# Chapter 2 – Analysis

# 2.1 Introduction of Analysis

Analysis is an important step for software project as it is critical in the success and failure of the system. Analysis is done to achieve and create a quality software that meets the user’s needs within budget and on time. There are four types of activities involved in analysis such as:

* Eliciting requirements: It is also known as requirements gathering and it communicates with customers and users to find what their requirements are.
* Analyzing requirements: Gathered requirements are determined and checked whether it is unclear, incomplete, ambiguous or contradictory and then resolving these issues.
* Requirements modelling: Requirements are documented using different formats or various forms, such as user stories, use cases, natural language documents or process specification.
* Review and Retrospective: During team meeting or review team members reflect on what happened in the iteration and identifies solution for improvement going forward.

Requirement Analysis is an important step of software development. Analysis activities is done to ensure that the product has all the functions and features required by the client. Requirements are gathered through user needs and document is created after using an analysis technique. Analysis ensures that the final product gives what the user demands rather than trying to mold user expectations to fit the requirements.

There are different types of analysis methodology and the methodology that I will be using for this project is Soft system methodology (SSM). SSM focuses more on people or users than technical during analysis. SSM consist of different analysis methods among them CATWOE is one of the analysis techniques which helps in root definition. Root definition is a step of SSM which terms the system or process in an organized way, making it clears who perform what task, for what purpose. CATWOE analysis is a tool to prepare a comprehensive root definition model. CATWOE can be used for any project and organization but done when user needs should be considered for better quality and enhance productivity. All the factors involved internally or externally in an organization can be analyzed using this method. It encourages open discussion of problems, perceptions and needs, joint problem solving as well as user participation and commitment. (CATWOE Analysis, 2019)

A close up of a device

Description automatically generated

Figure 1: CATWOE Analysis

CATWOE stands for

* Customers/Clients: Clients are the receiver of the outputs from the system. For this system they are asked questions about the problem with current system and how they will feel or react about proposed system.
* Actors/Agents: Agents/Actors are the user that undertake the activities involved with the system such as employee for this system who will use it. Following questions are asked for actors such as how they feel about proposed system and what are the impact of new system on them.
* Transformation: Transformation refers to the changes that occurs on data or processes after the development of a system. This step is carried out for this project by listing the inputs and finding the nature of changes it will go through while producing output.
* Worldview: For, this project the surroundings such as what is going on in and outside of the organization that may influence the development of this system is considered. This step involves engaging the process or system under analysis in its wider context to highlight the consequences or relevance of such process to the overall system.
* Owners: The person who owns an organization is also involved in an analysis during this system development. We need to consider what role they will play in analysis and will they help in analysis. Owner need to be considered for the system as they can stop the project and decide whether to go ahead with changes or not.
* Environment: The analysis is to find about the external constraints under which this system works, and which might hamper or restrict the development or changes to the system. Usually political, legal, economic, social, technological factor e.t.c are considered for their impact on system development.

# 2.2 Feasibility Study

Feasibility Study means the study about this project or its idea to find out whether it is legally, technically, socially, economically feasible or not to develop. It helps us study whether this project is worth the investment or not. Its an initial design stage of project which shows or indicate if a project is possible or not. (Feasibility study, 2019) Different factors or types of feasibility are studied to measure the likelihood and ability of completing a project successfully.

Feasibility Study types:

1. Technical Feasibility: Technical feasibility refers to the feasibility or availability of the hardware and software to complete the proposed project. For this project the technology such as hardware resources, programming software and other all software tools are available to complete it without hindrance. So, this project is technologically feasible to complete it.
2. Economic Feasibility: Economic feasibility is known as financial or cost evaluation studies to find whether the project is economically feasible or not. This project is economically feasible as the cost does not extend from the estimated budget. The cost was estimated for things such as electricity, internet for research, hardware or software components e.t.c.
3. Operational Feasibility: Operational feasibility refers to the study of proposed software whether it will be fully operational after development or not. This project is operationally feasible as the current manual system will be an automated user-friendly system. With the required training the users will be able to operate it easily. Usability will be high as this system has high operational feasibility.
4. Legal Feasibility: Legal feasibility is about the software being developed whether it is legally available to develop or not. This project is legally feasible as there is nothing unethical about it. The proposed system is legal, and it is not against the law or rules of our country.
5. Schedule Feasibility: It refers to the task that will be needed to be completed on different schedule as proposed. This project is feasible as the task for software development are sub-divided into small task and time and days is assigned for each task. So, completing each task in their deadline helps achieving schedule feasibility for this project.

# 2.3 Requirement Analysis

## Functional Requirements

Functional Requirements are the specific functions and behaviors of the system. It specifies the things system perform. Functional Requirements document are prepared for the user and they should be able to understand it without any technical knowledge. Functional Requirements shows outlines of work flows performed by the system, include functions performed by specific screens and other organization requirements it must meet. The functional requirements of this system are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** |
| FR1 | Register User | User should be registered to log in the system | To create user profile and access system | N/A |
| FR2 | User login | User login is needed for the security of system | To allow access and secure the system | FR1 |
| FR3 | Update User | User can update profile and change password | To allow user to edit their details | FR2 |
| FR4 | Remove User | User can remove their account | To allow user to delete their account | FR2 |
| FR5 | Add Employee | Employee details can be added | To allow employee information to be added | FR2 |
| FR6 | Update and Remove Employee | User can update and delete employee details | To edit, delete employee details | FR5 |
| FR7 | Add Product | Product can be added | To allow user to add product sock | FR2 |
| FR8 | Update and Remove Product | User can update and delete product details | To edit, delete product or stock details | FR7 |
| FR9 | Generate Bills | Bills can be generated for customer | To allow customer bills to be generated | FR2 |
| FR10 | Add Customer | Customer or sales information can be added | To allow sales details to be added | FR2 |
| FR11 | Update and Remove Customer | Customer or sales information can be edited and removed | To edit and delete customer or sales details | FR10 |
| FR12 | View Report | Generating sales, stock and employee report | To generate sales, stock report | FR5, FR7, FR10 |
| FR13 | View Data | Inserted data of employee, customer and stock can be viewed | To allow user to view inserted data | FR5, FR7, FR10 |

## Non-Functional Requirements

Non-Functional Requirements refers to the indirect supporting features of the system. It covers all the other requirements that is not included in functional requirements. It describes how the system works and specifies how the system should behave. It is a constraint upon the systems behavior and known as quality attributes of a system. The non-functional requirements for this system are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** |
| NF1 | Security | System should be secured, and unauthorized access not allowed. | Security is needed to secure stored data | NF2 |
| NF2 | Performance | Output of system should be fast, and system should not lag or stuck | Performance helps complete task faster and gives faster response time | N/A |
| NF3 | Reliability | Output of system should be precise or correct and data integrity is needed | Performance should be fast with reliable output | NF2 |
| NF4 | Usability | System should be easy to use and understand. Non-technical user should also feel easy | Accessibility should also be included for greater usability | NF3 |
| NF5 | Scalability | The system should be able to store any amount of data given. | To store data inserted by users without any system or memory issue | NR1, NR2, NR4 |
| NF6 | Portability | System should be cross platform that is run on different platform | To increase portability different platform should be supported by system | NR2, NR3, NR4 |
| NF7 | Maintainability | System should be maintained regularly | To prevent errors and remove bugs system should be maintained | NR4 |

## MOSCOW Prioritization

## 

MoSCoW is the popular requirements managing prioritization technique and it is also known as MoSCow analysis. This method is used to identify or prioritize the user stories gathered according to the importance or team vote. It was initially designed as a prioritization framework for time boxed projects and DSDM. It stands for four categories known as must have, should have, could have, won’t have.

Must have: It is used for prioritizing a basic and non-negotiable requirement of the project or system. It prioritizes the requirements without which the system might not function as needed.

Should have: It is below the must have for prioritization and important but not vital as must have. It tells about requirements without which system function properly but if added adds significant value to the system.

Could have: It is also known as nice to have function. If left out it will not impact on core function and have smaller impact on outcome but if such requirements are added it creates a significant system.

