# Chapter 2: Analysis

## 2.1 Introduction

Analysis means a systematic examination and evaluation of data and facts to uncover and understand cause-effect relationships, thus providing basis for problem solving and decision-making. (Analysis, 2020) We should do a proper analysis before starting any project. Proper analysis of the project will help to make the project much effective, and also facilitates us in different ways.

## 2.2 Analysis Methodologies

Analysis is one the most important part of developing a system which will help us to point out the features and functionality that an application needs to have. It helps us to point out all of the examination that are needed in an application to make the application much more applicable.

For analysis of this project, object-oriented analysis [OOA] methodology has been chosen. It analyzes and designs an application based on the system’s logical components that interact with one another. It also combines all characteristics, states, and behaviors together into a single analysis process, instead of splitting them into different stages, as many other analysis methodologies would do.

Following interaction diagrams have been used in my project;

* Class Diagram

Class Diagram in the Unified Modeling Language (UML) is a type of static structures diagram that shows the collection of classes, attributes, interfaces, their association and relationships among objects.

* Activity Diagram

Activity Diagram is a behavioral diagram in Unified Modeling Language (UML) which describes the dynamic aspects of the system. It is an advanced version of flow chart that models the flow of one activity to another activity.

* Sequence Diagram

Sequence Diagram are interaction diagrams that detail how operations are carried out. They are time focus and show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. (Sequence Diagram, 2020)

Detailed insight of above mentioned interaction diagrams will be given in the design phase.

## 2.3 Information Gathering

The simplest way to define information gathering would be the process of collecting information about something you are interested in. Information are gathered in order to understand the requirements of the system. The more the information gathered about the system, the more the probability to obtain relevant results. I have used the following two information gathering techniques which involves direct communication with the clients; interview and Questionnaire.

Interview

An information gathering interview is a directed conversation with a specific purpose that uses a question-and-answer format. In the interview you want to get the opinions of the interviewee and his/her feelings about the current state of the system, personal and organizational goals, and informal procedures for interacting with information technologies. While interviewing, seek the opinions of the person you are interviewing. Opinions may be more important and more revealing than facts. (Interviewing in Information Gathering, 2020) Some of the questions that I prefer to ask to the interviewee are;

* What are the general and specific features that you want in the system?
* How can we make the system much more user friendly?
* Are you having any issue while using the system?

Questionnaire

Questionnaire in an information gathering technique helps system analysts to study attitudes, beliefs, behavior, and characteristics of several people in the system/organization who may be affected by the current and proposed systems. Responses gained through questionnaire using closed questions can be quantified and can be used to survey a large sample of system users to sense problems or raise important issues before interviews are scheduled. It helps to distinguish the behavior and preferences of people in a specific field. (System Analysis, 2020) Some of the questionnaire that I prefer to ask are;

* What is your budget estimation for the project?
* Do you need further maintenance services?
* Have you registered domain for the company?
* Does your company have branding inclusions like logo/banners/established images?
* Do you have technicians to run the website?
* How often do you update the contents in website?
* Do you have any hosting company preference?
* Do you want us to train your staff to operate the website?
* What is your deadline for the project?
* Who are going to use the website?
* Would you like us to handle your other web aspects like digital marketing, search engine optimization (SEO) or you already have other dedicated member?
* Who are your targeted audience?
* Who will provide the contents for the website?
* Do you want to include any 3rd party application in your website?
* What are the features you want in your website?

## 2.4 Feasibility Study

A feasibility study is an analysis that takes all of a project’s relevant factors into account-including economic, technical, legal, and scheduling considerations- to ascertain the likelihood of completing the project successfully. Feasibility studies are used to discern the pros and cons of undertaking a project before they invest a lot of time and money into it. It also provides a company’s management with crucial information which could prevent the company from entering blindly into risky businesses. It is always good to have a contingency plan that you test to make sure it’s a viable alternative in case the first plan fails while doing a feasibility study. (feasibility study, 2020)

For the project I have done the following types of feasibility study;

1. Technical Feasibility

Technical feasibility study focuses on the technical resources available to the organization. The project is a total website which is made on Laravel (a php web framework). To study technical feasibility of the project, we need to confirm if the existing computer system are compatible with the technologies that we will use. The project will work perfectly across all the devices, browsers since it will be optimized properly. The requirements needed for project to run successfully are freely available and manageable. Therefore, the project is technically feasible.

1. Economic Feasibility

Economic feasibility typically involves a cost/benefits analysis of the project which helps organizations to determine the viability, cost, and benefits associated with a project before the financial resources are allocated. Since the project is made up of freely available technologies, basic operating computer system, no cost will be charged. Some cost will be charged during the hosting of the website, for the development team. The project will be small, efficient so there will be training for short period of time which minimizes the cost of training. And while the records of clients and other credentials will be off the paper it will also minimize the cost. Therefore, the project is economically feasible.

1. Legal Feasibility

Legal feasibility determines whether any aspect of the proposed project conflicts with legal requirements like data protection acts or social media laws. (Feasibility Study, 2020) For the project, I have used open source tools that have free license which will not permit the violation of the legal conflicts. And as for the website, we have a privacy policy which will clear the data related to the users. Thus, it will help the website to be fully secure which depicts no legal crimes. Therefore, the project is legally feasible.

1. Operational Feasibility

Operational feasibility involves undertaking a study to analyze and determine whether and how well the organization’s needs can be met by completing the project. It also examines how a project plan satisfies the requirements identified in the requirements analysis phase of system development. (Feasibility Study, 2020) A proper orientation and basic trainings will be provided to the users so the introduction of new system will not affect anyone. In this way, the project will be operationally feasible.

1. Scheduling Feasibility

Scheduling feasibility is the most important for project success because a project will fail if not completed on time. An organization estimates how much time the project will take to complete. My project will have enforced several time estimates on which several chunks of works related to the project will be done which will balance the work load and help the project to be completed on time. In this way, my project will be on time.

## 2.5 System Requirement Specification

A System Requirements Specification (SRS) is a documents or set of documentation that describes the features and behavior of a system or software application. It includes a variety of elements that attempts to define the intended functionality required by the customer to satisfy their different users. (System Requirements Specifications, 2020)

## 2.5.1 Functional Requirement

The functional requirement specification documents the operations and activities that a system must be able to perform. It is designed to be read by a general audience. Readers should be able to understand the system, but no particular technical knowledge should be required to understand the document. (Functional Requirements, 2020)

List of functional requirements are shown below;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Title | Description | Dependency | Rational |
| FR\_101 | Registration | Add new user | N/A | To access secure system |
| FR\_102 | Login | Registered user can login to the system | FR\_101 | To provide access to a valid user |
| FR\_103 | Verify password | To check the password entered with the database | FR\_101,  FR\_102 | To authenticate the user |
| FR\_104 | Display Dashboard | To provide a functional front page | FR\_101,  FR\_102 | To navigate entire system easily |
| FR\_105 | Edit Profile | Change personal information | FR\_101,  FR\_102 | To update user’s information in case of emergency |
| FR\_106 | Add Category | Add new category of fruits and vegetables | FR\_102 | Add new type of category of the product |
| FR\_107 | Update Category | Add new details on the existing category details | FR\_102,  FR\_106 | For any additional information is needed to be add |
| FR\_108 | Delete Category | Remove the category | FR\_102,  FR\_106 | To remove the unnecessary category |
| FR\_109 | Add Product | Add new product | FR\_102,  FR\_106 | To add new product under different category |
| FR\_110 | Update Product | Add new details on the existing product details | FR\_102,  FR\_109 | To add extra information about the product or remove the unwanted details |
| FR\_111 | Delete Product | Remove the product | FR\_102,  FR\_109 | To remove the product that is unavailable |
| FR\_112 | Add to cart | Add product to the cart | FR\_102,  FR\_109 | To buy the product |
| FR\_113 | Delete from cart | Delete the added product | FR\_102,  FR\_112 | To not to buy the product incase of change of mind |
| FR\_114 | View Order Details | Lists out all the order | FR\_102,  FR\_112 | Details the list of all the order placed by the user |
| FR\_115 | Approve Order | Approve the order for delivery | FR\_102,  FR\_112 | To approve the order placed by the user. |
| FR\_116 | Delete Order | Deletes the order | FR\_102,  FR\_114 | To delete the order incase of lack of the product |
| FR\_117 | Add Message | Sends message to the admin | FR\_102 | To send any information that a user possesses to the admin |
| FR\_118 | View Message | Lists out all the message | FR\_102,  FR\_117 | To view the message the users have send to the admin |
| FR\_119 | Delete Message | Deletes the message | FR\_102,  FR\_117 | To delete the message the admin has already seen |
| FR\_120 | Manage Access | Maintains the level of access | FR\_101,  FR\_102 | To provide certain level of authority |
| FR\_121 | Logout | Logout at desired time | FR\_102 | To secure users’ data to be misused |

Table 1: Functional Requirement

## 2.5.2 Non-Functional Requirement

Non functional requirements define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

List of non-functional requirements are shown below;

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Title | Description | Dependency |
| NFR\_101 | Scalability | The system should be able to manage the users’ information. | N/A |
| NFR\_102 | Verification | The system should verify the password and the point of access to verify who the user claim to be. | NFR\_104 |
| NFR\_103 | Reliable | The system should be strongly reliable so that the users’ information is strongly preserved and protected. | NFR\_109 |
| NFR\_104 | Security | The system should be secure since it contains all the valuable information about the users. | NFR\_102 |
| NFR\_105 | Efficiency | The system should be efficient so that the users can access it without any errors. | NFR\_105 |
| NFR\_106 | Acceptance | The system should accept all the users new or old registered to the system. | NFR\_102 |
| NFR\_107 | Intuitiveness | The system should be easily understandable so that the users find it more way friendly to use. | NFR\_106 |
| NFR\_108 | Usability | It defines that the services provided by the system should be easily useable with no problems. | NFR\_103,  NFR\_105 |
| NFR\_109 | Availability | The system should be accessible form anywhere so that the user can do what ever they can with the help of the site. | NFR\_103,  NFR\_106 |
| NFR\_110 | Quality | The system should be qualified for performing all the functional and non-functional requirement of the system. | NFR\_105,  NFR\_107,  NFR\_108 |

Table 2: Non-Functional Requirement

## 2.5.3 MoSCoW Prioritization

MoSCoW Prioritization, also known as the MoSCoW method or MoSCoW analysis, is a popular prioritization technique for managing requirements. The method is commonly used to help key stakeholders understand the significance of initiatives in a specific release.

The acronym, MoSCoW stands for 4 different categories of initiatives: must-haves, should-haves, could-haves, and will not have. (MoSCoW Prioritization, 2020) The MoSCoW prioritization for my project is as follows:

|  |  |  |
| --- | --- | --- |
| ID | Functional and Non-Functional Requirements | MoSCoW |
| MP\_01 | Registration | Must have |
| MP\_02 | Login | Must have |
| MP\_03 | Verify Password | Must have |
| MP\_04 | Display Dashboard | Should have |
| MP\_05 | Edit Profile | Could have |
| MP\_06 | Add Category | Must have |
| MP\_07 | Update Category | Must have |
| MP\_08 | Delete Category | Should have |
| MP\_09 | Add Product | Must have |
| MP\_10 | Update Product | Must have |
| MP\_11 | Delete Product | Should have |
| MP\_12 | Add to cart | Must have |
| MP\_13 | Delete from cart | Must have |
| MP\_14 | View order details | Should have |
| MP\_15 | Approve order | Must have |
| MP\_16 | Delete order | Should have |
| MP\_17 | Add Message | Must have |
| MP\_18 | View Message | Must have |
| MP\_19 | Delete Message | Should have |
| MP\_20 | Manage access | Must have |
| MP\_21 | Logout | Must have |
| MP\_22 | Scalability | Could have |
| MP\_23 | Verification | Must have |
| MP\_24 | Reliable | Could have |
| MP\_25 | Security | Must have |
| MP\_26 | Efficiency | Must have |
| MP\_27 | Acceptance | Should have |
| MP\_28 | Intuitiveness | Must have |
| MP\_29 | Usability | Should have |
| MP\_30 | Availability | Should have |
| MP\_31 | Quality | Must have |

Table 3: MoSCoW Prioritization

## 2.5.4 Requirement Specification

A requirement specification is a document that captures complete description about how the system is expected to perform. The requirement specification for the project is as follow:

Software Requirement

Programming Language: PHP

Framework Used: Laravel

User Interface Design: HTML, BOOTSTRAP, jQuery

Operating System: Windows 7 and above

Server: Apache

Database: MySQL

Web Server Solution Package: Xampp

Hardware Requirement

RAM: 2GB and above

Processor: Intel Processor Dual Core and above

## 2.6 Use Case Diagram

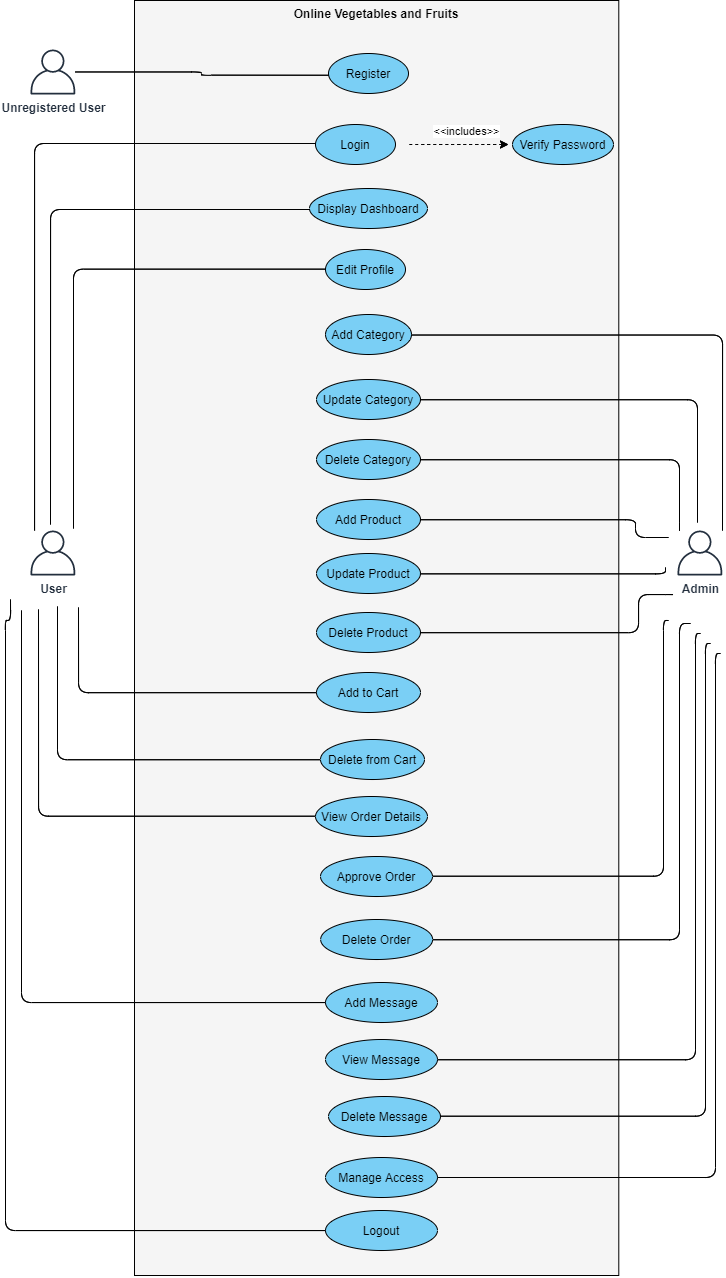


Figure 1: Use Case Diagram

A use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). A key concept of use case modeling is that it helps us design a system from the end user’s perspective. It is an effective technique for communicating system behavior in the user’s terms by specifying all externally visible system behavior. (Use Case Diagram, 2020) The use case diagram shown above describes the relationship between the users and the system including the system’s functions. The relation is explained in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Use Case Title | Summary | Alternative Sequence | Actor |
| UC\_01 | Register | User need to register themselves to the system to access the system. | N/A | Unregistered  User |
| UC\_02 | Login | Administrative and regular users’ login is done. | In case of incorrect password a message pop ups. | Admin,  User |
| UC\_03 | Verify password | The system verifies the password entered with the password on the database | N/A | The System |
| UC\_04 | Display dashboard | Both users land firstly here. | N/A | Admin,  User |
| UC\_05 | Edit Profile | Both users can edit their profile. | Message is displayed after editing the profile. | Admin,  User |
| UC\_06 | Add Category | Admin adds new category. | Message is displayed after a category is added. | Admin |
| UC\_07 | Update Category | Admin updates the category. | Message is displayed when the category is updated. | Admin |
| UC\_08 | Delete Category | Only admin can delete the category. | Message is displayed when the category is deleted. | Admin |
| UC\_09 | Add Product | Admin can add new products. | N/A | Admin |
| UC\_10 | Update Product | Admin can update the existing product. | N/A | Admin |
| UC\_11 | Delete Product | Admin can delete the existing product. | N/A | Admin |
| UC\_12 | Add to cart | Users can add the product to their cart | N/A | User |
| UC\_13 | Delete form cart | Users can delete the product they added to their cart | Message will be displayed when a product form the cart is deleted. | User |
| UC\_14 | View order details | Admin will view all of the orders made by the users. | N/A | Admin |
| UC\_15 | Approve Order | Admin is capable of approving the orders. | N/A | Admin |
| UC\_16 | Delete Order | Admin can delete the orders. | N/A | Admin |
| UC\_17 | Add Message | Users can send message to the admin. | N/A | Users |
| UC\_18 | View Message | Admin will see the message send by the users. | N/A | Admin |
| UC\_19 | Delete Message | Admin can delete the message. | A message saying message deleted will be displayed | Admin |
| UC\_20 | Manage Access | Only admin is capable of controlling the access provided to the users. | N/A | Admin |
| UC\_21 | Logout | Both users and admin can logout of the system as per their choice. | Logged out successfully message will be displayed. | Users,  Admin |

