# Chapter 2: Analysis

## 2.1: Introduction to Analysis

An analysis is a process of determining the user needs and expectation for the new or upgraded products. These features are called requirements. In this process, a systematic examination and evaluation of information or data are performed to discover the important components to build the system.

The analysis is the first phase of the software development life cycle (SDLC). In this software development conceptual model, analysis focuses on the following parts:

* Gather, analyze, and ratify the information.
* Define the requirements and prototypes for new system.
* Evaluate the alternatives and prioritize the requirements.
* Examine the needs of end-user and enhance to meet the system goal.
* Prepare the Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system.

## 2.2: Analysis Methodology

The project uses Object-Oriented Analysis and Design Methodology. It is a technical method of analyzing and designing a system based on their object models. An object is an instance of anything that represents a real-world object and has all the same types of characteristics (properties), behavior (methods), and states (data). This methodology not only focuses on processes or data of the system but outlook the system as a collection of object that can interact with each other to accomplish tasks.

Object-Oriented Analysis and Design (OOAD) often include stages i.e. requirements, planning, design, coding, testing, deployment, and maintenance. These stages are similar to the waterfall SDLC and does not require additional tasks for the project as the requirement are well defined. That’s why I have decided to use OOAD for this project.

In the Object-oriented Analysis, we undertake the following tasks as mentioned below:

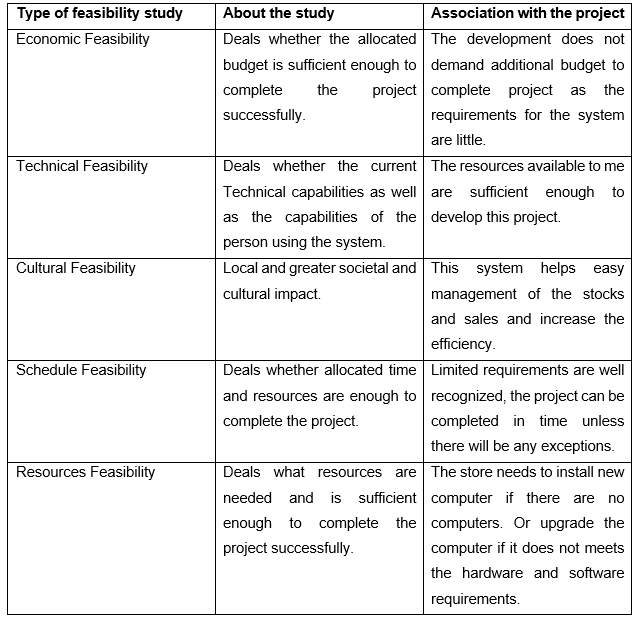
1) Elicit requirements: Define what the problem the system is trying to solve, what the system needs to perform.

2) Specify requirements: Describe the requirements i.e. use cases or user stories.

3) Model: Identify the important objects, their relationships and functionality/behavior.

## 2.3: Feasibility Study

The feasibility study deals whether the project’s practical extent that can be performed successfully. Basically, feasibility study is performed to determine whether the solution to a problem is practical in real world scenario. There are different types of feasibility studies, i.e.:

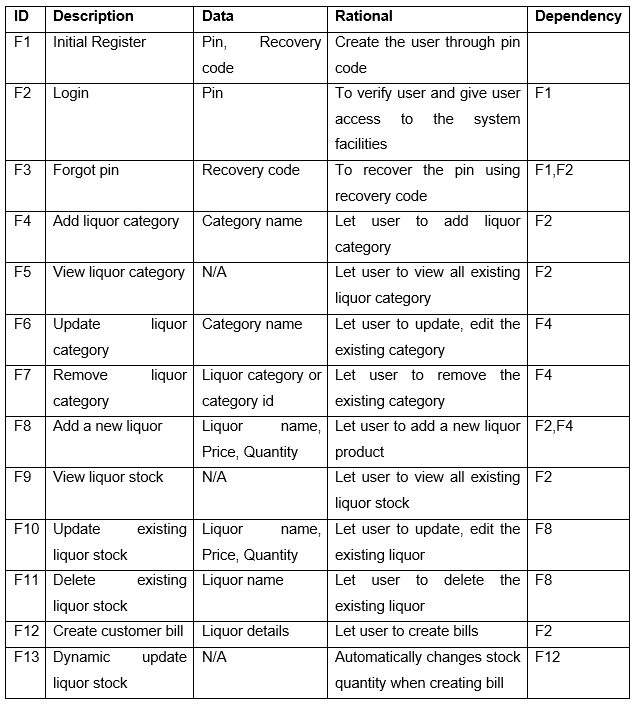


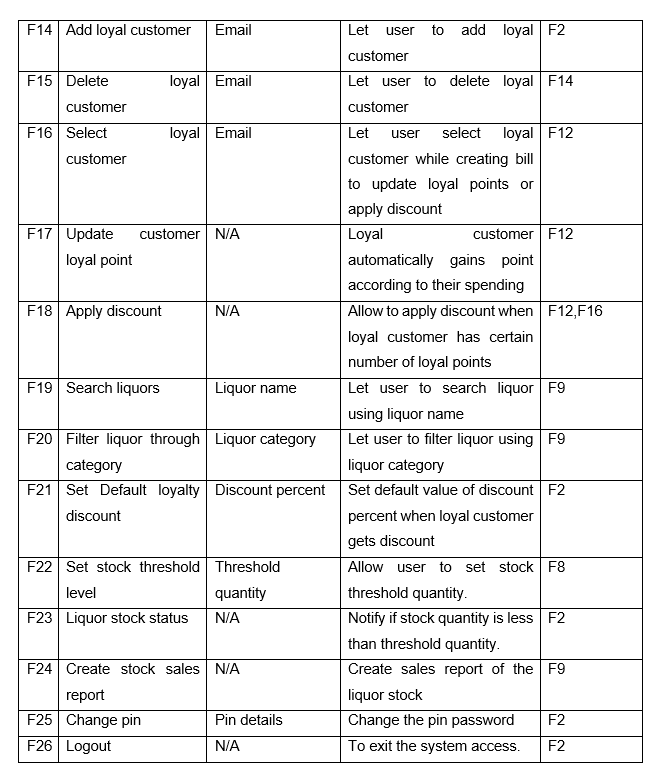
## 2.4: Requirement Analysis

The requirement analysis is the process of gathering, analyzing and defining the technical requirements of the users for the system.

## 2.4.1: Functional Requirement

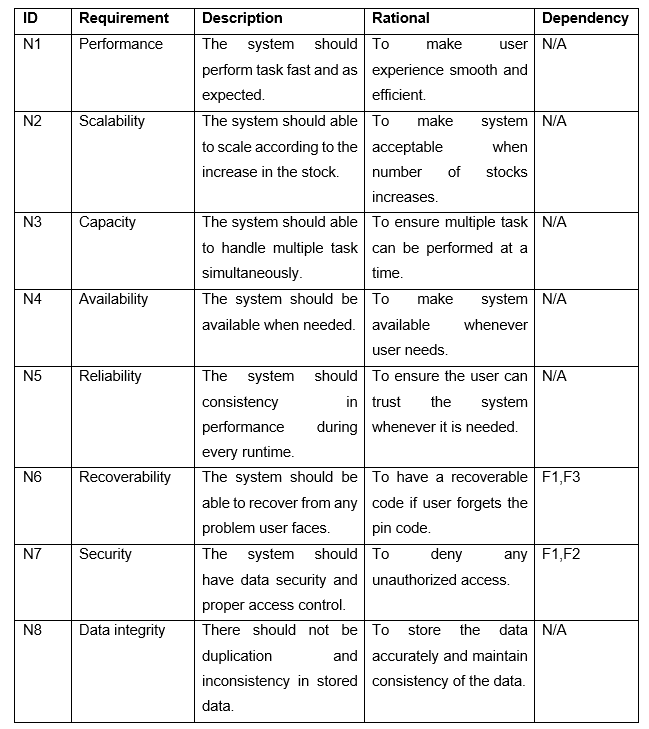
The functional requirements generally defines tasks or processes of what a system should do.

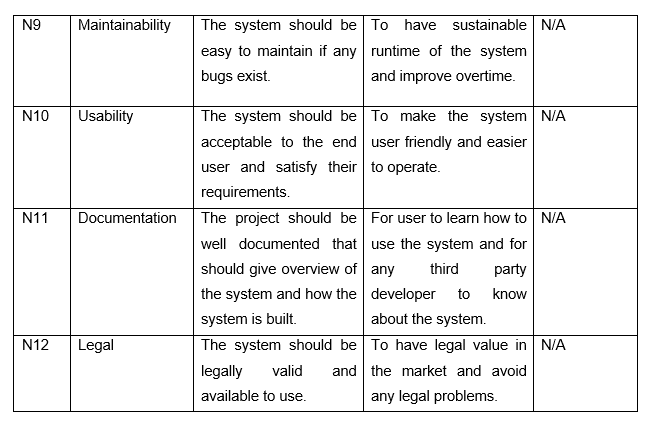




## 2.4.2: Non-functional Requirement

The non-functional requirements are important features that defines the quality of a system. It covers all the remaining requirements which are not covered by the functional requirements.

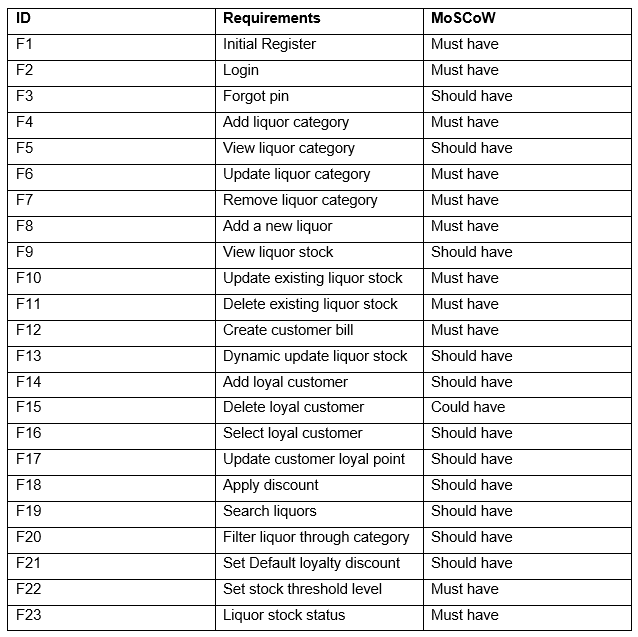


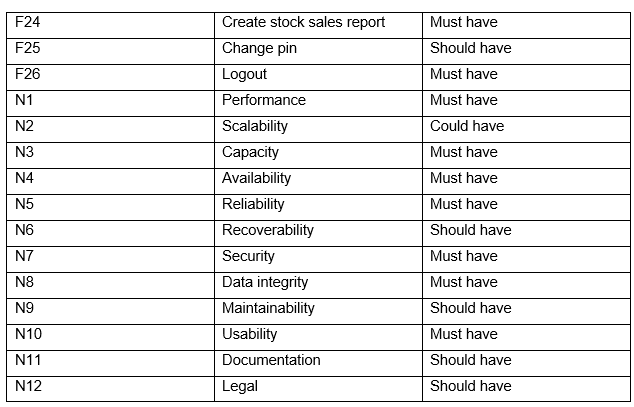


## 2.4.3: MOSCOW Prioritization

MoSCoW is a prioritization technique for assisting to understand and manage the priorities of the project. After gathering the requirements, they are divided based on their priorities to help stakeholders understand the importance of each requirement. The letters stands for

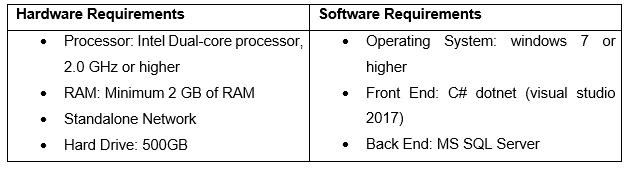
* Must Have
* Should Have
* Could Have
* Won’t Have this time





## 2.4.4: Software Requirement Specification

A Software Requirement Specification is a document, which specifies the software, hardware, functional, and network requirements of a system.



## 2.5: Use Case Diagram

The use case a diagram is a graphical representation used by an analyst to plan the boundaries of the business system that currently being analyzed, the expectations of the system and the potential user of the system.

It provides a high-level view of the solutions of the business problem and allows the analyst and designer to consider multiple implementation strategies. It helps to present the project scope to all the stakeholders. It allows people involved in the project to understand better about system processes and workflow. That’s why the use case diagram is made for this project analysis.

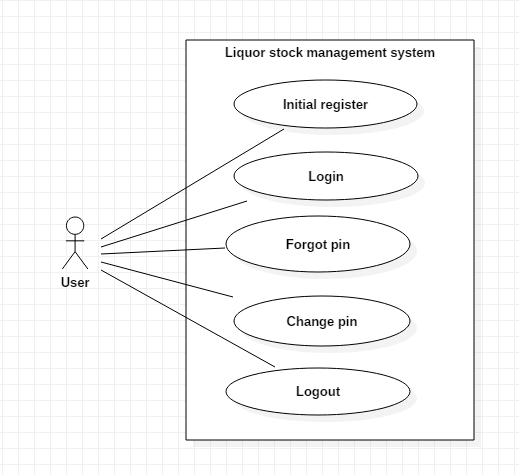


Figure : use case diagram 1

There is only a single actor for this system as the system is built for the standalone computer.

Scenario:

* User can register an account (pin) and recovery code is provided after registration.
* User can log in to the system using pin password.
* If the user forgets the password, they can recover the password using recovery code.
* User can change the pin password using the recovery code.
* User can log out from the system to stop access.

After logging in:

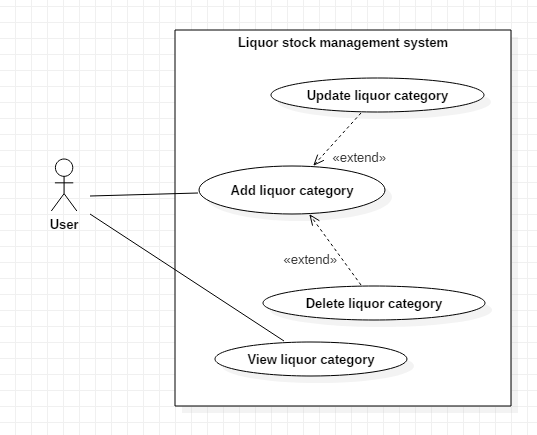


Figure : use case diagram 2

Scenario:

* User can add liquor category according to their wish.
* User can update or delete the existing liquor category.
* User can view all the existing liquor category.

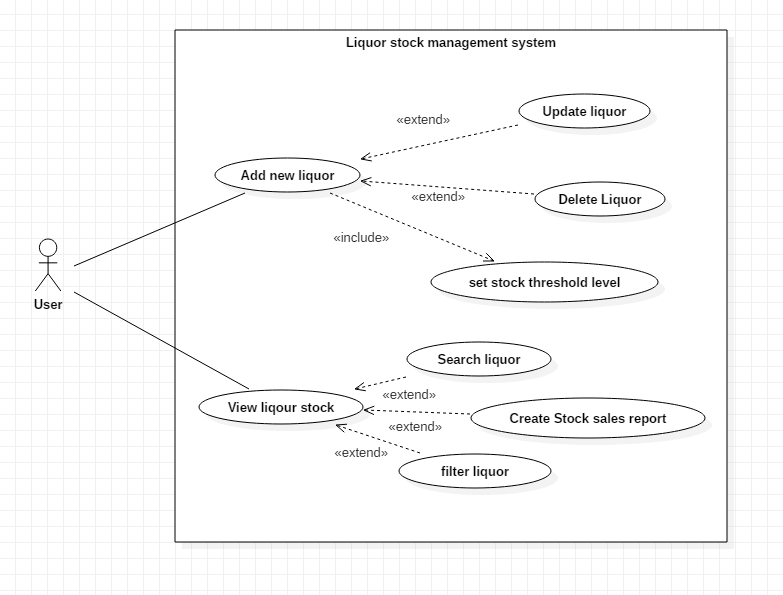


Figure : use case diagram 3

Scenario:

* User can add new liquor item on the database.
* User can update and delete the existing liquor stock.
* User can view all the existing liquor stock, filter using liquor category, search through liquor name and create stock sales report

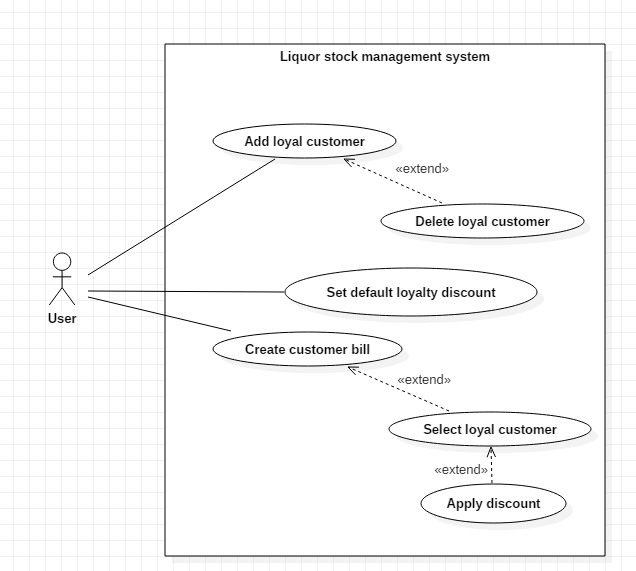


Figure : use case diagram 4

Scenario:

* User can add loyal customer for discount according to their points. The Loyal customer can obtain points when they buy any liquor and stacks with their previous purchase.
* User can set default loyalty discount to the loyal customer.
* User can delete the loyal customer.
* User can create customer bill and select loyal customer to update the loyal points and apply discount according to discount rate set by the user.

## 2.6: NLA and Initial Class Diagram

Liquor stock management system is a desktop based application system for stock management of a liquor store. The main aim of the system is to build an automated database system for managing the liquor stocks based on each item sold. The system allows to make work efficient and cut human errors.

The system should have a log-in system through a pin password. The system should also allow a user to change their log-in pin password.

The system must allow the user to add, edit and delete liquor items and its detail. The liquor item should contain its name, price, and quantity. The liquor has different categories so, the system should allow the user to view liquor stock according to the category. The user can also update and remove the liquor categories. The system should also allow the user to search liquor according to liquor name while viewing the stocks. The stock should have a threshold quantity level and should notify the user when the stock level is lower than the threshold.

The system should allow the user to create bills for the daily transaction of the store. The system should have a special feature for the customer through their emails to have a discount according to their earlier spending. The discount feature should be adjustable. The system should also create a report of sales.

From the scenario, the Natural Language Analysis (NLA) is performed and candidate classes, attributes and methods are picked up from the nouns, verbs and adjectives.

|  |  |  |
| --- | --- | --- |
| **Candidate classes** | **Candidate attributes** | **Candidate methods** |
| * User * Liquor * Category * Customer * Bill | * Pin * Liquor name * Price * Quantity * Category name * Threshold quantity * Discount * Email | * Login * Change * Add * Delete * Update * Search * View * Notify * Create report |

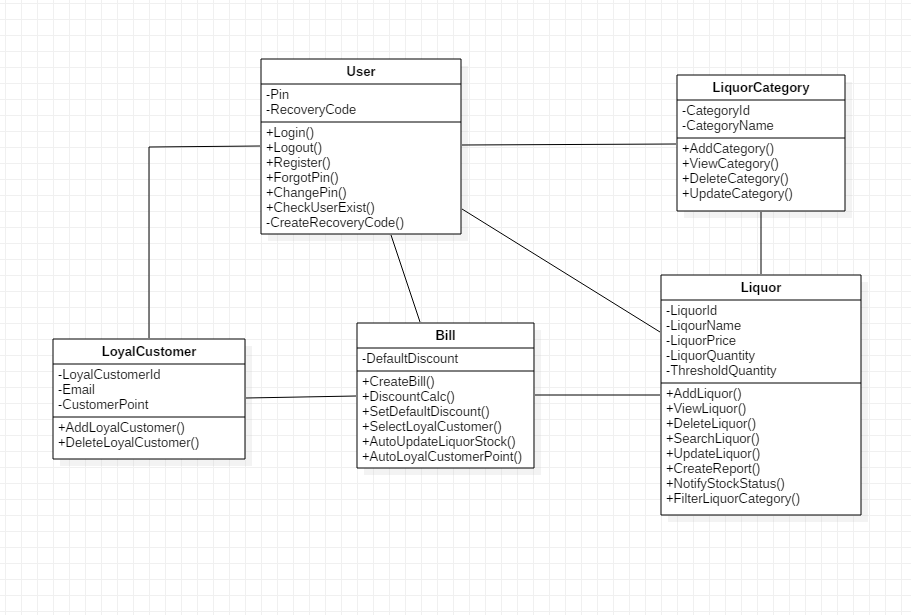


Figure : Initial Class Diagram