# **Chapter 2: Analysis**

## **2.1 Introduction**

Successful project required analysis which must be done in order to meet user’s requirement and desirable functionality as the needs of the stakeholders. Analysis refers to the process of breaking down the project in order to gain knowledge regarding of better understanding of the users need, project aim and objectives. So for that, we should study and research all the components that is included in the project in order to solve the upcoming problems and list the significant features and characteristics of the product.

## **Analysis Methodology**

In the initiative phase of analysis, every project needs appropriate methodology. There are different types of analysis methodology which comes in practice in day to day project. Some of them are Soft-system approach, Hard-system approach, Combined system approach, Object-oriented analysis and Organization-oriented analysis. There are many approach which we can follow to accomplish or assigned task. So for that, Soft-system approach will be the suitable one for our project.

There are plenty of methodology which have already been listed in the above paragraph. Among different methodology, we prefer Soft-system approach. It is so because due to following reasons;

1. This approach is people-focused analysis which can help for better understanding of our project needs.
2. It keeps the users at the center point which is even called as user-centric. This helps to get immediate feedback from the stakeholders and works to give their desire outcome.
3. It took some times but will clear the view of our project.

### **Rich Picture**

A rich picture is the pictorial representation of a situation which illustrates the main element and relationships that needs to be considered in order to create some improvement. Likewise, the rich picture consists of pictures, text, symbols and icons which are used to illustrate graphical representation of the particular situation. Thus, it is the best practice which must be done in the pre-analysis phase.

**Root definition**

It represented as the structured description of the system. This phase is practically done after accomplished a rich picture. The practice helps to clarify what the system process and how to sought out the problems. It also helps to describe and explain the required functionality or aims of the system.

Thus the above rich pictures can use to reflect the ***aim*** and ways about ***how does our system do the task***. Besides that, it will help to identify, ***what sorts of works can be done in the system***?

The main aim of our system is to provide the platform where users can check out nearby futsal and book it as the desirable time by the users. While in content aspect, the application provides the genuine list of the futsal which is provided by the futsal owner in purpose of booming their futsal business. Moreover, users can read all the genuine sports news from the trusted channel through online sports news ***API (Application Programming Interface).***

Conceptual Model

## **Flexibility Study**

It is the practice which is done in the initial design stage of any project which illustrate all the required information and indicate if a project is viable or not. In simple term, it means that whether the develop application can generate an equal or higher rate of return from the markets during its lifetime. Besides that, it will check the way of measuring various things like estimated time, allocated budget and others factor too. There are couples of times of feasibility studies that are performed in order to find out the likelihood of a project. But not all the feasibility studies are relevant in every project. Some of the pertinent feasibility studies due to its significance are explained below

* **Financial feasibility**

The crucial stage of the analysis is to estimate the project cost and make sure that it does not exceed the limited budget. So that, there always be some extra revenue in order to make some good sum of money.

* **Technical feasibility**

It is the complete understanding or an act done by researching and fully study of the project in terms of input, output, processes including program or procedure. This sort of study will help to identify whether current hardware or a software can handle the system. For our case, doing the sorts of study, we come to know that our system has more than adequate to support hardware or software processes.

* **Legal feasibility**

Legal feasibility is the complete study which is done in order to know if the future project conform the both requirement or not i.e. ***Legal*** or ***ethical*** ***requirements***. It is important that our project must be in track by following both the requirements which we have mention in above statement. Likewise, it is important to start a project including project licenses, certificates, copyrights, health and safety and many more.

* **Social/Cultural feasibility**

It is the complete study of the project to determine whether the developing project does not hurt the people emotionally or sentimentally or maybe associate with any ethnic group, cast or a culture. So before learning and implementing it our project we will be able to reduce the chances of hurting the sentimental of the people or maybe the group of a culture, race or ethnic.