Table 4: Use Case

## 2.7 System Architecture

The system architecture describes how a product is composed of components and how the components are related to each other via interfaces. (System Architecture, 2020)

## 2.7.1 National Language Architecture [NLA]

Salaka Vegetables and fruits is a farm where fresh, local fruits and vegetables are grown and are sold. It is located at Salleri, Solukhumbu and is at service since the early 2000. Here a large amount of fresh vegetables and fruits are grown and sold at the local market. All the works in this farm are done in a traditional method. They sell their products only at local markets. Since it produces vegetables and fruits at large number the consumption of these goods at local market is lower. Due to this these products are going to waste.

So, the owner of this farm intends to expand its service to other places. So, he has asked to make a web application that enables users from other areas to be able to buy their products.

Here are his requirements:

There are many farmers and people who wish to sell their fruits and vegetables online. These people who produce fresh vegetables and fruits sometimes have to let their fruits and vegetables go to waste because the fruits and vegetables they produce are in high quantity and the number of consumption is low. So the owner wants me to make a web application that will help these people. The web application should be able to help user register to the system. It should be user-friendly, easily accessible. It should also take users message so that they can elevate the level of application to another level. Customers should be able to order the product they want to purchase and admin can add as many product as they want to provide the information about products to the customers.

List of nouns (potential class) are:

Users, Customer, Order, Farmers, Fruits, Vegetables, Product, Admin.

Final candidate class has been finalized following the rules given below:

* Removed the matching, irrelevant, synonyms and outside the scope nouns.
* Removed the architectural level action noun. Also, the noun that could be design during development were skipped.

Final Candidate class

* Customer
* Order
* Product
* Admin

## 2.7.2 Initial Class Diagram

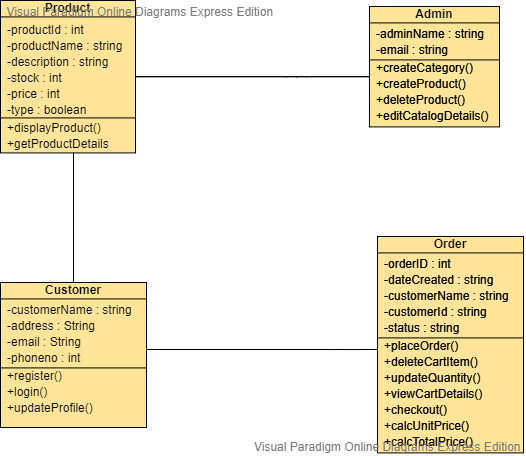


Figure 2: Initial Class Diagram

## 2.7.3 System Architecture

3-Tier system architecture has been used which holds the user interface, data storage, isolated platform and business logic. It includes client-server relation like client tier, application and database tier. It is used because all the tasks are separated which are handled by each layer in client-server application. Interface services are displayed to user in client tire whereas all the business logic is maintained in application tier. All the information are received and preserved in the data tier independently from other tiers. It also helps to keep the availability, scalability and performance of the overall system.

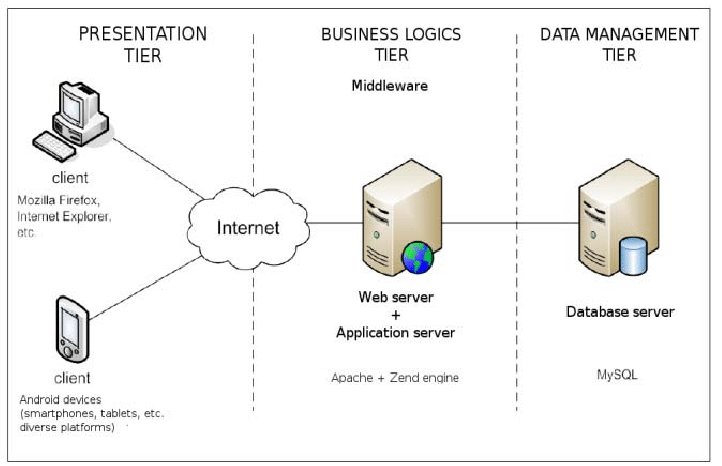


Figure 3: System Architecture