Won’t have: It tells about the requirements which are irrelevant for the current system or user stories that are not to be a priority for this current time frame. It helps listing the functions that will not be in the system.

The MoSCoW prioritization of this system are as follows:

|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **MoSCow** |
| FR1 | Register User | Must have |
| FR2 | User login | Must have |
| FR3 | Update User | Must have |
| FR4 | Remove User | Must have |
| FR5 | Add Employee | Must have |
| FR6 | Update and Remove Employee | Should have |
| FR7 | Add Product | Must have |
| FR8 | Update and Remove Product | Should have |
| FR9 | Generate Bills | Must have |
| FR10 | Add Customer | Must have |
| FR11 | Update and Remove Customer | Should have |
| FR12 | View Report | Could have |
| FR13 | View Data | Should have |
| NF1 | Security | Must have |
| NF2 | Performance | Should have |
| NF3 | Reliability | Should have |
| NF4 | Usability | Could have |
| NF5 | Scalability | Should have |
| NF6 | Portability | Could have |
| NF7 | Maintainability | Could have |

## System Requirement Specification (SRS)

SRS is an important information for user as it tells about the specific hardware or system that is needed to run the developed system. SRS gives information about minimum or maximum requirements for the software to run properly. It is a document that describes the functions and features of a system as well. Every software should have requirement specification so that user knows the required hardware and software components that will be needed to run the software.

System Requirements

|  |  |
| --- | --- |
| Operating System | Windows Vista/7/8/8.1/10 |
| System Type | 32-bit or 64-bit |
| Processor | Intel Core 2 Duo or higher |
| RAM | 2 GB or higher |
| Programming Language Used | C# |
| Database | Microsoft SQL Server |

# 2.4 Use Case

Use case is a part of analysis which identify, clarify and organize system requirements. It is also known as diagram representing high level interactions with a system. It defines an interaction between system and user. It is important because it defines the roles of different users in a system. It is a behavioral diagram that represents functionality using different notations such as Actor, Use-case and association. The use case diagram of the Mobile Shop Management system created using Star UML is shown below.

A close up of text on a white background

Description automatically generated

Figure 2: Use Case Diagram for Mobile MS

# 2.5 Class Diagram

Class Diagram is also the part of analysis that defines the structure of a system. It is known as structural diagram in UML as it shows system’s classes, their attributes, methods and relationship among class. For, creating class diagrams there are other steps that’s need to be completed.

Natural Language Analysis (NLA) is an analysis process to identify the class, attributes and methods of a system. For, performing NLA we need a scenario about the organization or business as shown below.

**Scenario:**

The system is about the mobile shop which manage the product stock, customer sales and employee information. It's a desktop-based application which stores data onto a database located on the same device. The system will be managed or handled by an employee or owner and they are the user or admin of the system. The system has the login system that ensures the security as the employee that has registered can only use it. The registered user can edit and remove their details as well. The user will record the information about an employee, and they are able to edit and delete the data as well. The main task of the system is keeping track of product or stock of the shop, the product and its detail will be added, edited and removed by the user. Other features include creating bills for customers and storing their information as well. User can also edit and remove the customer information. The inserted data can also be viewed by user and they can search for a report as well.

The repeated nouns, verbs and adjectives are refined and below table shows the candidate classes, attributes and operations.

|  |  |  |
| --- | --- | --- |
| **List of Candidate Classes (noun)** | **List of Candidate Attributes (Adjective)** | **List of Candidate operations (Verb)** |
| System, mobile, shop, product, customer, employee, application, database, owner, user, admin, bills, report | stock, sales, task, store, inserted | Manage product, information, edit, remove, record, delete, track, add, create, view, search |

In, the next step the synonyms and out of context words are removed for choosing possible class, attributes and operations.

|  |  |  |
| --- | --- | --- |
| **List of possible class(noun)** | **List of possible attributes (Adjective)** | **List of possible operations (Verb)** |
| User, employee, customer, product, bill | Stock, sales, store | Create, Add, edit, remove, track, view, search |

A close up of text on a white background

Description automatically generated

Figure 3: Class Diagram for Mobile shop MS