# **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:

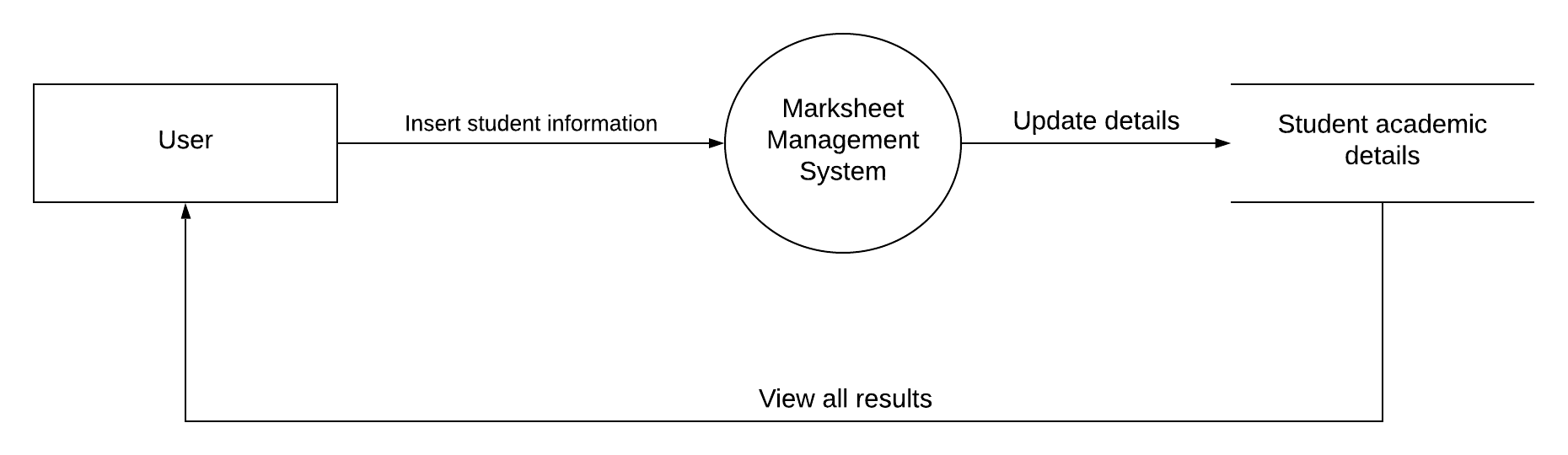
Process view, defines functions like storing database of the students, preparing mark sheet that system should carry, how data like students details, students obtained marks will move in the entire system and how it changes as it is processed.

