**Chapter 2**

**2.1 Introduction to analysis**

Any detailed examination of anything complex in order to understand nature or to determine its essential features is called analysis **(Merriam-webster.com, 2019)**. In other words, it is the process of breaking down complex thing into small parts that can be understand easily. For any project it is important to understand what is happening and how the things will work. This all is done through a process called analysis. It is an important phase in Software development life cycle. In this phase the requirements are captured accordingly, in this phase a customer/client expresses what he wants the system to do in order to fulfill his requirements. Technical person or the person from developer team analyse each requirement provided by the users/clients and make sure that the requirements can be fulfilled in the software in the system without affecting its system functionality and without causing any problems.

Analysis is done for following reasons:

**Studying the current system**

Any project is initiated because there might be problem with the existing system. In order to make the current system efficient and make it free from errors and bugs new system needs to be designed. Analysis will help in collecting the facts from existing users also it will help to collect information about current system boundary, details of the system and the people affected by the system.

**Defining new system objectives**

Analysis will help to prioritize user requirements. It will give a clear idea of what the system should be and will make the developer team (Technical person) clear about the needs of user/client from the system. Analysis will help in understanding each and every aspects of current system and indicate how the things can be made efficient with the deployment of new system.

**2.2. Analysis Methodology**

The literal meaning of methodology is theoretical, systematic study of the methods and ways applied to any fields of study. Analysis refers to studying and methodology refers to the systematic approach, so combing these, analysis methodology means the methods of studying the system by help of various resources that will help in making the study efficient and less time-consuming. There are mainly 9 analysis methodology that can be used in studying the system. Among them I will use Soft System Methodology. Soft approach involves seven steps.

* First step is to find out problematic situation. Problems faced are recorded from staffs with the help of Interviews, Questionnaire etc.
* Second step is to express problem situation, rich picture is used to express problem situation.
* Third step is to derive root definitions of relevant systems. Root definition is a definition of purpose of the system of the human-activity. CATWOE framework is used to define root definition.
* Fourth step is to derive conceptual model. In this step an ideal diagram that could solve the most of the problem is shown.
* Fifth step is to compare conceptual model with the real world.
* Sixth step is to analyse feasible and desirable change. In this step we will understand how various activities are performed. And also, only those changes are made which is feasible.
* Seventh step is to take action. The conceptual diagram with feasible changes is implemented.

**Advantages**

* It helps to structure a complex organisational and political situations.
* It also focuses on user involvement rather than technical specification.
* It allows developer to roll back and repeat the phages if necessary.

**Disadvantages**

* It has no definite technique modelling tool, so it is hard to know about the correctness of diagrams.
* Since no specific rules and process are followed it is impossible to track the progress.
* This can sometime take too long time to reach an agreement, so there is chance that the software will not get delivered.

**2.3. Feasibility study**

It is a study which involves an estimation of the level of proficiency mandatory for a project that can provide qualitative, qualitive assessments of various other resources, identification of mandatory points, general timetable and general cost estimation. **(My Accounting Course, 2019)**. This type of study helps for determining the possibility of an idea. And also, it helps in confirming that the project is legally, technically, socially, economically feasible. It also gives an idea on whether a project is worth the investment. Mainly there are five types of feasibility that should be done:

1. Technical feasibility: It gives a clear idea of capability of the developer/technical team of any organisation.
2. Economic feasibility
3. Legal feasibility
4. Operational feasibility
5. Scheduling feasibility.

**2.4. Software Requirement Specification**

Software requirement specification is an explanation of system that is to be developed. SRS helps to lay out non-functional and functional requirement and this involves a set of use cases that explains user interaction that the software must have for the user satisfaction. It helps in maintaining an agreement between client and developer team on how the software will be and how a software should function. It helps in providing a realistic estimation on product costs, risks, schedules etc. It can help to prevent software project failure by identifying risk in early phase and the ways to mitigate them. Software Requirement specification helps to minimize the efforts and time that developers require a specific goal and also help in minimizing development cost.

Software requirement specification serves other purposes. They are listed below:

* It provides feedback which guarantee to the client that the developer team understand the problems and issues that needs to be solved and the behavior of the system that are necessary to address those problems.
* It helps to make agreement between user and the developer, and also it helps user to determine whether the stated requirements are fulfilled.
* It helps in determining the requirements of system which further helps in rough estimation of time and cost.

**2.4.1 Functional Requirement**

It is a document which provides list of the operations and task or activities that the system must be able to perform. Functional requirements should include description of data to be entered into the system, operations performed by each interface, system reports or other outputs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Title | Description | Rational | Dependencies |
| FR-001 | Login into system | Admin as well as normal users can login with the credentials provided | Authentication of username and password | FR-002 |
| FR-002 | Registration | Users can register themselves in the system by providing necessary credentials |  |  |
| FR-003 | View Staffs | Users can view staffs according to different category |  | FR-001 |
| FR-004 | Update Details | Users can update their credentials after being logged in to the system. |  | FR-001 |
| FR-005 | Delete Account | Users can delete their account and take break from the system if they want |  | FR-001 |
| FR-006 | View Individual Staffs account | User can check individual staffs accounts that are added in the system and get some information |  | FR-001 |
| FR-007 | Rate staffs | User can give stars to individual staffs by going on their profile |  | FR-001 |
| FR-008 | Give Feedback | Users can log into the system, select the staff and give feedback |  | FR-001 |
| FR-009 | View other important notices | Users once logged in the system can view notices posted by the admin. |  | FR-001 |
| FR-010 | Post Query | Users can post query in community forum |  | FR-001 |
| FR-011 | Reply To posted Queries | Users can reply to others query too. |  | FR-001 |

Functional requirements for admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Title | Description | Rational | Dependencies |
| FR-012 | Login into system | Admin can login to the system by entering username and password | Authentication of Username and password |  |
| FR-013 | Update User details | Admin can update user details if required |  | FR-012 |
| FR-014 | Delete User | Admin can also remove user from the system if he/she violates the rules. |  | FR-012 |
| FR-015 | Add Staffs | Admin can add staffs according to various category |  | FR-012 |
| FR-016 | Remove Staffs | Admin can remove staffs if necessary |  | FR-012 |
| FR-017 | View Feedback | Admin can view feedbacks posted by various users. |  | FR-012 |
| FR-018 | Add notices | Admin can add notice so that the user can view them |  | FR-012 |
| FR-019 | View graphs of individual staffs | Admin can view rating graphs of various staffs. |  | FR-012 |

**Non-functional requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Title | Description | Rational | Dependencies |
| NF-001 | Security | Data of users as well as the system must be kept secure from theft | Conserving security | NF-003 |
| NF-002 | Availability | System should not be down. It should function 24\*7 | Conserving availability | NF-003 |
| NF-003 | Reliability | System must be reliable. User input and output must be handled properly | Data loss and incorrect data must be avoided | NF-001, NF-002, NF-005 |
| NF-004 | Testability | System must be tested thoroughly | Maintain Testability and correctness of software |  |
| NF-005 | Scalability | It should fit in any conditions. It should function even if users of system grow rapidly and in large number. | For expansion of business | NF-003 |
| NF-006 | Maintainability | System components must be maintained easily. | Solving small problems and fixing bugs. | NF-001 |
| NF-007 | Serviceability | Help service should be provided by developer team if necessary. | Helping users to solve the problems that they are facing. |  |
| NF-008 | Performance | Software should be optimized and less resources should be used. | Helps to boost performance and response time. | NF-001 |
| NF-009 | Recoverability | Data should be recovered in case of accidental deletion and damage. | Ensuring availability | NF-003 |
| NF-010 | Interoperability | Data are allowed for unrestricted sharing between different system. | To share information and resources. |  |

**Mosque prioritization**

In a DSDM project time is fixed. It is very necessary to understand the importance of work that needs to be done in specific order to meet deadlines. Moscow is a tool or technique that helps to understand and manage priorities in which following letters stand for:

* **M**ust Have
* **S**hould Have
* **C**ould Have
* **W**on’t Have this time

Prioritization of requirement of Feedback management system is given below:

|  |  |  |
| --- | --- | --- |
| S.N. | Functional Requirements | MOSCOW |
| FR-001 | Login into system | Must Have |
| FR-002 | Registration | Must Have |
| FR-003 | View Staffs | Must Have |
| FR-004 | Update Details | Should Have |
| FR-005 | Delete Account | Should Have |
| FR-006 | View Individual Staffs account | Should have |
| FR-007 | Rate staffs | Must have |
| FR-008 | Give Feedback | Must have |
| FR-009 | View other important notices | Should have |
| FR-010 | Post Query | Should have |
| FR-011 | Reply To posted Queries | Should have |

MOSCOW prioritization for Admin

|  |  |  |
| --- | --- | --- |
| S.N. | Functional Requirements | MOSCOW |
| FR-012 | Login into system | Must have |
| FR-013 | Update User details | Should have |
| FR-014 | Delete User | Must have |
| FR-015 | Add Staffs | Must have |
| FR-016 | Remove Staffs | Should have |
| FR-017 | View Feedback | Must have |
| FR-018 | Add notices | Must have |
| FR-019 | View graphs of individual staffs | Should have |

**Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| S.N. | Non-functional requirements | MOSCOW |
| NF-001 | Security | Must have |
| NF-002 | Availability | Should have |
| NF-003 | Reliability | Should have |
| NF-004 | Testability | Must have |
| NF-005 | Scalability | Should have |
| NF-006 | Maintainability | Must have |
| NF-007 | Serviceability | Won’t have |
| NF-008 | Performance | Won’t have |
| NF-009 | Recoverability | Won’t have |
| NF-010 | Interoperability | Won’t have |

**2.4.4. Hardware Software Specification**

**2.5. Use case Diagram**