# Chapter 2 Analysis

## 2.1 Introduction

Analysis is the main phase in the lifecycle of software development, it assumes an important job when creating fresh frameworks and separate task into littler components in order to gain a superior understanding of the needs of the framework. At this point, all the prerequisites of the structure to be developed are differentiated and broken down

## 2.2 Analysis Methodologies

Analysis is the way to distinguish and characterize customers ' demands for an implementation that will be produced or altered. Analysis fuses most of the coordinating that facilitates to perceive the requirements of client. In this manner, the Analysis hopes to explore, approve and supervise the needs of the structure or programming.

Methodology of OOA includes sequence, class and activities diagram. Sequence diagram are the diagrams showing the relationship of information between classes validating and visualizing runtime scenarios. Class diagram explains the structure of the system by displaying system classes, attributes, or methods along with object relationships. Activity diagram explains the system's dynamic aspect which shows the system's flow events. These three diagrams are drawn for my project but in the design phase the illustration of the sequence diagram, class diagram and activity diagram is described.

## 2.3 Information Gathering

It's the gathering of functional, technical and system requirement from user, staff and employee. Before constructing a task, there should be a lot of exchanges that should be completed. Getting the correct requirement from the client can accomplish the projects in time. The reason for the information gathering structure is to assist arrange our association's efforts to turn out more inclusive, different methods are to be used to assemble information.

Interview: This approach is the most relevant information collection strategy where both parties will speak face to face about fulfilling the requirement. Using this approach, better info can be gathered that leads to a proper understanding of requirements and can also strengthen mutual understanding between both parties. Below is a list of the questions I asked during the interview

**Advantage:**

Interactive way for data collection.

Ideas and emotions are better understood, so there is less misunderstanding.

Enables in-depth collection of the information required.

**Disadvantage:**

Its time consuming and costly

There is probability of being bias, by both interviewer and the person giving the answer.

**SWOT**

SWOT is an important strategy for understanding characteristics and shortcomings of my project. It separates openings that are accessible to you and the risks that you face while constructing the project. I choose SWOT because it will collect the prerequisite as it is easy and useful techniques to examine the quality of the structure, its shortcomings, its openings and its hazards. In Swot we focus on the following areas:

**Strength:**

Benefit of your blood bank registration system?

What feature does your system offers compared other system in the m2.arket?

**Weakness:**

Things to avoid while developing the system?

What scion and environmental things that affects the system?

What improvement can be made in the system?

## 2.3 DEVELOPMENT METHODOLOGY

Methodology used

Since this project will have a single person in charge of evaluating, designing, developing, implementing and testing and no other teammates are to be involved, so waterfall methodology will be the best to use in this project. The project size is also tiny, so it is unlikely that project demands will alter. Methodology of waterfalls has some of these benefits over others such as agile. Some of the reason for choosing this methodology are as follow;

* Non-repetitive stages unlike agile.
* Easy and uncomplicated steps, beneficial for short project.
* Next stage only begins when the requirement of last stage is completed.
* Saves time and energy.
* Uncomplicates the project

This methodology isn’t without shortcomings some of them are as follow,

* Less room for errors.
* Going back to the previous stage is not possible
* If the requirement changes then you will have to run the stages all over again.
* No guarantee of user satisfaction as stockholders are not involved.



## 2.4 Feasibility Study

Feasibility study demonstrates if a task is conceivable to assemble or not. It examines the perceivability of a thought. This investigation decides whether the task is specialized, monetarily, socially, doable inside the assessed expense.

The various type of feasibility studies needed to be performed are as follow,

**Technical feasibility**: This evaluation center around the technical resource that are available, it chooses whether the specific portion of the association meets all the prerequisites. It also involves equipment evaluation, programming and other specific prerequisites.

**Economic feasibility:** Cost and benefit advantages inquires is tried in this feasibility. Money saving advantage inquiry gages and contrast benefits and similar expenditure plan designated for the task. The price of the assignment is explored, and it is decided whether it is compatible and conceivable to complete. It also acknowledges the economic benefits that a job will offer to the association.

**Legal feasibility:** It investigates whether the suggested framework conflicts with the legitimate necessities. Governmental problems may be the problem of the project being developed. It understands that the constitution of the country gives permission to create such a site or not.

**Operational feasibility:** It is used to differentiate the importance of certain issues and how to fathom them. It makes a step as it needs to follow that problem and undertaking an inquiry to break down and decide whether and how well the need of the association can be met by completing the assignment.

**Scheduling feasibility:** This feasibility is the most important for time accomplishment, all things considered, if not completed on timetable, a task will be bound to fail. In the event of booking, an organization assesses how much time the task will take to complete and chooses whether the undertaking will work as set out in the Gantt diagram.

## SRS (Software Requirement Specification):

The record that specifies the systems highlights such as functional and non-functional requirement is known as software requirement specification or SRS. A progressively formal procedure is required if separate gathering will develop the requirements.

### Functional requirements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FRID | Functional requirement | Data | Rational | Dependencies |
| FR1 | Registration | Valid user information with valid organization info | To login into the system |  |
| FR2 | Login | Valid ID and password | To authenticate the admin | FR1 |
| FR3 | Add donor | Information about donor and their blood type | Adding donor information | FR 2 |
| FR4 | View all the information about the Donor including blood type | Information about donor and their blood type | Viewing donor information | FR 3 |
| FR5 | Delete or update donor | Information about donor | Deleting or updating the donor and its information. | FR 3 |
| FR6 | Add campaigns | Information about campaigns and their location. | Adding campaign information | FR 2 |
| FR7 | View all the information about the campaigns including their location. | Information about campaigns and their location. | Viewing campaign information | FR 6 |
| FR8 | Delete or update  Campaign | Information about campaigns and their location. | Deleting or updating the campaign info and its Location | FR 6 |
| FR9 | Log out | N/A | Logging out of the system | FR 2 |

### 

### Non-functional requirements:

|  |  |  |
| --- | --- | --- |
|  | Non-functional requirements | Rational |
| NFR1 | Privacy | Secure confidential information. |
| NFR2 | Effectiveness | Generating user-friendly, efficient website with better usability. |
| NFR3 | Dependability | Building a web-based application that customers can rely on. |
| NFR4 | Accessibility | Developing a website that can be accessed and used at any moment. |
| NFR 5 | Interoperable | making the website readily available on many platforms |
| NFR6 | Productivity | Creating website that is simple to navigate. |
| NFR 7 | Maintenance | Providing regular adequate maintenance. |

## 2.5 MoSCoW Prioritization

It is an agile methodology for understanding the priorities properly. It can be applied for tasks and requirements.

The word MOSCOW stands for;

M: Must have

S: Should have

C: Could have

W: wont' have

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional and**  **Non-Functional Requirement** | **MoSCow** |
| FR1 | Registration | Must have |
| FR2 | Login | Must have |
| FR3 | Add donor | Must have |
| FR4 | View all the information about the Donor including blood type | Must have |
| FR5 | Delete or update donor | Should have |
| FR6 | Add campaigns | Must have |
| FR7 | View all the information about the campaigns including their location. | Must have |
| FR8 | Delete or update  Campaign | Should have |
| FR9 | Log out | Must have |
| NFR1 | Privacy | Must have |
| NFR2 | Effectiveness | Should have |
| NFR3 | Dependability | Must have |
| NFR4 | Accessibility | Should have |
| NFR 5 | Interoperable | Could have |
| NFR6 | Productivity | Must have |
| NFR 7 | Maintenance | Should have |

## 2.6 Requirement Specification

**Hardware requirements:**

**Hardware:** desktop/laptop

**Ram needed**- 4GB

**Processor used** - i3/ i5 5th generation.

**Storage**- 500GB

**Software requirement:**

**Software used:** Note++, my php admin, google chrome, Mozilla

**OS used**- Windows 8,10

**Database used**- Oracle or MySQL

## 2.7 Use Case Diagram

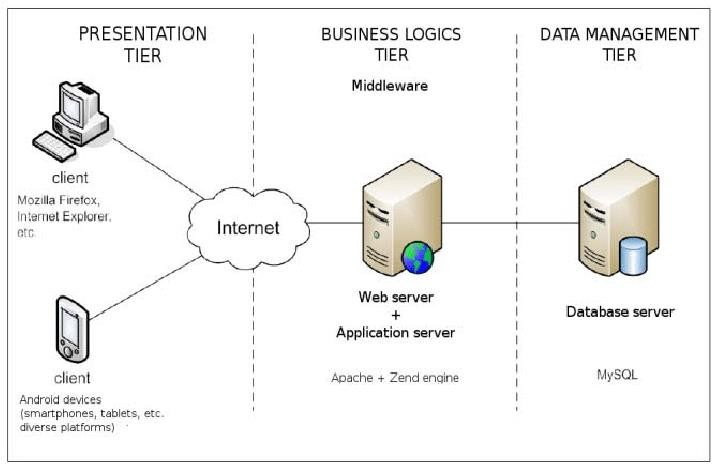
A close up of a map

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Use Case Title** | **Summery** | **Alternative sequence** | **Actor** |
| UC 1 | Register using valid organization detail | User registers using valid organization details | Error while registering is showed | User From organization 1 and 2 |
| UC 2 | Log in | User login using valid credentials | Error while message is displayed, if login info isn’t correct | User From organization 1 and 2 |
| UC 3 | Add blood donors with blood type. | User adds donor information by clicking add donor. | Error while adding is displayed if the information is invalid | User From organization 1 and 2 |
| UC 4 | View donor list from both user | User view all the donor information by clicking View donor. | Nothing is displayed | User From organization 1 and 2 |
| UC 5 | Update donor list | User updates donor information by clicking update donor | Error while Updating is displayed if the information is invalid | User From organization 1 and 2 |
| UC 6 | Delete donor list | User deletes donor information by clicking remove donor. | Deletion is unsuccessful is displayed | User From organization 1 and 2 |
| UC 7 | Add campaign with location. | User adds campaign information by clicking add campaign. | Error while adding is displayed if the information is invalid | User From organization 1 and 2 |
| UC 8 | View campaign list from both user | User view all the campaign information by clicking View donor. | Nothing is displayed | User From organization 1 and 2 |
| UC 9 | Update campaign list | User updates campaign information by clicking update donor | Error while Updating is displayed if the information is invalid | User From organization 1 and 2 |
| UC 10 | Delete campaign list | User deletes campaign information by clicking remove donor. | Deletion is unsuccessful is displayed | User From organization 1 and 2 |
| UC 11 | Logout | User logs outs by clicking log out button | Error message is displayed. | User From organization 1 and 2 |

## 2.8 System Architecture

We had a need for an architecture that efficiently distributes user / server functionality which provides increased reusability, performance, flexibility and maintenance. We therefore chose 3 tire Architecture as our program architecture in which user interface, functional process logic and system data storage are built and maintained on separate platforms as independent modules. Within client tier interface services are kept view users while all business logic is retained within application tier. All information that is kept independent of application and client tier is processed and retrieved in the data layer. Hence, this design helps my software improve its performance, availability and scalability.



## 2.9 NLA

**Scenario:**

Blood bank Nepal is NGO organization that has asked to develop a web-based application, to keep record of various donor detail and campaign detail. It must enable any individual or staff of any hospital or blood bank can keep a online record of donors with blood types and also keep record of campaigns done by their organization with their location. The main aim of this system is creating a digital record of blood donors and make the record available online so that various blood banks/hospital can share each other’s record of donors and blood type, so that they can find proper blood for an immediate patient on time. They have asked to have the following functionality,

1. Registration with valid organization detail such as Username, User\_id, number, locations and organization registration number.
2. Secure database and login-logout system
3. Add donor detail with blood type.
4. View, update and delete donor detail
5. Add Campaign detail with blood type.
6. View, update and delete campaign detail.

List of candidate noun and verbs:

|  |
| --- |
| Blood, bank, Nepal, Organization, blood, types, online, Donor, donors, proper,  Record, records, hospitals, login, add, view, delete, find, main, aim, details, registration, keep, location, ask, campaign, hospitals, details, web, donors, logout, number, username, user\_id |

Final candidate class and methods are selected using the following rules:

* Repeated nouns and verbs are removed
* Synonyms are removed
* Irrelevant nouns and verbs are removed.
* Complex meaning noun and verb are removed along with those that could be in future development.

Following the above rules following final class and method are selected.

Final class:

|  |
| --- |
| Login  Registration  Donor  Campaign |

Final Method:

|  |
| --- |
| Add  Update  Delete  Login  Logout  Username  User\_id |

2.9.1 Initial Class diagram:

A close up of text on a white background

Description automatically generated

Figure 1 initial class diagram