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

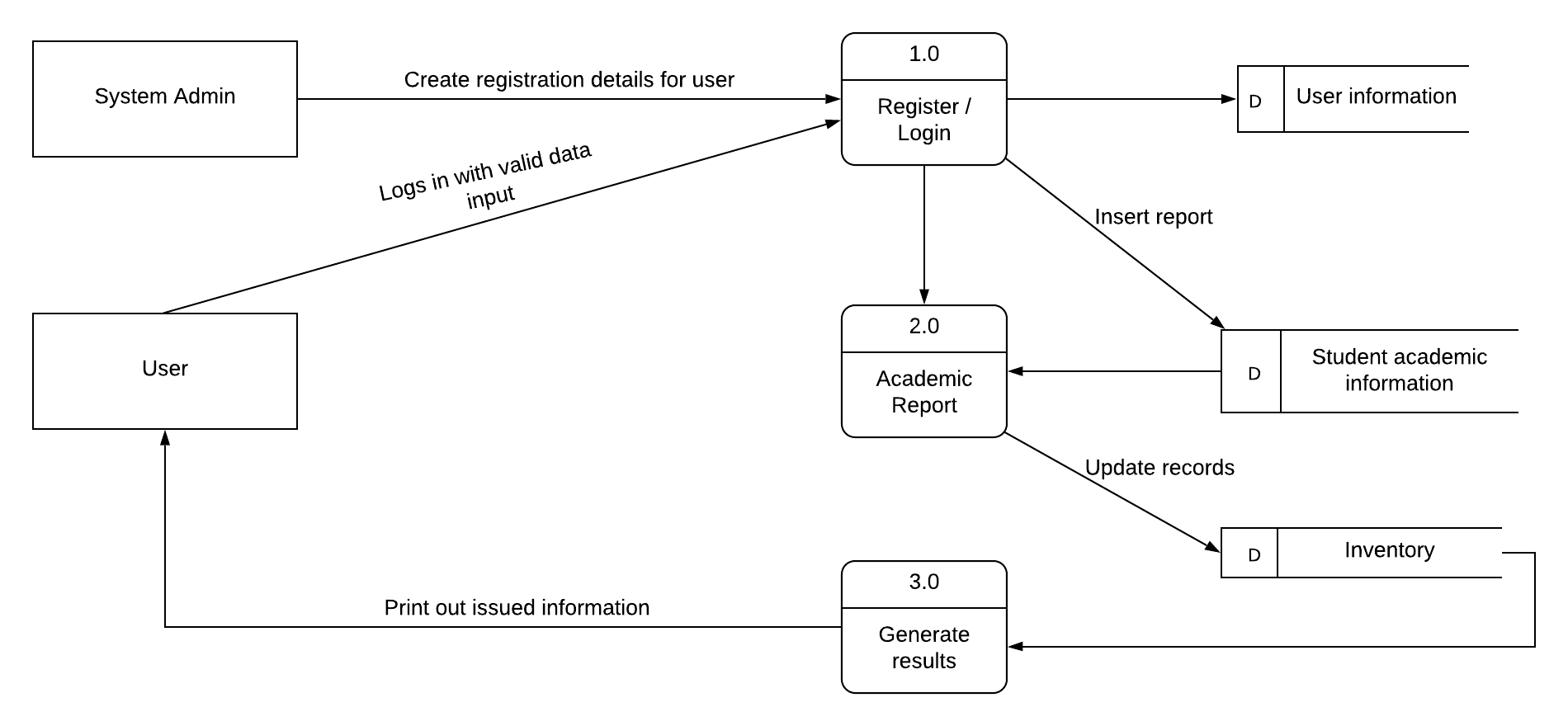
Event view, describes the events (print view, exporting data and so on) which helps in setting up of the processes running and the effect of the external events on the data.

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:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** | **Remarks** |
| F001 | User Registration | New users are to fill some data to be registered to use the system. | To register user with valid their valid identity. | N/A |  |
| F002 | User login | Users’ username and password are to be provided to get access into the system. | To sustain confidentiality,  integrity, and authenticity by  providing access to verified users  Only. | F001 |  |
| F003 | Class Entry | User is able to entry the different level of grades as per the need. | To add classes for sorting out the students records. | F002 |  |
| F004 | Update &Delete Class | Any changes in insertion of class data is possible. | To edit existing records and to maintain integrity. | F003 |  |
| F005 | Section Entry | Section entry is available to user for multiple number of sections in each classes. | To add sections as per need for easy in sorting. | F002 |  |
| F006 | Update &Delete Section | Modification in sections is made possible to user. | To edit in available records and maintain the data integrity. | F005 |  |
| F007 | Examination Type Entry | Different categories of examinations can be enter by users. | To classify the different types of examination. | F002 |  |
| F008 | Update &Delete Examination Type | Changes if needed is possible to perform. | To edit in available records and maintain the data integrity. | F007 |  |
| F009 | Subjects Entry | Various subjects name can be entry by the user. | To add subjects to entry obtained marks in different subjects. | F002 |  |
| F0010 | Update &Delete Subjects | The users can perform required modification. | Edit of data to maintain the accuracy and to maintain integrity of records. | F009 |  |
| F0011 | Remarks Entry | Various remarks can be written. | To add remarks in mark sheet to provide feedbacks for encouragement. | F002 |  |
| F0012 | Update &Delete Remarks | Changes in remarks are possible | Editing in existed records to sustain the accuracy. | F0011 |  |
| F0013 | Students Detail Entry | Students’ detail information can be filled up. | To add classes for sorting out the students records. | F002 |  |
| F0014 | Update & Delete Students Detail | Any changes or delete can be made. | To edit the records to maintain integrity. | F0014 |  |
| F0015 | Full/ Pass marks Entry | The users can provide full marks and pass marks for each subjects. | To add classes for sorting out the students records. | F002 |  |
| F0016 | Update & Delete Full/ Pass marks | Marks can be alter as per need. | To edit the records to maintain integrity. | F0016 |  |
| F0017 | Obtained Marks Entry | Students marks obtained can be enter by the users | To add classes for sorting out the students records. | F002 |  |
| F0018 | Update & Delete Obtained Marks | The users if required can make certain changes. | To edit the records to maintain integrity. | F0017 |  |
| F0019 | Generate Mark Sheet | Mark Sheet can be prepared by calculating all the marks obtained. | To view the students performance. | F002, F003, ,F005, F007, F009, F0011, F0013, F0015, F0017 |  |
| F0020 | View Mark Sheet in pdf | Prepared mark sheet can be viewed in pdf | To view and download in the system so to view easily. | F0019 |  |
| F0021 | Import / export Database in excel file format | Student database can be import and export to excel file | To keep the backups and maintain the availability. | F002 |  |
| F0022 | Search student by name, class, roll number | Searching of students data by their name, class and roll number is possible.. | To view the records and to maintain availability. | F002 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** | **Remarks** |
| NF001 | Security | System should be secure from different attacks and malwares. | To protect and secure the data. | N/A |  |
| NF002 | Reliability | System should perform the task as intended. | To make system reliable | NF001 |  |
| NF003 | Concurrency | System should able to create and maintain security to multiple users. | To maintain privacy in multiple users. | NF001 |  |
| NF004 | Backup and recovery | Data backup provision must be available in the system. | To maintain backups. | NF002 |  |
| NF005 | Accuracy | System should provide accurate data. | To make system accurate. | NF001,NF002,NF006 |  |
| NF006 | Scalability | System should hold data for long term use as per the change in volume of data. | To make the system to hold large volume data by making it scalable, | NF002, NF004 |  |
| NF007 | Error-Handling | Error handling guides should be provided to sort problems accurately. | To make the system work functioning to solve errors. | NF005, NF002 |  |
| NF008 | Configurability | System should provide to update the user profile. | To facilitate users to edit their profiles if required. | NF002 |  |
| NF009 | Testing | System should be able perform testing without difficulty. | To make the system easily testable. | NF002,NF006 |  |

