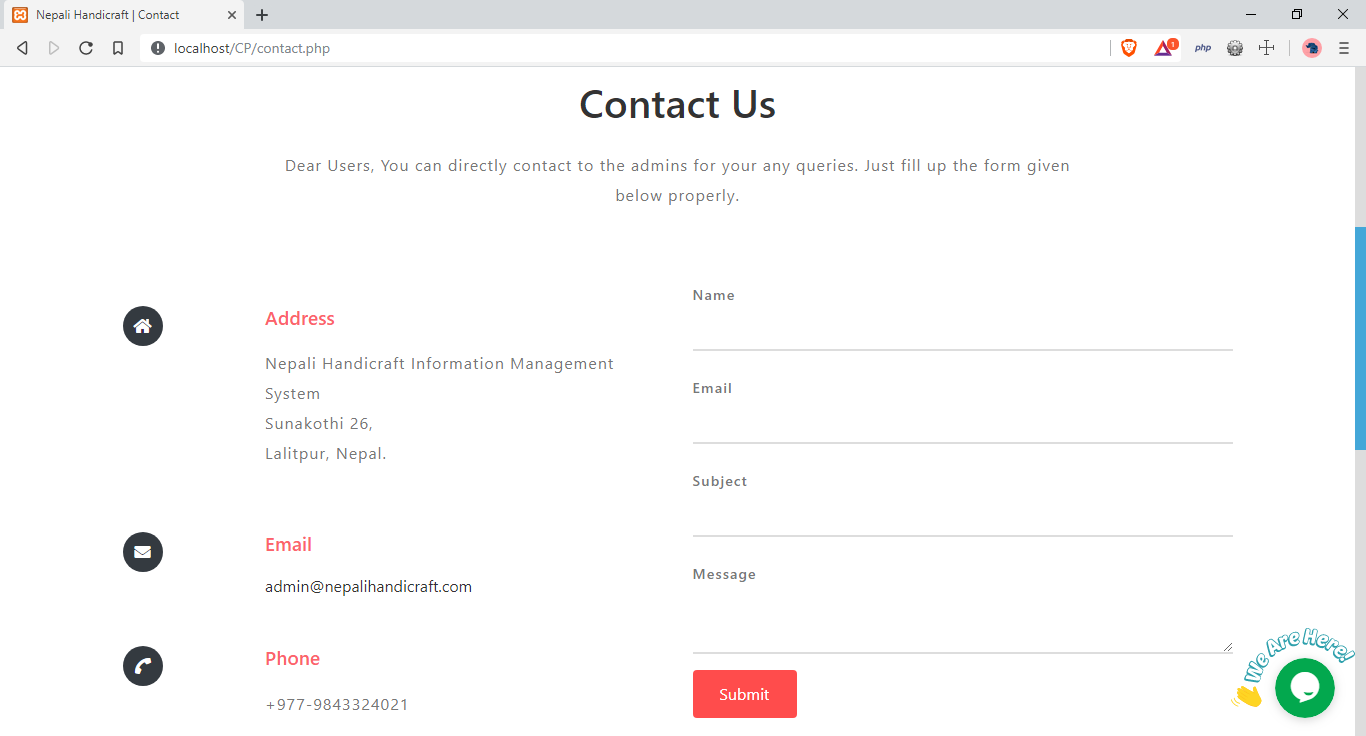


Figure :Contact Form

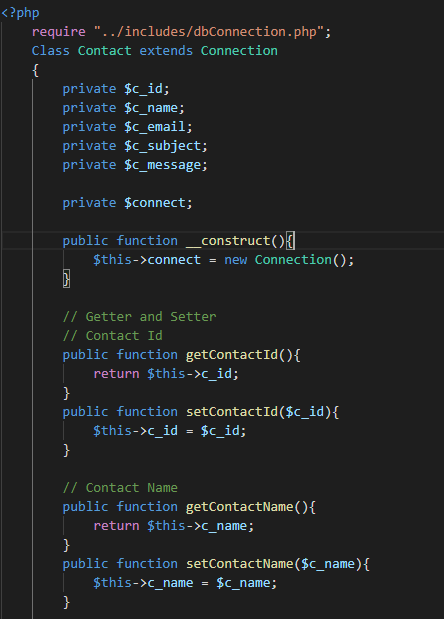




Figure : Contact Form Code

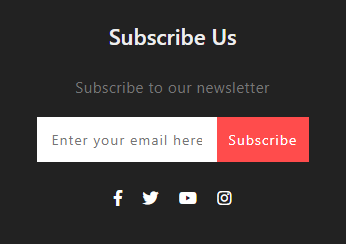


Figure : Subscribe Form

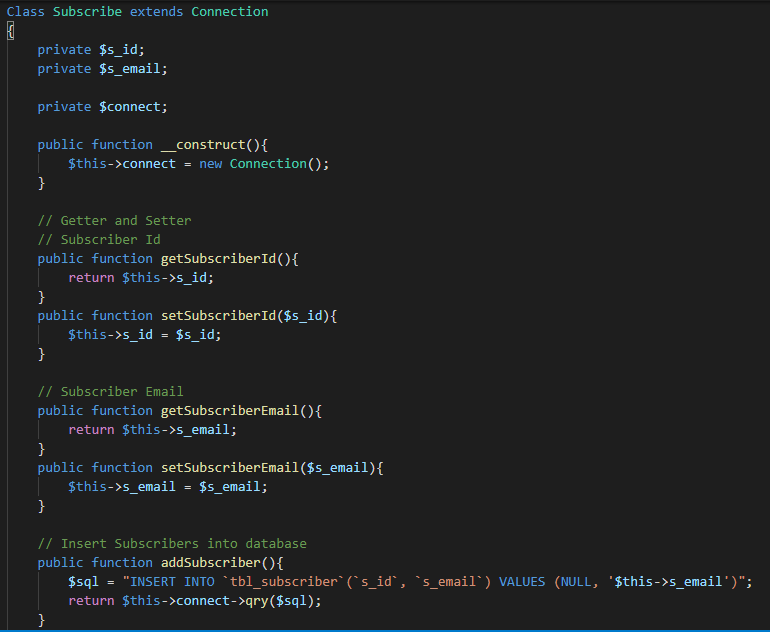


Figure : Subscribe Code

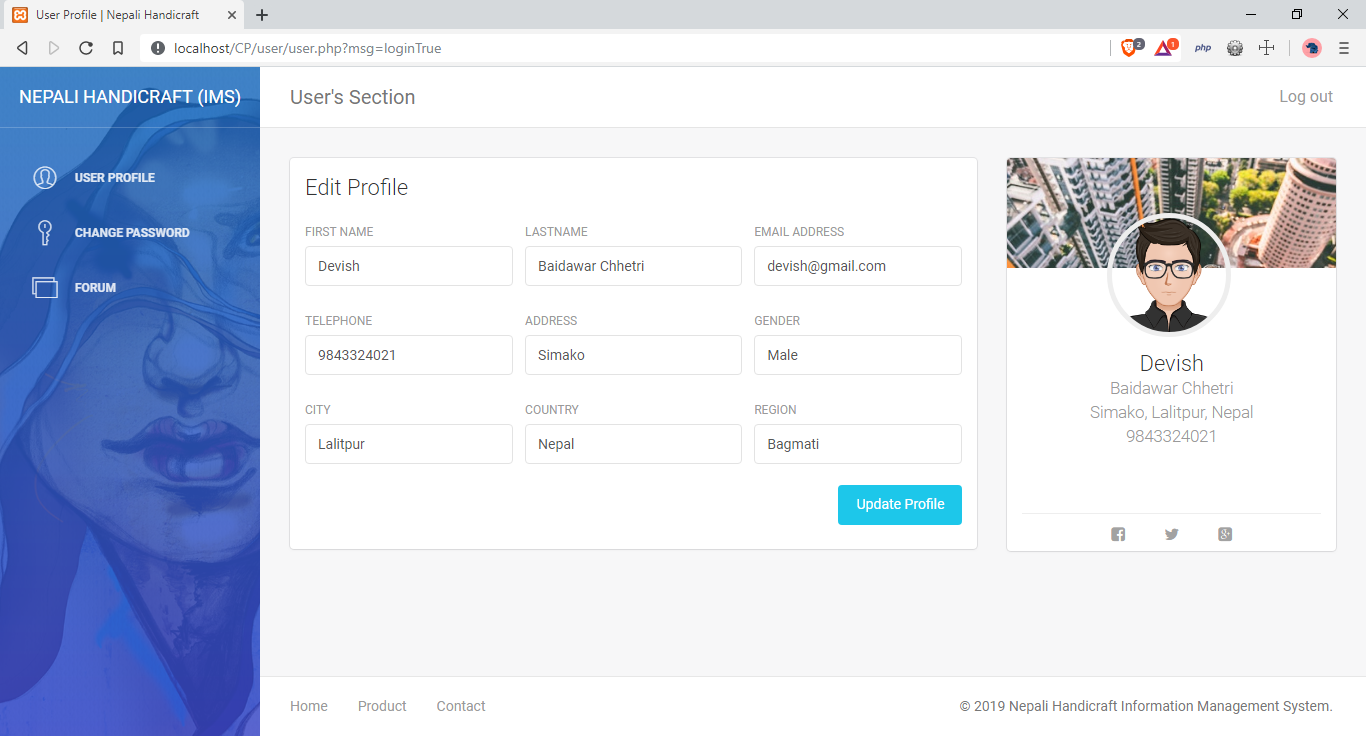


Figure : User dashboard and User Update Form







Figure : User Update Code

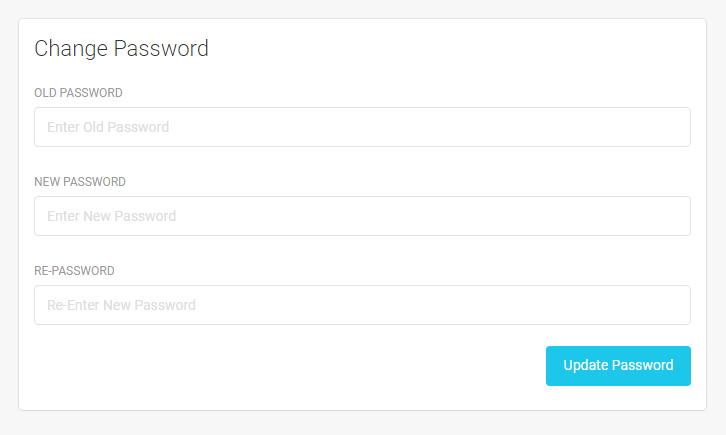


Figure : User Password Change Form



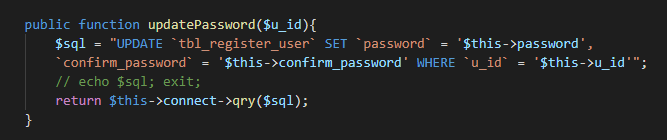


Figure : User Password Change Code

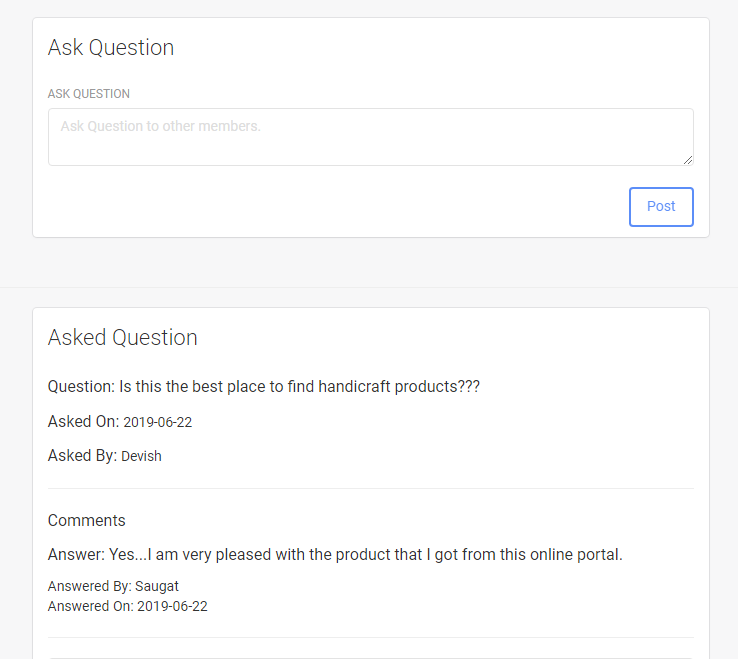


Figure : Users Forum Form



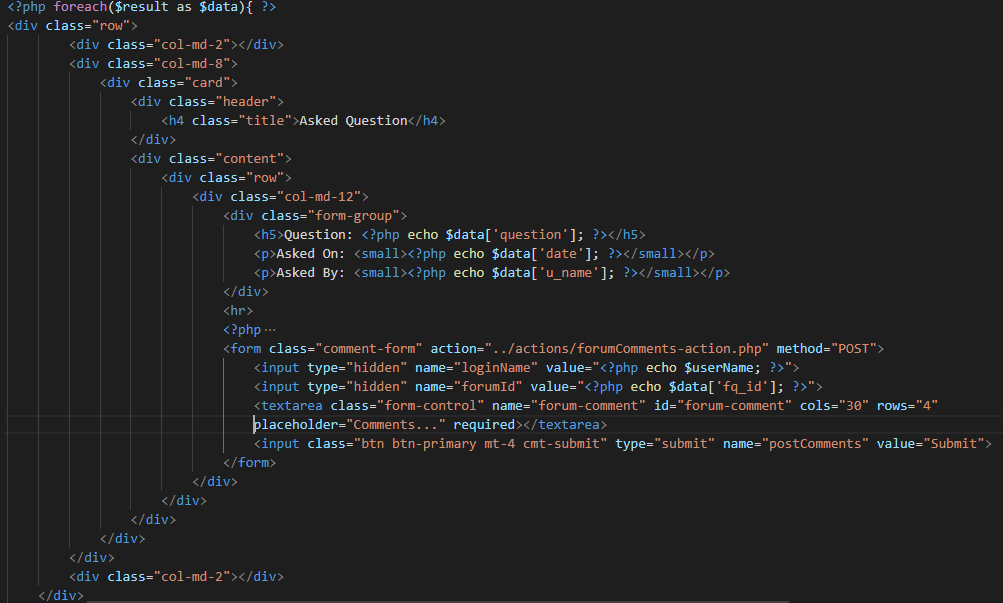
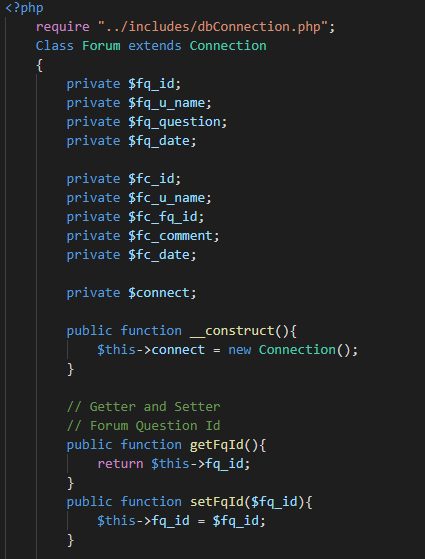
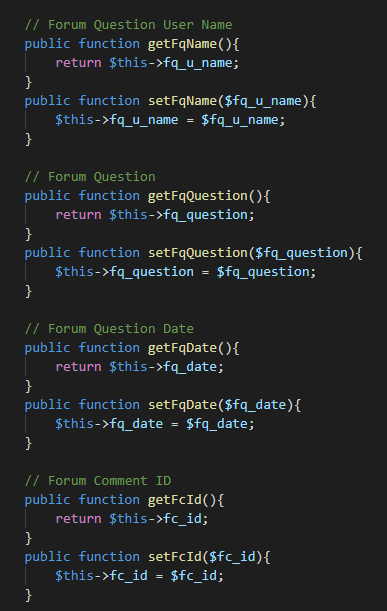


Figure : Forum Question code and display forum question and comments





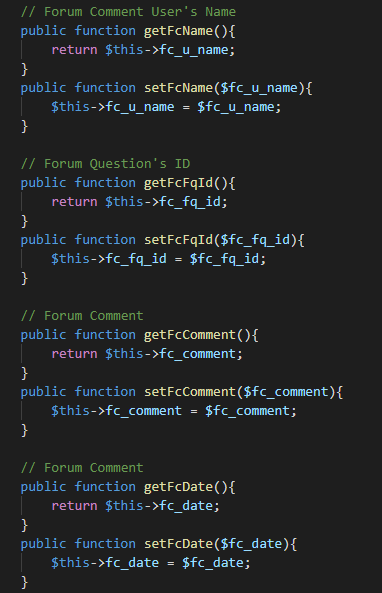




Figure : Forum Post and See Question and Comment Code



Figure : Forum Question Code



Figure : Forum Comment code

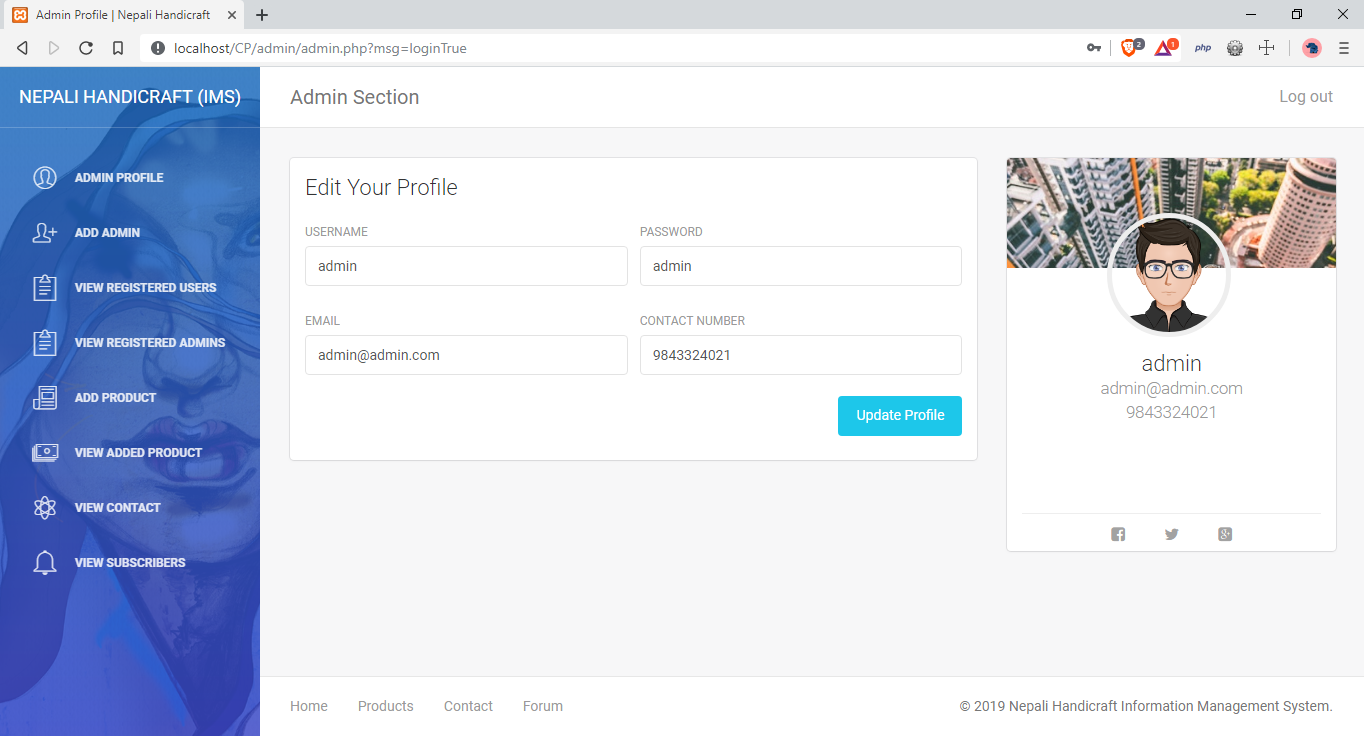


Figure : Admin Dashboard



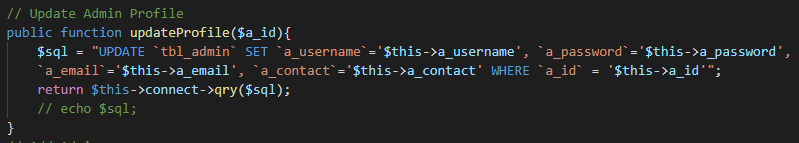


Figure : Update Admin Profile Code

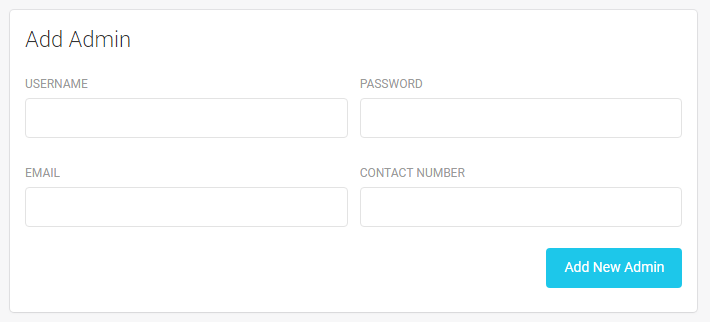


Figure : Add New Admin form



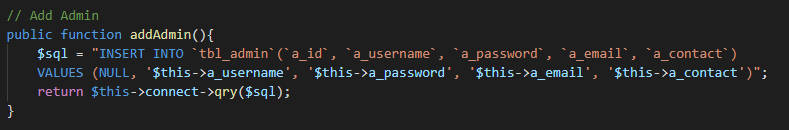


Figure : Add Admin Code

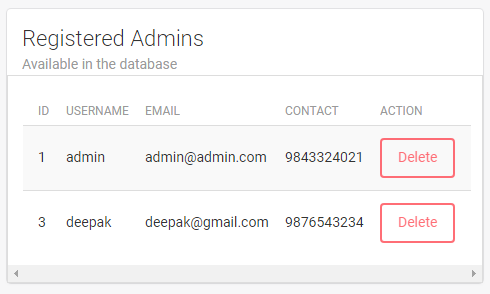
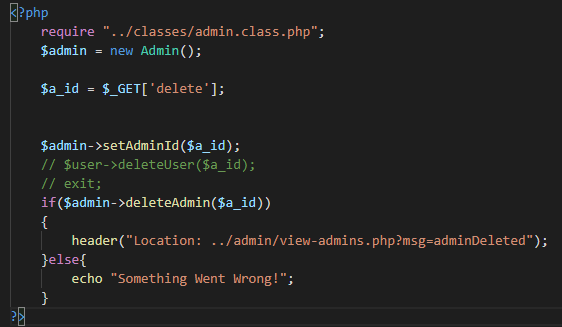


Figure : Delete Admin Form



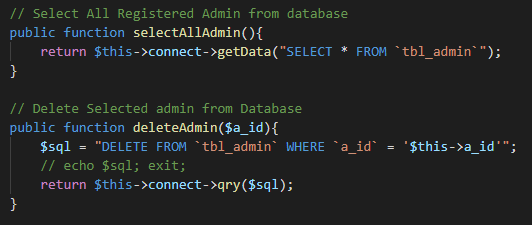


Figure : Select and Delete admin code

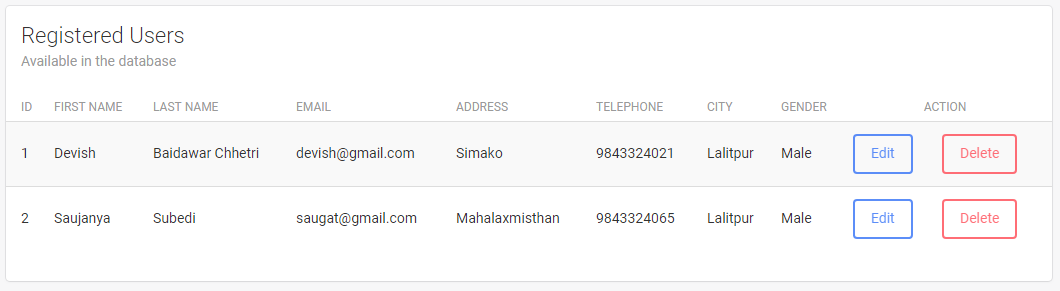


Figure : Select registered users

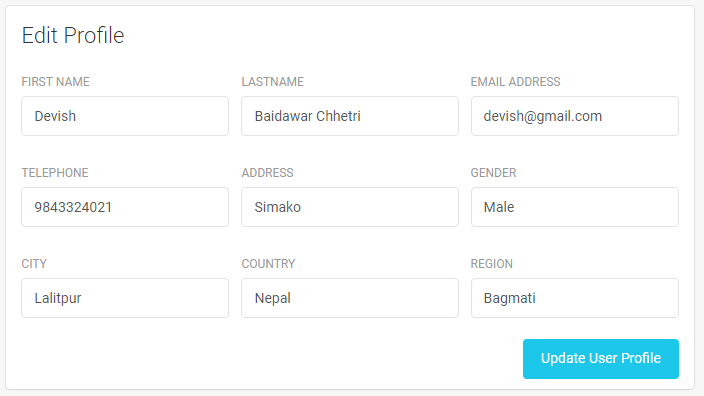
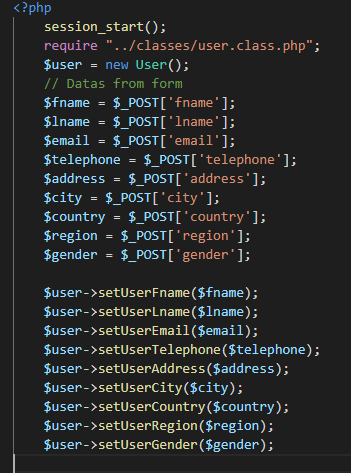


Figure : Edit selected user by admin



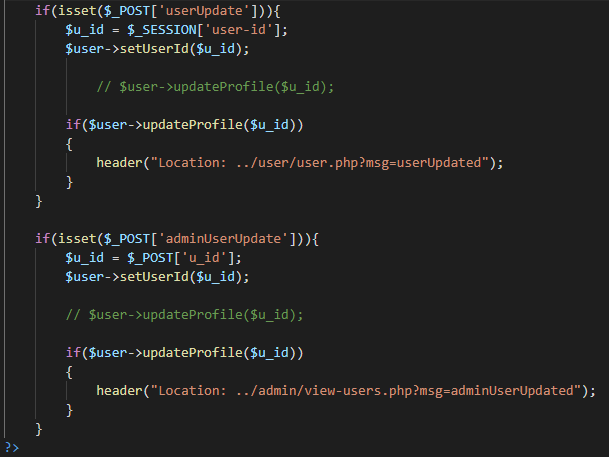




Figure : Update and Delete selected user

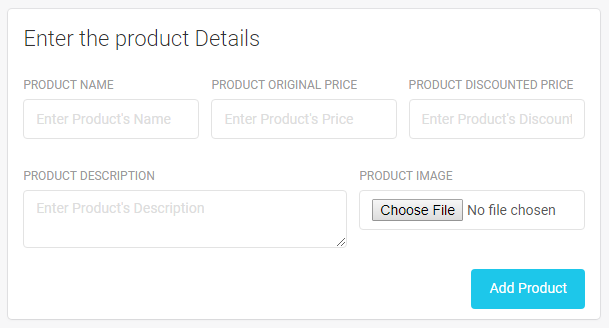
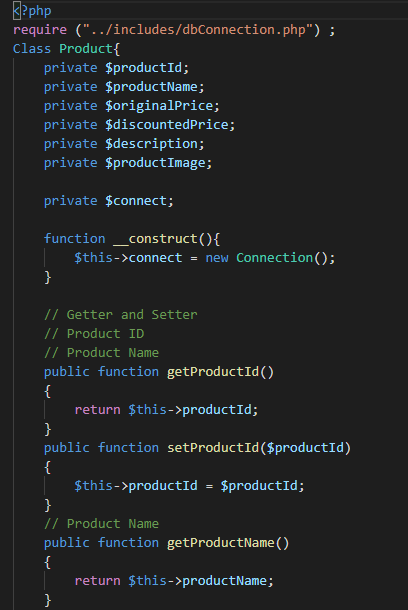
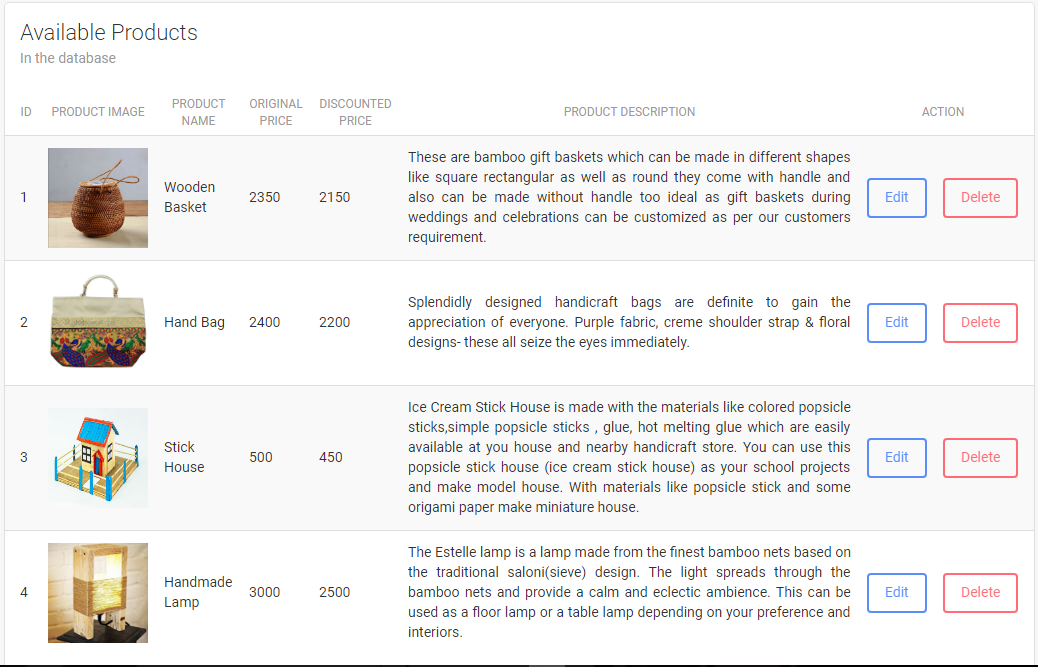
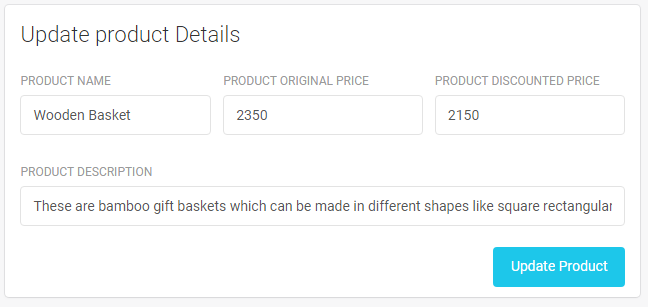


Figure : Add Product Form







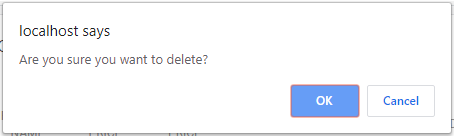


Figure : Add Update and Delete Product

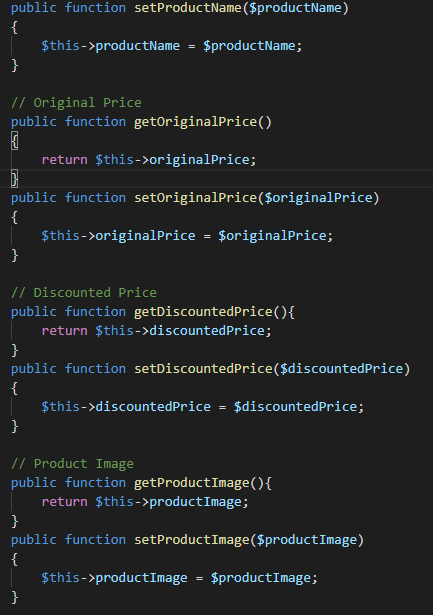








Figure : Add Update and Delete Product Code

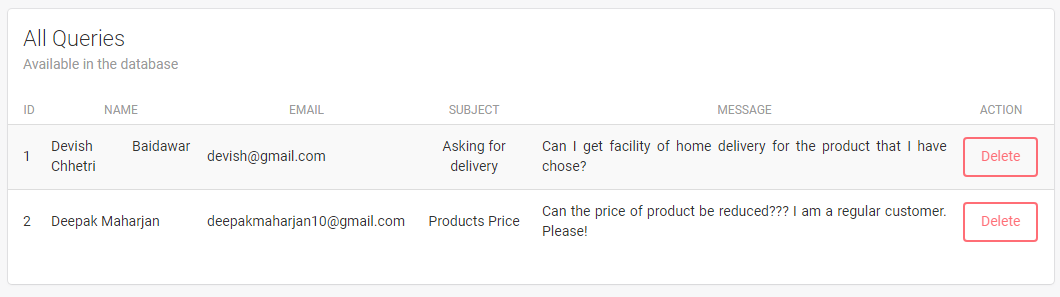


Figure : View Queries

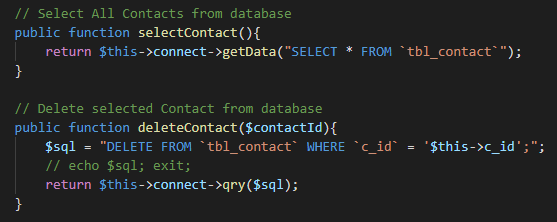


Figure : View and Delete Contact Code

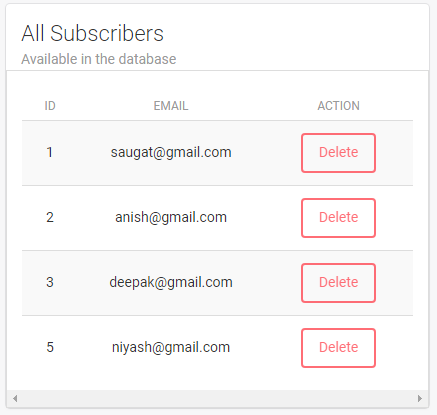
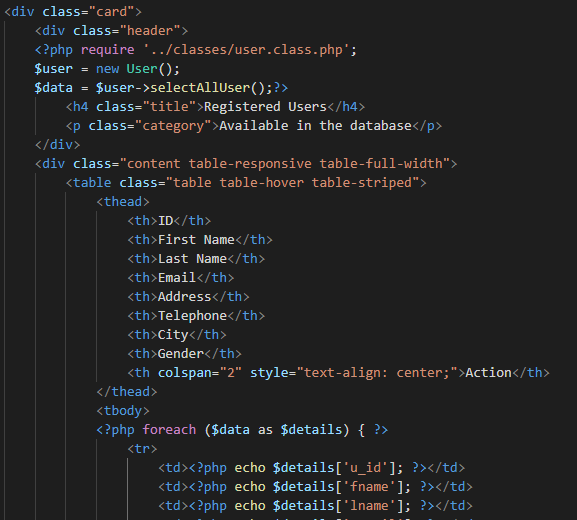
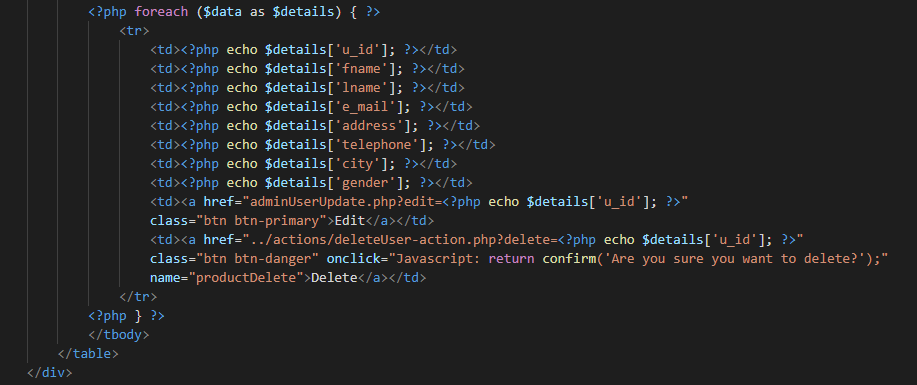


Figure : View Subscribed Users





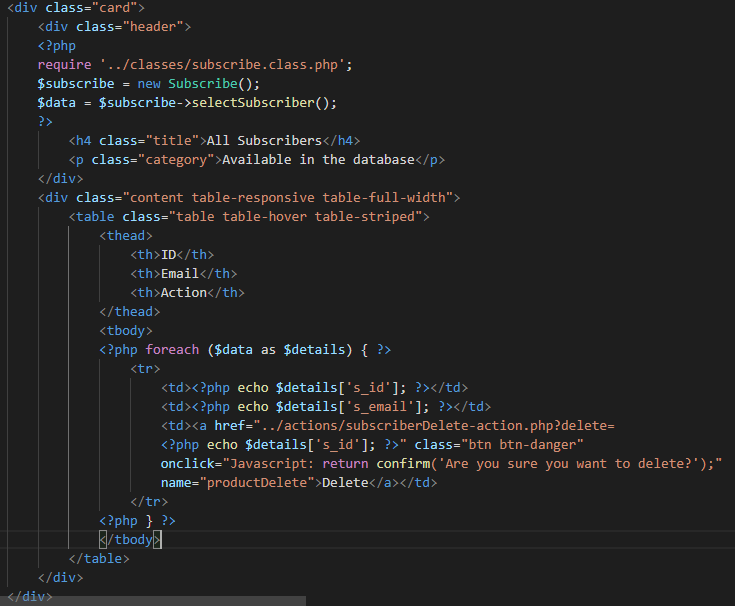


Figure : View Subscriber Code

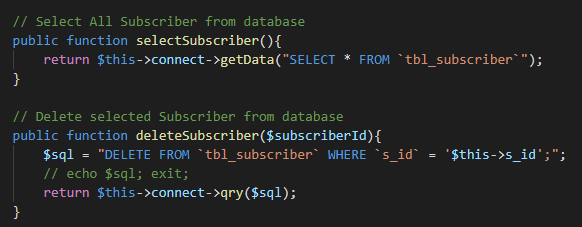


Figure : Select and Delete Subscriber Code

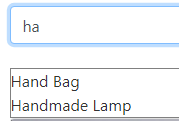


Figure : Search Form and Result

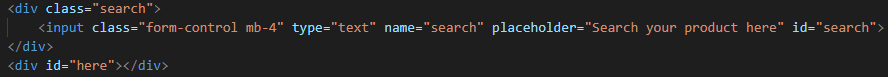


Figure : Search Form Code

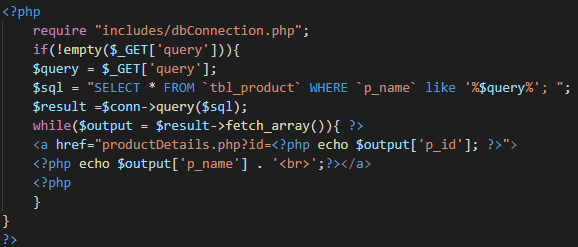


Figure : Ajax and Fetch Search Data Code

# Chapter 5: Testing

## 5.1 Introduction

## 5.2 Types of Testing

There are different types of testing and are good at its own. Usability testing, Integration testing, Unit testing, Black box testing, etc. are different type of testing.

I choose to do Unit testing and black box testing for this project.

### 5.2.1 Unit Testing

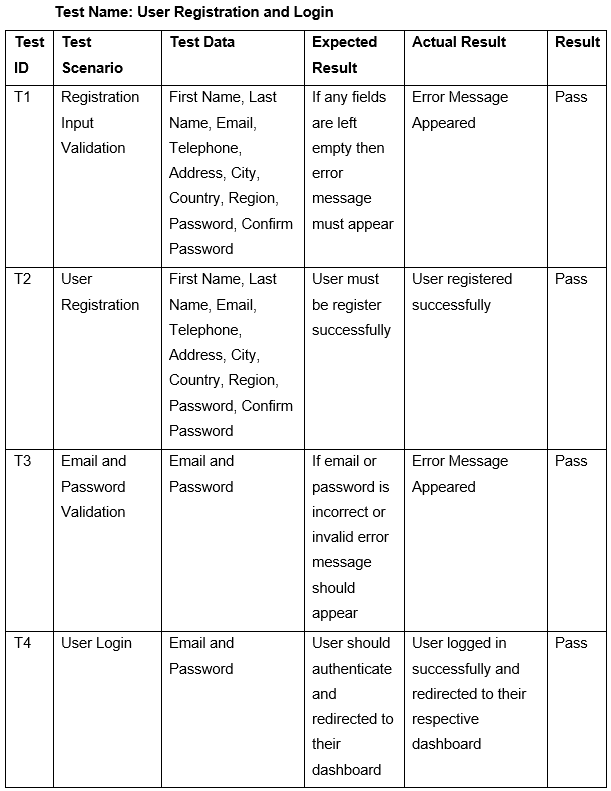
Unit Testing is a level of software where every individual features of an application is tested. It guarantees every unit of the product performs are in true form. Unit testing builds certainty while changing code and we can identify and changes happen while changing any code. (Software Testing Fundamantals, 2019)

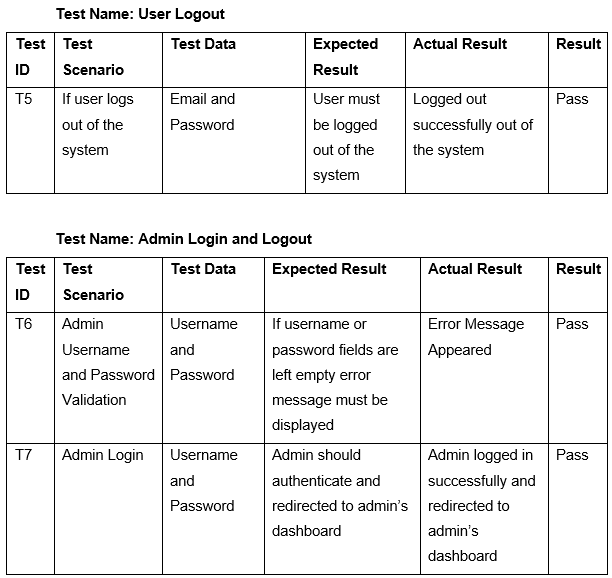
This type of testing can be done manually as well as automatically using prebuilt tools like PHPunit and so on.

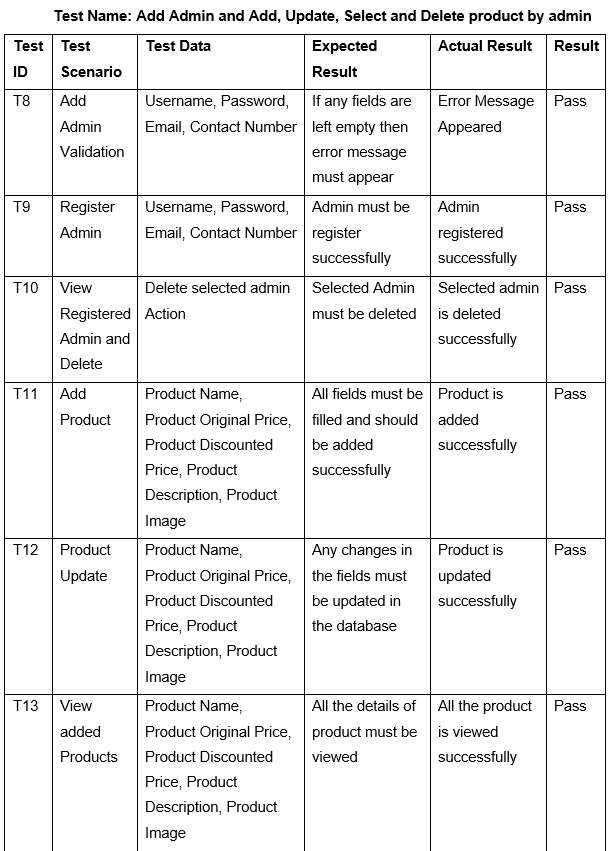
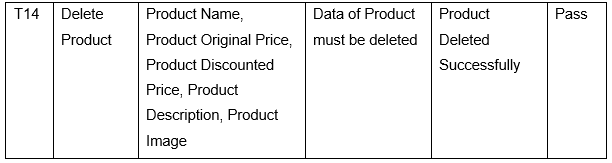
### 5.2.2. Black Box Testing

