# **2.1 Introduction to Analysis**

Analysis is a vital aspect in any system development. Here, we identify the requirements, feasibility of the system under several areas using appropriate analysis methodology. After that we under go in preparation of preparing various models like Use Cases and Initial Class Diagram, which provides certain vision what we actually identified during analysis.

Regarding to my project, Mark Sheet Generator, which is totally based on student’s data management dealing with students’ progress reports. Therefore, for this project data management concept is essential aspects.

# **2.2 Analysis Methodology**

Several Analysis Methodology can contribute in analysis. Despite, Hard System Methodology seem to be appropriate for my project, Mark Sheet Generator.

Hard System Methodology is analysis methodology that undergoes by following certain rules, guidelines and standards. Therefore, it is called as highly structured approach. It is suitable for small project like ours. In Hard approach we will be preparing logical data modeling, Data flow Diagram, etc. Hard approach is likely to called as Structured System Analysis Design Methodology (SSADM), which poses mainly three views that helps in analysis. Three views of SSADM are:

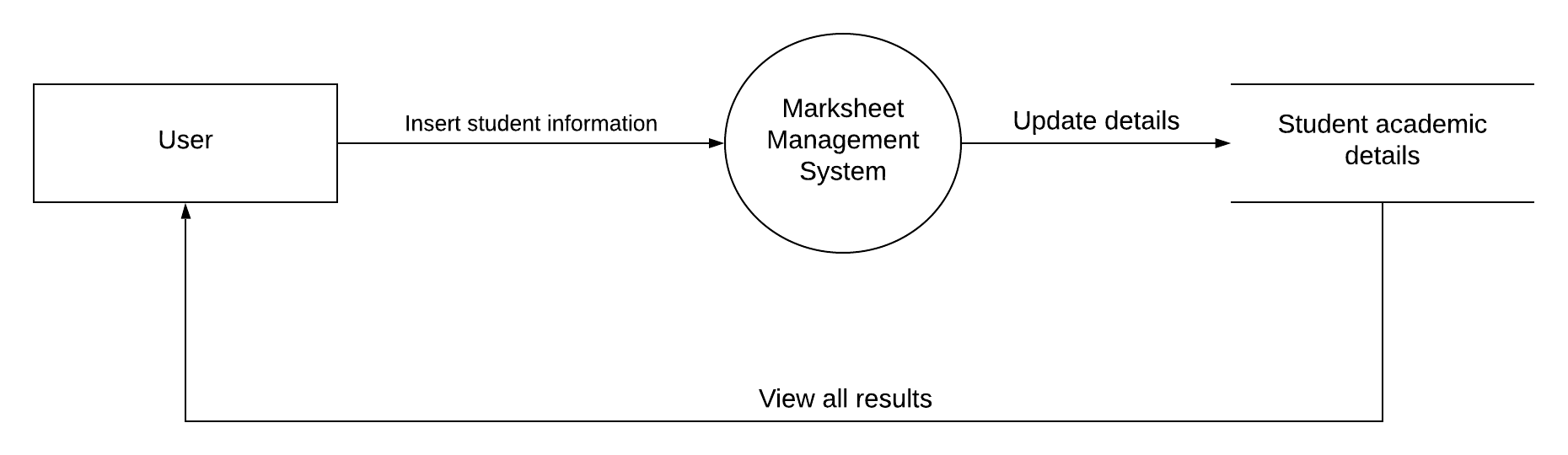
Process view, defines functions like storing database of the students, preparing mark sheet that system should carry, how data like students details, students obtained marks will move in the entire system and how it changes as it is processed.

Data view, defines the data like students marks in respective test and information which system will be using.

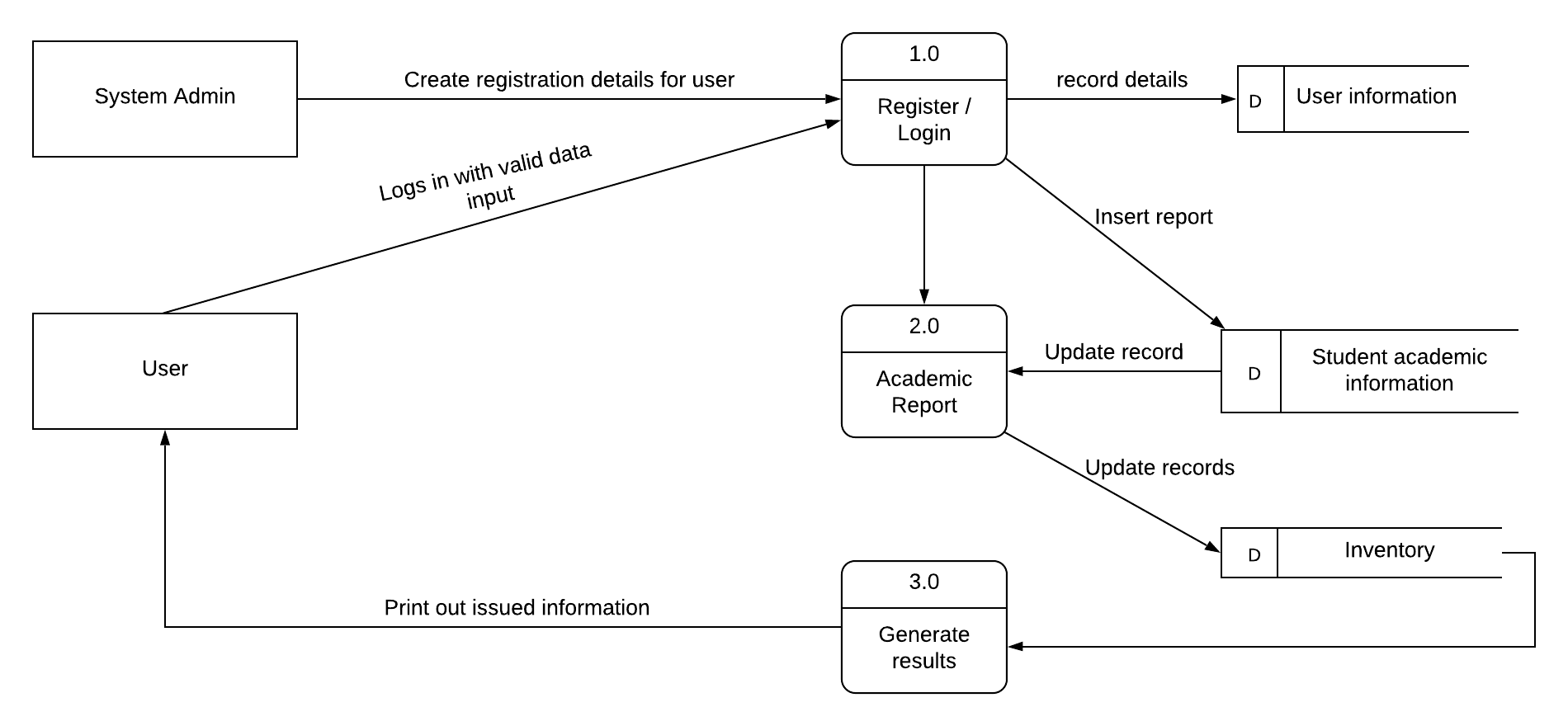
Event view, describes the events (print view, exporting data and so on) which helps in setting up of the processes running and the effect of the external events on the data.

As SSADM is being applied in this system for the development, we require to illustrate how data is stored, manipulated and flow throughout the system. These can be shown in Data Flow Diagram (DFD. The DFDs of this system are:

Level 0 DFD:



Level 1 DFD:



# **2.3 Feasibility Study**

Feasibility study is what we do at initial level of system development that contribute in determining the technical, operational, financial, time factor, economical, legal, social and political aspects of the project. In these aspects of the study, we make sure that any of the feasible factors may not have any sorts of negative impact on our project. This is why we perform at initial level of the system development to sort out the negative sides, which can be obstacles or might cause problems in upcoming days.

The aspects of the feasibility study view based on this project is briefly explained below.

