# 2.Analysis

# 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 Tic-Tac-Toe game, analysis of requirement of the gameplay is crucial. The internal system logic and the overall functioning of the system has to be thoroughly studied.

The main purpose of analysis for Tic-Tac-Toe game are:

Gather all the requirement for the system

Analyze the information for the feasibility study

## 2.1Requirement Elicitation Technique:

Requirement Elicitation is the practice of collecting the requirement s of the system from users, customers and other stakeholders. This can 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 Tic-Tac-Toe Game, 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 Tic-Tac-Toe Game, 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 Tic-Tac-Toe Game I have addressed the following requirements

Normal Requirements: easy user-interface

Expected Requirements: multi-player gaming

Excited Requirements: winner announce

**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 Tic-Tac-Toe Game, I have observed the regular people’s interest in the gaming. People prefer offline, simple games where there is instant gameplay.

**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 multiplayer gameplay where two friends can have fun gameplay.

# 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 Tic-Tac-Toe Game:

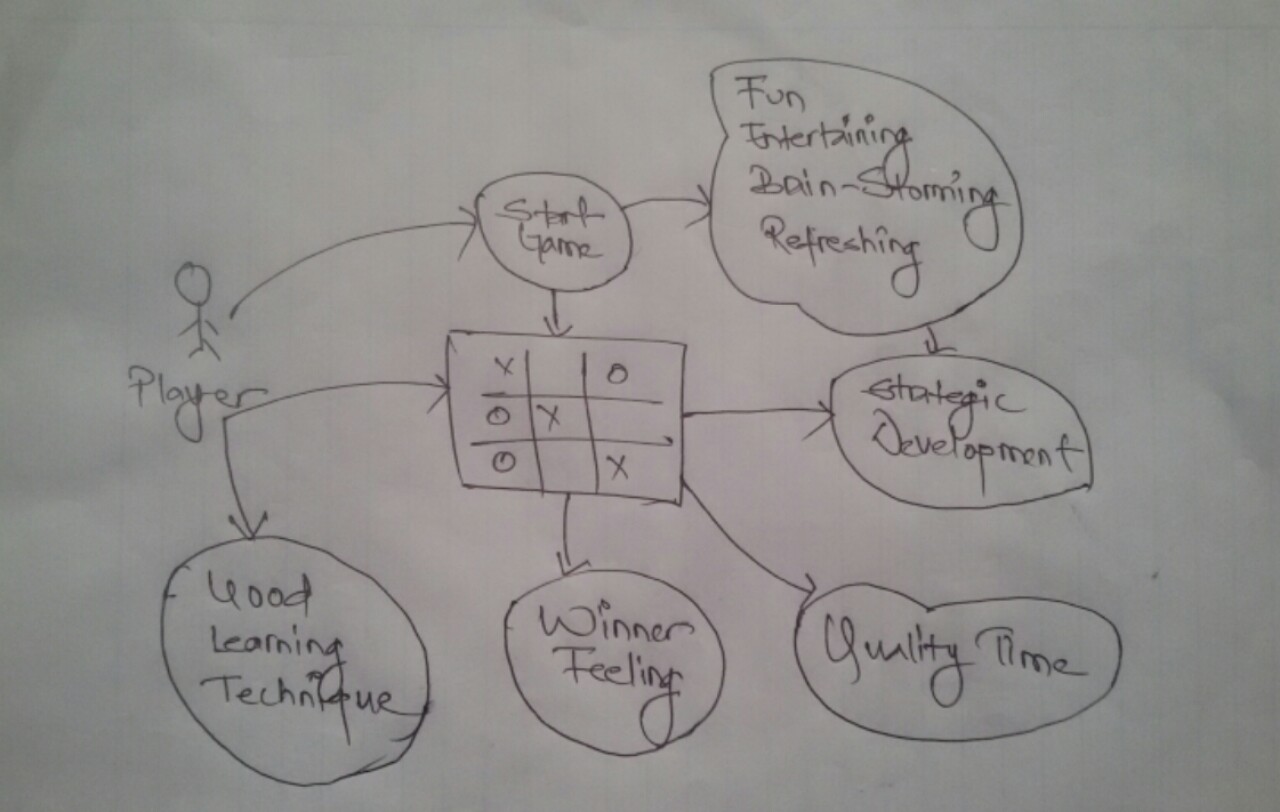


Fig: Rich Picture of Tic-Tac-Toe Game

## 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?

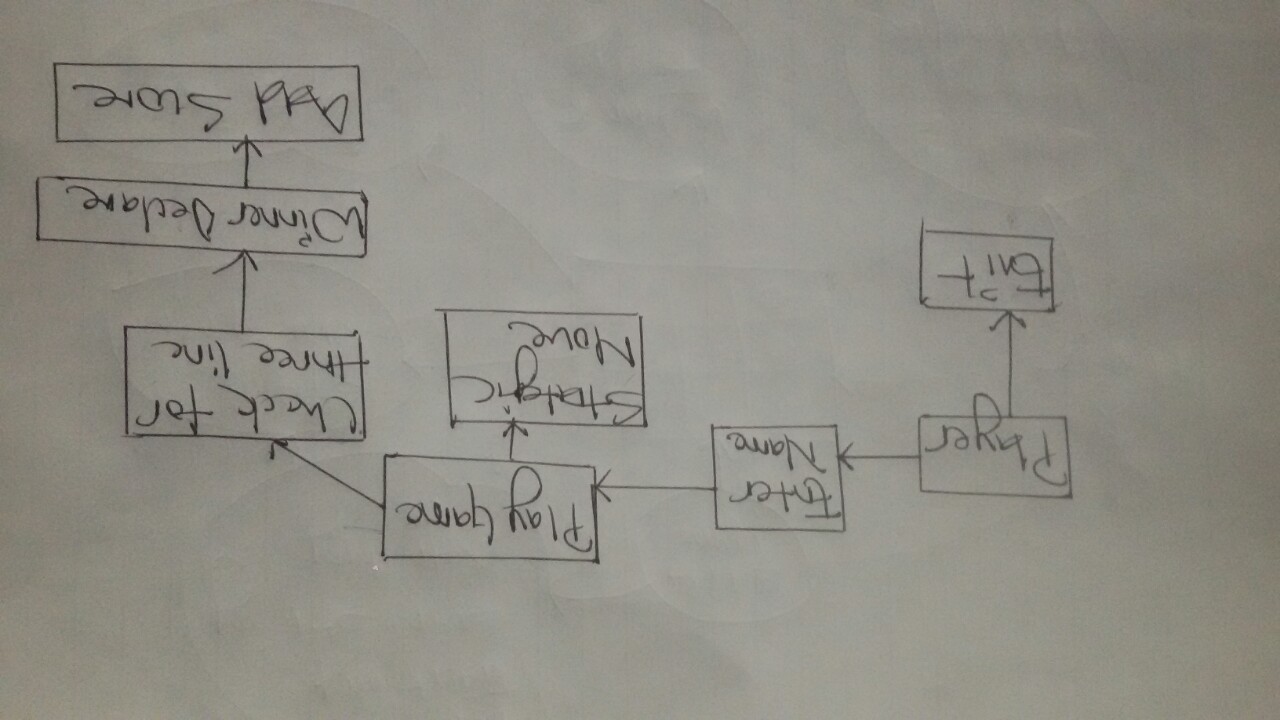


Fig: Conceptual Model Of Tic-Tac-Toe Game

# 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 Tic-Tac-Toe Game 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 Tic-Tac-Toe Game:

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Requirement** | **Remarks** |
| FR1 | **Player name** | Player must enter their name for the gameplay |
| FR2 | **Game symbols** | There are two symbols representing the gameplay |
| FR3 | **Score board** | Score boards of player 1 and player 2 are required |
| FR4 | **Reset button** | In-order to have a new gameplay, one can reset |
| FR5 | **result** | One who wins must be displayed at the end of the game |

## 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 Tic-Tac-Toe Game.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | In case, 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**



Fig: 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.

## Tic-Tac-Toe Game (MoSCow Prioritization)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Functional Requirement** | **MoSCoW** | **Remarks** |
| FR1. | Player name | **SHOULD HAVE** | Player name is not so of concern |
| FR 2. | Game symbols | **MUST HAVE** | There are only two options of symbols |
| FR 3. | Score board | **MUST HAVE** | Users should know their score |
| FR 4. | Reset button | **SHOULD HAVE** | User can either exit the gameplay and have new one |
| FR 5. | result | **MUST HAVE** | Winner must be notified at the end |

|  |  |  |  |
| --- | --- | --- | --- |
| **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

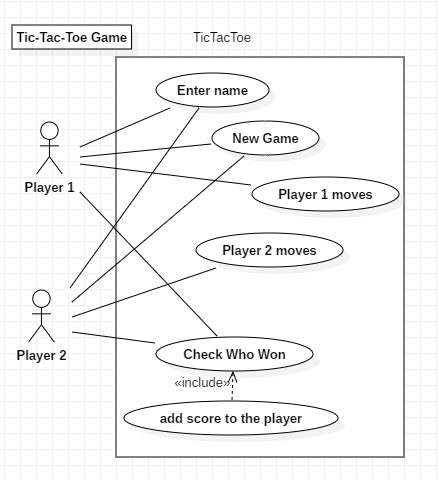


Fig: Tic-Tac-Toe Use case

Scenario:

Here for the Tic-Tac-Toe game, I have designed the Use case diagram. Here I have showed the necessary operations that goes in the system, with focus on what user is able to perform the task.

# NLA/ Initial Class Diagram

Tic-Tac-Toe game is a fun game developed for entertainment purpose. As per the above information, 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.

# CONCLUSION

Here in this way, I have done all the necessary analysis required for the project which includes many things like Usecase diagram, Moscow prioritization, analysis meythodology.