# 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.:

|  |  |  |
| --- | --- | --- |
| **Type of feasibility study** | **About the study** | **Association with the project** |
| Economic Feasibility | Deals whether the allocated budget is sufficient enough to complete the project successfully. | The development does not demand additional budget to complete project as the requirements for the system are little. |
| Technical Feasibility | Deals whether the current Technical capabilities as well as the capabilities of the person using the system. | The resources available to me are sufficient enough to develop this project. |
| Cultural Feasibility | Local and greater societal and cultural impact. | This system helps easy management of the stocks and sales and increase the efficiency. |
| Schedule Feasibility | Deals whether allocated time and resources are enough to complete the project. | Limited requirements are well recognized, the project can be completed in time unless there will be any exceptions. |
| Resources Feasibility | Deals what resources are needed and is sufficient enough to complete the project successfully. | The store needs to install new computer if there are no computers. Or upgrade the computer if it does not meets the hardware and software requirements. |

## 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Data** | **Rational** | **Dependency** |
| F1 | Initial Register | Pin, Recovery code | Create the user through pin code |  |
| F2 | Login | Pin | To verify user and give user access to the system facilities | F1 |
| F3 | Forgot pin | Recovery code | To recover the pin using recovery code | F1,F2 |
| F4 | Add liquor category | Category name | Let user to add liquor category | F2 |
| F5 | View liquor category | N/A | Let user to view all existing liquor category | F2 |
| F6 | Update liquor category | Category name | Let user to update, edit the existing category | F4 |
| F7 | Remove liquor category | Liquor category or category id | Let user to remove the existing category | F4 |
| F8 | Add a new liquor | Liquor name, Price, Quantity | Let user to add a new liquor product | F2,F4 |
| F9 | View liquor stock | N/A | Let user to view all existing liquor stock | F2 |
| F10 | Update existing liquor stock | Liquor name, Price, Quantity | Let user to update, edit the existing liquor | F8 |
| F11 | Delete existing liquor stock | Liquor name | Let user to delete the existing liquor | F8 |
| F12 | Create customer bill | Liquor details | Let user to create bills | F2 |
| F13 | Dynamic update liquor stock | N/A | Automatically changes stock quantity when creating bill | F12 |
| F14 | Add loyal customer | Email | Let user to add loyal customer | F2 |
| F15 | Delete loyal customer | Email | Let user to delete loyal customer | F14 |
| F16 | Select loyal customer | Email | Let user select loyal customer while creating bill to update loyal points or apply discount | F12 |
| F17 | Update customer loyal point | N/A | Loyal customer automatically gains point according to their spending | F12 |
| F18 | Apply discount | N/A | Allow to apply discount when loyal customer has certain number of loyal points | F12,F16 |
| F19 | Search liquors | Liquor name | Let user to search liquor using liquor name | F9 |
| F20 | Filter liquor through category | Liquor category | Let user to filter liquor using liquor category | F9 |
| F21 | Set Default loyalty discount | Discount percent | Set default value of discount percent when loyal customer gets discount | F2 |
| F22 | Set stock threshold level | Threshold quantity | Allow user to set stock threshold quantity. | F8 |
| F23 | Liquor stock status | N/A | Notify if stock quantity is less than threshold quantity. | F2 |
| F24 | Create stock sales report | N/A | Create sales report of the liquor stock | F9 |
| F25 | Change pin | Pin details | Change the pin password | F2 |
| F26 | Logout | N/A | To exit the system access. | F2 |

## 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Requirement** | **Description** | **Rational** | **Dependency** |
| N1 | Performance | The system should perform task fast and as expected. | To make user experience smooth and efficient. | N/A |
| N2 | Scalability | The system should able to scale according to the increase in the stock. | To make system acceptable when number of stocks increases. | N/A |
| N3 | Capacity | The system should able to handle multiple task simultaneously. | To ensure multiple task can be performed at a time. | N/A |
| N4 | Availability | The system should be available when needed. | To make system available whenever user needs. | N/A |
| N5 | Reliability | The system should consistency in performance during every runtime. | To ensure the user can trust the system whenever it is needed. | N/A |
| N6 | Recoverability | The system should be able to recover from any problem user faces. | To have a recoverable code if user forgets the pin code. | F1,F3 |
| N7 | Security | The system should have data security and proper access control. | To deny any unauthorized access. | F1,F2 |
| N8 | Data integrity | There should not be duplication and inconsistency in stored data. | To store the data accurately and maintain consistency of the data. | N/A |
| N9 | Maintainability | The system should be easy to maintain if any bugs exist. | To have sustainable runtime of the system and improve overtime. | N/A |
| N10 | Usability | The system should be acceptable to the end user and satisfy their requirements. | To make the system user friendly and easier to operate. | N/A |
| N11 | Documentation | The project should be well documented that should give overview of the system and how the system is built. | For user to learn how to use the system and for any third party developer to know about the system. | N/A |
| N12 | Legal | The system should be legally valid and available to use. | To have legal value in the market and avoid any legal problems. | N/A |

## 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

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirements** | **MoSCoW** |
| F1 | Initial Register | Must have |
| F2 | Login | Must have |
| F3 | Forgot pin | Should have |
| F4 | Add liquor category | Must have |
| F5 | View liquor category | Should have |
| F6 | Update liquor category | Must have |
| F7 | Remove liquor category | Must have |
| F8 | Add a new liquor | Must have |
| F9 | View liquor stock | Should have |
| F10 | Update existing liquor stock | Must have |
| F11 | Delete existing liquor stock | Must have |
| F12 | Create customer bill | Must have |
| F13 | Dynamic update liquor stock | Should have |
| F14 | Add loyal customer | Should have |
| F15 | Delete loyal customer | Could have |
| F16 | Select loyal customer | Should have |
| F17 | Update customer loyal point | Should have |
| F18 | Apply discount | Should have |
| F19 | Search liquors | Should have |
| F20 | Filter liquor through category | Should have |
| F21 | Set Default loyalty discount | Should have |
| F22 | Set stock threshold level | Must have |
| F23 | Liquor stock status | Must have |
| F24 | Create stock sales report | Must have |
| F25 | Change pin | Should have |
| F26 | Logout | Must have |
| N1 | Performance | Must have |
| N2 | Scalability | Could have |
| N3 | Capacity | Must have |
| N4 | Availability | Must have |
| N5 | Reliability | Must have |
| N6 | Recoverability | Should have |
| N7 | Security | Must have |
| N8 | Data integrity | Must have |
| N9 | Maintainability | Should have |
| N10 | Usability | Must have |
| N11 | Documentation | Should have |
| N12 | Legal | Should have |

## 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.

|  |  |
| --- | --- |
| **Hardware Requirements** | **Software Requirements** |
| * Processor: Intel Dual-core processor, 2.0 GHz or higher * RAM: Minimum 2 GB of RAM * Standalone Network * Hard Drive: 500GB | * Operating System: windows 7 or higher * Front End: C# dotnet (visual studio 2017) * Back End: MS SQL Server |

## 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.

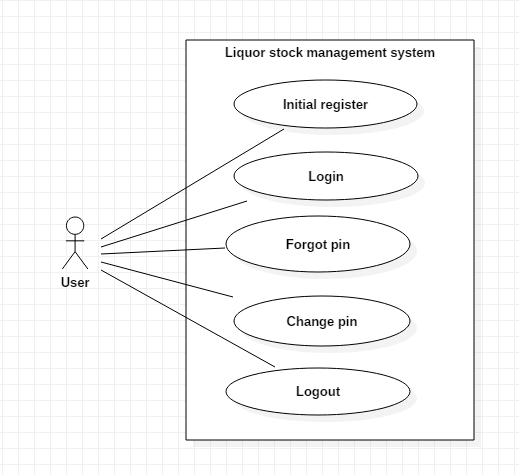


Figure : use case diagram 1

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 log out from the system to stop access.

After logging in:

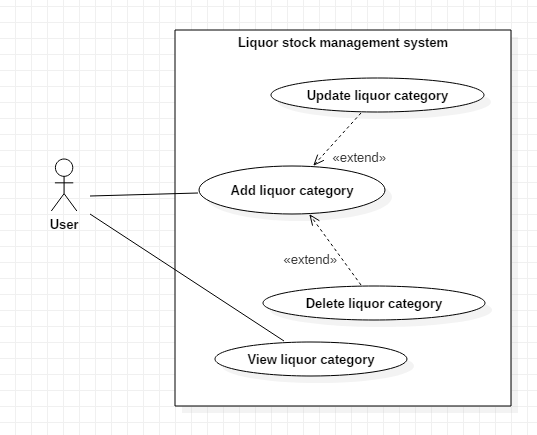


Figure : use case diagram 2

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.

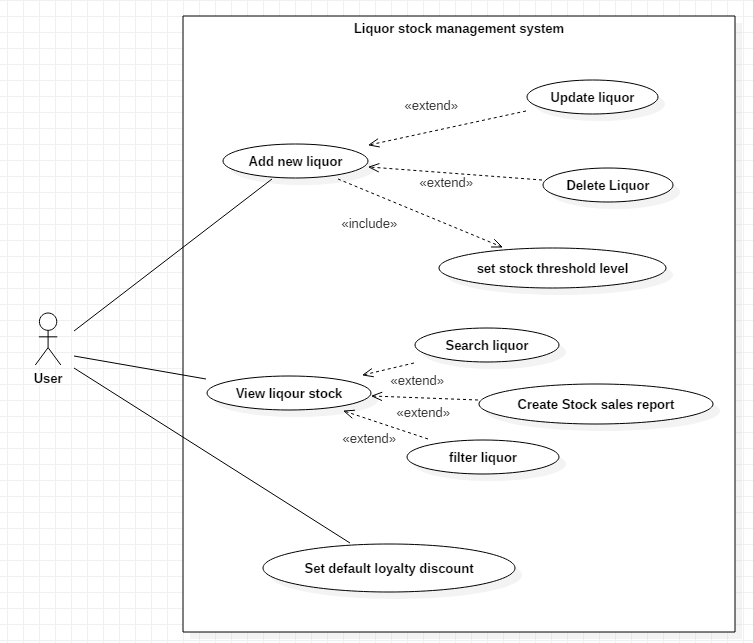


Figure : use case diagram 3

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.
* User can set default loyalty discount to the loyal customer.

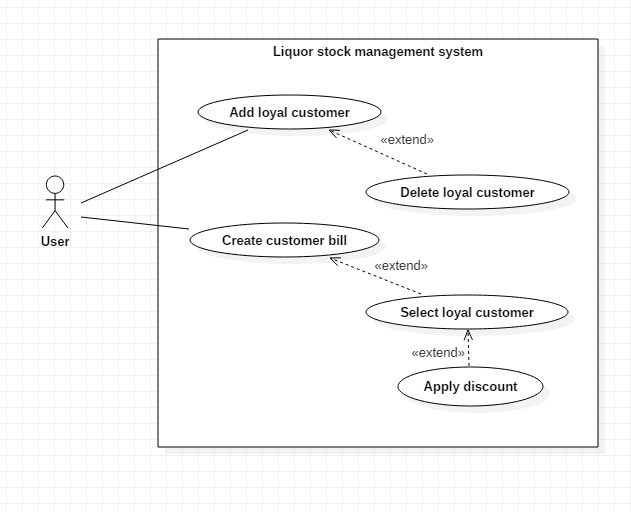


Figure : use case diagram 4

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 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 |

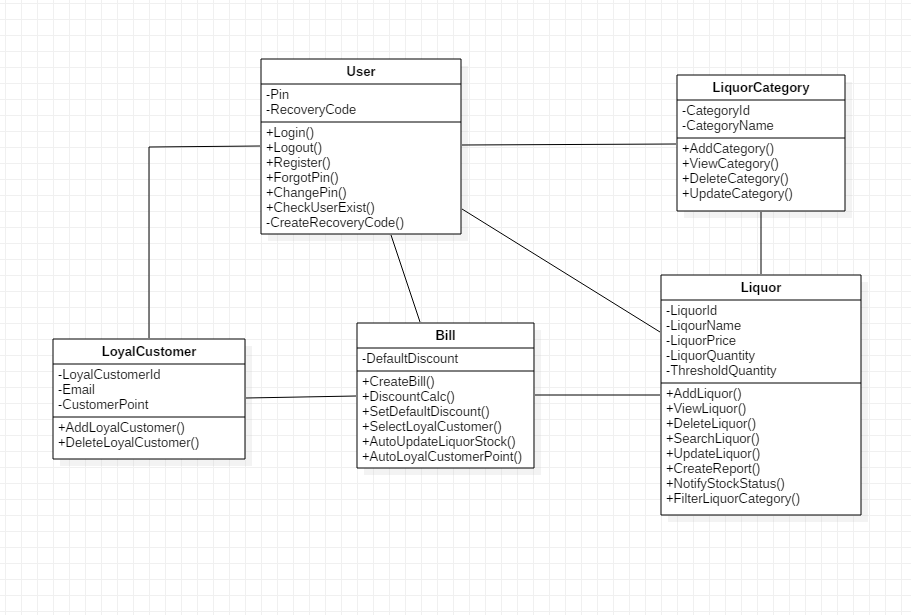


Figure : Initial Class Diagram