## **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

|  |  |  |
| --- | --- | --- |
| **Functional** | | |
| **Requirement ID** | **Title** | **MoSCoW Prioritization** |
| F001 | User Registration | M |
| F002 | User login | M |
| F003 | Class Entry | M |
| F004 | Update &Delete Class | M |
| F005 | Section Entry | S |
| F006 | Update &Delete Section | S |
| F007 | Examination Type Entry | M |
| F008 | Update &Delete Examination Type | M |
| F009 | Subjects Entry | M |
| F0010 | Update &Delete Subjects | M |
| F0011 | Remarks Entry | C |
| F0012 | Update &Delete Remarks | C |
| F0013 | Students Detail Entry | S |
| F0014 | Update & Delete Students Detail | S |
| F0015 | Full/ Pass marks Entry | M |
| F0016 | Update & Delete Full/ Pass marks | M |
| F0017 | Obtained Marks Entry | M |
| F0018 | Update & Delete Obtained Marks | M |
| F0019 | Generate Mark Sheet | M |
| F0020 | View Mark Sheet in pdf | S |
| F0021 | Import / export Database in excel file format | M |
| F0022 | Search student by name, class, roll number | C |
| **Non- Functional** | | |
| NF001 | Security | S |
| NF002 | Reliability | C |
| NF003 | Concurrency | W |
| NF004 | Backup and recovery | M |
| NF005 | Accuracy | M |
| NF006 | Scalability | M |
| NF007 | Error-Handling | C |
| NF008 | Configurability | W |
| NF009 | Testing | S |

