# **Introduction:**

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. The technique has been applied in the study of mathematics and logic since before Aristotle though analysis as a formal concept is a relatively recent development. It is the systematic study of real and complex-valued continuous functions.

# **Feasibility Study**

Feasibility study is that style of study that is usually done by developers which inaugurates whether or not the conditions are correct to start out a selected project. it's typically done to appear the whether or not the new hardware and software system are possible or not. it's done to specialize in the project and description the alternatives.

For this project, we've got studied 3 elements of the practicableness study and that they are listed and explained below:

## Economical

cargo plays in economic development and factors that condition its impact. After presenting basic empirical relationships between air cargo and both trade and gross domestic product per capita, we discuss three factors that can enhance cargo’s positive impact: air service liberalization, improving customs quality, and reducing corruption.

## Financial

We offer easy digital trade finance solutions to assist you along with your capital wants. A simplified answer which will handle each the flow of your merchandise and funding at a competitive value to assist you expand and develop your business.

## Legal

From the study, I found that the code will maintain the safety. It helps the shopper or the owner to depend on the code. Also, it doesn't have any privacy problems. this method is often employed by every kind of the users within the outlets where they need. they will allow it by collateral from the upper authorities.

# **Analysis Methodology**

For this project, I’ve got chosen the Soft System Methodology (SSM) analysis as a result of its additional people-focused analysis. it'll facilitate Pine Tree State to assemble necessities. It tries to understand and learn the matter scenario between the stakeholders instead of resolution the pre-defined issues (Anon., n.d.).

I have chosen this technique as a result of this helps America to focus additional on the peoples read rather than that specialize in system. In my read the system ought to be developed as per the purchasers would like as they are the one WHO are mistreatment the system right along. we should always recognize their read which is able to facilitate America gather additional demand for the system throughout analysis. I’ve got followed the 3 steps of the SSM likewise as done the CATWOE analysis

# **Software requirement Specification**

It is a nitty gritty portrayal of a product framework to be created with its utilitarian and non-practical necessities. The SRS is created based the understanding among client and contractual workers. It might incorporate the utilization instances of how client will collaborate with programming framework. The product prerequisite determination archive steady of every single vital necessity required for venture improvement. To build up the product framework, we ought to have clear comprehension of Software framework. To accomplish this, we have to ceaseless correspondence with clients to assemble all necessities.

Hardware requirements Specification

Operating System: Windows 7 or higher

RAM: 2GB or higher

Processor: 1 GHz or higher

Screen resolution: 1024 \* 768 resolution for better experience

# **Functional Requirements**

It is the announcement of expected capacity of a framework and its segment. It can identify with equipment, programming or both regarding computation, specialized subtleties, information control and handling or other explicit usefulness that characterizes what a framework should achieved.

Below is the function requirement which of my project

ID: FUN1

Title user Registration

Description: User should register with valid information

Rational: It helps the user to acquire log in credentials

Dependencies: N/A

ID: FUN2

Title: Login

Description: User can login with the right username and password

Rational: It helps the user to enter the system.

Dependencies: N/A

ID: FUN3

Title: Forgot Password

Description: User can change password.

Rational: It helps user to change password when they forget it.

Dependencies: N/A

ID: FUN4

Title: Add, update, delete data

Description: User can use the CRUD function

Rational: It helps them to manipulate the data.

Dependencies: FUN1

ID: FUN5

Title: View report

Description: User can view the debits and credit

Rational: It helps the user to keep the track of profit and loss

Dependencies: FUN1

ID: FUN6

Title: Logout

Description: User can logout from the system.

Rational: It helps to user to logout when they have used the system and helps in the security reason.

Dependencies: FUN1

ID: FUN7

Title: Internet for booking/reservation

Description: Customers can reserve tables in the café and restaurants.

Rational: Reserving the table in advance

Dependencies: N/A

# **Non-Functional Requirements**

In non-utilitarian testing the quality attributes of the part or framework is tried. Non-useful alludes to parts of the product that may not be identified with a particular capacity or client activity, for example, adaptability or security. E.g. What number of individuals can sign in without a moment's delay? Non-utilitarian testing is likewise performed at all dimensions like useful testing.

ID: NFUN1

Title: Performance

Description: The system should run fast without any lag.

Rational: Productivity maintained

Dependencies: N/A

ID: NFUN2

Title: Usability

Title: The software should be user friendly.

Rational: User will not feel monitors.

Dependencies: N/A

ID: NFUN3

Title: Response time

Description: The software should be quick to response to user.

Rational: rapid responding.

Dependencies: N/A

ID: NFUN4

Title: Security

Description: User with the right password and username can access data.

Rational; Data confidentiality

Dependencies: N/A

ID: NFUN5

Title: Reliability

Description: The system should be accurate and reliable.

Rational: Maintenance of reliability:

Dependencies: N/A

ID: NFUN6

Title: Availability

Description: Software should run whenever the user wants.

Rational; Upholding the availability

Dependencies: N/A

ID: NFUN7

Title: Maintainability

Description: the software can be maintaining in the future.

Rational: upgrading the system will have better experience

Dependencies: N/A

ID: NFUN8

Title: Recoverability

Description: There should be backup plan to recover the data in case there is any problem.

Rational: Data availability

Dependencies: N/A

# **MoSCoW Prioritization**

Functional Requirement

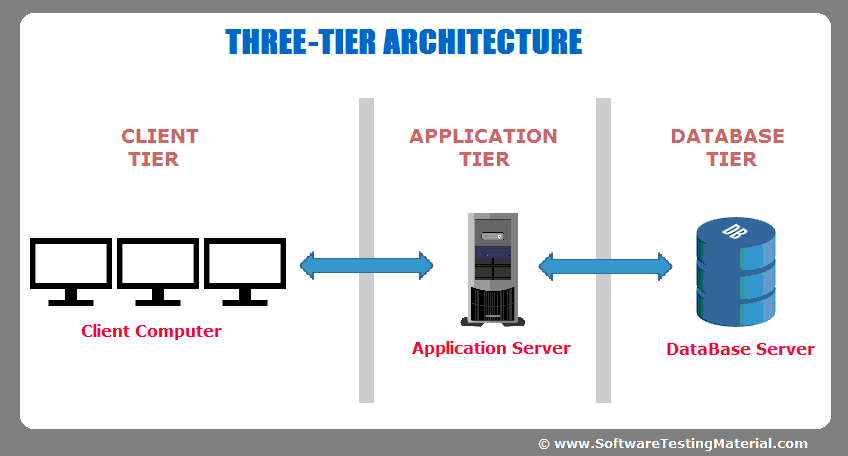
|  |  |  |
| --- | --- | --- |
| ID | Functional Requirement | MoSCoW |
| FUN1 | Registration | Must Have |
| FUN2 | Login | Must Have |
| FUN3 | Forgot Password | Could Have |
| FUN4 | Add, Update, Delete | Must Have |
| FUN5 | View Report | Should have |
| FUN6 | Logout | Must Have |
| FUN7 | Internet For Booking | Wont Have |

Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Non-Functional requirements | MoSCoW |
| NFUN1 | Performance | Could Have |
| NFUN2 | Usability | Could Have |
| NFUN3 | Response Time | Could Have |
| NFUN4 | Security | Should have |
| NFUN5 | Reliability | Should have |
| NFUN6 | Availability | Could have |
| NFUN7 | Maintainability | Could have |
| NFUN8 | Recoverability | Could Have |

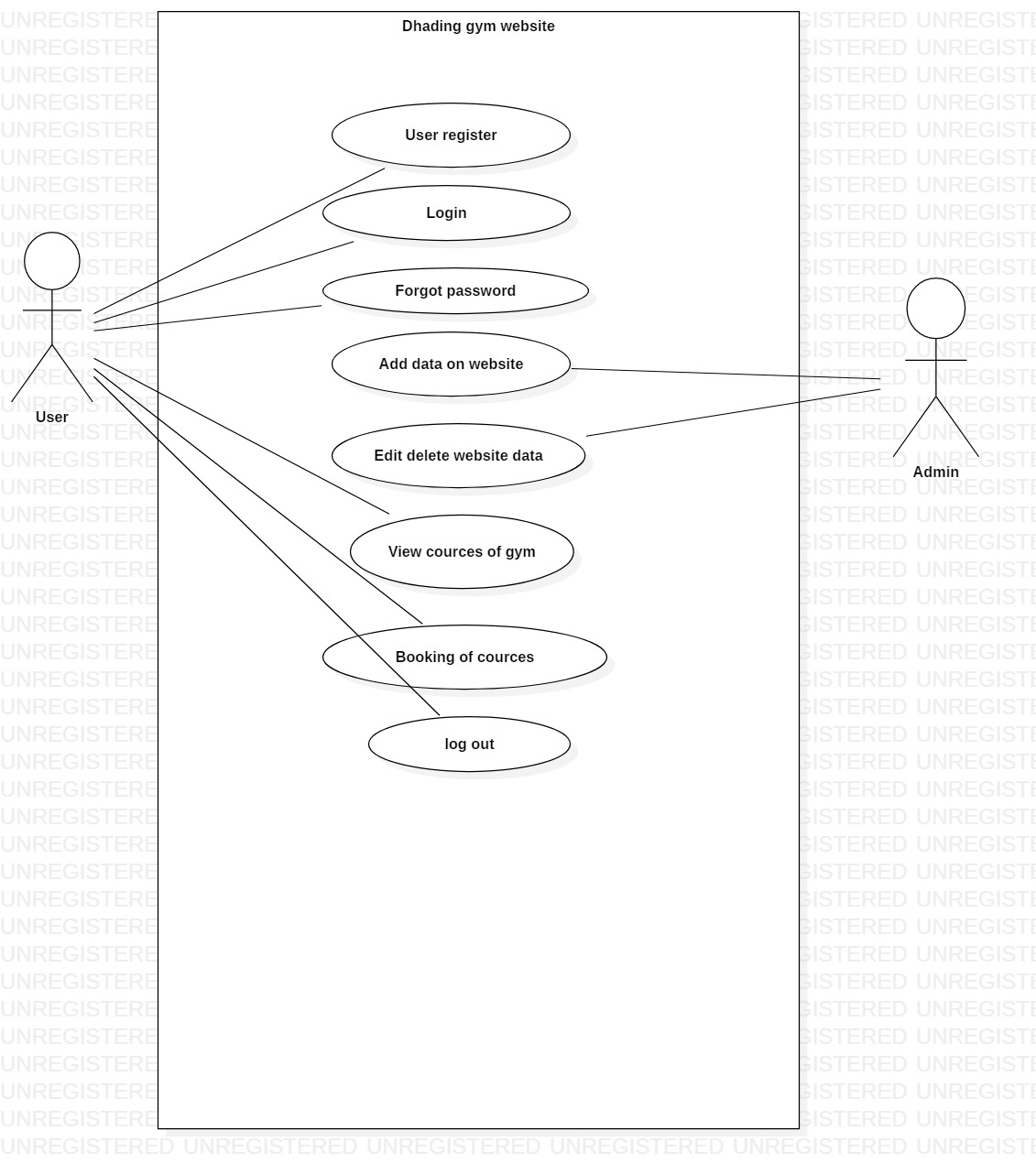
Architecture

Software architecture is the characterizing and organizing of an answer that meets specialized and operational necessities. Programming engineering improves traits including a progression of choices, for example, security, execution and reasonability. These choices eventually sway application quality, support, execution and by and large achievement.



Use case diagram

Use case is one of the dynamic behavioral diagrams of UML. It shows what the Function are in the system and who operates. Use case consist of actors, use cases, relationship between them etc. A single use case diagram captures the functionality of a system. The Purpose of using use case is to capture the dynamic aspect of a system. The use case diagram of Dhading gym website is given below:



# **NLA (Natural language analysis)**

The Scenario of the Dhading website.

For the college project I have decided to develop a website for the Dhading gym center. which will briefly describe about the Dhading gym and will show the program/ courses that are run in there.

My website contains registration system by which user can see what they have register and update the register. My website helps the people to select Courses like (Zumba dance, body building) learn about the courses and which will be beneficial for them to take.

* User can be added, deleted and updated from the system.
* Courses that are registered can also be updated or deleted form the system.

NLA (Natural language analysis)

The NLA is a process of finding out the class, methods and attributes with the help of nouns, verbs and adjective from the scenario. The short form of Dhading gym website is shown below:

Identifying classes

Selecting all the nouns from the scenario

|  |
| --- |
| College, project, website, gym, website, program, courses, user, register |

Candidate class

|  |  |
| --- | --- |
| SN. | Class |
| 1. | User |
| 2. | Courses |
| 3. | Register |

Selecting all the verbs from the scenario

|  |
| --- |
| Add user, delete user, update user, register courses, update courses, delete courses |

Class diagram of the Dhading gym website is shown below:

