

3.9 STUDENT RESULT MANAGEMENT SYSTEM—EXAMPLE

A university has decided to engage a software company for the automation of student result management system of its M. Tech. Programme. The following documents are required to be prepared.

- (i) Problem statement
- (ii) Context diagram
- (iii) Data flow diagrams
- (iv) ER diagrams
- (v) Use case diagram
- (vi) Use cases
- (vii) SRS as per IEEE std. 830-1993

These seven documents may provide holistic view of the system to be developed. The SRS will act as contract document between developers (software company) and client (University).

3.9.1 Problem Statement

The problem statement is the first document which is normally prepared by the client. It only, gives superficial view of the system as per client's perspective and expectations. It is the input to the requirement engineering process where final product is the SRS.

The problem statement of student result management system of M. Tech. (Information Technology) Programme of a University is given below:

"A University conducts a 4-semester M. Tech. (IT) program. The students are offered four theory papers and two Lab papers (practicals) during Ist, IIInd and IIIrd semesters. The theory papers offered in these semesters are categorized as either 'Core' or 'Elective'. Core papers do not have an alternative subject, whereas elective papers have two other alternative subjects. Thus, a student can study any subject out of the 3 choices available for an elective paper.

In Ist, IIInd and IIIrd semesters, 2 core papers and 2 elective papers are offered to each student. The students are also required to submit a term paper/minor project in IIInd and IIIrd semesters each. In IVth semester the students have to give a seminar and submit a dissertation on a topic/subject area of their interest.

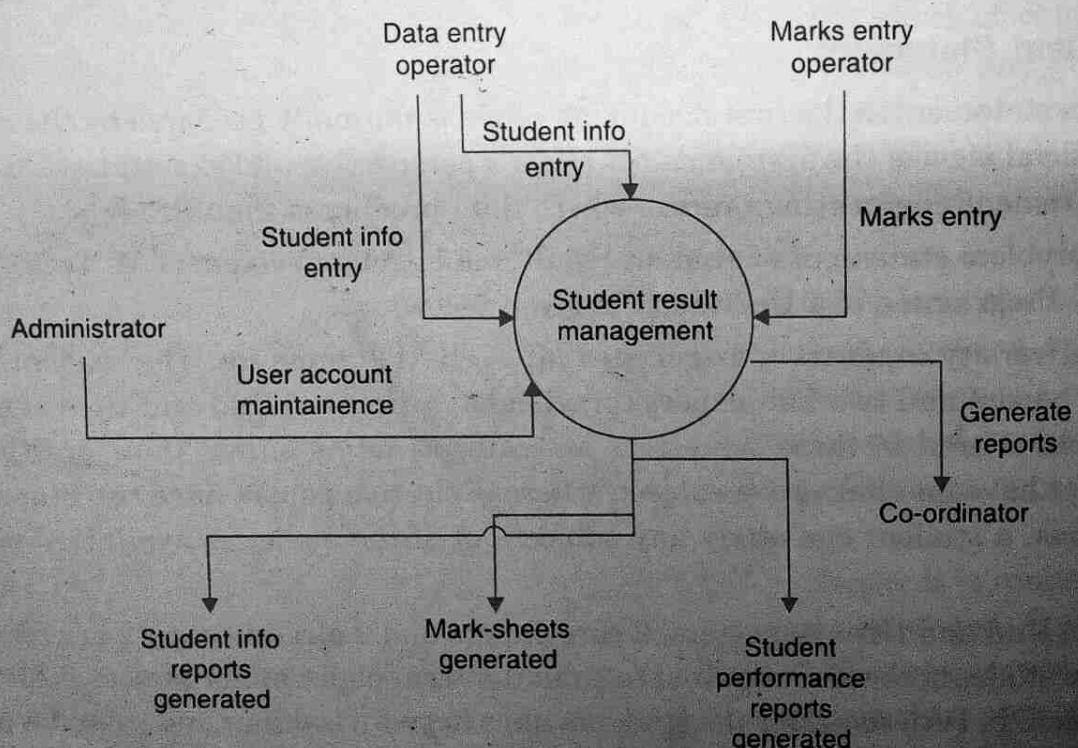
The evaluation of each subject is done out of 100 marks. During the semester, minor exams are conducted for each semester. Students are also required to submit assignments as directed by the corresponding faculty and maintain Lab records for practicals. Based on the students' performance in minor exams, assignments, Lab records and their attendance, marks out of 40 are given in each subject and practical paper. These marks out of 40 account for internal evaluation of the students. At the end of each semester major exams are conducted in each subject (theory as well as practical). These exams are evaluated out of 60 marks and account for external evaluation of the students. 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 credit points assigned to it. If the total marks of a student are $>= 50$ in a subject, he/she is considered 'Pass' in that subject otherwise the student is considered 'Fail' in that subject. If a student passes in a subject, he/she earns all the credit points assigned to that subject, but if the student fails in a subject he/she does not earn any credit point in that subject. At any time, the latest information about subjects being offered in various semesters and their credit points can be obtained from university's website.

It is required to develop a system that will manage information about subjects offered in various semesters, students enrolled in various semesters, elective (s) opted by various students in different semesters, marks and credit points obtained by students in different semesters. The system should also have the ability to generate printable mark-sheets for each student. Semester-wise detailed mark lists and student performance reports also need to be generated."

3.9.2 Context Diagram

The context diagram is given below:



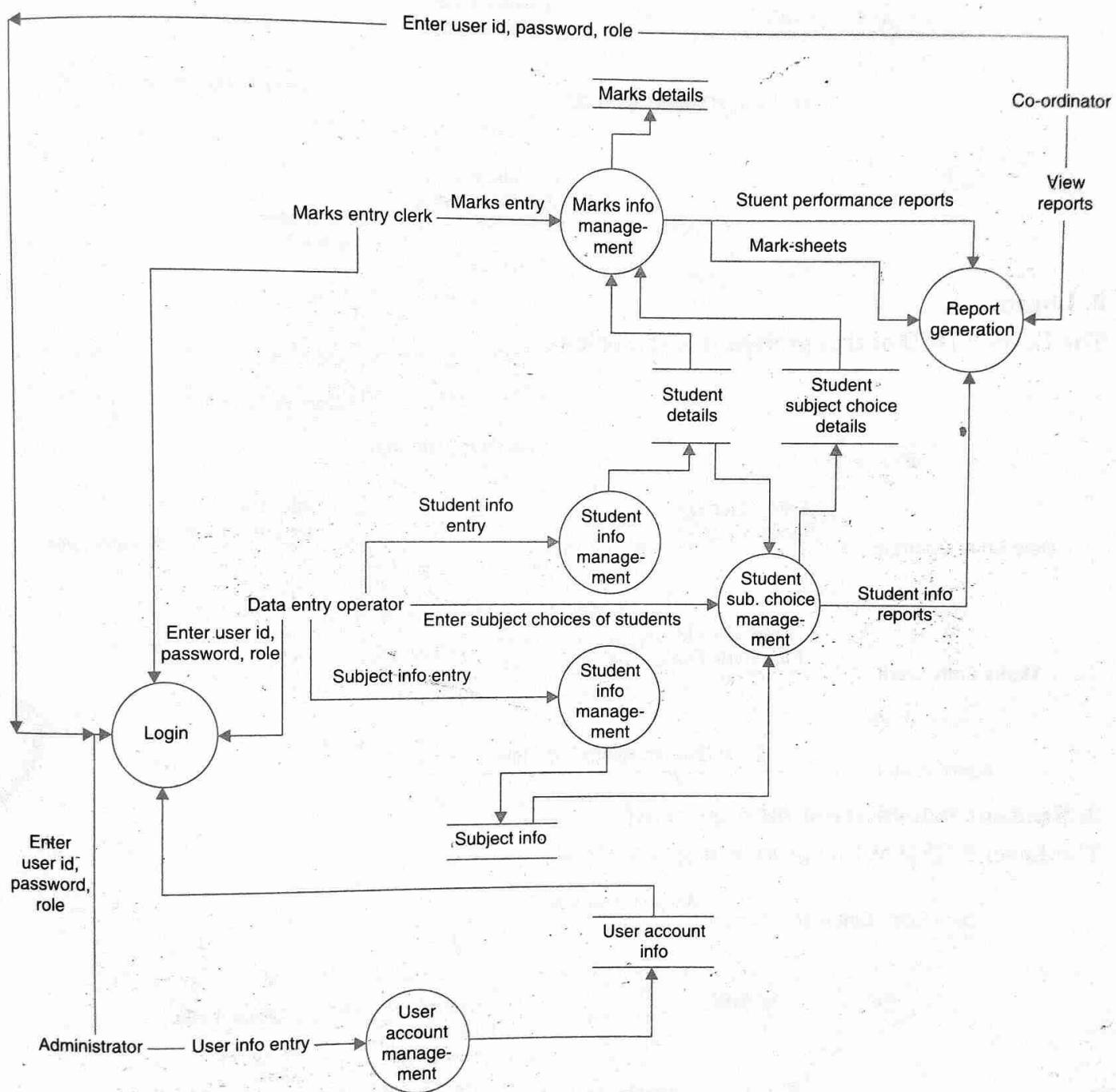
The following persons are interacting with the "student result management system"

- (i) Administrator
- (ii) Marks entry operator
- (iii) Data entry operator
- (iv) Co-ordinator

3.9.3 Level-n DFD

Level-1 DFD

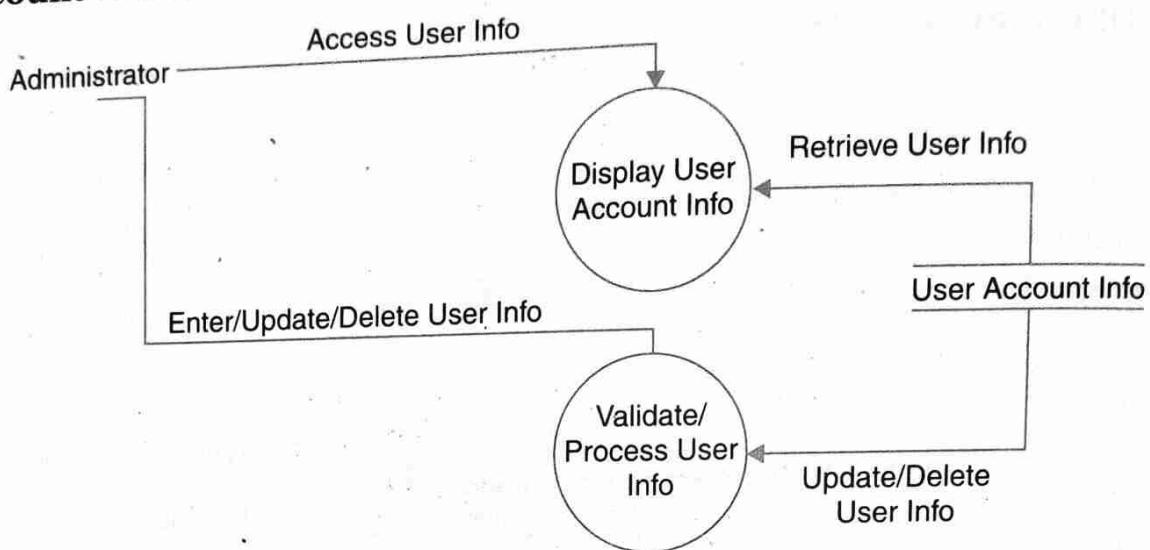
The Level-1 DFD is given below:



Level 1 DFD of result management system

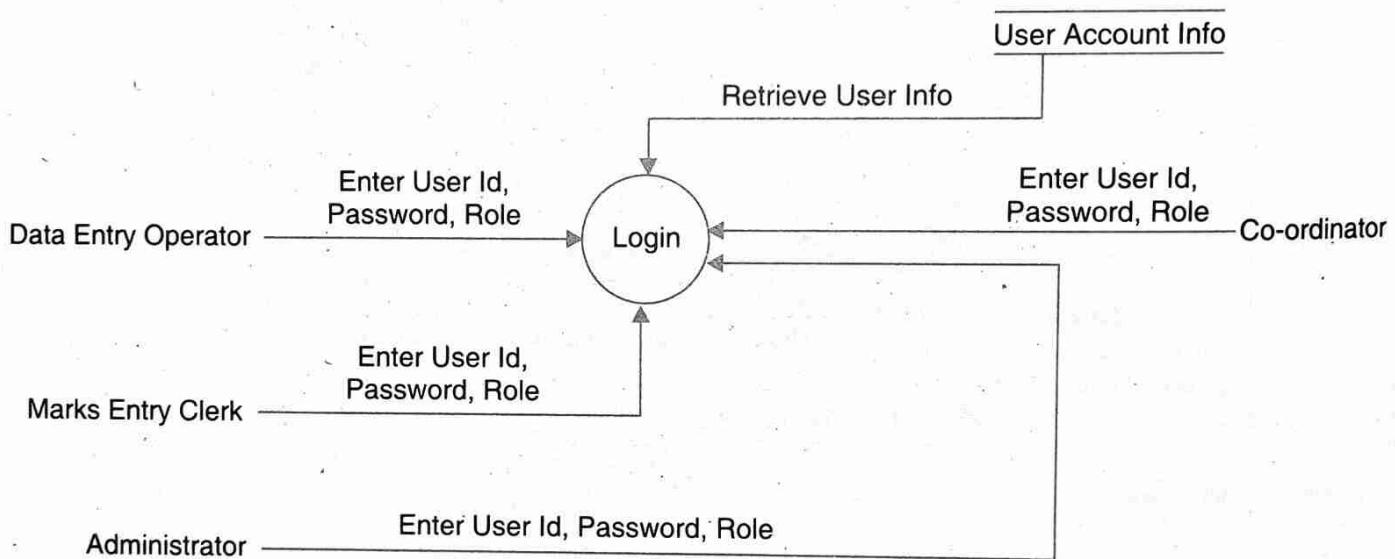
Level-2 DFDs

1. User account maintenance



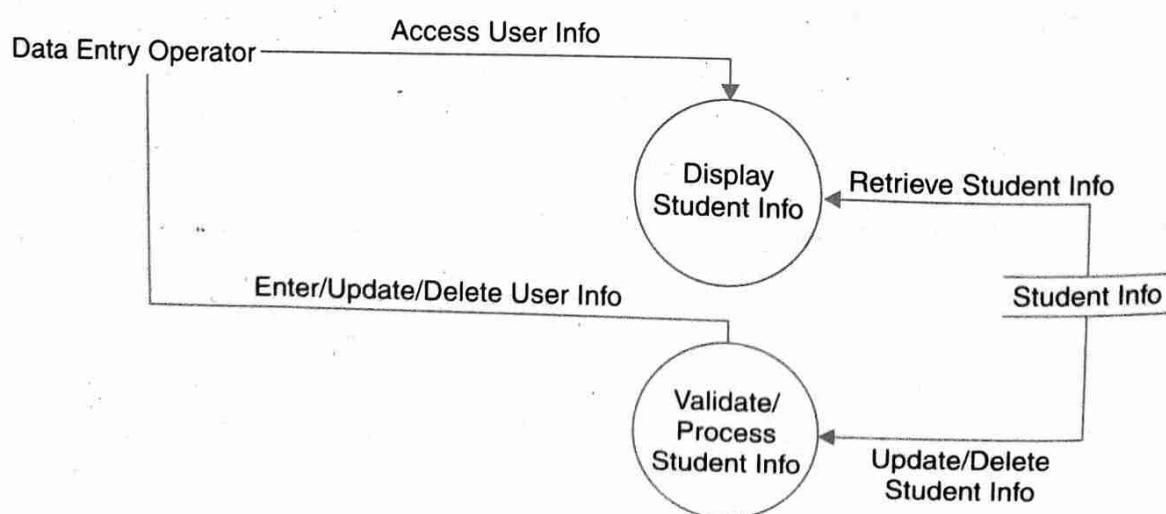
2. Login

The Level 2 DFD of this process is given below:



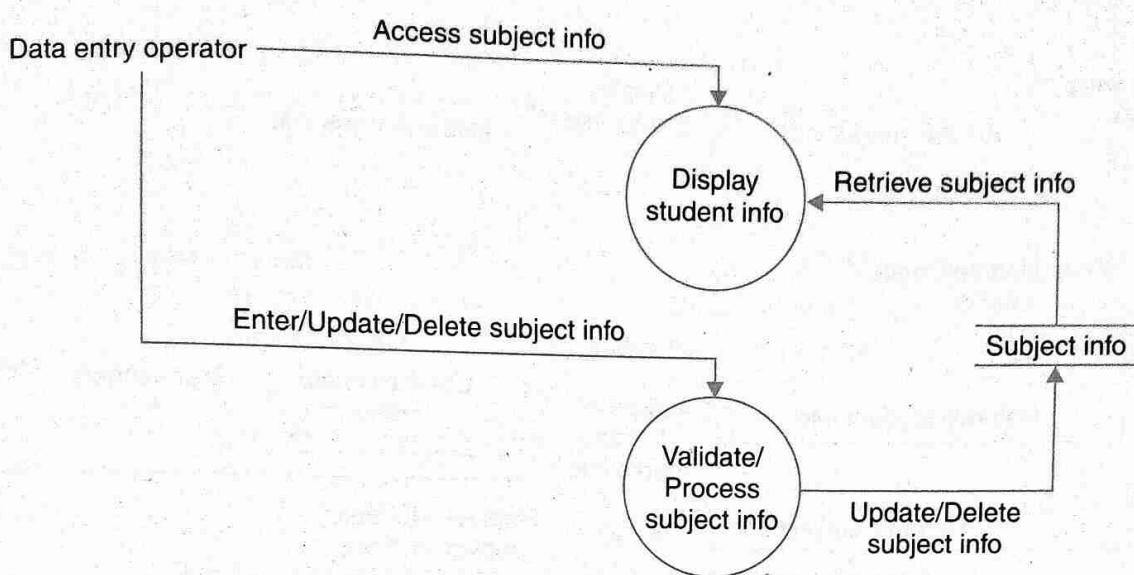
3. Student information management

The Level 2 DFD of this process is given below:



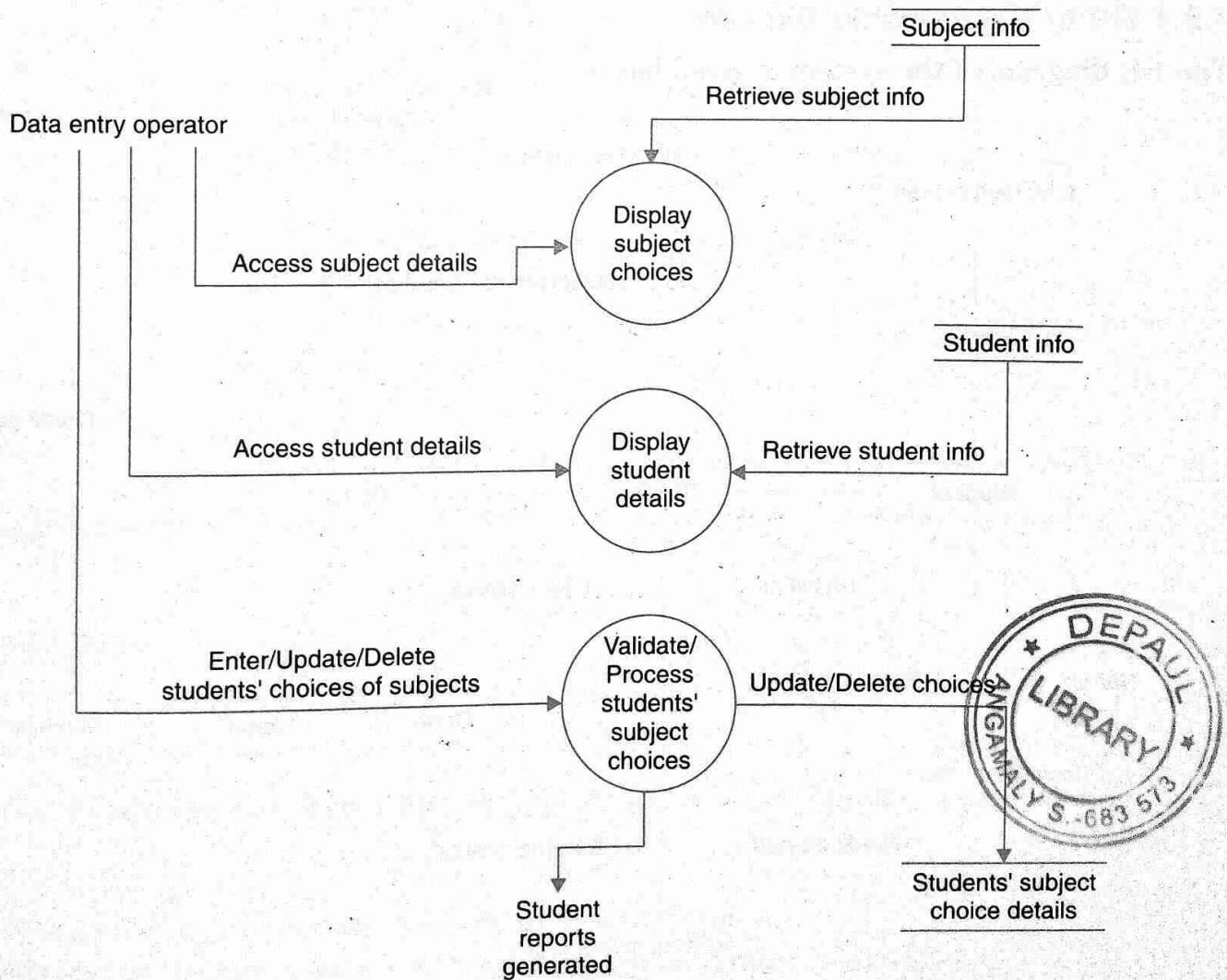
4. Subject information management

The Level 2 DFD of this process is given below:



5. Students' subject choice management

The Level 2 DFD of this process is given below:



1.3 Flow of Events

1.3.1 Basic Flow

This use case starts when the actor wishes to Login to the Student Result Management System.

1. The system requests that the actor enter his/her name, password and role. The role can be any one of Data Entry Operator, Marks Entry Clerk, Co-ordinator, and Administrator.
2. The actor enters his/her name, password and role.
3. The system validates the entered name, password, role and logs the actor into the system.

1.3.2 Alternative Flows

1.3.2.1 Invalid Name/Password/Role

If in the Basic Flow, the actor enters an invalid name, password and/or role, the system displays an error message. The actor can choose to either return to the begining of the Basic Flow or cancel the login, at which point the use case ends.

1.4 Special Requirements

None

1.5 Pre-Conditions

All users must have a User Account (*i.e.*, User ID, Password and Role) created for them in the system (through the Administrator), prior to executing the use cases.

1.6 Post-Conditions

If the use case was successful, the actor is logged into the system. If not, the system state is unchanged.

If the actor has the role 'Data Entry Operator' he/she will have access to only screens corresponding to the Subject Info Maintenance, Student Info Maintenance and Students' Subject Choice Info Maintenance modules of the system.

If the actor has the role 'Marks Entrey Clerk', he/she will have access to only screens corresponding to the Marks Info Maintenance module of the system. If the actor has the role 'Co-ordinator', he/she will only be able to view/print the various reports generated by the system.

If the actor has the role 'Administrator' he/she will have access to only screens corresponding to User Account maintenance module and Reset System feature of the system.

1.7 Extension Points

None

2 Maintain Student Information

2.1 Brief Description

This use case allows the actor with role 'Data Entry Operator' to maintain student information. This includes adding, changing and deleting student information from the system.

2.2 Actors

The following actor (s) interact and participate in this use case:

Data Entry Operator.

2.3 Flow of Events

2.3.1 Basic Flow

This use case starts when the Data Entry Operator wishes to add, change, and/or delete student information from the system.

1. The system requests that the Data Entry Operator specify the function he/she would like to perform (either Add a Student, Update a Student, or Delete a Student).
2. Once the Data Entry Operator provides the requested information, one of the sub-flows is executed.
 - If the Data Entry Operator selected "Add a Student", the **Add a Student** sub-flow is executed.
 - If the Data Entry Operator selected "Update a Student", the **Update a Student** sub-flow is executed.
 - If the Data Entry Operator selected "Delete a Student", the **Delete a Student** sub-flow is executed.

2.3.1.1 Add a Student

1. The system requests that the Data Entry Operator enter the student information. This includes:
 - (a) Name
 - (b) Enrollment Number—should be unique for every student
 - (c) Year of Enrollment
2. Once the Data Entry Operator provides the requested information, the student is added to the system and an appropriate message is displayed.

2.3.1.2 Update a Student

1. The system requests that the Data Entry Operator enters the student enrollment number.
2. The Data Entry Operator enters the student enrollment number. The system retrieves and displays the student information.
3. The Data Entry Operator makes the desired changes to the student information. This includes any of the information specified in the **Add a Student** sub-flow.
4. Once the Data Entry Operator updates the necessary information, the system updates the student record with the updated information.

2.3.1.3 Delete a Student

1. The system requests that the Data Entry Operator enters the student enrollment number.
2. The Data Entry Operator enters the student enrollment number. The system retrieves and displays the student information.
3. The system prompts the Data Entry Operator to confirm the deletion of the student.
4. The Data Entry Operator confirms the deletion.
5. The system deletes the student record.

2.3.2 Alternative Flows

2.3.2.1 Student Not Found

If in the **Update a Student** or **Delete a Student** sub-flows, a student with the specified enrollment number does not exist, the system displays an error message. The Data Entry Operator can then enter a different enrollment number or cancel the operation, at which point the use case ends.

2.3.2.2 Update Cancelled

If in the **Update a Student** sub-flow, the Data Entry Operator decides not to update the student information, the update is cancelled and the **Basic Flow** is re-started at the beginning.

2.3.2.3 Delete Cancelled

If in the **Delete a Student** sub-flow, the Data Entry Operator decides not to delete the student information, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

2.4 Special Requirements

None

2.5 Pre-Conditions

The Data Entry Operator must be logged onto the system before this use case begins.

2.6 Post-Conditions

If the use case was successful, the student information is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

2.7 Extension Points

None

3 Maintain Subject Information

3.1 Brief Description

This use case allows the actor with role 'Data Entry Operator' to maintain subject information. This includes adding, changing and deleting subject information from the system.

3.2 Actors

The following actor (s) interact and participate in this use case:
Data Entry Operator

3.3 Flow of Events

3.3.1 Basic Flow

This use case starts when the Data Entry Operator wishes to add, change, and/or delete subject information from the system.

1. The system requests that the Data Entry Operator specify the function he/she would like to perform (either Add a Subject, Update a Subject, or Delete a Subject).
2. Once the Data Entry Operator provides the requested information, one of the sub-flows is executed.
 - If the Data Entry Operator selected “Add a Subject”, the **Add a Subject** sub-flow is executed.
 - If the Data Entry Operator selected “Update a Subject”, the **Update a Subject** sub-flow is executed.
 - If the Data Entry Operator selected “Delete a Subject”, the **Delete a Subject** sub-flow is executed.

3.2.1.1 Add a Subject

1. The system requests that the Data Entry Operator enters the subject information. This includes:
 - (a) Name of the subject
 - (b) Subject Code—should be unique for every subject
 - (c) Semester
 - (d) Subject Type—can be Core 1/Core 2/Dissertation/Elective 1/Elective 2/Lab 1/Lab 2/Minor Project/Seminar/Term Paper.
 - (e) Credits.
2. Once the Data Entry Operator provides the requested information, the subject is added to the system and an appropriate message is displayed.

3.3.1.2 Update a Subject

1. The system requests that the Data Entry Operator enters the subject code.
2. The Data Entry Operator enters the subject code. The system retrieves and displays the subject information.
3. The Data Entry Operator makes the desired changes to the subject information. This includes any of the information specified in the **Add a Subject** sub-flow.
4. Once the Data Entry Operator updates the necessary information, the system updates the subject record with the updated information.

3.3.1.3 Delete a Subject

1. The system requests that the Data Entry Operator enter the subject code.
2. The Data Entry Operator enters the subjects code. The system retrieves and displays the subject information.
3. The system prompts the Data Entry operator to confirm the deletion of the subject.
4. The Date Entry Operator confirms the deletion.
5. The system deletes the subject record.

3.3.2 Alternative Flows

3.3.2.1 Subject Not Found

If in the **Update a Subject** or **Delete a Subject** sub-flows, a subject with the specified subject code does not exist, the system displays an error message. The Data Entry Operator can then enter a different subject code or cancel the operation, at which point the use case ends.

3.3.2.2 Update Cancelled

If in the **Update a Subject** sub-flow, the Data Entry Operator decides not to update the subject information, the update is cancelled and the **Basic Flow** is re-started at the beginning.

3.3.2.3 Delete Cancelled

If in the **Delete a Subject** sub-flow, the Data Entry Operator decides not to delete the subject information, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

3.4 Special Requirements

None

3.5 Pre-Conditions

The Data Entry Operator must be logged onto the system before this use case begins.

3.6 Post-Conditions

If the use case was successful, the student information is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

3.7 Extension Points

None

4 Maintain Students' Subject Choice Information

4.1 Brief Description

This use case allows the actor with role 'Data Entry Operator' to maintain information about the choice of different Elective subjects opted by various students. This includes displaying the various available choices of Elective subjects available during a particular semester and updating the information about the choice of Elective Subject (s) opted by different students of that semester.

4.2 Actors

The following actor (s) interact and participate in this use case:
Data Entry Operator

4.3 Flow of Events

4.3.1 Basic Flow

This use case starts when the Data Entry Operator wishes to update students' Subject Choice information from the system.

1. The system requests that the Data Entry Operator specify the semester and enrollment year of students, for which the Students' Subject Choices have to be updated.
2. Once the Data Entry Operator provides the requested information, the system displays the list of available choices for Elective I and Elective II subjects for that semester and the list of students enrolled in the given enrollment year (alongwith their existing subject choices, if any).
3. The system requests that the Data Entry Operator specify the information regarding Students' Subject Choices. this includes
 - (a) Student's Enrollment Number
 - (b) Student's Choice for Elective I subject (the corresponding subject code)
 - (c) Student's Choice for Elective II subject (the corresponding subject code).
4. Once the Data Entry Operator provides the requested information, the information regarding Student's Subject Choices is added/updated in the system and an appropriate message is displayed.

4.3.2 Alternative Flows

4.3.2.1 Subject Information Does Not Exist

If no or incomplete subject information exists in the system for the semester specified by the Data Entry Operator, the system displays an error message. The Data Entry Operator can then enter a different semester or cancel the operation, at which point the use case ends.

4.3.2.2 Student Information Does Not Exist

If no student information exists in the system for the enrollment year specified by the Data entry Operator, the system displays an error message. The Data Entry Operator can then enter a different enrollment year or cancel the operation, at which point the use case ends.

4.3.2.3 Incorrect Choice Entered for Elective I/Elective II Subjects

If the subject code entered by the Data Entry Operator for Elective I/Elective II subject does not exist in the system, the system displays an error message.

The Data Entry Operator can then enter the correct subject code or cancel the operation, at which point the use case ends.

4.3.2.4 Update Cancelled

If in the **Basic Flow**, the Data Entry Operator decides not to update the subject information, the update is cancelled and the **Basic Flow** is re-started at the beginning.

4.4 Special Requirements

None

4.5 Pre-Conditions

The Data Entry Operator must be logged onto the system before this use case begins.

4.6 Post-Conditions

If the use case was successful, information about students' choices for opting different Elective Subjects is added/updated in the system. Otherwise, the system state is unchanged.

4.7 Extension Points

None

5 Maintain Result Details

5.1 Brief Description

This use case allows the actor with role 'Marks Entry Clerk' to maintain subject-wise marks information of each student, in different semesters. This includes adding, changing and deleting marks information from the system.

5.2 Actors

The following actor (s) interact and participate in this use case:
Marks Entry Clerk.

5.3 Flow of Events

5.3.1 Basic Flow

This use case starts when the Marks Entry Clerk wishes to add, change, and/or delete marks information from the system.

1. The system requests that the Marks Entry Clerk specify the function he/she would like to perform (either Add Marks, Update Marks, Delete Marks, or Generate Mark-sheet).
2. Once the Marks Entry Clerk provides the requested information, one of the sub-flows is executed.
 - If the Marks Entry Clerk selected "Add Marks", the **Add Marks** sub-flow is executed.
 - If the Marks Entry Clerk selected "Update Marks", the **Update Marks** sub-flow is executed.
 - If the Marks Entry Clerk selected "Delete Marks", the **Delete Marks** sub-flow is executed.
 - If the Marks Entry Clerk selected "Generate Mark-sheet", the **Generate Mark-sheet** sub-flow is executed.

5.3.1.1 Add Marks Record

1. The system requests that the Marks Entry Clerk enters the marks information. This includes:

- (a) Selecting a semester
 - (b) Selecting a subject code
 - (c) Selecting the student enrollment number
 - (d) Entering the internal/external marks for that semester, subject code and enrollment number.
2. Once the Marks Entry Clerk provides the requested information, the system saves the marks and an appropriate message is displayed.

5.3.1.2 Update Marks Record

1. The system requests the Marks Entry Clerk to make following entries:
 - (a) Selecting the semester
 - (b) Selecting the subject code for which marks have to be updated
 - (c) Selecting the student enrollment number.
2. Once the Marks Entry Clerk provides the requested information, the system retrieves and displays the corresponding marks details.
3. The Marks Entry Clerk makes the desired changes to the internal/external marks details.
4. The system updates the marks record with the changed information.

5.3.1.3 Delete Marks Record

1. The system requests the Marks Entry Clerk to make following entries:
 - (a) Selecting the semester
 - (b) Selecting the subject code for which marks have to be updated
 - (c) Selecting the student enrollment number.
2. Once the Marks Entry Clerk provides the requested information, the system retrieves and displays the corresponding marks record from the database.
3. The system verifies if the Marks Entry Clerk wishes to proceed with the deletion of the record. Upon confirmation, the record is deleted from the system.

5.3.1.4 Compute Result

1. Once all the marks are added to the database, the result is computed for each student.
2. If the student has scored more than 50% in a subject, the associated credit points are allotted to that student.
3. The average percentage marks are calculated for the student and his/her division is also derived based on the percentage.

5.3.1.5 Generate Mark-Sheet

1. The system requests that the Marks Entry Clerk specify the Enrollment Number of the student and the semester for which mark-sheet is to be generated.
2. Once the Marks Entry Clerk provides the requested information, the system generates a printable mark-sheet for the specified student and displays it.
3. The Marks Entry Clerk can then issue a print request for the mark-sheet to be printed.

5.3.2 Alternative Flows

5.3.2.1 Record Not Found

If in the **Update Marks**, **Delete Marks** or **Generate Mark-sheet** sub-flows, a record with the specified information does not exist, the system displays an error message. The Marks Entry Clerk can then enter different information for retrieving the record or cancel the operation, at which point the use case ends.

5.3.2.2 Update Cancelled

If in the **Update Marks** sub-flow, the Marks Entry Clerk decides not to update the marks, the update is cancelled and the **Basic Flow** is re-started at the beginning.

5.3.2.3 Delete Cancelled

If in the **Delete Marks** sub-flow, the Marks Entry Clerk decides not to delete the marks, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

5.4 Special Requirements

None

5.5 Pre-Conditions

The Marks Entry Clerk must be logged onto the system before this use case begins.

5.6 Post-Conditions

If the use case was successful, the marks information is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

5.7 Extension Points

None

6 Generate Reports

6.1 Brief Description

This use case allows the actor with role 'Co-ordinator' to generate various reports. The following reports can be generated:

- (a) Student List Report
- (b) Students' Subject Choices List Report
- (c) Marks List Report
- (d) Rank-wise List Report

6.2 Actors

The following actor(s) interact and participate in this use case:
Co-ordinator

6.3 Flow of Events

6.3.1 Basic Flow

This use case starts when the Co-ordinator wishes to generate reports:

1. The system requests the Co-ordinator specify the report he/she would like to generate.
2. Once the Co-ordinator provides the requested information, one of the sub-flows is executed:
 - If the Co-ordinator selected "Student List Report", the **Generate Student List Report** sub-flow is executed.
 - If the Co-ordinator selected "Students' Subject Choices List Report", the **Generate Students' Subject Choices List Report** sub-flow is executed.
 - If the Co-ordinator selected "Marks List Report", the **Generate Marks List Report** sub-flow is executed.
 - If the Co-ordinator selected "Rank-wise List Report", the **Generate Rank-wise List Report** sub-flow is executed.

6.3.1.1 Generate Student List Report

1. The system requests that the Co-ordinator provide the enrollment year for which the Student List report is to be generated.
2. Once the Co-ordinator provides the requested information, the system generates the Student List report, containing the list of students enrolled in the given year.
3. The Co-ordinator can then issue a print request for the report to be printed.

6.3.1.2 Generate Student's Subject Choices List Report

1. The system requests that the Co-ordinator provides the enrollment year and the semester for which the Students' Subject Choices List report is to be generated.
2. Once the Co-ordinator provides the requested information, the system generates the Students' subject Choices List report, containing the choices for Elective I and Elective II subjects, opted by the students of the given enrollment year and semester.
3. The Co-ordinator can then issue a print request for the report to be printed.

6.3.1.3 Generate Marks List Report

1. The system requests that the Co-ordinator provides the enrollment year and the semester for which the Marks List report is to be generated.
2. Once the Co-ordinator provides the requested information, the system generates the Marks List report, containing the marks details of various students in all the subjects for the given enrollment year and semester.
3. The Co-ordinator can then issue a print request for the report to be printed.

6.3.1.4 Generate Rank-wise List Report

1. The system requests that the Co-ordinator provide the enrollment year and the semester for which the Rank-wise List report is to be generated.

2. Once the Co-ordinator provides the requested information, the system generates the Rank-wise List report, containing the percentage wise and rank-wise list of all students (alongwith their total marks and division) for the given enrollment year and semester.
3. The Co-ordinator can then issue a print request for the report to be printed.

6.3.2 Alternative Flows

6.3.2.1 Student Not Found

If no student information exists in the system for the enrollment year specified by the Co-ordinator, the system displays an error message. The Co-ordinator can then enter a different enrollment year or cancel the operation, at which point the use case ends.

6.4 Special Requirements

None

6.5 Pre-Conditions

The Co-ordinator must be logged onto the system before this use case begins.

6.6 Post-Conditions

If the use case was successful, the desired report is generated. Otherwise, the system state is unchanged.

6.7 Extension Points

None.

7 Maintain User Accounts

7.1 Brief Description

This use case allows the actor with role 'Administrator' to maintain User Account. This includes adding, changing and deleting user account information from the system.

7.2 Actors

The following actor (s) interact and participate in this use case:

Administrator.

7.3 Flow of Events

7.3.1 Basic Flow

This use case starts when the Administrator wishes to add, change, and/or delete user account information from the system.

1. The system requests that the Administrator specify the function he/she would like to perform (either Add a User Account, Update a User Account, or Delete a User Account).
2. Once the Administrator provides the requested information, one of the sub-flows is executed.

- If the Administrator selected “Add a User Account”, the **Add a User Account** sub-flow is executed.
- If the Administrator selected “Update a User Account”, the **Update a User Account** sub-flow is executed.
- If the Administrator selected “Delete a User Account”, the **Delete a User Account** sub-flow is executed.

7.3.1.1 Add a User Account

1. The system requests that the Administrator enters the user information. This includes:
 - (a) User Name
 - (b) User ID-should be unique for each user account
 - (c) Password
 - (d) Role
2. Once the Administrator provides the requested information, the user account information is added to the system and an appropriate message is displayed.

7.3.1.2 Update a User Account

1. The system requests that the Administrator enters the User ID.
2. The Administrator enters the User ID. The system retrieves and displays the user account information.
3. The Administrator makes the desired changes to the user account information. This includes any of the information specified in the **Add a User Account** sub-flow.
4. Once the Administrator updates the necessary information, the system updates the user account record with the updated information.

7.3.1.3 Delete a User Account

1. The system requests that the Administrator enters the User ID.
2. The Administrator enters the User ID. The system retrieves and displays the user account information.
3. The system prompts the Administrator to confirm the deletion of the user account.
4. The Administrator confirms the deletion.
5. The system deletes the user account record.

7.3.2 Alternative Flows

7.3.2.1 User Not Found

If in the **Update a User Account** or **Delete a User Account** sub-flows, a user account with the specified User ID does not exist, the system displays an error message. The Administrator can then enter a different User ID or cancel the operation, at which point the use case ends.

7.3.2.2 Update Cancelled

If in the **Update a User Account** sub-flow, the Administrator decides not to update the user account information, the update is cancelled and the **Basic Flow** is re-started at the beginning.

7.3.2.3 Delete Cancelled

If in the **Delete a User Account** sub-flow, the Administrator decides not to delete the user account information, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

7.4 Special Requirements

None

7.5 Pre-Conditions

The Administrator must be logged onto the system before this use case begins.

7.6 Post-Conditions

If the use case was successful, the user account information is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

7.7 Extension Points

None

8. Reset System

8.1 Brief Description

This Use case allows the actor with role 'Administrator' to reset the system by deleting all existing information from the system.

8.2 Actors

The following actor (s) interact and participate in this use case:

Administrator

8.3 Flow of Events

8.3.1 Basic Flow

This use case starts when the Administrator wishes to reset the system.

1. The system requests the Administrator to confirm if he/she wants to delete all the existing information from the system.
2. Once the Administrator provides confirmation, the system deletes all the existing information from the backend database and displays an appropriate message.

8.3.2 Alternative Flows

8.3.2.1 Reset Cancelled

If in the **Basic Flow**, the Administrator decides not to delete the entire existing information, the reset is cancelled and the use case ends.

8.4 Special Requirements

None

8.5 Pre-Conditions

The Administrator must be logged onto the system before this use case begins.

8.6 Post-Conditions

If the use case was successful, all the existing information is deleted from the backend database of the system. Otherwise, the system state is unchanged.

8.7 Extension Points

None

3.9.7 SRS Document

- 1 Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
- 2 Overall Description
 - 2.1 Product Perspective
 - 2.1.1 System Interfaces
 - 2.1.2 User Interfaces
 - 2.1.3 Hardware Interfaces
 - 2.1.4 Software Interfaces
 - 2.1.5 Communications Interfaces
 - 2.1.6 Memory Constraints
 - 2.1.7 Operations
 - 2.1.8 Site Adaptation Requirements
 - 2.2 Product Functions
 - 2.3 User Characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and Dependencies
 - 2.6 Apportioning of Requirements
- 3 Specific Requirements
 - 3.1 External Interface Requirements
 - 3.1.1 User Interfaces
 - Login Screen:
 - Subject Info Parameters Screen:
 - Subject Information Screen:
 - Student Info Parameters Screen:
 - Student Information Screen:
 - Students' Subject Choice Parameters Screen:
 - Students' Subject Choice Information Screen:

(Contd.)...

Marks Entry Parameters Screen:
 Marks Entry Screen:
 Mark-sheet Parameters Screen:
 Students List Report Parameters Screen:
 Marks List Report Parameters Screen:
 Rank-wise List Report Parameters Screen:
 Students' Subject Choices List Report Parameters Screen:

- 3.1.2 Hardware Interfaces
- 3.1.3 Software Interfaces
- 3.1.4 Communications Interfaces
- 3.2 Software Product Features
 - 3.2.1 Subject Information Maintenance
 - Validity Checks
 - Sequencing Information
 - Error Handling/ Response to Abnormal situations
 - 3.2.2 Student Information Maintenance
 - Validity Checks
 - Sequencing Information
 - Error Handling / Response to Abnormal situations
 - 3.2.3 Marks Info Maintenance
 - Validity Checks
 - Sequencing Information
 - Error Handling/ Response to Abnormal situations
 - 3.2.4 Mark sheet Generation
 - 3.2.5 Report Generation
 - Student List Reports
 - Students' Subject Choices List Reports
 - Semester-wise Mark lists
 - Rank-wise List Report
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software System Attributes
 - 3.5.1 Security
 - 3.5.2 Maintainability
 - 3.5.3 Portability
- 3.6 Logical Database Requirements
- 3.7 Other Requirements

1 Introduction

This document aims at defining the overall software requirements for 'Student Result Management System'. Efforts have been made to define the requirements exhaustively and

accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing/using this product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

1.1 Purpose

This specification document describes the capabilities that will be provided by the software application ‘Student Result Management System’. It also states the various required constraints by which the system will abide. The intended audience for this document are the development team, testing team and end users of the product.

1.2 Scope

The software product ‘Student Result Management System’ will be an MIS and Reporting application that will be used for result preparation and management of M. Tech. Program of a University. The application will manage the information about various students enrolled in this course in different years, the subjects offered during different semesters of the course, the students’ choices for opting different subjects, and the marks obtained by various students in various subjects in different semesters. Printable reports regarding list of students, marks obtained by all students in a particular semester and performance of students (rank-wise, percentage-wise, pass/fail, division-wise.) will be generated. The system will also generate printable mark-sheets for individual students.

The application will greatly simplify and speed up the result preparation and management process.

1.3 Definitions, Acronyms, and Abbreviations

Following abbreviations have been used throughout this document:

M.Tech: Master of Technology

IT: Information Technology

DBA: Database Administrator

1.4 References

- (i) *University website*: For information about course structure of M.Tech. Program
- (ii) IEEE Recommended Practice for Software Requirements Specifications—IEEE Std 830-1993

1.5 Overview

The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail.

2 Overall Description

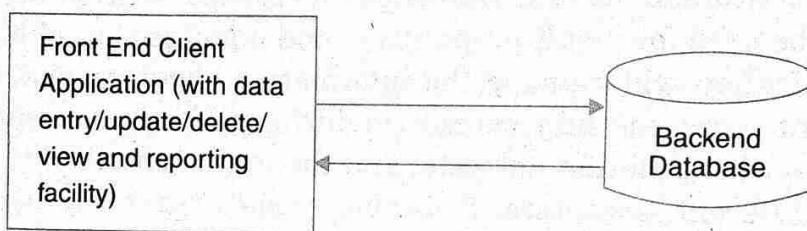
M.Tech. Program is a 4-semester course. The students are offered 4 subjects (theory) and 2 Labs (practical) during first, second and third semesters. Students also have to submit a term paper/minor project in 2nd and 3rd semesters. The fourth semester consists of a seminar and

dissertation. Each subject/lab/term paper/seminar/dissertation has credits associated with it. When a student secures pass marks in a paper he/she also earns all the credit (s) assigned to that paper.

The 'Student Result Management System' will have capability to maintain information about students enrolled in the course, the subjects offered to students during different semesters, the students' choices for opting different Elective subjects (out of the available ones) and the marks obtained by students in different subjects in various semesters. The software will also generate summary reports regarding student information, semester-wise mark lists and performance reports. Printable mark-sheets of individual students will also be generated by the application.

2.1 Product Perspective .

The application will be a windows-based, self-contained and independent software product.



2.1.1 System Interfaces

None

2.1.2 User Interfaces

The application will have a user-friendly and menu based interface. Following screens will be provided:

- (i) A Login screen for entering the username, password and role (Administrator, Data Entry Operator, Marks Entry Clerk, Co-ordinator) will be provided. Access to different screens will be based upon the role of the user.
- (ii) There will be a screen for capturing and displaying information regarding what all subjects are offered during which semester, how many credit points are assigned to that subject and whether the subject is an elective, a core paper, a lab paper, a term paper or a dissertation.
- (iii) There will be a screen for capturing and displaying information regarding various students enrolled for the course in different years.
- (iv) There will be a screen for capturing and displaying information regarding which student is currently enrolled in which semester and what all elective subjects he/she has opted.
- (v) There will be a screen that will capture information regarding which student has scored how many marks (internal + external evaluation) in each subject (in a particular semester). Credits in each subject will be calculated depending upon the marks obtained in that subject.

- (vi) There will be a screen for capturing and displaying information regarding which all user accounts exist in the system, thus showing who all can access the system. The following reports will be generated:
- (i) *Students' List Report*: Printable reports will be generated to show the list of students enrolled in a particular batch year.
 - (ii) *Students' Subject Choices List Report*: For Ist, IInd and IIIrd semester, there will be printable reports showing the different elective subjects opted by various students (enrolled in a particular batch year) of the corresponding semester.
 - (iii) *Marks List Report*: For each semester there will be a printable report showing the subject-wise marks details for all students of that semester.
 - (iv) *Rank-wise List Report*: For each semester there will be a printable report showing the percentage-wise and rank-wise list of students alongwith the division secured.
 - (v) *Mark-sheet*: For each student of each semester, a printable mark-sheet will be generated, showing the subject-wise marks details, Total marks, total credits, Percentage, Pass/Fail status for that student.

2.1.3 Hardware Interfaces

- (i) Screen resolution of at least 800×600 —required for proper and complete viewing of screens. Higher resolution would not be a problem.
- (ii) Support for printer (dot-matrix/deskJet/inkjet etc.—any will do)—that is, appropriate drivers are installed and printer connected printer will be required for printing of reports and mark-sheets.
- (iii) Standalone system or network based—not a concern, as it will be possible to run the application on any of these.

2.1.4 Software Interfaces

- (i) Any windows-based operating system (Windows 95/98/2000/XP/NT)
- (ii) MS Access 2000 as the DBMS—for database. Future release of the application will aim at upgrading to Oracle 8i as the DBMS.
- (iii) Crystal Reports 8—for generating and viewing reports.
- (iv) Visual Basic 6—for coding/developing the software.

Software mentioned in pts. (iii) and (iv) above, will be required only for development of the application. The final application will be packaged as an independent setup program that will be delivered to the client (University in this case).

2.1.5 Communications Interfaces

None

2.1.6 Memory Constraints

At least 64 MB RAM and 2 GB space on hard disk will be required for running the application.

2.1.7 Operations

This product release will not cover any automated housekeeping aspects of the database. The DBA at the client site (*i.e.*, University) will be responsible for manually deleting old/non-required data. Database backup and recovery will also have to be handled by the DBA.

However, the system will provide a 'RESET SYSTEM' function that will delete (upon confirmation from the Administrator) all the existing information from the database.

2.1.8 Site Adaptation Requirements

The terminals at client site will have to support the hardware and software interfaces specified in above sections.

2.2 Product Functions

The system will allow access only to authorised users with specific roles (Administrator, Data Entry Operator, Marks Entry Clerk and Co-ordinator). Depending upon the user's role, he/she will be able to access only specific modules of the system.

A summary of the major functions that the software will perform:

- (i) A Login facility for enabling only authorised access to the system.
- (ii) User (with role Data Entry Operator) will be able to add/modify/delete information about different students that are enrolled for the course in different years.
- (iii) User (with role Data Entry Operator) will be able to add/modify/delete information about different subjects that are offered in a particular semester. The semester-wise list of subjects along with their credit points and type (*i.e.*, elective/core/lab/term paper/dissertation) will also be displayed.
- (iv) User (with role Data Entry Operator) will be able to add/modify/delete information about the Elective subjects opted by different students in different semesters.
- (v) User (with role Marks Entry Clerk) will be able to add/modify/delete information regarding marks obtained by different students in different semesters.
- (vi) User (with role Marks Entry Clerk) will also be able to print mark-sheets of students.
- (vii) User (with role Co-ordinator) will be able to generate Printable reports (as mentioned in section 2.1.2 above).
- (viii) User (with role Administrator) will be able to 'Reset' the system-leading to deletion of all existing information from the backend database.
- (ix) User (with role Administrator) will be able to create/modify/delete new/existing user accounts.

2.3 User Characteristics

- **Educational level:** At least graduate should be comfortable with English language.
- **Experience:** Should be well versed/informed about the course structure of M. Tech. program of University. Entry of marks or their modification can be done only by user who is authorised for this job by the result preparation committee of University.

- *Technical expertise:* Should be comfortable using general-purpose applications on a computer.

2.4 Constraints

- Since the DBMS being used is MS Access 2000, which is not a very powerful DBMS, it will not be able to store a very huge number of records.
- Due to limited features of DBMS being used performance tuning features will not be applied to the queries and thus the system may become slow with the increase in number of records being stored.
- Due to limited features of DBMS being used, database auditing will also not be provided.
- Users at University will have to implement a security policy to safeguard the marks-related information from being modified by unauthorised users (by means of gaining access to the backend database).

2.5 Assumptions and Dependencies

- The number of subjects to be taken up by a student in each semester does not change.
- The subject types (*i.e.*, elective, core, lab, term paper and dissertation) do not change.
- The number of semesters in the M. Tech. Program does not change.

2.6 Apportioning of Requirements

Not Required.

3 Specific Requirements

This section contains the software requirements to a level of detail sufficient to enable designers to design the system, and testers to test that system.

3.1 External Interface Requirements

3.1.1 User Interfaces

The following screens will be provided:

Login screen:

This will be the first screen that will be displayed. It will allow user to access different screens based upon the user's role. Various fields available on this screen will be

- (i) *User ID:* Alphanumeric of length upto 10 characters
- (ii) *Password:* Alphanumeric of length upto 8 characters
- (iii) *Role:* Will have the following values:
Administrator, Marks Entry Clerk, Co-ordinator, Data Entry Operator

Subject info parameters screen:

This screen will be accessible only to user with role Administrator. It will allow the user to enter the semester number for which the user wants to access the subject information.

Subject information screen:

This screen will be accessible only to user with role Administrator. It will allow user to add/modify/delete information about new/existing subject (s) for the semester that was selected in the 'Subject Info Parameters' screen. The list of available subjects for that semester, will also be displayed. Various fields available on this screen will be:

- (i) **Subject Code:** of the format IT-### (# represents a digit)
- (ii) **Subject Name:** Alphanumeric, of length upto 50 characters
- (iii) **Category / Type:** Will have any of the following values:-
Elective 1/Elective 2/Core/Lab/Term paper/Seminar/Dissertation
- (iv) **Credits:** Numeric, will have any value from 0 to 20.

Student info parameters screen:

This screen will be accessible only to user with role Administrator. It will allow the user to enter the Batch Year for which the user wants to access the student information.

Student information screen:

This screen will be accessible only to user role Administrator. It will allow the user to add/modify/delete information about new/existing student (s) for a particular Batch Year. Batch Year-wise list of students will also be displayed. Various fields available on these screens will be:

- (i) **Student Enrollment No:** of the format ##/M.Tech. (IT)/YYYY (# represents a digit and YYYY represents the batch year).
- (ii) **Student Name:** will have only alphabetic letters and length upto 40 characters.
- (iii) **Batch Year:** of the format YYYY (representing the year in which the student enrolled for the course).

Students' subject choice parameters screen:

This screen will be accessible only to user with role Administrator. It will allow the user to enter the Batch Year and the semester number for which the user wants to access the students' subject choice information.

Students' subject choice information screen:

This screen will be accessible only to user with role Administrator. It will allow user to add/modify/delete students' choices for elective subjects of the semester and batch year selected in "Students' Subject Choice Parameters" screen. For the selected semester it will display the list of available choices for Elective I and for Elective II. The screen will display the list of students enrolled during the selected batch year and currently studying in the selected semester and the user will be able to view/add/modify/delete the subject choices for each student in the list.

Marks entry parameters screen:

This screen will be accessible only to user with role Marks Entry Clerk. It will allow the user to enter the Batch Year, the semester number and the Subject for which the user wants to access the marks information.

Marks entry screen:

This screen will be accessible only to user with role Marks Entry Clerk. It will allow user to add/modify/delete information about marks obtained in the selected subjects by different students of that semester who were enrolled in the Batch Year selected in the 'Marks Entry Parameters' Screen. The screen will display the list of students enrolled during the selected batch year and currently studying the selected subject in the selected semester and the user will be able to view/add/modify/delete the marks for each student in the list.

Various fields available on this screens will be:

- (i) *Student Enrollment No.:* will display the enrollment numbers of all students of the selected Batch Year studying the selected subject in the selected semester.
- (ii) *Student Name:* will display the name of the student
- (iii) *Internal Marks:* between 0 and 40
- (iv) *External Marks:* between 0 and 60
- (v) *Total Marks:* sum of Internal Marks and External Marks

Mark-sheet parameters screen:

This screen will be accessible only to user with role Marks Entry Clerk. It will allow the user to enter the Enrollment Number and the semester number of the student for whom the user wants to view/print the mark-sheet.

Students list report parameters screen:

This screen will be accessible only to user with role Co-ordinator. It will allow the user to enter the Batch Year for which the user wants to view/print the students list report.

Marks list report parameters screen:

This screen will be accessible only to user with role Co-ordinator. It will allow the user to enter the Batch Year and the semester for which the user wants to view/print the marks list report.

Rank-wise list report parameters screen:

This screen will be accessible only to user with role Co-ordinator. It will allow the user to enter the Batch Year and the semester for which the user wants to view/print the rank-wise list report.

Students' subject choices list report parameters screen:

This screen will be accessible only to user with role Co-ordinator. It will allow the user to enter the Batch Year and the semester for which the user wants to view/print the students' subject choices list report.

3.1.2 Hardware Interfaces

As stated in Section 2.1.3.

3.1.3 Software Interfaces

As stated in section 2.1.4.

3.1.4 Communications Interfaces

None

3.2 System Features

3.2.1 Subject Information Maintenance

Description

The system will maintain information about various subjects being offered during different semesters of the course. The following information would be maintained for each subject:

Subject code, Subject name, Subject Type (Core/Elective 1/Elective 2/Lab 1/Lab 2/Term Paper/Minor Project/Dissertation/Seminar), Semester, Credits.

The system will allow creation/modification/deletion of new/existing subjects and also have the ability to list all the available subjects for a particular semester.

Validity checks

- (i) Only user with role Data Entry Operator will be authorised to access the Subject Information Maintenance module.
- (ii) Ist, IIInd and IIIrd semesters will have 2 core papers, 2 Elective papers, 2 Lab papers and 1 term paper/Minor Project.
- (iii) Ist, IIInd and IIIrd semesters will have 3 choices (subjects) each of type Elective 1 and of type Elective 2.
- (iv) IVth semester will have only 1 dissertation and 1 seminar.
- (v) No two semesters will have the same subject i.e., A subject will be offered only in a particular semester.
- (vi) Subject code will be unique for every subject.
- (vii) Subject code cannot be blank.
- (viii) Subject name cannot be blank.
- (ix) Credits cannot be blank.
- (x) Credits can have value only between 0 and 20.
- (xi) Subject Type cannot be blank.
- (ii) Semester cannot be blank.

Sequencing information

Subject info for a particular semester will have to be entered in the system before any student marks information for that semester can be entered.

Error handling/response to abnormal situations

If any of the above validations/sequencing flow does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.2 Student Information Maintenance

Description

The system will maintain information about choice of different Elective subjects opted by various students of different enrollment years in different semesters. The following information would be maintained:

Student Enrollment number, Semester, Student's Choice for Elective 1 subject Student's Choice for Elective 2 subject.

The system will allow creation/modification/deletion of new/existing students and also have the ability to list all the students enrolled in a particular year.

Validity checks

- (i) Only user with role Data Entry Operator will be authorised to access the Student Information Maintenance module.
- (ii) Every student will have a unique Enrollment Number.
- (iii) Enrollment Number cannot be blank.
- (iv) Student name cannot be blank.
- (v) Enrollment Year cannot be blank.

Sequencing information

Student Info for a particular student will have to be entered in the system before any marks info can be entered for that student.

Error handling/response to abnormal situations

If any of the above validations/sequencing flow does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.3 Students' Subject Choices Information Maintenance

Description

The system will maintain information about choice of different Elective subjects opted by various students of different enrollment years in different semesters. The following information would be maintained:

Student Enrollment number, Semester, Student's Choice for Elective 1 subject, Student's Choice for Elective 2 subject.

The system will allow creation/modification/deletion of students' subject choices and also have the ability to list all the available students' subject choices for a particular semester.

Validity checks

- (i) Only user with role Data Entry Operator will be authorised to access the Students' Subject Choices Information Maintenance module.
- (ii) The subject choice for Elective 1 and Elective 2 can be made only from the list of available choices for that semester.

Sequencing information

Students' Subject Choices Info for a particular student can be entered in the system only after Subject Info has been entered in the system for the given semester and the Student Info for that student has been entered in the system.

Students' Subject Choices Info for a particular student will have to be entered in the system before any marks info can be entered for that student in the given semester.

Error handling/response to abnormal situations

If any of the above validations/sequencing flow does not hold true appropriate error messages will be prompted to the user for doing the needful.

3.2.4 Marks Information Maintenance

Description

The system will maintain information about marks obtained by various students of different enrollment years in different semesters. The following information would be maintained:

Student Enrollment Number, Semester, Subject Code, Internal Marks, External Marks, Total Marks, and Credits.

The system will allow creation/modification/deletion of marks information and also have the ability to list all the available marks information for all students for a particular subject in the given semester.

Validity checks

- (i) Only user with role Marks Entry Clerk will be authorised to access the Marks Information Maintenance module.
- (ii) Internal Marks for any subject cannot be less than 0 and greater than 40.
- (iii) External marks for any subject cannot be less than 0 and greater than 60.
- (iv) Total marks in any subject will be calculated as: Internal Marks in that subject + External Marks in that subject.
- (v) If the total Marks in a subject are $>= 50$, all the credit points associated with that subject will be given to the student, else the credit points earned by the student will be 0 for that subject.

Sequencing information

Marks Info for a particular student can be entered in the system only after Subject Info has been entered in the system for the given semester, the Student Info for that student has been entered in the system, and the Students' Subject Choice Info has been entered in the system for that student in the given semester.

Marks info for a particular student will have to be entered in the system before that student's mark-sheet can be generated.

Error handling/response to abnormal situations

If any of the above validations/sequencing flow does not hold true appropriate error messages will be prompted to the user for doing the needful.

3.2.5 Mark-sheet Generation

Description

The system will generate mark-sheet for every student in different semesters.

Mark-sheet will have the following format:

Name of the University Name of Program Semester <no.> Mark-sheet						
Student Enrollment No. _____ Student Name: _____						
S.No.	Subject	Internal Marks (Out of 40)	External Marks (Out of 60)	Total Marks (Int. + Ext.)	Pass/Fail	Credits Earned
1.						
2.						
3.						
4.						
5.						
6.						

Marks Grand Total: _____/600 Total Credits: _____
Result: (Pass/Fail)

Date: _____ Signature of Controller of Examinations

There will be a 'Print' icon at the top of mark-sheet for printing the mark-sheet.

Validity checks

- (i) Only user with role Marks Entry Clerk will be authorised to access the Mark-sheet Generation module.

Sequencing information

Marks-sheet for a particular student can be generated by the system only after Subject info has been entered in the system for the given semester, the Student Info for that student has been entered in the system, the Students' Subject Choice Info has been entered in the system for that student in the given semester, and the Marks Info has been entered for that student for the given semester.

Error handling/response to abnormal situations

If any of the above validations/sequencing flow does not hold true appropriate error messages will be prompted to the user for doing the needful.

3.2.6 Report Generation

Student list reports

For each year a report will be generated containing the list of students enrolled in that batch year.

Report format:

*Name of University
Name of the Program
List of students enrolled in year xxxx*

S.No.	<i>Student Enrollment Number</i>	<i>Student Name</i>
1.		
2.		

Students' subject choices list reports

For each batch year a report will be generated containing the list of students and their choices for Elective subjects in the selected semester. For Ist, IInd and IIIrd semesters the list will contain the names of elective subjects opted by each student. For IVth semester the list will contain the topic/subject area of dissertation for each student.

Report format (for Ist, IInd and IIIrd Semesters):

*Name of University
Name of the Program*

S.No.	<i>Student Enrollment Number</i>	<i>Student Name</i>	<i>Elective 1</i>		<i>Elective 2</i>	
			<i>Code</i>	<i>Name</i>	<i>Code</i>	<i>Name</i>
1.						
2.						

Report format (for IVth Semester):

*Name of University
Name of the Program*

S.No.	<i>Student Enrollment Number</i>	<i>Student Name</i>	<i>Topic / Subject Area of Dissertation</i>
1.			
2.			

Semester-wise mark lists

For each semester a mark list will be generated that will have the total marks (internal + external) of all students (enrolled in the selected Batch Year) of that semester in all subjects.

Report format:

<i>Name of University</i> <i>Name of the Program</i>							
S.No.	Enrollment Number	Subject 1 Total Marks	Subject 2 Total Marks	Subject 3 Total Marks	Subject 4 Total Marks	Subject 5 Total Marks	Subject 6 Total Marks
1.							
2.							

Rank-wise list report

This report will be generated for each semester of every Batch Year. It will show the Grand Total marks, Percentage, Rank and Division secured of all students of that semester. The report will be sorted in increasing order of Percentage/Rank.

Report format:

<i>Name of University</i> <i>Name of the Program</i>						
S.No.	Enroll. No.	Name	Grand Total Marks	%age	Rank	Division
1.						
2.						

3.2.7 User Accounts Information Maintenance

Description

The system will maintain information about various users who will be able to access the system. The following information would be maintained:

User Name, User ID, Password, and Role.

Validity checks

- (i) Only user with role Administrator will be authorised to access the User Accounts Information Maintenance module.

- (ii) User Name cannot be blank.
- (iii) User ID cannot be blank.
- (iv) User ID should be unique for every user.
- (v) Password cannot be blank.
- (vi) Role cannot be blank.

Sequencing information

User Account for a particular user has to be created in order for the system to be accessible to that user. At system startup, only a default user account for 'Administrator' would be present in the system.

Error handling/response to abnormal situations

If any of the above validations/sequencing flow does not hold true, appropriate error message will be prompted to the user for doing the needful.

3.3 Performance Requirements

None

3.4 Design Constraints

None

3.5 Software System Attributes

3.5.1 Security

The application will be password protected. Users will have to enter correct username, password and role in order to access the application.

3.5.2 Maintainability

The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules (*i.e.*, subject info, student info, students' subject choices info, marks info, report generation and user accounts info).

3.5.3 Portability

The application will be easily portable on any windows-based system that has MS-Access 2003 installed.

3.6 Logical Database Requirements

The following information will be placed in a database:

- (i) **Subject info:** Subject Name, Code, Credit points, Type and Semester
- (ii) **Student info:** Student enrollment number, Student Name, enrollment year
- (iii) **Students' subject choice info:** Student Enrollment No, Semester, Choice of Elective 1 subject, Choice of Elective 2 subject

- (iv) **Marks info:** Student Enrollment No., Semester, internal Marks in each subject, External Marks in each subject
- (v) **User account info:** User Name, User ID, Password, Role.

3.7 Other Requirements

None

REFERENCES

- [BERR98] Berry D.M. & B. Lawrence, "Requirements Engineering", IEEE Software, Mar/April, 26–29, 1998.
- [BITT03] Bittner K. and I spence, "Use Case Modelling", Pearson Education, 2003.
- [BROO95] Brooks F.P., "No Silver Bullet", The Mythical Man Month: Essay on Software Engineering (Second Edition), Addison Wesley Longman, Reading, Mass, 179–209, 1995.
- [CAIN75] Caine S. & E. Gordon, "A Tool for Software Design", Proceeding of the AFIPS, National Computer Conference, Vol. 47, AFIPS Press, Montvale, NJ, 271–276, 1975.
- [CAPE94] Caper Jones, "Charting the seas of Information Technology", 1994.
- [CARE90] Carey J. M., "Prototyping: Alternative Systems Development Methodology", Information and Software Technology, Vol. 32, No. 2, March 1990, 119–126.
- [CERI88] Ceri S. et al., "Software Prototyping by Relational Techniques: Expenses with Program Construction Systems", IEEE Transaction on Software Engineering, Vol. 14, No. 11, Nov. 1988, 1597–1609.
- [CHEN76] Chen P., "The Entity–Relationship Model—Towards the Unified View of Data", ACM Trans on database systems, March, 9–36, 1976.
- [COAD89] Coad P. & E. Yourdon, "OOA—Object Oriented Analysis", Englewood Cliffs, N.J. Prentice Hall, 1989.
- [COUN99] Councill B., "Third–Party Testing and Stirrings of the New Software Engineering", IEEE Software, Nov./Dec., 76–88, 1999.
- [DAV194] Davis A. & P. Hsia, "Giving Voice to Requirements Engineering", IEEE Software, March, 12–16, 1994.
- [DAVI90] Davis A.M., "Software Requirements Analysis and Specification", PH, Englewood Cliffs, NJ, 1990.
- [DEMA79] DeMacro T., "Structured Analysis and System Specification", Englewood Cliffs, N.J., PH, 1979.
- [HEKM87] HeKmatpour S., "Experience Evolutionary Prototyping in Large Software Project", ACM Software Engineering Notes, Vol. 12, No. 1, January, 1987, 38–41.
- [HEME82] Hemenway K. & L. X. McCusker, "Prototyping & Evaluating a User Interface", Proc. of the 6th International Computer Software & Applications Conference, Chicago, Illinois, Nov. 1982, 175–180.
- [HICK03] Hickey A.M. and A.M. Davis, "Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge Intensive Software Development Processes", Proc. of 36th Hawaii Int. conf. on system science, 2003.
- [HOFF99] Hoffer J.A. et al., "Modern Systems Analysis and Design", Addison–Wesley, 1999.