* **Time/Scheduling feasibility**

The project always need to be accomplished in the fix time or an interval which have already been decided in the initial phase of the project. If the project seems not to be completed in the given time the others resources and budget must be allocated so that the project must be completed within the deadline. Thus, even from the recent proposal we have visualized the milestone of the completion of the project too.

## **S.R.S (Software Requirement Specification)**

The abbreviation of SRS stands for ***(Software Requirement Specification).*** It is a document which content entire description of the system and even learn about how the system is expected to perform. This consists of many qualities and some of them are

* Correct
* Complete or Consistent
* Verifiable
* Modifiable
* Traceable
* Unambiguous

There are two different requirements which we can use under ***S.R.S*** i.e. ***Functional requirement*** and ***Non-functional requirements***.

### **Functional Requirement**

The definition of ‘a functional requirement’ is that it basically state the functionality which show what the system should do. Typically, the main role of this is to specify a behavior or function. All the required behavior and function must have in order to achieve the goals of the project. Some of the functional requirement examples in our project are “add futsal” and “booking invoice”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **R.ID** | **R.Name** | **Data** | **Rational** | **Dependency** | **Remarks** |
| FA1 | User Signup/Registration | Fullname: String  Email: String  Uername: String  Password: String  DOB: date  Gender: String  Phone: String | To register User as an Audience | - | - |
| FA2 | User login | Username: String  Password: String | To let users for searching futsal and booking it | FA1 | - |
| FA3 | Update Users | Fullname: String  Email: String  Password: String  DOB: date  Gender: String  Phone: String | To let users for update their personal details except username | FA1 | - |
| FA4 | Add Futsal | Futsalname: String  Details: String  Address: String  Facilities: String  Image: String  Contactno: String  Price/Cost: int | To add futsal for the users to checkout nearby futsal details | - | Critical and admin are only allowed to add this |
| FA5 | Update Futsal | Image: String  Contactno: String  Facilities: String  Price/Cost: int | To update futsal attributes | FA4 | - |
| FA6 | Delete Futsal | Futsalname: String  Details: String  Address: String  Facilities: String  Image: String  Contactno: String  Price/Rentalrate: int | To delete recorded futsal | FA4 | - |
| FA7 | Add Review | Review: String | To add review of the futsal | - | - |
| FA8 | Edit Review | Review: String | To edit the added review | FA7 | - |
| FA9 | Delete Review | Review: String | To remove the added review | FA7 | - |
| FA10 | Add Sport News | Title: String  Description: String  Postedtime: String  Source: String  Image: String | To post sports news for users to make application more useful | - | Use different free distributed **API** and manually written news |
| FA11 | Update Sport News | Title: String  Description: String  Postedtime: String  Source: String  Image: String | To update news content | FA10 | - |
| FA11 | Delete Sport News | Title: String  Description: String  Postedtime: String  Source: String  Image: String | To delete old and unwanted news | FA10 | - |
| FA12 | Futsal Booking | Fullname: String  Email: String  Phone: String  Futsalname: String  Bookeddate: date  Bookfor.date: date  Rentalrate/Price: int | Create reservation system for users to book futsal | - | - |
| FA13 | Cancel Booking | Fullname: String  Email: String  Phone: String  Futsalname: String  Bookeddate: date  Bookfor.date: date  Rentalrate/Price: int | Remove booking within certain time | FA12 | Not critical - users can cancel their booking |

The functional requirement in the above tabular are highly priorities which must be implement in the project. Since, the term APIs and using of different libraries are used in the project, we do need some times for research to learn about them. It directly helps our project by increasing it usability and authentication of the users.

### **Non-Functional Requirement**

Non-functional requirement state about what the system should do. In other word, it specifies how the system should behave which is constraint upon the system behavior. Likewise, Non-functional requirement focuses on ***quality characteristics*** or ***quality attributes*** for of a system. It covers all the remaining requirements which are not covered in the above ***Functional requirement.***

There are some non-functional requirements which is listed below

* Performance
* Security
* Reliability
* Maintainability
* Availability and many more

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R.ID** | **R.Name** | **Rational** | **Dependency** | **Remarks** |
| NFR A1 | Smooth system interaction | To make smooth and lag free system | - | The developing system must be smooth in use and lag free |
| NFR A2 | Proper validation and security mechanism | To make system less vulnerable and secure | - | The system must be secure and appropriate input are only allowed |
| NFR A3 | Reliable and Dependable | To make the system more reliable, trustful and dependable | NFR A2 | The system must be effective and the work must be accomplished at time as the desire of the time. |
| NFR A4 | System must be repaired or enhanced | To make system more useful and bug free | - | The system must be bug free and helps to enhance the use of it. |
| NFR A5 | Ease to use | To make the system user-friendly | NFR A1, NFR A2, NFR A4 | The system must be easy to use even for non-technical users which must be user-friendly |
| NFR A6 | Scalability | To let system easily to handle any number of users | NFR A1, NFR A3 | System should be able to handle any number of users |
| NFR A7 | Maintainable | Make system to easily identify bugs and update friendly | NFR A1, NFR A2, NFR A5 | System must be easier to find bugs and maintain it |
| NFR A8 | Accuracy | To check the functionality so that it provide accurate result | NFR A1, NFR A2, NFR A6, NFR A7 | System should give accurate result |
| NFR A9 | Testing | To make system easily testable and bug free | NFR A1, NFR A2, NFR A6, NFR A7 | System testing should be done |

### **2.4.3 Moscow Prioritization**

Simply, Prioritization refers to the action of deciding the comparative significance or urgency of a thing or things in order of its. M0SCow is the widely using prioritization technique which helps to pin the important task, features and implement it in the real work by their level of prioritization. As our discussion, the MoSCow prioritization is divided into following categories.

* **Most have**
* **Should have**
* **Could have**
* **Won’t have**

We implement it into the tasks as their level of prioritization. The demonstration of the prioritization is discussed in the below table.

|  |  |  |
| --- | --- | --- |
| **Functional** | | |
| **RID** | **R.N** | **MoSCow** |
| FA1 | User Signup/Registration | Must have |
| FA2 | User login | Must have |
| FA3 | Update Users | Should have |
| FA4 | Add Futsal | Must have |
| FA5 | Update Futsal | Should have |
| FA6 | Delete Futsal | Should have |
| FA7 | Add Review | Must have |
| FA8 | Edit Review | Should have |
| FA9 | Delete Review | Should have |
| FA10 | Add Sport News | Must have |
| FA11 | Update Sport News | Should have |
| FA11 | Delete Sport News | Should have |
| FA12 | Futsal Booking | Must have |
| FA13 | Cancel Booking | Must have |

|  |  |  |
| --- | --- | --- |
| **Non Functional** | | |
| **RID** | **R.N** | **MoSCow** |
| NFR A1 | Smooth system interaction | Must have |
| NFR A2 | Proper validation and security mechanism | Should have |
| NFR A3 | Reliable and Dependable | Could have |
| NFR A4 | System must be repaired or enhanced | Should have |
| NFR A5 | Ease to use | Must have |
| NFR A6 | Scalability | Could have |
| NFR A7 | Maintainable | Should have |
| NFR A8 | Accuracy | Must have |
| NFR A9 | Testing | Should have |

So above date can be seen in the set of tabular form where it has given a prioritization to a fixed task and now we can be able to work as there given prioritization.

### **2.4.3 Hardware/Software Specification**

Every project needs to identify appropriate hardware and software specification as the needs of the system. Computers do have different specification regarding to the needs and the works of the users. Almost every individual have set their own computing configuration as their requirement so that it could fulfilled the desirable and gain some value from them. Nowadays, both component i.e. Hardware and software are closely coupled to provide agile integration as well as to quickly harvest data and utilize it as for actionable intelligence. Thus, our application is an android application so we already have to figure out what sorts of hardware and software **i.e.** O.S (Operating System) is required to run our android application. So for that, we need to learn about required hardware specification including software specification too.

|  |  |  |  |
| --- | --- | --- | --- |
| **System Requirement** | | | |
| **Minimum** | | **Recommended** | |
| **OS** | Android – version Lollipop | | Android (64 Bit) – version above Lolipop |
| **RAM** | 2GHz dual core processor | | 2GHz dual core processor |
| **Storage** | 1 GB | | 2 GB |
| **Others** | Google map support needed | | Browser and google map is required |

## **Use case diagram**

It is the primary form of system/software requirements for a new software program under developed. Here, use case diagram help to specify the expected behavior i.e. What, and not the exact method of making it happen i.e. How. Likewise, the use case diagram is usually denoted by both visual and textual representation such as UML. Moreover, the main advantage of using use case diagram is to helps us design a system from end users’ perspective. There are some elements used to create the diagram. They are listed below

1. It helps to show the relationships between actors, use cases and systems.
2. It does not show sequence order which helps to achieve the goals of each use case.

## **Initial Class Diagram (NLA)**

NLA stands for Natural class diagram. It is the simple i.e. non-technical technique which is used to identify classes, attributes and the function of a system from certain scenario. There always be the scenario where we have a role to identify all the required information and gather and extract them as the needs of the users. So in simple terms, NLA helps to easily identify the important classes, attributes and the functions.

#### **Scenario**

Sport fanatic is an online android application which deals with various futsal. It is a free distribute android application which provides a platform for both futsal owners to boom their business while helps user to search the nearby futsal. It is a platform for all futsal lovers where they can check out nearby futsal and search for the best futsal to gain more facilities and more services in less cost. Likewise, it will be the first android application in the market which will cover the issues of futsal booking and reading sports news. So in my opinion, not having such type of application in the market helps to gain it aims to be the dominant system in the market.

Here, in our scenario, we need 2 different users. The first users are a simple user who prefer the application in order to check out the futsal and track it location and for booking purposes. While the second type of users is called as an admin. He must have a privilege of adding futsal and updating application by posting sports news through manual. Likewise, API can still be used to generate automated sports news from trusted sources like BBCSPORTS, CNNSPORTS and many more. Moreover, the users can do various sorts of activities in application which even consists of reviewing and rating. Thus for all this, the users must be logged in in-order to gain and see the nearby futsal. Likewise, futsal can be view using location search through addresses, and by its name too.

|  |
| --- |
| Name |
| Email |
| Username |
| Password |
| Phone number |
| Location |
| Futsal Name |
| Futsal location |
| Futsal phone number |
| Rental |
| Facilities |
| Sports news |
| Sources |
| Time |

**Finding out the classes, Attributes and Methods**

|  |  |  |
| --- | --- | --- |
| **Nouns** | **Adjective** | **Verbs** |
| Sport fanatic | Name | Add |
| System | Type | Update |
| Futsal List | Categories | Delete |
| Platform | Email | View |
| Futsal | Date | Post |
| Opinion | Normal |  |
| Review | Particular |  |
| Audience | Generic |  |

Note: From the above tabular form, we remove all the repeating/vague from them

|  |  |  |
| --- | --- | --- |
| **Nouns** | **Adjective** | **Verbs** |
| Sport fanatic (Irrelevant) | Name | Add |
| System (Irrelevant) | Type | Update |
| Futsal List | Categories | Delete |
| Platform (Irrelevant and vague) | Email | View |
| Futsal | Date | Post |
| Opinion | Normal |  |
| Review (Normal review) | Particular |  |
| Audience (Duplicate with users) | Generic |  |

Note: From the above table, we can get the following NLA things

|  |  |  |
| --- | --- | --- |
| Nouns | Adjective | Verbs |
| Futsal | Name | Add |
| Review | Type | Update |
| User | Email | Delete |

## **References**

* <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>
* https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/