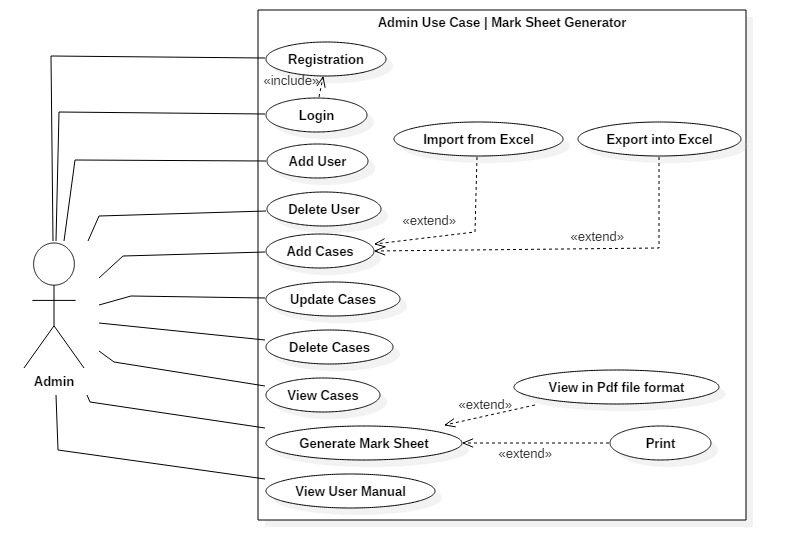
## **2.4.4 Hardware Software Specification**

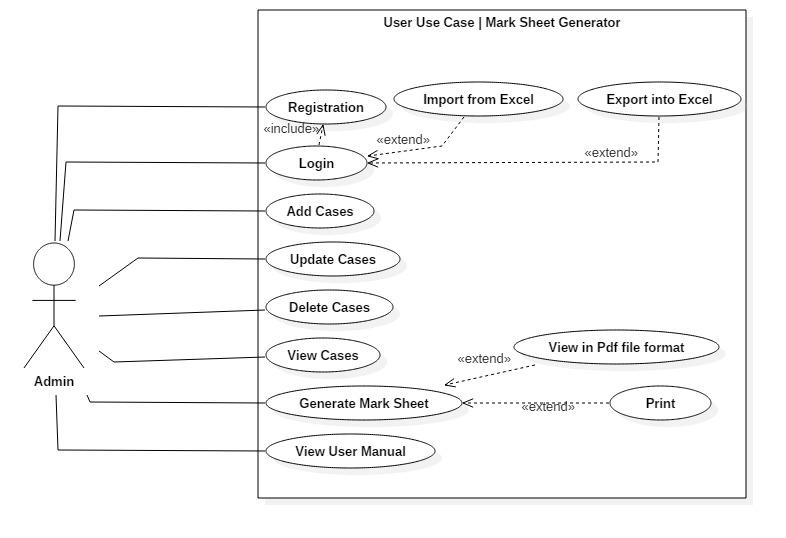
|  |  |  |
| --- | --- | --- |
| **Hardware Software Specification** | | |
|  | **Minimum** | **Recommended** |
| **OS** | Windows (x86) | Windows (x64) |
| **Processor** | 1.8GHz or faster i3 coffee lake edition or higher required | i7 8700K intel |
| **RAM** | 2 GB | 4 GB or higher preferable |
| **Storage** | 20 GB | Higher is recommended |

# **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”**

|  |  |
| --- | --- |
| ID | UC1 |
| Justification | A use case where user inputs their valid credentials to get authorized. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User will save their data by clicking on save button. 3. System will save the data into the database. 4. System will navigates to the login Page. |
| Alternative Flow | 1. User inputs invalid data into the field. 2. System identifies the validation annotation. 3. System sent error message for invalid inputs. 4. User expected to correct the field data. 5. Repeat from step 2 of Alternative Flow until correct input. |

**Title: “Login”**

|  |  |
| --- | --- |
| ID | UC2 |
| Justification | A use case which valid other use cases to operate unless it is authorized. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. |
| Alternative Flow | 1. User inputs invalid data into the field. 2. System identifies the validation annotation. 3. System sent error message for invalid inputs. 4. User expected to correct the field data. 5. Repeat from step 2 of Alternative Flow until correct input. |

**Title: “Add Case”**

|  |  |
| --- | --- |
| ID | UC3 |
| Justification | A use case which facilitate to insert data into the system. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. 4. User navigates to Add section and click on Add button, 5. User inputs the data in the field. 6. System saves the data into database after user clicks on Save button. |
| Alternative Flow | 1. User inputs invalid data into the field. 2. System identifies the validation annotation. 3. System sent error message for invalid inputs. 4. User are expected to correct the field data. 5. Repeat from step 2 of Alternative Flow until correct input. |

**Title: “View Case”**

|  |  |
| --- | --- |
| ID | UC4 |
| Justification | A use case that displays all the existing data .i.e. Retrieves all the existence data. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. 4. User navigates to View and click on View button, 5. System shows all the present data. |
| Alternative Flow | N/A |

**Title: “Update Case”**

|  |  |
| --- | --- |
| ID | UC5 |
| Justification | Vital use case to edit unidentified invalid inputs in order to maintain data integrity and reliability. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. 4. User navigates to View and click on View button, 5. System shows all the present data. 6. User will change the invalid inputs and enter the correct one. 7. User will click on the button Update. 8. System will save the changes. 9. System navigates to the view page where all the is displayed. |
| Alternative Flow | 1. User inputs invalid data into the field. 2. System identifies the validation annotation. 3. System sent error message for invalid inputs. 4. User are expected to correct the field data. 5. Repeat from step 2 of Alternative Flow until correct input. |

**Title: “Delete Case”**

|  |  |
| --- | --- |
| ID | UC6 |
| Justification | Any irrelevant data entered in the system can be removed by using this use case. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. 4. User navigates to View and click on View button, 5. System shows all the present data. 6. User select the irrelevant data. 7. User will click on the button Delete. 8. System will send a confirmation message box. 9. User accepts and then selected data is erased from the system. 10. User will now click on save button. 11. System will save the changes. 12. System navigates to the view page where all the is displayed. |
| Alternative Flow | 1. User might use check marks for multiple selection of data to delete. 2. User might reject the confirmation. |

**Title: “Search Case”**

|  |  |
| --- | --- |
| ID | UC7 |
| Justification | User can get the specific data they require by possible match inputs. This is beneficiary to take data from large volume. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. 4. User navigates to View and click on View button, 5. System shows all the present data. 6. User inputs specific strings to search. 7. System will display the data without reload of the page. 8. System navigates to the view page where all the is displayed. |
| Alternative Flow | 1. User enter data could not be matched. 2. No record found message would be displayed. |

|  |  |
| --- | --- |
| ID | UC8 |
| Justification | User can get Mark Sheet using this use case where percentage, grades, ranks are being calculated and is being displayed. |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User enter their valid credential. 2. User entered data will be verified. 3. User get access into the system. 4. User navigates to View Result and click on View Result button. 5. System automatically calculates the total obtained marks and gives percentage. 6. System produce rank automatically based on calculated obtained marks and percentage. 7. User will view Ledger view of Mark sheet. |
| Alternative Flow | 1. User can view in Mark Sheet Format by clicking on Print View. 2. User can download in pdf format. 3. User can print hard copy by local printer. |

**Title: “Generate Mark Sheet”**

**Title: “User Manual”**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| ID | UC9 |
| Justification | A use case where user will get the help and guidance to deal with |
| Primary Actor(s) | Admin or Normal User |
| Supporting Actor(s) | N/A |
| Primary Flow | 1. User logged into the system by giving input their valid credential. 2. User will navigate to View User manual. 3. System displays the Manual Page. 4. User read the manual displayed. |

# **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.

|  |  |  |
| --- | --- | --- |
| **Nouns** | **Adjectives** | **Verbs** |
| Mark Sheet Generator | Name | Add |
| System | Address | Update |
| Students | Email | Delete |
| User Type | Contact | Create |
| Admin | Rank | View |
| Examination | Marks obtained | Generate |
| Result | Section | Produces |
| Subjects | Parents Name |  |
| Class | Date |  |
|  |  |  |

Now we filter the similar, irrelevant values

|  |  |  |
| --- | --- | --- |
| **Nouns** | **Adjectives** | **Verbs** |
| Mark Sheet Generator (Irrelevant) | Name | Add |
| System (Irrelevant) | Address | Update |
| Students | Email | Delete |
| User (User Type is renamed as User ) | Contact | Create |
| Admin (Duplicate with users) | Rank | View |
| Examination | Marks obtained | Generate(Duplicate with Create) |
| Result | Section | Produces (Duplicate with Create) |
| Subjects | Parents Name |  |
| Class | Date |  |

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

|  |  |  |
| --- | --- | --- |
| **Nouns** | **Adjectives** | **Verbs** |
| Students | Name | Add |
| User | Address | Update |
| Examination | Email | Delete |
| Result | Contact | Create |
| Subjects | Rank | View |
| Class | Marks obtained |  |
|  | Section |  |
|  | Parents Name |  |
|  | Date |  |

The Initial Class Diagram of the Mark Sheet Generator is:

