Experiment 1:

STUDENT RESULT MANAGEMENT SYSTEM

Aim: Prepare problem statement for suggested system.

Requirements: Microsoft Word

Theory:

1) Problem statement for Student Result Management System:
A university conducts 8 – semester B.Tech program divided into 4 branches namely CSE, IT, ECE and EEE. The students are offered 6 theory papers and 5 lab papers in all semesters. The theory papers offered in the 4th year are categorized as either ‘Core’ or ‘Elective’. Core papers do not have alternative subjects whereas elective papers have 2 other alternative subjects. Thus, a student can take up any subject out of the 3 given choices.
The evaluation of each subject is done out of 100 marks. For theory paper, the division is 75+25 for externals and internals respectively. For lab paper, the division is 60+40 for externals and internals respectively. During the semester, minor exams are conducted for each semester. Based on the students’ performance in minor exams, assignments, Lab records and attendance, marks are given out of 25 in theory paper and 40 in lab paper. At the end of each semester major exams are conducted in each subject (theory as well as practical). The theory papers are evaluated out of 75 marks and lab papers are evaluated out of 60 marks. Thus, the total marks of a student in a subject are obtained by adding the marks obtained in internal and external evaluation.Every subject has some credits assigned to it. If the total marks of a student are >=50, he/she is considered ‘PASS’ otherwise ‘FAIL’.

It is required to develop a system that will manage information about subjects offered in various semesters, students enrolled in various semesters, electives opted by the students and marks and credit points obtained by the students in different semesters. The system would also have the ability to generate mark sheets for each student semester-wise. Semester-wise detailed marks list and student performance report also need to be generated.

Features:

- Displaying list of students (mention detail)
- Generating mark-sheets semester-wise.
- Calculation of GPA and CGPA by marks entered by administrator on basis of credits allocated to each subject.
- Average overall performance dashboard, and a comparative analysis chart.
- Displaying final evaluation status, PASS/FAIL.

Conclusion:

Problem statement of STUDENT RESULT MANAGEMENT SYSTEM has been written successfully.

Experiment 2

Aim: To draw the ER Diagram for STUDENT RESULT MANAGEMENT SYSTEM.

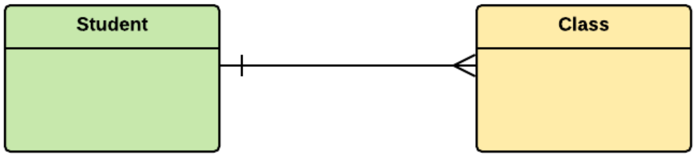
Requirements: Microsoft Word, StarUML

Theory:

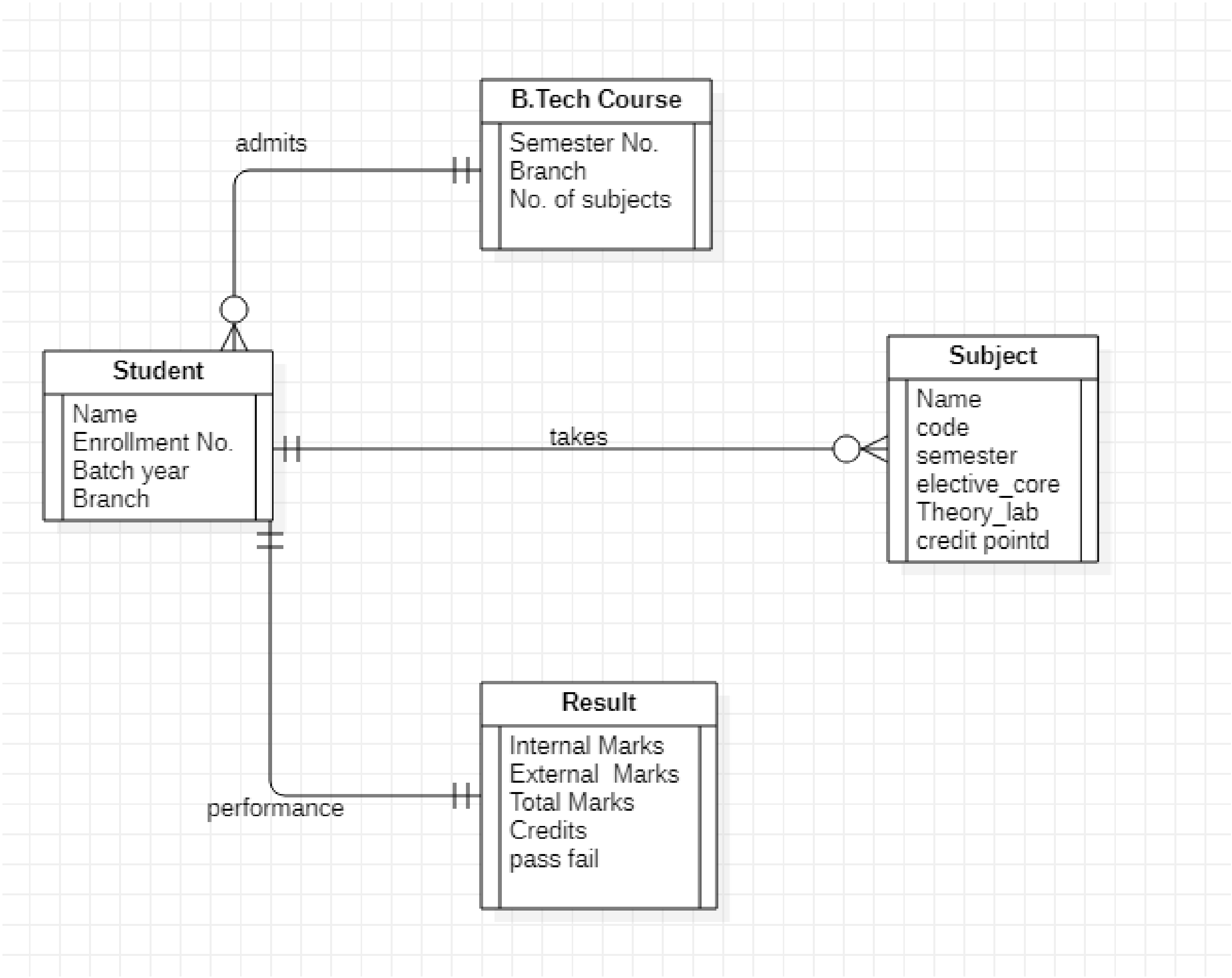
An *Entity Relationship (ER) Diagram* is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. They use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

Entity set:

An entity set is a group of similar kinds of entities. It may contain entities with attributes sharing similar values. Entities are represented by their properties, which are also called attributes. All attributes have their separate values. For example, a student entity may have a name, age, class, as attributes.



ER Diagram:


B.Tech Course
Semester No.
Branch
No. of subjects
admits
Student
Name
Enrollment No.
Batch year
Branch
takes
Subject
Name
code
semester
elective_core
Theory_lab
credit pointd
Result
Internal Marks
External Marks
Total Marks
Credits
pass fail
performance

Experiment 3

Aim: To draw the Level – 0 and Level – 1 Data Flow Diagram (DFD) for STUDENT RESULT MANAGEMENT SYSTEM.

Software Used: Microsoft Word, StarUML

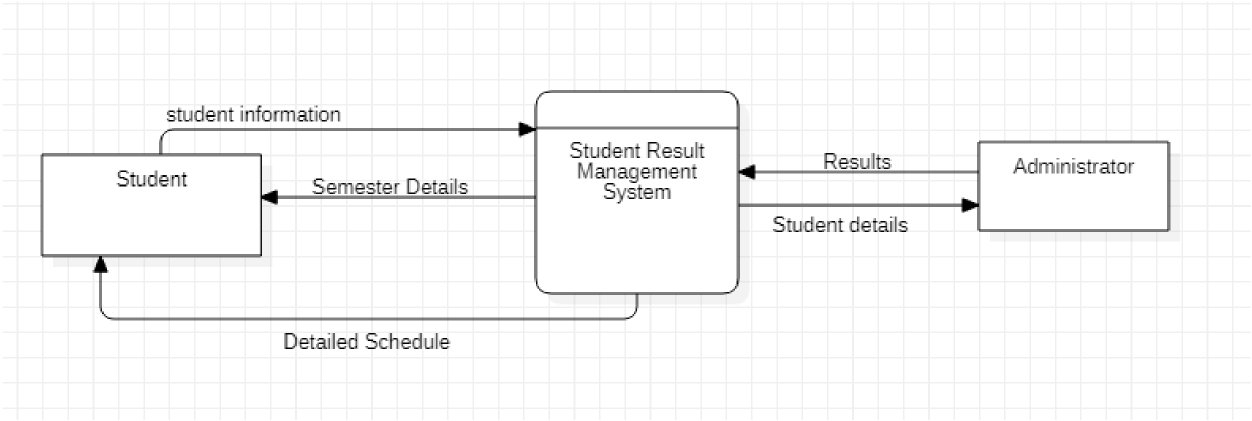
Theory: A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

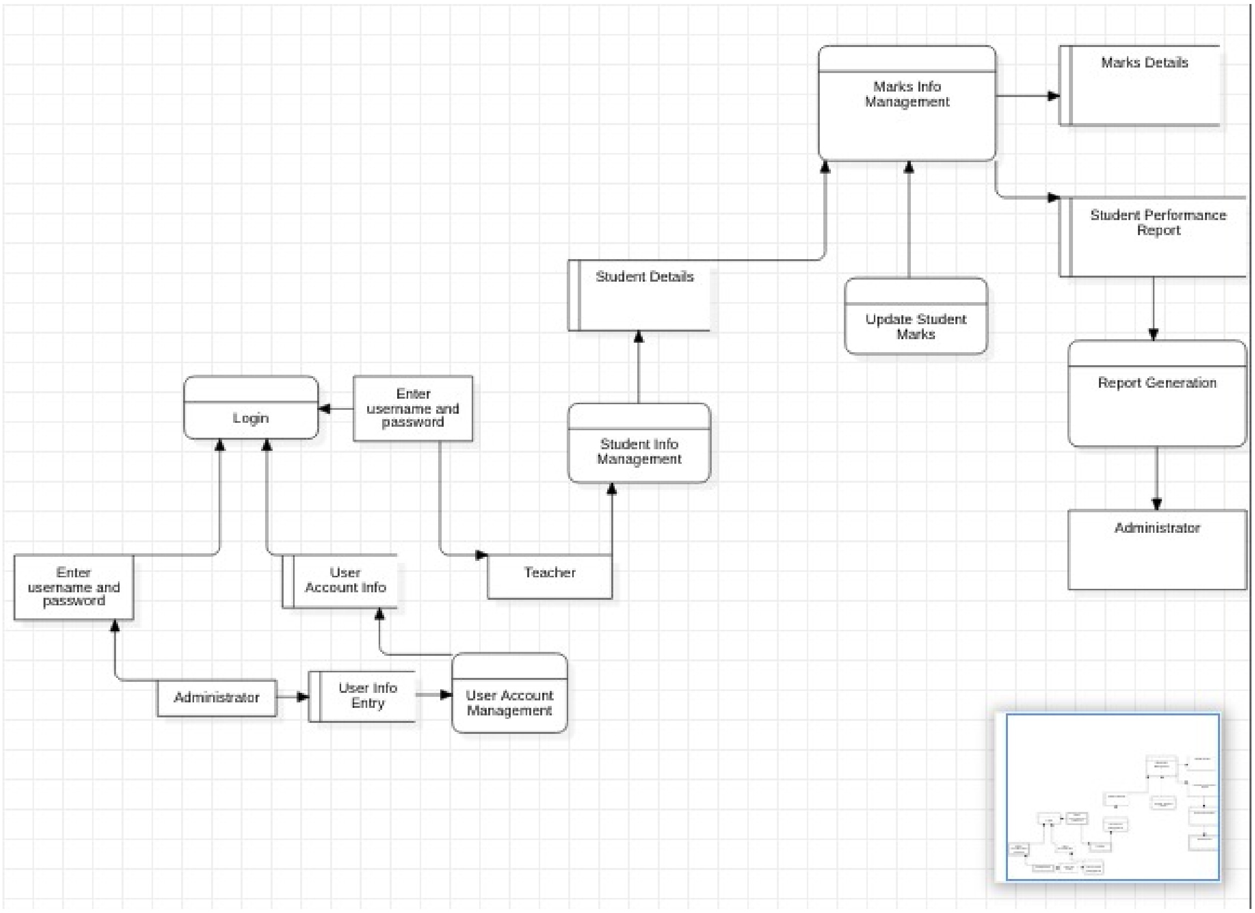
It is usually beginning with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower-level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required.

DFD Diagram:

Level 0 dfd:

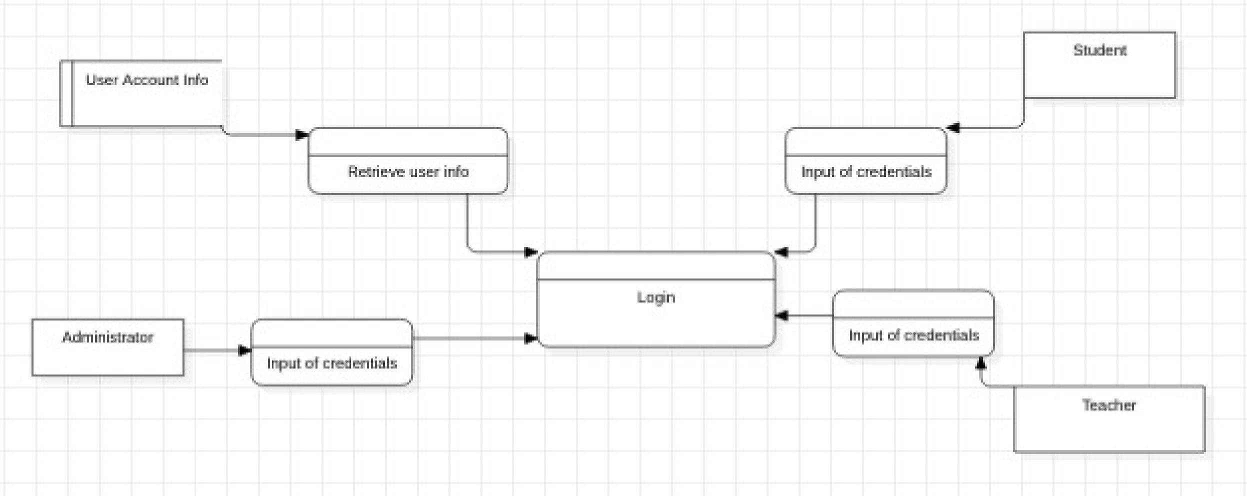


Level 1 dfd:

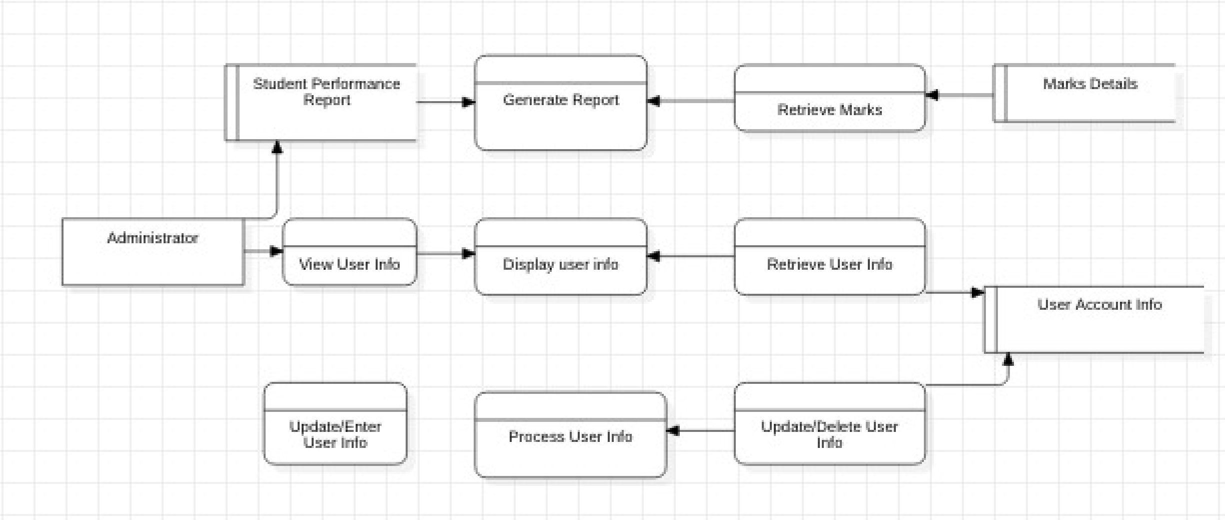


Level 2 dfd:

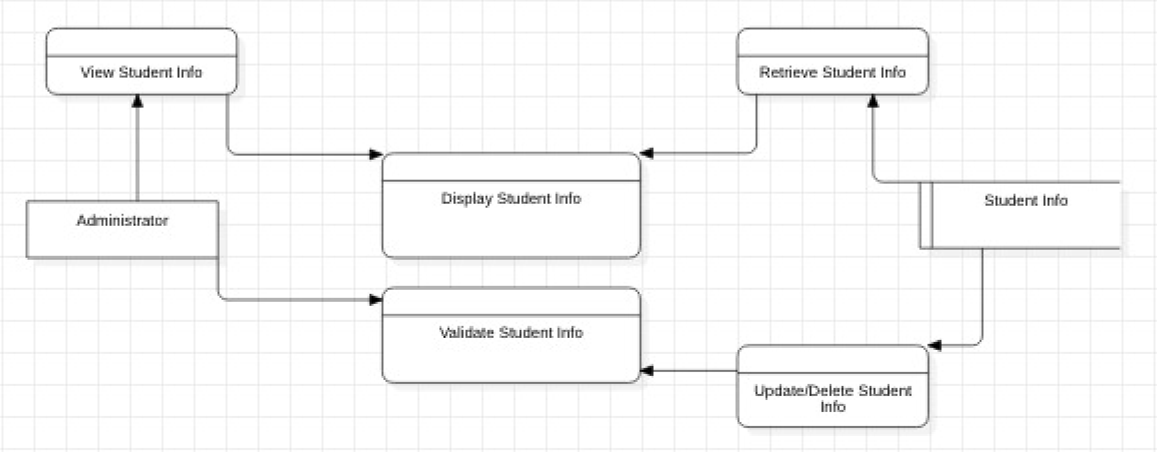
a) Login



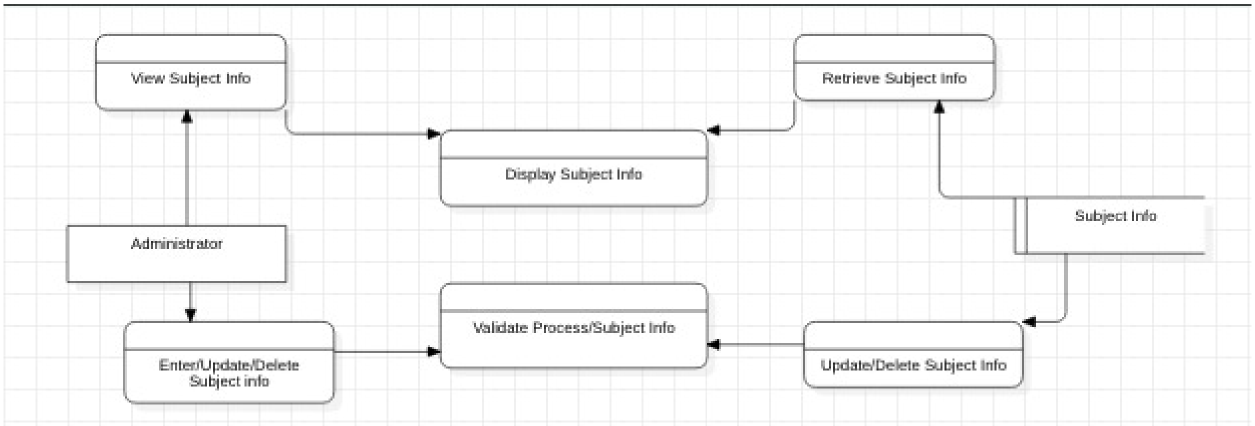
b) User Account Maintenance



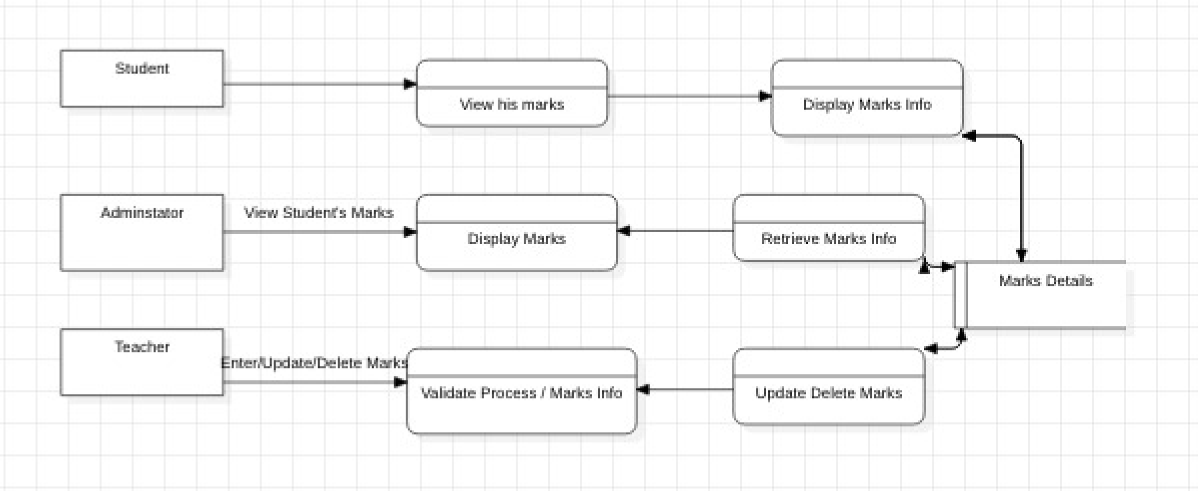
c) Student Information Management



d) Subject Information Management


View Subject Info
Retrieve Subject Info
Display Subject Info
Subject Info
Administrator
Validate Process/Subject Info
Enter/Update/Delete Subject info
Update/Delete Subject Info

e) Marks Information Management



Experiment 4

Aim: To draw a UseCase diagram of the Student Result Management System.

Software Used: Star UML

Theory: A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

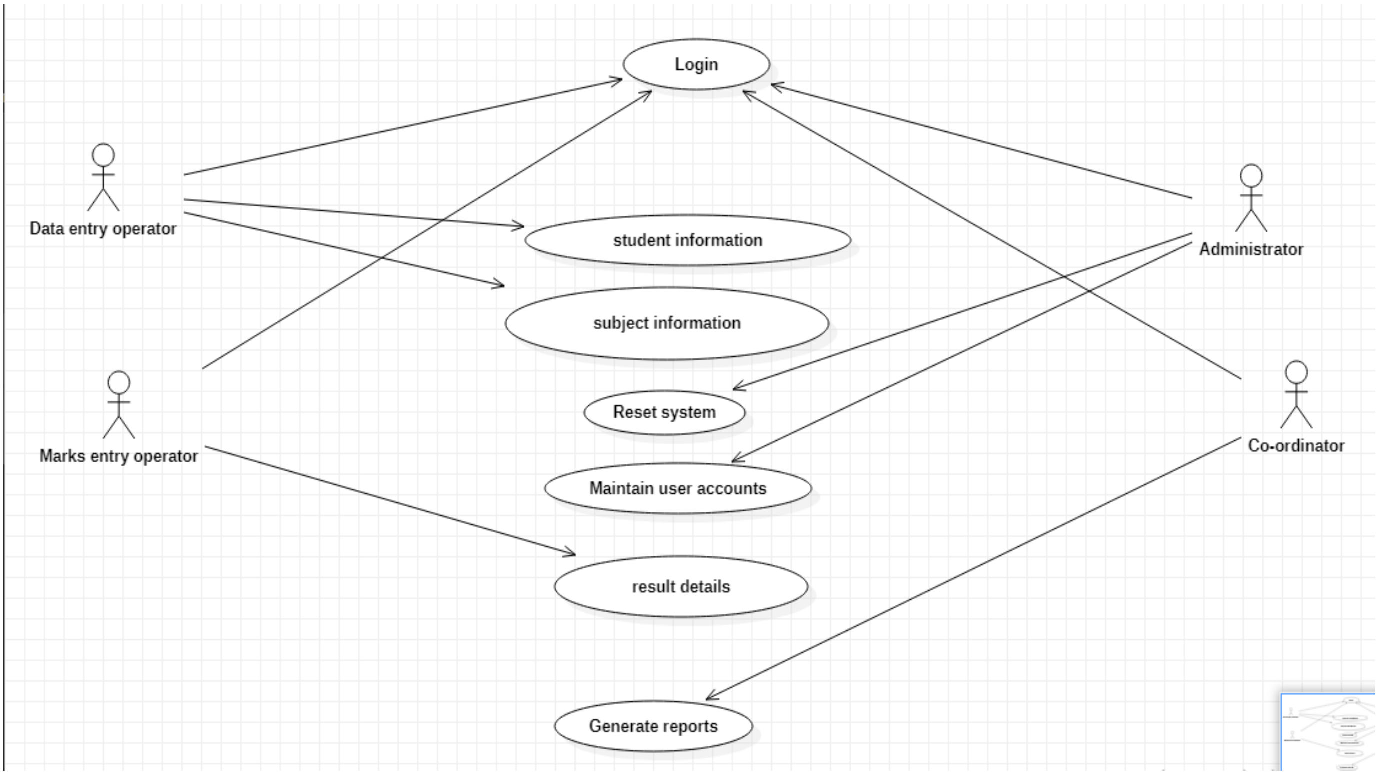
A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

A use case diagram is usually simple. It does not show the detail of the use cases:

- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.

<u>Output</u>:

Use case diagram for Student Result Management System is :


Login
Data entry operator
student information
subject information
Administrator
Reset system
Marks entry operator
Maintain user accounts
Co-ordinator
result details
Generate reports