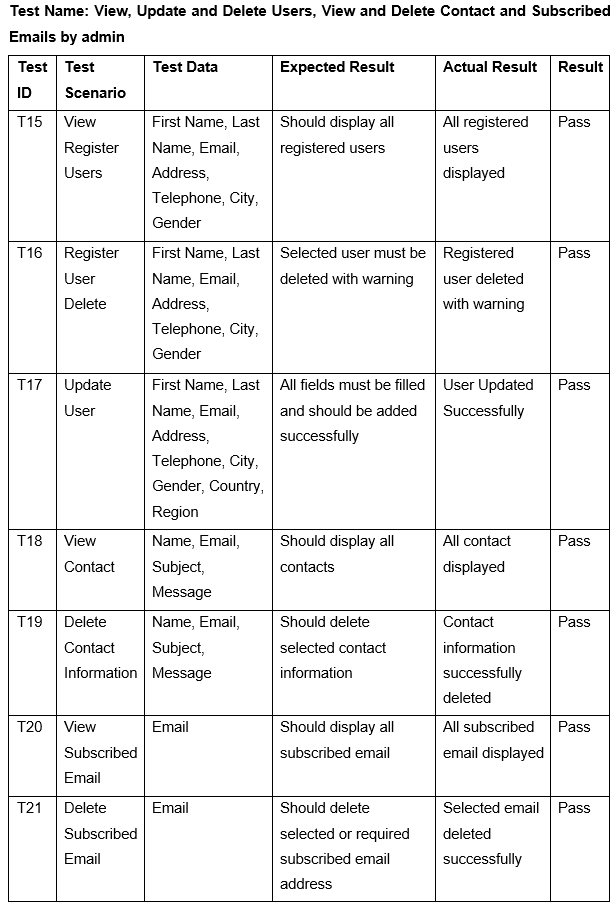
Black Box Testing, which is also known as Behavioral Testing, is a software testing method in which internal structure, design and implementation of the item is tested. This type of testing can be both functional and non-functional. Almost all the time this type of testing is functional. (Software Testing Fundamentals, 2019)

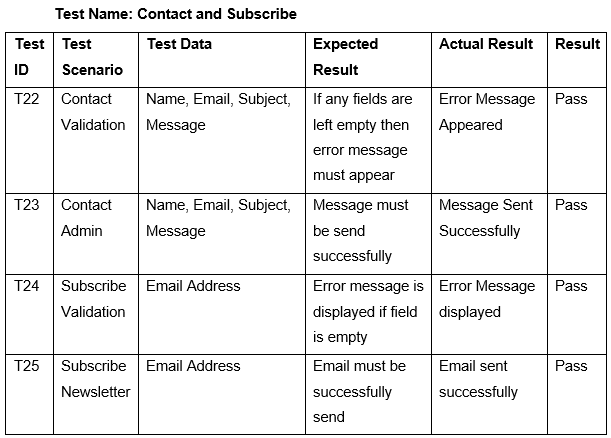
It checks whether function of system is working appropriately or not and discovers the defect in the functionality/behavior of the system. It distinguishes the imperfection right on time during development stage.

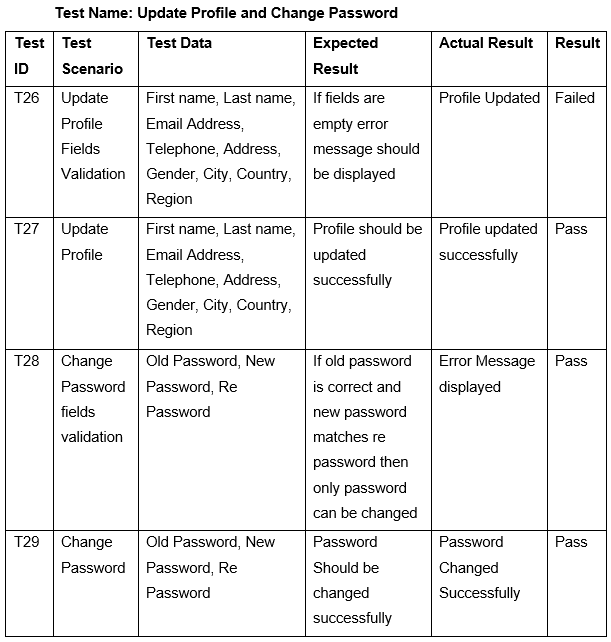


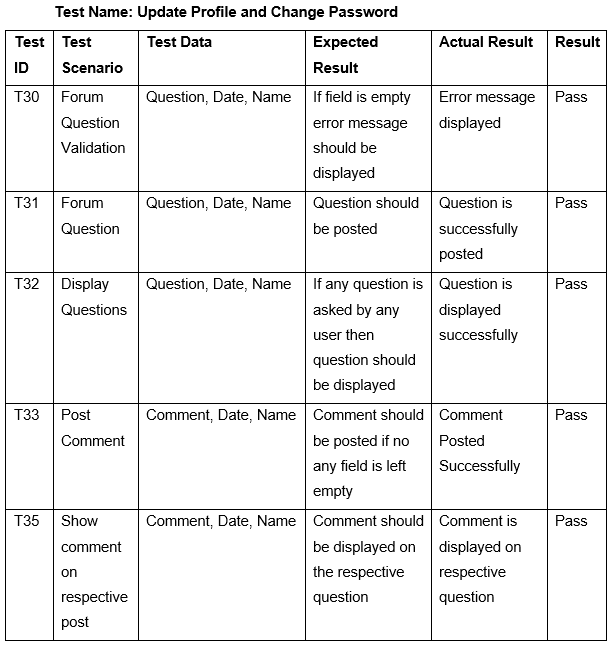


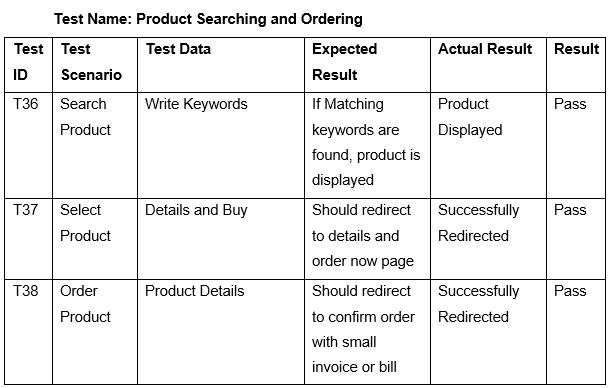












# Chapter 6: Other Project Issues

A part of arranged action that are embraced and finished over a fixed period, inside estimated cost and to accomplish certain goal is known as project. A condition that has negative effect for the task and which is happening in a project can be said as project issues. This issues are viable through good decisions.

## 6.1 Project Limitation

Some of the limitations of the project are listed below:

* Online payment is currently not available
* Multiple items cannot be requested at once
* Quality of the product is not 100% assured
* No any third-party login

## 6.2 Risk Management

Risk Management is the process of identifying the strength, weaknesses, opportunities, and threats of our project. It is important in the project because without it, we cannot define our objectives for the future. And if we define our objectives without taking risk management then we may lose the track once any risk occurs. Our project will go effectively if the risk is managed wisely.

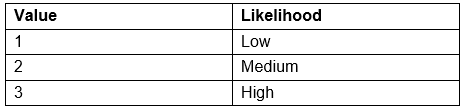


Fig: Risk Likelihood

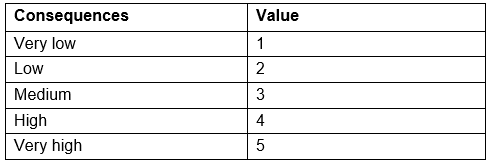


Fig: Risk Consequences Values

For identifying risk, we will use the following formula:

**Impact = Likelihood x Consequence**

**Risk Management Chart**



Fig: Risk Management Table

Some of the basic steps of risk management lifecycle are explained below:

1. **Identification:** We cannot manage our risk if we don’t know the risk. So identifying risk is the first step for risk management.
2. **Assessment:** Once the risk is identified, it needs to be examined in terms of likelihood and impact. This will then identify which risks should be in first priority.
3. **Treatment:** After accessing the risk, an approach for treating each risk should now be characterized.
4. **Monitoring:** Once the risk is identified, accessed and treated now we cannot leave it. It needs to be processed and reviewed, which is essential for proactive risk management.
5. **Reporting:** After all these four stages, the report of each four stages above should be described briefly for effective risk management.

## 6.3 Configuration Management

Configuration Management is the process of establishing and maintaining the regularity or uniformity of a project’s performance and functional attributes with its information throughout its life. It helps to manage all the configurations of the project through five key stages:

1. **Planning:** It deals with how will we record, track, control, and audit the project.
2. **Identification:** It includes functionality, requirements, design requirement and any other specifications of the project.
3. **Controlling:** It must be assessed, approved, and well documented.
4. **Status accounting:** It helps to control the version of the project and also trace out the changes made throughout the project.
5. **Auditing:** It includes all the tests to prove that the product conforms with the configuration requirements. (Wrike, 2019)

Following is the folders that manage the project:



Fig: Tree Chart

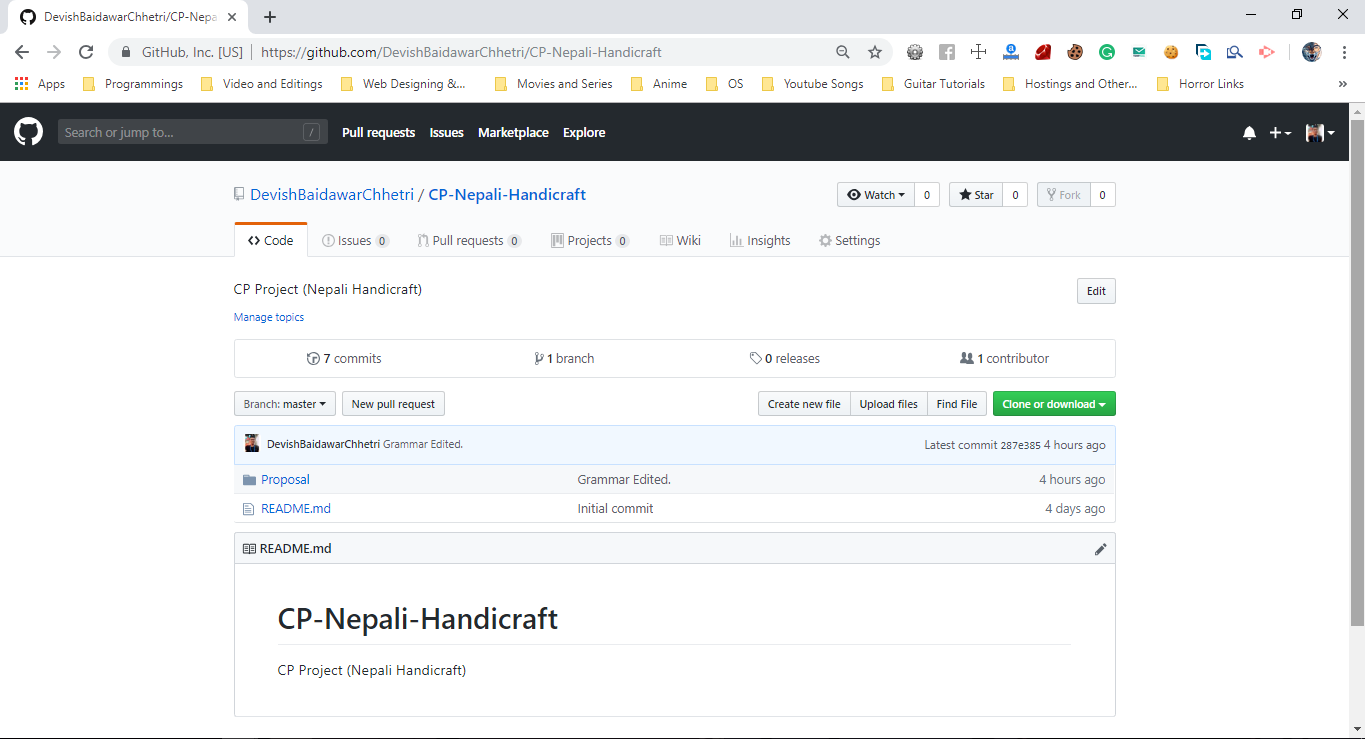


Fig: Screenshot of GitHub’s Repository

**GitHub Profile:** <https://github.com/DevishBaidawarChhetri>

**GitHub Repository URL:** <https://github.com/DevishBaidawarChhetri/CP-Nepali-Handicraft>

# Chapter 7: Conclusion

# Chapter 8: References

# Chapter 9: Appendix

# References

Anon., 2019. *Tutorialpoint.* [Online]   
Available at: https://www.tutorialspoint.com/software\_engineering/software\_design\_basics.htm  
[Accessed 15 May 2019].

Burge, S., 2015. *Root-Definition-Tool-v1.* [Online]   
Available at: https://www.burgehugheswalsh.co.uk/Uploaded/1/Documents/Root-Definition-Tool-v1.pdf  
[Accessed 29 April 2019].

folks, t., 2014. *softwaretestingclass.* [Online]   
Available at: https://www.softwaretestingclass.com/difference-between-black-box-testing-and-white-box-testing/

Rouse, M., 2017. *TechTarget.* [Online]   
Available at: https://searchsoftwarequality.techtarget.com/definition/unit-testing

Software Testing Fundamantals, 2019. *Unit-Testing.* [Online]   
Available at: http://softwaretestingfundamentals.com/unit-testing/  
[Accessed 25 June 2019].

Software Testing Fundamentals, 2019. *black-box-testing.* [Online]   
Available at: http://softwaretestingfundamentals.com/black-box-testing/  
[Accessed 25 June 2019].

TutorialsPoint, 2017. *UML Class Diagram.* [Online]   
Available at: https://www.tutorialspoint.com/uml/uml\_class\_diagram.htm  
[Accessed 7 May 2019].

Wrike, 2019. *what-is-configuration-management-in-project-management.* [Online]   
Available at: https://www.wrike.com/project-management-guide/faq/what-is-configuration-management-in-project-management/  
[Accessed 07 April 2019].