# Chapter 2: Analysis

# Chapter 2.1: Introduction to Analysis

Analysis defines as a detailed examination of anything complex in order to understand its nature or to determine its essential features from a thorough study. We perform analysis to know about the functional requirement and non-functional requirements of the material. Analysis pull the facts on the requirement that is needed for that particular subject. In software analysis as well, we gather all the requirement that the software should fulfill so that the software can be use by people with ease.

# Chapter 2.2: Analysis Methodology

Methodology is defined as the systematic and theoretical analysis of the methods applied to a field of study. This also refers as the steps that we need to follow while undertaking a project and the process that must be followed during it. **Analysis Methodology** whereas refers to the various steps taken to collect information, analyzing that information and the providing a document containing the requirement. For choosing what analysis methodology to be selected in doing any project we must know about various factors like how efficient it is, is it functional, is it accurate, is it ease to use, how reliable it is and so on. While focusing on these factors any analysis methodology chosen for the project will be able to provide the requirement information.

For my project ***Nepal Handicraft Online Market,*** I choose ***Hard Approach methodology*** as the main methodology. This approach refers as taking a highly structured approach to the analysis of information system that follow logical sequence of steps and adheres to rules, guidelines and standards. My project is small so this approach is perfect for my system. This approach follows the SSADM (Structured Systems Analysis and Design) method. For this method we have to create and design the Data Flow Diagram (DFD) which show how the system work. This approach ensures thorough planning and scheduling and also is easy to measure the progress of the work that can be shown through each step. I chose this methodology because of the merits it has for my system as this follow a waterfall model where you don’t go forward until you finish the step as a result, we get a better system.

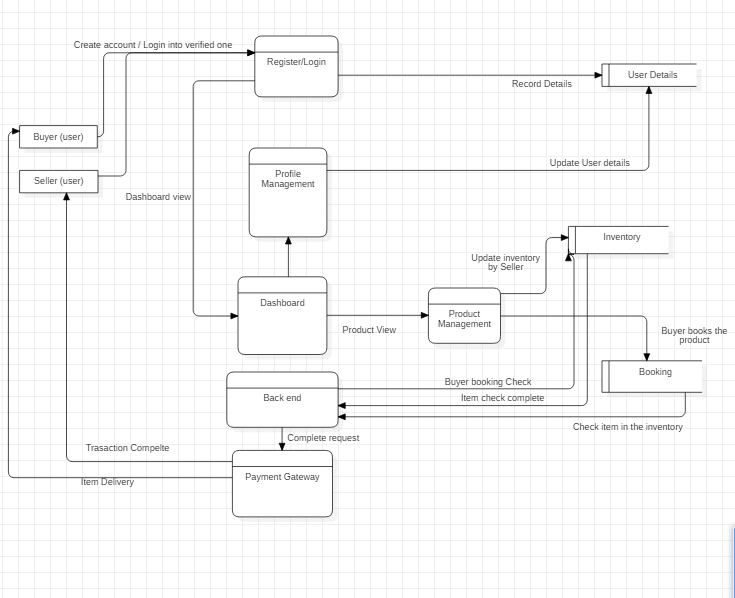


Figure 1: Data Flow Diagram for User

The following data flow diagram show how the user use the system. This DFD shows how the system process flow from one part to another with detail information on the working mechanism. This helps to get the grasp of the system process.

# Chapter 2.3: Feasibility Study

Feasibility study defines as an analysis that takes all of a project’s relevant factors into account that includes the economic, technical, legal and scheduling considerations to ascertain that the likelihood of completion of the project is successful. This is mainly done by project managers to discern the pros and cons of undertaking a project before they invest a lot of time and money in it. This also provides company management with vital information that could prevent any risk that may occurs. The areas that feasibility study examines and what determine factor determine the project are given and explained below:

* **Technical Feasibility**

This study focuses on the availability of the technical needs and resources that are needed for the software development phase that includes resources like hardware, software, memory and so on.

My project has enough technical equipment, hardware and software requirement which determines that the project is technically feasible.

* **Social Feasibility**

This type of feasibility study adjusts the social factors that includes political condition, environment of the targeted area covered by that particular project. This also includes the social norms and values of the area that influences the project.

My project has none features that doesn’t follow the social norm and that may hamper the society in any way.

* **Legal Feasibility**

Legal Feasibility is a type of feasibility study that check whether the project fulfill the legal requirements such as data protection law, social media law and so on. This allows us to create a project that follow the legal laws.

In my project there are no legal issues that may take place as well as any features that have been added are legally feasible.

* **Financial Feasibility**

This study is dependent on how much the project can benefit financially. This study performs the cost-benefit analysis of the project, determining viability, cost and benefits of the project as well as the financial benefits.

Since my project is based on E-commerce there is no issues that have been found which determines my project to be financially feasible.

* **Market Feasibility**

This feasibility study checks the market of the project. This also determines the users of the project after the development process is over and how much impact will occur in the market.

I have research thoroughly and found different system like the one I am trying to develop but none on the product that I have focused for the development so this project will cover the market just fine as a result this project is market feasible.

# Chapter 2.4: SRS (Software Requirement Specification)

Software requirement specification is a detailed description on how the software is being developed which have fulfilled its functional and non-functional requirements. This is made based on the agreement between customer and contractors. This document includes all the necessary requirement that are required for the project development. A good SRS defines how the software will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real-life scenarios.

# Chapter 2.4.1: Functional Requirements

In the terms of software engineering and system engineering, functional requirement defines a function of a system or its components where functions are described as a specification of behavior between outputs and inputs. There is various functional requirement in my project and some of them are listed in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Functional Requirement** | **Data Required** | **Rational** | **Dependency** |
| F1 | Admin Signup | Name, Username, Password, Email, Phone number | Admin Login Data |  |
| F2 | Admin Login | Username, Password | Security and privacy | F1 |
| F3 | Add Category | Name, Type | Category Add | F1, F2 |
| F4 | User Signup | Name, Username, Password, Email, Phone number | User Login Data |  |
| F5 | User Login | Username, Password | Security and Privacy | F4 |
| F6 | Upload Products | Name, Price, Quality, Type, Details | Uploading product information | F1, F2, F3, F4, F5 |
| F7 | View Products | Name, Price, Quality, Type, Details | View product information | F1, F2, F3, F4, F5 |
| F8 | Update Products | Name, Price, Quality, Type, Details | Update product information | F1, F2, F7, F4, F5 |
| F9 | Delete Products | Name, Price, Quality, Type, Details | Delete Product | F1, F2, F7, F4, F5 |
| F10 | Manage Profile | Name, Username, Password, Email, Phone number | Update or change personal details | F1, F4, F2, F5 |
| F11 | Chat with admin | Username, Message | Interaction with admin | F4, F5 |
| F12 | Chat with customer | Message, Name, Phone number | Interaction with product owner | F4, F5, F7 |
| F13 | View User Details | Name of user, Username, Email | Get data of users and their interaction with system | F1, F2 |
| F14 | Manage Users | Name of user, Username, Email | Suspend and delete unwanted and untrusted users | F1, F2, F13 |
| F15 | Buy Products | Name, Email, Phone number, Payment method detail, Username | Buying product through the software | F4, F5, F7 |
| F16 | Book Products | Name, Email, Phone Number, Username | Book the product listed in the site | F4, F5, F7 |
| F17 | Booking Cancellation | Product name, Name, Email, Phone number, Username | Cancel the booked item | F4, F5, F10, F16 |
| F18 | Deploy Message when product is booked or bought | Name of user, Username, Phone number, Time, Message | Informs the product owner about the transaction | F1, F2 |

# Chapter 2.4.2: Non-Functional Requirements

In system and requirement engineering, a non-functional requirement is a requirement that specifies criteria used to judge the operation of the system rather specific behavior. This is also known as quality attributes of a system. This is very important for a system to make it secure, user-friendly, reliable, effective and more reliable. Some of the non-functional requirements are:

|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **Description** |
| 1 | Security | This is very important requirement for developing a software which can be done by using strong password and encryption algorithms. |
| 2 | Performance | This measures how fast and effective the system is. By using the latest technology, we can increase the performance of the system |
| 3 | User-friendly | The system developed should be easily usable and accessible. High user interface makes it more usable and user-friendly. |
| 4 | Reliability | The developed system must be reliable and there should be no failures while operating. |
| 5 | Availability | The system should be easily available and accessible by all. |
| 6 | Scalable | The system much scale according to the devices it is used on as well as any operating system. This make the system more robust and adaptive to large market. |
| 7 | Data Integrity | The data and information should be secured and only admin should be allowed to delete and change the data in the system. |
| 8 | Maintainability | The system should be maintained in regular interval |
| 9 | Recoverability | The system should be recoverable if any problem occurs during any process |
| 10 | Manageability | The system should be easy to manage |

# Chapter 2.4.3: MoSCoW Prioritization

Prioritization is the process of arrangement of the requirements and features according to their order of importance. MoSCoW is a prioritization technique for helping to understand and manage priorities. The letters stand for

* Must have
* **S**hould Have
* **C**ould Have
* **W**on’t have.

This prioritization allows us to know what features are the must in the system and what features are less needed for the system. Some of the priority order for the requirement are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Features | Prioritization | Rational |
| 1 | User/ Admin registration | Must have | For registration in the system by user and admin |
| 2 | User/ Admin Login | Must have | To login in the system by user and admin |
| 3 | Upload/ update/ delete / view Products | Must have | To upload, update, delete and view product |
| 4 | Manage Users | Should have | To manage users |
| 5 | Manage Profile | Should have | For handling the profile data of the user |
| 6 | Chat with Customer | Would have | To chat with the customer |
| 7 | View User details | Would have | To display the user details |
| 8 | Chat with admin | Would have | To chat with the admin |
| 9 | Inform information on product transaction | Must have | To send message after transaction of product |
| 10 | Book and buy products | Must have | To book and buy the product |
| 11 | Security | Must have | Fundamental Function within the system |
| 12 | User-friendly and reliable | Must have | Fundamental function within the system |
| 13 | Data Integrity | Must have | Fundamental function within the system |
| 14 | Availability | Must have | Fundamental function within the system |
| 15 | Scalable | Must have | Fundamental function within the system |
| 16 | Performance | Must have | Fundamental function within the system |
| 17 | Booking Cancellation | Should have | For cancelling the product that have been booked |
| 18 | Add Category | Must have | To sort base on the what category the product type is |
| 19 | Manage Users | Should have | To suspend and delete the users from the system |
| 20 | Manage Profile | Must have | To manage the profile of user |
| 21 | Manageability | Must have | Fundamental function within the system |
| 22 | Recoverability | Must have | Fundamental function within the system |
| 23 | Maintainability | Must have | Fundamental function within the system |

# Chapter 2.4.4: Hardware software Specification

For the system some specific hardware and software are required for the development and design of the project. Below are the list of hardware and software required for the project:

**Hardware Specification**

* Processor: Celeron 500MHz or any Pentium processor
* Ram: 4GB or higher
* Hard Disk: 100Gb or higher
* Display Type: Standard VGA or SVGA card
* Peripherals: Keyboard, Mouse

**Software Specification**

* Operating system: Windows 7 or higher, Linux
* Front-end: Bootstrap 4.0.0
* Back-end: PHP, MySQL, XAMPP 7.3.2

# Chapter 2.5: Use Case Diagram

Use case diagram is a dynamic or behavior diagram drawn in UML which shows the functionality of the system using actors and use cases. Use cases are a set of actions, services and functions that the system needs to perform. Various components of use case diagram are

* Actors: It represents the people who use the system and represented in UML by stick figures.
* Use cases: It defines the process in the system represented by oval shapes.
* Associations: This shows the relation between use cases and actors.
* System Boundary Boxes: It refers to rectangular box where system is posted.

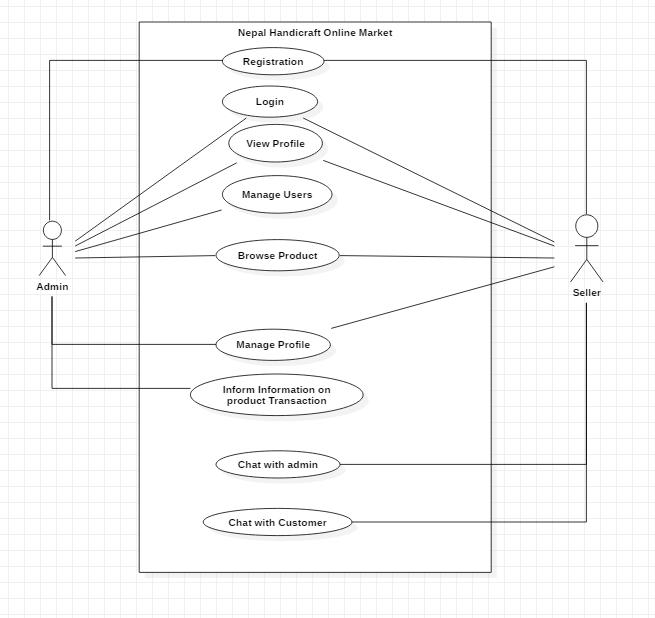


Figure 2: Use Case Diagram of Nepal Handicraft Online Market

|  |  |
| --- | --- |
| Justification | flow how user and admin logs and register in the system |
| Primary Actors | Admin/User |
| Secondary Actors | N/A |
| Primary Flow | 1. Normal User Initially register into the system leaving profile information 2. The user logs the system through the profile information |
| Alternative Flow | N/A |

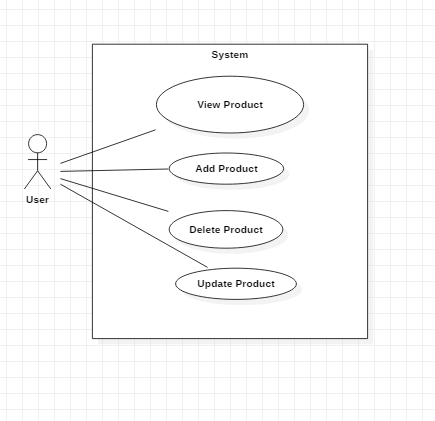


Figure 4: Product Entry/ Delete/ Update/ View

|  |  |
| --- | --- |
| Justification | This use case shows how the user can upload the product data, update the data if any mistake, delete the product information if not needed and view the product detail |
| Actors | User |
| Supporting Actors | N/A |
| Primary Flow | 1. User Add the product data in the system. 2. User update the product data by searching the product and changing it data 3. Delete the product data if not needed |
| Alternative Flow | N/A |

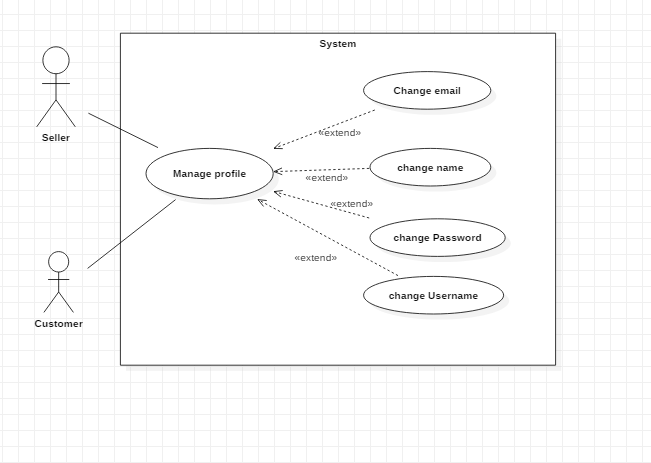


Figure 5: Profile Management

|  |  |
| --- | --- |
| Justification | This use case show how user can manage their profile |
| Actors | Users |
| Alternate Actor | N/A |
| Primary Flow | 1. User change their name, email, phone number, username and password if needed. |
| Alternate Flow | N/A |

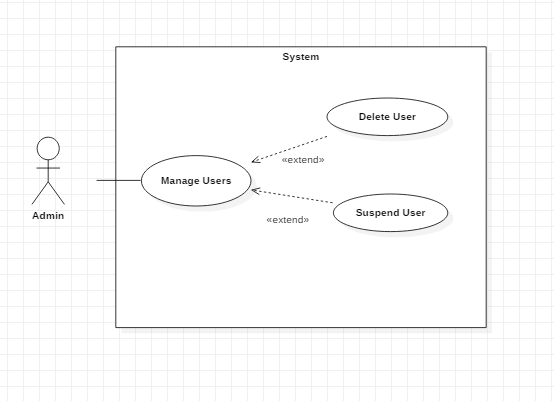
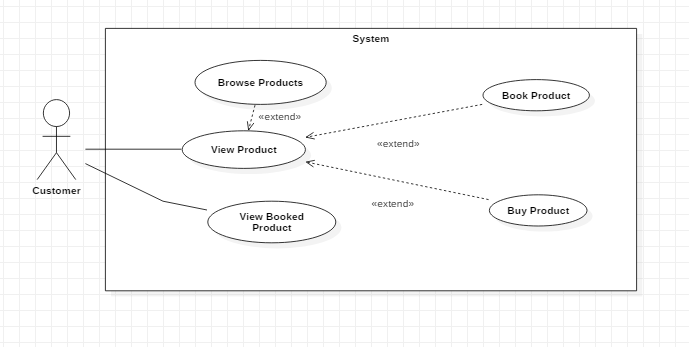


Figure 6: User Management

|  |  |
| --- | --- |
| Justification | This use case show what the admin can do to manage the users. |
| Actors | Admin |
| Alternate Actor | N/A |
| Primary Flow | 1. The admin can suspend the user who are doing the work that doesn’t follow the rule. 2. The admin can delete the user who are not using. |
| Alternate Flow | N/A |



|  |  |
| --- | --- |
| Justification | This show the what customer can do when they view product |
| Actors | Customer |
| Secondary Actors | N/A |
| Primary Flow | 1. The customer can book and buy the product after they have viewed the product. 2. They can see what product they have booked |
| Secondary Flow | N/A |

The figures above represent the use case diagram of this project. The interaction of users and admin in the system in shown in the figure above. These diagram shows what functions are performed by the users and admin. I have also shown the various behaviors done by the actors in the use cases diagram above.

# Chapter 2.6: NLA (Natural Language Analysis)

Natural Language Analysis is defined as the use of ability of systems to process sentences in a natural language such as English rather than in specialized artificial computer language such as C++ or C#. NLA helps to understand the system. The process of NLA involves in the division of the noun, verbs and adjectives that are sorted out to obtain the candidate classes, functions and attributes.

|  |  |  |
| --- | --- | --- |
| Nouns (Classes) | Verb (Function) | Adjective (Attributes) |
| Admin | Signup, Login, Add\_category, View\_profile | Admin\_id, Admin\_name |
| User | Buyproduct, view\_product | User\_id, Name, username, email, password |
| Product | Edit\_product, Add\_product, Delete\_product, view\_product, Book\_product | Product\_id, product\_name, price, details |
| Ordered \_item | Create\_order, Update\_order, Cancel\_order | Username, product\_id, Order\_date, Order\_id |

# Chapter 2.7: Initial Class Diagram

Class Diagram is static diagram that represents the static view of an application. Class Diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. Through the help of the NLA above I have designed the class diagram which is shown below:

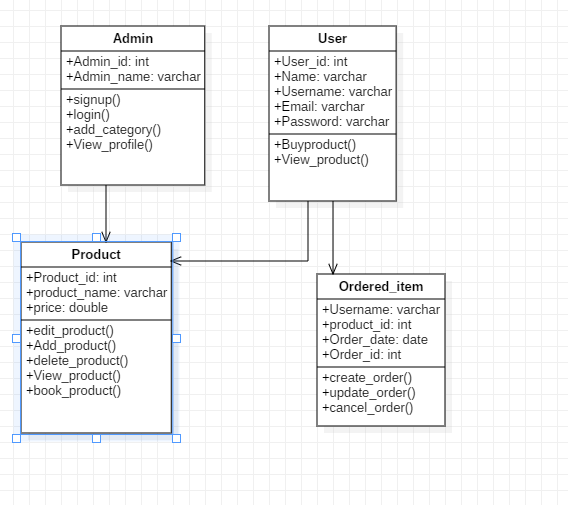
1

Figure 7: Initial Class Diagram for Nepal Handicraft Online Market

The following class diagram is prototype for the final class diagram that will be made. In the following class diagrams, I have made 4 classes which contains their respective attributes and functions. The classes are linked with one another with direct association. The following class diagram will help in designing the final class diagram.