* Technical: Technically this project Mark Sheet Generator is can be said as having good feasibility. As backend programming will be done using Object Oriented Principles, which makes codes systematic.
* Economical: As this project is small- scale project, this does not require huge budgeting. Economical prospective for this project is also feasible which can contribute for success of this project.
* Time: By making a time- schedule, we can complete this project within certain time. Allocation of specific time for specific task can result in positive outcomes. This way time is also feasible for this project.
* Social: Socially, this system has wider acceptance. As this system will be in use by any educational institution to serve the progress reports of the students and keeping proper records of the students. This means this system does not affect any sorts of social believes and has greater feasibility.
* Legal: Legally it is does not hamper in system development. The system will be developed under legal alliances.
* Political: This system does not concern about any political issues. As the system will be concerned with educational institution, not to any governmental bodies.

# **2.4 SRS (Software Requirement Specification)**

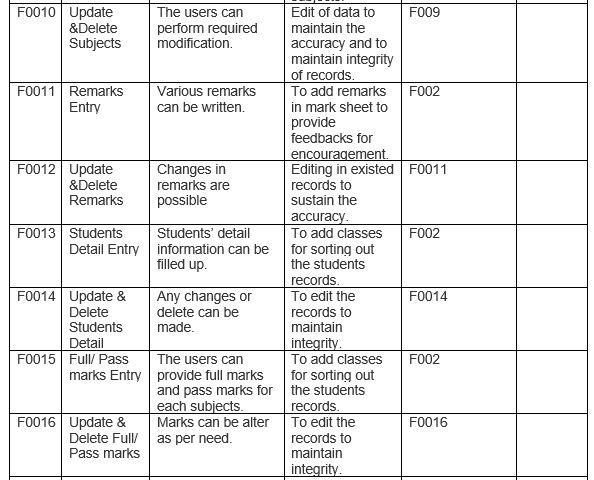
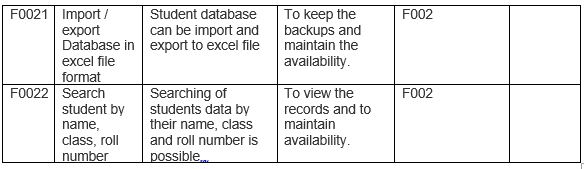
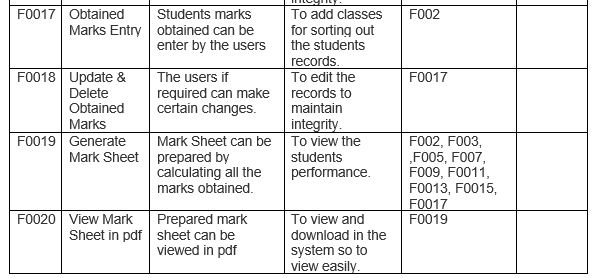
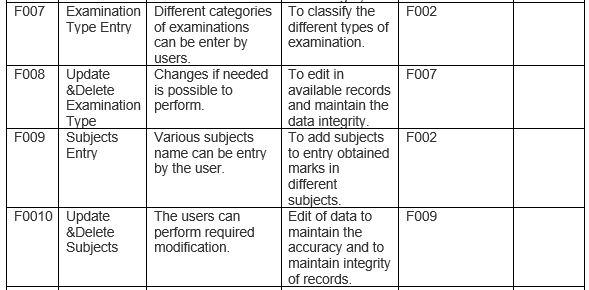
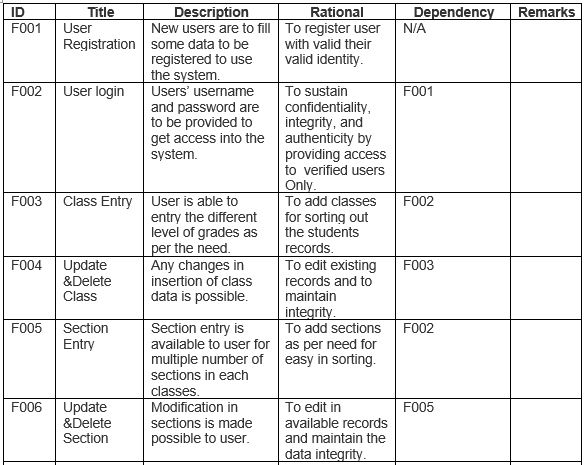
Software Requirement Specification is where all the system requirements are mentioned. SRS allows users to analyze if the requirements in it are as per their requirements. It consist of user requirements of system also complete specifications of the system requirements. ***(softwarerequirementsspecification, 2019)***

## **2.4.1 Functional Requirement**

A statement of all the planned function of a system is term as functional requirement. It let know about the preferred end function of the system to guarantee the design is suitable to create the chosen product.

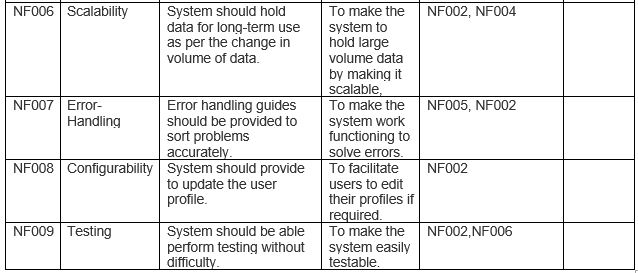
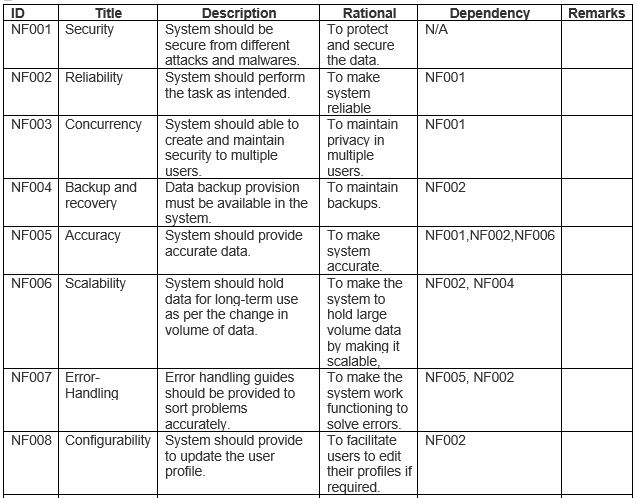
***(functional-requirements, 2019)***

The functional requirement of this system is:



## **2.4.1 Non- Functional Requirement**

Nonfunctional requirements are quality attributes of the system that deals with the usability, reliability, security, performance, scalability and so on.   
Some of the nonfunctional requirements for this system are:

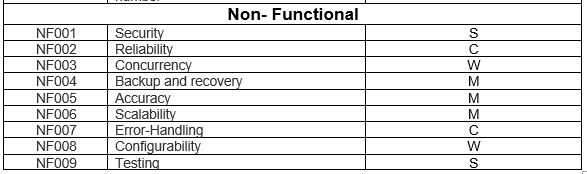
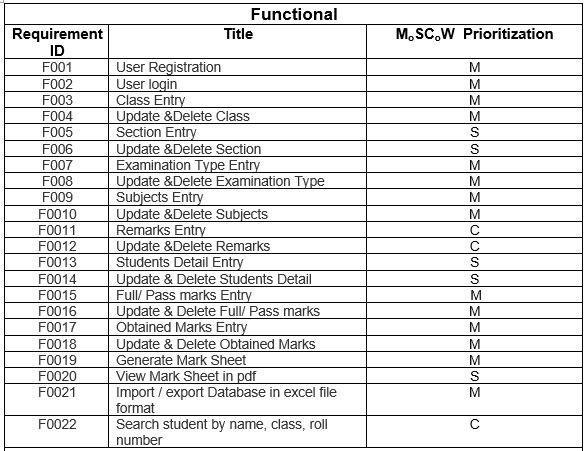


## **2.4.3 MoSCoW Prioritization**

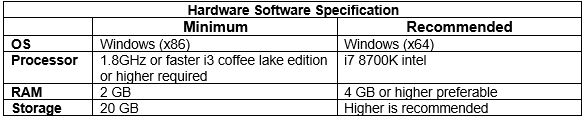
MoSCoW Prioritization is a methodology, which prioritized the requirements to prevent them from being expensive or unrealistic. The prime motive is to take such requirements that are the most valuable for the system.

The acronym of MoSCoW is:

M= Must have  
S= Should have  
C= Could have  
W= Won’t have



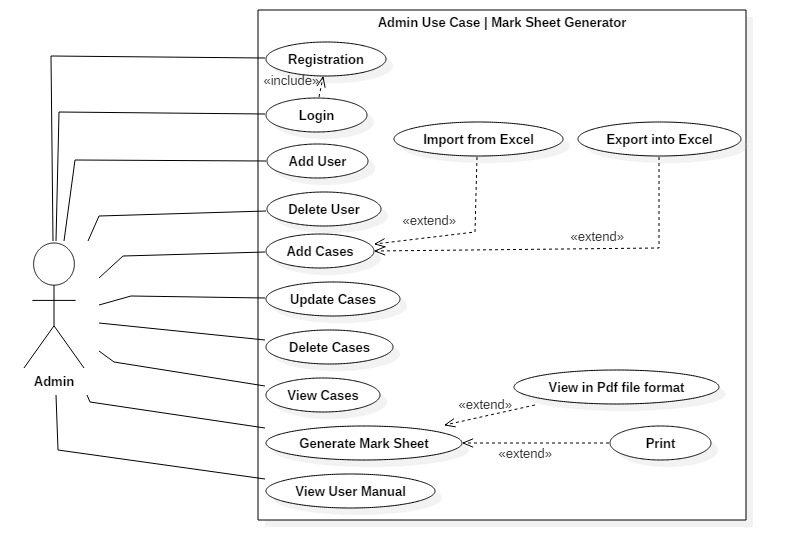
## **2.4.4 Hardware Software Specification**

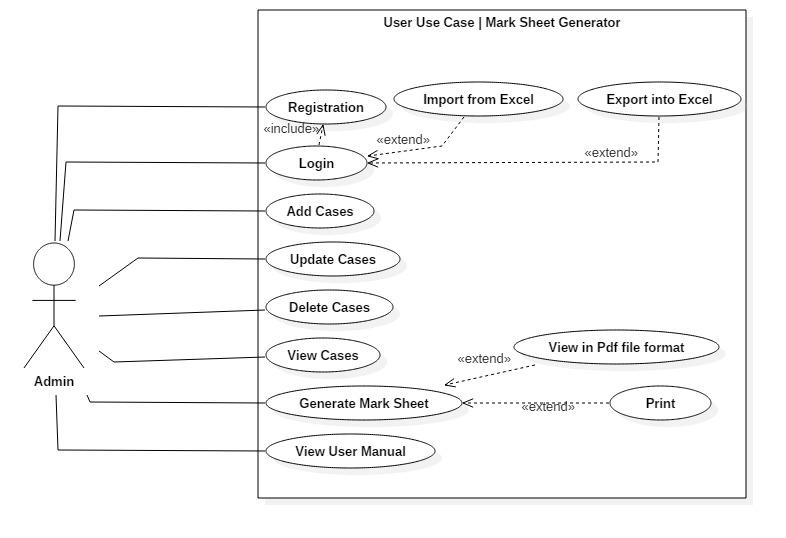


# **2.5 Use Case Diagram**

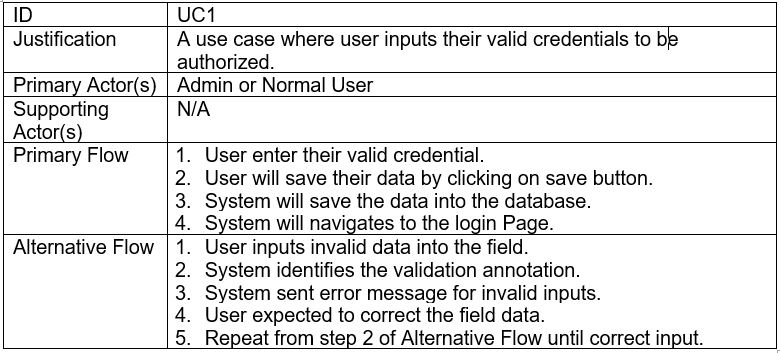
Use case Diagram is a diagram that provide the details of the interaction between the users (actors) and the system. Here, we use specialized symbols and connectors. It displays the different ways of interactions of users and the system.

The Use Cases for this system are:

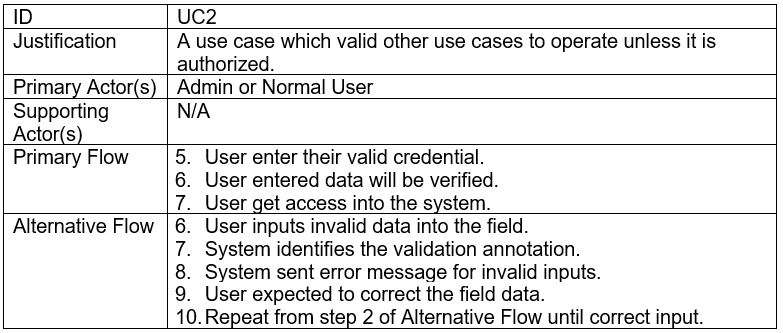




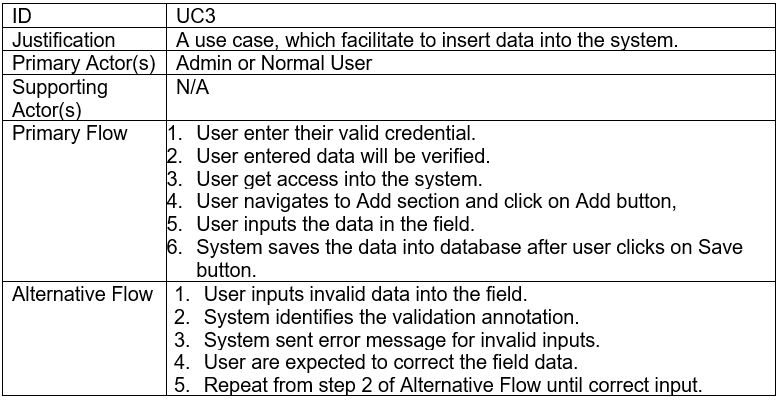
**Title: “Registration”**



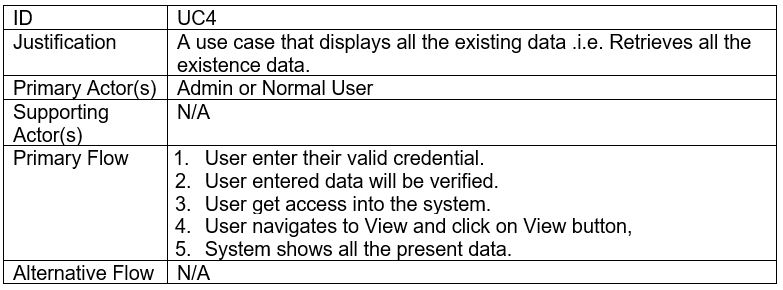
**Title: “Login”**



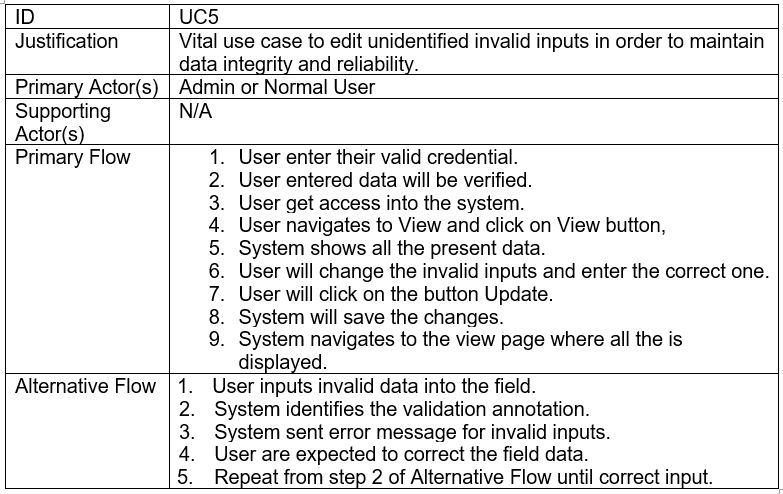
**Title: “Add Case”**



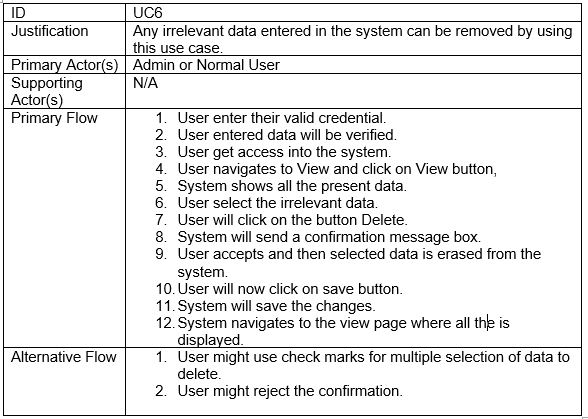
**Title: “View Case”**



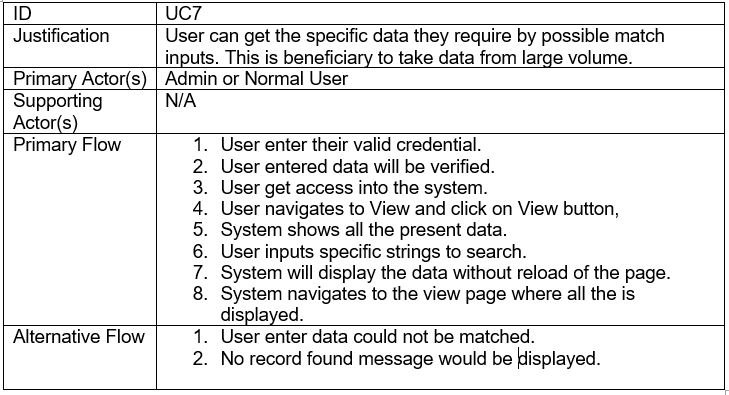
**Title: “Update Case”**



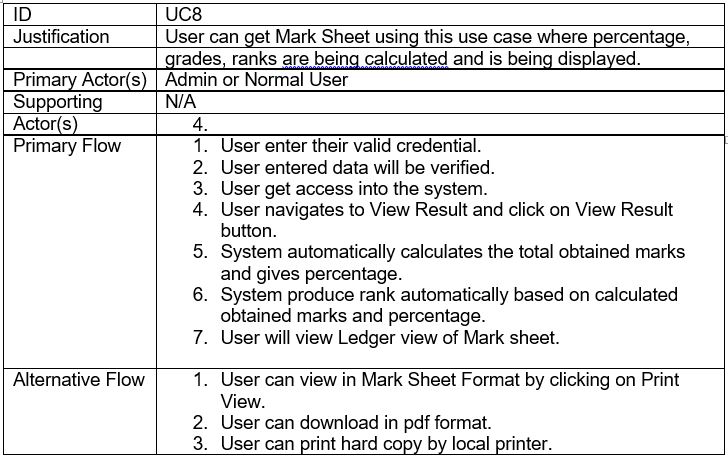
**Title: “Delete Case”**



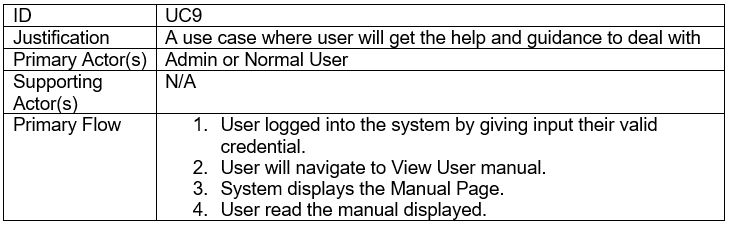
**Title: “Search Case”**



**Title: “Generate Mark Sheet”**



**Title: “User Manual”**



# **Initial Class Diagram**

A diagram that is produced to represent the relationship between classes of the system is called class diagram. It also consists of attributes of the classes with their operations. Class diagram is part of a unified modeling language (UML).

In order to make class diagram we perform NLA (Natural Language Analysis).

## **Scenario**

Mark Sheet Generator is an academic system can be fruitful in any educational institution. Students detail information is maintained by this system. It is desktop application. This system prepare mark sheet of students. For this user will have to input the subjects, examination type and obtained marks. After this this system produces mark sheet with automation of rank.

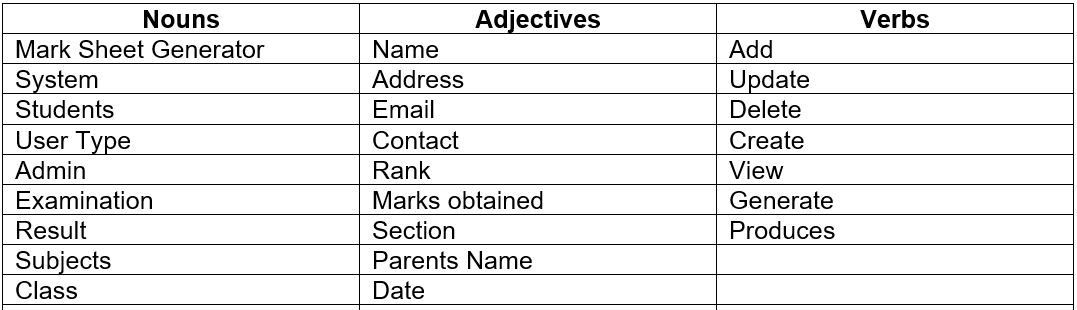
User level privilege is also provided in this system. Admin can create and delete the users. Rest other features are provided to both admin and normal users.

## **NLA (Natural Language Analysis)**

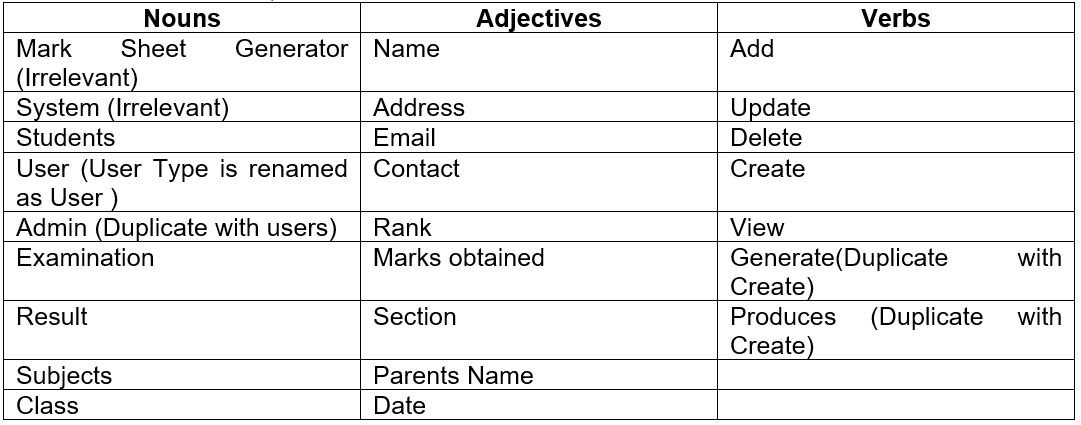
NLA is the process of identifying nouns as potential classes, verbs as potential function and adjectives as potential attributes.

Initial step to proceed NLA is to identify the nouns, adjectives and verbs from the scenario of the system.

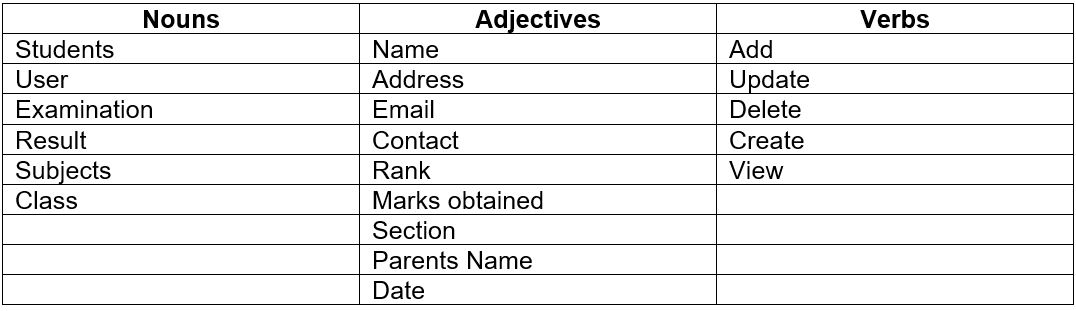
Here we find out the nouns, adjectives and verbs as classes, attributes and functions from the scenario.



Now we filter the similar, irrelevant values



Therefore, we get the following information to produce initial class diagram.



The Initial Class Diagram of the Mark Sheet Generator is:

