INDIRA GANDHI NATIONAL OPEN UNIVERSITY

MCSP-232

PROJECT

INTRANET MAIL SYSTEM

SUBMITTED BY

Sandeep kumar
Enrollment:-2200096169

UNDER GUIDANCE

OF

Mr. ABHISHEK KUMAR

SUBMITTED TO THE SCHOOL OF COMPUTER AND INFORMATION SCIENCES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE
MASTER OF COMPUTER APPLICATION (MCA-NEW)

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

Maidan Garhi
New Delhi-110068

ABHISHEK KUMAR
TECHNICAL CONSULTANT

EXPERTISE

Dynamics 365 for Financial &Operations,
Ax 2009, X++, MorphX
Excellent in problem solving skills, communication, interpersonal skills

CERTIFICATIONS

EDUCATION

1. MCA -2021from GGSIPU,DELHI
2. B.Sc. in Computer science in 2016BRAB University

PERSONAL DETAILS:

1. **Name**:- ABHISHEK KUMAR
2. **Date of Birth**: 15- APRL - 1995
3. **Gender**: Male
4. **Nationality**: Indian
5. **Language**: English, Hindi
6. Email:- abhishekstm2018stm@gmail.com

CAREER SNAPSHOT

Total Experience – 2years Dynamics 365 experi...

Working astechnicalconsultant inaRinfy, from FEB 2021 to till date.

DYNAMICS AX HIGHLIGHTS

- 2 years of experience in Microsoft Dynamics 365 design, and development
- 2+ project full cycle implementation.
- Having experience to support Onsite and offshore client.
- Having hands on experience D365 F&O, AX 2009,MorphX, SQLserver 2008,X++, SSRS, Workflow and Model store deployment and deploying UAT and, Dynamics 365 for sales integration with D365 finance.
- Data integration from sales force.

DYNAMICS-365 PROJECT EXPERIENCE:

1-CLIENT: - AGRICULTURAL PRODUCT TRADING

PRODUCT: D365 FOR FINANCE AND OPERATION

PROJECT TYPE:- DEVELOPMENT (Inventory fair valuation functionality).

SUMMARY OF PROJECT: -Inventory fair valuation project , that project is a part of inventory management here we have created the full scenario. The main functionality of this project is calculating the fair values on the basis of rate and quantity of objects.

- First, we enter a serial number and create a line
- Click on new button and select parameters on dialog .
- Enter the Unit rate value then all value fields are auto filled.

Abhishek Kumar
23-10-2023

- Click on post button the amount will be create.
- After that we have clicking on ok button then journal number created on header form and
- The journal will also post and
- Field of Reversal will check.
- The reversal date field fulfill by the one day after the posted date
- The next line will be created.
- Click on print button to create a report.

ROLES &RESPONSIBILITIES: -

- Work as associate technical consultant
- I have worked on the functionality inventory fair valuation full development.
- New Customization in form, Classes, SSRS Report according to business process.
- Understand current solution and provide a feasible design fit of D365.

2-CLIENT: -AGRICULTURAL PRODUCT TRADING

PRODUCT: DYNAMICS 365 FINANCE AND OPERATION

PROJECT TYPE: UPGRADE AND REIMPLEMENTATION (PURCHASE REGISTER REPORT)

SUMMARY OF PROJECT:-In this projectas per the requirements, I have customized and manage the report data .validating the values of the fields according to the requirement put some validation on the report also. that was the better experience for me to arranging and applying the logics on the report .

ROLES & RESPONSIBILITIES:

- Work as associate technical consultantk
- Customization &development on report.
- Manage LCS and deployment on cloud.
- Role involved estimation, technical design, and development.

3-CLIENT: AGRICULTURAL PRODUCT TRADING

PRODUCT: DYNAMICS 365 FINANCE AND OPERATION

PROJECTTYPE:DEVELOPMENTANDSUPPORT(DATA INTIGRATION)

SUMMARY OF PROJECT: -In this project we must use the OData and DMF for data integration.

- Create the entity of the table.
- In UI data management module creating the data group.
- Data entity show on the module form and add table entity.
- Select the target data format.
- And export the data in selected format.
- Also import the same format after entering the data.

ROLES &RESPONSIBILITIES:

- Workasassociatetechnicalconsultant.
- Roleinvolvedestimation, technicaldesignanddevelopment.
- ManageLCSanddeploymentoncloud.
- Customizationonclass,table,SSRSreport.

Ashish Kumar

28-10-2023

SUMMARY OF PROJECT: - Quality Order Quantity per Actual Receipt or Report As Finished functionality FDD we have add the functionality for report as finished.

8-CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (The system shall allow authorized users the ability to see the BATCH and SERIAL numbers while vouchering invoices)

SUMMARY OF PROJECT: - The system shall allow authorized users the ability to see the BATCH and SERIAL numbers while vouchering invoices FDD we have add the two field on the form and populate the data in the field as per the requirement.

9-CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (Automate Hold Functionality for Batches and Batch Orders)

SUMMARY OF PROJECT: - Automate Hold Functionality for Batches and Batch Orders this FDD is related from Batch status in which we have create three field in warehouse management parameter batches OK ,Hold, Reject Disposition code. It was related from production and Quality order also

10-CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (Product Reconciliation Report)

SUMMARY OF PROJECT: - Product Reconciliation Report here we have working on report data populating from deferent table as per the requirement.

11 -CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (Bartender Labelling Customizations)

SUMMARY OF PROJECT:- Bartender Labelling Customizations this FDD we created five form and six table that functionality is working on request xml and get response from printer API in Logging form .

12 -CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT(Manage Returns(SF -D365) Power Automate)

SUMMARY OF PROJECT:- In salesforce integration from dynamics we use power Automate as a third party to creating flow and connect with condition hare we create business event and data entity .

Place:India

Abhishek Kumar

Abhishek Kumar

23-10-2023

4-CLIENT: LIQUOR RETAIL and consumer services

PRODUCT: DYNAMICS AX 2009

PROJECT TYPE: CUSTOMIZE AND SUPPORT (STORE ROSTER)

SUMMARY OF PROJECT:-In Store Roster project our task was that to customize the scenario of the form and also work on backend to creating the table and menu items and applying the objects on the form with business logics and it was working on proper way as per the requirement .

ROLES & RESPONSIBILITIES:

- Work as associate technical consultant.
- Customized the form and implement.
- Developed the tables, forms, menu items.
- Role involved estimation, technical design, and development.
- Manage LCS and deployment on cloud.

5-CLIENT: AGRICULTURAL PRODUCT TRADING

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (VENDOR BALANCE LIST)

SUMMARY OF PROJECT: -Vendor Balance List project provide the better understanding about the grouping the report data .as per the requirement we have to fully customized the report where the value of each field come from the deferent table then after exact calculation, we have to show the accurate value on the report . the data also grouped on the bases of location.

ROLES & RESPONSIBILITIES:

- Work as associate technical consultant.
- Customization on Tables, class and SSRS report.
- Role involved estimation, technical design, and development.
- Manage LCS and deployment on cloud.

6 -CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (Sample Requests and Movement Journal)

SUMMARY OF PROJECT: - In Sample Requests and Movement Journal creating a email functionality where we add a button when we click a button the system will fair an email and movement journal will be created. This functionality is added in inventory management.

7-CLIENT: REAL DYNAMICS

PRODUCT: DYNAMICS 365

PROJECT TYPE: DEVELOPMENT AND SUPPORT (Quality Order Quantity per Actual Receipt or Report As Finished)

Abhishek Kumar

23-10-2023

Guru Gobind Singh Indraprastha University, Delhi

COE/F52/MAY16
18544883

CONSOLIDATED GRADE SHEET

MASTER OF COMPUTER APPLICATIONS

NAME:	ABHISHEK KUMAR	TOTAL CREDITS OF PROGRAMME:	160
ENROLLMENT:	00117004418	MINIMUM CREDITS REQUIRED:	150
FATHER'S NAME:	ARUN KUMAR SINGH	YEAR OF COMPLETION:	Jun, 2021
YEAR OF ADMISSION:	2018	PROGRAMME DURATION:	THREE YEARS
INSTITUTE:	TECNIA INSTITUTE OF ADVANCED STUDIES		

CODE	PAPER	CS	INT	EXT	TOTAL	GRADE	GP
	FIRST SEMESTER						
MCA101	FUNDAMENTALS OF IT	4	20	47	67	A	8
MCA103	PROGRAMMING IN C	4	19	32	51	B	6
MCA105	DISCRETE MATHEMATICS	4	22	45	67	A	8
MCA107	COMPUTER ORGANIZATION	4	23	30	53	B	6
MCA109	PRINCIPLES AND PRACTICES OF MANAGEMENT	4	21	44	65	A	8
MCA151	FUNDAMENTALS OF IT LAB.	1	33	54	87	A+	9
MCA153	PROGRAMMING IN C LAB	2	35	46	81	A+	9
MCA155	COMPUTER ORGANIZATION LAB.	2	36	48	84	A+	9
MCA161	GENERAL PROFICIENCY - I	1	-	71	71	A	8
	SECOND SEMESTER						
MCA102	DATA AND FILE STRUCTURES	4	25	57	82	A+	9
MCA104	OBJECT ORIENTED PROGRAMMING IN C++	4	25	31	56	B+	7
MCA106	OPERATING SYSTEMS	4	20	44	64	B+	7
MCA108	DATABASE MANAGEMENT SYSTEMS	4	24	47	71	A	8
MCA110	SOFTWARE ENGINEERING	4	21	44	65	A	8
MCA152	DATA AND FILE STRUCTURES LAB	1	39	58	97	O	10
MCA154	OBJECT ORIENTED PROGRAMMING IN C++ LAB	2	39	55	94	O	10
MCA156	DATABASE MANAGEMENT SYSTEMS LAB	1	36	49	85	A+	9
MCA158	SOFTWARE ENGINEERING LAB.	1	36	53	89	A+	9
MCA162	GENERAL PROFICIENCY – II	1	-	84	84	A+	9
	THIRD SEMESTER						
MCA201	THEORY OF COMPUTATION	4	25	30	55	B+	7
MCA203	COMPUTER GRAPHICS	4	25	50	75	A+	9
MCA205	JAVA PROGRAMMING	4	25	48	73	A	8
MCA207	DATA COMMUNICATIONS & NETWORKING	4	25	28	53	B	6
MCA209	C# PROGRAMMING	4	25	43	68	A	8
MCA251	COMPUTER GRAPHICS LAB	1	40	55	95	O	10
MCA253	JAVA PROGRAMMING LAB	2	40	56	96	O	10
MCA255	C# PROGRAMMING LAB	2	40	55	95	O	10
MCA261	GENERAL PROFICIENCY -III	1	-	96	96	O	10
	FOURTH SEMESTER						
MCA202	DESIGN AND ANALYSIS OF ALGORITHMS	4	25	65	90	O	10
MCA204	DATA WAREHOUSING AND DATA MINING	4	25	65	90	O	10
MCA206	ADVANCED COMPUTER NETWORKS	4	25	65	90	O	10
MCA208	OBJECT ORIENTED ANALYSIS AND DESIGN	4	25	65	90	O	10
MCA210	WEB TECHNOLOGIES	4	25	65	90	O	10
MCA252	DESIGN AND ANALYSIS OF ALGORITHMS LAB	1	40	50	90	O	10
MCA254	DATA WAREHOUSING AND DATA MINING LAB	1	40	50	90	O	10
MCA256	ADVANCED COMPUTER NETWORKS LAB	1	40	50	90	O	10
MCA258	OBJECT ORIENTED ANALYSIS AND DESIGN LAB	1	40	50	90	O	10
MCA260	WEB TECHNOLOGIES LAB	1	40	50	90	O	10
MCA262	GENERAL PROFICIENCY – IV	1	-	100	100	O	10
	FIFTH SEMESTER						
MCA301	LINUX PROGRAMMING	4	25	26	51	B	6
MCA303	SOFTWARE TESTING	4	25	45	70	A	8
MCA305	ENTERPRISE COMPUTING WITH JAVA	4	25	35	60	B+	7
MCA313	MULTIMEDIA TECHNOLOGIES	4	25	64	89	A+	9
MCA333	SOFTWARE QUALITY MANAGEMENT	4	25	50	75	A+	9
MCA351	LINUX PROGRAMMING LAB	1	40	49	89	A+	9
MCA353	SOFTWARE TESTING LAB	1	39	59	98	O	10
MCA355	ENTERPRISE COMPUTING WITH JAVA LAB	2	39	55	94	O	10
MCA357	LAB BASED ON ELECTIVE - I	1	40	51	91	O	10
MCA361	GENERAL PROFICIENCY - V	1	-	98	98	O	10
	SIXTH SEMESTER						
MCA302	DISSERTATION	26	40	51	91	O	10
MCA362	GENERAL PROFICIENCY – VI (SEMINAR AND PROGRESS REPORT)	4	-	97	97	O	10

CREDITS EARNED: 160	CGPA: 8.73	EQUIVALENT PERCENTAGE: 87.3	DIVISION: FIRST

CS: Credit Secure; INT: Internal Marks; EXT.: External Marks; ABS: Absent; CAN: Cancel; GP: Grade Point; *: Passed with Grace
Minimum Cumulative Grade Point Average (CGPA) required for the award of the Degree is 4.

CRMID: 160000024299
Date of Print: 31-Aug-2021

Place: Delhi, India
Abhishek Kumar
23-10-2023

Officer In-Charge

Controller of Examinations

XI. CERTIFICATE OF ORIGINALITY

This is to certify that the project report entitled INTRANET MAIL SYSTEM submitted to **Indira Gandhi National Open University** in partial fulfilment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA)** , is an authentic and original work carried out by Mr. / Ms. SANDEEP KUMAR with enrolment no. 22000961 69 under my guidance.

The matter embodied in this project is genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirements of any course of study.

Sandeep Kumar

Signature of the Student:

Date: 21/10/23

Name and Address
of the student

vill- VISHANPUR
P.O.- PATDAURA
P.S.- BAJPATTI
DIS - SITAMARHI

Enrolment No. 22 000 96 169

Ashok Kumar

Signature of the Guide

Date: 23-10-2023.

Name, Designation
and Address of the
Guide:

Technical
Consultant
Shivnagar Riga
Sitamarhi Bihar
843327

INDEX OF PROJECT REPORT

TITLE OF THE PROJECT

“E-FACILITATOR”

PROJECT SYNOPSIS

INDEX OF SYNOPSIS

1. Introduction and Objective:

AUTOMATION OF E-FACILITATOR MONITORING is an intranet based Java application that automates the working of a professional institute that imparts professional training in the field of IT, Management, Accounting etc. this application is created as a product and can be customized according the specific needs of the client.
All such professional institutions have same kind of problems to solve such as they run professional courses for which queries are received, students are enrolled, classes are conducted, batches are scheduled, dues are to be collected, record of batches, students and study materials is to maintained, various type of reports are generated, etc.

AUTOMATION OF E-FACILITATOR MONITORING provides solution of all these problems in the form of easy to configure and use application by automating all these functionality. Its major modules are:

- Enquiry Management
- Registration
- Query Management
- Batch Scheduling
- Fee and Recovery Management
- Attendance Management
- Student Management (Transfer, Break, Resume & Backup)
- Study Material Management
- Result & Certification

2. Project Category:

The product is a web application that is to be implemented on local Intranet. The advantage of developing the product as web application is that it need not be installed on individual machines of clients. A web application is installed on the server and client access it using a browser.
The product comes under the category of MIS as it captures information related to different operations of an educational institution and generates various reports which help the management in decision making.

Input of the project:-

- Login page get the input of user id and password.
- Create the user id for register yourself.
- Student details.
- View responses.
- Change password

Output of the project:-

- View student details.
- Help
- Home

Benefits of proposed System:-

- Fully works as a online
- Reducing the time
- Don't Thinking
- Centralized maintain all information.
- Easy to Searching the information.

3. TOOLS AND PLATFORM USED

This software is a database application. The tools used for developing the projects are as follows:

1. **MYSQL**
2. **J2EE (servlet, jsp and beans)**
3. **Weblogic 10.3**

Reasons for using these tools:

J2EE-

J2EE is a Product of Sun Microsystems That is used for developing Web based application. This software include **Servlet, jsp, Bean** .These component a large number of graphical components that are generally used in developing GUI applications.

The J2EE programming system provides with many kinds of tools to create attractive and useful applications. this makes us more productive by providing appropriate tools for the different aspects of GUI development. The objects like text box, list box, picture box, labels can easily be drawn thus helps of Jsp. And after designing the interface we make this interface interact with the user by writing code help of Bean class that responds to events that occur in the interface.

MYSQL-

MYSQL is Object Relational database Management System (ORDBMS). It offers capabilities of both relational and object oriented database system. In general objects can be defined as reusable software codes which are location independent and perform a specific task on any application environment with little or no change to the codes. SQL is used to access the data within the MYSQL. It contains a set of commands, which make it very easy to maintain the database. It has for sub parts DDL, DML, DCL, and TCL. DDL includes the commands, which allows us to create objects and to manipulate the structure of the objects. DML includes the commands to manipulate the information stored in a database. DCL includes the commands for controlling the data access and TCL includes the commands for controlling the transactions like commit and rollback.The database server or back-end is used to manage the database files optimally among multiple clients who concurrently request the server for the same data. It also enforces Data Integrity across all client application and controls database access and other security requirements.

HARDWARE AND SOFTWARE REQUIREMENT:-

Hardware Specification:

It is recommended that the minimum configuration for clients is as appended below Suggested Configuration of Windows clients:-

Microprocessor	**: - Pentium-4 class processor, 450 (MHz)**
Ram	**: - 512 MB of RAM**
CD ROM Drive	**: - 52 X CD ROM Drive**
Hard Disk	**: - 40 Gigabytes (GB) on installation drive,**

Software requirements:

- **Windows 98 / XP operating system**
- **MYSQL**
- **JDK 1.6**
- **Weblogic 10.3**

Problem Definition

Analysis Reports

Requirements Definition - A software requirement is an abstract description of the services that the system shall provide and the constraints under which the system must operate.

Requirements determination involves studying the current business system to find out how it works and where improvements should be made. System studies result in an evaluation of how current methods are working and whether adjustments are necessary or possible.

A requirement is a feature that must be included in a new system. It may include a way of capturing or processing data, producing information, controlling a business activity, or supporting management. The determination of requirements thus entails studying the existing system and collecting details about it to find out what these requirements. Requirements can either be functional or non functional**.**

Requirement Analysis - Requirement Analysis is a Software Engineering task that bridges the gap between system level requirements engineering and software design. In the proposed project Software Requirements Analysis have been divided into five areas of effort.

1. Problem recognition
2. Evaluation and Synthesis
3. Modeling
4. Specification
5. Review

Requirements Elicitation for the Software:- Before requirements can be analyzed, modeled or specified they are gathered through an elicitation process.

Context free questions were asked to the management people belonging to different large organizations/ institutes regarding how they would characterize a good output that would generate a successful solution, what kind of problems will this solution address, how they describe the environment in which the solution will be used and will special performance issues or constraints effect the way the solution is approached.

SOFTWARE REQUIREMENT SPECIFICATIONS

Software requirement specification (SRS) is the starting point of the software development activity. The SRS is the means of translating the ideas in the minds of the client (the input), into a formal document (the output of the requirement phase). Thus, the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input have none of these properties. Clearly the process of specifying requirements cannot be formal.

Any formal translation process producing a formal output must have a precise and unambiguous input. A procedure for identifying requirements can therefore be at best a set of guidelines.

The requirement specification phase consists of two basic activities: - Problem or requirement analysis, and requirement specification. The first aspects, perhaps the harder and more nebulous of the two, deals with understanding the problem, the goals, and constraints. In the second, the focus is on clearly specifying what has been found during analysis. Issues such as representation, specification languages and tools and checking the specification are addressed during this activity. The requirements phase terminates with the production of the validated software requirement specification document. Producing the SRS is the basic goal of this phase.

Role of SRS:

The origin of most software systems is in the need of a client who either wants to automate an existing manual system or desires a new software system. The Developer creates the software system itself. Finally, the end users will use the complete system. Thus there are three major parties interested in the new software system *the client, the user and the developer.* The problem is that the client usually does not understand the software or the software development process and the developer often does not understand the client's problem and application area.

This cause the communication gap between the parties involved in the development project.

The purpose of software requirement specification, SRS, is to bridge this communication gap. SRS is the medium through which the client and the user needs are accurately specified: indeed, SRS forms the basis of software development. A good SRS should specify ass the parties.

A good SRS provides many benefits:

Some of the goals it accomplishes are:-

- Establishes the basis for agreement between client and software developer on what the software product will do.
- Reducing the development cost.
- Providing a reference for validation of the final product. The SRS assists the client in determining if the software meets the requirements.
- Overview of a software requirement.

Purpose: -

The system has been developed on client server architecture. The Server has a database SQL Server 7.0 and the client has Visual Basic 6.0. Providing connectivity through ADODC (ADO data control) object makes the communication between these two. They communicate via a middle ware component called ODBC.

PROBLIM ANALYSIS: -

The first of the two basic activities performed during the requirement phase is analyzing the problem. Problem analysis is done to obtain a clear understanding of the needs of the client and the users and what exactly is the desired from the software. Analysis leads to the actual specification. Analysis involves interviewing the clients and the users. Typically, analysts research a problem by asking questions from clients and users, and by reading existing documents.

REQUIREMENT SPECIFICATIONS: -

Once the analysis is complete, the requirements must be written or specified. The final, output is the Software Requirement Specifications documents (SRS). For smaller problems or the problems that can easily be comprehended , the specification activity might come after the entire analysis is complete. However it is more likely that problem analysis and specification are done concurrently. And analyst typically will analyze some parts of the problem and then writes the requirements for that part.

CHARACTERISTICS OF AN SRS: -

To properly satisfy the basic goals, an SRS should have certain properties and should contain different types of requirements. In this section we should discuss some of the desirable characteristics of an SRS, and different components of and SRS. A good SRS is:-

- Understandable.
- Unambiguous.
- Complete.
- Verifiable.
- Consistent.
- Modifiable.
- Traceable.

Clearly an SRS should be understandable, as one of the goals of the requirement phase is to produce a document upon which a client, the user and the developer can agree. Since multiple parties need to understand and approve the SRS, it is most utmost importance that the SRS should be understandable.

And SRS is complete if everything the software is supposed to do is in the SRS. A complete SRS defines the responses of the software to all classes of input data. For specifying al thee requirements, the requirements relating to functionality, performance, design constraints,

attributes and external interfaces must be specified. In addition, the responses to both valid and invalid in input values must also be specified.

A requirement is verifiable if there exists some cost effective process that can check if the final software meets that requirement. An SRS is verifiable if and only if every stated requirement is verifiable.

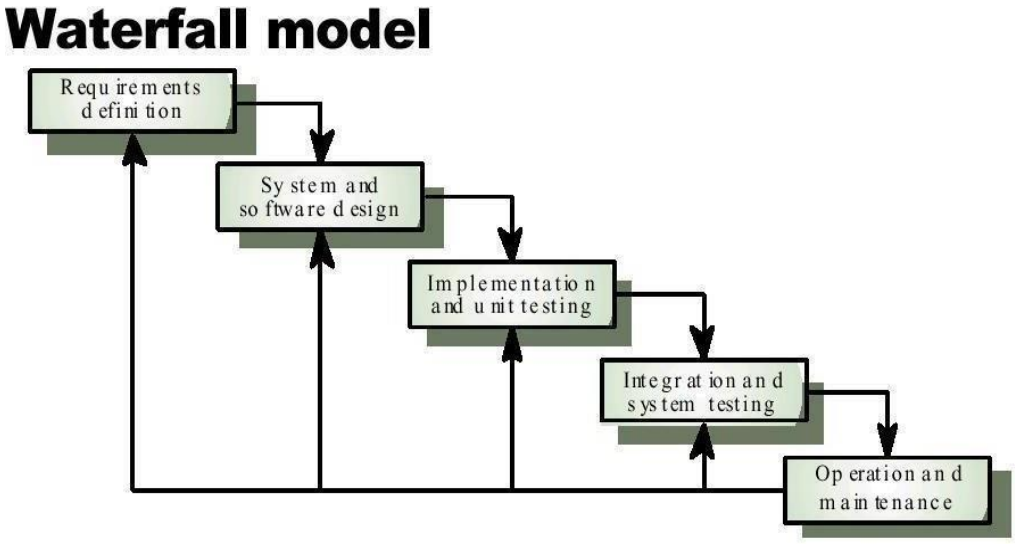
Writing an SRS is an iterative process. Even when the requirements of a system are specified, they are later modified as the need of thee client change with time. Hence an SRS should be easy to modify. An SRS is modifiable if its structure and style is such that any necessary change can be made easily, while preserving the completeness and consistency.
An SRS is traceable if the origin of each of its requirement is clear and if it facilitates the reference of each requirement in future development. Forward trace ability means that each requirement should be traceable to some design and code elements. Backward trace ability requirement should be possible to trace design and code element to requirement they support. Trace ability adds verification and validation.

4. Software Engineering paradigm applied*:*

A. METHODOLOGY.

For develop the software the basic thing is that what is the model will be used. The model will be used is of great importance for development of the software.

In the waterfall model, the sequence of the activities performed in a software development project: requirement analysis, project planning, system design, detailed design, coding, and unit testing, system integration and testing. This is the order the different phases will be apply for develop of the project.



Linear ordering of the phases is the important for the development of the software because by using this model you can follow the phases in the linear order. First the requirement analysis is done then you can proceed the next phase like
Designing. This model is used when you know all the phases in advance i.e. you know the requirement of the client. Because requirement analysis is important for the development of the software. For the development of the software there are two parties are involved in the development of the software – a client and a developer. Most often the requirement of the client does not know and the requirement of the client will be changed. So before using this model requirement of the client is know in advanced.

To overcome these problems we can used the prototype model. The basic idea for using this model is that the requirements have not known initially. In this model a prototype is build to understand the requirements. This prototype is developed based on the currently known requirements. The prototype means to develop the formal model of the software which meets the current requirements. By using this prototype, the client can get an actual feeling of the system because the interaction with the prototype can enable the client to better understand the requirements of the desired system. By using

this model there is no work manually. By using this model we can create a prototype, which meets the primary requirements of the Advertisement ads. By seeing this model the user know the requirements of the Advertisement ads. In the waterfall model the requirements had been known in advanced. To solve this problem we can used the prototype model.

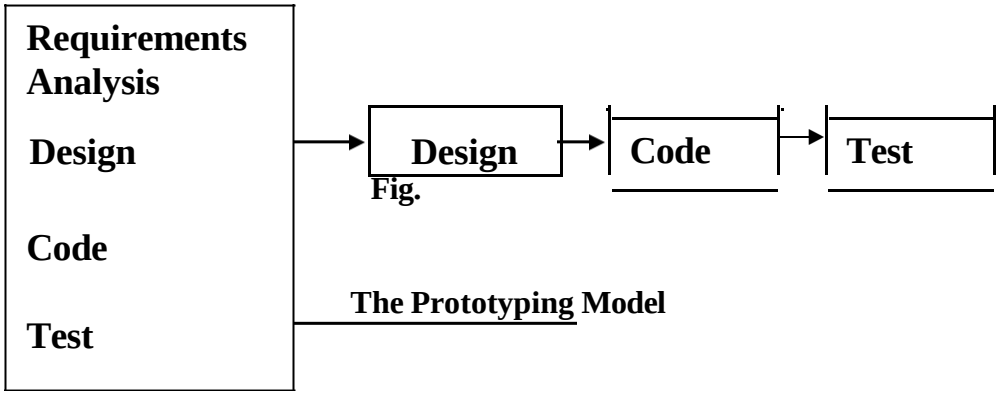


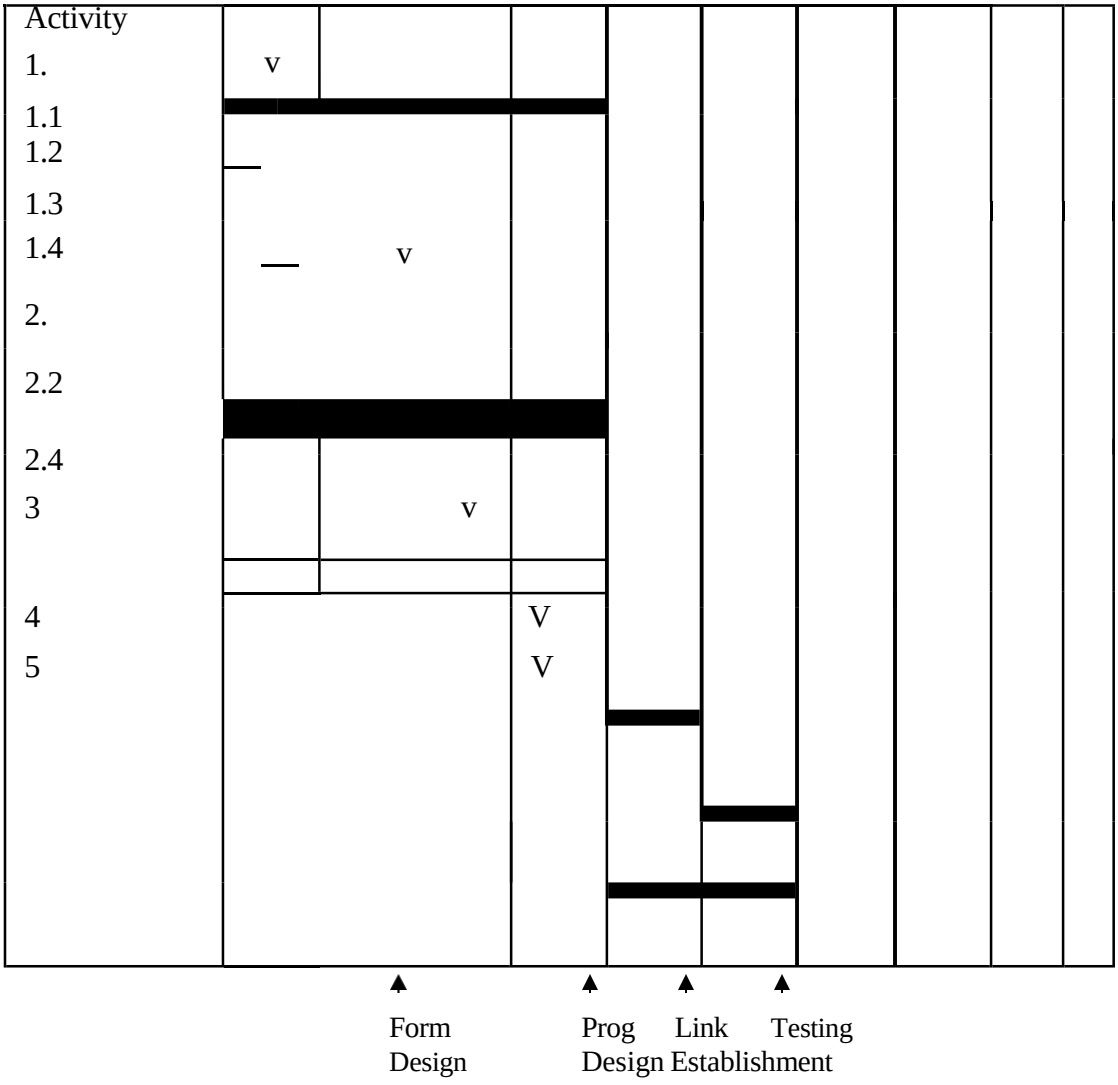
Fig.

The Prototyping Model

The development of the prototype typically starts when the preliminary version of the requirements specification document has been developed. After developed the prototype, the end users and clients are given an opportunity to use the prototype. Based on the experience, they provide the feedback to the developers regarding the prototype: what is correct, what is needs to be modified, what is missing, what is not needed, etc. In this software when we give the prototype to the client & they give the feedback like the needed of the login in the software .

5. Project Planning

GANTT CHART



Use Case Diagram

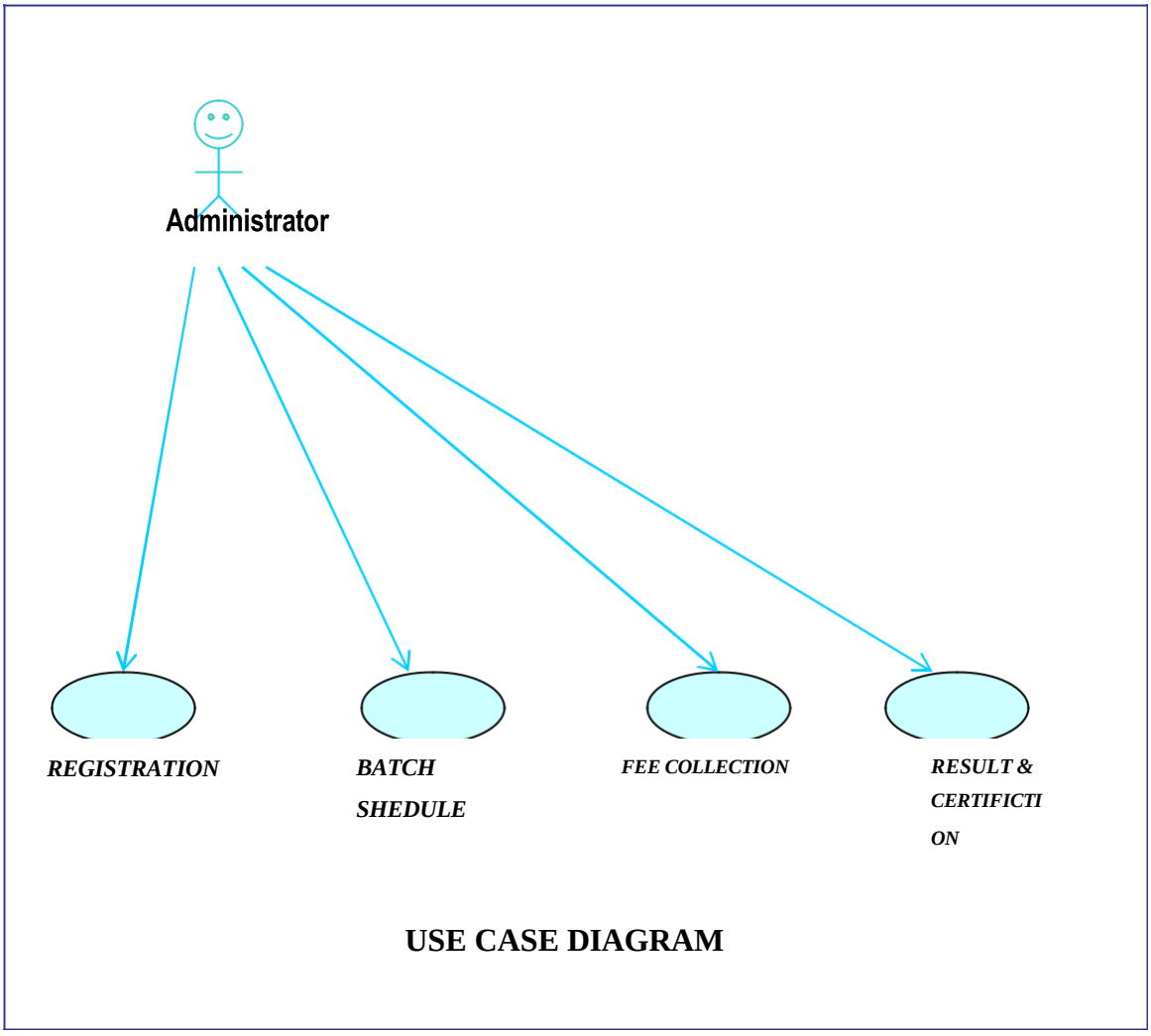
A **use case diagram** is a type of **behavioral diagram** defined by the Unified Modeling Language (UML) created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals—represented as use cases—and any dependencies between those use cases.

UML standard defines a graphical notation for modeling use cases with diagrams, but no format for describing these use cases. While the graphical notation and descriptions are important, they are documentation of the use case—a purpose that the actor can use the system for—

The true value of a use case lies in two areas:

- ☐ The written description of system behavior regarding a business task or requirement. This description focuses on the value provided by the system to external entities such as human users or other systems.

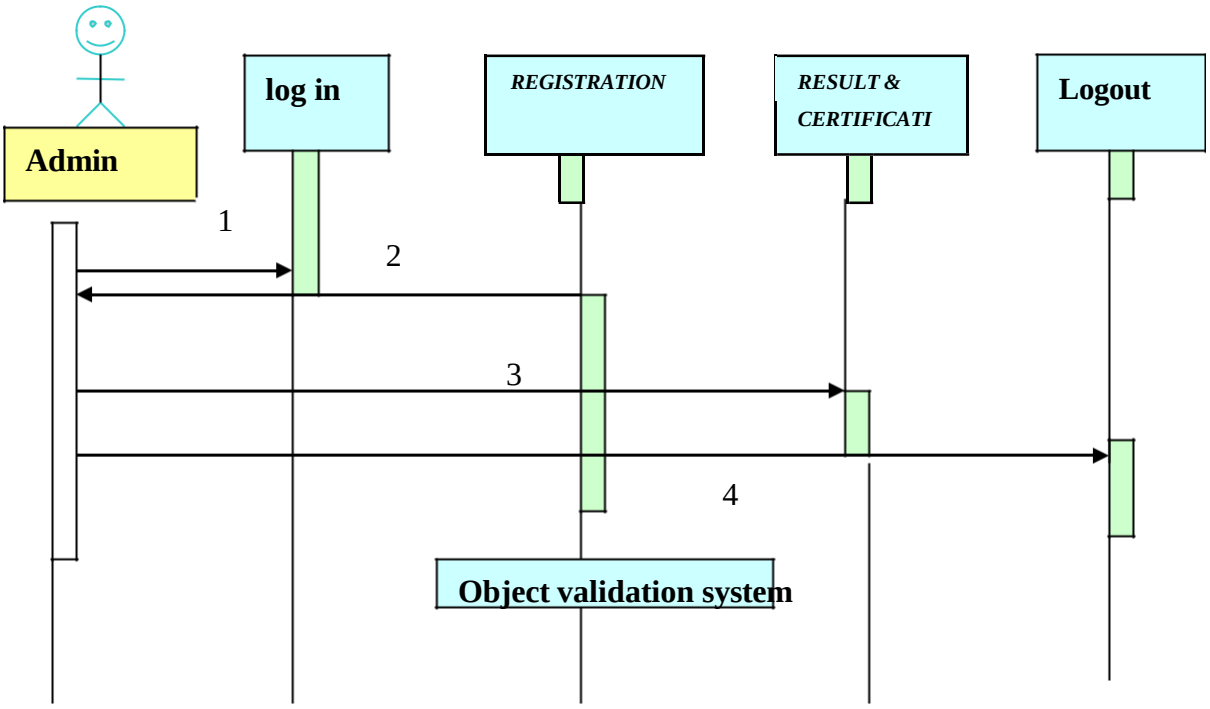
- The position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the customer/owner/user and the development team.



USE CASE DIAGRAM

Sequence Diagram

The Message Sequence Chart technique has been incorporated into the Unified Modeling Language (UML) diagram under the name of **Sequence Diagram**. A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



Sequence Diagram

DATA FLOW DIAGRAM

Graphical description of a system's data and how the processes transform the data is known as Data Flow Diagram (or DFD).

Unlike detail flowcharts, DFDs do not supply detailed descriptions of modules but graphically describe a system's data and how the data interact with the system.

To construct data flow diagrams, we use:

i. arrows,
ii. circles,
iii. open-ended boxes, and
iv. squares

An arrow identifies data flow - data in motion. It is a pipeline through which information flows. Like the rectangle in flowcharts, circles stand for a process that converts data/into information.

An open-ended box represents a data store - data at rest, or a temporary repository of data.

A square defines a source (originator) or destination of system data.

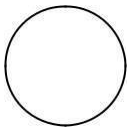
The following rules govern construction of data flow diagrams(DFD):

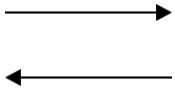
1. Arrows should not cross each other.
2. Squares, circles, and files must bear names.
3. Decomposed data flows must be balanced (all data flows on the decomposed diagram must reflect flows in the original diagram).
4. No two data flows, squares, or circles can have the same name.
5. Draw all data flows around the outside of the diagram.
6. Choose meaningful names for data flows, processes, and data stores. Use strong verbs followed by nouns.
7. Control information such as record counts, passwords, and validation requirements are not pertinent to a data-flow diagram.

If too many events seem to be occurring at a given point, an analyst can decompose a data conversion (circle). The new data conversions form a parent-child relationship with the original data conversion: the child circle

Symbols Used in DFD :

Square  **source/Destination of Data**

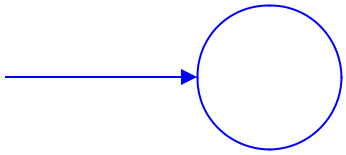
Bubble  **Process For Transformation of Data**

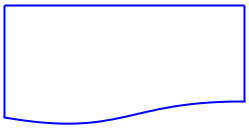
Arrows  **Data in motion (Data flow)**

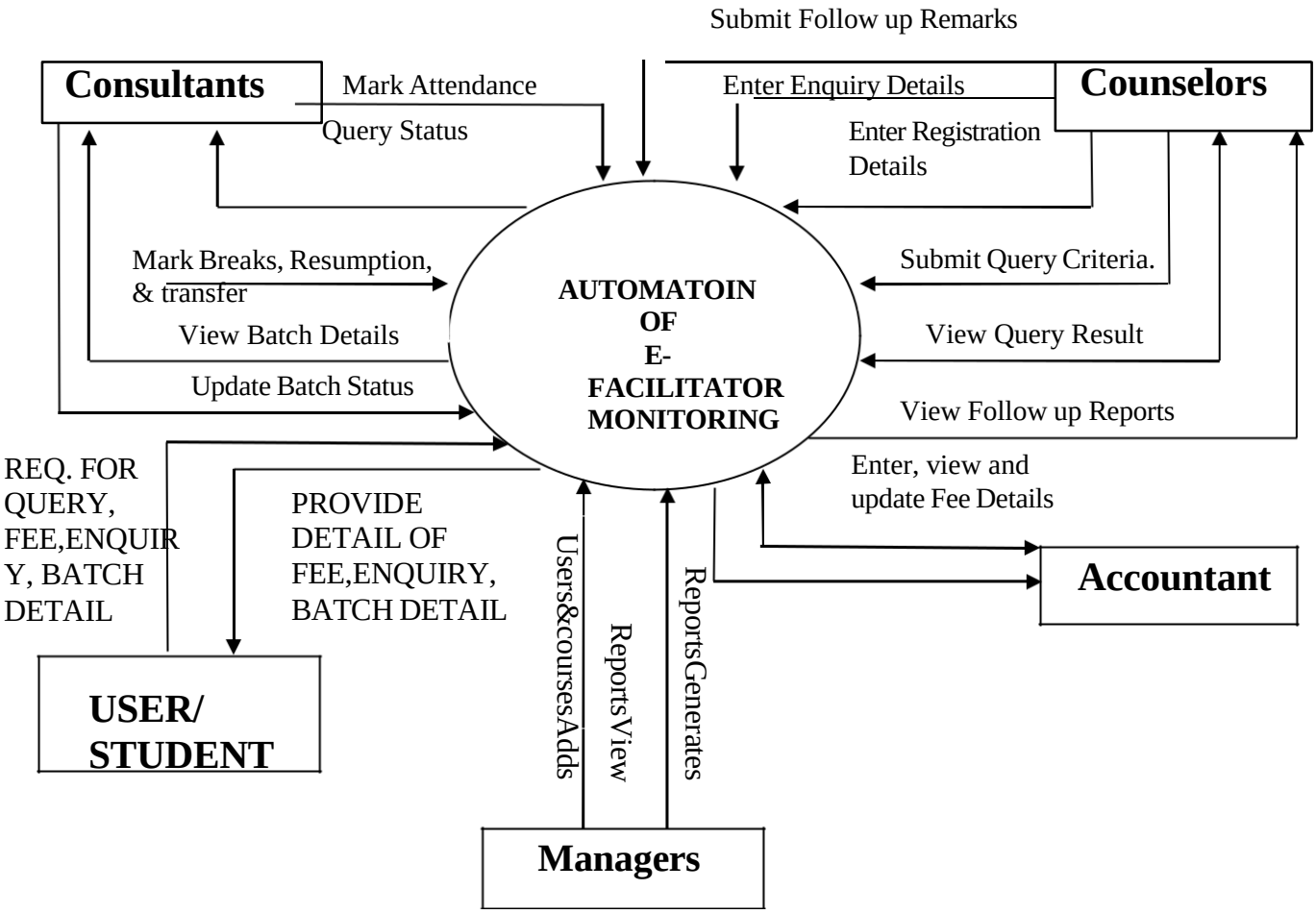
Rectangle  **Data at rest (Data Source)**

Open Rectangle

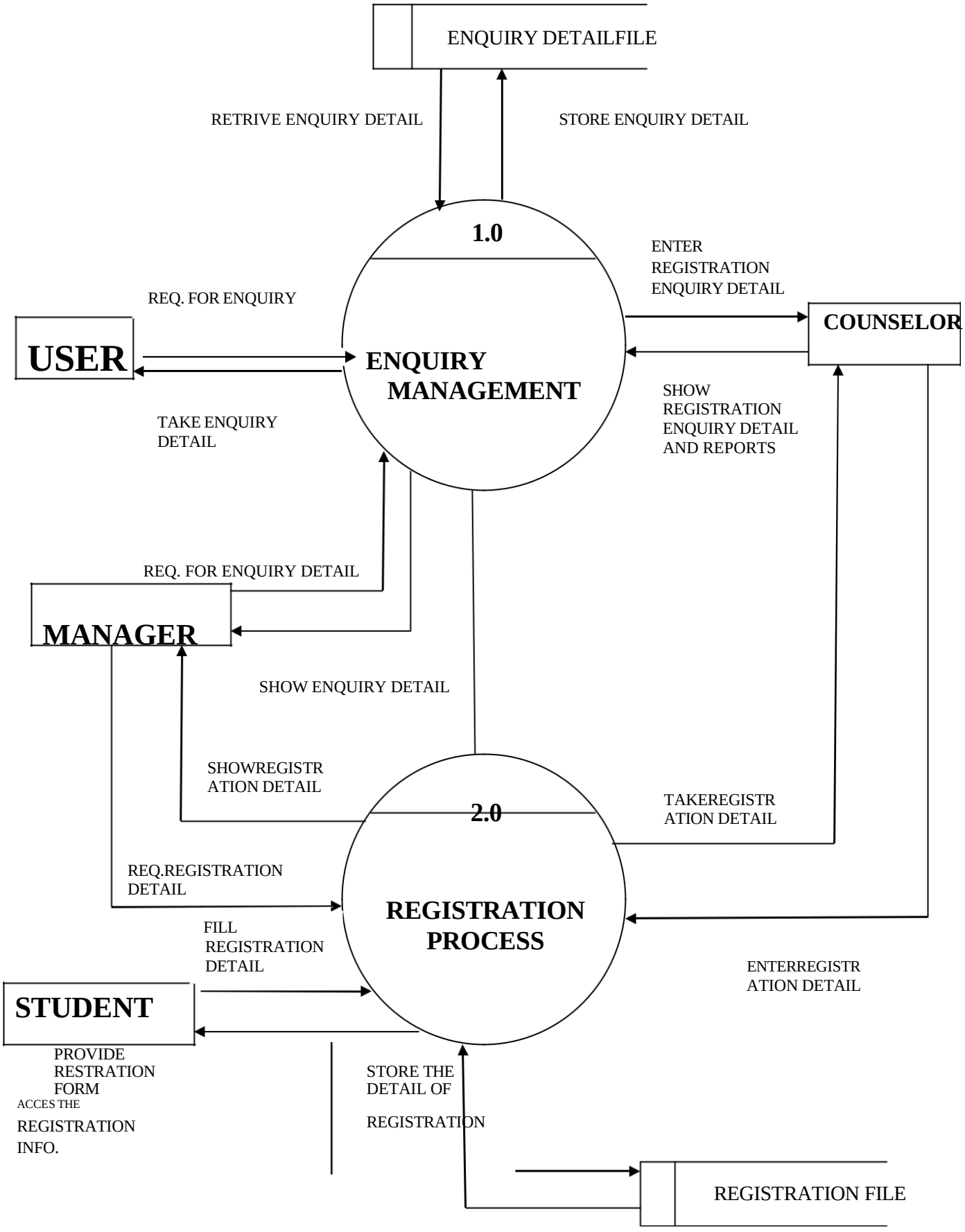
Data at rest (Data Source) 

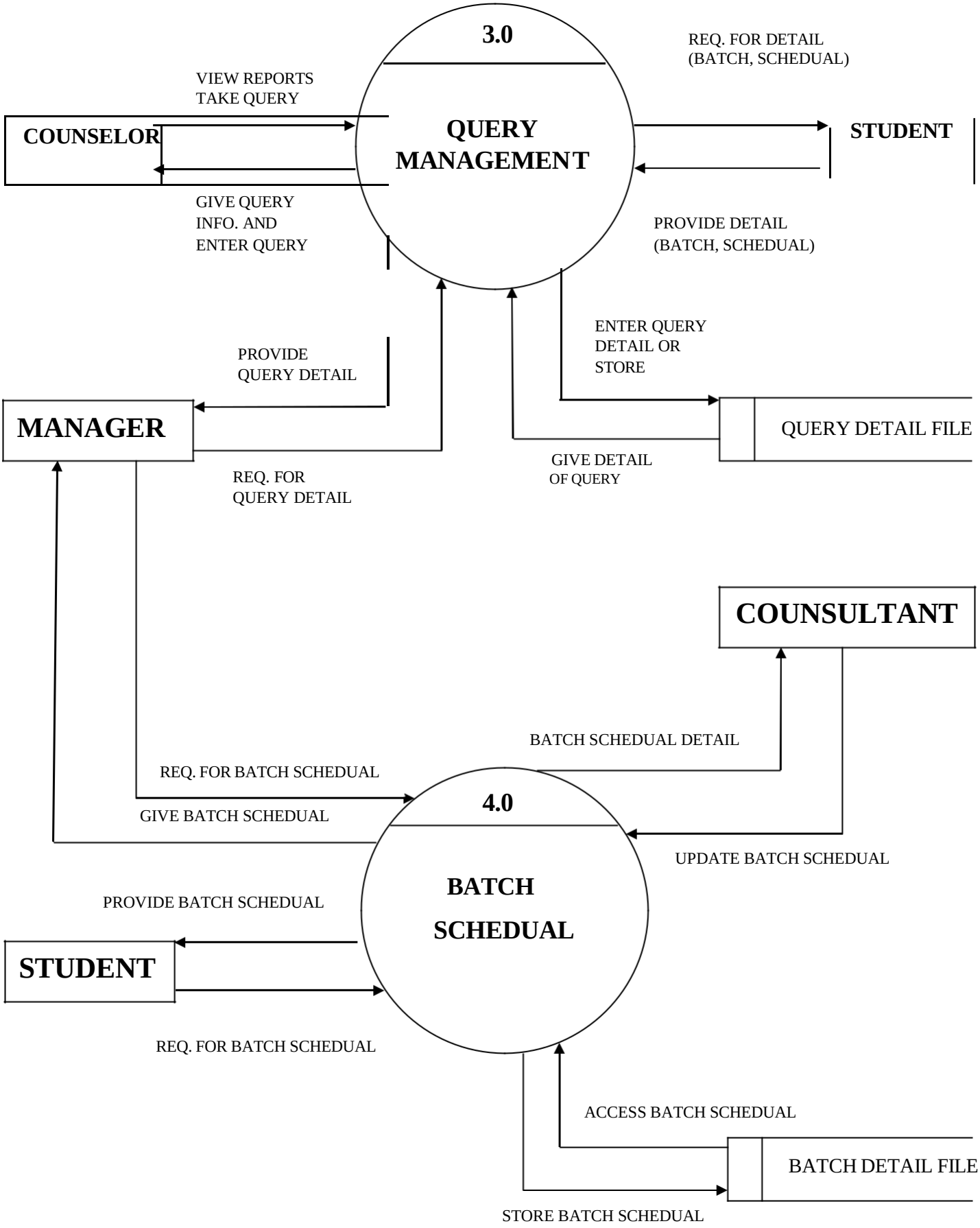
Connector 

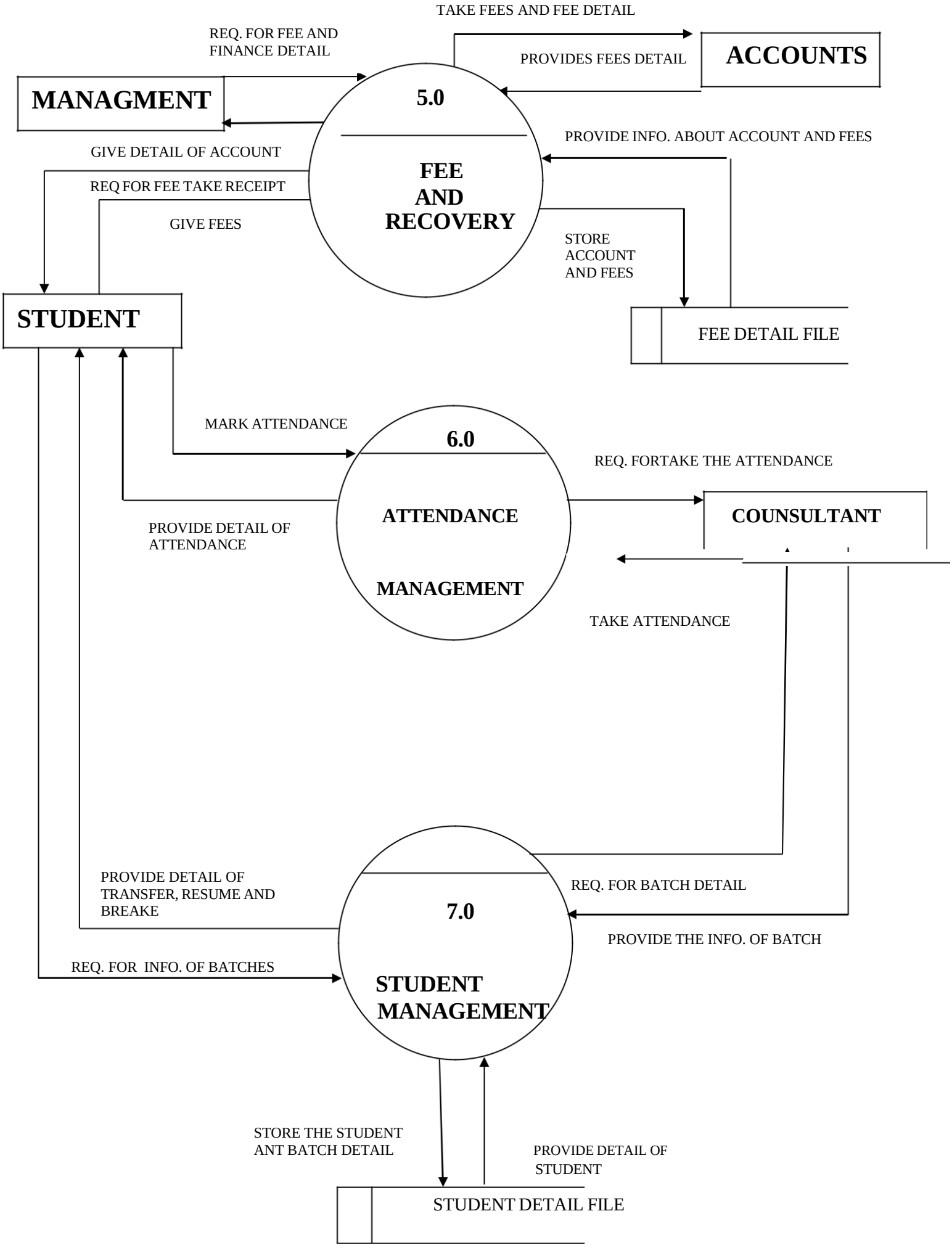
Document 

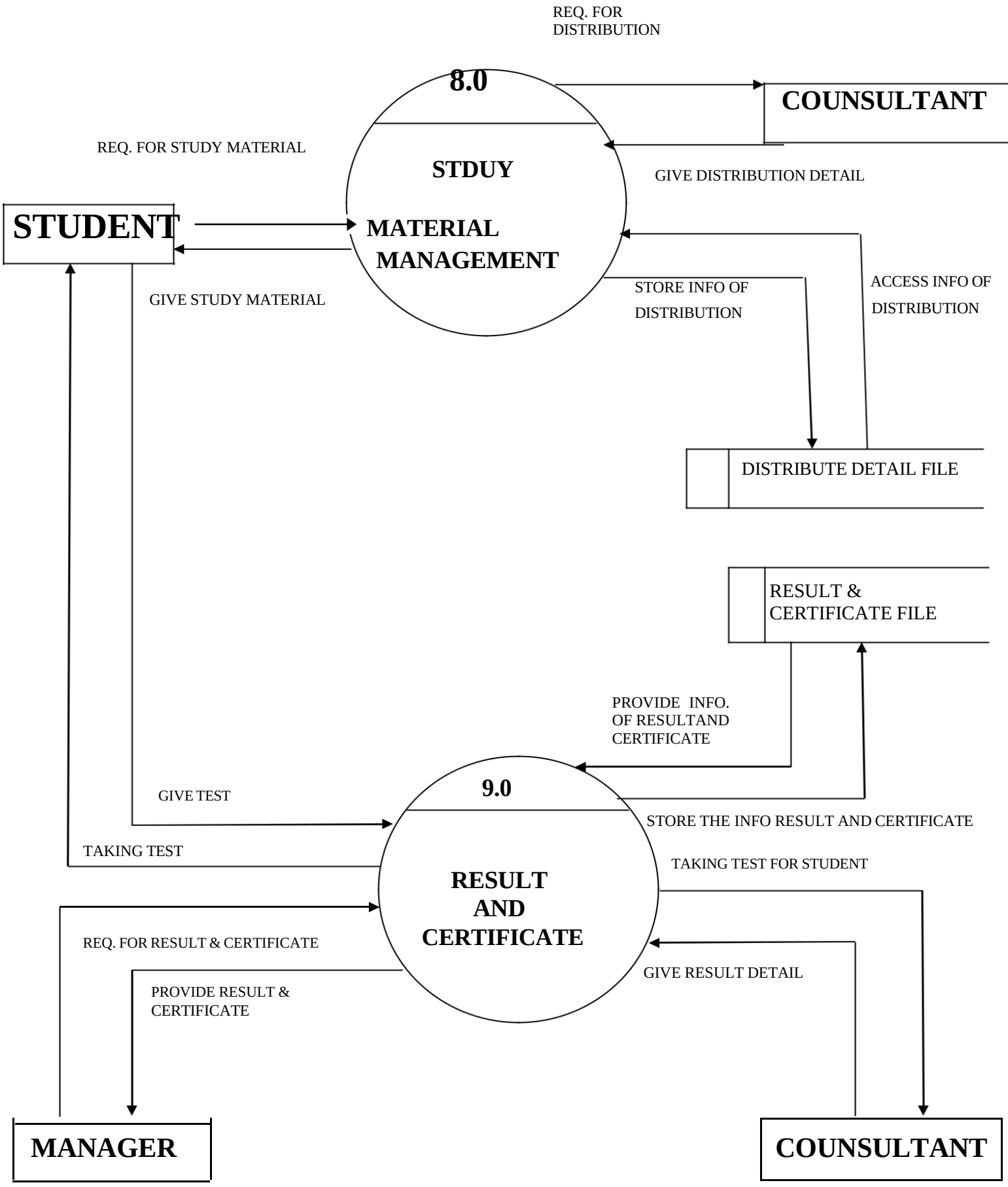

Submit Follow up Remarks
Consultants
Mark Attendance
Query Status
Enter Enquiry Details
Counselors
Enter Registration Details
Mark Breaks, Resumption, & transfer
View Batch Details
Update Batch Status
AUTOMATOIN OF E-FACILITATOR MONITORING
Submit Query Criteria.
View Query Result
View Follow up Reports
REQ. FOR QUERY, FEE,ENQUIRY, BATCH DETAIL
PROVIDE DETAIL OF FEE,ENQUIRY, BATCH DETAIL
Enter, view and update Fee Details
Accountant
USER/ STUDENT
Users&coursesAdds
ReportsView
ReportsGenerates
Managers

FIRST LEVEL DFD

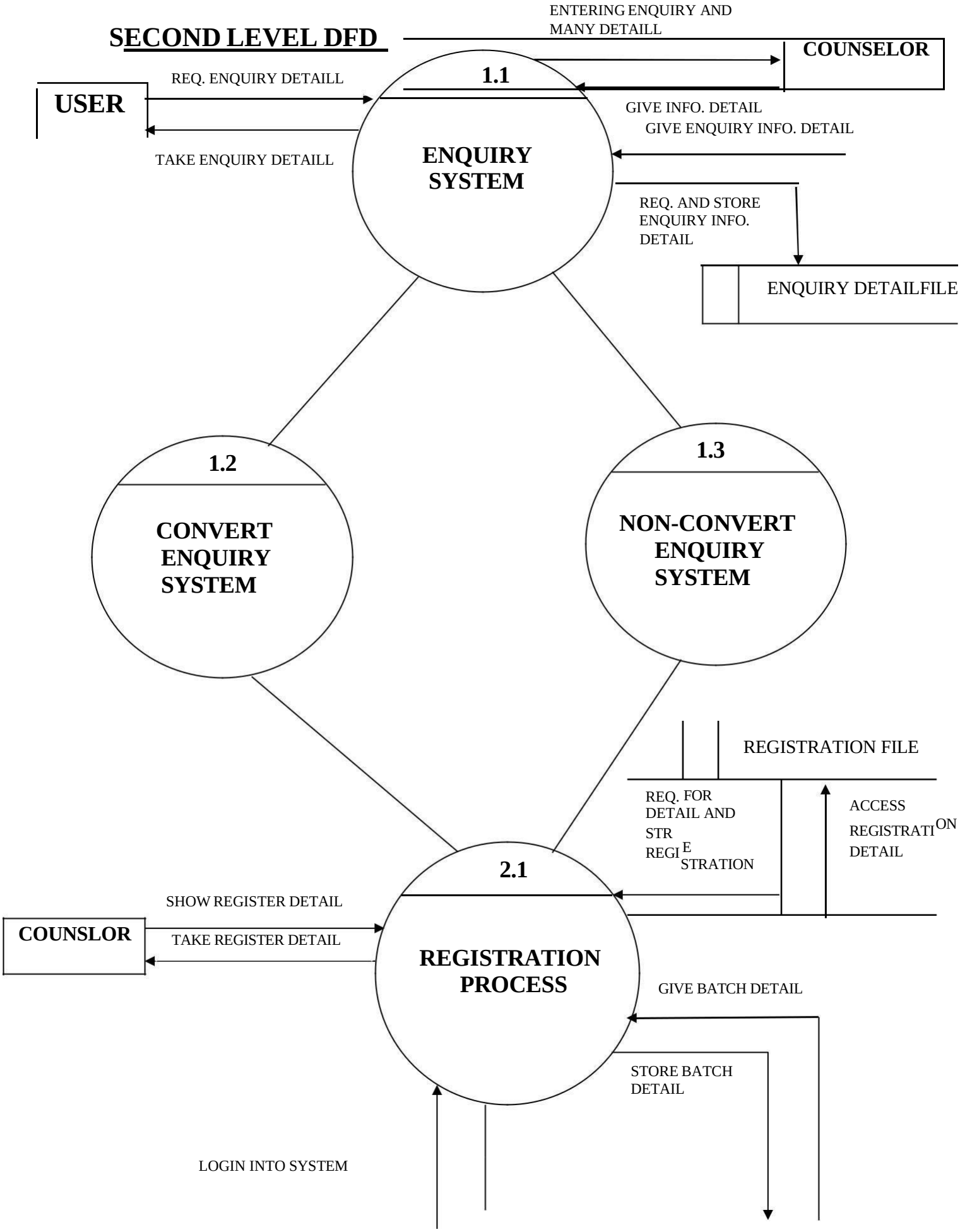


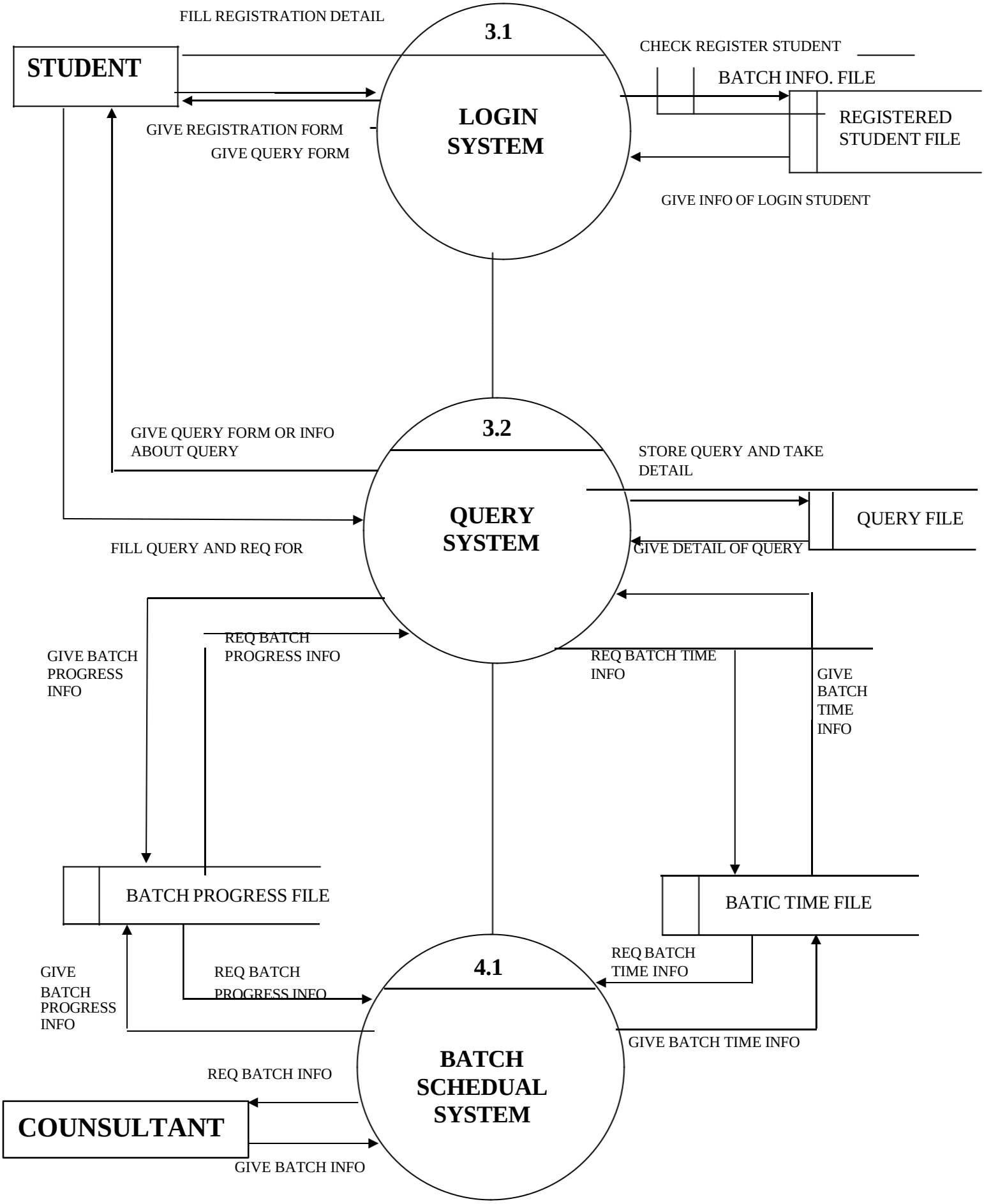

3.0
QUERY MANAGEMENT
REQ. FOR DETAIL (BATCH, SCHEDUAL)
VIEW REPORTS
TAKE QUERY
COUNSELOR
STUDENT
GIVE QUERY INFO. AND ENTER QUERY
PROVIDE DETAIL (BATCH, SCHEDUAL)
ENTER QUERY DETAIL OR STORE
PROVIDE QUERY DETAIL
MANAGER
QUERY DETAIL FILE
GIVE DETAIL OF QUERY
REQ. FOR QUERY DETAIL
COUNSULTANT
BATCH SCHEDUAL DETAIL
REQ. FOR BATCH SCHEDUAL
4.0
GIVE BATCH SCHEDUAL
UPDATE BATCH SCHEDUAL
BATCH SCHEDUAL
PROVIDE BATCH SCHEDUAL
STUDENT
REQ. FOR BATCH SCHEDUAL
ACCESS BATCH SCHEDUAL
BATCH DETAIL FILE
STORE BATCH SCHEDUAL

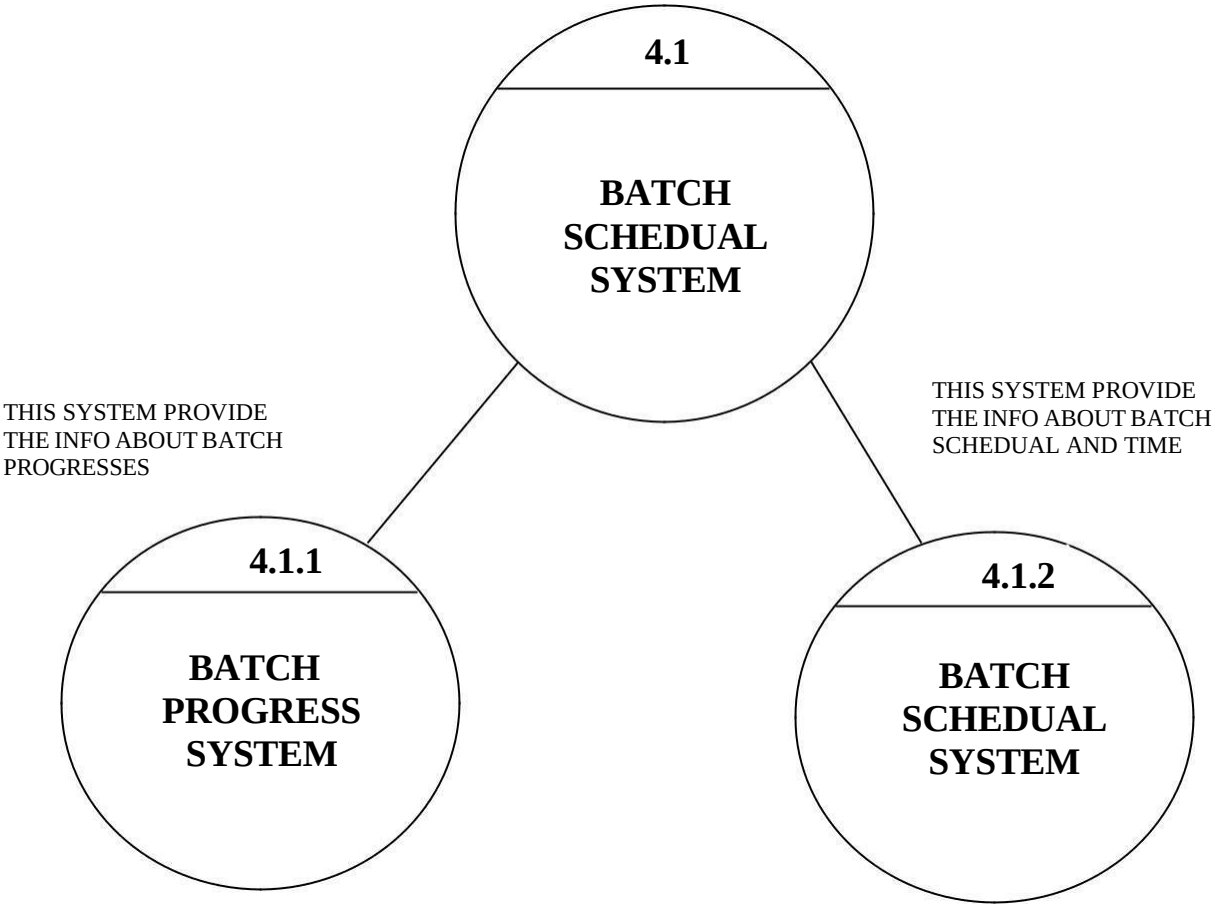

TAKE FEES AND FEE DETAIL
REQ. FOR FEE AND FINANCE DETAIL
PROVIDES FEES DETAIL
ACCOUNTS
MANAGMENT
5.0
FEE AND RECOVERY
PROVIDE INFO. ABOUT ACCOUNT AND FEES
GIVE DETAIL OF ACCOUNT
REQ FOR FEE TAKE RECEIPT
GIVE FEES
STORE ACCOUNT AND FEES
STUDENT
FEE DETAIL FILE
MARK ATTENDANCE
6.0
ATTENDANCE MANAGEMENT
REQ. FORTAKE THE ATTENDANCE
COUNSULTANT
PROVIDE DETAIL OF ATTENDANCE
TAKE ATTENDANCE
PROVIDE DETAIL OF TRANSFER, RESUME AND BREAKE
REQ. FOR BATCH DETAIL
7.0
STUDENT MANAGEMENT
PROVIDE THE INFO. OF BATCH
REQ. FOR INFO. OF BATCHES
STORE THE STUDENT ANT BATCH DETAIL
PROVIDE DETAIL OF STUDENT
STUDENT DETAIL FILE

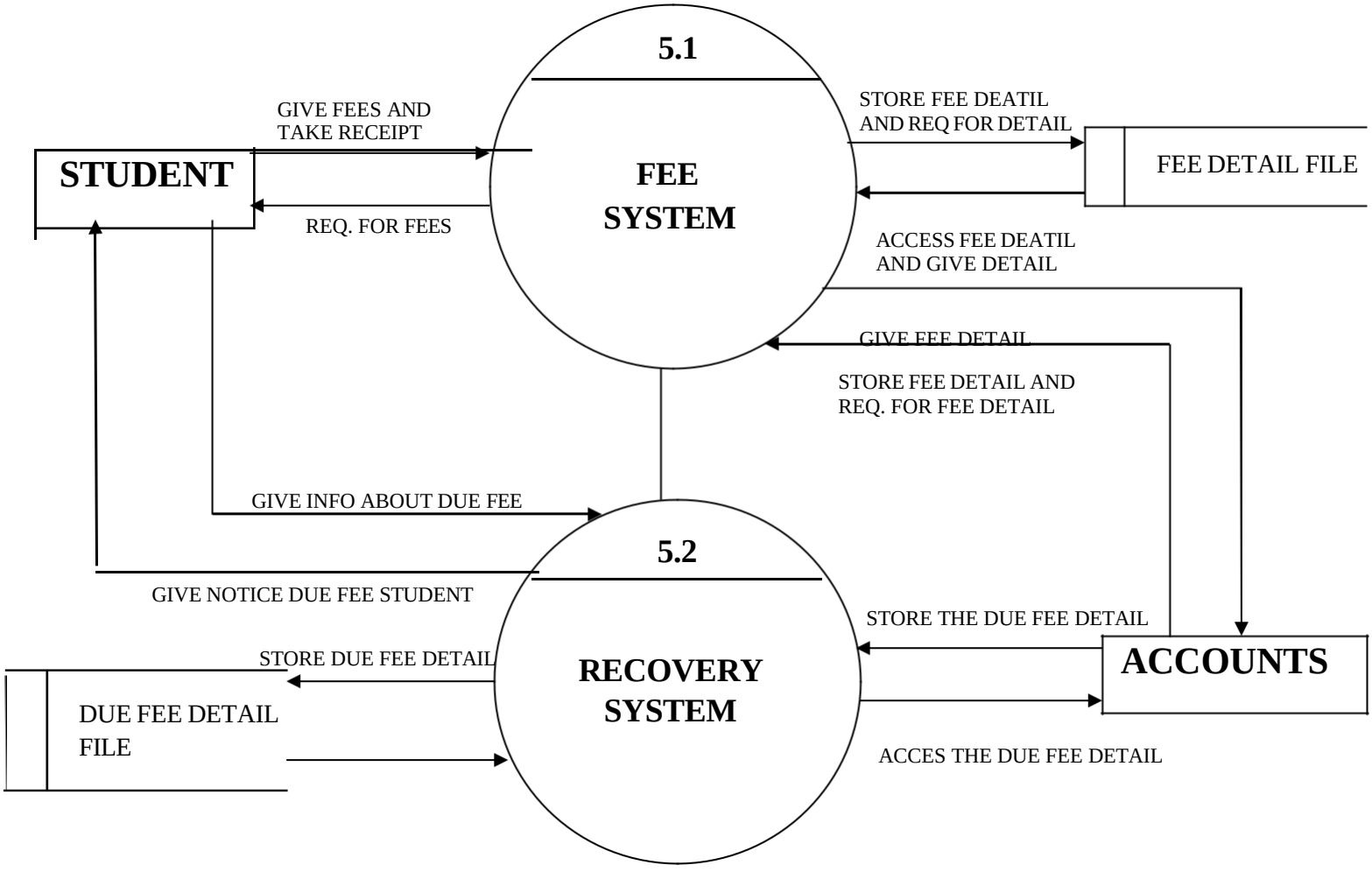

REQ. FOR DISTRIBUTION
8.0
COUNSULTANT
REQ. FOR STUDY MATERIAL
STDUY
MATERIAL MANAGEMENT
GIVE DISTRIBUTION DETAIL
STUDENT
GIVE STUDY MATERIAL
STORE INFO OF DISTRIBUTION
ACCESS INFO OF DISTRIBUTION
DISTRIBUTE DETAIL FILE
RESULT & CERTIFICATE FILE
PROVIDE INFO. OF RESULTAND CERTIFICATE
GIVE TEST
9.0
STORE THE INFO RESULT AND CERTIFICATE
TAKING TEST
TAKING TEST FOR STUDENT
RESULT AND CERTIFICATE
REQ. FOR RESULT & CERTIFICATE
GIVE RESULT DETAIL
PROVIDE RESULT & CERTIFICATE
MANAGER
COUNSULTANT

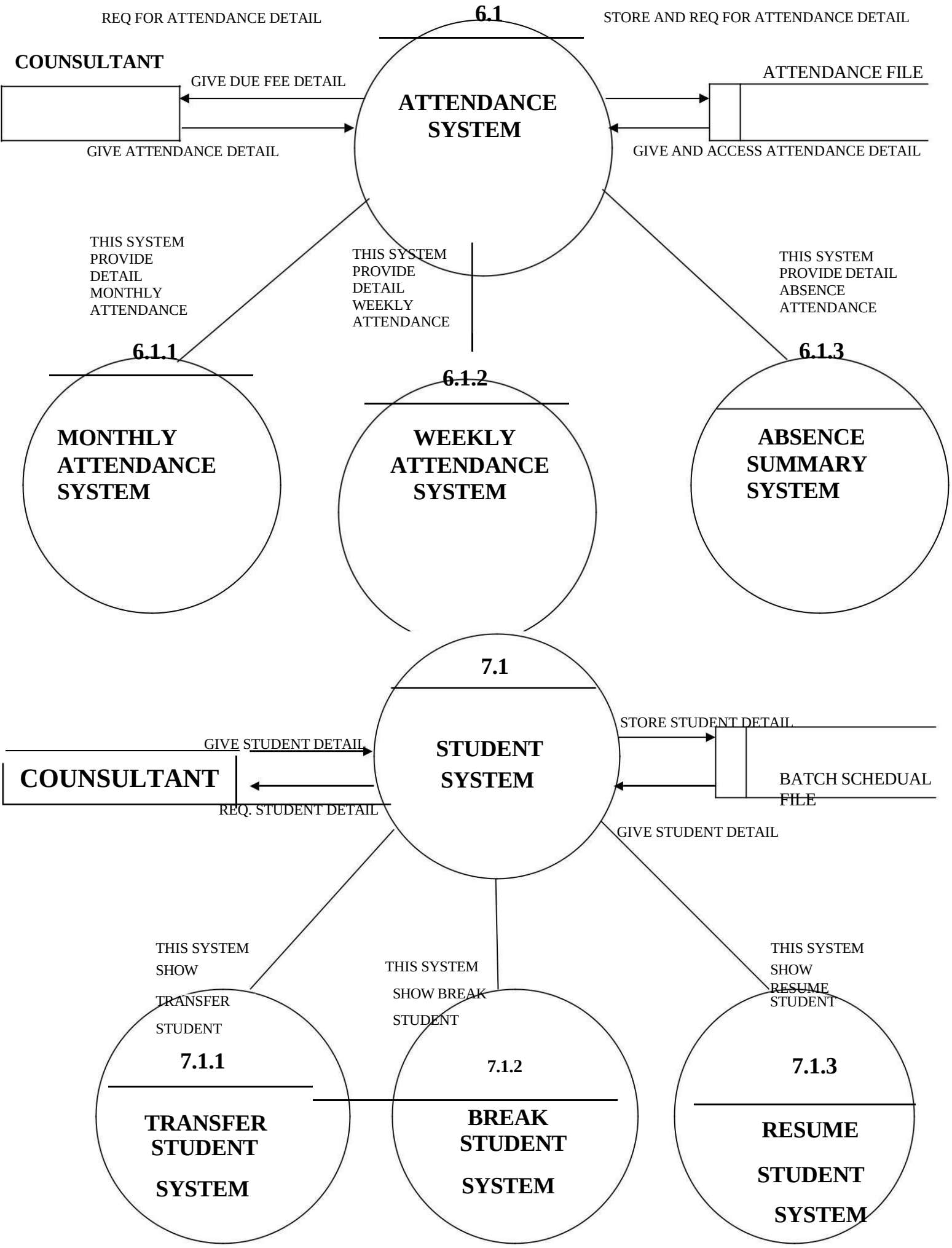
SECOND LEVEL DFD

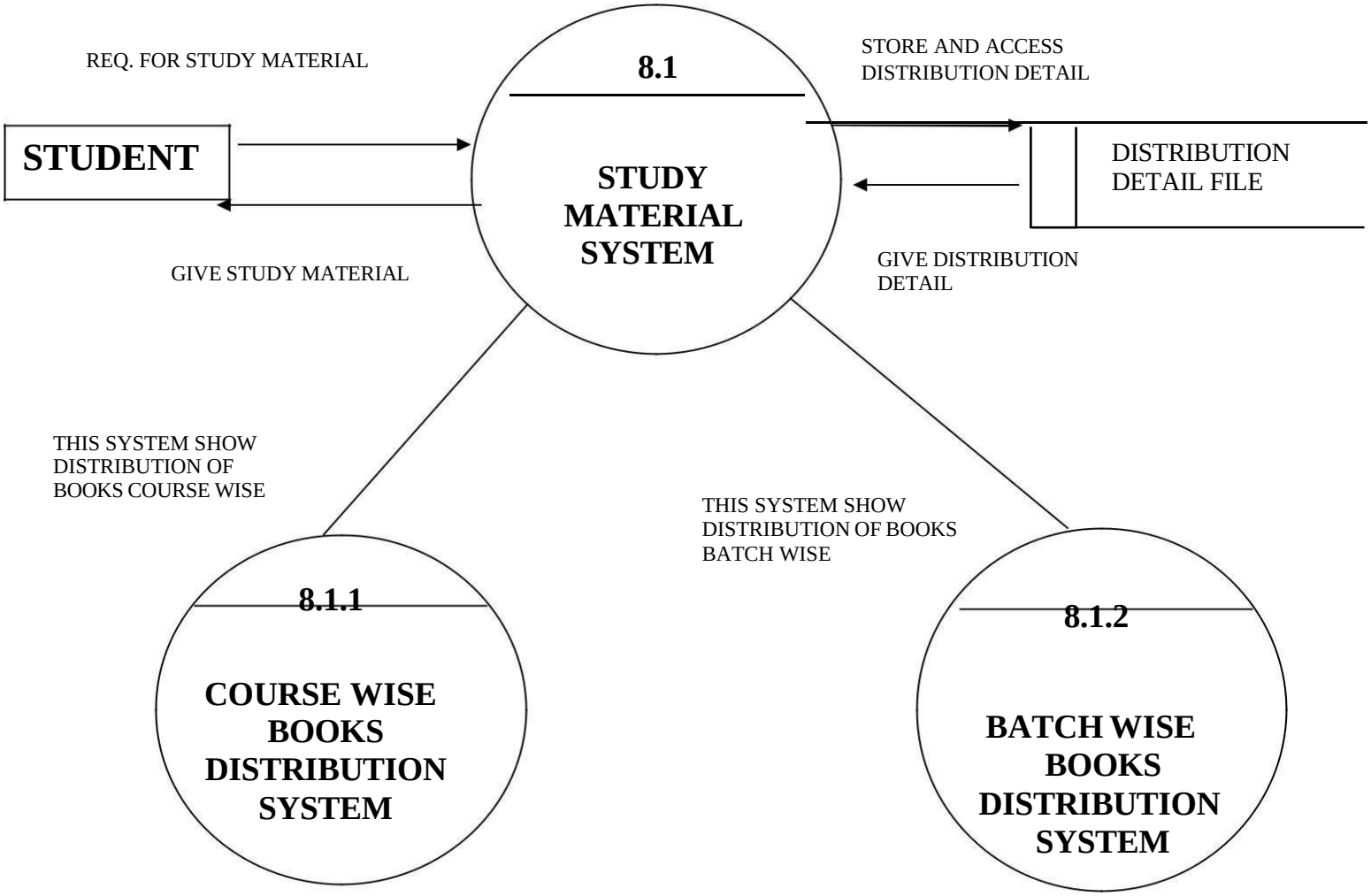


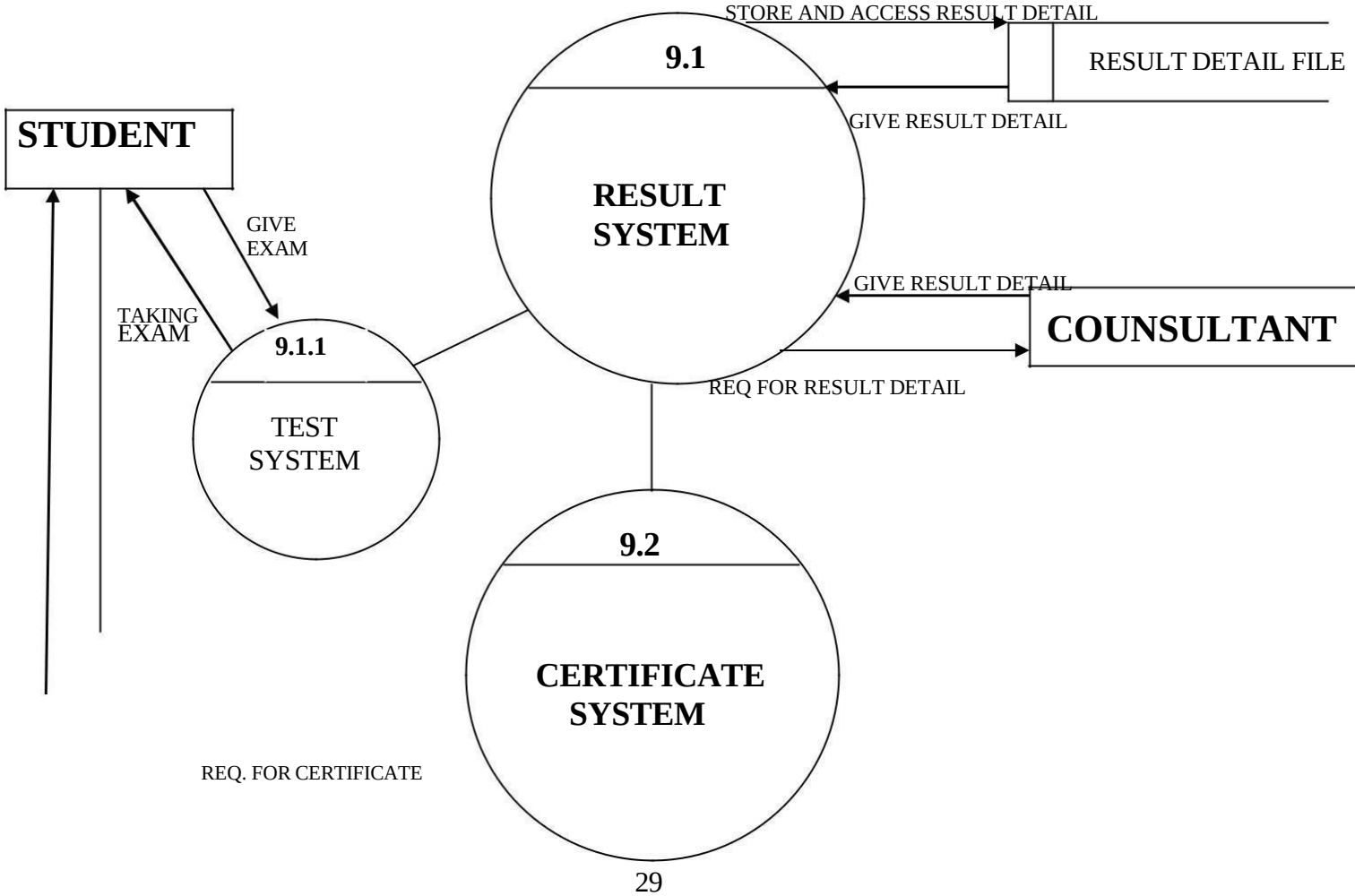

FILL REGISTRATION DETAIL
3.1
STUDENT
CHECK REGISTER STUDENT
BATCH INFO. FILE
LOGIN SYSTEM
REGISTERED STUDENT FILE
GIVE REGISTRATION FORM
GIVE QUERY FORM
GIVE INFO OF LOGIN STUDENT
GIVE QUERY FORM OR INFO ABOUT QUERY
3.2
STORE QUERY AND TAKE DETAIL
QUERY SYSTEM
QUERY FILE
FILL QUERY AND REQ FOR
GIVE DETAIL OF QUERY
GIVE BATCH PROGRESS INFO
REQ BATCH PROGRESS INFO
REQ BATCH TIME INFO
GIVE BATCH TIME INFO
BATCH PROGRESS FILE
BATIC TIME FILE
REQ BATCH TIME INFO
GIVE BATCH PROGRESS INFO
REQ BATCH PROGRESS INFO
4.1
GIVE BATCH TIME INFO
REQ BATCH INFO
BATCH SCHEDUAL SYSTEM
COUNSULTANT
GIVE BATCH INFO


4.1
BATCH SCHEDUAL SYSTEM
THIS SYSTEM PROVIDE THE INFO ABOUT BATCH PROGRESSES
THIS SYSTEM PROVIDE THE INFO ABOUT BATCH SCHEDUAL AND TIME
4.1.1
BATCH PROGRESS SYSTEM
4.1.2
BATCH SCHEDUAL SYSTEM


5.1
STUDENT
GIVE FEES AND TAKE RECEIPT
REQ. FOR FEES
FEE SYSTEM
STORE FEE DEATIL AND REQ FOR DETAIL
FEE DETAIL FILE
ACCESS FEE DEATIL AND GIVE DETAIL
GIVE FEE DETAIL
STORE FEE DETAIL AND REQ. FOR FEE DETAIL
GIVE INFO ABOUT DUE FEE
5.2
GIVE NOTICE DUE FEE STUDENT
STORE THE DUE FEE DETAIL
STORE DUE FEE DETAIL
RECOVERY SYSTEM
ACCOUNTS
DUE FEE DETAIL FILE
ACCES THE DUE FEE DETAIL

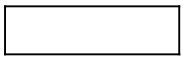

REQ FOR ATTENDANCE DETAIL
6.1
STORE AND REQ FOR ATTENDANCE DETAIL
COUNSULTANT
GIVE DUE FEE DETAIL
ATTENDANCE SYSTEM
ATTENDANCE FILE
GIVE ATTENDANCE DETAIL
GIVE AND ACCESS ATTENDANCE DETAIL
THIS SYSTEM PROVIDE DETAIL MONTHLY ATTENDANCE
THIS SYSTEM PROVIDE DETAIL WEEKLY ATTENDANCE
THIS SYSTEM PROVIDE DETAIL ABSENCE ATTENDANCE
6.1.1
6.1.2
6.1.3
MONTHLY ATTENDANCE SYSTEM
WEEKLY ATTENDANCE SYSTEM
ABSENCE SUMMARY SYSTEM
7.1
STORE STUDENT DETAIL
GIVE STUDENT DETAIL
STUDENT SYSTEM
COUNSULTANT
BATCH SCHEDUAL FILE
REQ. STUDENT DETAIL
GIVE STUDENT DETAIL
THIS SYSTEM SHOW TRANSFER STUDENT
THIS SYSTEM SHOW BREAK STUDENT
THIS SYSTEM SHOW RESUME STUDENT
7.1.1
7.1.2
7.1.3
TRANSFER STUDENT SYSTEM
BREAK STUDENT SYSTEM
RESUME STUDENT SYSTEM


REQ. FOR STUDY MATERIAL
8.1
STORE AND ACCESS DISTRIBUTION DETAIL
STUDENT
STUDY MATERIAL SYSTEM
DISTRIBUTION DETAIL FILE
GIVE STUDY MATERIAL
GIVE DISTRIBUTION DETAIL
THIS SYSTEM SHOW DISTRIBUTION OF BOOKS COURSE WISE
THIS SYSTEM SHOW DISTRIBUTION OF BOOKS BATCH WISE
8.1.1
COURSE WISE BOOKS DISTRIBUTION SYSTEM
8.1.2
BATCH WISE BOOKS DISTRIBUTION SYSTEM

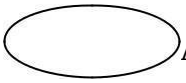

STORE AND ACCESS RESULT DETAIL
9.1
RESULT DETAIL FILE
GIVE RESULT DETAIL
STUDENT
RESULT SYSTEM
GIVE EXAM
TAKING EXAM
GIVE RESULT DETAIL
COUNSULTANT
9.1.1
TEST SYSTEM
REQ FOR RESULT DETAIL
9.2
CERTIFICATE SYSTEM
REQ. FOR CERTIFICATE

ENTITY RELATIONSHIP DIAGRAM

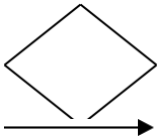
E-R Diagram is a way to represent conceptual database tools. The overall logical structure of a database can be expressed graphically by an E-R diagram, which consists of the following components :



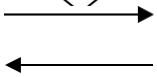
Entity



Attribute



Relationship

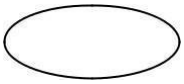


M : 1

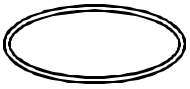
1 : M

1 : 1

M : M



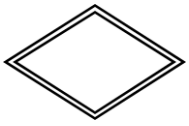
Derived Attribute

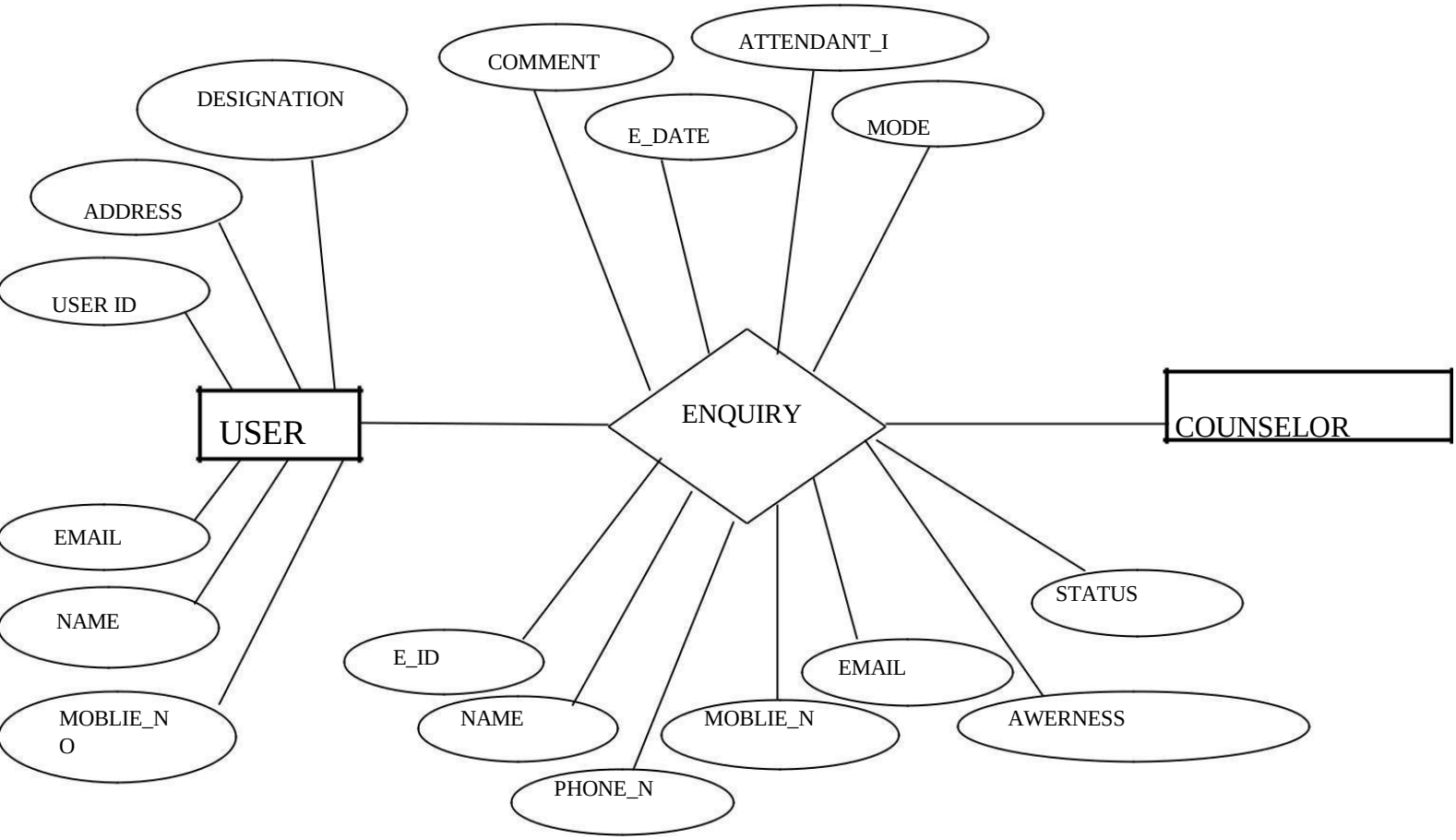


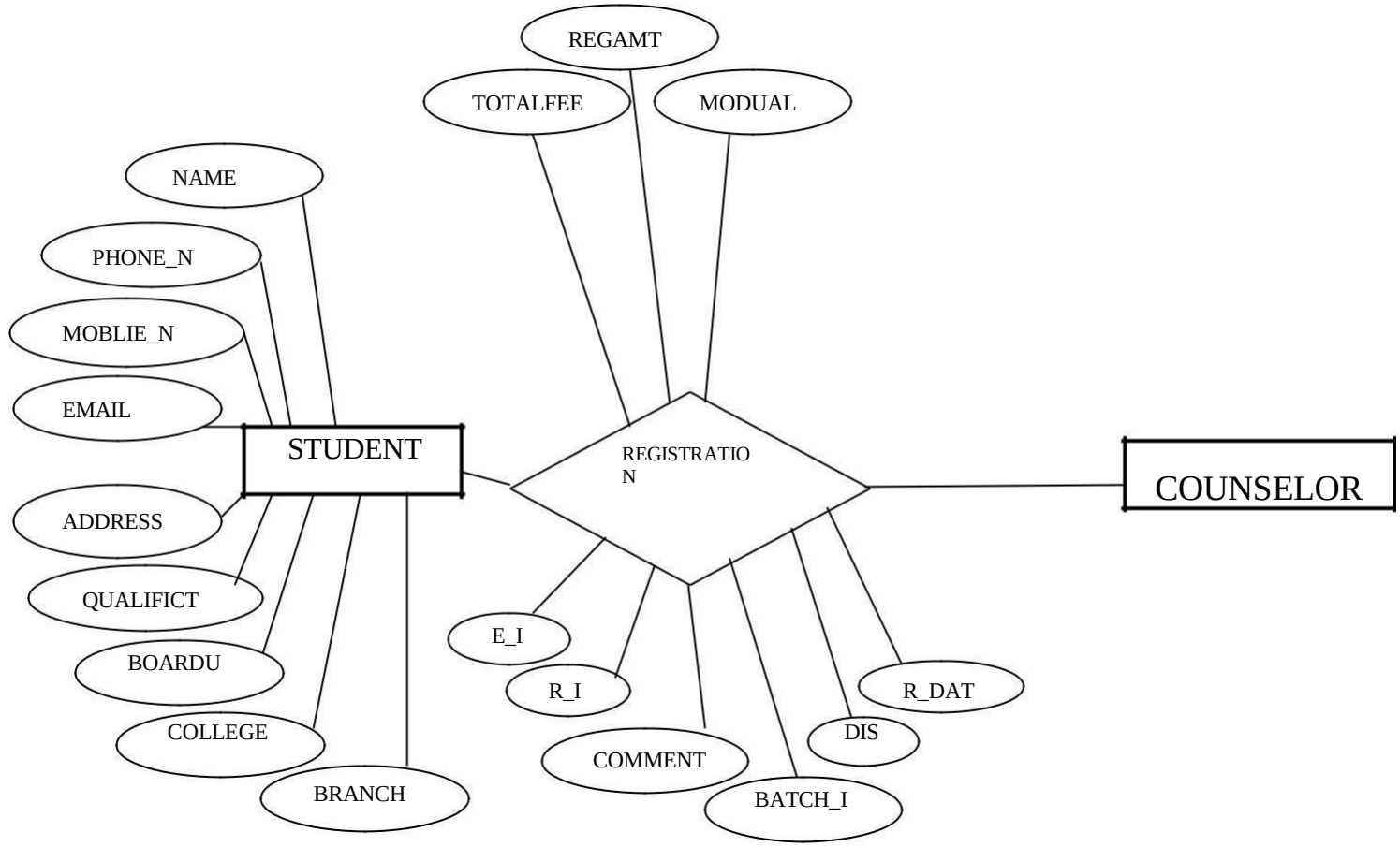
Multi-valued Attribute

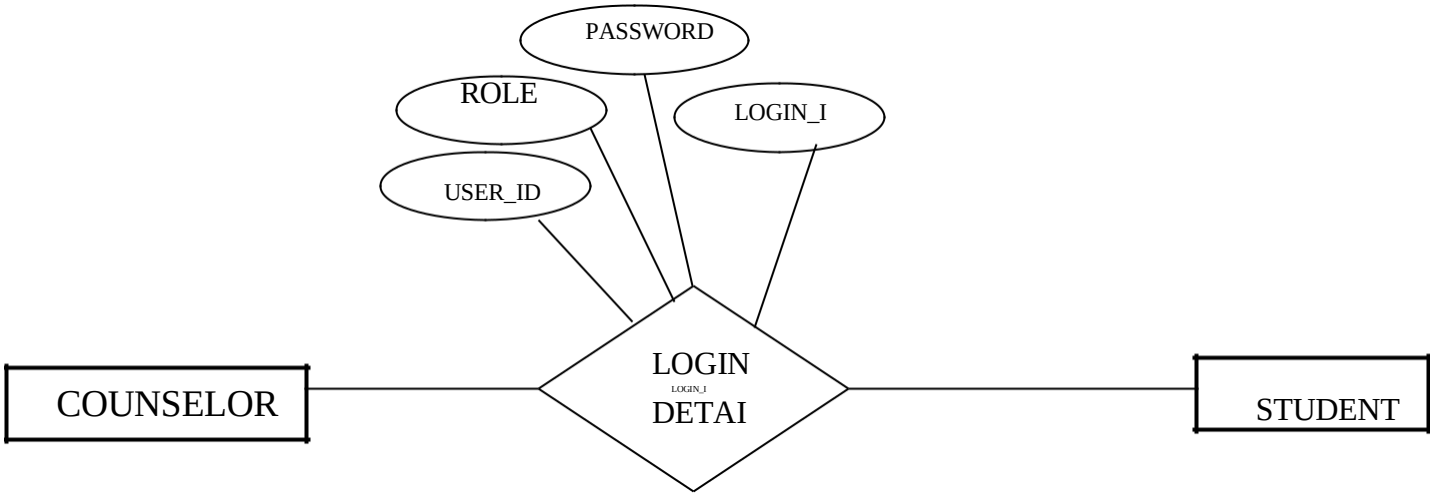
Weak Entity

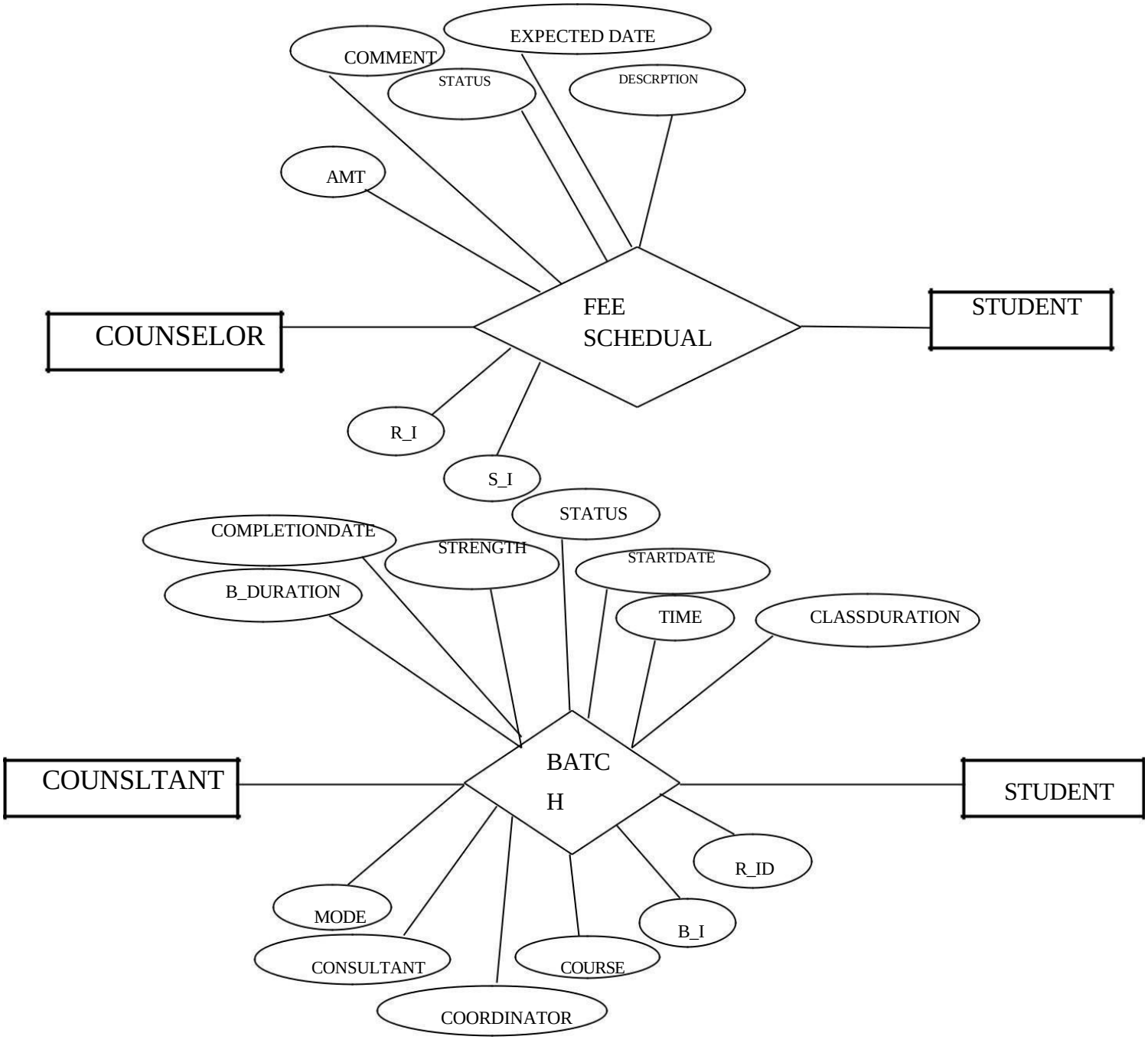
Weak Relationship

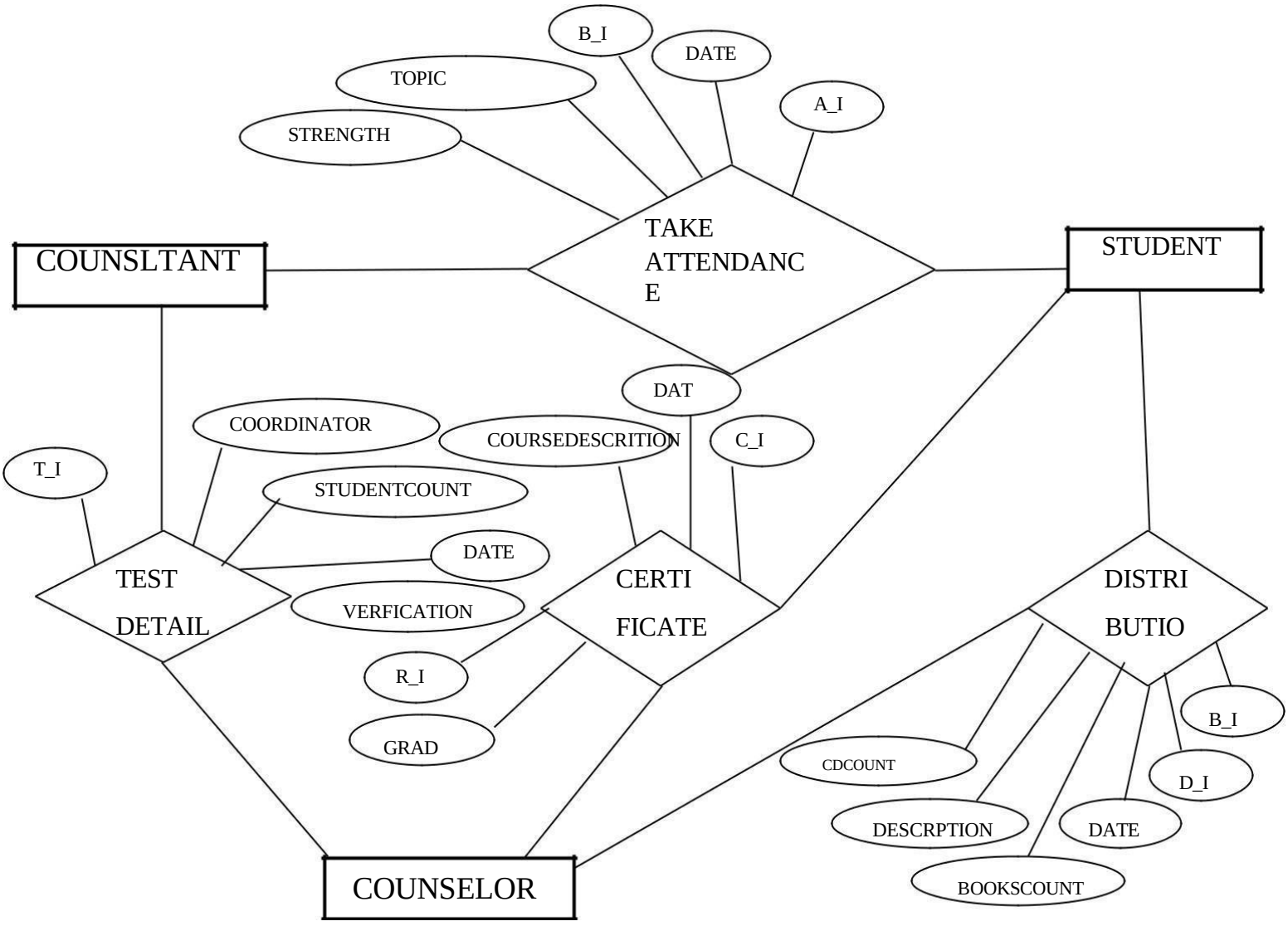



COMMENT
ATTENDANT_I
DESIGNATION
E_DATE
MODE
ADDRESS
USER ID
USER
ENQUIRY
COUNSELOR
EMAIL
NAME
MOBLIE_N
O
E_ID
NAME
MOBLIE_N
PHONE_N
EMAIL
STATUS
AWERNESS


REGAMT
TOTALFEE
MODUAL
NAME
PHONE_N
MOBLIE_N
EMAIL
STUDENT
REGISTRATIO
N
COUNSELOR
ADDRESS
QUALIFICT
BOARDU
COLLEGE
BRANCH
E_I
R_I
COMMENT
BATCH_I
DIS
R_DAT


PASSWORD
ROLE
LOGIN_I
USER_ID
LOGIN
LOGIN_I
DETAI
COUNSELOR
STUDENT


EXPECTED DATE
COMMENT
STATUS
DESCRPTION
AMT
FEE
SCHEDUAL
COUNSELOR
STUDENT
R_I
S_I
STATUS
COMPLETIONDATE
STRENGTH
STARTDATE
B_DURATION
TIME
CLASSDURATION
BATC
H
COUNSLTANT
STUDENT
R_ID
MODE
B_I
CONSULTANT
COURSE
COORDINATOR


B_I
DATE
TOPIC
A_I
STRENGTH
TAKE ATTENDANCE
COUNSLTANT
STUDENT
DAT
COORDINATOR
COURSEDESCRITION
C_I
T_I
STUDENTCOUNT
DATE
TEST DETAIL
CERTI FICATE
DISTRI BUTIO
VERFICATION
R_I
GRAD
B_I
CDCOUNT
D_I
DESCRPTION
DATE
BOOKSCOUNT
COUNSELOR

6. Complete Structure

Brief Description of the modules:

Enquiry Management: An Enquiry represents the query of a prospective customer about various courses offered by the institution. An enquiry can be made in the following three ways- In Person, by Phone, by Internet.
This module is responsible for storing the information of all enquiries into the system so that it can be retrieved later to generate various reports, to follow up non-converted enquires etc. An Enquiry that is stored into the system is either converted or becomes dead.

Registration: The Bread and Butter of all such institution depends on the no. of students that enroll themselves into the courses. This module is responsible for capturing and submitting the details of all the enrolled students into the system. This information is later retrieved to generate various reports. This module receives input from Enquiry Module and its output acts as input to the Fee and Recovery Module.

Query Management: This module deals with the replying of queries of existing students. Once students are enrolled, they request various type of information from time to time such as current topic of a specific batch, schedule of batches of a specific consultant etc.
This module mainly fetches required information stored in the system and presents it in easy to understand manner.

Batch Scheduling: An Institution that offers various courses and has various consultants needs a mechanism for proper scheduling of the batches. This module provides an interface to its user to insert the details of courses and consultants as well as to plan and view schedules.

Fee and Recovery Management: This module is responsible for entering fee details of the registered students into the system, generating due fee reports, collected fee reports, arrears and dead arrears. It also has the provision of factoring service tax into the collected fee, inclusively or exclusively as desired by the client.

Attendance Management: This module provides an interface to mark attendance of students, and generates batch wise, course wise daily, weekly and monthly attendance and absentee summary reports.

Student Management: Registering in one batch and then seeking transfer to another batch, discontinuing one batch and resuming another batch is a common phenomenon in all such institutions. This module deals with keeping track of all such transfers, breaks and resumption. It provides an interface to facilitate registering all transfers, breaks or resumptions into the system and generates student wise, course wise, consultant wise reports of breaks, transfers and resumptions.

Study Material Management: Mostly all such institutions provide study material in the form of books, printed notes and CDs to their students. This module deals with keeping track of course wise, batch wise distribution of study material. It provides an interface to submit the details of all available study materials into the system, details of all distribution into the system, and generates course wise, batch wise distribution reports as well as study material availability reports.

Result & Certification: All the professional institutions have the provision of awarding certificates after successful completion of the courses for this, in house examinations are conducted. This module provides an interface to submit results of examination as well as details of awarding certificates into the system. It generates course wise, month wise certificate distribution report as well as facilitate query of students results.

Data Structures

(1) **User:** This table shall store general information of the users. It contains following fields.

Field name	Data type	Constraint/Description
UserId	Varchar2	Primary key
Name	Varchar2	Not Null
Address	Varchar2	Not Null
MobileNo	Varchar2	Not Null
MailId	Varchar2	Not Null
designation	Varchar2	Not Null

(2) **Login Details:** This table shall store user information to be used at the time of login. It contains following fields.

Field name	Data type	Constraint/Description
UserId	Varchar2	Foreign key (User)
loginId	Varchar2	Login Id of the user, Unique
password	Varchar2	Not Null
Role	Varchar2	Not Null, (manager, accountant, consultant etc.)

(3) **Enquiry:** This table shall store enquiry information. It contains following fields.

Field name	Data type	Constraint/Description
enquiryId	Varchar2	Primary key

Name	Varchar2	Not Null
mailId	Varchar2	
phoneNo	Varchar2	
mobileNo	Varchar2	
enquiryDate	Date	Not Null
Mode	Varchar2	Not Null, (In Person, Mail, telephonic)
Status	Varchar2	(To be Attended, converted, on hold,dead)
attendantId	Varchar2	Foreign key (userId of userDetails)
courseName	Varchar2	Not Null
awernessSource	Varchar2	(Friend, News Paper Ad, etc.)
comment	Varchar2	Any comment marked by attendant.
followupDate	Date	Date on which enquiry is to be followed up.

(4) **Enquiry Follow up:** This table shall store enquiry follow-up information. It contains following fields.

Field name	**Data type**	**Constraint/Description**
enquiryId	Varchar2	Primary key
followupDate	Date	Not Null, Date on which enquiry is followed-up.
Mode	Varchar2	Not Null, (In Person, Mail, telephonic)
attendantId	Varchar2	Foreign key (userId of user Details)
comment	Varchar2	Any comment marked by attendant at the time of follow-up.

(5) **Student:** This table shall store general information of students. It contains following fields.

Field name	Data type	Constraint/Description
registrationId	Varchar2	Foreign key (Registration)
Name	Varchar2	
Address	Varchar2	
phoneNo	Varchar2	
mobileNo	Varchar2	Not Null
mailId	Varchar2	Not Null
qualification	Varchar2	Not Null
boardUniv	Varchar2	
College	Varchar2	Name of study Center in case of correspondence.
Branch	Varchar2	IT, CS etc.

(6) **Registration:** This table shall store information regarding registration of students in a batch. It contains following fields.

Field name	Data type	Constraint/Description
registrationId	Varchar2	Primary key
regDate	Date	Not Null, Registration Date
batchId	Varchar2	Foreign key (Batch)
regAmount	Number	Not Null
totalFee	Number	Not Null
Discount	Number	
comment	Varchar2	Specific timing or consultant etc.

enquiryId	Varchar2	Foreign key (Enquiry)
Modules	Varchar2	Not Null. (Modules to be covered.)

(7) **Fee Schedule:** This table shall store information about fee schedule of students. It contains following fields.

Field name	Data type	Constraint/Description
scheduleId	Varchar2	Primary key
registrationId	Varchar2	Foreign key (Registration)
Amount	Varchar2	Not Null, (Amount to be collected.)
description	Varchar2	Not Null, (First/second Installment etc)
expectedDate	Date	Not Null
Status	Varchar2	Not Null, (due, collected, arrear, dead)
comment	Varchar2	

(8) **Collection Details:**This table shall store information about fee collection. It contains following fields.

Field name	Data type	Constraint/Description
collectionId	Varchar2	Primary Key
scheduleId	Varchar2	Foreign key (Fee Schedule)
Amount	Varchar2	Not Null, (Amount collected.)
collection Date	Date	Not Null
Mode	Varchar2	Not Null (cash, check, credit card etc.)
comment	Varchar2	
Fee	Number	Not Null (amount – service Tax)

service Tax	Number	Not Null

(9) **Recovery Follow up:** This table shall store recovery follow-up information. It contains following fields.

Field name	**Data type**	**Constraint/Description**
scheduleId	Varchar2	Foreign Key (Fee Schedule)
followupDate	Date	Not Null
Mode	Varchar2	Not Null, (Phone, SMS, In person)
COORDINATOR	Varchar2	Foreign key, userId (User)
comment	Varchar2	Not Null

(10) **Batch:** This table shall store general information about batches. It contains following fields.

Field name	**Data type**	**Constraint/Description**
batchId	Varchar2	Primary Key
Course	Varchar2	Not Null
consultant	Varchar2	Not Null
COORDINATOR	Varchar2	Foreign key, userId (User)
Mode	Varchar2	Not Null, (WD, WE, MWF,TTS)
batchDuration	Varchar2	Not Null (Six Weeks, Four Moths, Six Months, etc.
classDuration	Varchar2	Not Null (1 hr, 2 hr, 3 hr etc.)
Time	Varchar2	Not Null
startDate	Date	Not Null (Expected in case status is Planned)
Status	Varchar2	Not Null (Planned, Running, On Break, Completed.)

completionDate	Varchar2	(Expected in case status is not Completed)
Strength	Number	(Expected in case status is Planned)

(11) **Batch Progress Detail:** This table shall store information about current and next topics of batches. It contains following fields.

Field name	**Data type**	**Constraint/Description**
batchId	Varchar2	Primary Key
currentTopic	Varchar2	Not Null
startedOn	Date	Not Null
nextTopic	Varchar2	Not Null
expDate	Date	Not Null.
updationDate	Date	Not Null
comment	Varchar2	

(12) **Batch Schedule:** This table shall represent schedule of the batches on a day. It contains following fields.

Field name	**Data type**	**Constraint/Description**
batchId	Varchar2	Foreign Key (Batch)
roomNo	Varchar2	Not Null
scheduleDate	Date	Not Null
Status	Varchar2	(Tentative, Permanent)

(13) **Attendance:** This table shall store summary of attendance in a batch on a day. It contains following fields.

Field name	**Data type**	**Constraint/Description**
attendanceId	Varchar2	Primary Key
batchId	Varchar2	Foreign Key (Batch)

Date	Date	Not Null
Strength	Varchar2	Not Null, (No. of Students Present)
topicCovered	Varchar2	Not Null

(14) **Attendance Details:** This table shall store details of attendance in a batch on a day. It contains following fields.

Field name	**Data type**	**Constraint/Description**
attendanceId	Varchar2	Foreign Key (Attendance)
registrationId	Varchar2	Foreign Key (Registration)

(15) **Break Details:** This table shall store details of students on break. It contains following fields.

Field name	**Data type**	**Constraint/Description**
registrationId	Varchar2	Foreign key (Registration)
batchId	Varchar2	Not Null (Identifies Batch form which break is taken.)
startDate	Date	Not Null
endDate	Date	Not Null
Topic	Varchar2	Not Null
authorisedBy	Varchar2	Foreign Key (userId (User))
comment	Varchar2	Not Null

(16) **Transfer Details:** This table shall store details of transfer of students from one batch to another. It contains following fields.

Field name	**Data type**	**Constraint/Description**
registrationId	Varchar2	Foreign key (Registration)
toBatch	Varchar2	Not Null
fromBatch	Varchar2	Not Null

transferDate	Date	Not Null
authorisedBy	Varchar2	Foreign Key (userId (User))
comment	Varchar2	Not Null

(17) **Distribution:** This table shall store batch wise summary of study material distribution. It contains following fields.

Field name	**Data type**	**Constraint/Description**
DistId	Varchar2	Primary Key
batchId	Varchar2	Foreign Key (Batch)
Date	Date	Not Null
books Count	Number	Not Null (Count of distributed books.)
cdCount	Number	Not Null (Count of distributed CDs.)
description	Varchar2	Not Null (Description of distributed books and CDs)

(18) **Distribution Details:** This table shall store details of study material distribution. It contains following fields.

Field name	**Data type**	**Constraint/Description**
DistId	Varchar2	Foreign Key (Distribution)
registrationId	Varchar2	Foreign Key (Registration)

(19) **Test:** This table shall store summary of tests conducted for awarding grades. It contains following fields.

Field name	**Data type**	**Constraint/Description**
TestId	Varchar2	Primary Key
Date	Date	Not Null

studentsCount	Number	Not Null (Count of students.)
COORDINATOR	Varchar2	Foreign key (userId (User))

(20) **Test Details:** This table shall store student wise details of tests. It contains following fields.

Field name	**Data type**	**Constraint/Description**
TestId	Varchar2	Foreign Key (Test)
registrationId	Varchar2	Foreign Key (Registration)
Grade	Varchar2	Not Null

(21) **Certificate Details:** This table shall store student wise details of certificates. It contains following fields.

Field name	**Data type**	**Constraint/Description**
CertId	Varchar2	Primary Key
registrationId	Varchar2	Foreign Key (Registration)
Grade	Varchar2	Not Null
Date	Date	Date on which certificate is distributed.
courseDesc	Varchar2	Not Null (Desc.Of the Course & Modules mentioned in the Certificate.)
verified By	Varchar2	Foreign Key (userId (User))

7. Security Mechanism:

Security means different things to different people depending upon their perspective. In the context of our product it means only valid users can login into the system and each user can only access the functionality authorized to the user. To prevent unauthenticated access, form based authentication implemented through Front Controller is to be used. To prevent unauthorized access, different roles are to be created by the administrator and access to features is to be controlled through these roles.

8. Future Scope and Further Enhancement:

Any product despite of its meticulous design and features needs enhancement with time. AUTOMATION OF E-FACILITATOR MONITORING being no exception needs active enhancement of features and functionality. Following features are proposed to be implemented in future to make the product more feature rich.

Intranet Messaging: In each organization, there is always a need of efficient paperless, secure, and private communication medium that has the retention capabilities. We are proposing to add Intranet messaging facility to fill this requirement as an independent module in future release.

Template Based Look & Feel: Being a product, it is proposed to be implemented at the site of different clients. Each client has different set of preferences for look and feel of the application. To minimize the customization process, look and feel of the application is proposed to be based on templates in future releases.

9. BIBLIOGRAPHY

Book Name	Author Name
Java-2 Complete Reference	by Patrick Naughton
Java Servlet Programming	by O'Reilly
MYSQL Developer 2000	by Wrox
Pure JavaScript	by Jason Gilliam, Charlton & Try, R.Allen Wyke
HTML Complete	BPB publications.
Java Server Programming	by Apress Publication.

All of the IGNOU blocks are also very useful.

PROJECT REPORT

INTRODUCTION OF THE PROJECT

E-F ACILITATOR is an intranet based Java application that automates the working of a professional institute that imparts professional training in the field of IT, Management, Accounting etc. this application is created as a product and can be customized according the specific needs of the client.

In “E-FACILITATOR” project, the two special advance programming languages are used as “J2EE” and “MYSQL”. The former is used as front end tool / for foreground and the latter is as back end tool / for background.

1. OBJECTIVE OF THE PROJECT

This project is being developed to serve the following objectives.

- ❑ Improve processing speed .
- ❑ Ability to handle large volume of information.
- ❑ Improve accuracy and reliability.
- ❑ Faster retrieval of information.
- ❑ Better security.

All professional institutions have same kind of problems to solve such as they run professional courses for which queries are received, students are enrolled, classes are conducted, batches are scheduled, dues are to be collected, record of batches, students and study materials is to maintained, various type of reports are generated, etc.

E-F ACILITATOR provides solution of all these problems in the form of easy to configure and use application by automating all these functionality. Its major modules are:

- Enquiry Management
- Attendance Management
- Student Management (Transfer, Break, Resume & Backup)

2. TOOLS AND PLATFORM

This software is a database application. The tools used for developing the projects are as follows:

1. **MYSQL**
2. **J2EE (servlet, jsp and beans)**
3. **Weblogic 10.3**

J2EE-

J2EE is a Product of Sun Microsystems That is used for developing Web based application. This software include SERVLET,JSP,BEAN .These component a large number of graphical components that are generally used in developing GUI applications.

The J2EE programming system provides with many kinds of tools to create attractive and useful applications. This makes us more productive by providing appropriate tools for the different aspects of GUI development. The objects like text box, list box, picture box, labels can easily be drawn with the helps of JSP. And after designing the interface we make this interface interact with the user by writing code help of Bean class that responds to events that occur in the interface.

MYSQL-

MYSQL is Object Relational database Management System (ORDBMS). It offers capabilities of both relational and object oriented database system. In general objects can be defined as reusable software codes which are location independent and perform a specific task on any application environment with little or no change to the codes.

SQL is used to access the data within the MYSQL. It contains a set of commands, which make it very easy to maintain the database. It has for sub parts DDL, DML, DCL, and TCL.

- ✓ DDL includes the commands, which allows us to create objects and to manipulate the structure of the objects.
- ✓ DML includes the commands to manipulate the information stored in a database.
- ✓ DCL includes the commands for controlling the data access and
- ✓ TCL includes the commands for controlling the transactions like commit and rollback.The database server or back-end is used to manage the database files optimally among multiple clients who concurrently request the server for the same

data. It also enforces Data Integrity across all client application and controls database access and other security requirements.

HARDWARE AND SOFTWARE REQUIREMENT:-

Hardware Specification:

It is recommended that the minimum configuration for clients is as appended below.Suggested Configuration of Windows clients:-

Microprocessor : Pentium-4 class processor, 450 (MHz)
Ram : 512 MB of RAM
CD ROM Drive : 52 X CD ROM Drive
Hard Disk : 40 Gigabytes (GB) on installation drive,

Software requirements:

- **Windows 98 / XP operating system**
- **MYSQL**
- **JDK 1.6**
- **Weblogic 10.3**

SYSTEM ANALYSIS

- **Requirements Definition**-

A software requirement is an abstract description of the services that the system shall provide and the constraints under which the system must operate.Requirements determination involves studying the current business system to find out how it works and where and where improvements should be made. System studies results in an evaluation of how current methods are working and weather adjustment are necessary or possible.

A requirement is a feature that must be included in a new system. It may include a way of capturing of processing data, producing information, controlling a business activity, or supporting management. The determination of requirements thus entails studying the existing system and collecting details about it to find out what these requirements. Requirements can either be functional or non functional.

- **Requirement Analysis**– Requirement Analysis is a software Engineering task that bridges the gap between system level requirements engineering and software design. In the proposed project software requirements analysis have been divided in to five areas of effort.

1. problem recognition
2. evaluation and synthesis
3. modeling
4. specification
5. review

Requirements Elicitation for the software:- Before requirements can be analyzed, modeled or they are gathered through an elicitation process.

Context free questions were asked to the management people belonging to different large organizations /institute regarding how they would characterize a good output that would generate a successful solution, what kind of problems will this solution address, how they describe the environment in which the solution will be used and will special performance issues or constraints effects the way the solution is approached.

- **QUALITY FUNCTION REQUIREMENT** :

Quality function development (QFD) is a quality management technique that translates the needs of the customer into

technical requirements for software.

In QFD three types of requirement are identified

Normal Requirements:

1. Graphical displays:

a) Fully menu driven.

b) Intuitive key assignments and user interactive screen.

c) User configurable.

2. Backup and restore facilities.

3. Facility to add, delete, modify a user record.

4. Report generation.

❖ Expected Requirements:

These requirements are implicit to the product or system and may be so fundamental that the customer dose not state them.

The following are listed.

1. Indexing
2. Ease of human / machine interaction
3. Reliability and operational correctness
4. Ease of software installation
5. Single point data storage for data element
6. Maintenance of integrity and inter – linkage of data
7. Extensive query facility to provide immediate answers for management
8. Matching of physical and logical movement of file
9. Should be upgradeable to incorporate new features
10. Should be expandable
11. Should have fastest possible response while processing queries, reports and updates

❖ Exciting requirements:

1. Error control mechanism
2. Graphical animations
3. Other look and feel appeals

❖ Security requirements:

The following security requirements are considered in this project.

1. User Level Authentication
2. Restricted Menu access
3. Backup and Restore

❖ Functional requirements:

- **Users Details** – This report contains the complete personal details about the user like user's id, name, address, mobile no, mail id , designation, login id, password, role and technology.
- **Course details** – This report includes the description of the course. report contains customer's name, course fee and technology provide to student by institution.
- **Technology detail report –** This report include the description of the technology such as technology id, name and description.

❖ Non – functional requirements:

The Non-functional requirements can be classified into three main categories that are:

1. Product requirements
2. Organizational requirements
3. External requirements

Product requirements :- These are the expectations from a system. Product requirements include the following:

Usability requirements:-

The system should have a good interface and it should be used friendly because it the user who would be using the system most of the time. Along with the user friendly interface the system should be easy to use and should not be confusing. A list of command or instructions on how to work with the system and operate it should be provided with the system.

Efficiency Requirements:-

To make the system run efficiently system requirements of the software will have to be kept minimum.

Reliability Requirements:-

A system is said to be reliable is a user can depend on it. The process be execute in the same manner it has been programmed.

And the out puts of the project must be very correct. If outputs are not reliable then there is no advantage of using the software. Moreover, the database must be handled very carefully because loss of the data may result in bankrupting or closing the organization.

❖ Feasibility Study:-

Feasibility study is the determination of whether or not a project is worth dong. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalize the project requirements.

Different type of feasibility study undertaken is:

Technical feasibility:-

This is concerned with specifying equipments and will satisfy the user requirements. The technical needs of the system may vary considerably but might include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.
- Facility to communicate data to distant location.

Technical feasibility centers on the existing computer system, hardware, software, etc and to what extent it can support the system. In examination technical feasibility configuration of the system is given more important that the actual make of hardware. The configuration should give the complete picture of the system requirements e.g. how many workstations are required, how these units are interconnected so that they would operate smoothly, etc. the result of technical feasibility study is used as the basic for documents against which dealer and manufactures can make the bids. Specified hardware and software product can then be evaluated keeping in view the logical needs.

Economic Feasibility:-

Economic feasibility is the most frequently used technique for evaluating the effectiveness of the proposed system. More commonly knows, as cost/benefit analysis; the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement a system. Otherwise, further justification or alternative in a proposed system will have to be made if it is to have a chance of being approved. This is an outgoing efforts that improves in accuracy at each phase of system lifecycle.

Time feasibility:-

It is determination of whether a proposed project can be implemented fully with in a stipulated time frame. If a project takes too much time it is likely to be rejected. Time feasibility is important because the client companies usually give deadlines for the system to be changed. Usually give deadlines for the system to be changed. Usually deadlines are very strict and should be followed by companies.

Resource feasibility:-

This was also important to check whether required resources will be available or not. For the software, there were no such constraints as most of the web-related language is platform independent and freely available. Here, I decided to use J2EE as the most prominent tool, the compiler for which is freely available. For the database, a general approach was adopted by means of which the same coding can be implied on all the web-related database management system software available. The database may be implemented on mysql.

Behavioral feasibility:-

Normal human psychology of human being indicates that people are resistant to change and computers are knows to facilitate. Any project formulations should consider this factor also. Before the development of the project titled "telephone billing system", the need to study the feasibility of the successful execution of the project was felt and thus the following factors are considered for a feasibility study.

1. Need Analysis.
2. Provide the users information pertaining to the above requirements.

Feasibility study report:

Questionnaires were used to gather information from the key users i.e. the customers, operators and managements to determine the following:

- Whether there is a need for an automated system.
- If the need is there, what are the drawbacks of existing system that have to be rectified in the new system?
- That omissions are there in the existing systems, which have to be incorporated in the new system?

The result of the feasibility study provided us with the following facts:

1. The automated system would increases the efficiency of the system.
2. The automated system would increases customer's satisfaction.
3. The automated system has many requirements such as
    a. Efficiency
    b. Cost effectiveness
    c. Prompt services
    d. Reliability
4. The automated system would add to the security features of the system
5. The automated system should be
6. Simple to use
7. Incorporate all necessary services
8. Maintainable
9. This will cause some changes in the organization. These are:-
10. Change in staffing policies-present employees will have to be sent for training.

- New employees to be recruited will have the knowledge about the automated system.

Software Engineering Paradigm

Software Engineering Paradigm is also referred as software Life Cycle Model or a process model. A software life cycle is a series of identifiable stage that a software product undergoes during its life time.

A software product development effort usually starts with a feasibility study stage, and then subsequently requirements analysis & specification, design, coding, testing and maintenance are undertaken. Each of these stages is called life cycle phase.

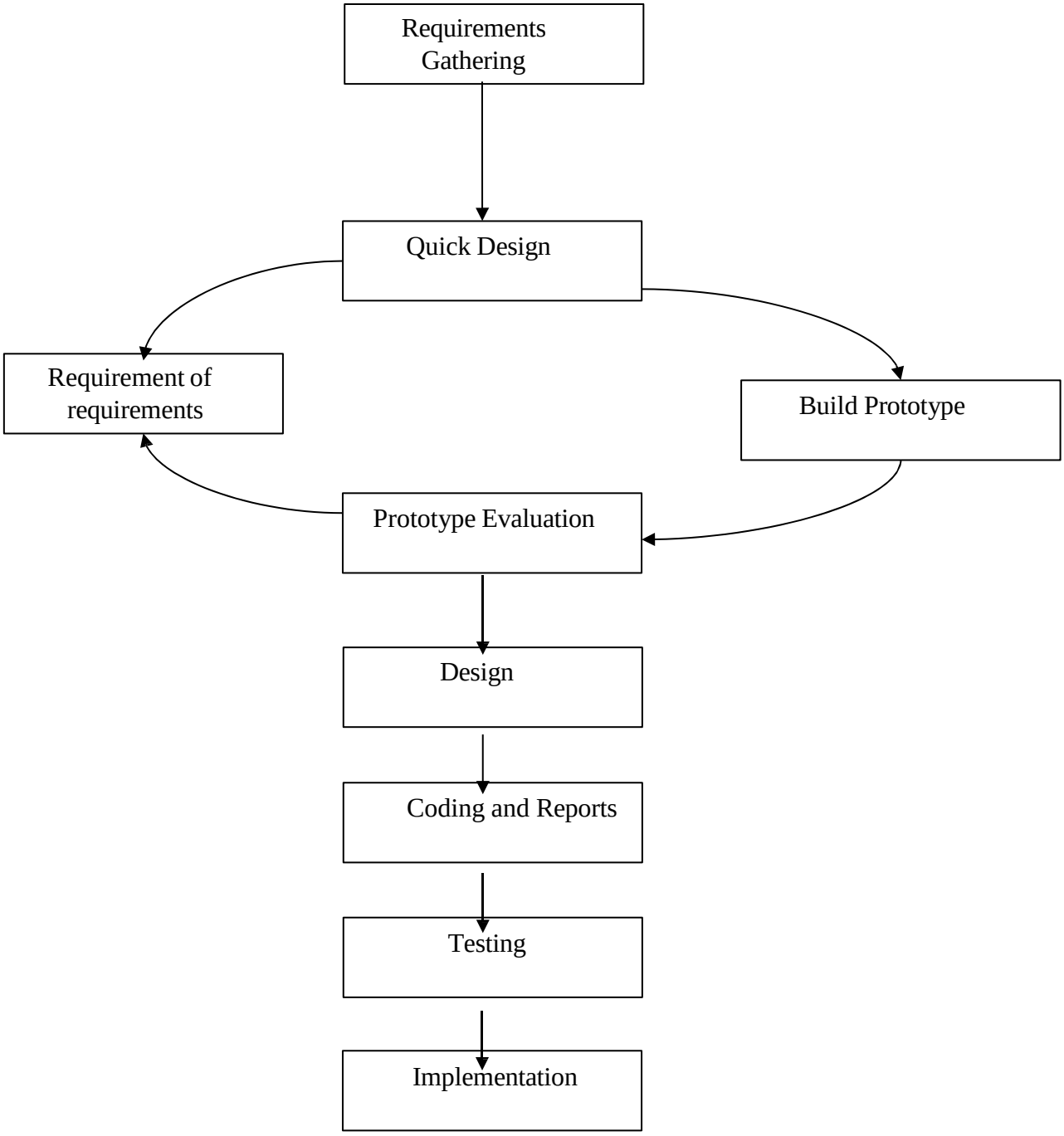
While developing a software product, it is necessary for the development team to identify a suitable life cycle model and adhere to it. The primary advantage of adhering to a life model is that it encourage development of software in a systematic and disciplined manner. There are several software engineering paradigms (process module) for developing software. Some of the important and commonly used models are as under:

- Linear Sequential model (also called "classis life cycle" or the "waterfall model")
- Iterative waterfall model
- Prototyping model – applied in to the "project information system"
- Evolutionary model
- Rapid application development (RAD) model
- Spiral model

Prototyping paradigm of software Engineering is applied in the "telephone billing system" the prototyping model suggests that before development of the software, a working prototype of the system should be built first. There were several reasons for developing a prototype. An important purpose was to illustrate the input data formats, reports, and the interactive dialogues to the end user. This was a valuable mechanism for gaining better understanding of the Business Development Division's needs. For much easier for the user to form his option by experimenting with a working model rather than just trying to imaging the working of a hypo- thetical system.

After gathering the requirements from the Business Development division, a quick design (prototype) was made and showed to the HODs of information Technology and Business Development Division. The system prototype built was quit close to the requirements of the Business Development Division. Some additional requirements were also added to the system on the system on the pretext of slight increase in the earlier scope. Following figure depicts the prototyping paradigm applied in the "E-FACILITATOR".

As shown in the figure the model starts with an initial requirements gathering phase. A quick design was carried out and the system proto- type was built. The developed prototype was then shown to the end user for their suggestion on it. Based on the feedback, the requirements were refined and classical waterfall approach was used to develop the"E-FACILITATOR".


Requirements Gathering
Quick Design
Requirement of requirements
Build Prototype
Prototype Evaluation
Design
Coding and Reports
Testing
Implementation

PROJECT SCHEDULING:

Scheduling is an important activity of project Managers. In order to schedule a project, a software project manager to do the following:

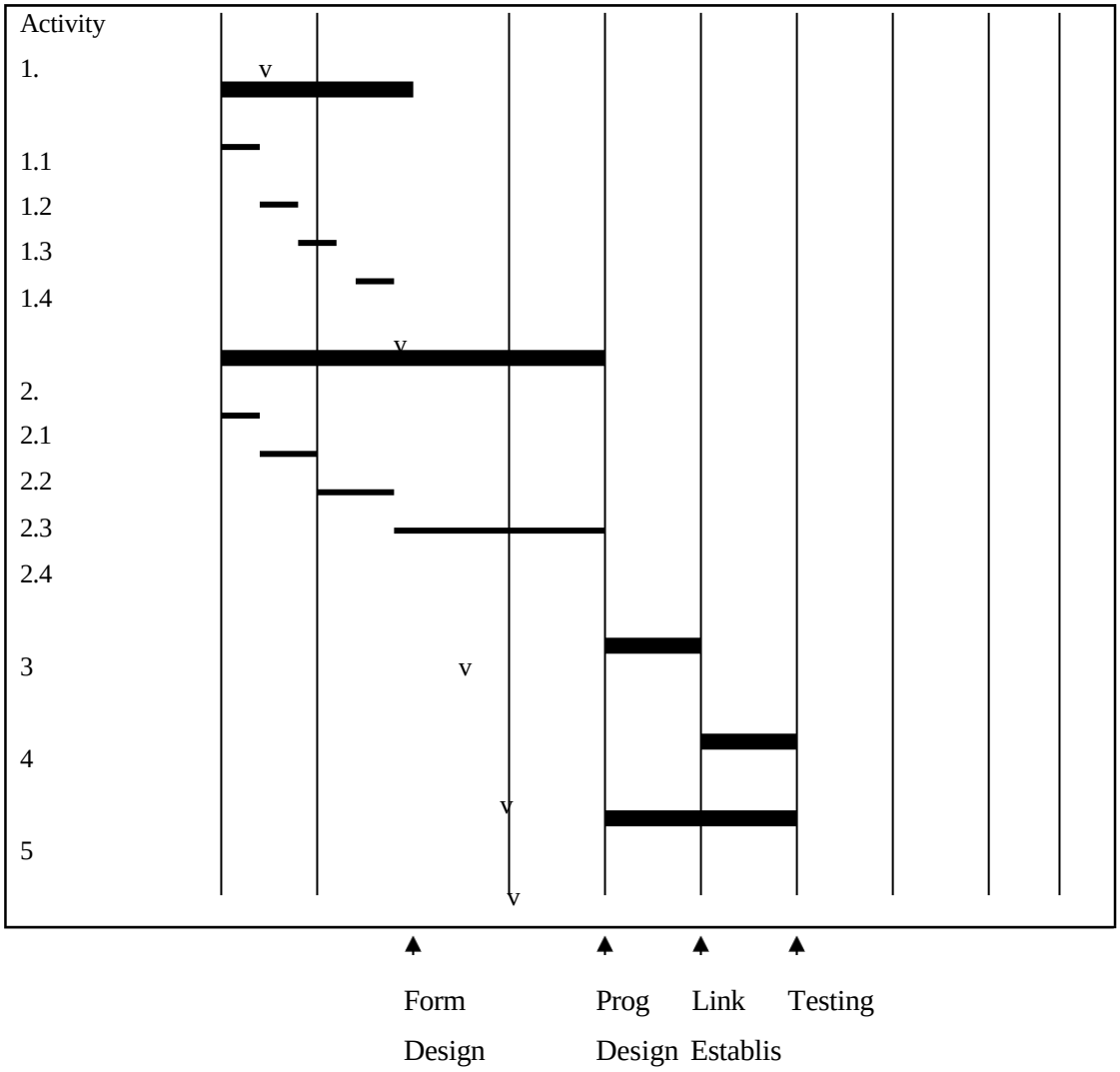
- Identify the tasks needed to complete the project.
- Determine the dependency among different tasks.
- Establish the most likely estimates for the duration of the identified tasks.
- Plan the starting and ending dates for various tasks.
- Determine the critical path (i.e. the chain of tasks that determine the duration of the project).

As for as the "E-FACILITATOR" is concern, the whole system was divided in to main activities shown in **PERT charts**. Dependencies between these activities; duration of each activity, its starting and ending dates; and total duration of the project can be easily identified by going through these charts.

GANTT CHART

Gantt charts (developed by Henry L. Gantt) are a project control technique that can be used for several purposes including scheduling, budgeting and resource planning. A Gantt chart is a bar chart with each bar representing an activity. The bars are drawn against a time line. The length of each bar is proportional to the length of time planned for the activity.

Gantt chart can take different form depending on their intended use.



PERT CHART

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.

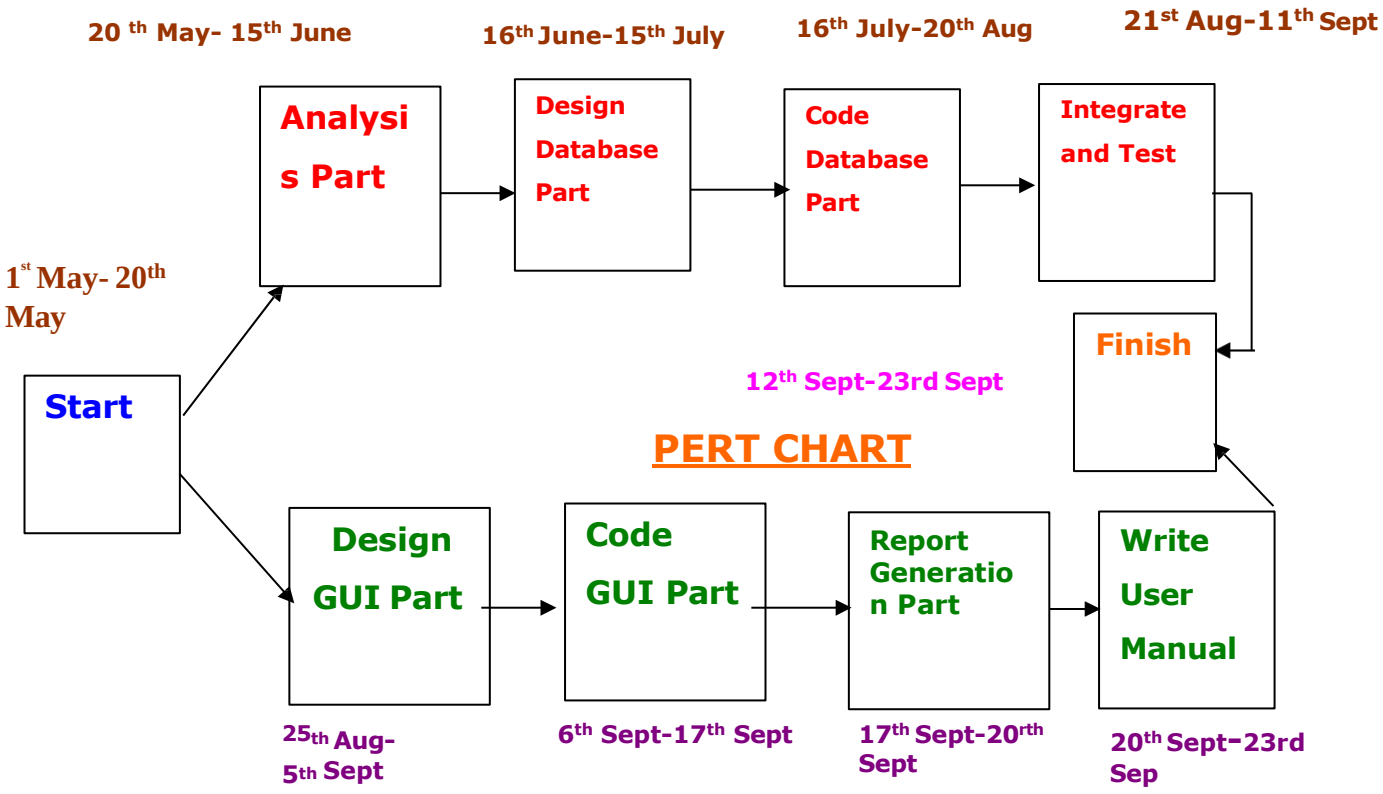
PERT is organized by events and activities or tasks. PERT have more advantages and they are likely to be used for more complex projects.

Through PERT chart the various task paths are defined. PERT enables the calculation of critical path. Each path consists of combination of tasks, which must be completed. The time and the cost associated with each task along a path are calculated, and the path that requires the greatest amount of elapsed time is the critical path. Calculation of the critical path enables project manager to monitor this series of task more closely than others to shift resources to it, if it begins to fall behind schedule.

PERT controls time and cost during the project and also facilitate finding the right balance between completing a project on time and completing it within the budget.

There are thus not one but many critical paths, depending on the permutations of the estimates for each task. This makes analysis of critical path in PERT charts very complex. The PERT chart representation of the club Management system.

PERT CHART OF E-FACILITATOR:-



❖ USE CASE DIAGRAM

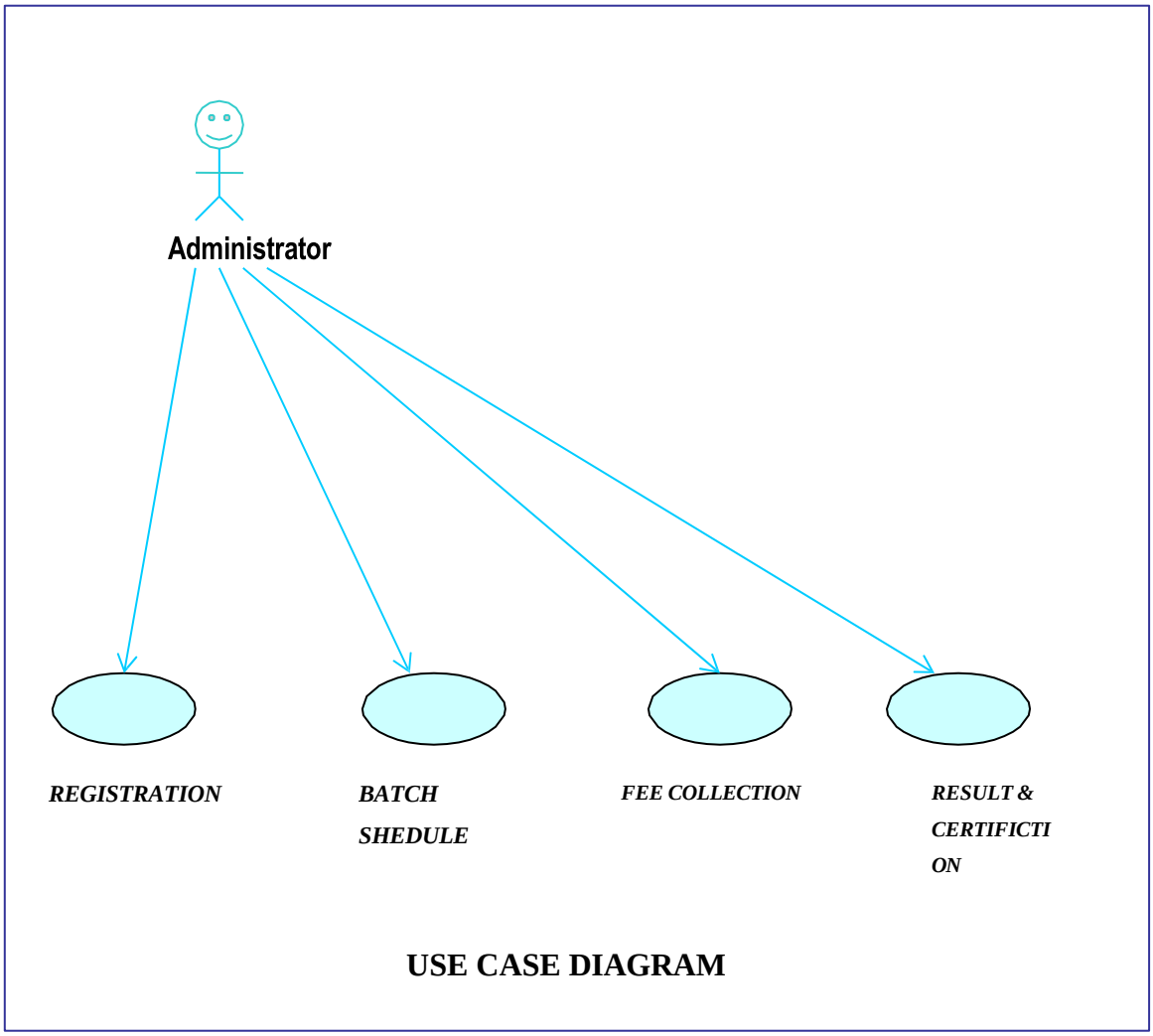
A **use case diagram** is a type of **behavioral diagram** defined by the Unified Modeling Language (UML) created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals—represented as use cases—and any dependencies between those use cases.

UML standard defines a graphical notation for modeling use cases with diagrams, but no format for describing these use cases. While the graphical notation and descriptions are important, they are documentation of the use case—a purpose that the actor can use the system for—

The true value of a use case lies in two areas:

The written description of system behavior regarding a business task or requirement. This description focuses on the value provided by the system to external entities such as human users or other systems.

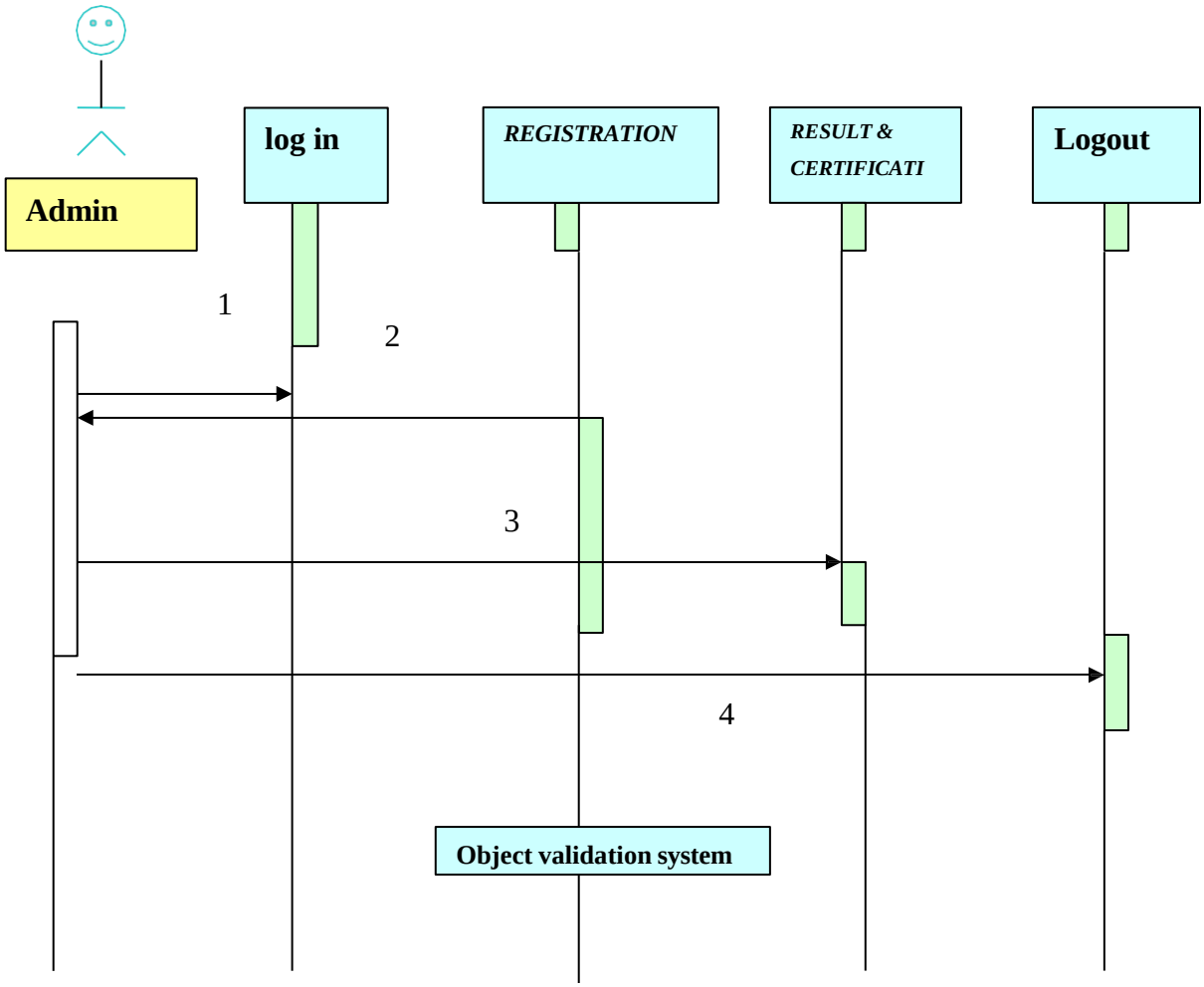
The position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the customer/owner/user and the development team.



USE CASE DIAGRAM

❖ SEQUENCE DIAGRAM

The Message Sequence Chart technique has been incorporated into the Unified Modeling Language (UML) diagram under the name of **Sequence Diagram**. A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



Sequence Diagram

DATA FLOW DIAGRAM

Graphical description of a system's data and how the processes transform the data is known as Data Flow Diagram (or DFD).

Unlike detail flowcharts, DFDs do not supply detailed descriptions of modules but graphically describe a system's data and how the data interact with the system.

To construct data flow diagrams, we use:

i. arrows,

ii. circles,

iii. open-ended boxes, and

iv. squares

An arrow identifies data flow - data in motion. It is a pipeline through which information flows. Like the rectangle in flowcharts, circles stand for a process that converts data/into information.

An open-ended box represents a data store - data at rest, or a temporary repository of data.

A square defines a source (originator) or destination of system data.

The following rules govern construction of data flow diagrams(DFD):

1. Arrows should not cross each other.

2. Squares, circles, and files must bear names.

3. Decomposed data flows must be balanced (all data flows on the decomposed diagram must reflect flows in the original diagram).

4. No two data flows, squares, or circles can have the same name.

5. Draw all data flows around the outside of the diagram.

6. Choose meaningful names for data flows, processes, and data stores. Use strong verbs followed by nouns.

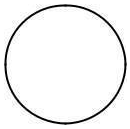
7. Control information such as record counts, passwords, and validation requirements are not pertinent to a data-flow diagram.

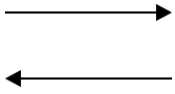
If too many events seem to be occurring at a given point, an analyst can decompose a data conversion (circle). The new data conversions form a parent-child relationship with the original data conversion: the child circle

.

Symbols Used in DFD :

Square  so**urce/Destination of Data**

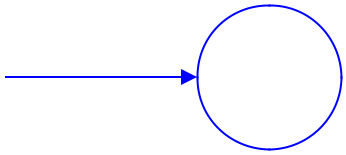
Bubble  **Process For Transformation of Data**

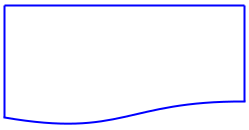
Arrows  **Data in motion (Data flow)**

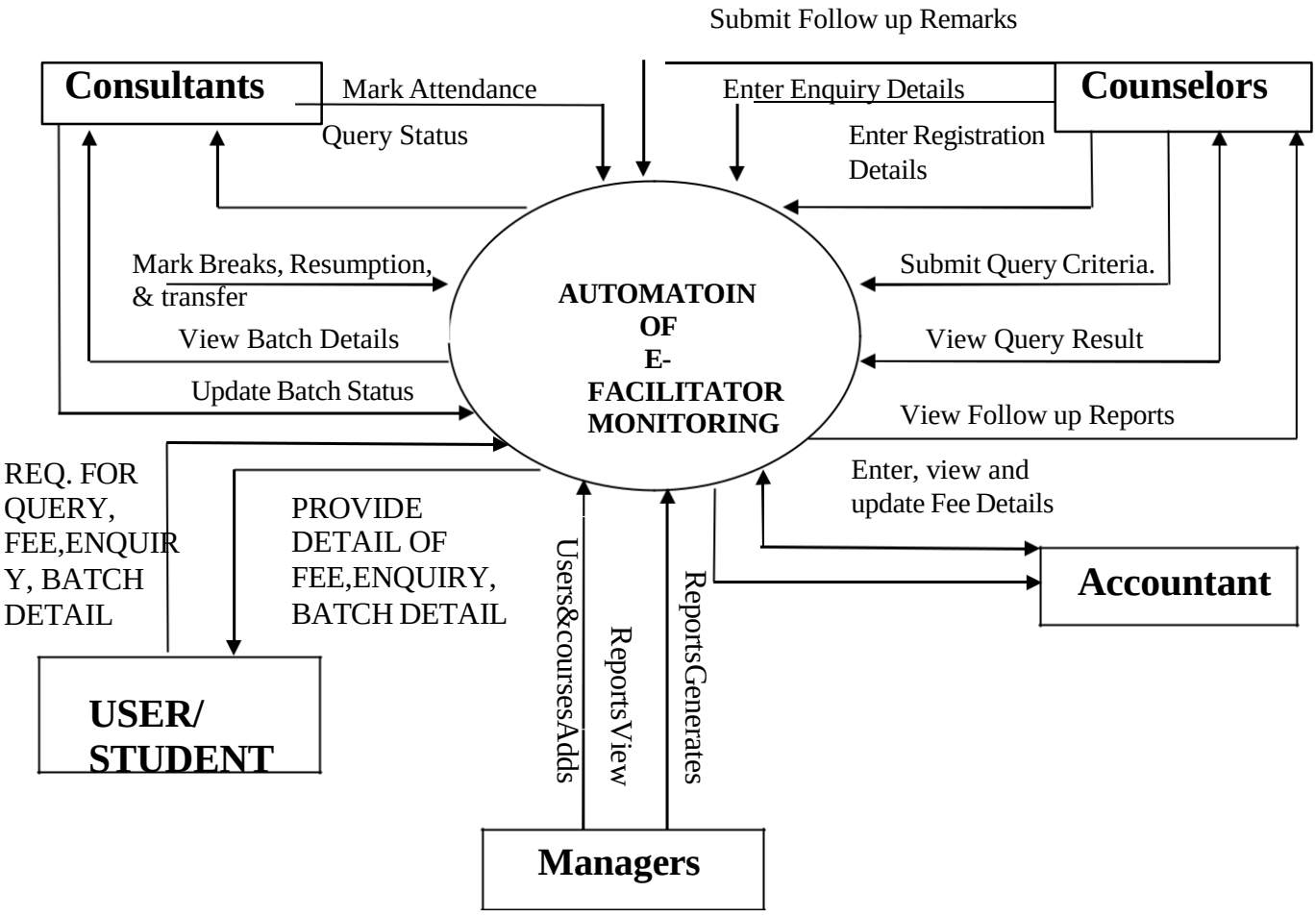
Rectangle  **Data at rest (Data Source)**

Open Rectangle

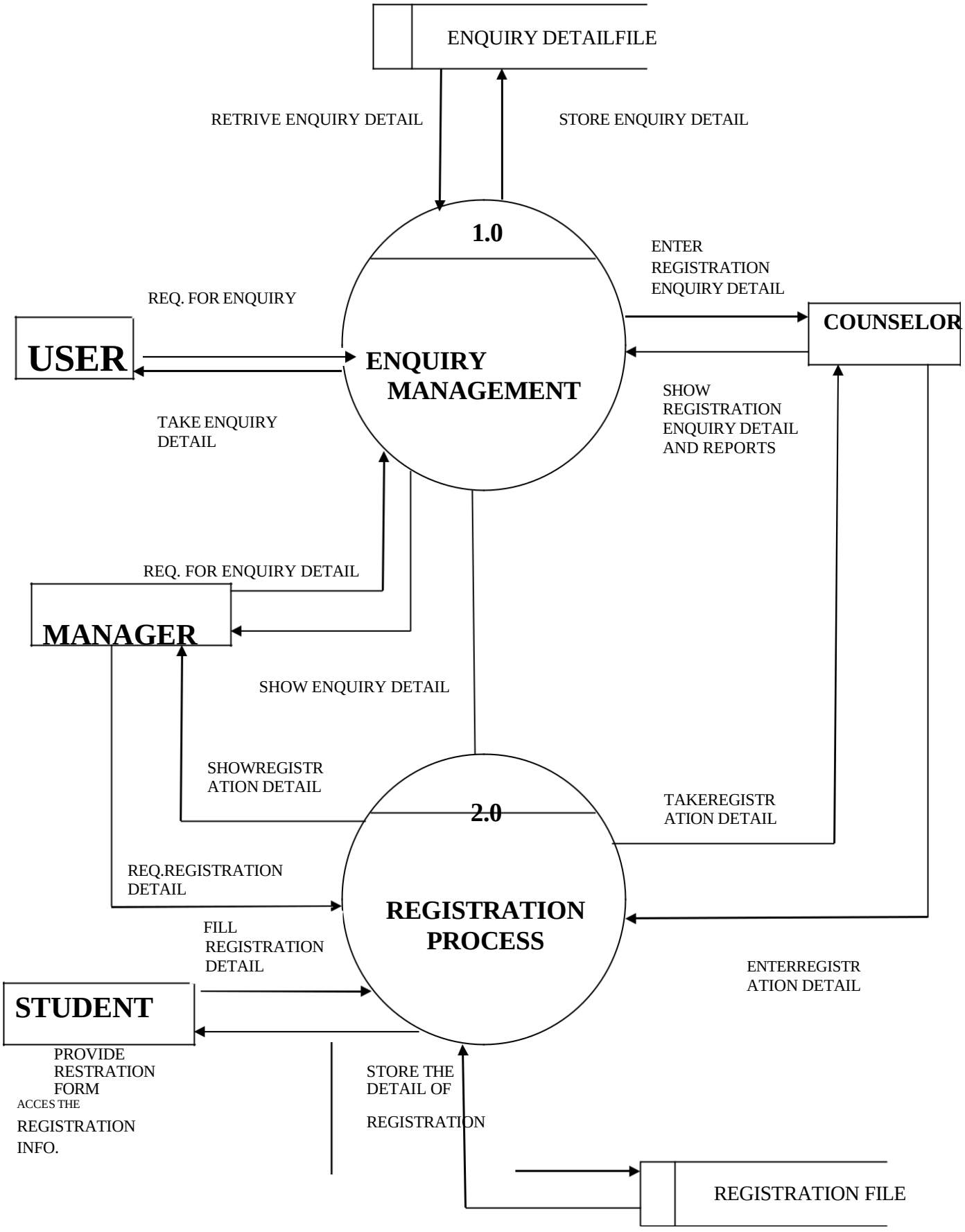
Data at rest (Data Source) 

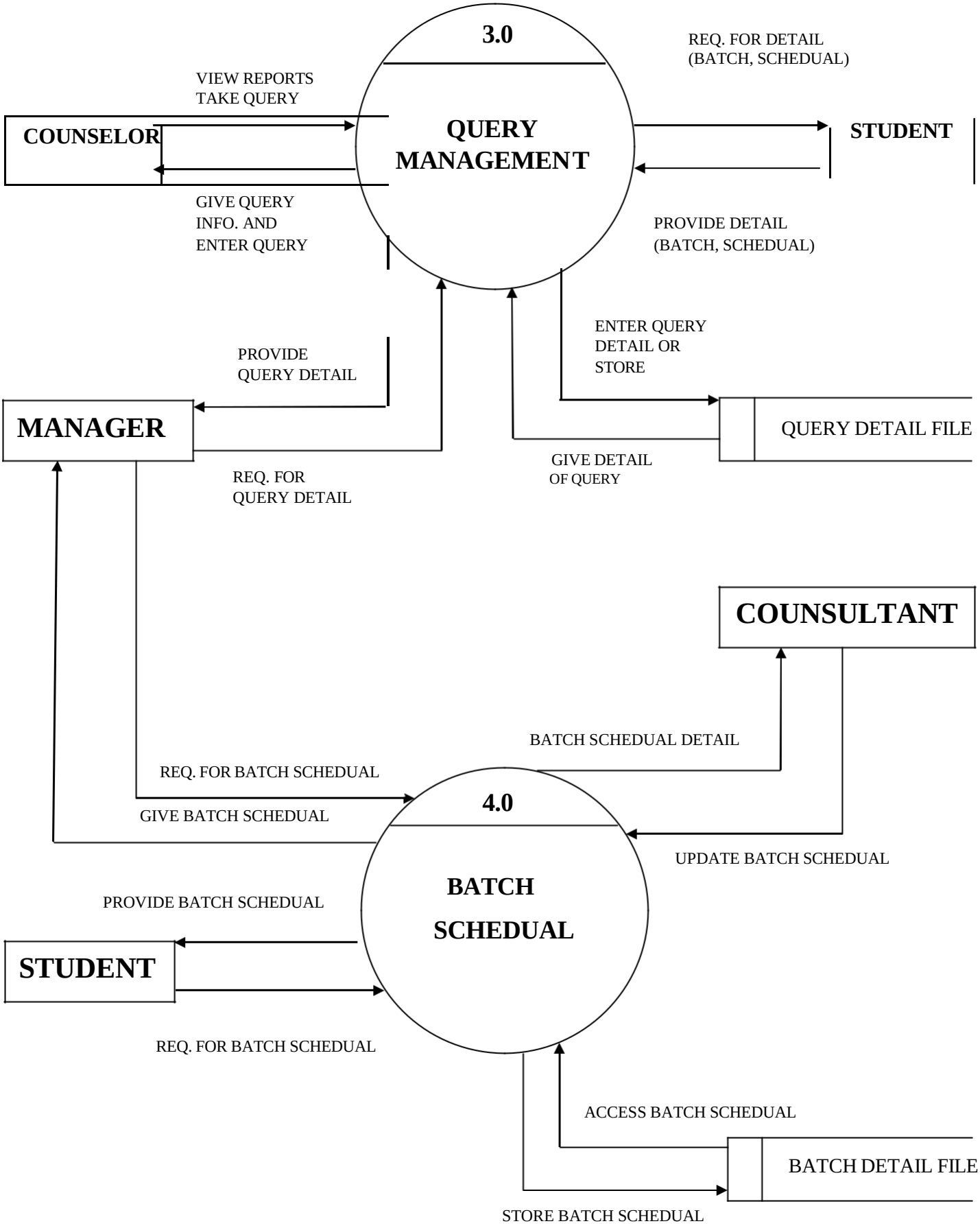
Connector 

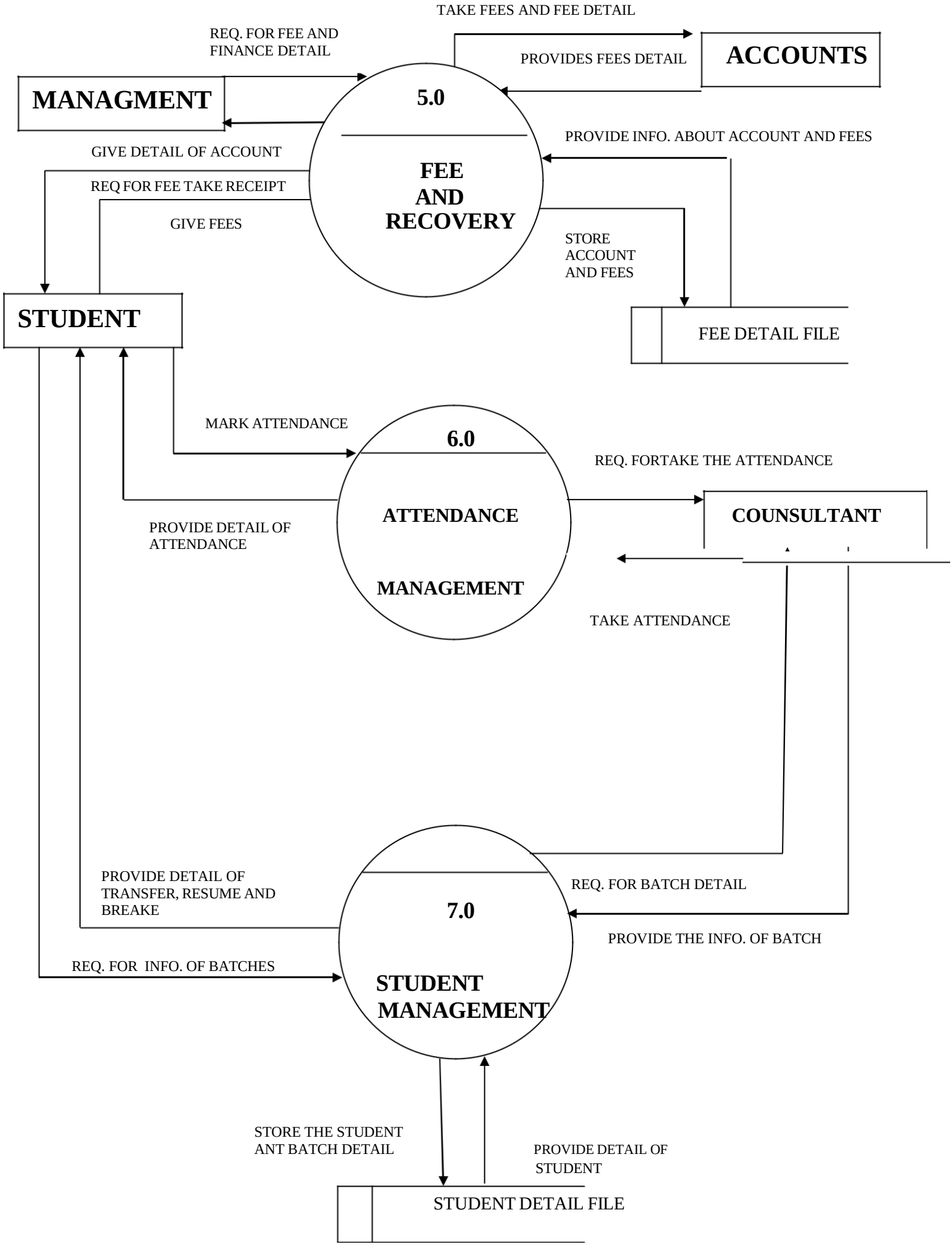
Document 

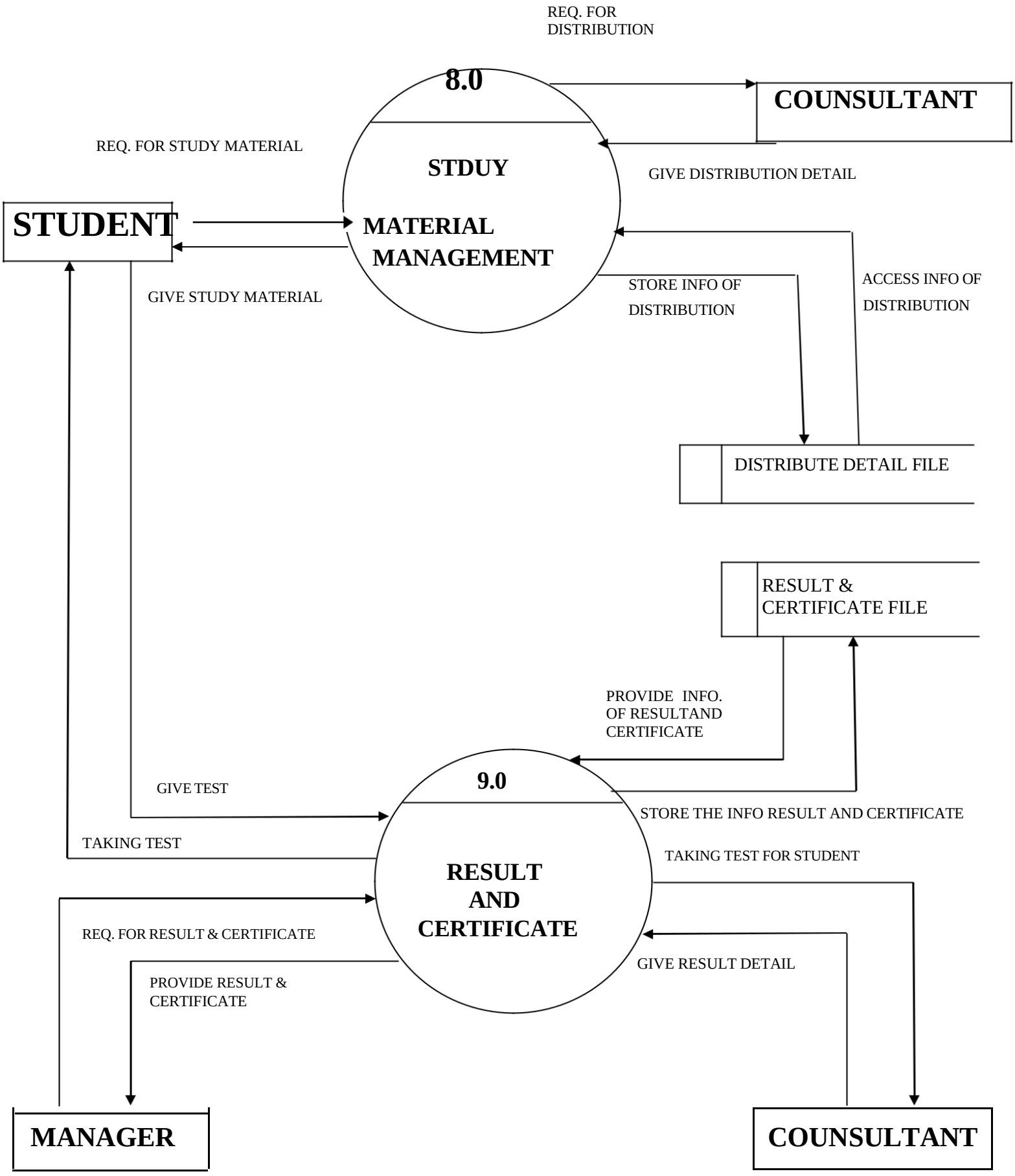

Submit Follow up Remarks
Consultants
Mark Attendance
Query Status
Enter Enquiry Details
Counselors
Enter Registration Details
Mark Breaks, Resumption, & transfer
View Batch Details
Update Batch Status
AUTOMATOIN OF E-FACILITATOR MONITORING
Submit Query Criteria.
View Query Result
View Follow up Reports
REQ. FOR QUERY, FEE,ENQUIRY, BATCH DETAIL
PROVIDE DETAIL OF FEE,ENQUIRY, BATCH DETAIL
Users&coursesAdds
ReportsView
ReportsGenerates
Enter, view and update Fee Details
Accountant
USER/ STUDENT
Managers

FIRST LEVEL DFD

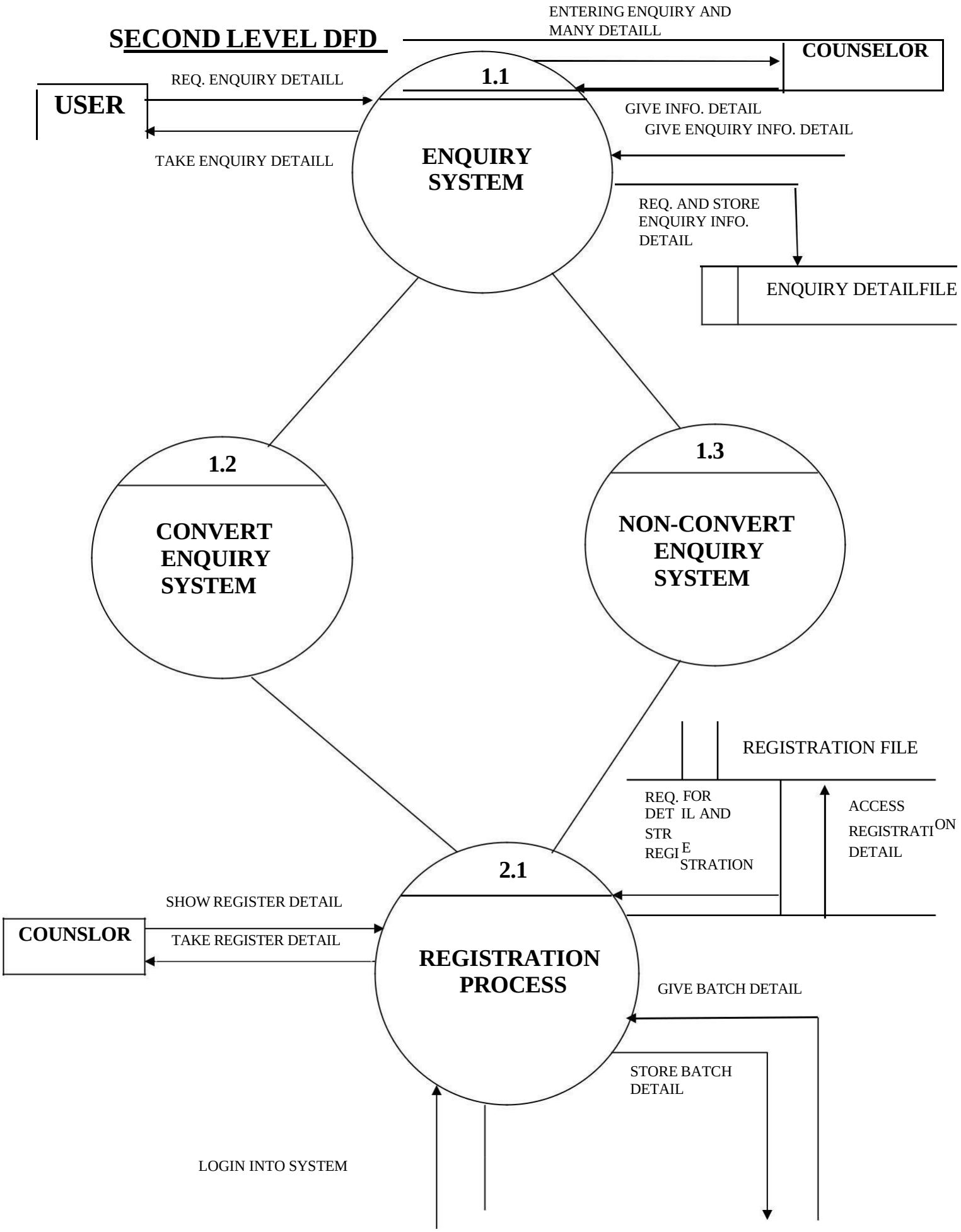


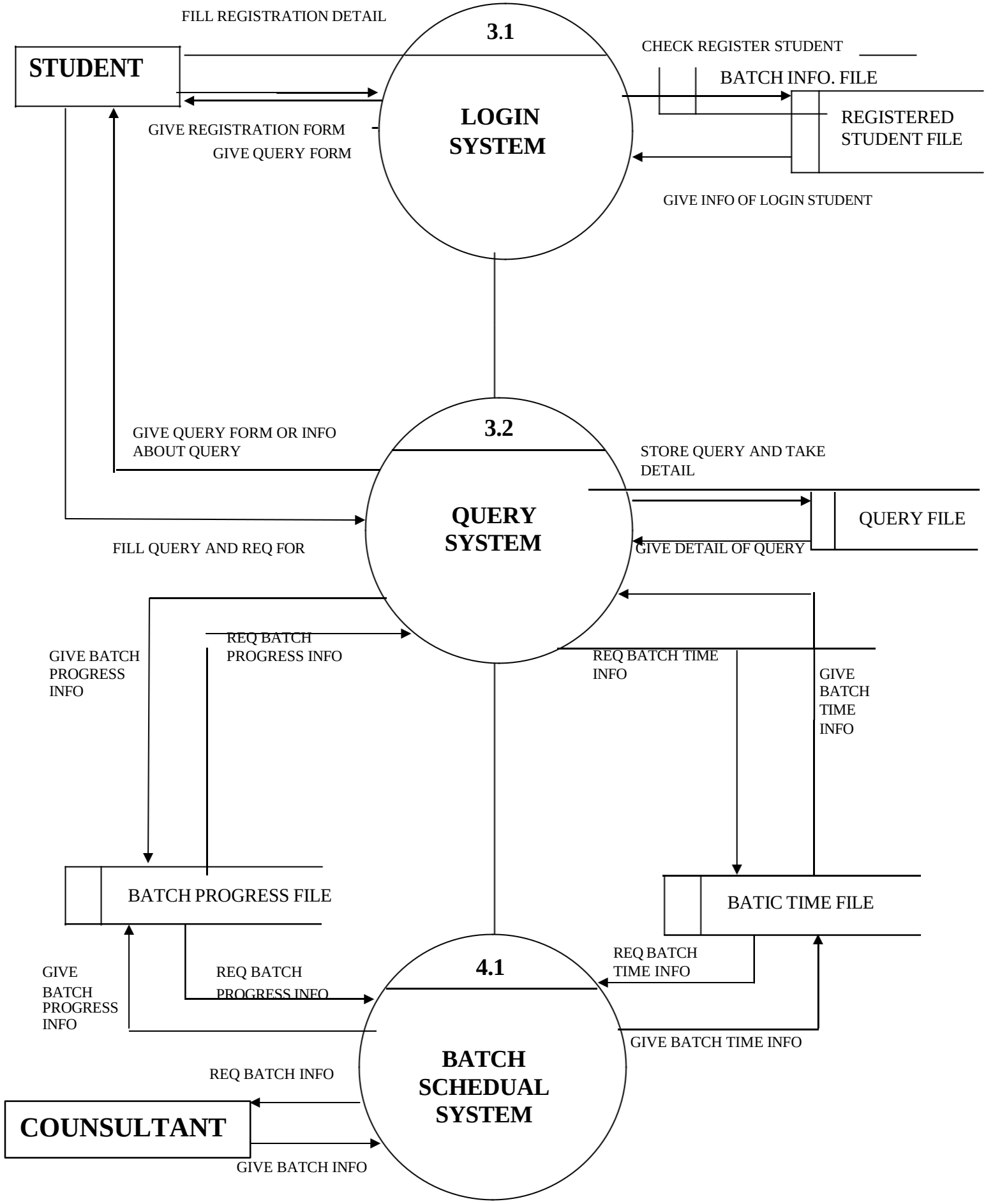

3.0
QUERY MANAGEMENT
REQ. FOR DETAIL (BATCH, SCHEDUAL)
VIEW REPORTS
TAKE QUERY
COUNSELOR
STUDENT
GIVE QUERY INFO. AND ENTER QUERY
PROVIDE DETAIL (BATCH, SCHEDUAL)
ENTER QUERY DETAIL OR STORE
PROVIDE QUERY DETAIL
MANAGER
QUERY DETAIL FILE
GIVE DETAIL OF QUERY
REQ. FOR QUERY DETAIL
COUNSULTANT
BATCH SCHEDUAL DETAIL
REQ. FOR BATCH SCHEDUAL
4.0
GIVE BATCH SCHEDUAL
UPDATE BATCH SCHEDUAL
BATCH SCHEDUAL
PROVIDE BATCH SCHEDUAL
STUDENT
REQ. FOR BATCH SCHEDUAL
ACCESS BATCH SCHEDUAL
BATCH DETAIL FILE
STORE BATCH SCHEDUAL

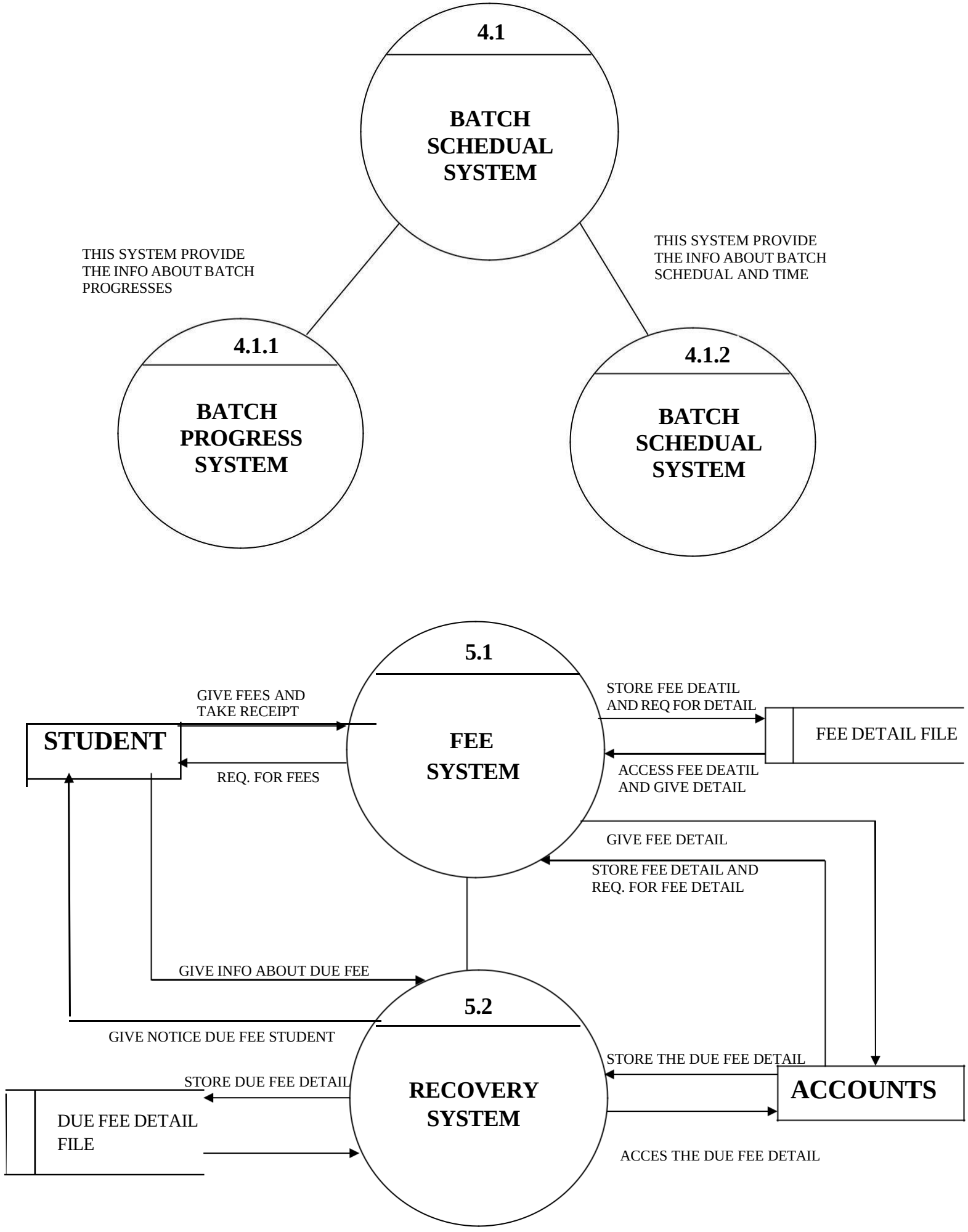

TAKE FEES AND FEE DETAIL
REQ. FOR FEE AND FINANCE DETAIL
PROVIDES FEES DETAIL
ACCOUNTS
MANAGMENT
5.0
FEE AND RECOVERY
PROVIDE INFO. ABOUT ACCOUNT AND FEES
GIVE DETAIL OF ACCOUNT
REQ FOR FEE TAKE RECEIPT
GIVE FEES
STORE ACCOUNT AND FEES
STUDENT
FEE DETAIL FILE
MARK ATTENDANCE
6.0
REQ. FORTAKE THE ATTENDANCE
ATTENDANCE MANAGEMENT
COUNSULTANT
PROVIDE DETAIL OF ATTENDANCE
TAKE ATTENDANCE
PROVIDE DETAIL OF TRANSFER, RESUME AND BREAKE
REQ. FOR BATCH DETAIL
7.0
PROVIDE THE INFO. OF BATCH
REQ. FOR INFO. OF BATCHES
STUDENT MANAGEMENT
STORE THE STUDENT ANT BATCH DETAIL
PROVIDE DETAIL OF STUDENT
STUDENT DETAIL FILE

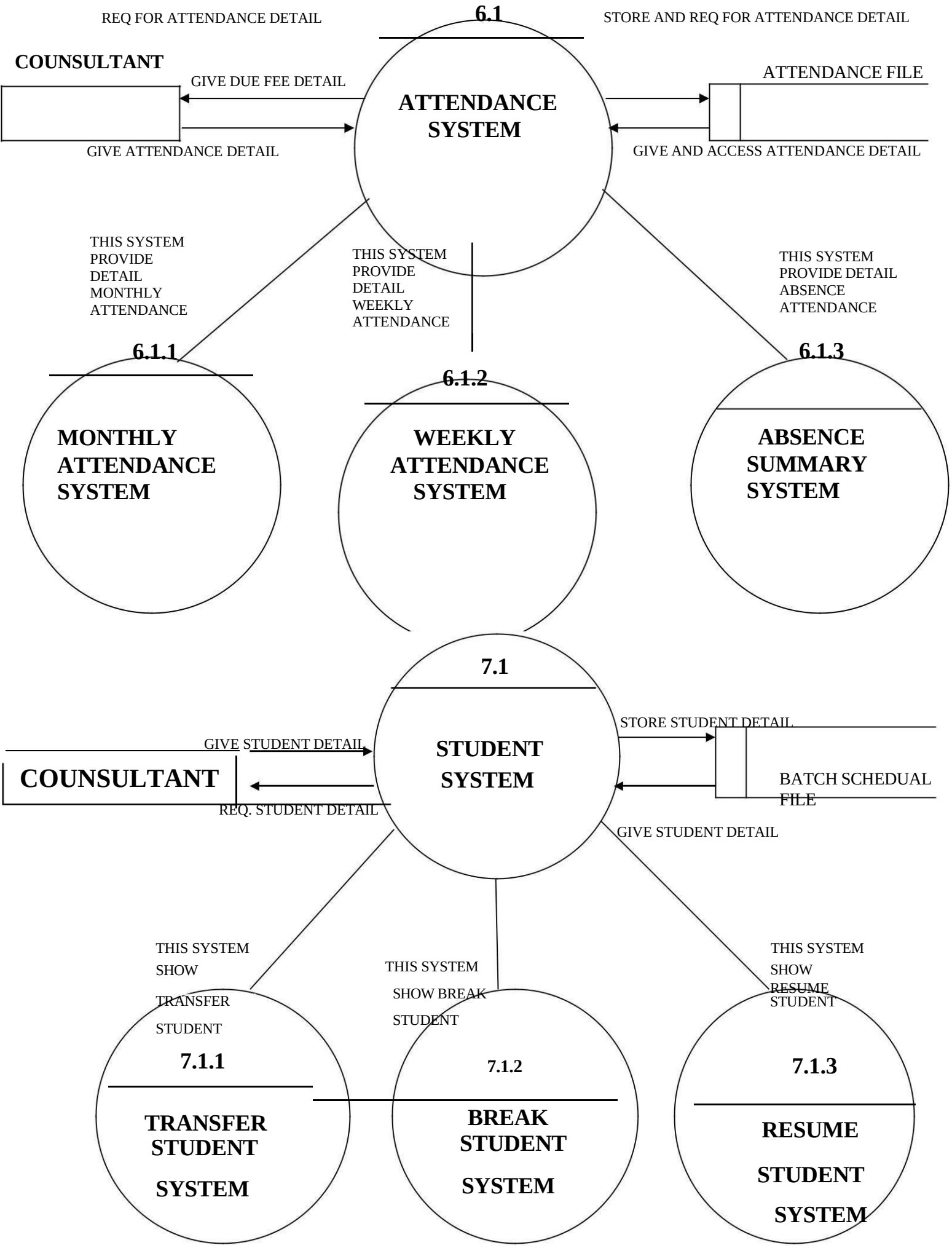

REQ. FOR
DISTRIBUTION
8.0
COUNSULTANT
REQ. FOR STUDY MATERIAL
STDUY
MATERIAL
MANAGEMENT
GIVE DISTRIBUTION DETAIL
STUDENT
GIVE STUDY MATERIAL
STORE INFO OF
DISTRIBUTION
ACCESS INFO OF
DISTRIBUTION
DISTRIBUTE DETAIL FILE
RESULT &
CERTIFICATE FILE
PROVIDE INFO.
OF RESULTAND
CERTIFICATE
GIVE TEST
9.0
STORE THE INFO RESULT AND CERTIFICATE
TAKING TEST
TAKING TEST FOR STUDENT
RESULT
AND
CERTIFICATE
REQ. FOR RESULT & CERTIFICATE
GIVE RESULT DETAIL
PROVIDE RESULT &
CERTIFICATE
MANAGER
COUNSULTANT

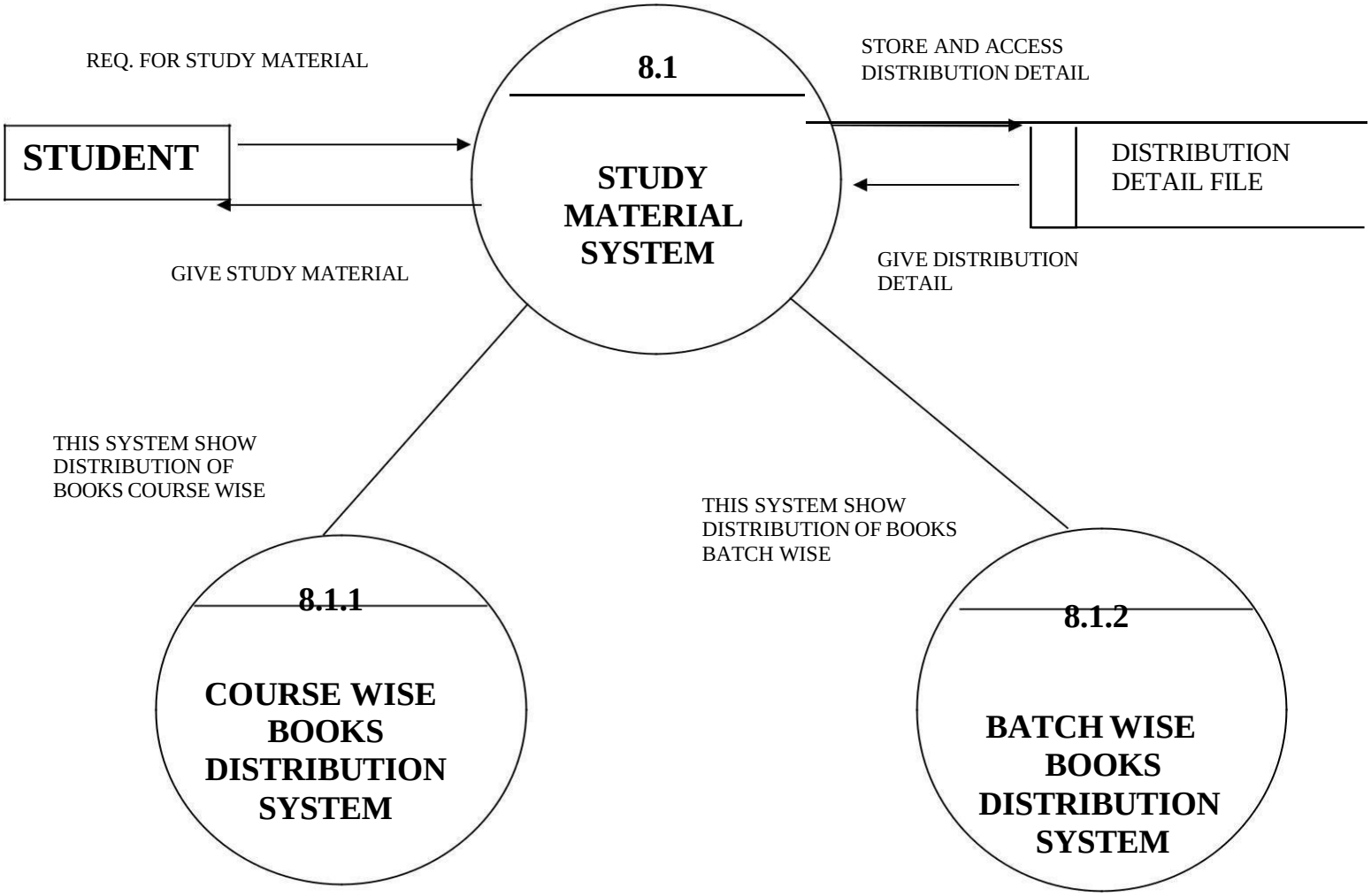
SECOND LEVEL DFD

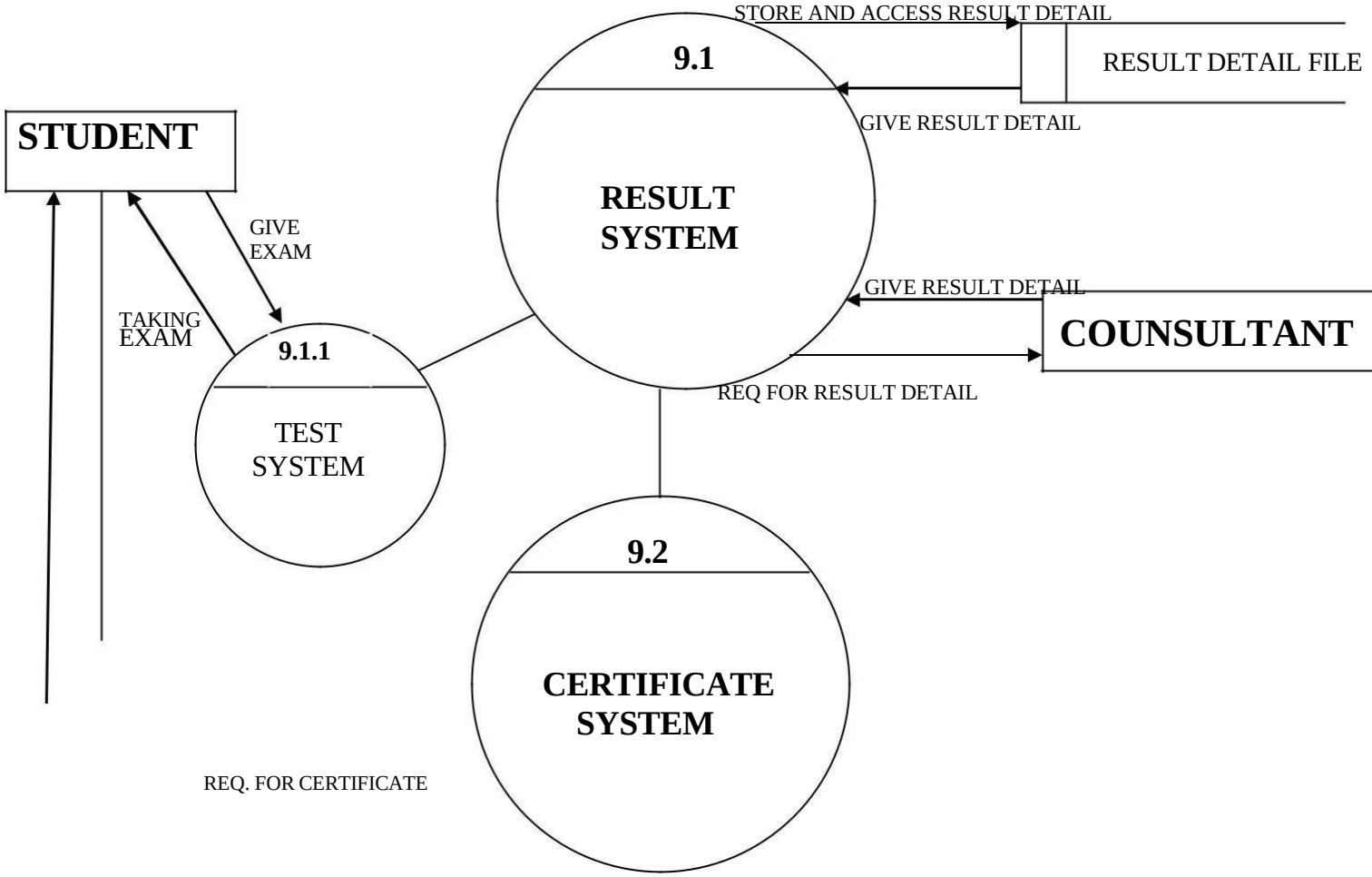



FILL REGISTRATION DETAIL
3.1
STUDENT
CHECK REGISTER STUDENT
BATCH INFO. FILE
LOGIN SYSTEM
REGISTERED STUDENT FILE
GIVE REGISTRATION FORM
GIVE QUERY FORM
GIVE INFO OF LOGIN STUDENT
GIVE QUERY FORM OR INFO ABOUT QUERY
3.2
STORE QUERY AND TAKE DETAIL
QUERY SYSTEM
QUERY FILE
FILL QUERY AND REQ FOR
GIVE DETAIL OF QUERY
GIVE BATCH PROGRESS INFO
REQ BATCH PROGRESS INFO
REQ BATCH TIME INFO
GIVE BATCH TIME INFO
BATCH PROGRESS FILE
BATIC TIME FILE
GIVE BATCH PROGRESS INFO
REQ BATCH PROGRESS INFO
4.1
REQ BATCH TIME INFO
GIVE BATCH TIME INFO
REQ BATCH INFO
BATCH SCHEDUAL SYSTEM
COUNSULTANT
GIVE BATCH INFO


4.1
BATCH SCHEDUAL SYSTEM
THIS SYSTEM PROVIDE THE INFO ABOUT BATCH PROGRESSES
THIS SYSTEM PROVIDE THE INFO ABOUT BATCH SCHEDUAL AND TIME
4.1.1
BATCH PROGRESS SYSTEM
4.1.2
BATCH SCHEDUAL SYSTEM
5.1
FEE SYSTEM
STUDENT
GIVE FEES AND TAKE RECEIPT
REQ. FOR FEES
STORE FEE DEATIL AND REQ FOR DETAIL
ACCESS FEE DEATIL AND GIVE DETAIL
FEE DETAIL FILE
GIVE FEE DETAIL
STORE FEE DETAIL AND REQ. FOR FEE DETAIL
GIVE INFO ABOUT DUE FEE
GIVE NOTICE DUE FEE STUDENT
5.2
RECOVERY SYSTEM
STORE THE DUE FEE DETAIL
ACCOUNTS
STORE DUE FEE DETAIL
DUE FEE DETAIL FILE
ACCES THE DUE FEE DETAIL

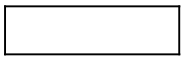

REQ FOR ATTENDANCE DETAIL
6.1
STORE AND REQ FOR ATTENDANCE DETAIL
COUNSULTANT
GIVE DUE FEE DETAIL
ATTENDANCE SYSTEM
ATTENDANCE FILE
GIVE ATTENDANCE DETAIL
GIVE AND ACCESS ATTENDANCE DETAIL
THIS SYSTEM PROVIDE DETAIL MONTHLY ATTENDANCE
THIS SYSTEM PROVIDE DETAIL WEEKLY ATTENDANCE
THIS SYSTEM PROVIDE DETAIL ABSENCE ATTENDANCE
6.1.1
6.1.2
6.1.3
MONTHLY ATTENDANCE SYSTEM
WEEKLY ATTENDANCE SYSTEM
ABSENCE SUMMARY SYSTEM
7.1
STORE STUDENT DETAIL
GIVE STUDENT DETAIL
STUDENT SYSTEM
COUNSULTANT
BATCH SCHEDUAL FILE
REQ. STUDENT DETAIL
GIVE STUDENT DETAIL
THIS SYSTEM SHOW TRANSFER STUDENT
THIS SYSTEM SHOW BREAK STUDENT
THIS SYSTEM SHOW RESUME STUDENT
7.1.1
7.1.2
7.1.3
TRANSFER STUDENT SYSTEM
BREAK STUDENT SYSTEM
RESUME STUDENT SYSTEM


REQ. FOR STUDY MATERIAL
8.1
STORE AND ACCESS DISTRIBUTION DETAIL
STUDENT
STUDY MATERIAL SYSTEM
DISTRIBUTION DETAIL FILE
GIVE STUDY MATERIAL
GIVE DISTRIBUTION DETAIL
THIS SYSTEM SHOW DISTRIBUTION OF BOOKS COURSE WISE
THIS SYSTEM SHOW DISTRIBUTION OF BOOKS BATCH WISE
8.1.1
COURSE WISE BOOKS DISTRIBUTION SYSTEM
8.1.2
BATCH WISE BOOKS DISTRIBUTION SYSTEM

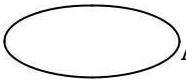

STORE AND ACCESS RESULT DETAIL
9.1
RESULT DETAIL FILE
GIVE RESULT DETAIL
STUDENT
RESULT SYSTEM
GIVE EXAM
TAKING EXAM
GIVE RESULT DETAIL
COUNSULTANT
9.1.1
TEST SYSTEM
REQ FOR RESULT DETAIL
9.2
CERTIFICATE SYSTEM
REQ. FOR CERTIFICATE

ENTITY RELATIONSHIP DIAGRAM

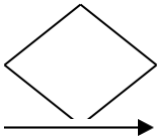
E-R Diagram is a way to represent conceptual database tools. The overall logical structure of a database can be expressed graphically by an E-R diagram, which consists of the following components :



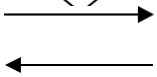
Entity



Attribute

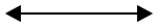


Relationship



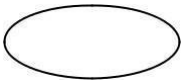
M : 1

1 : M

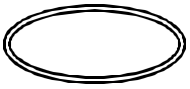


1 : 1

M : M



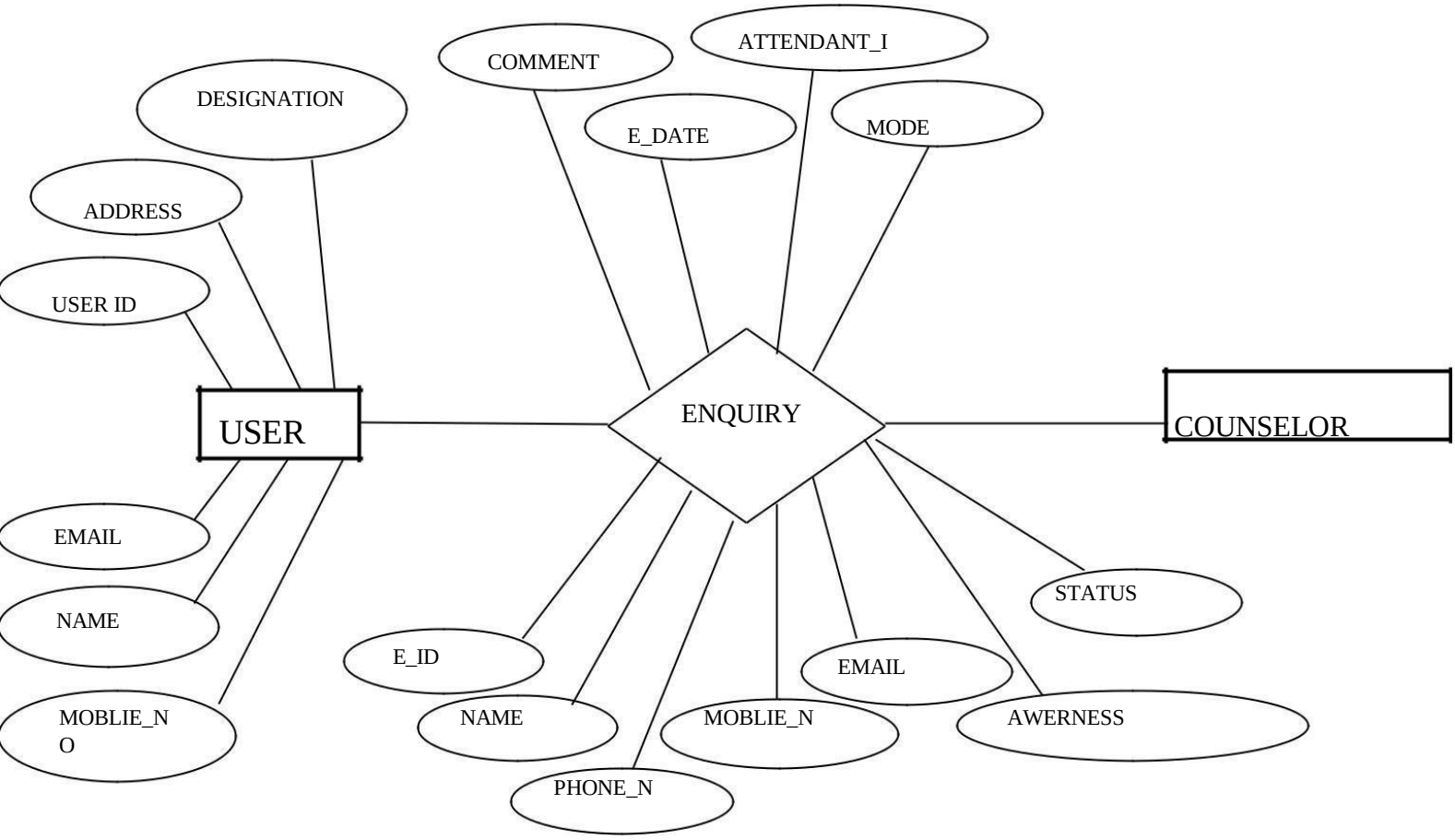
Derived Attribute

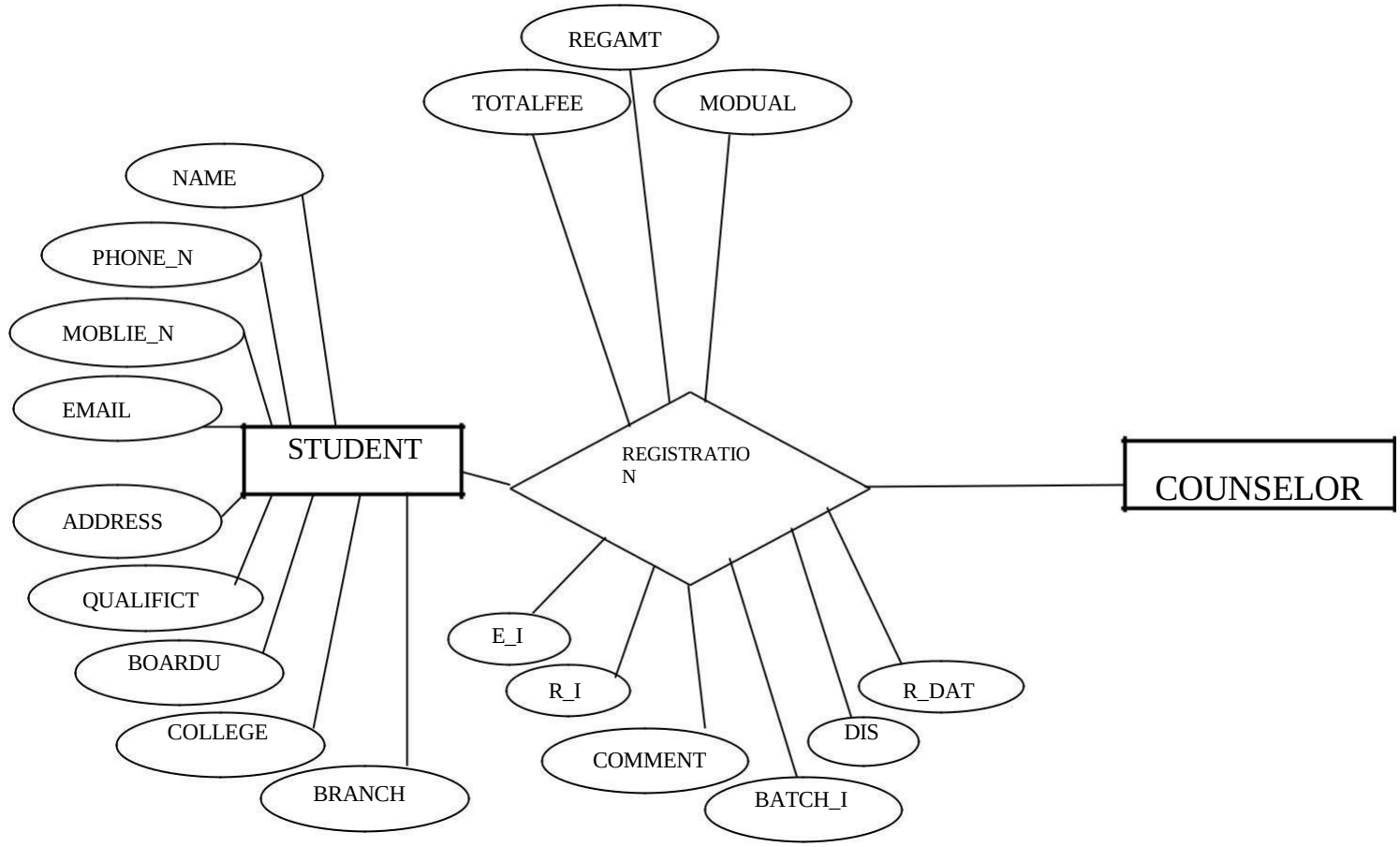


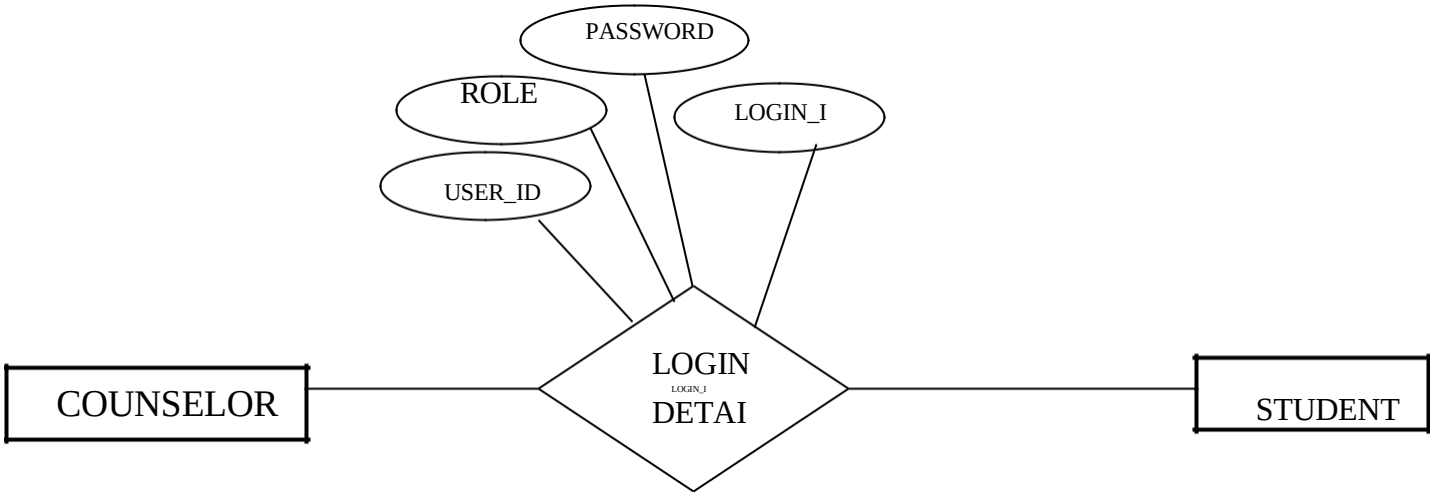
Multi-valued Attribute

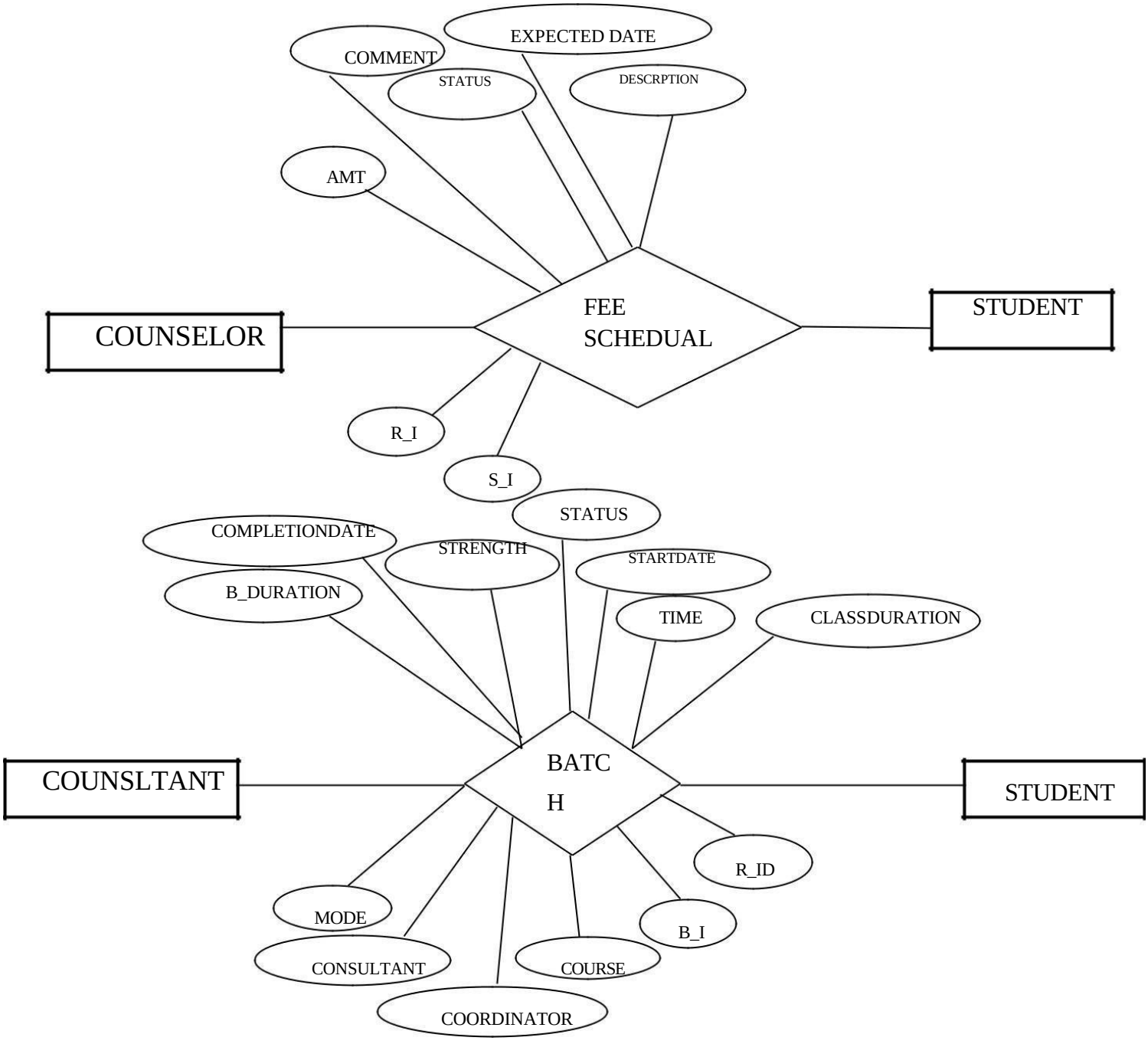
Weak Entity

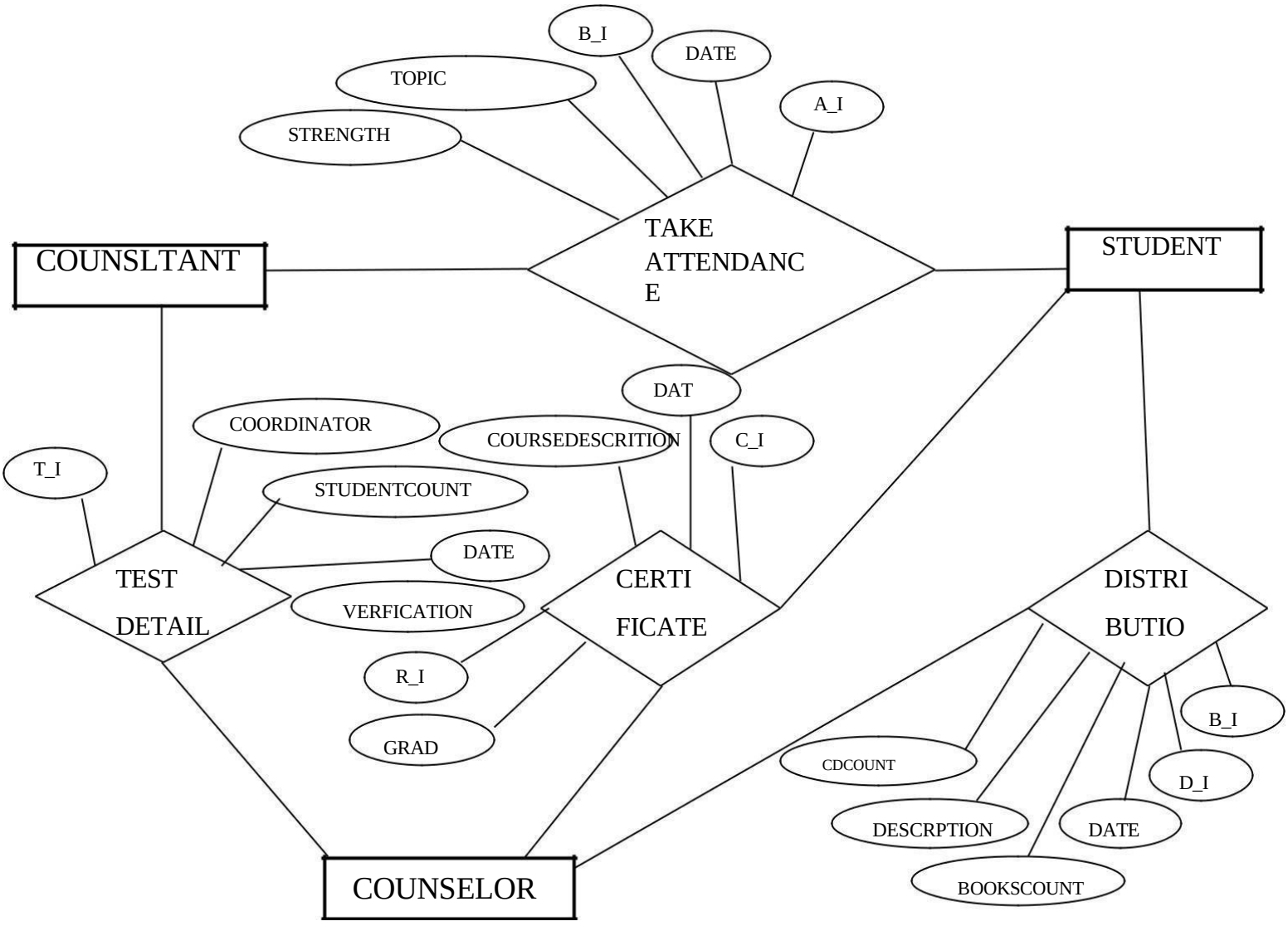
Weak Relationship


COMMENT
ATTENDANT_I
DESIGNATION
E_DATE
MODE
ADDRESS
USER ID
USER
ENQUIRY
COUNSELOR
EMAIL
NAME
MOBLIE_N
O
E_ID
NAME
MOBLIE_N
PHONE_N
EMAIL
STATUS
AWERNESS


REGAMT
TOTALFEE
MODUAL
NAME
PHONE_N
MOBLIE_N
EMAIL
STUDENT
REGISTRATIO
N
COUNSELOR
ADDRESS
QUALIFICT
BOARDU
COLLEGE
BRANCH
E_I
R_I
COMMENT
BATCH_I
DIS
R_DAT


PASSWORD
ROLE
LOGIN_I
USER_ID
LOGIN
LOGIN_I
DETAI
COUNSELOR
STUDENT


EXPECTED DATE
COMMENT
STATUS
DESCRPTION
AMT
FEE
SCHEDUAL
COUNSELOR
STUDENT
R_I
S_I
STATUS
COMPLETIONDATE
STRENGTH
STARTDATE
B_DURATION
TIME
CLASSDURATION
BATC
H
COUNSLTANT
STUDENT
R_ID
MODE
B_I
CONSULTANT
COURSE
COORDINATOR


B_I
DATE
TOPIC
A_I
STRENGTH
TAKE ATTENDANCE
COUNSLTANT
STUDENT
DAT
COORDINATOR
COURSEDESCRITION
C_I
T_I
STUDENTCOUNT
DATE
TEST DETAIL
CERTI FICATE
DISTRI BUTIO
VERFICATION
R_I
GRAD
B_I
CDCOUNT
D_I
DESCRPTION
DATE
BOOKSCOUNT
COUNSELOR

SRS

SOFTWARE REQUIREMENT SPECIFICATIONS

Software requirement specification (SRS) is the starting point of the software development activity. The SRS is the means of translating the ideas in the minds of the client (the input), into a formal document (the output of the requirement phase). Thus, the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input have none of these properties. Clearly the process of specifying requirements cannot be formal.

Any formal translation process producing a formal output must have a precise and unambiguous input. A procedure for identifying requirements can therefore be at best a set of guidelines.

The requirement specification phase consists of two basic activities: - Problem or requirement analysis, and requirement specification. The first aspects, perhaps the harder and more nebulous of the two, deals with understanding the problem, the goals, and constraints. In the second, the focus is on clearly specifying what has been found during analysis. Issues such as representation, specification languages and tools and checking the specification are addressed during this activity. The requirements phase terminates with the production of the validated software requirement specification document. Producing the SRS is the basic goal of this phase.

- ***Role of SRS:***

The origin of most software systems is in the need of a client who either wants to automate an existing manual system or desires a new software system. The Developer creates the software system itself. Finally, the end users will use the complete system. Thus there are three major parties interested in the new software s*ystem the client, the user and the developer*. The problem is that the client usually does not understand the software or the software development process and the developer often does not understand the client's problem and application area.

This cause the communication gap between the parties involved in the development project.

The purpose of software requirement specification, SRS, is to bridge this communication gap.

SRS is the medium through which the client and the user needs are accurately specified: indeed, SRS forms the basis of software development. A good SRS should specify ass the parties.

A good SRS provides many benefits:

Some of the goals it accomplishes are:-

Establishes the basis for agreement between client and software developer on what the software product will do.

Reducing the development cost.

Providing a reference for validation of the final product. The SRS assists the client in determining if the software meets the requirements.

Overview of a software requirement.

Purpose: -

The system has been developed on client server architecture. The Server has a database SQL Server 7.0 and the client has Visual Basic 6.0. Providing connectivity through ADODC (ADO data control) object makes the communication between these two. They communicate via a middle ware component called ODBC.

PROBLEM ANALYSIS: -

The first of the two basic activities performed during the requirement phase is analyzing the problem. Problem analysis is done to obtain a clear understanding of the needs of the client and the users and what exactly is the desired from the software. Analysis leads to the actual specification. Analysis involves interviewing the clients and the users. Typically, analysts research a problem by asking questions from clients and users, and by reading existing documents.

REQUIREMENT SPECIFICATIONS: -

Once the analysis is complete, the requirements must be written or specified. The final, output is the Software Requirement Specifications documents (SRS). For smaller problems or the problems that can easily be comprehended , the specification activity might come after the entire analysis is complete. However it is more likely that problem analysis and specification are done concurrently. And analyst typically will analyze some parts of the problem and then writes the requirements for that part.

CHARACTERISTICS OF AN SRS: -

To properly satisfy the basic goals, an SRS should have certain properties and should contain different types of requirements. In this section we should discuss some of the desirable characteristics of an SRS, and different components of and SRS. A good SRS is:-

- Understandable.
- Unambiguous.
- Complete.
- Verifiable.
- Consistent.
- Modifiable.
- Traceable.

Clearly an SRS should be understandable, as one of the goals of the requirement phase is to produce a document upon which a client, the user and the developer can agree. Since multiple parties need to understand and approve the SRS, it is most utmost importance that the SRS should be understandable.

And SRS is complete if everything the software is supposed to do is in the SRS. A complete SRS defines the responses of the software to all classes of input data. For specifying al thee requirements, the requirements relating to functionality, performance, design constraints, attributes and external interfaces must be specified. In addition, the responses to both valid and invalid in input values must also be specified.

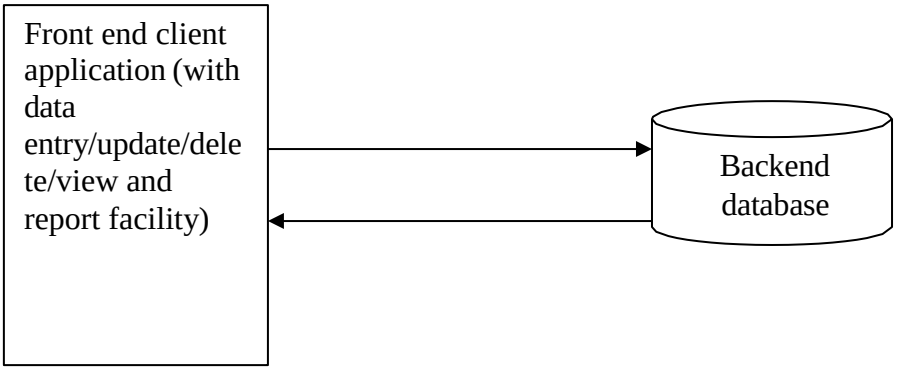
A requirement is verifiable if there exists some cost effective process that can check if the final software meets that requirement. An SRS is verifiable if and only if every stated requirement is verifiable.

Writing an SRS is an iterative process. Even when the requirements of a system are specified, they are later modified as the need of thee client change with time. Hence an SRS should be easy to modify. An SRS is modifiable if its structure and style is such that any necessary change can be made easily, while preserving the completeness and consistency.

An SRS is traceable if the origin of each of its requirement is clear and if it facilitates the reference of each requirement in future development. Forward trace ability means that each requirement should be traceable to some design and code elements. Backward trace ability requirement should be possible to trace design and code element to requirement they support. Trace ability adds verification and validation.

PRODUCT PERSPECTIVE

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



(A) User interface:-

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

(i) A login screen for entering the Login id and password will be provided to Access main project.

(ii) There will be a screen for add to profile of users.

(iii) There will be a screen for change password for login.

(iv) There will be a screen for add and displaying information regarding to course management.

(v) There will be a screen for add and displaying information regarding to user management.

(vi) There will be a screen for add and displaying the information regarding to technology management.

(B) Hardware interface:

It is recommended that the minimum configuration for clients is as appended beiow.Suggested Configuration of Windows clients:-

Microprocessor : - Pentium-4 class processor, 450 (MHz)

Ram : - 512 MB of RAM

CD ROM Drive : - 52 X CD ROM Drive

Hard Disk : - 40 Gigabytes (GB) on installation drive,

(C) Software interface:

- **Windows 98 / XP operating system**
- **MYSQL**
- **JDK 1.6**
- **Weblogic 10.3**

(1) Specific requirements:

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

Login screen

This will be first screen that will be displayed .it will allow user to access the main form .After login he will be displayed the main page where he can go though different menus various fields available on this screen will be:

(a) User ID:alphanumeric of length up to 20 characters.

(b)Password:alphanumeric of length up to 20 characters.

(A)Main form

(i) profile: This will be allowing the user to edit the profile stored in database. information about users various fields available on his screen will be:

(a) name: Alphanumeric letters and length up to 20 characters.

(b) address*:* Alphanumeric letters and length up to 30 characters.

(c) Mail id: Alphanumeric letters and length up to 25 characters.

(d) Mobile no: number letters is up to 12.

(e) Change password: Alphanumeric letters and length up to 10 characters.

(ii) Technology management:this will allow the user to add new/update/delete technology and description about the technology record. various fields available on these screen will be:

(a) Name: alphanumeric, of length up to 30 characters.

(b) description: alphanumeric, of length up to 100 characters.

(c) Technology id: number letters is up to 10.

(iii) Course management: This will be allowing the user to add/update/delete information about the course. various fields available on his screen will be:

(a) name:alphanumeric, of length up to 40 characters

(b) description*:* alphanumeric, of length up to 80 characters

(c) fee*: number letters,* of length up to 10 characters.

(d) technology: alphanumeric, of length up to 10 characters**.**

(iv) User management: this will allow the users to add new/update/delete the information about personal details. Various fields available on these screens will be:

(a) name: Alphanumeric, of length upto 20 characters.

(b) description: Alphanumeric, of length upto 100 characters.

(c) Role : Alphanumeric, of length upto 20 characters.

(d) address: Alphanumeric, of length upto 30 characters.

(e) Mail id: Alphanumeric, of length upto 20 characters.

(f) Mobile no: Numeric letters up to 12.

***(2)* Logout:** this will allow the users to logout and reach on E-FACILITATOR login foam.

(3) Admin login Table:this will allow the administrator/user to add new/update/delete the record. After entering the correct user id or password administrator will delete the records.

various fields available on these screen will be:

(a) Administrator_Id: alphabetic, of length up to 20characters.

(b) Administraator password:alphabetic, of length up to 20characters.

(D) Software product features:

Prompt users:

Description:

The system will maintain information about us*ers*. The following information would be maintained for each us*er*: userid, name, mobileno, address, mailed, designation, loginid, password, role, technology.

❖ Validity check

(a) userid: should not be greater than 10numeric and cannot be null.

(b) mobileno: should not be greater than 12 numeric and cannot be null.

(c) Name: should not be less than 20 alphanumeric and cannot be null.

(d) Address: should not be less than 30 alphanumeric and cannot be null.

(e) mailid: should not be less than 20 alphanumeric and cannot be null.

(f) Designation: should not be less than 25 alphanumeric and cannot be null.

(g) Loginid: should not be less than 8 numeric and can not be null.

(h) Password: should not be less than 10 alphanumeric and cannot be null.

(i) Role: should not be less than 20 alphanumeric and cannot be null.

(j) Technology id: should not be less than 5 numeric and cannot be null.

❖ Error handing/response to abnormal situation

Error handling is applied to the save submenu to handle the error at run time. If any error is occurred during save operation then it will handle and save operation will be terminated and project will continued.

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

- **Technology:**

Description

The system will maintain information about technology details. The following information would be maintained for each user: technologyid, name, description.

Validity checks

(a) Technologyid:will be unique for every technology and cannot be blank and should be less than 10 numeric.

(b) Name:cannot be null and should be less than 30 alphanumeric.

(c)Description: cannot be null and should be less than 80 alphanumeri

Error handing/response to abnormal situation

Error handling is applied to the save submenu to handle the error at run time. If any error is occurred during save operation then it will handle and save operation will be terminated and project will continued.

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

- **AttendedEnquiry:**

Description

The system will maintain information about attendedenquiry. The following information would be maintained for each student: attending id, enquiryid, attendantid, attendingdate, prospect, comments.

Validity check

(a) Attendingid: should not be greater than 10 numeric and cannot be null.

(b) Enquiryid: should not be greater than 10 numeric and cannot be null.

(c) Attandantid: should not be greater than 10 numeric and cannot be null.

(d) Attandingdate: cannot be null and should be less than 20 alphanumeric.

(e) Prospect: should not be greater than 10 numeric and cannot be null.

(f) Comments: cannot be null and should be less than 100 alphanumeric.

Error handing/response to abnormal situation

Error handling is applied to the save submenu to handle the error at run time. If any error is occurred during save operation then it will handle and save operation will be terminated and project will continued.

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

- **Login:**

Description

The system will maintain information about user. The following information would be maintained for each user: login id,password.

The system will be allowing modification to password.

Validity check

(a) Loginid:should not be less than 20 alphanumeric & not null.

(b)Password: Password should not be less than 20 alphanumeric & not null.

Error handing/response to abnormal situation

Error handling is applied to the save submenu to handle the error at run time. If any error is occurred during save operation then it will handle and save operation will be terminated and project will continued. If any of the above validation does not hold true, appropriate error messages will be prompted to the user for doing the needful.

- **Enquiry:**

Description

The system will maintain information about Enquiry of user detail and course The following information would be maintained for each user: enquiryid, name,mailed, mobileno.

Validity check

- ***(a)*** **Enquiryid:**should not be greater than 10 numeric and must be unique for each user.
- ***(b)*** **name:** Password should not be less than 25 alphanumeric & not null.
- ***(c)*** **mailid:** Password should not be less than 25 alphanumeric & not null.
- ***(d)*** **mobileno:** should not be greater than 12numeric and should not be null.
- ***(e)*** **Enquiry date:** should not be less than 20 alphanumeric & not null.
- ***(f)*** **Awereness source:** should not be greater than 10 numeric.
- ***(g)*** **Course id:** should not be greater than 10 numeric.
- ***(h)*** **Enquiry mode:** should not be greater than 10 numeric.
- ***(i)*** **Status:** should not be greater than 10 numeric.

Error handing/response to abnormal situation

Error handling is applied to the save submenu to handle the error at run time. If any error is occurred during save operation then it will handle and save operation will be terminated and project will continued.

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

(E) Software system Attributes:

(i) Security:

The application will be password protected .users will have to enter correct username, and password in lower case in order to access the application.

(ii) Maintainability:

The application will be designed in a manner. It will be easy to incorporate new requirement the individual modules (i.e.Administrator /User Login, profile, technology management, course management, user management, logout details.)

(iii) Portability:

The application will be portable on any window –based system that has mysql.

(F) Logical Database Requirement:

The following information will be placed in a database:

(i)Technology:TECHNOLOGYID,NAME,DESCRIPTION.

(ii)COURSE:OURSEID,TECHNOLOGYID,NUMBER,NAME,DESCRIPTION,FEE NUMBER.

(iii) Login: loginid,password.

(iv)promptUsers:user_id,name,Address,mobileNo,mailid,Designation,loginId,Password,Rol technologyId.

(iv) ENQUIRY:NQUIRYID,NAME,MAILID,MOBILENO,Enquiry date,awereness source, course id, enquiry mode,status.

(v)ATTENDEDENQUIRY:ATTENDINGID,ENQUIRYID,ATTENDANTID,ATTENDIN GDATE,PROSPECT,COMMENTS.

-

SYSTEM DESIGN

MODULES AND THEIR DESCRIPTION

- **Login:** This module contain the detail about users login id and passwordj.
- **Technology-** This includes the details about the technology which is provide by the institute to the student in IT field.
- **Course** - This includes the complete details about the course provideBy institutute.
- **Promptusers –** This includes the complete details about the users
  Or student Incuding role and technology.
- **Enquiry-** This module includes the details about enquiry for various courses offered by the institution..
- **Attendedenquiry-**This module provides an interface to mark attendance of students who have enrolled for technology.

DATA DICTIONARY:

It is a collection of elements that have been used in the database. It is actually "data about the data". The database is the actual place that stores, retrieves and manipulates data through the front-end tool. Our application initially has only 5 tables and the end-user has freedom to create their own tables which will have the same structure as the tables defined already. The following are the table definitions with their data types.

DATA STRUCTURE

Technology

Field Name	Data Type	Size	Constraint	Comment
Technology id	Number	(10,0)	Primary key	This field are store of the technology id
Name	Varchar2	30		This field are store of the technology name
Description	Varchar2	80		This field are store of the technology description

ATTENDEDENQUIRY				
Field Name	**Data Type**	**Size**	**Constraint**	**Comment**
Attendingid	Number	(10,0)	Primary key	This ID are store of the attandaning id
Enquiryid	Number	(10,0)		This field are store of the Enquiryid
Attendantid	Number	(10,0)		This field are store of the Attandantid
Attendingdate	Varchar2	20		This field are store of the attending date
Comments	Varchar2	100		This field are store of the Comment

ENQUIRY				
Field Name	**Data Type**	**Size**	**Constraint**	**Comment**
EnquiryId	Number	(10,0)	Primary key	This field are store of the equiry id of the user
Name	Varchar2	25		This field are store of the user Name
Mailid	Varchar2	25		This field are store of the mail id of the user
MobileNo	Varchar2	12		This field are store of the mobile no of user
Enquiry date	Varchar2	20		This field are store of the enquiry date of user.
Awereness source	Number	(10,0)		This field are store of the awereness source
Course id	Number	(10,0)	Foreingn key	This field are store of the course id
Enquiry mode	Number	(10,0)		This field are store of the enquiry mode
Status	Number	(10,0)		This field are store of the status.

Login				
Field Name	**Data Type**	**Size**	**Constraint**	**Comment**
User_id	Varchar2	20	Primary key	This ID are store of User ID
Password	Varchar2	20	Not null	This Password are store of the User Password

Course				
Field Name	**Data Type**	**Size**	**Constraint**	**Comment**
Course id	Number	(10,o)	Primary key	This ID are store of course id.
Technology id	Number	10	Foreign key	This ID are store of the technology id.
Name	Varchar2	40		This field are store of the technology name.
Description	Varchar2	80		This field are store of technology description
Fee	Number	(10,0)		This field are store of the technology fee.

Prompusers				
Field Name	**Data Type**	**Size**	**Constraint**	**Comment**
UserId	Number	(10,0)	Primary key	This field are store of the user id.
Name	Varchar2	20		This field are store of the user name.
Address	Varchar2	30		This field are store of the user address.
MobileNo	Varchar2	10		This field are store of the user mobile no.
Mailid	Varchar2	20		This field are store of tie user mail id.
Designation	Varchar2	20	Foreign key	This field are store of the user designation.
Loginid	Number	20		This field are store of the user login id.
Password	Varchar2	10		This field are store of the password.
Role	Varchar2	20		This field are store of the role.
Technology id	Number	(10,0)		This field are store of the technology id.

,

❖ **USER INTERFACE:-**

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

(i) A login screen for entering the Login id and password will be provided to Access main project.

(ii) There will be a screen for add to profile of users.

(iii) There will be a screen for change password for login.

(iv) There will be a screen for add and displaying information regarding to course management.

(v) There will be a screen for add and displaying information regarding to user management.

(vi) There will be a screen for add and displaying the information regarding to technology management.

❖ LIST OF REPORTS TO BE GENERATED

Inputs to the Projects are as follows:

- Login page get the input of user id and password.
- Create the user id for register yourself.
- Change password
- Course edit
- Technology edit
- User edit

Output(s) of the Project are as follows:

- View student details.
- Home
- Technology details
- Course detail

❖ **TEST CASE**

Validation check:

✓ **USER LOGIN**

Session 1

➢ **Before execution**

(a) Purpose: to check authorized person who can access the data.

(b) Pre-condition: None

(c) Input:login_id (a) AND password (a)

(d) Executed output: login is successful and opens the next page.

(e) Post-condition (if any): if login is successful, the open the next page and close the login form.

f) written by: Pooja

g) date: 15-09-2023

➢ **After execution**

(a) Execution history: login failed

(b) Result: invalid user_id or password.

(c) If fail, any possible reason(if any): wrong password or login_id

(d) Any other observation: case sensitive

(e) Any suggestion: change the input case

f) run by: Pooja

g) date: 15-09-2023

- ✓ **ADMINISTRATOR_LOGIN**

Session 2

- ➢ **Before execution**

a) **purpose:** to check authorized person

b) **pre-condition:** none

c) **input:**login_id AND password

d) **expected output:** login is successful and open the next page

e) **post-condition:** if login is successful then open next page

f) **written by: Pooja**

g) **date:** 15-09-2023

- ➢ **After execution**

a) **execution history:** login successfull

b) **result:** login is successful

c) **if fails, any possible reason(if any):** invalid user id or password

d) **Any observation:** none

e) **Any suggestion:** none

f) **Run by: Pooja**

g) **date:** 15-09-2023

- ✓ **PROFILE**

Session 1

- ➢ **Before execution**

a) **Purpose:** save the record.

b) **Pre-condition(if any):** none

c) **Input:** name, mail id, mobile no, address etc.

d) **Executed output:** save the new entry and stored in database.

e) **Post condition(if any):** none

f) **Written by: Pooja**

g) **date:** 15-09-2023

- **After execution**

a) Execution history: ODBC error generates.

b) Result: integrity constraint violated

c) if fail, any reasons(if any): edit profile will be blank.

d) Any observation: none

e) Any suggestion: none

f) Run by: Pooja

g) date: 15-09-2023

Session 2

- **Before execution**

a) Purpose: save the record.

b) pre-condition: none

input:

Name of the Field
Address
Name
Mail id
Mobile no

c) Executed output: save the record and save in the database.

d) post-condition: none

e) written by: Pooja

f) date: 15-09-2023

- **After execution:**

a) Execution history: successfully stored the record.

b) result: record is save

c) if fail , any reason (if any): no

d) any reason: no

e) **any suggestion:** no

f) **run by: Pooja**

g) **date:** 15-09-2023

- ✓ **Technology Management**

Session 1

- **After execution**

a) **Purpose:** save the record.

b) **Pre-condition(if any):** none

Input:

Field name
Name
Description

c) **Executed output:** save the new entry and stored in database.

d) **Post condition(if any):** none

e) **Written by: Pooja**

f) **date:** 15-09-2023

- **After execution**

a) **Execution history:** ODBC error generates.

b) **Result:** record does not save

c) **if fail, any reasons(if any):** name and description will be left blank

d) **Any observation:** none

e) **Any suggestion:** none

f) **Run by: Pooja**

g) **date:** 15-09-2023

CODING

PROGRAM CODE

```
<%@ page import="prompt.helper.RequestMapper,prompt.beans.Login"
buffer="16kb"%>

<%! String resource;
   RequestMapper helper;
%>

<HTML><HEAD><TITLE>E-FACILITATOR</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-
1252">
<LINK href="css/leftmenu.css" type=text/css rel=stylesheet>
<LINK href="css/style.css" type=text/css rel=stylesheet>
<LINK href="css/topmenu.css" type=text/css rel=stylesheet>
<LINK href="css/ajax-tooltip.css" type=text/css rel=stylesheet>

<SCRIPT language=javascript src="js/date.js" type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/mainPublic.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/index.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/leftmenu.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/datetimepicker.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/inputfix.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/shortcut.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/popcalendar.js"></SCRIPT>
<SCRIPT language=javascript type=text/javascript>

        function showHideMenu()
        {
                var menu = document.getElementById('divleftmenu');
                var img = document.getElementById('imgShowHide');
                var tdleft = document.getElementById('tdleftmenu'); ;

                var cookie = getCookie("leftmenu");
                if(cookie == 'none')
                {
```

```
                img.src = "images/menu_hide.jpg";
                tdleft.style.width = 150;
                menu.style.display = 'block';
                document.cookie = "leftmenu=block";
        }
        else
        {
                img.src = "images/menu_show.jpg";
                tdleft.style.width = 0;
                menu.style.display = 'none';
                document.cookie = "leftmenu=none";
        }
}
        var getHeight = screen.height - 252;

</SCRIPT>
<style type="text/css">
.options{display: none;}
</style>
<script language="javascript" type="text/javascript"
src="js/datetimepicker.js"></script>
<script language="javascript" type="text/javascript"
src="js/inputfix.js"></script>
<script language="javascript" type="text/javascript"
src="js/shortcut.js"></script>

<link rel="stylesheet" type="text/css" href="css/sample.css"><!--<script
type="text/javascript" src="includes/jscript/popup-window.js"></script>-->

<link rel="stylesheet" href="css/demo.htm" type="text/css">

<link rel="stylesheet" href="css/modal-message.css" type="text/css">

<link rel="stylesheet" href="css/scroll.css" type="text/css">
<link rel="stylesheet" href="includes/css/autosuggest_inquisitor.css"
type="text/css" media="screen" charset="utf-8" />

<style type="text/css">
input.b {
   width: 300px;
}
</style></head><body>
<div id="dhtmltooltip"></div>
```

```
        <table class="table_main" cellspacing="0">
        <tbody><tr>
        <td>

<%
                        if (helper==null)
                        {
                        helper=new RequestMapper();
                        String path=application.getRealPath("WEB-
INF/classes/prompt/resources/requestmap.properties");
        System.out.println(path);
        helper.load(path);

                        }
                        String key=request.getParameter("p");
                        resource=helper.getResource(key);

                        Login login=(Login) session.getAttribute("login");
%>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
<tbody>
<%
   String welcomeNote="",headerPage="";
   if(login!=null)
   {
   welcomeNote="welcome, "+login.getName()+"   you are logged in as
"+login.getRole();
        request.setAttribute("welcomeNote",welcomeNote);
        headerPage=helper.getResource(login.getRole());

  %>
<tr>
<td colspan="2">
<jsp:include page="<%=headerPage%>"/>
        </td></tr>
        <%
        }
        %>

        <tr>
        <td id="tdMain" style="padding-top: 5px;" valign="top" width="100%">

<jsp:include page="<%=resource%>"/>
```

```
                    </td>
                    </tr>
                </table>
        </td>
        </tr>
        <tr>

        </tr>
        </table>


</body></html>
```

```
<HTML><HEAD><TITLE>PromptBiz</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-
1252">
<LINK href="css/leftmenu.css" type=text/css rel=stylesheet>
<LINK href="css/style.css" type=text/css rel=stylesheet>

<LINK href="css/topmenu.css" type=text/css rel=stylesheet>
<LINK href="css/ajax-tooltip.css" type=text/css rel=stylesheet>

<SCRIPT language=javascript src="js/date.js" type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/mainPublic.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/index.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/leftmenu.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/datetimepicker.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/inputfix.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/shortcut.js"
type=text/javascript></SCRIPT>
<SCRIPT language=javascript src="js/popcalendar.js"></SCRIPT>
<SCRIPT language=javascript type=text/javascript>

        function showHideMenu()
        {
                var menu = document.getElementById('divleftmenu');
                var img = document.getElementById('imgShowHide');
                var tdleft = document.getElementById('tdleftmenu'); ;

                var cookie = getCookie("leftmenu");
                if(cookie == 'none')
                {
                        img.src = "images/menu_hide.jpg";
                        tdleft.style.width = 150;
                        menu.style.display = 'block';
                        document.cookie = "leftmenu=block";
                }
                else
                {
```

```
                    img.src = "images/menu_show.jpg";
                    tdleft.style.width = 0;
                    menu.style.display = 'none';
                    document.cookie = "leftmenu=none";
                }
        }
        var getHeight = screen.height - 252;

</SCRIPT>
<style type="text/css">
.options{display: none;}
</style>
<script language="javascript" type="text/javascript"
src="js/datetimepicker.js"></script>
<script language="javascript" type="text/javascript"
src="js/inputfix.js"></script>
<script language="javascript" type="text/javascript"
src="js/shortcut.js"></script>

<link rel="stylesheet" type="text/css" href="css/sample.css"><!--<script
type="text/javascript" src="includes/jscript/popup-window.js"></script>-->

<link rel="stylesheet" href="css/demo.htm" type="text/css">

<link rel="stylesheet" href="css/modal-message.css" type="text/css">

<link rel="stylesheet" href="css/scroll.css" type="text/css">
<link rel="stylesheet" href="includes/css/autosuggest_inquisitor.css"
type="text/css" media="screen" charset="utf-8" />
<style type="text/css">
input.b {
   width: 300px;
}
</style></head><body>


        <table border="0" cellpadding="0" cellspacing="0" width="100%"
align="center">
<tbody>
        <tr>
<td><img src="images/eFacilitator.jpg" width="103%" height="232" ></td>
</tr>
<TR class=textmid>
```

```
                <td >

                        <%
                        String msg=(String) request.getAttribute("message");
                        if (msg !=null)
                        out.println(msg);
                        %>

                </td>
                        </tr>
<tr>

        <td id="tdMain" style="padding-top: 5px;" valign="top" width="100%">
 <jsp:include page="includes/login.html"/>

                </td>
                        </tr>
                </tbody></table>
</body></html>
```

```
<%@ page
import="prompt.dao.UserDao,prompt.dao.DaoFactory,prompt.beans.Login" %>
<%
UserDao uDao=(UserDao) DaoFactory.getDao("UserDao");
String id=request.getParameter("loginId");
String pass=request.getParameter("password");
Login login=uDao.autenticate(id,pass);
if (login!=null)
{
session.setAttribute("login",login);
%>
<jsp:forward page="../index.jsp?p=home"/>

<%
}else
{
request.setAttribute("message","Invalid userName or Password!");
%>
<jsp:forward page="../promptHome.jsp" />
<%

}
%>
```

```
<table class="login_head" width="100%" >
<tbody><tr>
<td><img src="images/PromptBiz_main.jpg" width="100%" ></td>
</tr>
</tbody>
</table>
<tr class="textmin" width="100%">
	<td >
	<%
	String welcomeNote=(String)request.getAttribute("welcomeNote");
	if (welcomeNote ==null)
	welcomeNote="";
	%>
	<%=welcomeNote%>
	</td>

	</tr>

<tr>
	<td class="top_menu">

	<!--[if lt IE 6]>
<style type="text/css">ul.zzmenuNS li.v, ul.zzmenuNS li.sv { margin-left: -
20px; }</style>
<![endif]-->
<!--[if IE]>
<style type="text/css">ul.zzmenuNS span.h { float: left; }</style>
<![endif]-->
<script src="js/zzmenujs.js" type="text/javascript"></script>
<!--[if lt IE 7]>
<script src="js/zzmenuiehack.js" type="text/javascript"></script>
<![endif]-->




<DIV id=menuDiv
	 style="PADDING-RIGHT: 0px; BORDER-TOP: #000000 0px solid;
PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px;
BORDER-BOTTOM: #000000 1px solid"></DIV>

<script type="text/javascript">
```

```
<!--
var menuLibrary = new ZZEEMenus();
menuLibrary.libPath = 'zzmenu/';
var menu1 = menuLibrary.newMenu();
menu1.horizontal(true);
menu1.addItems([
        {caption: "Profile", subitems: [
        {caption: "View and Edit", url: "index.jsp?p=editProfile", icon:
"images/list_ico.gif"},
        {caption: "Change Password", url: "index.jsp?p=changePassword", icon:
"images/add_ico.gif"},
        ]},
        {caption: "Collection", subitems: [
        {caption: "New Fee Schedule", url: "index.jsp?p=newFeeSchedule", icon:
"images/list_ico.gif"},
   {caption: "Submit Fee", url: "index.jsp?p=submitFee", icon:
"images/add_ico.gif"},
        {caption: "View Dues", url: "index.jsp?p=viewDues", icon:
"images/add_ico.gif"},
        {caption: "View Fee Schedules", url: "index.jsp?p=viewFeeSchedules",
icon: "images/list_ico.gif"},
        ]},
        {caption: "Settings", subitems: [
        {caption: "Service Tax", url: "index.jsp?p=serviceTax", icon:
"images/list_ico.gif"},
   ]},
        {caption: "Logout", url: "index.jsp?p=logout"},
   ]);
        var mi = menu1.getItem([6]);
mi = mi.getItem([0]);

mi.setRadio(1);

mi.setChecked(true);

menu1.getItem([2, 1]).setRadio(1);
menu1.getItem([2, 2]).setRadio(1);
mi = menu1.getItem([2, 4]);
mi.onclick = function() {
   this.setChecked(!this.getChecked());
};
mi.setChecked(false);
```

```
mi = menu1.getItem([2, 5]);
mi.onclick = function() { this.setChecked(!this.getChecked()); };
mi.setChecked(true);
menuLibrary.initMenu(menu1, document.getElementById('menuDiv'));
messageObj = new DHTML_modalMessage();
messageObj.setShadowOffset(5);
function displayShortcut(url)
{
	messageObj.setSource(url);
	messageObj.setCssClassMessageBox(false);
	messageObj.setSize(650,520);
	messageObj.setShadowDivVisible(true);
	messageObj.display();
}

function closeShortcut(val)
{
	messageObj.close();
}

</script>
<!-- End of Javascript part for the menu -->
</td>
</tr>
</tbody></table>
```

```
<jsp:useBean id="course" class="prompt.beans.Course" />
<jsp:setProperty name="course" property="name"/>
<jsp:setProperty name="course" property="description"/>
<jsp:setProperty name="course" property="fee"/>
<jsp:setProperty name="course" property="technologyId" param="techId" />

<%@ page import="prompt.dao.*" %>
<%
CourseDao cDao=(CourseDao) DaoFactory.getDao("CourseDao");
cDao.insertCourse(course);
request.setAttribute("message","course Successfully Added");
%>
<jsp:include page="message.jsp"/>
```

```
<jsp:useBean id="user" class="prompt.beans.User"  />
<jsp:setProperty name="user" property="name"/>
<jsp:setProperty name="user" property="address"/>
<jsp:setProperty name="user" property="mobileNo"/>
<jsp:setProperty name="user" property="mailId"/>
<jsp:setProperty name="user" property="designation"/>
<jsp:setProperty name="user" property="loginId"/>
<jsp:setProperty name="user" property="password"/>
<jsp:setProperty name="user" property="role"/>

<%@ page import="prompt.dao.*" %>
<%
int techId=Integer.parseInt(request.getParameter("techId"));
user.setTechId(techId);
System.out.println("following techId is inserted."+techId);
UserDao udao=(UserDao) DaoFactory.getDao("UserDao");
udao.insertUser(user);
request.setAttribute("message","User Successfully Added");
%>
<jsp:include page="message.jsp"/>
```

```
<table class="login_head" width="100%" >
<tbody><tr>
<td><img src="images/PromptBiz_main.jpg" width="100%" ></td>
</tr>
</tbody>
</table>

<tr class="textmin" width="100%">
	<td >
	<%
	String welcomeNote=(String)request.getAttribute("welcomeNote");
	if (welcomeNote ==null)
	welcomeNote="";
	%>
	<%=welcomeNote%>
	</td>
	</tr>
<tr>
	<td class="top_menu">

	<!--[if lt IE 6]>
<style type="text/css">ul.zzmenuNS li.v, ul.zzmenuNS li.sv { margin-left: -
20px; }</style>
<![endif]-->
<!--[if IE]>
<style type="text/css">ul.zzmenuNS span.h { float: left; }</style>
<![endif]-->
<script src="js/zzmenujs.js" type="text/javascript"></script>
<!--[if lt IE 7]>
<script src="js/zzmenuiehack.js" type="text/javascript"></script>
<![endif]-->

<DIV id=menuDiv
	 style="PADDING-RIGHT: 0px; BORDER-TOP: #000000 0px solid;
PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px;
BORDER-BOTTOM: #000000 1px solid"></DIV>

<script type="text/javascript">
<!--

var menuLibrary = new ZZEEMenus();
menuLibrary.libPath = 'zzmenu/';
```

```
var menu1 = menuLibrary.newMenu();
menu1.horizontal(true);
menu1.addItems([
        {caption: "Profile", subitems: [
        {caption: "Edit Profile", url: "index.jsp?p=editProfile", icon:
"images/list_ico.gif"},
        {caption: "Change Password", url: "index.jsp?p=changePassword", icon:
"images/add_ico.gif"},
        ]},
        {caption: "Technology Management", subitems: [
        {caption: "Add", url: "index.jsp?p=newTechnology", icon:
"images/list_ico.gif"},
        {caption: "view and Edit", url: "index.jsp?p=viewTechnologies", icon:
"images/add_ico.gif"},
        ]},

        {caption: "Course Management", subitems: [
        {caption: "Add", url: "index.jsp?p=newCourse", icon:
"images/list_ico.gif"},
        {caption: "Home", url: "index.jsp?p=home", icon:
"images/add_ico.gif"},
        {caption: "view and Edit", url: "index.jsp?p=viewCourses", icon:
"images/add_ico.gif"},

        ]},

        {caption: "User Management", subitems: [
        {caption: "Add", url: "index.jsp?p=newUser", icon:
"images/list_ico.gif"},
        {caption: "view and Edit", url: "index.jsp?p=viewUsers", icon:
"images/add_ico.gif"},
        ]},

        {caption: "Logout", url: "index.jsp?p=logout"},


   ]);
        var mi = menu1.getItem([6]);
mi = mi.getItem([0]);

//-->
messageObj = new DHTML_modalMessage();
messageObj.setShadowOffset(5);
```

```
function displayShortcut(url)
{
	messageObj.setSource(url);
	messageObj.setCssClassMessageBox(false);
	messageObj.setSize(650,520);
	messageObj.setShadowDivVisible(true);
	messageObj.display();
}

function closeShortcut(val)
{
	messageObj.close();
}

</script>
<!-- End of Javascript part for the menu -->
</td>
</tr>

</tbody></table>

<%@ page
import="prompt.beans.*,prompt.dao.*,java.util.Date,java.text.DateFormat" %>

<%
int enquiryId=Integer.parseInt(request.getParameter("enquiryId"));
Login login=(Login) session.getAttribute("login");
int userId=login.getUser_id();
AttendedEnquiry enquiry=new AttendedEnquiry();
enquiry.setAttendantId(userId);
enquiry.setEnquiryId(enquiryId);
enquiry.setComments(request.getParameter("comments"));
enquiry.setProspect(Integer.parseInt(request.getParameter("prospect")));
DateFormat df=DateFormat.getDateInstance();
String eDate=df.format(new Date());
enquiry.setAttendingDate(eDate);
EnquiryDao eDao=(EnquiryDao) DaoFactory.getDao("EnquiryDao");
eDao.attendEnquiry(enquiry);
request.setAttribute("message","Enquiry Successfully attended.");
%>
<jsp:include page="message.jsp"/>
```

```
<table class="login_head" width="100%" >
<tbody><tr>
<td><img src="images/PromptBiz_main.jpg" width="100%" ></td>
</tr>
</tbody>
</table>


<tr class="textmin" width="100%">
	<td >
	<%
	String welcomeNote=(String)request.getAttribute("welcomeNote");
	if (welcomeNote ==null)
	welcomeNote="";
	%>
	<%=welcomeNote%>
	</td>
	</tr>
<tr>
	<td class="top_menu">



	<!--[if lt IE 6]>
<style type="text/css">ul.zzmenuNS li.v, ul.zzmenuNS li.sv { margin-left: -
20px; }</style>
<![endif]-->
<!--[if IE]>
<style type="text/css">ul.zzmenuNS span.h { float: left; }</style>
<![endif]-->
<script src="js/zzmenujs.js" type="text/javascript"></script>
<!--[if lt IE 7]>
<script src="js/zzmenuiehack.js" type="text/javascript"></script>
<![endif]-->




<DIV id=menuDiv
	 style="PADDING-RIGHT: 0px; BORDER-TOP: #000000 0px solid;
PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px;
BORDER-BOTTOM: #000000 1px solid"></DIV>
```

```
<script type="text/javascript">

menu1.addItems([

        {caption: "Admin", subitems: [

        {caption: "Add User", url: "index.jsp?p=newUser", icon:
"images/list_ico.gif"},
        {caption: "view All Users", url: "index.jsp?p=viewUser", icon:
"images/add_ico.gif"},
        {caption: "delete User", url: "index.jsp?p=deleteUser", icon:
"images/add_ico.gif"}
        ]},

        {caption: "User Profile", subitems: [
        {caption: "view Profile", url: "index.jsp?p=profile", icon:
"images/list_ico.gif"},
        {caption: "change Password", url: "index.jsp?p=changePassword", icon:
"images/add_ico.gif"}

        ]},

        {caption: "Enquiry", subitems: [

        {caption: "New Enquiry", url: "index.jsp?p=enquiry", icon:
"images/list_ico.gif"},
        {caption: "Edit Enquiry", url: "index.jsp?p=editEnquiry", icon:
"images/add_ico.gif"},
   {caption: "Enquiry Reports", url: "index.jsp?p=enqRep", icon:
"images/add_ico.gif"},
        {caption: "Enquiry Followup", url: "index.jsp?p=enquiryFollowup",
icon: "images/add_ico.gif"},
        {caption: "Followup Reports", url: "index.jsp?p=followupRep", icon:
"images/add_ico.gif"},

        ]},

        {caption: "Student", subitems: [
```

```
        {caption: "Register", url: "index.jsp?p=registration", icon:
"images/add_ico.gif"},
   {caption: "Schedule Batch", url: "index.jsp?p=student", icon:
"images/list_ico.gif"},
        {caption: "Allow Break", url: "index.jsp?p=breakDetails", icon:
"images/list_ico.gif"},
   {caption: "Transfer Batch", url: "index.jsp?p=transferDetails", icon:
"images/add_ico.gif"},
        {caption: "View & Edit Details", url: "index.jsp?p=viewStudent", icon:
"images/add_ico.gif"}

        ]},

        {caption: "Collection", subitems: [

        {caption: "New Fee Schedule", url: "index.jsp?p=feeSchedule", icon:
"images/list_ico.gif"},
        {caption: "Submit Fee", url: "index.jsp?p=collectionDetails", icon:
"images/add_ico.gif"},
   {caption: "View Dues", url: "index.jsp?p=dueDetails", icon:
"images/add_ico.gif"},
   {caption: "View & Edit Fee Schedule", url: "index.jsp?p=viewSchedule",
icon: "images/add_ico.gif"},
   {caption: "Recovery Followup", url: "index.jsp?p=recoveryFollowup", icon:
"images/add_ico.gif"}

        ]},


        {caption: "Batch", subitems: [

                        {caption: "New Batch", url: "index.jsp?p=batch", icon:
"images/list_ico.gif"},
                        {caption: "View & Edit Progress", url:
"index.jsp?p=batchProgressDetail", icon: "images/list_ico.gif"},
                        {caption: "View & Edit Details", url:
"index.jsp?p=viewBatchDetails", icon: "images/list_ico.gif"},
                        {caption: "Mark Attendance", url: "index.jsp?p=attendance",
icon: "images/list_ico.gif"}

                        ]},

        {caption: "Others", subitems: [
```

```
                    {caption: "Submit Test Details", url: "index.jsp?p=test",
icon: "images/list_ico.gif"},
                    {caption: "view Test Details", url: "index.jsp?p=testDetails",
icon: "images/add_ico.gif"},
                    {caption: "Certificate Details", url:
"index.jsp?p=certificateDetails", icon: "images/list_ico.gif"},

          ]},

                {caption: "Logout", url: "index.jsp?p=logout"},


   ]);
        var mi = menu1.getItem([6]);

        var menuprice = menu1.getItem([2,1]);



messageObj = new DHTML_modalMessage();
messageObj.setShadowOffset(5);
function displayShortcut(url)
{

        messageObj.setSource(url);
        messageObj.setCssClassMessageBox(false);
        messageObj.setSize(650,520);
        messageObj.setShadowDivVisible(true);
        messageObj.display();
}

function closeShortcut(val)
{

        messageObj.close();
}


</script>
<!-- End of Javascript part for the menu -->


</td>
```

```
</tr>

</tbody></table>


		<% session.invalidate();

		%>
			<jsp:forward page="../promptHome.jsp" />
<table class="login_head" width="100%" >
<tbody><tr>
<td><img src="images/E-FACILITATOR.jpg" width="100%" ></td>
</tr>
</tbody>
</table>


<tr class="textmin" width="100%">
	<td >
	<%
	String welcomeNote=(String)request.getAttribute("welcomeNote");
	if (welcomeNote ==null)
	welcomeNote="";
	%>
	<%=welcomeNote%>
	</td>

	</tr>

<tr>
	<td class="top_menu">



	<!--[if lt IE 6]>
<style type="text/css">ul.zzmenuNS li.v, ul.zzmenuNS li.sv { margin-left: -
20px; }</style>
<![endif]-->
<!--[if IE]>
<style type="text/css">ul.zzmenuNS span.h { float: left; }</style>
<![endif]-->
<script src="js/zzmenujs.js" type="text/javascript"></script>
<!--[if lt IE 7]>
```

```
<script src="js/zzmenuiehack.js" type="text/javascript"></script>
<![endif]-->



<DIV id=menuDiv
        style="PADDING-RIGHT: 0px; BORDER-TOP: #000000 0px solid;
PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px;
BORDER-BOTTOM: #000000 1px solid"></DIV>



<script type="text/javascript">
<!--

var menuLibrary = new ZZEEMenus();
menuLibrary.libPath = 'zzmenu/';
var menu1 = menuLibrary.newMenu();
menu1.horizontal(true);
menu1.addItems([
        {caption: "Profile", subitems: [
        {caption: "View and Edit", url: "index.jsp?p=editProfile", icon:
"images/list_ico.gif"},
        {caption: "Change Password", url: "index.jsp?p=changePassword", icon:
"images/add_ico.gif"},
        ]},
        {caption: "Enquiry Reports", subitems: [
        {caption: "Daily",  url: "index.jsp?p=dailyEnqRep", icon:
"images/add_ico.gif"},
   {caption: "Monthly", url: "index.jsp?p=monthlyEnqRep", icon:
"images/add_ico.gif"},
   {caption: "Quarterly", url: "index.jsp?p=qlyEnqRep", icon:
"images/list_ico.gif"},
   {caption: "yearly", url: "index.jsp?p=yearlyEnqRep", icon:
"images/add_ico.gif"},
        ]},

        {caption: "Registration Reports", subitems: [
        {caption: "Daily",  url: "index.jsp?p=dailyEnqRep", icon:
"images/add_ico.gif"},
```

```
   {caption: "Monthly", url: "index.jsp?p=monthlyRegRep", icon:
"images/list_ico.gif"},
         {caption: "Quarterly", url: "index.jsp?p=qlyRegRep", icon:
"images/add_ico.gif"},
   {caption: "Yearly", url: "index.jsp?p=yearlyRegRep", icon:
"images/add_ico.gif"},
         ]},
         {caption: "Collection Reports", subitems: [
         {caption: "Daily",  url: "index.jsp?p=dailyCollRep", icon:
"images/add_ico.gif"},
   {caption: "Monthly", url: "index.jsp?p=monthlyCollRep", icon:
"images/list_ico.gif"},
         {caption: "Quarterly", url: "index.jsp?p=qlyCollRep", icon:
"images/add_ico.gif"},
   {caption: "Yearly", url: "index.jsp?p=yearlyCollRep", icon:
"images/add_ico.gif"},
         ]},
   {caption: "Expected Collection Reports", subitems: [
   {caption: "Today", url: "index.jsp?p=CashFlowRep", icon:
"images/add_ico.gif"},
   {caption: "This Week",  url: "index.jsp?p=weeklyCashFlowRep", icon:
"images/list_ico.gif"},
         {caption: "This Month", url: "index.jsp?p=montlyCashFlowRep", icon:
"images/add_ico.gif"},
   {caption: "This Quarter", url: "index.jsp?p=qlyCashFlowRep", icon:
"images/add_ico.gif"},
         ]},
         {caption: "Batches Reports", subitems: [
    {caption: "Course Wise", url: "index.jsp?p=courseBatchRep", icon:
"images/list_ico.gif"},
         {caption: "Consultant Wise", url: "index.jsp?p=consultantBatchRep",
icon: "images/list_ico.gif"},
         ]},
         {caption: "Logout", url: "index.jsp?p=logout"},
   ]);
         var mi = menu1.getItem([6]);
mi.onclick = function() {
   this.setChecked(!this.getChecked());
};
messageObj = new DHTML_modalMessage();
messageObj.setShadowOffset(5);

function displayShortcut(url)
```

```
{

        messageObj.setSource(url);
        messageObj.setCssClassMessageBox(false);
        messageObj.setSize(650,520);
        messageObj.setShadowDivVisible(true);
        messageObj.display();
}

function closeShortcut(val)
{

        messageObj.close();
}


</script>
<!-- End of Javascript part for the menu -->


</td>
</tr>

</tbody></table>


<b>
<table>
<tr class="textmid">
<td>
<%=request.getAttribute("message") %>
</td>
<tr>
</table>
</b>
```

```
<%@ page import="prompt.dao.*,prompt.beans.*,java.util.*" %>
<%
TechDao tDao=(TechDao) DaoFactory.getDao("TechDao");
ArrayList list=tDao.findAllTech();
Iterator itr=list.iterator();
%>
<TABLE class=table_page cellSpacing=0 align=center>
<TBODY>
<TR>
<TD width=599>  </TD></TR>
<TR class=textlarge_dark>
<TD>New Course</TD></TR>
<TR>
<TD> </TD></TR>
<TR>
<TD>
<form action="index.jsp?p=addCourse" method="post">
<br><br>
<table  class=table_view cellSpacing=0>
<TBODY>
<TR class=table_head>
<TD colSpan=2>Submit Course Details</TD></TR>
<tr>
<td><b>Name</b></td>
<td><input type="text" size="20" name="name"></td>
</tr>
<tr>
<td><b>description</b></td>
<td><input type="text" size="20" name="description"></td>
</tr>
<tr>
<td><b>fee</b></td>
<td><input type="text" size="20" name="fee"></td>
</tr>
<tr>
<td><b>Technology</b></td>
<td>
<select  name="techId">
<%
        while (itr.hasNext())
        {
        Technology tech=(Technology) itr.next();
        %>
```

```
        <option
value="<%=tech.getTechnologyId()%>"><%=tech.getName()%></option>
        <%
        }
%>
</select>
</td></tr>
</table>
</td>
</tr>

<TR>

<TD> <INPUT type="submit" value="Submit"> 
<INPUT type="reset" value="Cancel"></TD></TR>
</table>
<P></P><P></P></form></TD></TR></TBODY></TABLE>

<%@ page import="prompt.dao.*,java.util.*,prompt.beans.*" %>

<%
CourseDao cDao=(CourseDao) DaoFactory.getDao("CourseDao");
ArrayList list=cDao.findAllCourse();
Iterator itr=list.iterator();
%>
<TABLE class=table_page cellSpacing=0 align=center>
<TBODY>
<TR>
<TD width=599>  </TD></TR>
<TR class=textlarge_dark>
<TD>New Enquiry</TD></TR>
<TR>
<TD> </TD></TR>
<TR>
<TD>
<form action="index.jsp?p=newEnquiry" method="post" >
<br><br>
<table class=table_view cellSpacing=0>
<TBODY>
<TR class=table_head>
<TD colSpan=2>Enquiry</TD></TR>
<tr>
<td><b>Name</b></td>
```

```
<td><input type="text" size="20" name="name"></td>
</tr>
<tr>
<td><b>MailId</b></td>
<td><input type="text" size="20" name="mailId"></td>
</tr>
<tr>
<td><b>MobileNo</b></td>
<td><input type="text" size="20" name="mobileNo"></td>
</tr>
<tr>
<td><b>Mode</b></td>
<td><select  name="mode">
        <option value="1">In Person</option>
        <option value="2" selected>Mail</option>
        <option value="3">Telephonic</option>
</select></td>
</tr>
<tr>
<td><b>course</b></td>
<td><select  name="courseId">
<%
        while (itr.hasNext())
        {
        Course course=(Course) itr.next();
        %>
        <option
value="<%=course.getCourseId()%>"><%=course.getName()%></option>
        <%
        }
%>
</select>
</td>
</tr>
<tr>
<td><b>AwarenessSource</b></td>
<td><select  name="awarenessSource">
        <option value="1">Friend</option>
        <option value="2" selected>News Paper</option>
        <option value="3">Ad..</option>
        <option value="4">Other</option>
</select></td>
</tr><TR>
```

```
<TD> </TD>
<TD><INPUT type="submit" value="Submit" > 
<INPUT type="reset" value="Cancel"></TD></TR>
</table>
</form></TD></TR></TBODY></TABLE>
<jsp:useBean id="tech" class="prompt.beans.Technology"  />
<jsp:setProperty name="tech" property="name"/>
<jsp:setProperty name="tech" property="description"/>

<%@ page import="prompt.dao.*" %>
<%
TechDao tDao=(TechDao) DaoFactory.getDao("TechDao");
tDao.insertTech(tech);
request.setAttribute("message","New Technology Successfully Added");
%>
<jsp:include page="message.jsp"/>
```

```
<%@ page import="prompt.pagination.*,prompt.helper.*" %>
<%
Paginate paginate=null;
String pg=request.getParameter("page");

if (pg==null)
{
System.out.println("Not Found Creating.");
paginate=ToBeAttendedPaginator.getPaginate(3);
paginate.setPreviousUrl("index.jsp?p=attendEnquiry&page=previous");
paginate.setNextUrl("index.jsp?p=attendEnquiry&page=next");
session.setAttribute("toBeAttended",paginate);
}
else
{
System.out.println("finding paginate in session");
paginate=(Paginate)session.getAttribute("toBeAttended");
}
if (pg !=null && pg.equals("previous"))
paginate.setPrevious();
else if (pg !=null && pg.equals("next"))
paginate.setNext();

paginate.setOut(out);
paginate.displayPage();
%>
```

```
<%@ page import="prompt.dao.*,prompt.beans.*,java.util.*" %>
<%
TechDao tDao=(TechDao) DaoFactory.getDao("TechDao");
ArrayList list=tDao.findAllTech();
Iterator itr=list.iterator();
%>
<TABLE class=table_page cellSpacing=0 align=center>
<TBODY>
<TR>
<TD width=599>  </TD></TR>
<TR class=textlarge_dark>
<TD>New User</TD></TR>
<TR>
<TD> </TD></TR>
<TR>
<TD>
<form action="index.jsp?p=addUser" method="post">
<br><br>
<table class=table_view cellSpacing=0>
<TBODY>
<TR class=table_head>
<TD colSpan=2>Personal Information</TD></TR>
<tr>
<td><b>Name</b></td>
<td><input type="text" size="20" name="name"></td>
</tr>
<tr>
<td><b>Address</b></td>
<td><input type="text" size="20" name="address"></td>
</tr>
<tr>
<td><b>MobileNo</b></td>
<td><input type="text" size="20" name="mobileNo"></td>
</tr>
<tr>
<td><b>MailId</b></td>
<td><input type="text" size="20" name="mailId"></td>
</tr>
<tr>
<td><b>Designation</b></td>
<td><input type="text" size="20" name="designation"></td>
</tr>
<tr>
```

```
<td><b>LoginId</b></td>
<td><input type="text" size="20" name="loginId"></td>
</tr>
<tr>
<td><b>Password   </b></td>
<td><input type="password" size="20" name="password"></td>
</tr>
<tr>
<td><b>Technology</b></td>
<td>
<select  name="techId">
<%
        while (itr.hasNext())
        {
        Technology tech=(Technology) itr.next();
        %>
        <option
value="<%=tech.getTechnologyId()%>"><%=tech.getName()%></option>
        <%
        }
%>
</select>
</td></tr>
<tr>
<td><b>Role</b></td>
<td>
<select  name="role">
        <option value="Admin">Admin</option>
        <option value="coor">Manager</option>
        <option value="Accountant" >Accountant</option>
        <option value="Consultant">Consultant</option>
        <option value="Counselor" selected>Counselor</option>
        <option value=Coordinator">Coordinator</option>


</select>
</td>
</tr>
</table>
</td>
</tr>

<TR>
```

<TD> <INPUT type="submit" value="Submit">
<INPUT type="reset" value="Cancel"></TD></TR>
</table>
<P></P><P></P></form></TD></TR></TBODY></TABLE>

```
<%@ page import="prompt.pagination.*,prompt.helper.*" %>
<%
Paginate paginate=null;
String pg=request.getParameter("page");

if (pg==null)
{
System.out.println("Not Found Created.");
paginate=CoursePaginator.getPaginate(3);
paginate.setPreviousUrl("index.jsp?p=viewCourses&page=previous");
paginate.setNextUrl("index.jsp?p=viewCourses&page=next");
session.setAttribute("coursePaginate",paginate);
}
else
{
System.out.println("finding paginate in session");
paginate=(Paginate)session.getAttribute("coursePaginate");
}
if (pg !=null && pg.equals("previous"))
paginate.setPrevious();
else if (pg !=null && pg.equals("next"))
paginate.setNext();

paginate.setOut(out);
paginate.displayPage();
%>
<%@ page import="prompt.pagination.*,prompt.helper.*" %>

<%
Paginate paginate=null;
String pg=request.getParameter("page");
if (pg==null)
{
System.out.println("new request is received.");
paginate=TechPaginator.getPaginate(3);
paginate.setPreviousUrl("index.jsp?p=viewTechnologies&page=previous");
paginate.setNextUrl("index.jsp?p=viewTechnologies&page=next");
session.setAttribute("techPaginate",paginate);

}
else
{
System.out.println("finding paginate in session");
```

```
paginate=(Paginate)session.getAttribute("techPaginate");
}
if (pg !=null && pg.equals("previous"))
paginate.setPrevious();
else if (pg !=null && pg.equals("next"))
paginate.setNext();

paginate.setOut(out);
paginate.displayPage();
%>
<%@ page import="prompt.pagination.*,prompt.helper.*" %>
<%
Paginate paginate=null;
String pg=request.getParameter("page");

if (pg==null)
{
System.out.println("Not Found Created.");
paginate=UserPaginator.getPaginate(3);
paginate.setPreviousUrl("index.jsp?p=viewUsers&page=previous");
paginate.setNextUrl("index.jsp?p=viewUsers&page=next");
session.setAttribute("paginate",paginate);
}
else
{
System.out.println("finding paginate in session");
paginate=(Paginate)session.getAttribute("paginate");
}
if (pg !=null && pg.equals("previous"))
paginate.setPrevious();
else if (pg !=null && pg.equals("next"))
paginate.setNext();

paginate.setOut(out);
paginate.displayPage();
%>
```

```
#ajax_tooltipObj {
	Z-INDEX: 1000000; TEXT-ALIGN: left
}
#ajax_tooltipObj DIV {
	POSITION: relative
}
#ajax_tooltipObj .ajax_tooltip_arrow {
	BACKGROUND-POSITION: left center; Z-INDEX: 1000005; LEFT:
0px; BACKGROUND-IMAGE: url(../../images/arrow_tip.gif); WIDTH: 20px;
BACKGROUND-REPEAT: no-repeat; POSITION: absolute; TOP: 0px;
HEIGHT: 30px
}
#ajax_tooltipObj .ajax_tooltip_content {
	BORDER-RIGHT: #317082 2px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #317082 2px solid; PADDING-LEFT: 5px; FONT-SIZE:
11px; Z-INDEX: 1000001; LEFT: 18px; PADDING-BOTTOM: 5px;
OVERFLOW: auto; BORDER-LEFT: #317082 2px solid; WIDTH: 450px;
LINE-HEIGHT: 15px; PADDING-TOP: 5px; BORDER-BOTTOM: #317082
2px solid; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; POSITION:
absolute; TOP: 0px; HEIGHT: auto; BACKGROUND-COLOR: #fff
}
```

```
#ajax_listOfOptions {
	BORDER-RIGHT: #317082 1px solid; BORDER-TOP: #317082 1px solid; FONT-SIZE: 0.9em; Z-INDEX: 100; OVERFLOW: auto; BORDER-LEFT: #317082 1px solid; WIDTH: 175px; BORDER-BOTTOM: #317082 1px solid; POSITION: absolute; HEIGHT: 250px; BACKGROUND-COLOR: #fff; TEXT-ALIGN: left
}
#ajax_listOfOptions DIV {
	PADDING-RIGHT: 1px; PADDING-LEFT: 1px; FONT-SIZE: 0.9em; PADDING-BOTTOM: 1px; MARGIN: 1px; CURSOR: pointer; PADDING-TOP: 1px
}
#ajax_listOfOptions .optionDiv {

}
#ajax_listOfOptions .optionDivSelected {
	COLOR: #fff; BACKGROUND-COLOR: #317082
}
#ajax_listOfOptions_iframe {
	 Z-INDEX: 5; POSITION: absolute; BACKGROUND-COLOR: #f00
}
```

```
.menu A {
	FONT-WEIGHT: bold; COLOR: #006699; TEXT-DECORATION: none
}
.options A {
	FONT-WEIGHT: bold; COLOR: #006699; TEXT-DECORATION: none
}
.menu A:hover {
	FONT-WEIGHT: bold; COLOR: #000000; TEXT-DECORATION: none
}
.options A:hover {
	FONT-WEIGHT: bold; COLOR: #000000; TEXT-DECORATION: none
}
.menu A:active {
	FONT-WEIGHT: bold; COLOR: #000000; TEXT-DECORATION: none
}
.options A:active {
	FONT-WEIGHT: bold; COLOR: #000000; TEXT-DECORATION: none
}
.Menu {
	BORDER-RIGHT: 0px; PADDING-RIGHT: 3px; BORDER-TOP: #fff
2px solid; PADDING-LEFT: 3px; FONT-SIZE: 11px; PADDING-BOTTOM:
3px; VERTICAL-ALIGN: middle; BORDER-LEFT: #b4b4b4 1px solid;
WIDTH: 150px; CURSOR: pointer; COLOR: #000000; PADDING-TOP: 3px;
BORDER-BOTTOM: #b4b4b4 1px solid; FONT-FAMILY: tahoma, verdana,
helvetica; BACKGROUND-COLOR: #eeeeee
}
.Option {
	PADDING-RIGHT: 4px; PADDING-LEFT: 4px; PADDING-BOTTOM:
4px; VERTICAL-ALIGN: middle; WIDTH: 139px; CURSOR: pointer;
PADDING-TOP: 4px; BACKGROUND-COLOR: #ffffff
}
.Options {
	PADDING-LEFT: 10px; FONT-WEIGHT: normal; FONT-SIZE: 10px;
VERTICAL-ALIGN: middle; COLOR: #000000; FONT-FAMILY: verdana;
BACKGROUND-COLOR: #ffffff
}



.modalDialog_transparentDivs {
	Z-INDEX: 1; FILTER: alpha(opacity=40); POSITION: absolute;
BACKGROUND-COLOR: #aaa; opacity: 0.4
}
```

```
.modalDialog_contentDiv {
	BORDER-RIGHT: #000 3px solid; PADDING-RIGHT: 2px; BORDER-TOP: #000 3px solid; PADDING-LEFT: 2px; Z-INDEX: 100; PADDING-BOTTOM: 2px; BORDER-LEFT: #000 3px solid; PADDING-TOP: 2px; BORDER-BOTTOM: #000 3px solid; POSITION: absolute; BACKGROUND-COLOR: #fff
}
.modalDialog_contentDiv_shadow {
	Z-INDEX: 90; FILTER: alpha(opacity=30); POSITION: absolute; BACKGROUND-COLOR: #555; opacity: 0.3
}
```

```
DIV.sample_popup {
	Z-INDEX: 1
}
DIV.sample_popup DIV.menu_form_header {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: #cccccc 1px
solid; FONT-WEIGHT: bold; FONT-SIZE: 14px; BACKGROUND: #3e92bf;
VERTICAL-ALIGN: middle; BORDER-LEFT: #cccccc 1px solid; WIDTH:
400px; CURSOR: default; COLOR: #ffffff; LINE-HEIGHT: 19px; BORDER-
BOTTOM: medium none; FONT-FAMILY: Arial; HEIGHT: 20px; TEXT-
ALIGN: left; TEXT-DECORATION: none
}
DIV.sample_popup DIV.menu_form_body {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: #cccccc 1px
solid; FONT-WEIGHT: 500; FONT-SIZE: 12px; BACKGROUND: #ffffff left
bottom; BORDER-LEFT: #cccccc 1px solid; WIDTH: 400px; CURSOR:
default; COLOR: #999999; BORDER-BOTTOM: #cccccc 1px solid; FONT-
FAMILY: Arial; TEXT-ALIGN: left; TEXT-DECORATION: none
}
DIV.sample_popup IMG.menu_form_exit {
	FLOAT: right; MARGIN: 4px 5px 0px 0px; CURSOR: pointer
}
DIV.sample_popup TABLE {
	WIDTH: 100%; BORDER-COLLAPSE: collapse
}
DIV.sample_popup TH {
	PADDING-RIGHT: 5px; PADDING-LEFT: 0px; FONT-WEIGHT: 900;
FONT-SIZE: 13px; PADDING-BOTTOM: 1px; WIDTH: 1%; COLOR:
#004060; PADDING-TOP: 0px; FONT-FAMILY: Times New Roman, Serif;
TEXT-ALIGN: left
}
DIV.sample_popup TD {
	PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM:
1px; WIDTH: 99%; PADDING-TOP: 0px
}
DIV.sample_popup FORM {
	PADDING-RIGHT: 10px; PADDING-LEFT: 10px; PADDING-
BOTTOM: 10px; MARGIN: 0px; PADDING-TOP: 8px
}
DIV.sample_popup INPUT.field {
	BORDER-RIGHT: #808080 1px solid; BORDER-TOP: #808080 1px
solid; FONT-SIZE: 12px; BORDER-LEFT: #808080 1px solid; WIDTH: 95%;
BORDER-BOTTOM: #808080 1px solid; FONT-FAMILY: Verdana, Sans-
Serif
```

```
}
DIV.sample_popup INPUT.btn {
	BORDER-RIGHT: #808080 1px solid; BORDER-TOP: #808080 1px
solid; MARGIN-TOP: 2px; FONT-SIZE: 11px; BORDER-LEFT: #808080 1px
solid; BORDER-BOTTOM: #808080 1px solid; FONT-FAMILY: Verdana,
Sans-Serif; BACKGROUND-COLOR: #ddffdd
}
.tableWidget_headerCell {
	BORDER-RIGHT: #aca899 1px solid; BORDER-LEFT: #fff 1px solid;
CURSOR: pointer; BORDER-BOTTOM: #c5c2b2 3px solid;
BACKGROUND-COLOR: #ece9d8
}
.tableWigdet_headerCellOver {
	BORDER-RIGHT: #aca899 1px solid; BORDER-LEFT: #fff 1px solid;
CURSOR: pointer; BORDER-BOTTOM: #c5c2b2 3px solid;
BACKGROUND-COLOR: #ece9d8
}
.tableWigdet_headerCellDown {
	BORDER-RIGHT: #aca899 1px solid; BORDER-LEFT: #fff 1px solid;
CURSOR: pointer; BORDER-BOTTOM: #c5c2b2 3px solid;
BACKGROUND-COLOR: #ece9d8
}
.tableWidget_headerCell {
	BORDER-TOP: #ece9d8 2px solid
}
.tableWigdet_headerCellOver {
	BORDER-TOP: #ffc83c 2px solid
}
.tableWidget TBODY .tableWidget_dataRollOver {
	BACKGROUND-COLOR: #fff
}
.tableWigdet_headerCellDown {
	BORDER-RIGHT: #fff 1px solid; BORDER-TOP: #ffc83c 2px solid;
BORDER-LEFT: #aca899 1px solid; BACKGROUND-COLOR: #dbd8c5
}
.tableWidget {
	 FONT-SIZE: 12px; WIDTH: 400px; FONT-FAMILY: arial; BORDER-
COLLAPSE: collapse
}
.tableWidget TD {
	BORDER-RIGHT: #e6e3df 1px solid; PADDING-RIGHT: 2px;
BORDER-TOP: 0px; PADDING-LEFT: 5px; FONT-SIZE: 12px; PADDING-
BOTTOM: 2px; MARGIN: 0px; BORDER-LEFT: #e6e3df 1px solid;
```

```
PADDING-TOP: 2px; BORDER-BOTTOM: #e6e3df 1px solid; BORDER-
COLLAPSE: collapse
}
.tableWidget TBODY {

}
DIV.widget_tableDiv {
	OVERFLOW-Y: auto; WIDTH: 400px
}
UNKNOWN {
	OVERFLOW: hidden; WIDTH: 400px
}
.tableWidget THEAD {
	POSITION: relative
}
.tableWidget THEAD TR {
	BOTTOM: 0px; POSITION: relative; ; TOP:
expression(offsetParent.scrollTop)
}
.tableWidget .scrollingContent {
	OVERFLOW-Y: auto; OVERFLOW-X: hidden; WIDTH: 100%
}
```

```
BODY {
	PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM:
0px; MARGIN: 0px; PADDING-TOP: 0px; FONT-FAMILY: Arial, Helvetica,
sans-serif; BACKGROUND-COLOR: #f3f3f3
}
INPUT {
	BORDER-RIGHT: #e5e5e5 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: #e5e5e5 1px solid; PADDING-LEFT: 1px; FONT-SIZE:
12px; PADDING-BOTTOM: 1px; BORDER-LEFT: #e5e5e5 1px solid;
PADDING-TOP: 1px; BORDER-BOTTOM: #e5e5e5 1px solid; FONT-
FAMILY: Arial, Helvetica, sans-serif
}
.hline {
	BACKGROUND: #333333; size: 1px
}
```

```
INPUT:unknown {
	BORDER-RIGHT: #b9b6aa 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: #b9b6aa 1px solid; PADDING-LEFT: 1px; FONT-SIZE:
12px; PADDING-BOTTOM: 1px; BORDER-LEFT: #b9b6aa 1px solid;
PADDING-TOP: 1px; BORDER-BOTTOM: #b9b6aa 1px solid; FONT-
FAMILY: Arial, Helvetica, sans-serif; BACKGROUND-COLOR: #eeedea
}
INPUT.sffocus {
	BORDER-RIGHT: #b9b6aa 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: #b9b6aa 1px solid; PADDING-LEFT: 1px; FONT-SIZE:
12px; PADDING-BOTTOM: 1px; BORDER-LEFT: #b9b6aa 1px solid;
PADDING-TOP: 1px; BORDER-BOTTOM: #b9b6aa 1px solid; FONT-
FAMILY: Arial, Helvetica, sans-serif; BACKGROUND-COLOR: #eeedea
}
SELECT {
	BORDER-RIGHT: #e5e5e5 1px solid; BORDER-TOP: #e5e5e5 1px
solid; FONT-SIZE: 12px; BORDER-LEFT: #e5e5e5 1px solid; BORDER-
BOTTOM: #e5e5e5 1px solid; FONT-FAMILY: Arial, Helvetica, sans-serif
}
.checkbox {
	BORDER-RIGHT: 0px; BORDER-TOP: 0px; BORDER-LEFT: 0px;
BORDER-BOTTOM: 0px
}
TEXTAREA {
	BORDER-RIGHT: #e5e5e5 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: #e5e5e5 1px solid; PADDING-LEFT: 1px; FONT-SIZE:
12px; PADDING-BOTTOM: 1px; BORDER-LEFT: #e5e5e5 1px solid;
PADDING-TOP: 1px; BORDER-BOTTOM: #e5e5e5 1px solid; FONT-
FAMILY: Arial, Helvetica, sans-serif
}
.btn {
	BACKGROUND-IMAGE: url(../../images/btn_bg.jpg); WIDTH: 50px;
CURSOR: pointer; COLOR: #ffffff; BACKGROUND-REPEAT: repeat-x;
HEIGHT: 22px
}
.btn_confirm {
	BACKGROUND-IMAGE: url(../../images/btn_bg.jpg); WIDTH: 90px;
CURSOR: pointer; COLOR: #ffffff; BACKGROUND-REPEAT: repeat-x;
HEIGHT: 22px
}
.btnlong {
```

```
	BACKGROUND-IMAGE: url(../../images/btn_bg.jpg); WIDTH: 115px;
CURSOR: pointer; COLOR: #ffffff; BACKGROUND-REPEAT: repeat-x;
HEIGHT: 22px
}
.btnremove {
	BACKGROUND-IMAGE: url(../../images/btn_bg.jpg); WIDTH: 55px;
CURSOR: pointer; COLOR: #ffffff; BACKGROUND-REPEAT: repeat-x;
HEIGHT: 22px
}
.btn_generate {
	BACKGROUND-IMAGE: url(../../images/btn_bg.jpg); WIDTH: 60px;
CURSOR: pointer; COLOR: #ffffff; BACKGROUND-REPEAT: repeat-x;
HEIGHT: 22px
}
.table_main {
	MARGIN: 0px; WIDTH: 100%
}
.table_main TD {
	PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM:
0px; PADDING-TOP: 0px
}
.colheadalign {
	TEXT-ALIGN: left
}
.login_head {
	WIDTH: 100%; BACKGROUND-COLOR: #ffffff
}
.login_head TD {
	PADDING-RIGHT: 10px; PADDING-LEFT: 10px; FONT-SIZE: 22px;
PADDING-BOTTOM: 0px; COLOR: #1667a8; PADDING-TOP: 0px;
LETTER-SPACING: 5px
}
.table_login {
	BORDER-RIGHT: #e5e5e5 3px solid; BORDER-TOP: #e5e5e5 3px
solid; BORDER-LEFT: #e5e5e5 3px solid; WIDTH: 35%; BORDER-
BOTTOM: #e5e5e5 3px solid; BACKGROUND-COLOR: #ffffff
}
.table_login TD {
	PADDING-RIGHT: 3px; PADDING-LEFT: 3px; FONT-SIZE: 12px;
PADDING-BOTTOM: 3px; COLOR: #000000; PADDING-TOP: 3px; TEXT-
ALIGN: left
}
.login_title TD {
```

```
	PADDING-RIGHT: 10px; PADDING-LEFT: 10px; FONT-SIZE: 20px;
PADDING-BOTTOM: 10px; COLOR: #0505d8; PADDING-TOP: 10px;
BORDER-BOTTOM: #e5e5e5 1px solid; LETTER-SPACING: 2px
}
.errmsg TD {
	FONT-SIZE: 12px; PADDING-BOTTOM: 5px; COLOR: #ff0000
}
.errmsg {
	FONT-WEIGHT: normal; FONT-SIZE: 11px; COLOR: #ff0000
}
.table_page {
	WIDTH: 95%
}
.table_page TD {
	FONT-SIZE: 11px
}
.table_menu {
	WIDTH: 100%; BACKGROUND-COLOR: #333333
}
.table_menu TD {
	FONT-SIZE: 12px; COLOR: #ffffff
}
.table_list {
	BORDER-TOP: 0px; WIDTH: 100%; BORDER-COLLAPSE: collapse
}
.table_list TD {
	BORDER-RIGHT: #e6e3df 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: 0px; PADDING-LEFT: 5px; FONT-SIZE: 12px; PADDING-
BOTTOM: 1px; BORDER-LEFT: #e6e3df 1px solid; PADDING-TOP: 1px;
BORDER-BOTTOM: #e6e3df 1px solid; BORDER-COLLAPSE: collapse
}
.table_list A {
	FONT-SIZE: 12px; COLOR: #0303d9; TEXT-DECORATION: none
}
.table_list A:visited {
	FONT-SIZE: 12px; COLOR: #0303d9; TEXT-DECORATION: none
}
.table_list A:hover {
	FONT-SIZE: 12px; COLOR: #ff0000; TEXT-DECORATION: none
}
.table_view {
```

```
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: 0px; BORDER-
LEFT: #cccccc 1px solid; WIDTH: 70%; BORDER-BOTTOM: #cccccc 1px
solid; BORDER-COLLAPSE: collapse
}
.table_view TD {
	BORDER-RIGHT: #cccccc 1px solid; PADDING-RIGHT: 3px;
BORDER-TOP: 0px; PADDING-LEFT: 3px; FONT-SIZE: 12px; PADDING-
BOTTOM: 3px; BORDER-LEFT: #cccccc 1px solid; PADDING-TOP: 3px;
BORDER-BOTTOM: #cccccc 1px solid; BORDER-COLLAPSE: collapse
}
.table_head TD {
	PADDING-RIGHT: 4px; PADDING-LEFT: 4px; FONT-WEIGHT:
normal; FONT-SIZE: 12px; PADDING-BOTTOM: 4px; COLOR: #000000;
PADDING-TOP: 4px; BORDER-BOTTOM: #333333 1px solid;
BACKGROUND-COLOR: #d4d0c8; TEXT-ALIGN: left
}
.table_head TH {
	BORDER-RIGHT: #e6e3df 1px solid; PADDING-RIGHT: 4px;
PADDING-LEFT: 4px; FONT-WEIGHT: normal; FONT-SIZE: 12px;
PADDING-BOTTOM: 4px; COLOR: #000000; PADDING-TOP: 4px;
BORDER-BOTTOM: #333333 1px solid; BACKGROUND-COLOR: #d4d0c8
}
.table_back {
	BORDER-RIGHT: #e5e5e5 1px solid; BORDER-TOP: #e5e5e5 1px
solid; MARGIN-TOP: 3px; MARGIN-BOTTOM: 3px; BORDER-LEFT:
#e5e5e5 1px solid; WIDTH: 100%; BORDER-BOTTOM: #e5e5e5 1px solid;
BACKGROUND-COLOR: #eeeeee
}
.table_back TD {
	PADDING-RIGHT: 2px; PADDING-LEFT: 4px; FONT-SIZE: 12px;
PADDING-BOTTOM: 2px; PADDING-TOP: 2px
}
.rpttable_back {
	BORDER-RIGHT: #e5e5e5 1px solid; BORDER-TOP: #e5e5e5 1px
solid; MARGIN-TOP: 3px; MARGIN-BOTTOM: 3px; BORDER-LEFT:
#e5e5e5 1px solid; WIDTH: 100%; BORDER-BOTTOM: #e5e5e5 1px solid;
BACKGROUND-COLOR: #eeeeee
}
.rpttable_back TD {
	PADDING-RIGHT: 0px; PADDING-LEFT: 1px; FONT-SIZE: 12px;
PADDING-BOTTOM: 0px; PADDING-TOP: 0px
}
.head_dark TD {
```

```
	PADDING-RIGHT: 5px; PADDING-LEFT: 5px; FONT-WEIGHT:
bold; FONT-SIZE: 14px; PADDING-BOTTOM: 5px; COLOR: #ffffff;
PADDING-TOP: 5px; LETTER-SPACING: 1px; BACKGROUND-COLOR:
#666666; TEXT-ALIGN: left
}
.back_dark TD {
	FONT-SIZE: 12px; COLOR: #ffffff; BACKGROUND-COLOR:
#1866a4
}
.textlarge_dark TD {
	FONT-SIZE: 24px; COLOR: "#ff8080"
}
.textmid TD {
	 FONT-SIZE: 18px; PADDING-LEFT: 25px; COLOR: "#ff8000"
}
.textmin TD {
	 FONT-SIZE: 12px; PADDING-LEFT: 5px; COLOR: "#ff8000"
}

.evenrow {
	BACKGROUND-COLOR: #f9f9f9
}
.oddrow {
	BACKGROUND-COLOR: #ffffff
}
.rolloverrow {
	BACKGROUND-COLOR: #f2f2ff
}
.noborder {
	BORDER-RIGHT: medium none; BORDER-TOP: medium none;
BORDER-LEFT: medium none; BORDER-BOTTOM: medium none
}
.table_pagging {
	WIDTH: 100%
}
.table_pagging TD {
	PADDING-RIGHT: 3px; PADDING-LEFT: 3px; PADDING-BOTTOM:
3px; PADDING-TOP: 3px
}
.table_pagging A {
	 FONT-SIZE: 12px; COLOR: #666666; TEXT-DECORATION: none
}
.table_pagging A:visited {
```

```
	FONT-SIZE: 12px; COLOR: #666666; TEXT-DECORATION: none
}
.table_pagging A:hover {
	FONT-SIZE: 12px; COLOR: #000000; TEXT-DECORATION:
underline
}
.text_mediumdark {
	FONT-SIZE: 12px; COLOR: #000000; LETTER-SPACING: 0px
}
.table_home TD {
	BORDER-RIGHT: #afae96 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #afae96 1px solid; PADDING-LEFT: 5px; PADDING-
BOTTOM: 5px; BORDER-LEFT: #afae96 1px solid; PADDING-TOP: 5px;
BORDER-BOTTOM: #afae96 1px solid; BACKGROUND-COLOR: #e8e9e3
}
.table_home TD:hover {
	BORDER-RIGHT: #94afc0 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #94afc0 1px solid; PADDING-LEFT: 5px; PADDING-
BOTTOM: 5px; BORDER-LEFT: #94afc0 1px solid; PADDING-TOP: 5px;
BORDER-BOTTOM: #94afc0 1px solid; BACKGROUND-COLOR: #d9d9cd
}
.table_home A {
	FONT-SIZE: 12px; COLOR: #246fa8
}
.table_home A:hover {
	FONT-SIZE: 12px; COLOR: #000000; TEXT-DECORATION: none
}
.box_home {
	BORDER-RIGHT: #b8b7a3 1px solid; BORDER-TOP: #b8b7a3 1px
solid; FLOAT: left; MARGIN: 2px; BORDER-LEFT: #b8b7a3 1px solid;
WIDTH: 110px; BORDER-BOTTOM: #b8b7a3 1px solid; HEIGHT: 110px;
BACKGROUND-COLOR: #c0bfad
}
.alertmsg {
	MARGIN-TOP: 3px; FONT-SIZE: 12px; MARGIN-BOTTOM: 3px;
WIDTH: 100%; COLOR: #ff0000; FONT-FAMILY: Arial, Helvetica, sans-
serif; TEXT-ALIGN: center
}
A.whitelink {
	FONT-SIZE: 11px; COLOR: #ffffff; TEXT-DECORATION: underline
}
A.whitelink:visited {
	FONT-SIZE: 11px; COLOR: #ffffff; TEXT-DECORATION: underline
```

```
}
A.whitelink:hover {
	FONT-SIZE: 11px; COLOR: #ffffff; TEXT-DECORATION: none
}
.table_home TD {
	BORDER-RIGHT: #afae96 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #afae96 1px solid; PADDING-LEFT: 5px; PADDING-
BOTTOM: 5px; BORDER-LEFT: #afae96 1px solid; PADDING-TOP: 5px;
BORDER-BOTTOM: #afae96 1px solid; BACKGROUND-COLOR: #e8e9e3
}
.table_home TD:hover {
	BORDER-RIGHT: #94afc0 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #94afc0 1px solid; PADDING-LEFT: 5px; PADDING-
BOTTOM: 5px; BORDER-LEFT: #94afc0 1px solid; PADDING-TOP: 5px;
BORDER-BOTTOM: #94afc0 1px solid; BACKGROUND-COLOR: #d9d9cd
}
.table_home A {
	FONT-SIZE: 12px; COLOR: #246fa8
}
.table_home A:hover {
	FONT-SIZE: 12px; COLOR: #000000; TEXT-DECORATION: none
}
.pointercursor {
	CURSOR: pointer
}
.rules {
	FONT-SIZE: 12px; COLOR: #ff0000; FONT-FAMILY: Arial, Helvetica,
sans-serif
}
.smalltext {
	FONT-SIZE: 10px; COLOR: #000000; FONT-FAMILY: Arial,
Helvetica, sans-serif
}
.home {
	BORDER-RIGHT: #b4b4b4 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #b4b4b4 1px solid; PADDING-LEFT: 5px; FONT-SIZE:
11px; FLOAT: left; PADDING-BOTTOM: 5px; MARGIN: 0px; BORDER-
LEFT: #b4b4b4 1px solid; WIDTH: 190px; PADDING-TOP: 5px; BORDER-
BOTTOM: #b4b4b4 1px solid; FONT-FAMILY: Verdana, Arial, Helvetica,
sans-serif; HEIGHT: auto; BACKGROUND-COLOR: #e1ded9
}
.hometitle {
```

```
	FONT-WEIGHT: bold; FONT-SIZE: 13px; FONT-FAMILY: Verdana,
Arial, Helvetica, sans-serif
}
.home DIV {
	FONT-SIZE: 11px; FONT-FAMILY: Verdana, Arial, Helvetica, sans-
serif
}
.home A {
	 FONT-SIZE: 10px; COLOR: #0303d9; TEXT-DECORATION: none
}
.home A:visited {
	 FONT-SIZE: 10px; COLOR: #0303d9; TEXT-DECORATION: none
}
.homeover {
	BORDER-RIGHT: #888888 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #888888 1px solid; PADDING-LEFT: 5px; FONT-SIZE:
11px; FLOAT: left; PADDING-BOTTOM: 5px; BORDER-LEFT: #888888
1px solid; WIDTH: 190px; CURSOR: pointer; PADDING-TOP: 5px;
BORDER-BOTTOM: #888888 1px solid; FONT-FAMILY: Verdana, Arial,
Helvetica, sans-serif; HEIGHT: auto; BACKGROUND-COLOR: #dbdbdb
}
.homeover DIV {
	FONT-SIZE: 11px; FONT-FAMILY: Verdana, Arial, Helvetica, sans-
serif; TEXT-ALIGN: justify
}
.homeover A {
	 FONT-SIZE: 10px; COLOR: #0303d9; TEXT-DECORATION: none
}
.homeover A:hover {
	 FONT-SIZE: 10px; COLOR: #ff0000; TEXT-DECORATION: none
}
.contest_desc {
	OVERFLOW-X: hidden; OVERFLOW: auto
}
.btncontest {
	BACKGROUND-IMAGE: url(../../images/btn_bg.jpg); VERTICAL-
ALIGN: top; WIDTH: 70px; CURSOR: pointer; COLOR: #ffffff;
BACKGROUND-REPEAT: repeat-x; HEIGHT: 15px; TEXT-ALIGN: center
}
.addimage {
	WIDTH: 12%; TEXT-ALIGN: right
}
.addimage A {
```

```
        FONT-SIZE: 12px; COLOR: #1866a4; TEXT-DECORATION: none
}
.addimage A:visited {
        FONT-SIZE: 12px; COLOR: #1866a4; TEXT-DECORATION: none
}
.addimage A:hover {
        FONT-SIZE: 12px; COLOR: #000000; TEXT-DECORATION:
underline
}
.cmbcategory_list TD {
        FONT-WEIGHT: normal; FONT-SIZE: 12px; COLOR: #000000
}
.option_view {
        BORDER-RIGHT: medium none; PADDING-RIGHT: 1px; BORDER-
TOP: medium none; PADDING-LEFT: 1px; FONT-SIZE: 12px; PADDING-
BOTTOM: 1px; BORDER-LEFT: medium none; PADDING-TOP: 1px;
BORDER-BOTTOM: medium none
}
.contestanswer_title TD {
        BORDER-TOP: #cccccc 1px solid; FONT-WEIGHT: bold; FONT-SIZE:
15px; COLOR: #246fa8
}
.tbl_noborder TD {
        BORDER-RIGHT: 0px; BORDER-TOP: 0px; BORDER-LEFT: 0px;
BORDER-BOTTOM: 0px
}
.category_title DIV {
        FONT-WEIGHT: normal; FONT-SIZE: 18px; PADDING-BOTTOM:
20px; COLOR: #246fa8; LETTER-SPACING: 0px
}
.category_title A {
        FONT-WEIGHT: normal; FONT-SIZE: 18px; COLOR: #246fa8;
LETTER-SPACING: 0px; TEXT-DECORATION: none
}
textlarge_dark A:visited {
        FONT-WEIGHT: normal; FONT-SIZE: 18px; COLOR: #246fa8;
LETTER-SPACING: 0px; TEXT-DECORATION: none
}
.category_title A:hover {
        FONT-WEIGHT: normal; FONT-SIZE: 18px; COLOR: #000000;
LETTER-SPACING: 0px; TEXT-DECORATION: underline
}
.category_list TD {
```

```
	FONT-SIZE: 12px; COLOR: #000000; TEXT-ALIGN: left
}
.article_list DIV {
	FONT-WEIGHT: normal; FONT-SIZE: 20px; COLOR: #246fa8;
LETTER-SPACING: 1px
}
.category_title DIV {
	FONT-WEIGHT: normal; FONT-SIZE: 18px; PADDING-BOTTOM:
20px; COLOR: #246fa8; LETTER-SPACING: 0px
}
.category_title SPAN {
	FONT-WEIGHT: normal; FONT-SIZE: 18px; PADDING-BOTTOM:
20px; COLOR: #246fa8; LETTER-SPACING: 0px
}
.category_title A {
	FONT-WEIGHT: normal; FONT-SIZE: 18px; COLOR: #246fa8;
LETTER-SPACING: 0px; TEXT-DECORATION: none
}
textlarge_dark A:visited {
	FONT-WEIGHT: normal; FONT-SIZE: 18px; COLOR: #246fa8;
LETTER-SPACING: 0px; TEXT-DECORATION: none
}
.category_title A:hover {
	FONT-WEIGHT: normal; FONT-SIZE: 18px; COLOR: #000000;
LETTER-SPACING: 0px; TEXT-DECORATION: underline
}
.table_check {
	WIDTH: 100%
}
.table_check TD {
	BORDER-RIGHT: 0px; PADDING-RIGHT: 0px; BORDER-TOP: 0px;
PADDING-LEFT: 0px; BACKGROUND: none transparent scroll repeat 0%
0%; PADDING-BOTTOM: 0px; BORDER-LEFT: 0px; PADDING-TOP: 0px;
BORDER-BOTTOM: 0px
}
A.logoutlink {
	FONT-SIZE: 11px; COLOR: #ff0000; TEXT-DECORATION: underline
}
A.logoutlink:visited {
	 FONT-SIZE: 11px; COLOR: #ff0000; TEXT-DECORATION: underline
}
A.logoutlink:hover {
	 FONT-SIZE: 11px; COLOR: #000000; TEXT-DECORATION: none
```

```
}
.top_menu {
	FONT-WEIGHT: bold; FONT-SIZE: 11px; COLOR: #000000;
BACKGROUND-COLOR: #e2e2e2
}
.page_title {
	FONT-SIZE: 24px; COLOR: #ff0000
}
.table_search {
	MARGIN-BOTTOM: 5px; WIDTH: 100%
}
.table_search TD {
	FONT-SIZE: 12px
}
.footer {
	BORDER-RIGHT: #ffffff 2px solid; BORDER-TOP: #c5c5c5 1px solid;
FONT-SIZE: 11px; BORDER-LEFT: #c5c5c5 1px solid; COLOR: #333333;
HEIGHT: 23px; BACKGROUND-COLOR: #e2e2e2
}
.po_view {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: 0px; BORDER-
LEFT: #cccccc 1px solid; WIDTH: 800px; BORDER-BOTTOM: #cccccc 1px
solid; BORDER-COLLAPSE: collapse
}
.po_view TD {
	BORDER-RIGHT: 0px; PADDING-RIGHT: 3px; BORDER-TOP: 0px;
PADDING-LEFT: 3px; FONT-SIZE: 12px; PADDING-BOTTOM: 3px;
BORDER-LEFT: 0px; PADDING-TOP: 3px; BORDER-BOTTOM: 0px;
BORDER-COLLAPSE: collapse
}
.sales_view {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: 0px; BORDER-
LEFT: #cccccc 1px solid; WIDTH: 850px; BORDER-BOTTOM: #cccccc 1px
solid; BORDER-COLLAPSE: collapse
}
.sales_view TD {
	BORDER-RIGHT: 0px; PADDING-RIGHT: 3px; BORDER-TOP: 0px;
PADDING-LEFT: 3px; FONT-SIZE: 12px; PADDING-BOTTOM: 3px;
BORDER-LEFT: 0px; PADDING-TOP: 3px; BORDER-BOTTOM: 0px;
BORDER-COLLAPSE: collapse
}
.table_detail {
```

```
	BORDER-RIGHT: 0px; PADDING-RIGHT: 0px; BORDER-TOP:
#cccccc 1px solid; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px;
MARGIN: 0px; PADDING-TOP: 0px; BORDER-BOTTOM: 0px; BORDER-
COLLAPSE: collapse
}
.table_detail TD {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-LEFT: #cccccc 1px
solid; BORDER-BOTTOM: #cccccc 1px solid; BORDER-COLLAPSE:
collapse
}
.table_status {
	BORDER-RIGHT: 0px; PADDING-RIGHT: 0px; BORDER-TOP:
#cccccc 1px solid; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px;
MARGIN: 0px; PADDING-TOP: 0px; BORDER-BOTTOM: 0px; BORDER-
COLLAPSE: collapse
}
.table_status TD {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-LEFT: #cccccc 1px
solid; BORDER-BOTTOM: #cccccc 1px solid; BORDER-COLLAPSE:
collapse
}
.detail_heading {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: #cccccc 1px
solid; FONT-WEIGHT: normal; FONT-SIZE: 13px; BORDER-LEFT: #cccccc
1px solid; BORDER-BOTTOM: #cccccc 1px solid; BORDER-COLLAPSE:
collapse; BACKGROUND-COLOR: #e2e2e2; TEXT-ALIGN: center
}
.errormsg {
	FONT-WEIGHT: normal; FONT-SIZE: 15px; COLOR: #ff0000
}
.item_remarks {
	OVERFLOW: auto
}
.txtBg {
	BORDER-RIGHT: 0px; BORDER-TOP: 0px; BORDER-LEFT: 0px;
BORDER-BOTTOM: 0px; BACKGROUND-COLOR: #f3f3f3; TEXT-
ALIGN: right
}
.reportdetail_column {
	BORDER-RIGHT: #000000 1px solid
}
.report_page {
```

```
	BORDER-RIGHT: #000000 1px solid; PADDING-RIGHT: 10px;
BORDER-TOP: #000000 1px solid; PADDING-LEFT: 10px; BORDER-LEFT:
#000000 1px solid; WIDTH: 100%; BORDER-BOTTOM: #000000 1px solid
}
.report_page_addressbook {
	BORDER-RIGHT: #000000 1px solid; BORDER-TOP: #000000 1px
solid; BORDER-LEFT: #000000 1px solid; WIDTH: 100%; BORDER-
BOTTOM: #000000 1px solid
}
.report_page_addressbook TD {
	PADDING-RIGHT: 10px; PADDING-LEFT: 10px; FONT-SIZE: 12px;
PADDING-BOTTOM: 2px; VERTICAL-ALIGN: top; PADDING-TOP: 2px
}
.report_page TD {
	PADDING-RIGHT: 10px; PADDING-LEFT: 10px; FONT-SIZE: 12px;
PADDING-BOTTOM: 2px; VERTICAL-ALIGN: top; PADDING-TOP: 2px
}
.report_page_head TD {
	PADDING-RIGHT: 2px; PADDING-LEFT: 2px; FONT-WEIGHT:
bold; FONT-SIZE: 13px; PADDING-BOTTOM: 2px; VERTICAL-ALIGN:
top; PADDING-TOP: 2px
}
.report_title {
	FONT-WEIGHT: bold; FONT-SIZE: 15px
}
.report_delvery_title {
	FONT-WEIGHT: bold; FONT-SIZE: 20px; FONT-FAMILY: "Times
New Roman", Times, serif; TEXT-DECORATION: underline
}
.mrp {
	FONT-WEIGHT: bold; FONT-SIZE: 11px; COLOR: #3333cc; FONT-
FAMILY: Arial, Helvetica, sans-serif
}
.topborder {
	BORDER-TOP: #cccccc 1px solid
}
.table_action {
	BORDER-RIGHT: #e5e5e5 1px solid; BORDER-TOP: #e5e5e5 1px
solid; MARGIN-TOP: 3px; MARGIN-BOTTOM: 3px; BORDER-LEFT:
#e5e5e5 1px solid; WIDTH: 100%; BORDER-BOTTOM: #e5e5e5 1px solid;
BACKGROUND-COLOR: #eeeeee
}
.table_action TD {
```

```
	PADDING-RIGHT: 2px; PADDING-LEFT: 4px; FONT-SIZE: 12px;
PADDING-BOTTOM: 2px; PADDING-TOP: 2px
}
HR.dashed {
	BORDER-RIGHT: #cccccc 1px dashed; BORDER-TOP: #cccccc 1px
dashed; BORDER-LEFT: #cccccc 1px dashed; BORDER-BOTTOM: #cccccc
1px dashed
}
.sizeheading {
	BORDER-RIGHT: medium none; PADDING-RIGHT: 0px; BORDER-
TOP: medium none; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px;
BORDER-LEFT: medium none; PADDING-TOP: 0px; BORDER-BOTTOM:
medium none
}
.sizeheading TD {
	BORDER-RIGHT: medium none; PADDING-RIGHT: 0px; BORDER-
TOP: medium none; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px;
BORDER-LEFT: medium none; PADDING-TOP: 0px; BORDER-BOTTOM:
medium none
}
.partialcolor {
	BACKGROUND: #fbf8ee
}
.pendingcolor {
	BACKGROUND: #fff3f2
}
.fullfillcolor {
	BACKGROUND: #ffffff
}
.partialtext {
	COLOR: #6fc8c4
}
.pendingtext {
	COLOR: #cc9999
}
.fullfilltext {
	COLOR: #5c8fcd
}
.payment_view {
	BORDER-RIGHT: #cccccc 1px solid; BORDER-TOP: 0px; BORDER-
LEFT: #cccccc 1px solid; WIDTH: 650px; BORDER-BOTTOM: #cccccc 1px
solid; BORDER-COLLAPSE: collapse
}
```

```
.payment_view TD {
	BORDER-RIGHT: #cccccc 1px solid; PADDING-RIGHT: 3px;
BORDER-TOP: 0px; PADDING-LEFT: 3px; FONT-SIZE: 12px; PADDING-
BOTTOM: 3px; BORDER-LEFT: #cccccc 1px solid; PADDING-TOP: 3px;
BORDER-BOTTOM: #cccccc 1px solid; BORDER-COLLAPSE: collapse
}
.payment_view TD {
	BORDER-RIGHT: 0px; BORDER-TOP: 0px; BORDER-LEFT: 0px;
BORDER-BOTTOM: 0px
}
#balanceDiv {
	FONT-WEIGHT: bold; FONT-SIZE: 12px; FONT-FAMILY: Arial,
Helvetica, sans-serif
}
.receive_view {
	PADDING-RIGHT: 3px; PADDING-LEFT: 3px; PADDING-BOTTOM:
3px; PADDING-TOP: 3px
}
.territory_detail {
	PADDING-RIGHT: 0px; PADDING-LEFT: 0px; FONT-WEIGHT:
bold; FONT-SIZE: 14px; PADDING-BOTTOM: 0px; COLOR: #1667a8;
PADDING-TOP: 0px; LETTER-SPACING: 0px; TEXT-ALIGN: right
}
.territory_detail SPAN {
	FONT-WEIGHT: normal; FONT-SIZE: 12px
}
.detail_list {
	 BORDER-TOP: 0px; WIDTH: 100%; BORDER-COLLAPSE: collapse
}
.detail_list TD {
	BORDER-RIGHT: #e6e3df 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: 0px; PADDING-LEFT: 5px; FONT-SIZE: 12px; PADDING-
BOTTOM: 1px; BORDER-LEFT: #e6e3df 1px solid; PADDING-TOP: 1px;
BORDER-BOTTOM: #e6e3df 1px solid; BORDER-COLLAPSE: collapse
}
.colorseprator {
	FONT-WEIGHT: bold; FONT-SIZE: 12px; COLOR: #ff0000; TEXT-
ALIGN: left
}
.selected {
	CURSOR: pointer; BACKGROUND-COLOR: #cfcffc
}
.tooltipbox {
```

```
	BORDER-RIGHT: #999999 1px solid; PADDING-RIGHT: 5px;
BORDER-TOP: #999999 1px solid; PADDING-LEFT: 5px; FONT-SIZE:
12px; PADDING-BOTTOM: 5px; BORDER-LEFT: #999999 1px solid;
PADDING-TOP: 5px; BORDER-BOTTOM: #999999 1px solid
}
.tooltipheader {
	BACKGROUND: #dddddd
}
.topbtn {
	BORDER-RIGHT: #cccccc 1px solid; PADDING-RIGHT: 0px;
BORDER-TOP: #cccccc 1px solid; PADDING-LEFT: 5px; PADDING-
BOTTOM: 2px; BORDER-LEFT: #cccccc 1px solid; WIDTH: 100%;
PADDING-TOP: 2px; BORDER-BOTTOM: #cccccc 1px solid;
BACKGROUND-COLOR: #eaeaea
}
.searchbar {
	BORDER-RIGHT: #cccccc 1px solid; PADDING-RIGHT: 0px;
BORDER-TOP: #cccccc 1px solid; PADDING-LEFT: 5px; PADDING-
BOTTOM: 2px; BORDER-LEFT: #cccccc 1px solid; WIDTH: 100%;
PADDING-TOP: 2px; BORDER-BOTTOM: #cccccc 1px solid;
BACKGROUND-COLOR: #eaeaea
}
.searchbar TD {
	FONT-SIZE: 13px
}
.date_title {
	FONT-SIZE: 12px
}
.podetail_list {
	BORDER-TOP: 0px; BORDER-COLLAPSE: collapse
}
.podetail_list TD {
	BORDER-RIGHT: #ffffff 1px solid; PADDING-RIGHT: 1px;
BORDER-TOP: 0px; PADDING-LEFT: 5px; FONT-SIZE: 12px; PADDING-
BOTTOM: 1px; BORDER-LEFT: #ffffff 1px solid; PADDING-TOP: 1px;
BORDER-BOTTOM: #ffffff 1px solid; BORDER-COLLAPSE: collapse
}


package prompt.beans;
public class Enquiry {
	int enquiryId;
	String name, mailId,mobileNo,enquiryDate;
```

```
int enquiryMode,courseId,awarenessSource,status;
public int getAwarenessSource() {
	return awarenessSource;
}
public String getAwarenessSourceName()
{String sourceName;
	switch (enquiryMode)
	{
	case PromptConstants.AwarenessSource_Friend:
		sourceName="Friend";
		break;
	case PromptConstants.AwarenessSource_NewsPaper:
		sourceName="NewsPaper";
		break;
	case PromptConstants.AwarenessSource_Ad:
		sourceName="Ad";
		break;

	default:
		sourceName="Other";

	}
	return sourceName;
}

public void setAwarenessSource(int awarenessSource) {
	this.awarenessSource = awarenessSource;
}
public String getEnquiryDate() {
	return enquiryDate;
}
public void setEnquiryDate(String enquiryDate) {
	this.enquiryDate = enquiryDate;
	System.out.println("set===="+enquiryDate);
}
public int getEnquiryId() {
	System.out.println("get===="+enquiryDate);
	return enquiryId;
}
public void setEnquiryId(int enquiryId) {
	this.enquiryId = enquiryId;
}
public String getMailId() {
```

```
        return mailId;
}
public void setMailId(String mailId) {
        this.mailId = mailId;
}
public String getMobileNo() {
        return mobileNo;
}
public void setMobileNo(String mobileNo) {
        this.mobileNo = mobileNo;
}
public int getEnquiryMode() {
        return enquiryMode;
}
public void setEnquiryMode(int mode) {
        this.enquiryMode = mode;
}
public String getModeName()
{String modeName;
        switch (enquiryMode)
        {
        case PromptConstants.Mode_InPerson:
                modeName="Personal";
                break;
        case PromptConstants.Mode_Email:
                modeName="Email";
                break;
        default:
                modeName="Telephonic";

        }
        return modeName;
}
public String getName() {
        return name;
}
public void setName(String name) {
        this.name = name;
}
public int getCourseId() {
        return courseId;
}
public void setCourseId(int technologyId) {
```

```
		this.courseId = technologyId;
	}
	public int getStatus() {
		return status;
	}
	public void setStatus(int status) {
		this.status = status;
	}

}


package prompt.beans;

public class Login {
private String name,loginId,password,role;
private int user_id,techId;
public String getLoginId() {
	return loginId;
}
public void setLoginId(String name) {
	this.loginId = name;
}
public String getPassword() {
	return password;
}
public void setPassword(String pass) {
	this.password = pass;
}
public String getRole() {
	return role;
}
public void setRole(String role) {
	this.role = role;
}
public int getUser_id() {
	return user_id;
}
public void setUser_id(int user_id) {
	this.user_id = user_id;
}
public String getName() {
	return name;
```

```
}
public void setName(String name) {
	this.name = name;
}
public int getTechId() {
	return techId;
}
public void setTechId(int techId) {
	this.techId = techId;
}

}

package prompt.beans;
public class PromptConstants {
public static final int Mode_InPerson=1;
public static final int Mode_Email=2;
public static final int Mode_Telephonic=3;
public static final int Status_ToBeAttended=1;
public static final int Status_Attended=2;
public static final int Prospect_Converted=1;
public static final int Prospect_ToBeFollowed=2;
public static final int Prospect_Dead=3;
public static final int AwarenessSource_Friend=1;
public static final int AwarenessSource_NewsPaper=2;
public static final int AwarenessSource_Ad=3;
public static final int AwarenessSource_Other=4;

}

package prompt.beans;

public class Technology {
private int  technologyId;
private String name,description;
public String getDescription() {
	return description;
}
public void setDescription(String description) {
	this.description = description;
}
public String getName() {
	return name;
```

```
}
public void setName(String name) {
	this.name = name;
}
public int getTechnologyId() {
	return technologyId;
}
public void setTechnologyId(int technologyId) {
	this.technologyId = technologyId;
}
}

package prompt.beans;

public class User {
	int userId,techId;
	String name, address, mobileNo,
mailId,designation,loginId,password,role;
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
	}
	public String getDesignation() {
		return designation;
	}
	public void setDesignation(String designation) {
		this.designation = designation;
	}
	public String getMailId() {
		return mailId;
	}
	public void setMailId(String mailId) {
		this.mailId = mailId;
	}
	public String getMobileNo() {
		return mobileNo;
	}
	public void setMobileNo(String mobileNo) {
		this.mobileNo = mobileNo;
	}
	public String getName() {
```

```
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public int getUserId() {
		return userId;
	}
	public void setUserId(int userId) {
		this.userId = userId;
	}
	public String getLoginId() {
		return loginId;
	}
	public void setLoginId(String loginId) {
		this.loginId = loginId;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getRole() {
		return role;
	}
	public void setRole(String role) {
		this.role = role;
	}
	public int getTechId() {
		return techId;
	}
	public void setTechId(int techId) {
		this.techId = techId;
	}


}
```

```
package prompt.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import prompt.beans.Course;

public class CourseDao extends PromptDao
{
	static int idCounter=40001;
	public CourseDao()
	{
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("select max
(courseId) from course");
			ResultSet rset=stmt.executeQuery();
			if (rset.next())
			{
				int c=rset.getInt(1);
				if (c>idCounter)
				idCounter=c;
			}
			con.close();
			System.out.println(idCounter);
		}catch(Exception e)
		{
			System.out.println(e);

		}

	}
	public int insertCourse (Course course ){
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("insert into
course values(?,?,?,?,?)");
```

```
                idCounter++;
                stmt.setInt(1,idCounter);
                stmt.setInt(2,course.getTechnologyId());
                stmt.setString(3,course.getName());
                stmt.setString(4,course.getDescription());
                stmt.setInt(5,course.getFee());
                stmt.executeUpdate();
                con.close();

            }catch(Exception e)
            {
                System.out.println(e);

            }
            return idCounter;
        }

        public void updateCourse(Course course)
        {

            try
            {
                Connection con=getConnection();
                PreparedStatement stmt=con.prepareStatement("update
course set name=?,description=?,fee=? where courseId=?");
                stmt.setString(1,course.getName());
                stmt.setString(2,course.getDescription());
                stmt.setInt(3,course.getFee());
                stmt.setInt(4,course.getCourseId());
                stmt.executeUpdate();
                con.close();

            }catch(Exception e)
            {
                System.out.println(e);

            }
            }
        public Map getNameMap  (){
            HashMap map=new HashMap();
            try
            {
                 Connection con=this.getConnection();
```

```
                PreparedStatement stmt=con.prepareStatement("select
courseId,name from course");
                ResultSet rset=stmt.executeQuery();
                while (rset.next())
                {


        map.put(String.valueOf(rset.getInt(1)),rset.getString(2));

                }
                con.close();

        }catch(Exception e)
        {
                System.out.println(e);

        }
        return map;

    }

    public ArrayList findAllCourse (){
        ArrayList list=new ArrayList();
        try
        {
                Connection con=this.getConnection();
                PreparedStatement stmt=con.prepareStatement("select *
from course");
                ResultSet rset=stmt.executeQuery();
                while (rset.next())
                {
                        Course course=new Course();
                        course.setCourseId(rset.getInt(1));
                        course.setTechnologyId(rset.getInt(2));
                        course.setName(rset.getString(3));
                        course.setDescription(rset.getString(4));
                        course.setFee(rset.getInt(5));

                        list.add(course);

                }
                con.close();
```

```
		}catch(Exception e)
		{
			System.out.println(e);

		}
		return list;

	}
	public void deleteCourse(int courseId)
	{
		try
		{
			Connection con=this.getConnection();
			PreparedStatement stmt=con.prepareStatement("delete from
course where courseId=?");
			stmt.setInt(1,courseId);
			stmt.executeUpdate();
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
	}

public Course findCourse(int courseId)
{ Course course=null;
	try
	{
		Connection con=this.getConnection();
		PreparedStatement stmt=con.prepareStatement("select * from
course where courseId=?");
		stmt.setInt(1,courseId);
		ResultSet rset=stmt.executeQuery();
		if (rset.next())
		{
		   course=new Course();
		   course.setCourseId(rset.getInt(1));
			course.setTechnologyId(rset.getInt(2));
			course.setName(rset.getString(3));
			course.setDescription(rset.getString(4));
			course.setFee(rset.getInt(5));
```

```
                }
                con.close();

        }catch(Exception e)
        {
                System.out.println(e);

        }
        return course;
}

}
```

```
package prompt.dao;
import java.util.HashMap;
public class DaoFactory {
private static HashMap daoMap=null;
static
{
daoMap=new HashMap();
}
public static PromptDao getDao (String className)
{
System.out.println("searcing "+className +" in map");
        PromptDao dao=(PromptDao) daoMap.get(className);
if (dao==null)
{
System.out.println("Not Found, creating and storing in Map...");
try{
        Class c=Class.forName("prompt.dao."+className);
        dao=(PromptDao)c.newInstance();
        daoMap.put(className,dao);
}catch(Exception e)
{
System.out.println(e);
}
}
else
        System.out.println("Found, returning...");
return dao;
```

```
}
}

package prompt.dao;
import java.sql.*;
import prompt.beans.AttendedEnquiry;
import prompt.beans.Enquiry;
import prompt.beans.PromptConstants;
import java.util.*;

public class EnquiryDao extends PromptDao {
	static int idCounter=20001;
	static int attendingId=5000;
	public EnquiryDao()
	{
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("select max
(enquiryId) from enquiry");
			ResultSet rset=stmt.executeQuery();
			if (rset.next())
			{	int c=rset.getInt(1);
			if (c>idCounter)
			idCounter=c;
			}
			PreparedStatement stmt1=con.prepareStatement("select max
(attendingId) from attendedEnquiry");
			 rset=stmt1.executeQuery();
			if (rset.next())
			{	int c=rset.getInt(1);
			if (c>attendingId)
			attendingId=c;
			}

				con.close();
			System.out.println(idCounter);
		}catch(Exception e)
		{
			System.out.println(e);

		}
```

```
	}
	public int attendEnquiry (AttendedEnquiry enquiry ){
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("insert into
attendedEnquiry values(?,?,?,?,?,?)");
			attendingId++;
			stmt.setInt(1,attendingId);
			stmt.setInt(2,enquiry.getEnquiryId());
			stmt.setInt(3,enquiry.getAttendantId());
			stmt.setString(4,enquiry.getAttendingDate());
			stmt.setInt(5,enquiry.getProspect());
			stmt.setString(6,enquiry.getComments());
			stmt.executeUpdate();
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		return idCounter;
	}

public int insertEnquiry (Enquiry enquiry ){
	try
	{
		Connection con=getConnection();
		PreparedStatement stmt=con.prepareStatement("insert into enquiry
values(?,?,?,?,?,?,?,?,?)");
		idCounter++;
		stmt.setInt(1,idCounter);
		stmt.setString(2,enquiry.getName());
		stmt.setString(3,enquiry.getMailId());
		stmt.setString(4,enquiry.getMobileNo());
		stmt.setString(5,enquiry.getEnquiryDate());
		stmt.setInt(6,enquiry.getAwarenessSource());
		stmt.setInt(7,enquiry.getCourseId());
		stmt.setInt(8,enquiry.getEnquiryMode());
		stmt.setInt(9,PromptConstants.Status_ToBeAttended);
		stmt.executeUpdate();
		con.close();
```

```
        }catch(Exception e)
        {
                System.out.println(e);

        }
        return idCounter;
}
public void deleteEnquiry (int enquiryId){
        try
        {
                Connection con=this.getConnection();
                PreparedStatement stmt=con.prepareStatement("delete from
enquiry where enquiryId=?");
                stmt.setInt(1,enquiryId);
                stmt.executeUpdate();
                con.close();

        }catch(Exception e)
        {
                System.out.println(e);

        }

}
public Enquiry findEnquiry(int id){
        Enquiry enquiry=null;
        try
        {
                Connection con=this.getConnection();
                PreparedStatement stmt=con.prepareStatement("select * from
enquiry where enquiryId=?");
                stmt.setInt(1,id);
                ResultSet rset=stmt.executeQuery();
                if (rset.next())
                {
                        enquiry=new Enquiry();
                        enquiry.setEnquiryId(rset.getInt(1));
                        enquiry.setName(rset.getString(2));
                        enquiry.setMailId(rset.getString(3));
                        enquiry.setMobileNo(rset.getString(4));
                        enquiry.setEnquiryDate(rset.getString(5));
                        enquiry.setAwarenessSource(rset.getInt(6));
```

```
                enquiry.setCourseId(rset.getInt(7));
                enquiry.setEnquiryMode(rset.getInt(8));
                enquiry.setStatus(rset.getInt(9));




            }
            con.close();

        }catch(Exception e)
        {
            System.out.println(e);

        }
        return enquiry;

}
public ArrayList findAllEnquiry (){
        ArrayList list=new ArrayList();
        try
        {
            Connection con=this.getConnection();
            PreparedStatement stmt=con.prepareStatement("select * from
enquiry ");
            ResultSet rset=stmt.executeQuery();
            while (rset.next())
            {
                Enquiry enquiry=new Enquiry();
                enquiry.setEnquiryId(rset.getInt(1));
                enquiry.setName(rset.getString(2));
                enquiry.setMailId(rset.getString(3));
                enquiry.setMobileNo(rset.getString(4));
                enquiry.setEnquiryDate(rset.getString(5));
                enquiry.setAwarenessSource(rset.getInt(6));
                enquiry.setCourseId(rset.getInt(7));
                enquiry.setEnquiryMode(rset.getInt(8));
                enquiry.setStatus(rset.getInt(9));

                list.add(enquiry);

            }
```

```
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		return list;

}
public ArrayList getToBeAttendedEnquires (){
		ArrayList list=new ArrayList();
		try
		{
			Connection con=this.getConnection();
			PreparedStatement stmt=con.prepareStatement("select * from
enquiry where status=?");
			stmt.setInt(1,PromptConstants.Status_ToBeAttended);
			ResultSet rset=stmt.executeQuery();
			while (rset.next())
			{
				Enquiry enquiry=new Enquiry();
				enquiry.setEnquiryId(rset.getInt(1));
				enquiry.setName(rset.getString(2));
				enquiry.setMailId(rset.getString(3));
				enquiry.setMobileNo(rset.getString(4));
				enquiry.setEnquiryDate(rset.getString(5));
				enquiry.setAwarenessSource(rset.getInt(6));
				enquiry.setCourseId(rset.getInt(7));
				enquiry.setEnquiryMode(rset.getInt(8));
				enquiry.setStatus(rset.getInt(9));

				list.add(enquiry);

			}
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		return list;
```

```
}

}
```

```
package prompt.dao;
import java.sql.*;
public abstract class PromptDao {
	public Connection getConnection()throws Exception
	{
		try{
			Class.forName("com.mysql.jdbc.Driver");
			Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/java","prompt
biz","promptbiz");
			   return con;
		}catch(Exception e)
		{
		throw (e);
		}

	}
}

package  prompt.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import  java.util.ArrayList;
import  java.util.HashMap;
import java.util.Map;

import prompt.beans.Technology;

public class TechDao extends PromptDao
{
	static int idCounter=30001;
	public TechDao()
	{
```

```
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("select max
(technologyId) from technology");
			ResultSet rset=stmt.executeQuery();
			if (rset.next())
			{
				int c=rset.getInt(1);
				if (c>idCounter)
				idCounter=c;
			}
			con.close();
			System.out.println(idCounter);
		}catch(Exception e)
		{
			System.out.println(e);

		}

	}
	public int insertTech (Technology tech ){
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("insert into
technology values(?,?,?)");
			idCounter++;
			stmt.setInt(1,idCounter);
			stmt.setString(2,tech.getName());
			stmt.setString(3,tech.getDescription());

			stmt.executeUpdate();
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		return idCounter;
	}
```

```
	public void updateTech(Technology tech)
	{

		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("update
technology set name=?,description=? where technologyId=?");
			stmt.setString(1,tech.getName());
			stmt.setString(2,tech.getDescription());
			stmt.setInt(3,tech.getTechnologyId());
			stmt.executeUpdate();
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		}
	public Map getNameMap  (){
		HashMap map=new HashMap();
		try
		{
			Connection con=this.getConnection();
			PreparedStatement stmt=con.prepareStatement("select *
from technology");
			ResultSet rset=stmt.executeQuery();
			while (rset.next())
			{


	map.put(String.valueOf(rset.getInt(1)),rset.getString(2));

			}
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		return map;
```

```
	}

	public ArrayList findAllTech (){
		ArrayList list=new ArrayList();
		try
		{
			Connection con=this.getConnection();
			PreparedStatement stmt=con.prepareStatement("select *
from technology");
			ResultSet rset=stmt.executeQuery();
			while (rset.next())
			{
				Technology tech=new Technology();
				tech.setTechnologyId(rset.getInt(1));
				tech.setName(rset.getString(2));
				tech.setDescription(rset.getString(3));
				list.add(tech);

			}
			con.close();

		}catch(Exception e)
		{
			System.out.println(e);

		}
		return list;

	}
	public void deleteTech(int techId)
	{
		try
		{
			Connection con=this.getConnection();
			PreparedStatement stmt=con.prepareStatement("delete from
technology where technologyId=?");
			stmt.setInt(1,techId);
			stmt.executeUpdate();
			con.close();

		}catch(Exception e)
		{
```

```
                System.out.println(e);

            }
        }

public Technology findTech(int techId)
{Technology tech=null;
        try
        {
            Connection con=this.getConnection();
            PreparedStatement stmt=con.prepareStatement("select * from
technology where technologyId=?");
            stmt.setInt(1,techId);
            ResultSet rset=stmt.executeQuery();
            if (rset.next())
            {
                tech=new Technology();
                tech.setTechnologyId(rset.getInt(1));
                tech.setName(rset.getString(2));
                tech.setDescription(rset.getString(3));
                System.out.println("technolgy found returning...");

            }
            con.close();

        }catch(Exception e)
        {
            System.out.println(e);

        }
        return tech;
}
}
```

```
package prompt.dao;
import java.sql.*;
import prompt.beans.Login;
import prompt.beans.User;
import java.util.*;
```

```
public class UserDao extends PromptDao {
	static int idCounter=10001;
	public UserDao()
	{
		try
		{
			Connection con=getConnection();
			PreparedStatement stmt=con.prepareStatement("select max
(user_id) from promptUsers");
			ResultSet rset=stmt.executeQuery();
			if (rset.next())
			{	int c=rset.getInt(1);
			if (c>idCounter)
			idCounter=c;
			}
				con.close();
			System.out.println(idCounter);
		}catch(Exception e)
		{
			System.out.println(e);

		}

	}

public int insertUser (User u ){
	try
	{
		Connection con=getConnection();
		PreparedStatement stmt=con.prepareStatement("insert into
promptUsers values(?,?,?,?,?,?,?,?,?,?)");
		idCounter++;
		stmt.setInt(1,idCounter);
		stmt.setString(2,u.getName());
		stmt.setString(3,u.getAddress());
		stmt.setString(4,u.getMobileNo());
		stmt.setString(5,u.getMailId());
		stmt.setString(6,u.getDesignation());
		stmt.setString(7,u.getLoginId());
		stmt.setString(8,u.getPassword());
		stmt.setString(9,u.getRole());
		stmt.setInt(10,u.getTechId());
```

```
                stmt.executeUpdate();
                con.close();

        }catch(Exception e)
        {
                System.out.println(e);

        }
        return idCounter;
}
public Login autenticate(String loginId,String password)
{
Login login=null;
try
{
        Connection con=this.getConnection();
        PreparedStatement stmt=con.prepareStatement("select user_id,
name,technologyId,role from promptUsers where loginId=? and password=?");
        stmt.setString(1,loginId);
        stmt.setString(2,password);
        ResultSet rset=stmt.executeQuery();
        if (rset.next())
        {
                login=new Login();
                login.setUser_id(rset.getInt(1));
                login.setLoginId(loginId);
                login.setPassword(password);
                login.setName(rset.getString(2));
                login.setTechId(rset.getInt(3));
                login.setRole(rset.getString(4));


        }
        con.close();

}catch(Exception e)
{
        System.out.println(e);

}
return login;
}
public boolean changePassword(int userId,String cPass,String nPass)
```

```
{boolean flag=false;
	try
	{
		Connection con=this.getConnection();
		PreparedStatement stmt=con.prepareStatement("update
promptUsers set password=? where user_id=? and password=?");
		stmt.setString(1,nPass);
		stmt.setInt(2,userId);
		stmt.setString(3,cPass);
		if (stmt.executeUpdate()==1)
		flag=true;

		con.close();

	}catch(Exception e)
	{
		System.out.println(e);

	}

return flag;
}

public void updateUser(User user)
{
	try
	{
		Connection con=this.getConnection();
		PreparedStatement stmt=con.prepareStatement("update
promptUsers set name=?, designation=?, role=?,
address=?,mailId=?,mobileNo=? where user_id=?");
		stmt.setString(1,user.getName());
		stmt.setString(2,user.getDesignation());
		stmt.setString(3,user.getRole());
		stmt.setString(4,user.getAddress());
		stmt.setString(5,user.getMailId());
		stmt.setString(6,user.getMobileNo());
		stmt.setInt(7,user.getUserId());
		stmt.executeUpdate();
		con.close();

	}catch(Exception e)
	{
```

```
                System.out.println(e);

        }


}
public void changeProfile(User user)
{
        try
        {
                Connection con=this.getConnection();
                PreparedStatement stmt=con.prepareStatement("update
promptUsers set name=?, address=?,mailId=?,mobileNo=? where user_id=?");
                stmt.setString(1,user.getName());
                stmt.setString(2,user.getAddress());
                stmt.setString(3,user.getMailId());
                stmt.setString(4,user.getMobileNo());
                stmt.setInt(5,user.getUserId());
                stmt.executeUpdate();
                con.close();

        }catch(Exception e)
        {
                System.out.println(e);

        }


}


public ArrayList findAllUser (){
        ArrayList list=new ArrayList();
        try
        {
                Connection con=this.getConnection();
                PreparedStatement stmt=con.prepareStatement("select * from
promptUsers ");
                ResultSet rset=stmt.executeQuery();
                while (rset.next())
                {
                        User u=new User();
                        u.setUserId(rset.getInt(1));
```

```
                u.setName(rset.getString(2));
                u.setAddress(rset.getString(3));
                u.setMobileNo(rset.getString(4));
                u.setMailId(rset.getString(5));
                u.setDesignation(rset.getString(6));
                u.setLoginId(rset.getString(7));
                u.setPassword(rset.getString(8));
                u.setRole(rset.getString(9));
                u.setTechId(rset.getInt(10));

                list.add(u);

            }
            con.close();

        }catch(Exception e)
        {
            System.out.println(e);

        }
        return list;

}


package prompt.helper;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Map;
import prompt.beans.Enquiry;
import prompt.dao.CourseDao;
import prompt.dao.DaoFactory;
import prompt.dao.EnquiryDao;
import prompt.pagination.Paginate;

public class ToBeAttendedPaginator {
    public static Paginate getPaginate(int rowPerPage)
    {
        CourseDao cDao=(CourseDao)DaoFactory.getDao("CourseDao");
        EnquiryDao
eDao=(EnquiryDao)DaoFactory.getDao("EnquiryDao");
        Map map=cDao.getNameMap();
        ArrayList list=eDao.getToBeAttendedEnquires();
```

```
		Iterator itr=list.iterator();
		int size=list.size();
		System.out.println(size+" records found.");
		String [][]data=new String[size][5];
		for (int i=0;i<size;i++)
		{
			Enquiry enquiry=(Enquiry)itr.next();
			data[i][0]=enquiry.getName();
			data[i][1]=enquiry.getEnquiryDate();
			data[i][2]=enquiry.getModeName();
			data[i][3]=
(String)map.get(String.valueOf(enquiry.getCourseId()));
			data[i][4]="<a
href=index.jsp?p=doAttend&enquiryId="+enquiry.getEnquiryId()+"> attend
</a>";
		}
		String []header={"Name","Date","Mode","course","Action"};
		String caption="To Be Attended Enquiries:";
		Paginate p=new Paginate(header,data,rowPerPage,caption);
		System.out.println("Paginate object created.");
		return p;}}
package prompt.helper;
import java.util.ArrayList;
import java.util.Iterator;
import prompt.beans.User;
import prompt.dao.*;
import prompt.pagination.Paginate;
public class UserPaginator {
public static Paginate getPaginate(int rowPerPage)
{
	UserDao uDao=(UserDao)DaoFactory.getDao("UserDao");
	ArrayList list=uDao.findAllUser();
	Iterator itr=list.iterator();
	int size=list.size();
	System.out.println(size+" records found.");
	String [][]data=new String[size][6];
	for (int i=0;i<size;i++)
	{
		User user=(User)itr.next();
		data[i][0]=user.getName();
		data[i][1]=user.getMailId();
		data[i][2]=user.getDesignation();
		data[i][3]=user.getRole();
```

```
                data[i][4]="<a
href=index.jsp?p=editUser&userId="+user.getUserId()+"> edit </a>";
                data[i][5]="<a
href=index.jsp?p=deleteUser&userId="+user.getUserId()+"> delete </a>";
                }
        String []header={"User Name","Mail
Id","Designation","Role","Action","Action"};

        String caption="User Details";
        Paginate p=new Paginate(header,data,rowPerPage,caption);
        System.out.println("Paginate object created.");
        return p;

}
}
```

```
package prompt.listener;
import prompt.helper.*;
import javax.servlet.*;
public class CtxListener implements ServletContextListener {

        public void contextDestroyed(ServletContextEvent c) {

        }

        public void contextInitialized(ServletContextEvent c) {
                ServletContext ctx=c.getServletContext();
                String param=ctx.getInitParameter("createTables");
                if (param.equals("yes"))
                {
                String p=ctx.getRealPath("WEB-
INF/classes/prompt/sql/promptBizTables.sql");
                        DBInitializer db=new DBInitializer();
                        db.createTables(p);
                }
        }

}
```

```
package prompt.pagination;
import javax.servlet.jsp.JspWriter;
public class Paginate {
private String[][] data;
private MetaData metaData;
private JspWriter out;
private String nextUrl,previousUrl;

public Paginate(String header[],String[][]data,int rowPerPage,String caption)
{
metaData=new MetaData(header,rowPerPage,caption);
this.data=data;
metaData.setTotalRows(data.length);
}

public void setOut(JspWriter out) {
        this.out = out;
}
public void reset()
{
metaData.setCurrentRow(0);
metaData.setPreviousRow(0);
}
public void setNext()
{

        metaData.setNext();

}
public void setPrevious()
{
metaData.setPrevious();
}
public String getNextUrl() {
        return nextUrl;
}

public void setNextUrl(String nextUrl) {
        this.nextUrl = nextUrl;
}
```

```
public String getPreviousUrl() {
	return previousUrl;
}

public void setPreviousUrl(String previousUrl) {
	this.previousUrl = previousUrl;
}

public void displayPage()
{
displayHeader();
displayData();
displayLinks();
}
private void displayHeader()
{
	try{
out.print("<TABLE class=table_page cellSpacing=0
align=center><TBODY><TR><TD width=599>  </TD></TR><TR
class=textlarge_dark>");
out.println("<TD>"+metaData.getCaption()+"</TD></TR>");
out.println("<TR><TD> </TD></TR><TR><TD><br><br><table
class=table_view cellSpacing=0><TBODY>");
out.println("<TR class=table_head>");
int c=metaData.getFieldCount();
for (int i=0;i<c;i++)
{
out.println("<TD colSpan=2>");
out.println(metaData.getFieldName(i));
out.println("</TD>");
}
out.println("</TR>");

	}catch(Exception e)
	{
		System.out.println(e);
	}
}
private void displayData()
{
try{
int r=metaData.getCurrentRow();
System.out.println("Current row count "+r);
```

```
int p=metaData.getRowPerPage();
System.out.println("row per page "+p);

int t=metaData.getTotalRows();
System.out.println("total rows "+t);

int rowCount=((r+p)<t)?p:(t-r);
System.out.println("row to be displayed in this pages "+rowCount);

int columnCount=metaData.getFieldCount();
for (int i=0;i<rowCount;i++,r++)
{
out.println("<tr>") ;
for (int c=0;c<columnCount;c++)
{
        out.println("<TD colSpan=2>");
        out.println(data[r][c]);
        out.println("</TD>");

}
out.print("</tr>");
}
out.println("</table><P></P><P></P></form></TD></TR></TBODY></TAB
LE>");
        metaData.setPreviousRow(r);
}catch(Exception e)
{
        System.out.println(e);
}
}
private void displayLinks()
{
try
{
        out.println("<table><tr>");



if (metaData.isPreviousPage())
        {

        out.println("<td><a href="+getPreviousUrl()+"> previous</a></td>");
        }
```

```
        if (metaData.isNextPage())
        {
        out.println("<td><a href="+getNextUrl()+">next</a></td>");
        }
        out.println("</table>");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

```
newUser=jspPages/user.jsp
enquiry=jspPages/newEnquiry.jsp
newEnquiry=jspPages/enquiry.jsp
addUser=jspPages/addUser.jsp
login=jspPages/login.jsp
logout=jspPages/logout.jsp
deleteUser=jspPages/deleteUser.jsp
home=includes/home.html
Manager=jspPages/managerHeader.jsp
Admin=jspPages/adminHeader.jsp
Accountant=jspPages/accountantHeader.jsp
Consultant=jspPages/consultantHeader.jsp
Coordinator=jspPages/coordinatorHeader.jsp
Counselor=jspPages/counselorHeader.jsp
newTechnology=includes/newTech.html
newTech=jspPages/newTech.jsp
viewTechnologies=jspPages/viewTech.jsp
editTech=jspPages/editTech.jsp
editTechnology=jspPages/editTechnology.jsp
deleteTech=jspPages/deleteTech.jsp
viewUsers=jspPages/viewUsers.jsp
editUser=jspPages/editUser.jsp
editDetails=jspPages/editDetails.jsp
deleteUser=jspPages/deleteUser.jsp
changePassword=includes/changePassword.html
changePass=jspPages/changePassword.jsp
editProfile=jspPages/editProfile.jsp
changeProfile=jspPages/changeProfile.jsp
```

```
newCourse=jspPages/newCourse.jsp
addCourse=jspPages/addCourse.jsp
viewCourses=jspPages/viewCourses.jsp
editCourse=jspPages/editCourse.jsp
changeCourse=jspPages/changeCourse.jsp
deleteCourse=jspPages/deleteCourse.jsp
attendEnquiry=jspPages/toBeAttended.jsp
doAttend=jspPages/attendEnquiry.jsp
enquiryAttended=jspPages/enquiryAttended.jsp
```

```
<web-app>
        <context-param>
        <param-name>createTables</param-name>
        <param-value>yes</param-value>
        </context-param>
        <listener>
        <listener-class>prompt.listener.CtxListener</listener-class>
        </listener>
<welcome-file-list>
<welcome-file>promptHome.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

❖ CODE EFFICIENCY

Every step has been considred in order to achieve the code efficiency J2EE has been used in order to achieve the high degree of optimality of code.It is an object oriented GUI tool that provides the facility of the rocedural programming in MY SQL as well as supports non-procedural like SQL.SQL drastically reduces the procedural steps to perform a task.SQL meets only the target(what has to be done)to be specified and it does not need the steps (how) to be followed to achiever the target.The data retrieved is done by SQL engine of the RDBMS,which is very efficient, and it increas4s the efficiency of the application as well.As far as user interface design of the application is concerned,object oriented paradigm has been applied which reduces the lengthy codification and achieve the inheritance,which further enhance the code efficiency.

❖ VALIDATION CHECKS

As discussed earlier the application has been broken in two tiers like Back End MY SQL and the Front End user interface in J2EE.In this architecture every type of validation check and constraints have been applied at the database level itself as per the code rules.Apart from this every type of data security rules have also been applied by creating appropriate synonyms and user roles.Access to the user has been provided in controlled way keeping the view at users level and their reqirements.Due to which user access is restricted and the security has enforced.

Another level of validation checks has been applied at data entry form level also in the user interface.These two levels of validation checks enforece high level of data validity and only valid data can be entered and processed.The validation rules cannot be bypassed by any means.

For example: -In Student entry Form,I have used a list item for student type field so that onlyeither'permanent'or 'ordinary'data value can be inserted into the database.It prevents the user from inserting any invalid data value for that field.

TESTING

❖ Testing objectives:

- Testing is a process of exceuting a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers as yet undiscovered error.

❖ Testing Techniques:

Once source code has been generated ,software must be tested to uncover(and correct)as many errors as possible before delivery to your customer.

Our goal is to design a series of test case that have a high likelihood of finding errors.

Software is tested from two different perspectives:-

- Internal program logic is exercised using "White Box"test case design techniques.
- Software requierements are exercised using "Black box" text case design techniques.In both cases,the intent is to find the

  Maximum number of errors with the minimum amount of effort and time.

❖ White Box Testing:

White box testing sometimes called Glass box testing.Is a test case design method that uses the control structure of the rocedural design to derive test cases?Using white box testing methods ,the software engineer can derive test case that

- ✓ Guarantee that all independent paths within a module have been exercised at least once.
- ✓ Exercise all logical decisions on their true and false sides.
- ✓ Exercise internal data structures to ensure their validity.

A reasonable question might be posed at this juncture."Why spend time and energy worrying about (and testing) Logical minutiate when we might better expend effort ensuring that program requirements have been met?" State another way ,why don't we spend all of our energy on black box tests?

Logic error and incorrect assumptions are inversely proportional to the probability that a program path will be executed.Errors tend to creep into our work when we design and implement function,conditions,or control that are out of the mainstream.

We often believe that a logical path is not likely to be executed when,in fact,it may be executed on a regular basis.

Typographical errors are random. When a program is translated into rogramming language source code,it is likely that some typing errors will occur. many will be uncovered by syntax and type checking mechanisms,but othersmy go undetected until begins. White-Box Testing is far more likely to uncover them.

❖ Black – Box Testing:-

Black Box testing ,also called Behavioral Testing,focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing is not an alternative to white – box techniques. Rather,it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

Black Box testing attempts to find errors in the following categories:-

1. Incorrect or missing function.
2. Interface errors.
3. Errors in data structures or external data base access.
4. Behavior or performance errors, and
5. Initializationa and termination errors.

Unlike white-Box testing. which is performed earluy in the testing process,black box testing purposely disregards control structure,attention is focused on the information domain. Tests are designed to answer the following questions:-How is functional validity tested?

1. How is system behavior and performance tested?
2. What classes of input will make good test cases?
3. Is the system particularly sensitive to certain input values?
4. How are the boundaries of a data class isolated?
5. What data rates and data volume can the system tolerate?
6. What effect will specific combinations of data have on system operation?

❖ Testing strategies

System engineering defines the role of software and leads to software requirements analysis where the information domain, function, behavior, performance, constraintsand validation criteria for software are established. Moving inward along the spiral inward along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral(shown in figure):-

1. Unit Testing :- Unit testing begins at the vertex of the spiral and concentrates on each unit(i.e component)of the software as implemented in source code.It focuses verification effort on the smallest unit of software design – the software component or module.The unit test is white-box oriented,and the step can be conducted in parallel for multiple components.

Unit Test Considerations:

- Interface
- local data structures
- Boundary conditions
- Independent paths
- Error handling paths

2. Integtarion Testing :- In this strategies it focus is on design and construction of the software architecture.Integration testing is a systematic technique for construction the program structure while at the same time conductiong tests to uncover errors associated with interfacing.The objective is to take unit tested components and build a program structure that has been dictated by design.

Components of Integration Testing:-

a) Top-down Integration: Top-down integration testing is an incremental approach to construction of rogram structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control (main program). Modules subordinate (and ultimately subordinate) to the main control module are incorporated into the structure in either a depth-first or breadth – first manner.

b) Bottom-up Integration: bottom-up integration testing ,as its name implies,begins construction and testing with atomic modules (i.e components at the lowest levels in the program structure). Because components are integrated from the bottom up, processing required for components subordinate to a given level is always available and the need for stubsis eliminated.

c) Regression Testing: It is the re-execution of some subset of tests that have alredy been conducted to ensure that changes have not propagated unintended side effects.

Smoke Testing: Smoke testing is an integration testing approach that is commonly used when"shrink-wrapped"software products are being developed.It is designed as pacing mechanism for time-critical projects,allowing the software team to assess its project on a frequent basis.

4. Validation Testing:

Validation can be defined in many ways but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the custome

a) Validation Test Criteria : Two possible conditions exist:

1. The function or performance characteristics conform to specification and are accepted
2. A deviation from specification is uncoverd and a deficiency list is created.

It is often necessary to negotiate with the customer to establish a method for resolving deficiencies.

b) Configuration Review: An important elements of the validation process is a configuration review. The configuration review, sometimes called an audit.

c) Alpha Testing : The alpha test is conducted at the developer's site by a customer. Alpha tests are conducted in controlled environments.

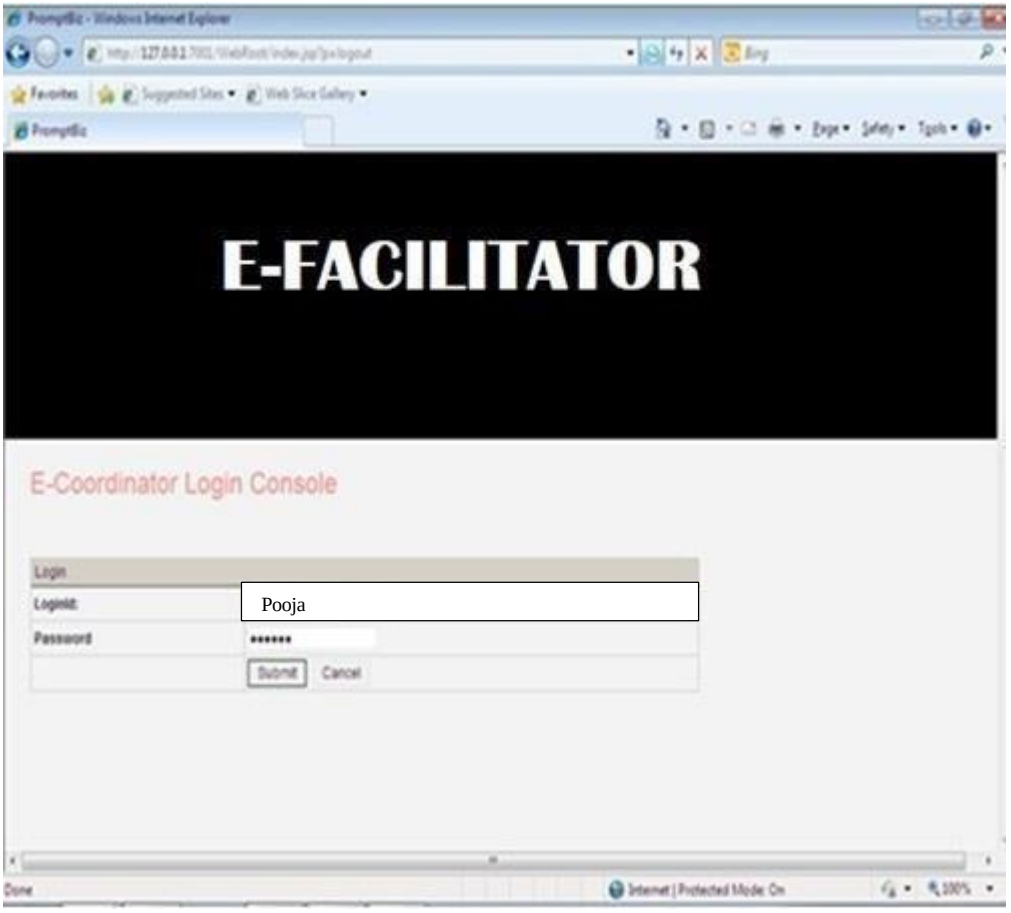
d) Beta Testing : The beta test is conducted at one or more customer sites by the end-user of the software. The beta test is a "live" application of the software in an environment that cannot be controlled by the developer.

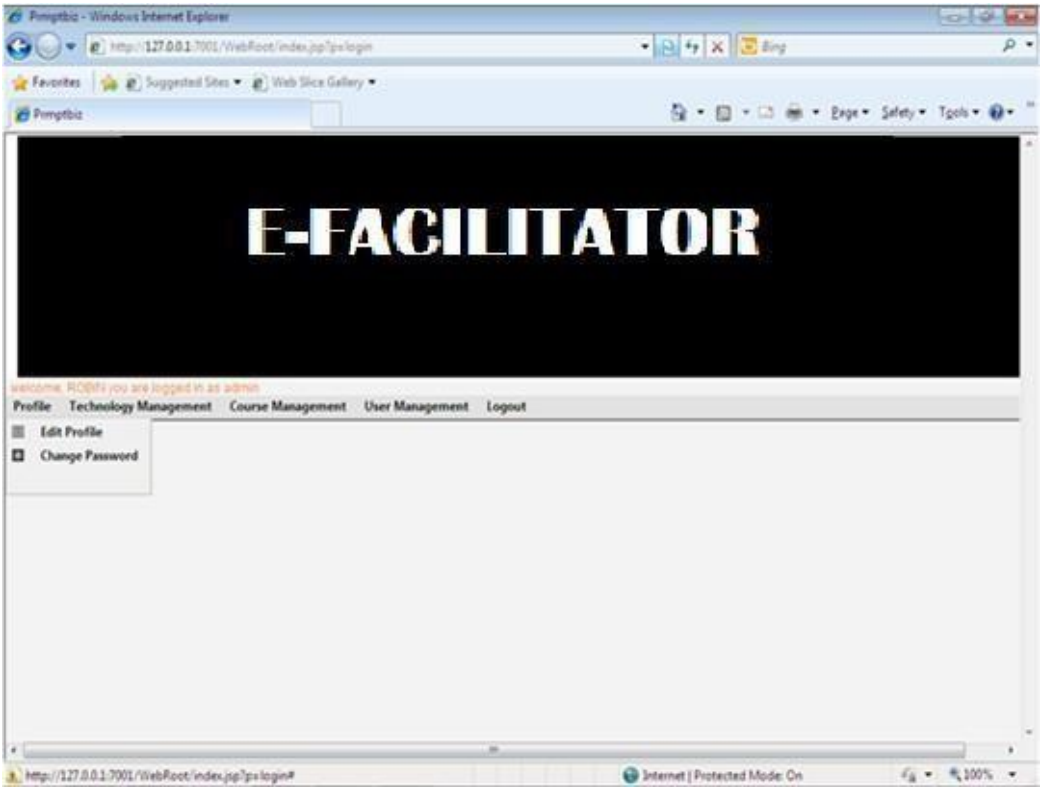
5. System Testing : In system Testing the software and other elements are tested as whole. System testing is actually as series of different tests whose primary purpose is to fully exercise the computer-based system.

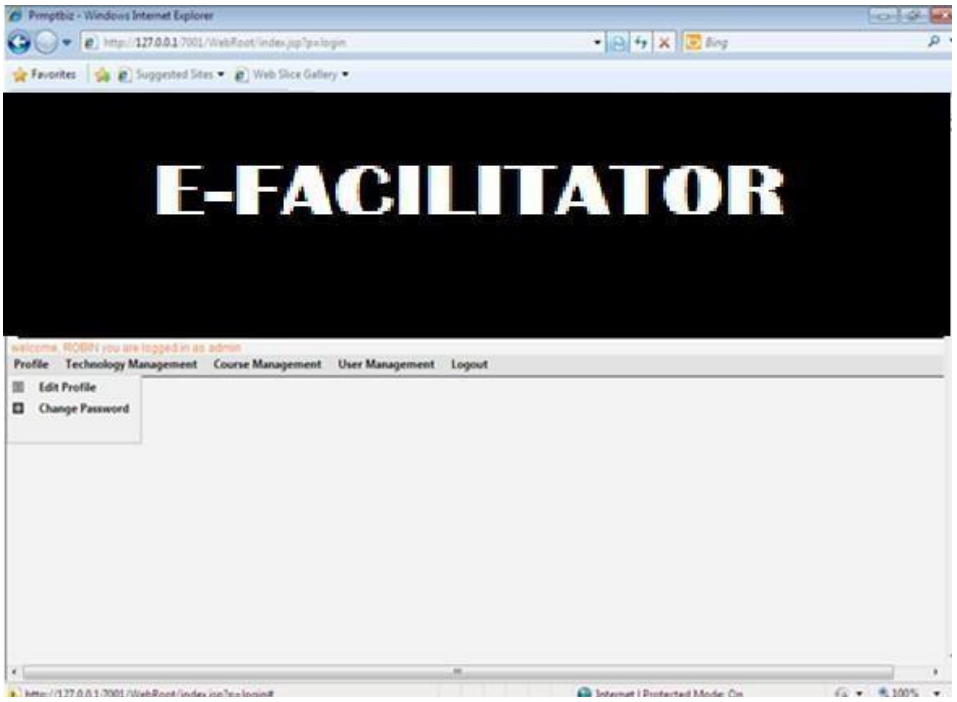
Although each test has a different perpose, all work to verify the system elements have been properly integrated and perform allocated functions.

To test computer software, we spiral out along streamlines that broaden the scope of testing with each turn.

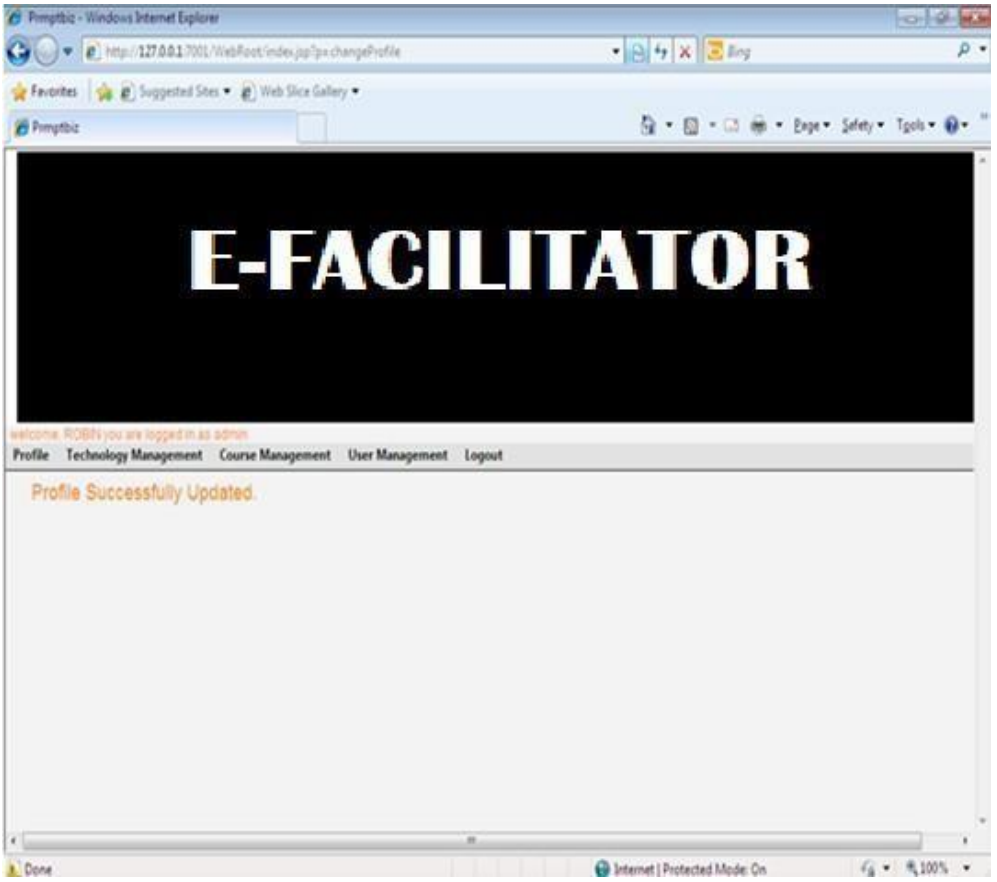
INPUT/OUTPUT SCREEN

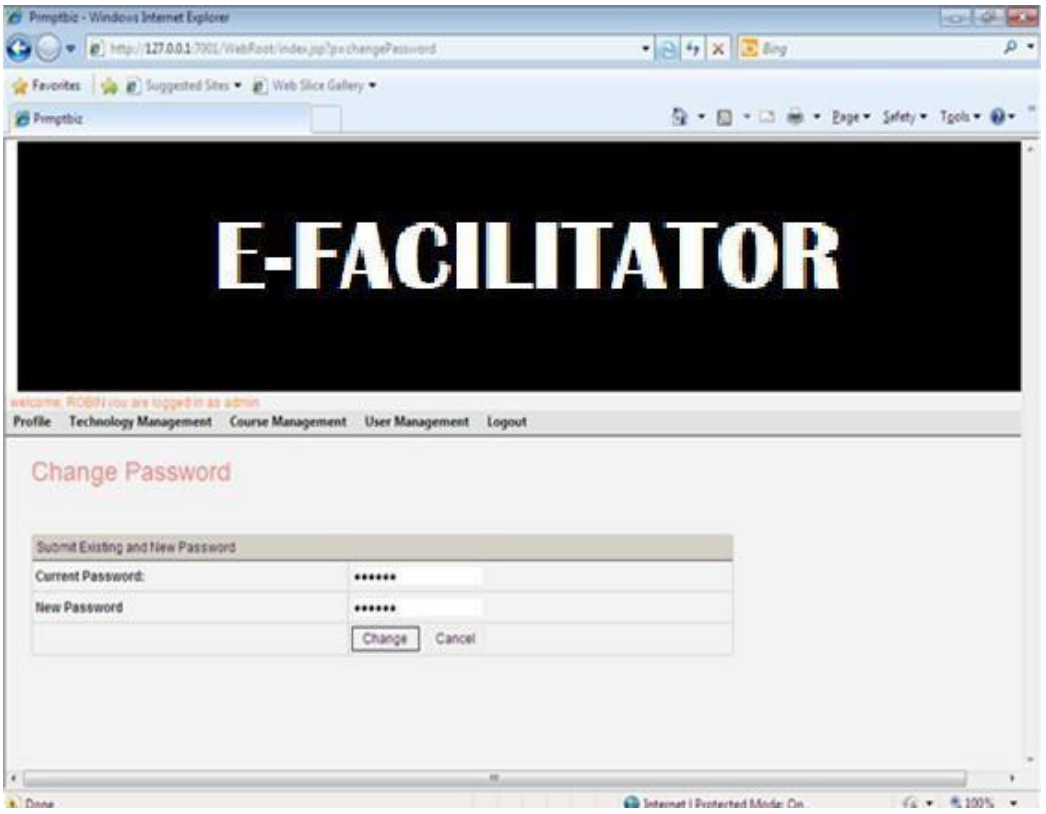

E-FACILITATOR
E-Coordinator Login Console
Login
LoginId:
Pooja
Password
Submit
Cancel


E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Edit Profile
Change Password

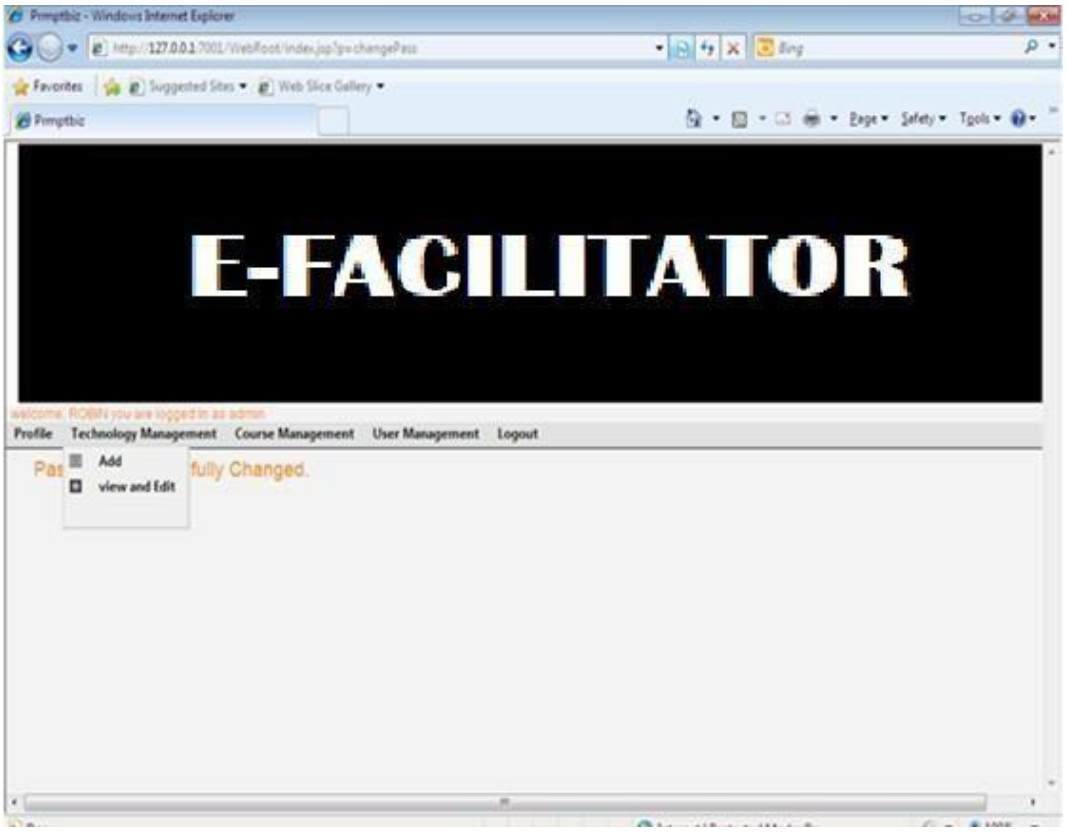

Pmptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Edit Profile
Change Password

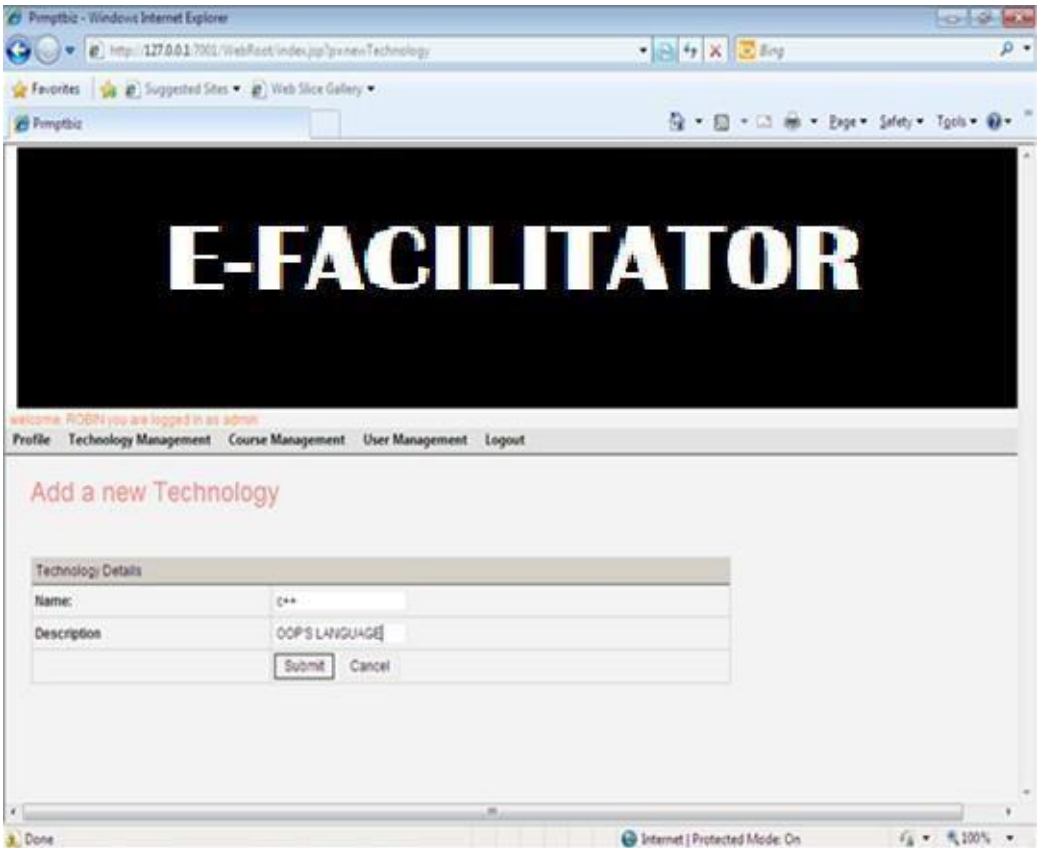

E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Edit Profile
User Details
name:
Pooja
address:
DELHI
mailId:
mobileNo:
Submit
Cancel

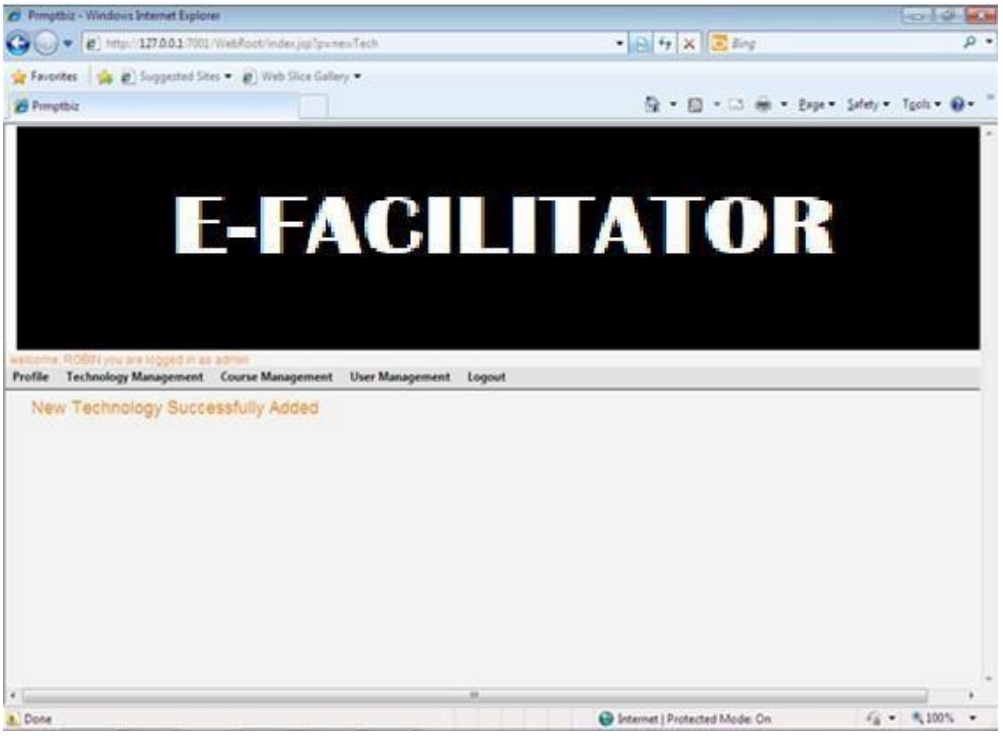

Pmgtbiz - Windows Internet Explorer
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Profile Successfully Updated.

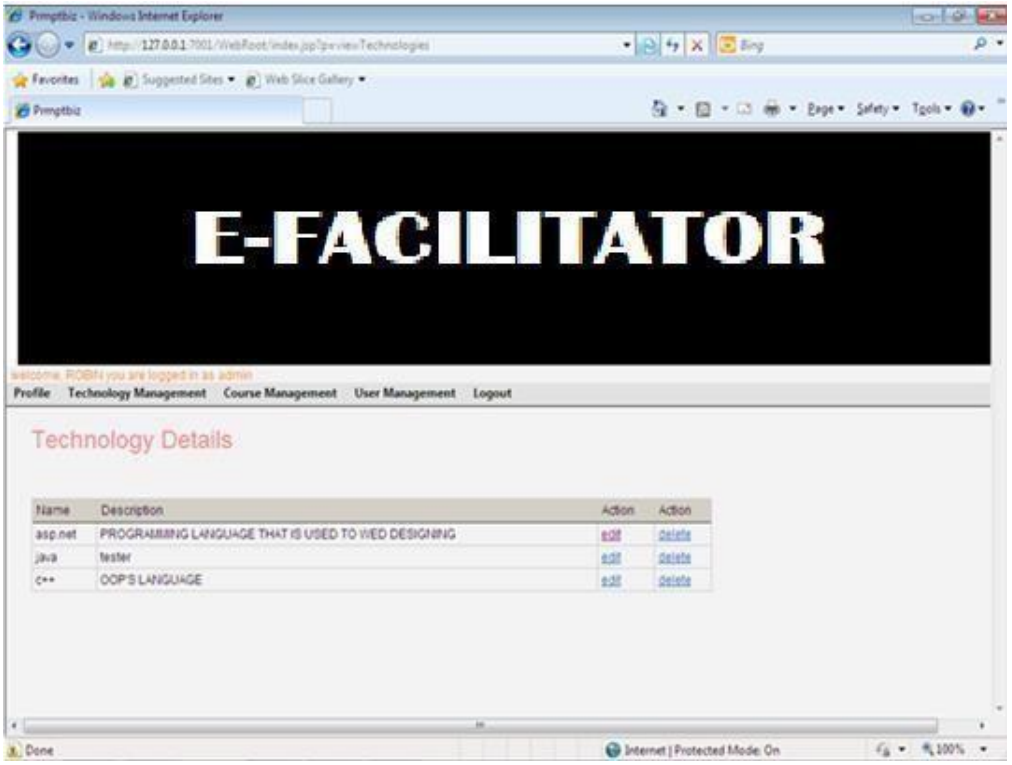

Pmgtbiz - Windows Internet Explorer
Favorites
Pmgtbiz
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Change Password
Submit Existing and New Password
Current Password:
New Password
Change
Cancel

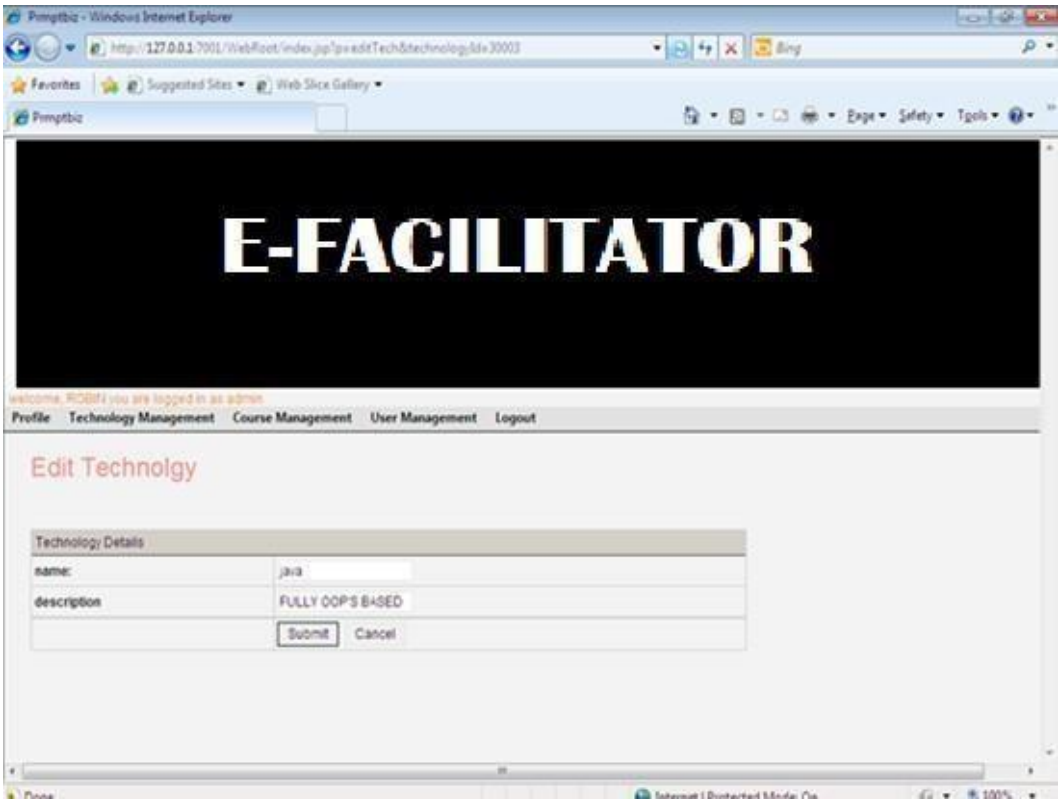

E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Password Successfully Changed.

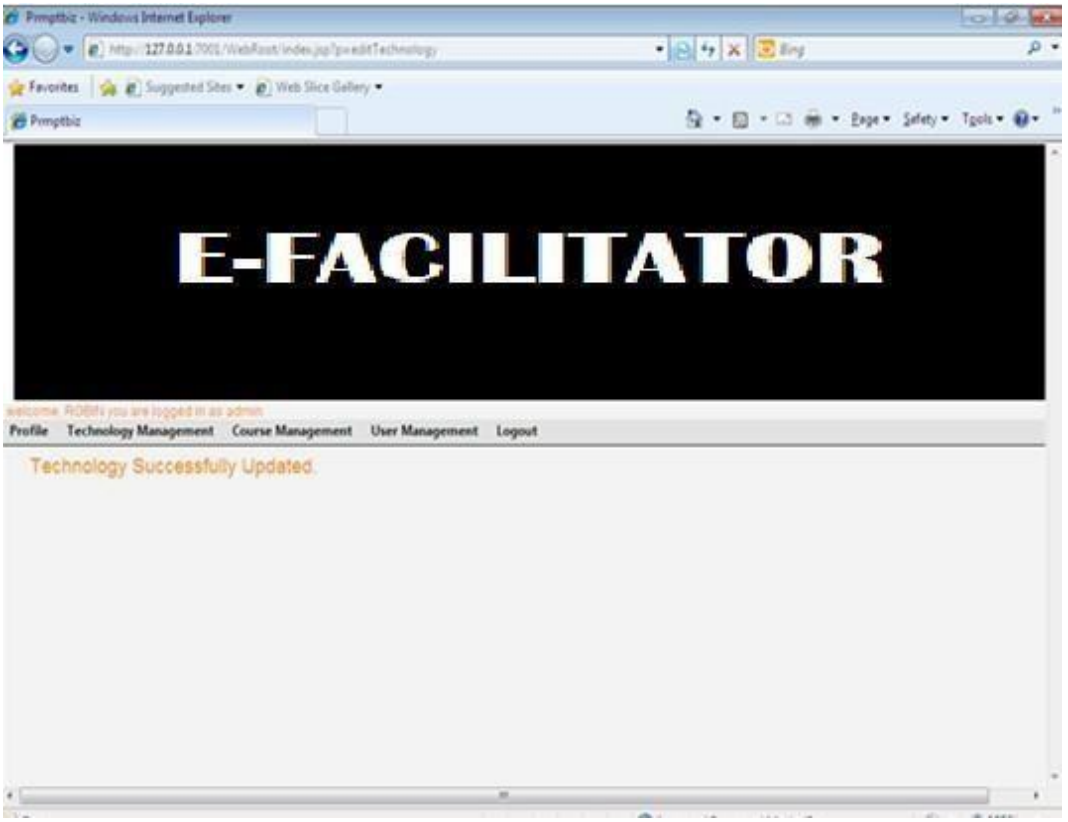

Pimptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pimptbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Add
view and Edit
fully Changed.

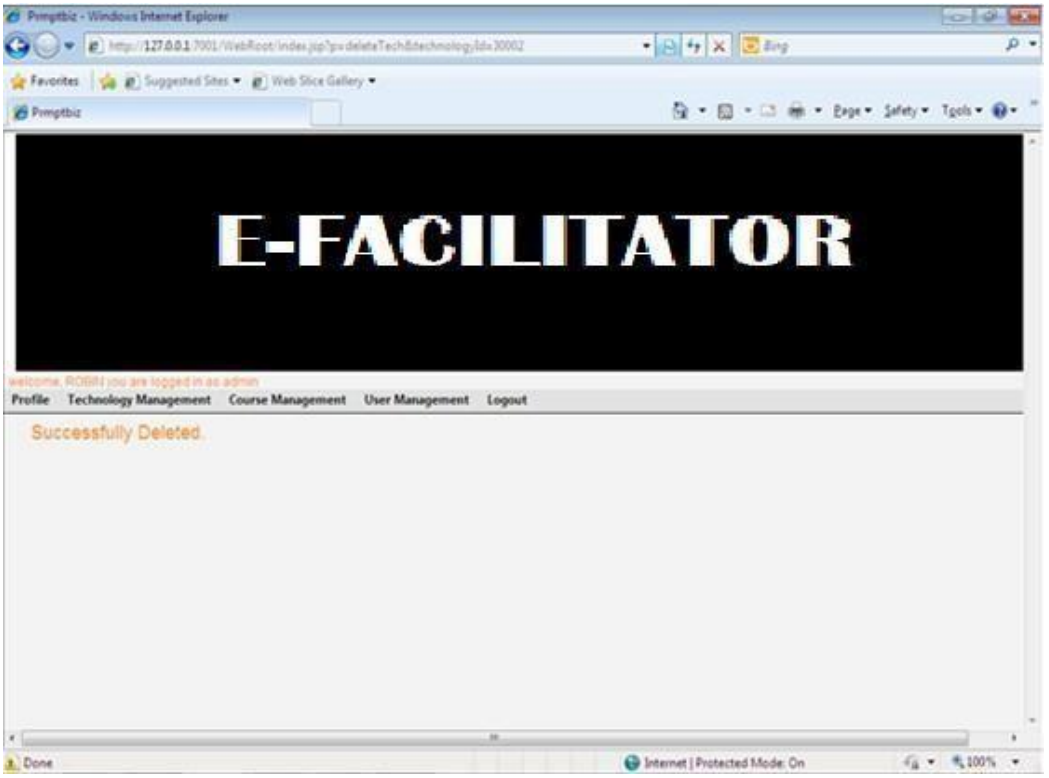

Pimpthiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pimpthiz
E-FACILITATOR
welcome ROBIN you are logged in as admin
Profile
Technology Management
Course Management
User Management
Logout
Add a new Technology
Technology Details
Name:
c++
Description
OOP'S LANGUAGE
Submit
Cancel
Done
Internet | Protected Mode: On
100%

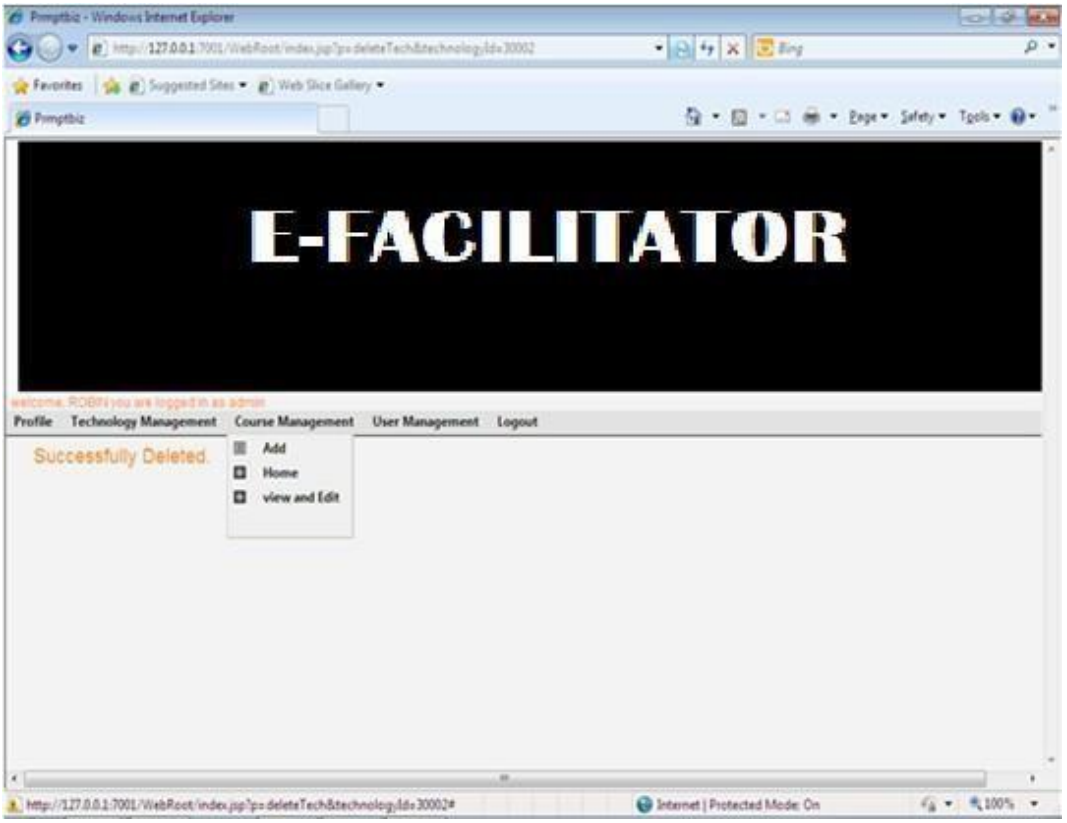

Prmptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Prmptbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
New Technology Successfully Added
Done
Internet | Protected Mode: On
100%

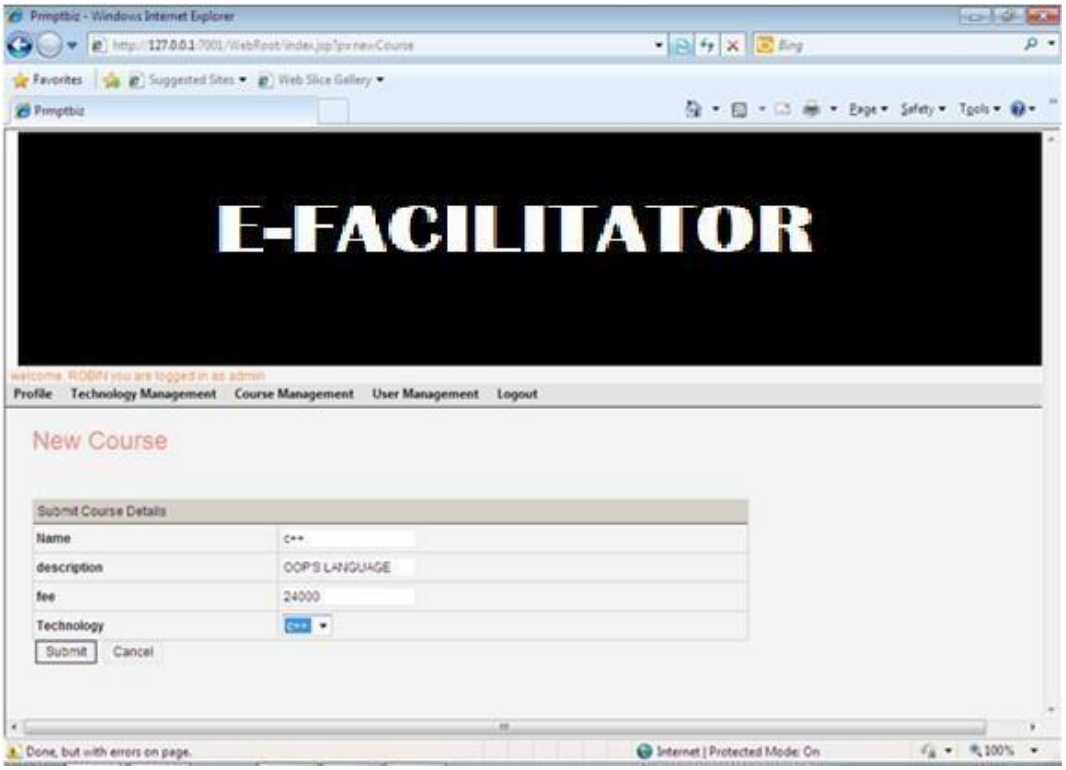

Pmgtbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pmgtbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Technology Details
Name
Description
Action
Action
asp.net
PROGRAMMING LANGUAGE THAT IS USED TO WED DESIGNING
edit
delete
java
tester
edit
delete
c++
OOP'S LANGUAGE
edit
delete
Done
Internet | Protected Mode: On
100%

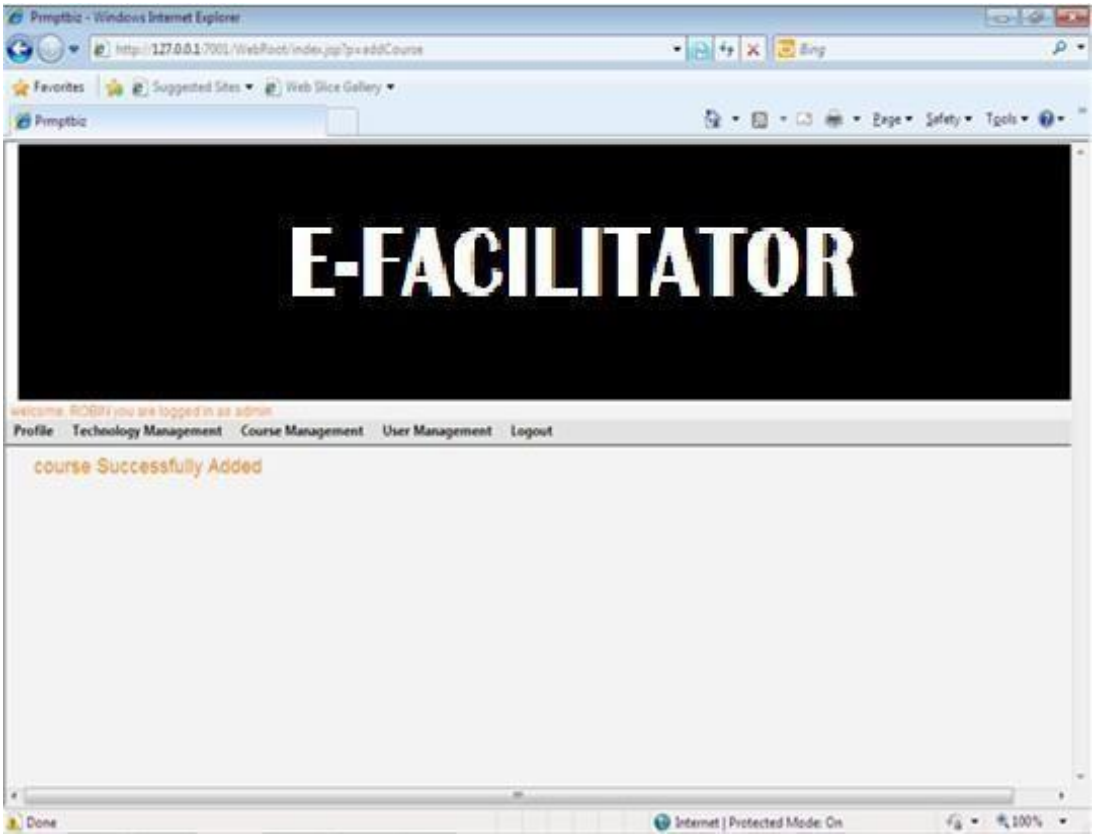

E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Edit Technolgy
Technology Details
name:
java
description
FULLY OOPS BASED
Submit
Cancel

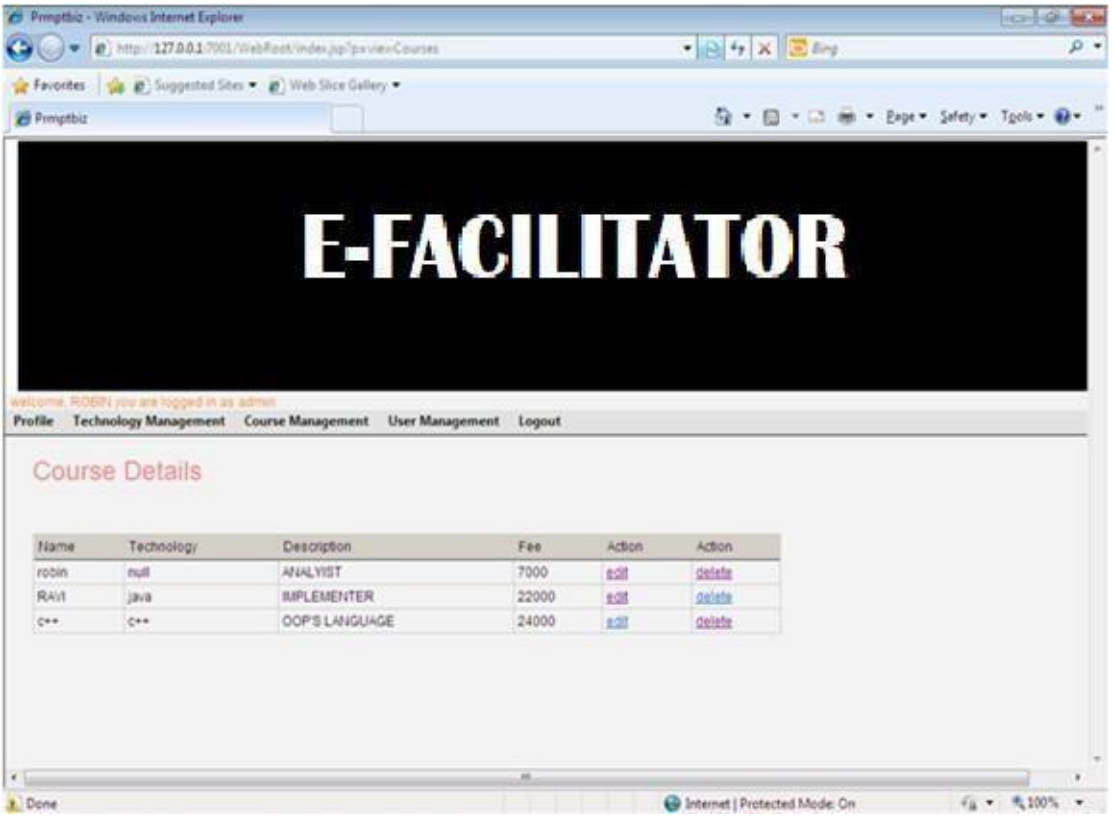

Pmptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pmptbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Technology Successfully Updated.

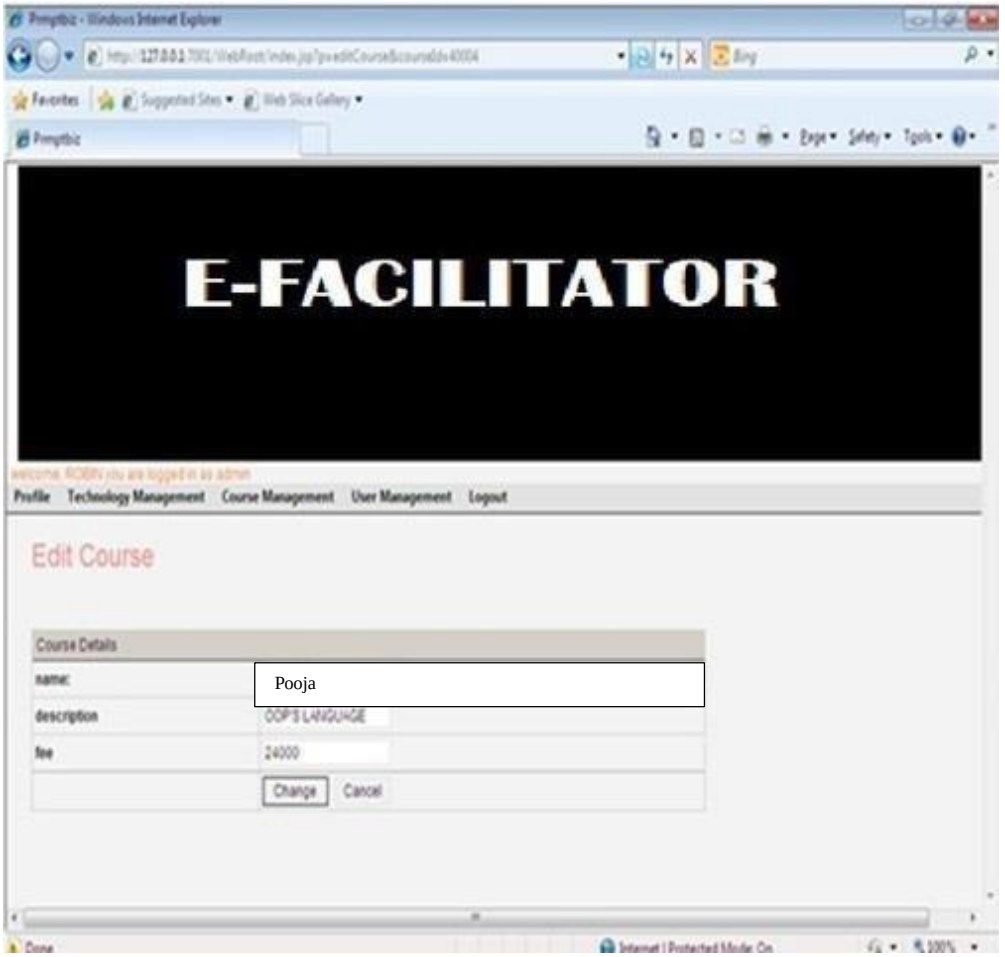

Pimptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pimptbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Successfully Deleted.
Done
Internet | Protected Mode: On
100%

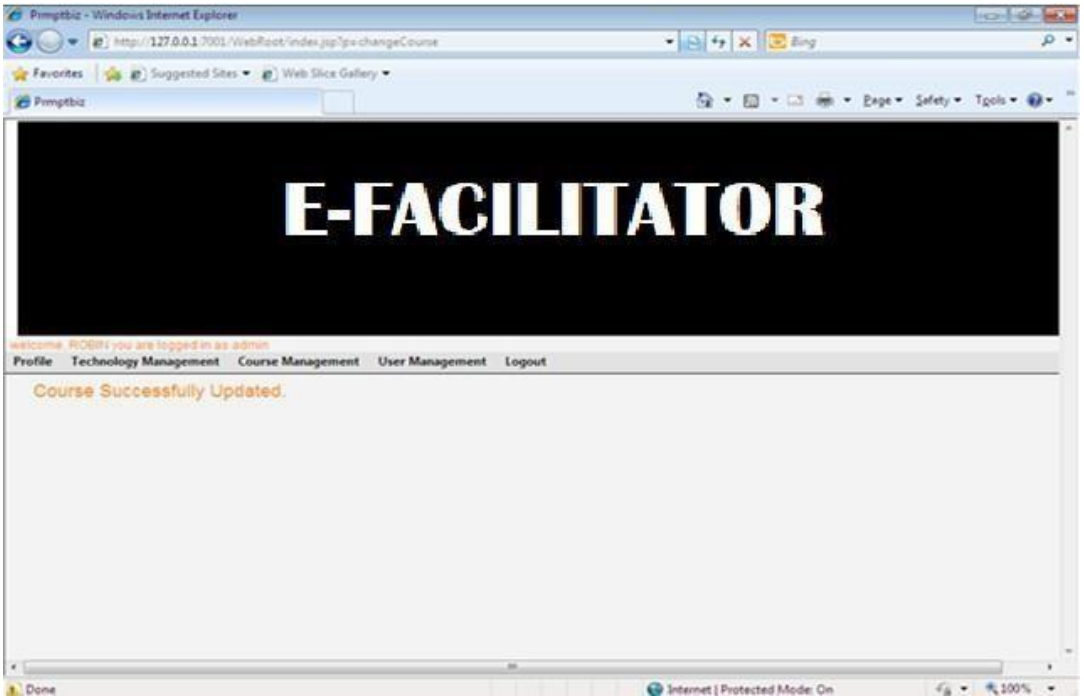

Pmgtbiz - Windows Internet Explorer
Favorites
Pmgtbiz
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Successfully Deleted.
Add
Home
view and Edit
Internet | Protected Mode: On

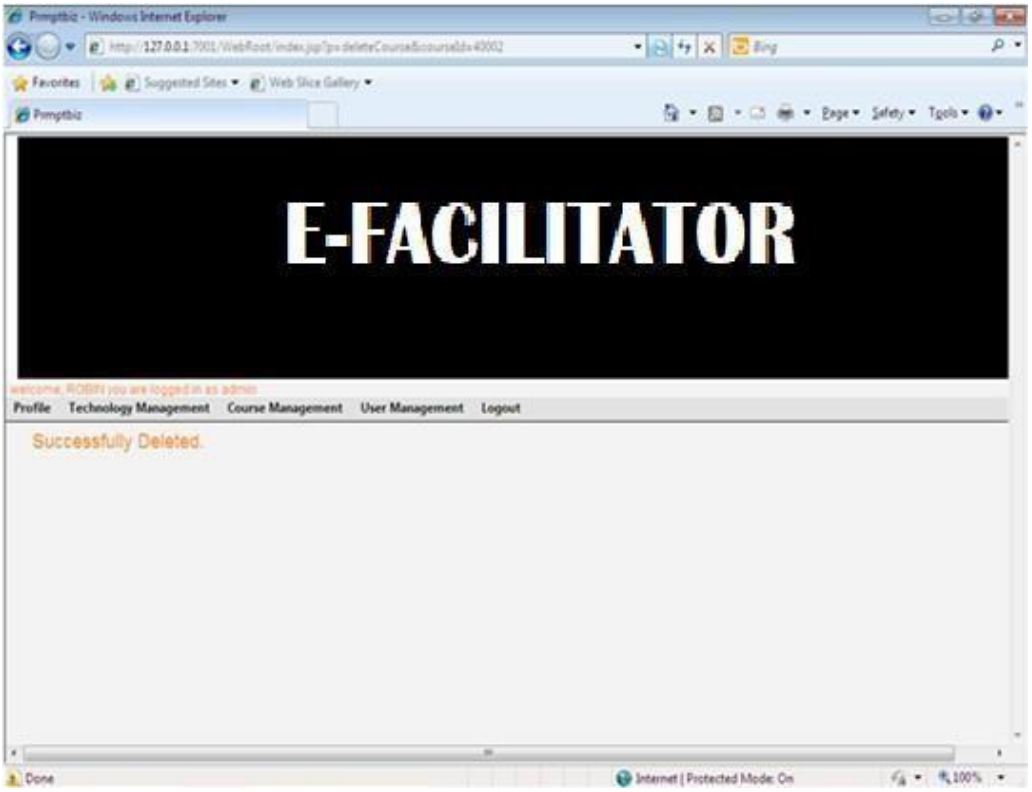

Prmptbiz - Windows Internet Explorer
http://127.0.0.1:7001/WebRoot/index.jsp?p=newCourse
Favorites
Suggested Sites
Web Slice Gallery
Prmptbiz
Page
Safety
Tools
E-FACILITATOR
welcome ROBIN you are logged in as admin
Profile
Technology Management
Course Management
User Management
Logout
New Course
Submit Course Details
Name
c++
description
OOP'S LANGUAGE
fee
24000
Technology
c++
Submit
Cancel
Done, but with errors on page.
Internet | Protected Mode: On
100%

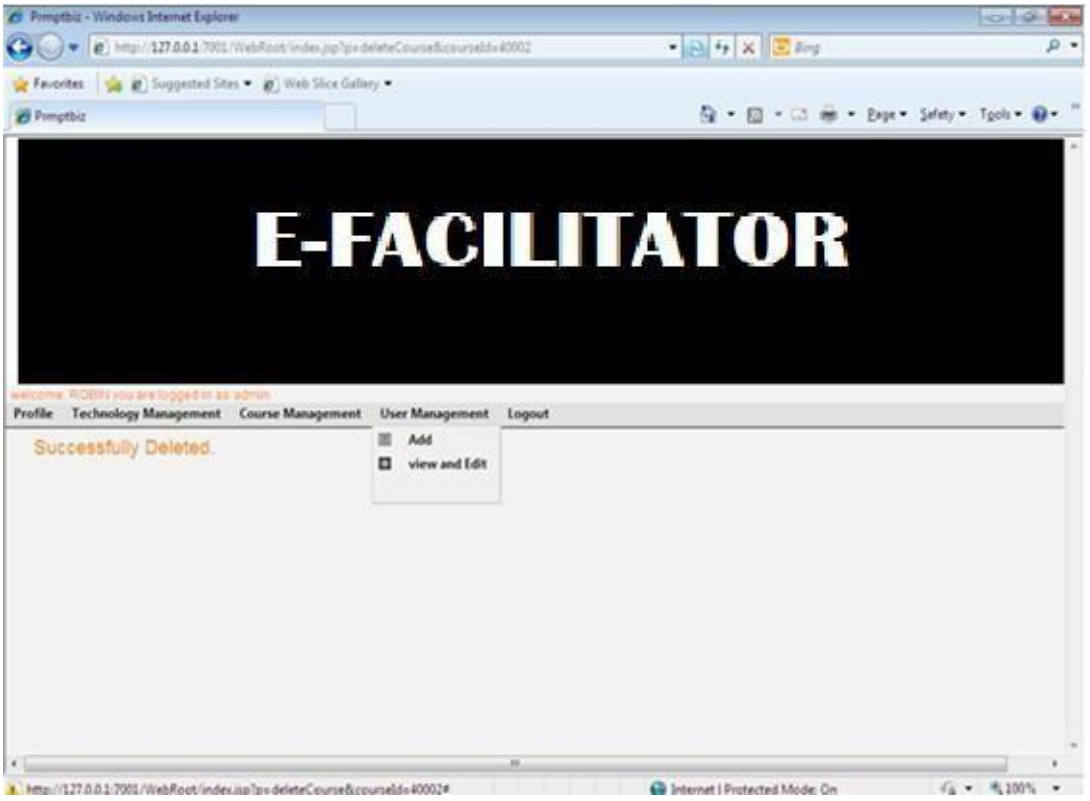

Pmptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pmptbiz
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
course Successfully Added
Done
Internet | Protected Mode: On
100%


Prmptbiz - Windows Internet Explorer
http://127.0.0.1:7001/WebRoot/index.jsp?pr=viewCourses
Favorites
Suggested Sites
Web Slice Gallery
Prmptbiz
Page
Safety
Tools
E-FACILITATOR
welcome, ROBIN you are logged in as admin
Profile
Technology Management
Course Management
User Management
Logout
Course Details
Name
Technology
Description
Fee
Action
Action
robin
null
ANALYIST
7000
edit
delete
RAVI
java
IMPLEMENTER
22000
edit
delete
c++
c++
OOP'S LANGUAGE
24000
edit
delete
Done
Internet | Protected Mode: On
100%

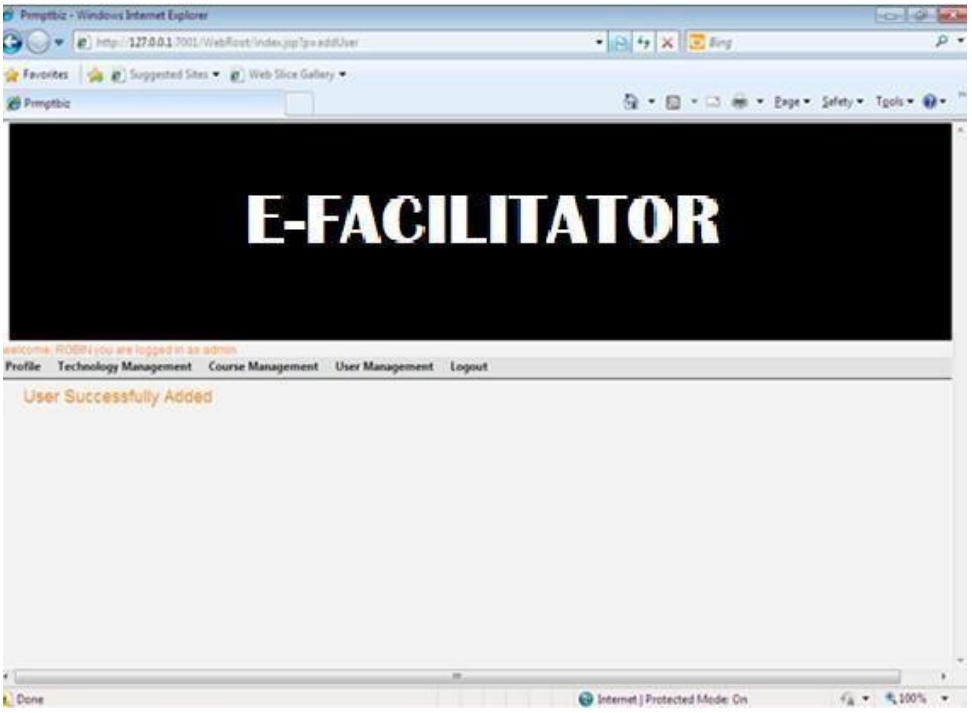

E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Edit Course
Course Details
name:
Pooja
description
OOP'S LANGUAGE
fee
24000
Change
Cancel

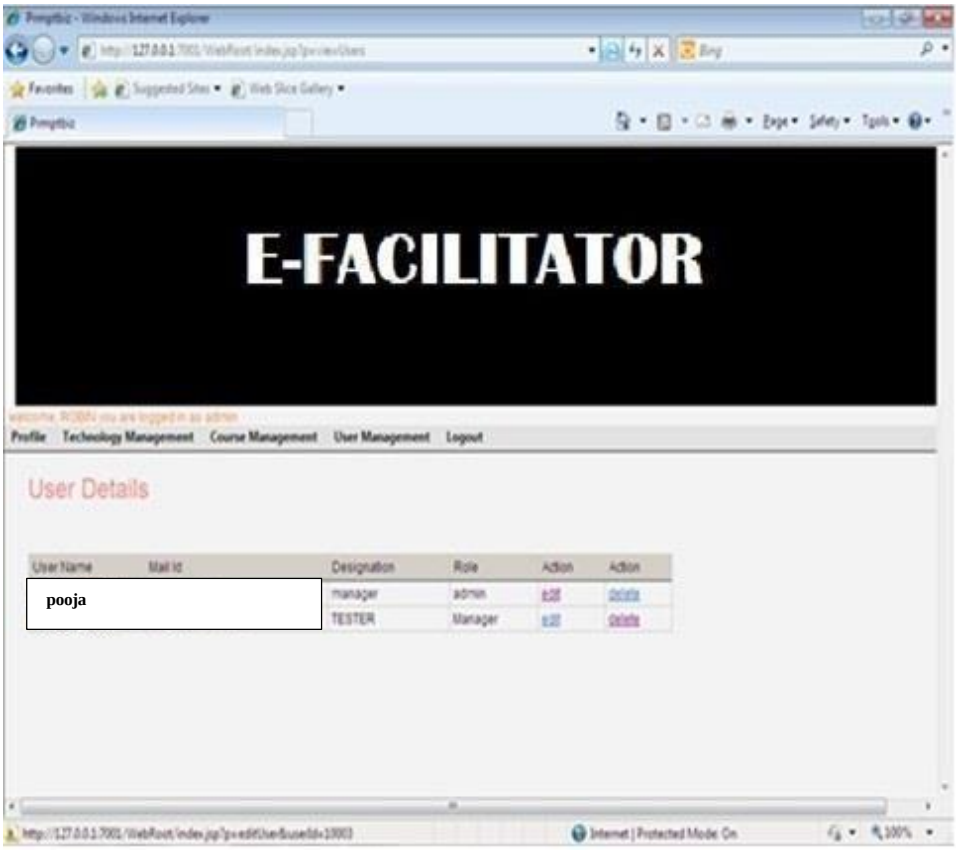

Pmptbiz - Windows Internet Explorer
http://127.0.0.1:7001/WebRoot/index.jsp?p=changeCourse
Favorites
Suggested Sites
Web Slice Gallery
Pmptbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Course Successfully Updated.
Done
Internet | Protected Mode: On
100%

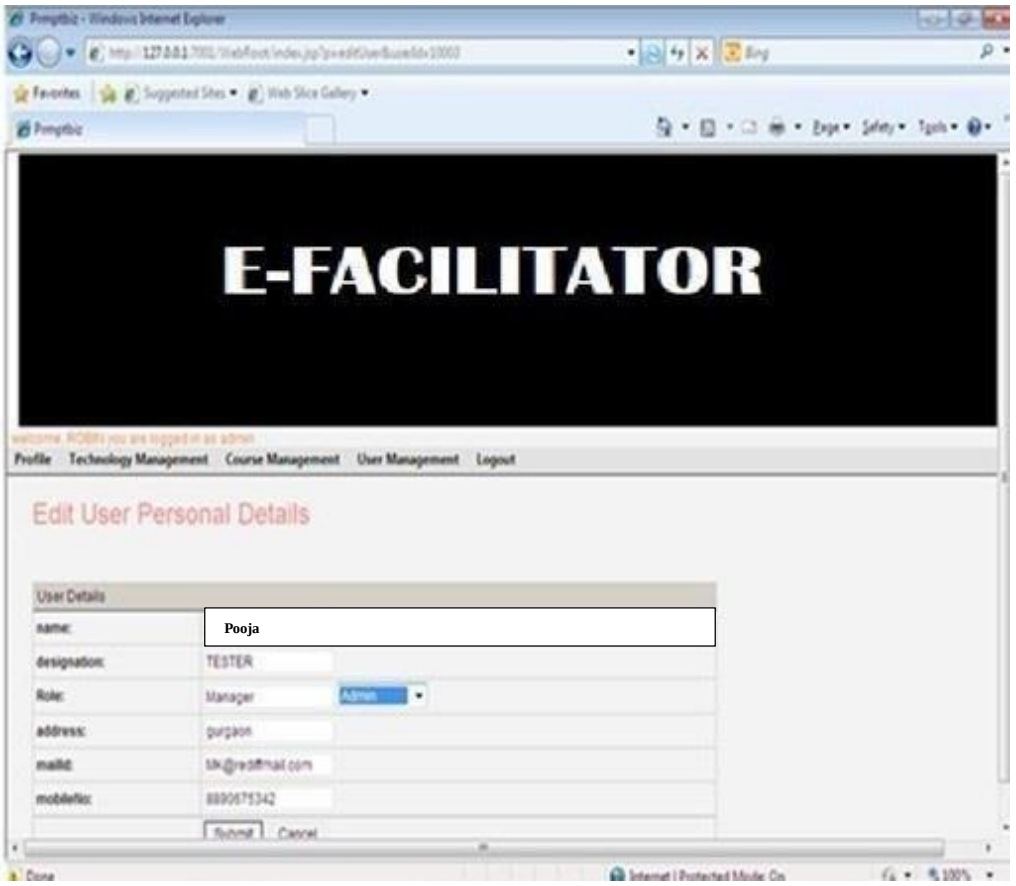

Pimpthiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pimpthiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Successfully Deleted.
Done
Internet | Protected Mode: On
100%

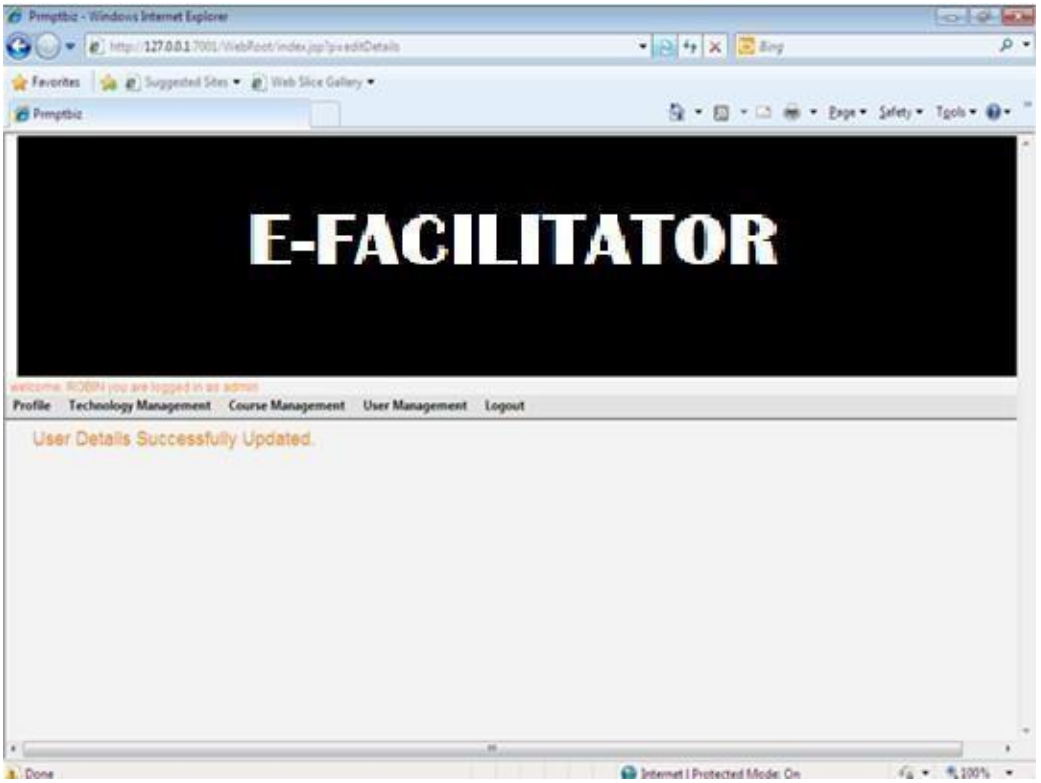

Pmptbiz - Windows Internet Explorer
Favorites
Pmptbiz
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Add
view and Edit
Successfully Deleted.

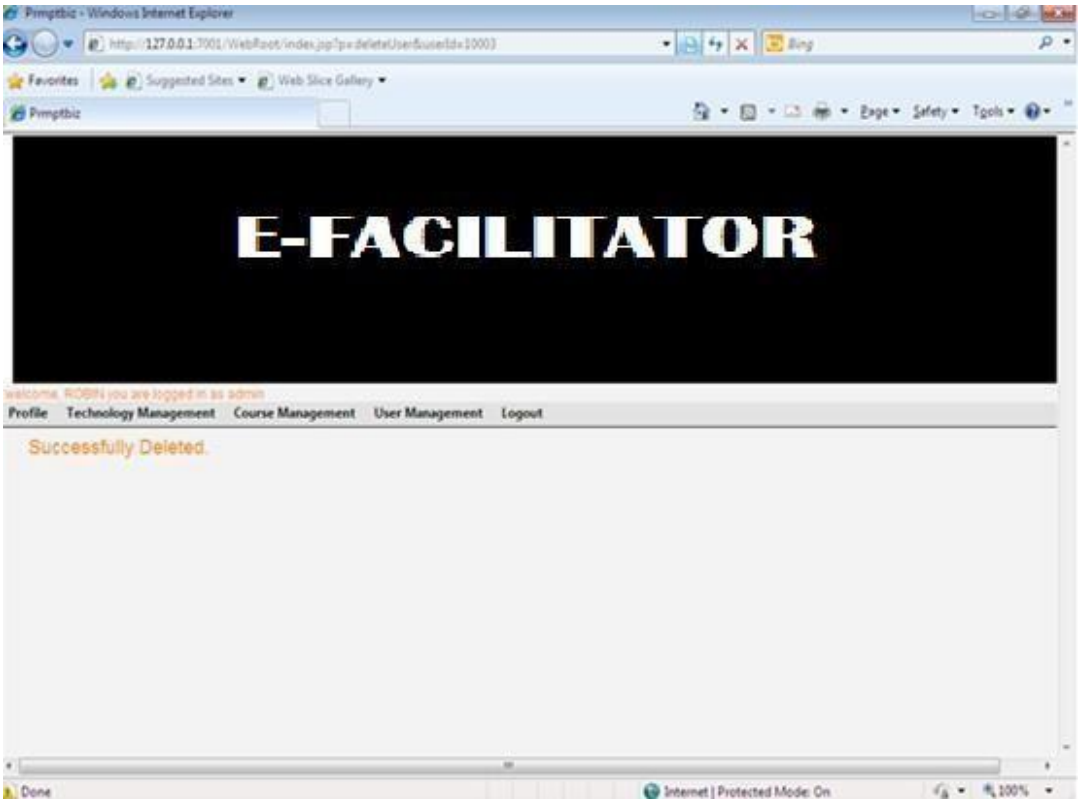

E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
New User
Personal Information
Name
Pooja
Address
Mobileno
Mailid
Designation
TESTER
Loginid
12345
Password
Technology
Role
Submit
Cancel

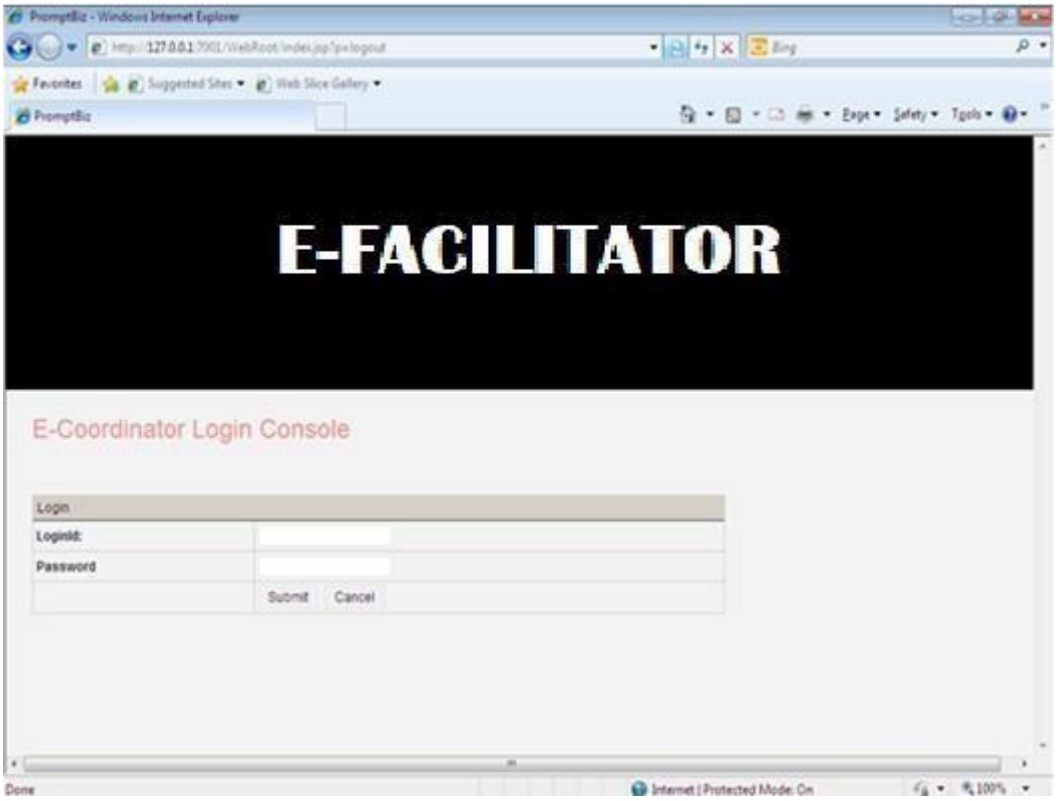

E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
User Successfully Added


E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
User Details
User Name
Mail Id
Designation
Role
Action
Action
pooja
manager
admin
edit
delete
TESTER
Manager
edit
delete


E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Edit User Personal Details
User Details
name:
Pooja
designation:
TESTER
Role:
Manager
Admin
address:
gurgaon
mailid:
mobileNo:
8890675342
Submit
Cancel


E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
User Details Successfully Updated.


Pmptbiz - Windows Internet Explorer
Favorites
Suggested Sites
Web Slice Gallery
Pmptbiz
Page
Safety
Tools
E-FACILITATOR
Profile
Technology Management
Course Management
User Management
Logout
Successfully Deleted.
Done
Internet | Protected Mode: On
100%


PromptBiz - Windows Internet Explorer
Favorites
PromptBiz
E-FACILITATOR
E-Coordinator Login Console
Login
LoginId:
Password
Submit
Cancel
Done
Internet | Protected Mode: On

SECURITY MEASURES

As far as the security is concern, "E-FACILITATOR " will provide a reliable security system for protecting the user databasefrom unauthorized access. One module (i.e. LOGIN) will take care of the security of the project database. To implement the LOGIN modules, one table has been included into the project database called LOGIN, which will store information about various users who will interact with the project. Details like USERID,PASSWARD will be stored in this Table. Users will be allowed to access project database only after feeding their LOGIN and PASSWARD correctly. LOGIN module and accordingly the access right to user database will be giver to the user.

❖ Database/data security

MYSQL-

MYSQL is Object Relational database Management System (ORDBMS). It offers capabilities of both relational and object oriented database system. In general objects can be defined as reusable software codes which are location independent and perform a specific task on any application environment with little or no change to the codes.

SQLis used to access the data within the MYSQL. It contains a set of commands, which make it very easy to maintain the database. It has for sub parts DDL, DML, DCL, and TCL. DDL includes the commands, which allows us to create objects and to manipulate the structure of the objects.

- DML includes the commands to manipulate the information stored in a database.
- DCL includes the commands for controlling the data access and
- TCL includes the commands for controlling the transactions like commit and rollback

.The database server or back-end is used to manage the database files optimally among multiple clients who concurrently request the server for the same data.

It also enforces Data Integrity across all client application and controls database access and other security requirements.

VALIDATION CHECK

Validation for any organization is necessary checks to make the entire projects more reliable,strong, effective, and more efficient. When the user passes the value it would be checked to fall within the range of software provided for E-Facilitator system. If values are incorrect or any data type error then the appropriate message would be flash and the user come to know the exact position fault for error in software. Due to the validation check software becomes more effective as it does not accept the wrong entries to the data.

As discussed earlier the application has been broken in two tiers like Back End MY SQL and the Front End user interface in J2EE.In this architecture every type of validation check and constraints have been applied at the database level itself as per the code rules.Apart from this every type of data security rules have also been applied by creating appropriate synonyms and user roles.Access to the user has been provided in controlled way keeping the view at users level and their reqirements.Due to which user access is restricted and the security has enforced.

Another level of validation checks has been applied at data entry form level also in the user interface.These two levels of validation checks enforece high level of data validity and only valid data can be entered and processed.The validation rules cannot be bypassed by any means.

For example: -In Student entry Form,I have used a list item for student type field so that onlyeither ‘permanent’or ‘ordinary’data value can be inserted into the database.It prevents the user from inserting any invalid data value for that field.

COST ESTIMATION

The software project management process begins with a set of activities that are collectively called project planning. The first of these activities is Estimation.
Whenever estimates are made, we look into future and accept some degree of uncertain as a matter of course estimation of resources cost and schedule of a software development effort requires experience access to good historical information, and the courage to commit to quantitative measures when qualitative data are all that exist. Estimation carries inherent risk that leads to uncertainity.
Project size is another important factor can affect the accuracy of estimates. As size increases, the interdependency among various elements of the software grows rapidly. Time also effect cost of the project. If time increases then cost is also increases then cost is also increased.

Risk is measured by the degree if uncertainly in the quantitative estimates established for resources, cost and schedule. If project requirements are subject to change, uncertainly and risk factors become dangerously high. The software planner should demand completeness of function, performance and interface definitions. The planner and customer, should recognize that variability in software requirements means instability in cost and schedules. A project manager should not become obsessive about estimation; modern software engineering approaches take an iterative view of the development. In such approaches it is possible to revisit the estimate and revise it when customer makes changes to requirements.

LIMITATIONS OF PROJECTS

E-F ACILITATOR may also suffer from the following limitations-

- Knowledge of MS WINDOWS NT, MS WINDOWS 98, Oracle 9i and Java is essential.
- Not compatible with all O/ S like Linux, Unix
- Does not provide handling of customer Queries.
- This project as any other project may suffer from mishandling by the users in some cases.

FUTURE SCOPE OF PROJECT

The development of this software has been done keeping in mind the future scope of this application. I find that it has good prospects in the future also. I find it is getting solved the work of the E-FACILITATOR system.After development of this software, I can say that the complete work of this project would be under control and easier to handle. Keeping in mind the needs of the E-FACILITATOR , which may crop up in the near future, I have tried and introduced certain features, which may be required by the club member, and so at that time implementation can be done without any problems. Right now I have developed this software keeping in mind just of E-FACILITATOR , but this job can be done country wise. It will run successfully.

Any product despite of its meticulous design and features needs enhancement with time. E-FACILITATOR being no exception needs active enhancement of features and functionality. Following features are proposed to be implemented in future to make the product more feature rich.

Intranet Messaging: In each organization, there is always a need of efficient paperless, secure, and private communication medium that has the retention capabilities. We are proposing to add Intranet messaging facility to fill this requirement as an independent module in future release.

Template Based Look & Feel: Being a product, it is proposed to be implemented at the site of different clients. Each client has different set of preferences for look and feel of the application. To minimize the customization process, look and feel of the application is proposed to be based on templates in future releases.

BIBLIOGRAPHY

Book Name	Author Name
Java-2 Complete Reference	by Patrick Naughton
Java Servlet Programming	by O'Reilly
MYSQL Developer 2000	by Wrox
Pure JavaScript	by Jason Gilliam, Charlton& Try, R.Allen Wyke
HTML Complete	BPB publications.
Java Server Programming	by Apress Publication.