# Introduction to Analysis

The Analysis Phase is where the project lifecycle begins. This is where the break down the deliverables in the Project Charter into the more detailed requirements is done. It’s also the part of the project where identification of the overall direction that the project will take through the creation of the project strategy documents takes place.

Gathering the requirement is the main attraction of this phase. For the Health and Fitness Management System, analysis of the requirement of the people, their basic need is crucial. What the requirement of the user in their perspective are, the health forum, articles, registration, login, accessibility and all the other required information are collected here.

The main purpose of analysis for Health and Fitness Management System are:

Gather all the requirement for the system

Analyze the information for the feasibility study

## Requirement Elicitation Technique:

Requirement Elicitation is the practice of collecting the requirement s of the system from users, customers and other stakeholders. This cam also be termed as Requirement Gathering. Here are few examples of the Requirement Elicitation Techniques:

Interviews

Brainstorming Sessions

Facilitated Application Specification Technique (FAST)

Quality Function Deployment (QFD)

Use Case Approach

Focus Group

Prototyping

Survey

Questionnaires

CATWOE (Customers, Actors, Transformation process, Worldview, Owners and Environmental constraints)

PEST (Political, Economic, Sociological, Technological)

Observation Social Analysis

Scenario Analysis

For the development of Health and Fitness Management System I have used the following

**Brainstorming Session:**

Brainstorming Session is intended to generate lots of new ideas. Here the imagination if the ones is let to roar to the fullest. For the Health and Fitness Management System, using Brainstorming session technique, I have collected as many as possible for the development of the system.

**Quality Function Deployment (QFD):**

This is the technique where the requirements are stated into three different categories as:

Normal Requirement: the objective and goals of the proposed software

Expected Requirement:  requirements are so obvious that the customer need not explicitly state them

Excited Requirement:  features that are beyond customer’s expectations and prove to be very satisfying when present

For the Health and Fitness Management System I have addressed the following requirements

Normal Requirements: articles regarding Healthy Lifestyle

Expected Requirements: login, sign-up

Excited Requirements: customizable workout-plans and healthy diet according to their lifestyle.

**Observation Social Analysis:**

This is the part where observation of the real life takes place. It is particle. Therefore, it enables us the ability to record and report all the finding that are true.

For Health and Fitness Management System I have observed the lifestyle of the regular person where I have found they have unhealthy lifestyle so basically now I have the idea of the workout plan and diet that could fit into their such lifestyle.

**Scenario analysis:**

Scenario analysis is a technique where different scenario with certain terms and conditions are placed in-order to gather the information.

I have kept certain health criteria such as Fat loss program, muscle mass-up program.

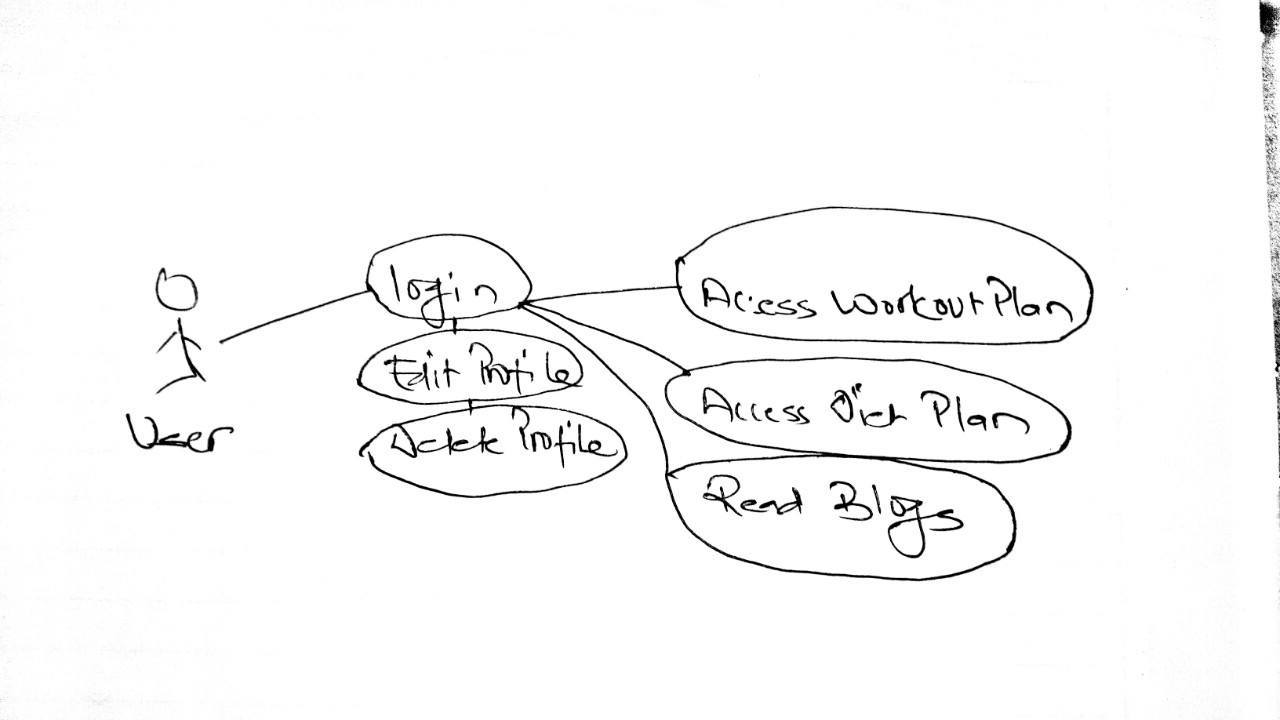
# Analysis Methodology

For the development of this project I have used Soft System Methodology i.e. Soft Approach. Soft approach methodology is where a more people focused analysis is carried out. Steps when undertaking the Soft approach are as following:

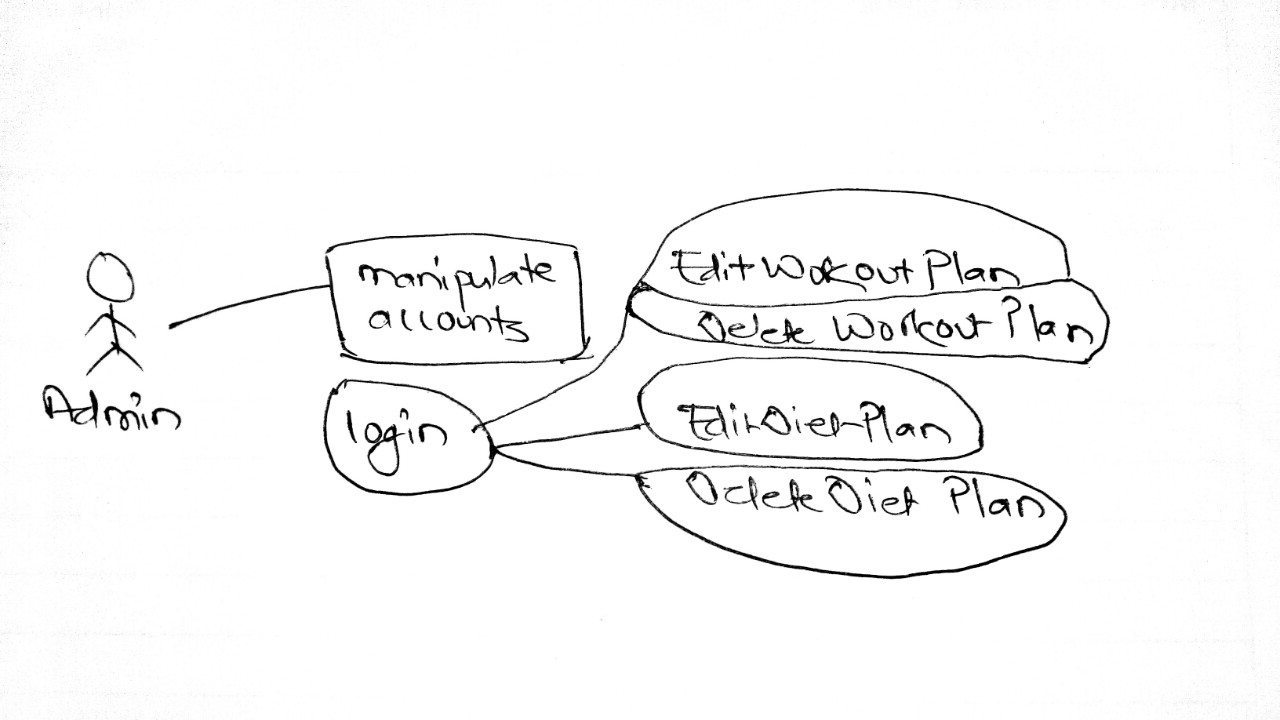
* Rich Picture
* Root Definition
* Conceptual Models

Rich Picture

This picture represents a view of the whole system and can enable better planning and understanding of a system. Here I have created a rich picture for Health and Fitness Management System:



User rich picture



Admin rich picture

## Root Definition

On defining the root definitions, we need to ask the following questions

What does the system do? (aim of the system)

How does the system do it? (means of achieving the aim)

Why is it being done? (longer term aim)

These questions are easily answered by the CATWOE analysis method.

|  |  |  |
| --- | --- | --- |
| CATWOE | Questions?? | remarks |
| Customer | Who are the customers of the system? | People with fitness goals |
| Actors/Agents | Who are the users of the system? | All the members including trainers, peoples |
| Transformation | What is transformed (changed) by the system? | This project makes easy for people to stay healthy |
| Worldview | What is the overall view of the system? | Very beneficial for everyone to adopt healthy lifestyle |
| Owners | Who owns, controls and pays for the system? | Admins of the system |
| Environment | What are the economic and/or social, political, technical and environmental constraints to the system? | Easily accessible for everyone |

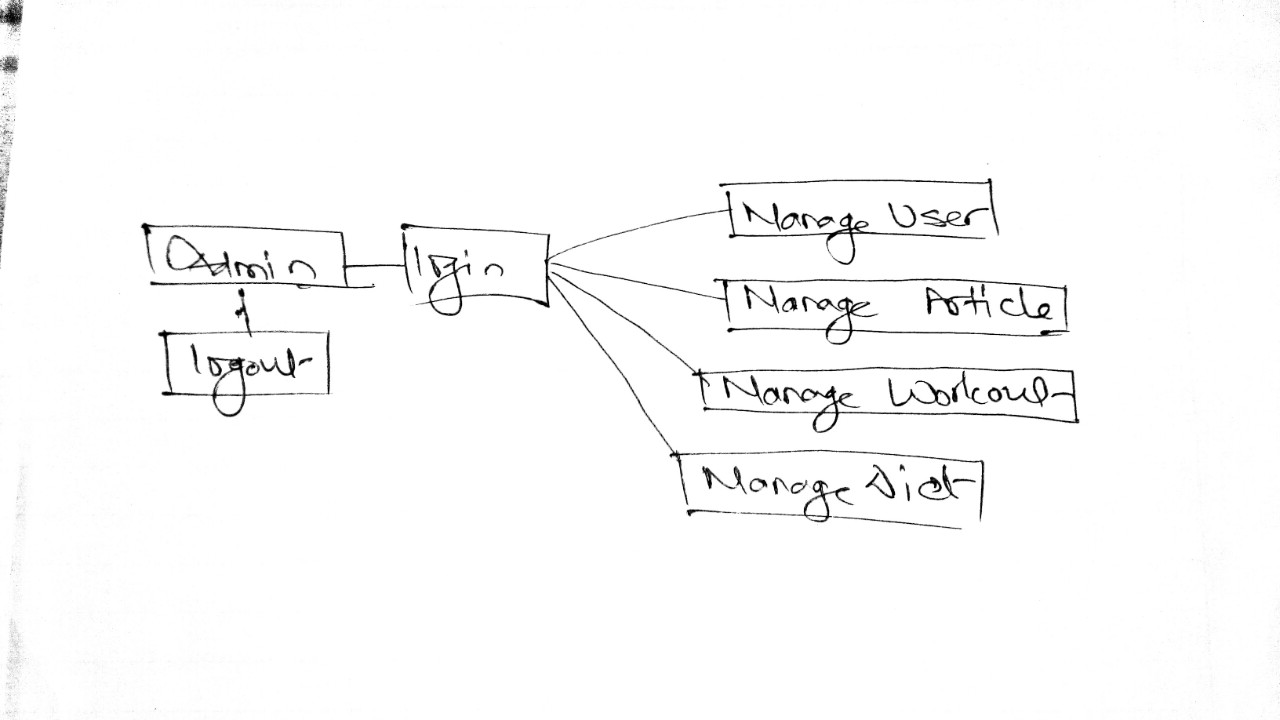
## Conceptual Model

This describes how the system should function and what activities are necessary for the processes to take place. Its basically defined by three E’s

Efficacy: will the system work and will the transformation be achieved?

Efficiency: - will it work with minimum resources?

Effectiveness: - will the system achieve its longer-term goals?



Why SSM??

I have used Soft System Methodology for the development of this project since this project focuses on the user moreover and their requirements. Other methodology such as hard approach is solely focused on technical requirement only. But SSM is more useful for the development of this project.

# 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.”

The main objective of this is to ensure that the project developed is acceptable, scalable, beneficial to the people, society.

For our Health and Fitness Management System I have conducted the following Feasibility Study:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Feasibility Study-Type** | **Purpose** | **With Aspect to Our Project** |
| 1. | Technical Feasibility | Defines and questions about the technical aspects (whether it will be able to met the requirement) | Current available resources are well enough for the completion of the project. |
| 2. | Economic Feasibility | involves a cost/ benefits analysis (helps in determine the viability, cost, and benefits associated) | The working paper work the online payment can be kind of troublesome where it requires security which is costly but can be deal in further improvements. |
| 3. | Legal Feasibility | Defines and questions whether any legal conflicts are related or not | So far, this project does not conflict with any legal requirement or any laws. |
| 4. | Operational Feasibility | Defines and questions on how the project will meet the stated requirements | In day to day life people are busy so online fitness forum can be really helpful for many. |
| 5. | Scheduling Feasibility | Defines and questions the timeframe for the completion of the project. | In the beginning I have set WBS and Gantt chart where each and every individual task are pointed. So far task is assessed accordingly. |

# Software Requirement Specification (SRS)

This is a document that captures the complete description about how the system is expected to perform. It lays out the functional and non-functional requirements.

## Functional Requirements:

This describes **WHAT** a software should do. In other words, a functional requirement will describe a particular behavior of function of the system when certain conditions are met.

The following are the Functional requirements for the Health and Fitness Management System:

**Authentication**

--The user must be able to login in-order to access the forums, articles of the healthy lifestyle. There is user validation in the system. for Authentication we have

|  |  |  |
| --- | --- | --- |
| ID | Functional Requirement | Remarks |
| FR1 | **Login form** | user must be able to login in-order to view contents |
| FR2 | **Register form** | Registration is compulsory for everyone |
| FR3 | **Edit Profile** | user are able to many necessary changes to their profile |
| FR4 | **Delete Profile** | incase of any certainty users are able to delete their profile |
| FR5 | **Edit the blogs** | Management (EDIT) of Program and related blogs |
| FR6 | **Delete the blogs** | Management (DELETE) of Program and related blogs |
| FR7 | **Manage articles** | users are able to manage their respective articles |
| FR8 | **Upload workout-plan** | access to and upload workout-plan |
| FR9 | **Delete workout-plan** | can modify their plan if needed |
| FR10 | **Edit workout plan** | can edit workout plan |
| FR11 | **Manage workout-plan** | manage their plans |
| FR12 | **Upload diet-plan** | access of diet plan |
| FR13 | **edit diet plan** | can edit diet plan |
| FR14 | **Manage diet-plan** | manage diet plan |
| FR15 | **logout** | after the completion of required task user can logout |

## Non-Functional Requirements:

Nonfunctional requirements describe how a system must behave and establish constraints of its functionality.

Here I have stated few Non-functional requirements for Health and Fitness Management System.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Non-Functional Requirements** | **Purpose** | **Remark** |
| **NFR1** | Efficiency | It is the parameter to measure the performance of the software application. | Overall performance of the system should be functional |
| **NFR2** | Quality | Quality is the overall factor contributing to non-functional requirement. A quality end product is what everyone demands | The end product should be fully functional in every aspect |
| **NFR3** | Response Time | The time taken to perform an action triggered by the user. | Responsive towards the action and command of the user |
| **NFR4** | Availability | The extent to which a system is available for use initially. If more technically defined then it can be stated as the proportion of time a system is in functioning condition. | End product is usable within the time frame |
| **NFR5** | Interoperability | Easy to use interface that should enable the user to work with any product with ease. | User-friendly interface for user |
| **NFR6** | Security | Application should not be vulnerable to any exploitation. That is, strict authentication mechanisms to be applied to the system. | Privacy should be maintained |
| **NFR7** | Fault Tolerance | A major aspect of any software product. It portrays the capability of a system to continue performing in the event of failure of some other components. | Incase, of any fault or any failure in the system, the system should not compromise on its functionality |
| **NFR8** | Documentation | Provides an overview about the system, clearly lays down the specifications in detail. | Proper written documentation is required at the end phase. |

# Hardware and Software Specification:

|  |  |
| --- | --- |
| **hardware** | **Software** |
| RAM: 2GB minimum  Processor: dual core @ 2.4 GHz (i5 or i7 Intel processor or equivalent AMD)  Hard disk: 256 GB | Operating system: Windows (7,8,8.1,10)  Browser: Google Chrome, Mozilla Firefox  Database: MySQL |

# **MoSCow Prioritization**

**Must Have**

These provide the Minimum Usable Subset (MUST) of requirements which the project guarantees to deliver. These may be defined using some of the following:

No point in delivering on target date without this; if it were not delivered, there would be no point deploying the solution on the intended date

Not legal without it

Unsafe without it

Cannot deliver a viable solution without it

**Should Have**

Should Have requirements are defined as:

Important but not vital

May be painful to leave out, but the solution is still viable

**Could Have**

Could Have requirements are defined as:

Wanted or desirable but less important

Less impact if left out (compared with a Should Have)

**Won’t Have this time**

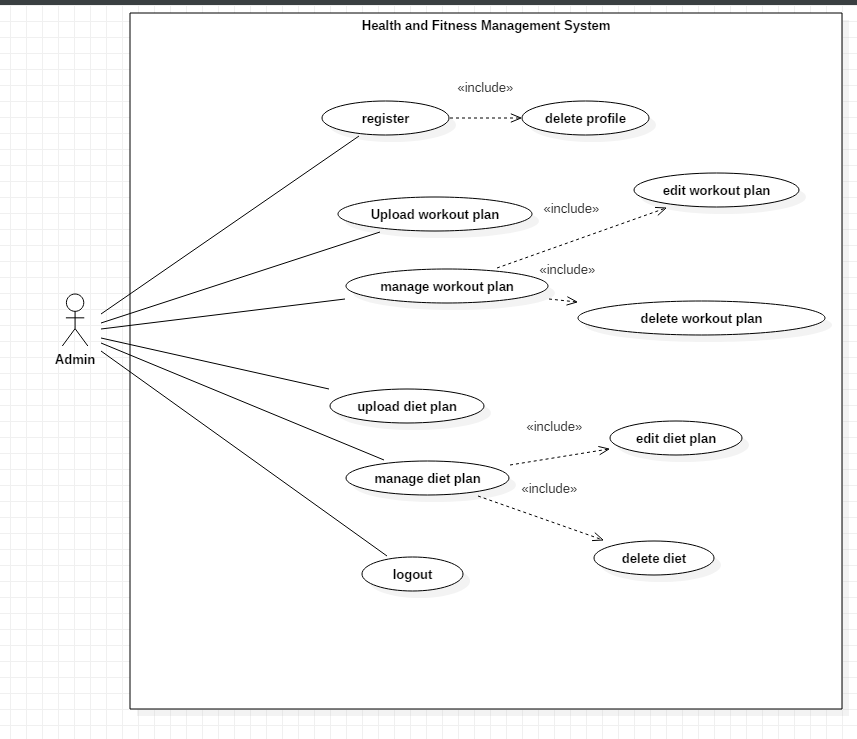
These are requirements which the project team has agreed will not be delivered (as part of this timeframe). They are recorded in the Prioritized Requirements List where they help clarify the scope of the project. This avoids them being informally reintroduced at a later date. This also helps to manage expectations that some requirements will simply not make it into the Deployed Solution, at least not this time around. Won’t Haves can be very powerful in keeping the focus at this point in time on the more important Could Haves, Should Haves and particularly the Must Haves.

## Health and Fitness Management System (MoSCow Prioritization)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Functional Requirement** | **MoSCoW** | **Remarks** |
| FR1. | Login | **MUST HAVE** | User login is required |
| FR 2. | Register | **MUST HAVE** | Registration of user is compulsory |
| FR 3. | Edit profile | **MUST HAVE** | Changes to the profile is must |
| FR 4. | Delete profile | **SHOULD HAVE** | Incase of any certainty |
| FR 5. | Edit the blog | **SHOULD HAVE** | Blog articles are optional |
| FR 6. | Delete the blog | **SHOULD HAVE** | Blog articles are optional |
| FR 7. | Manage articles | **SHOULD HAVE** | Blog articles are optional |
| FR 8. | Upload workout plan | **MUST HAVE** | Trainer are responsible for this task |
| FR 9. | Upload diet plan | **MUST HAVE** | Trainer are responsible for this task |
| FR 10. | Delete workout plan | **COULD HAVE** | Trainer are responsible for this task |
| FR 11. | Manage diet plan | **COULD HAVE** | Trainer are responsible for this task |
| FR 12. | Edit workout plan | **SHOULD HAVE** | Trainer are responsible for this task |
| FR 13. | Edit diet plan | **SHOULD HAVE** | Trainer are responsible for this task |
| FR 14. | Delete diet plan | **COULD HAVE** | Trainer are responsible for this task |
| FR 15. | logout | **MUST HAVE** | After the completion of task user must be logged out |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Non-Functional Requirement** | **MoSCoW** | **Remarks** |
| NRF1. | Efficiency | **MUST HAVE** | the performance of the software application. |
| NRF 2. | quality | **SHOULD HAVE** | overall factor contributing to non-functional requirement |
| NRF 3. | Response time | **MUST HAVE** | Minimal amount of time must be taken to perform an action triggered by the user. |
| NRF 4. | availability | **SHOULD HAVE** | The extent to which a system is available for use initially |
| NRF 5. | interoperability | **MUST HAVE** | User-Friendly interface is must |
| NRF 6. | Security | **MUST HAVE** | Security is top priority |
| NRF 7. | Fault Tolerance | **SHOULD HAVE** | the capability of a system to continue performing in the event of failure |
| NRF 8. | Documentation | **SHOULD HAVE** | Provides an overview about the system, clearly lays down the specifications in detail. |

# Use case diagram



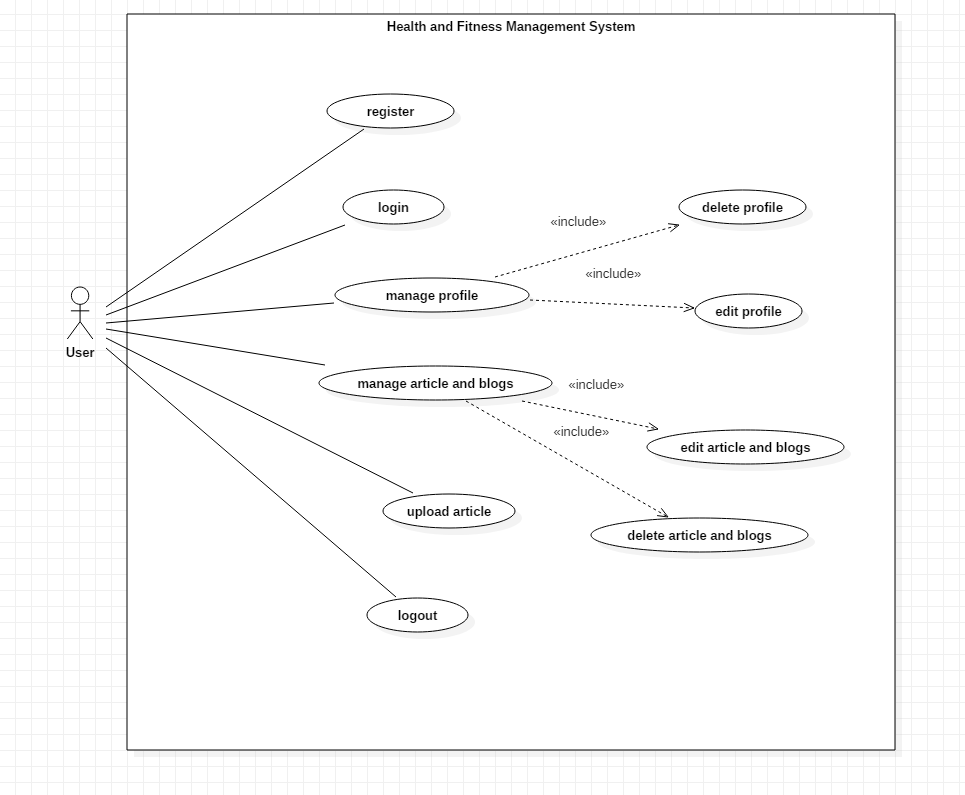
Admin-usecase diagram

Scenario:

Here I have created a working area in the aspect of the admin.

Admin is able to do the following task:

* Register user
* Upload workout plan
* Edit workout plan
* Delete workout plan
* Upload diet plan
* Edit diet plan
* Delete diet plan



User-usecase diagram

Scenario:

Here I have created the use case diagram in the aspect of the user. User is able to do the following task:

* Register
* Login
* Edit profile
* Delete profile
* Edit blogs
* Edit article
* Delete blogs
* Upload blog
* logout

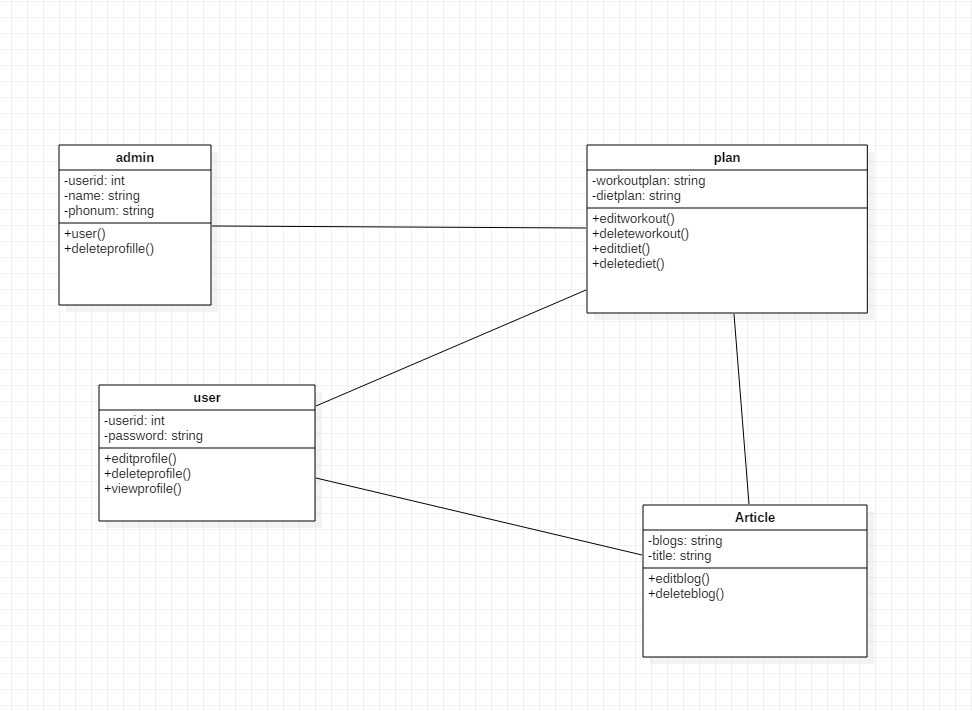
# NLA/ Initial Class Diagram

Health and Fitness Management System is a project intended for making the lifestyle of the people healthy. Adopting the healthy lifestyle with healthy eating habit. As mention in the previous task, I have collected the following data for the class diagram.

|  |  |  |
| --- | --- | --- |
| Noun (Candidate class) | Adjective (candidate attributes) | Verbs (candidate operations) |
| Admin, user,plan | Username, password, phone number | Edit, delete, add, register |

## Initial class diagram

With the above data I have created the following class diagram. This initial class diagram contains the basic set of functions of the overall system.



Initial class diagram