Course Name: Database Management

Course ID: CSE-303 (Section-2)

Group 2

Name	ID
Md. Johirul Islam	1731456
Jennifer Margaret Gomes Jenny	1721134
Fatematuz Jahura	1830481
Shah Newaz	1730747
S.M. Wasif Zaman Omee	1730049
Md Zahid Hasan	1730634

Table of Contents

Chapter 1: Introduction

BACKGROUND OF THE ORGANIZATION

Independent University, Bangladesh (IUB) established in 1993 is the leading private university in Bangladesh with an explicit focus on Research and Global partnerships. They are a full service, meritocratic ally elite university with a current enrollment of 7,378 students, 11,556 alumni and 401 faculty members. The student population is projected to grow at 10% annually. IUB facilitates three academic semesters: spring, summer and autumn. Admission tests are conducted in November, April and July, two months prior to the beginning of each semester.

OBJECTIVES OF THE PROJECT

The system will be used by the Independent University, Bangladesh. Accurate statistical data representation about the student's performance evaluation in Independent University, Bangladesh. Upgradable design for future expansion. The system will be accessible to the university students, faculties, admin and concerned parties. Creating a system which is more efficient than the current system in terms of determining PLO & CO scores, keeping track of past scores and providing future projections.

BACKGROUND OF THE PROJECT

The main idea of our project is to design a software that will be helpful for the universities everywhere to promote a more productive and effective way of evaluating students' performance. At the very first of our project, we introduced the idea of mapping Program Learning Outcomes (PLOs) with courses. Then the course outcome where every PLO map with cos. To evaluate the students efficiently the project intends to check whether the PLOs that are mapped to the COs requirement is fulfilled or not for each student. The system will give access to the Department to set PLO requirements. Then the department input the COs so that the system can map the COs to PLO accordingly. The system will also allow to the faculty member to update the according to the requirements of the course with this software, the user will be able to access the statistical information about student's departments or courses' performance evaluation in terms of completing COs in a particular course and the overall PLOs achieved in a particular semester at Independent University Bangladesh. The software will be able to analyze the annual student evaluation report of their performance in the program. This will help to get the accurate data of the students enrolled by majors, departments and schools.

Scope of the project

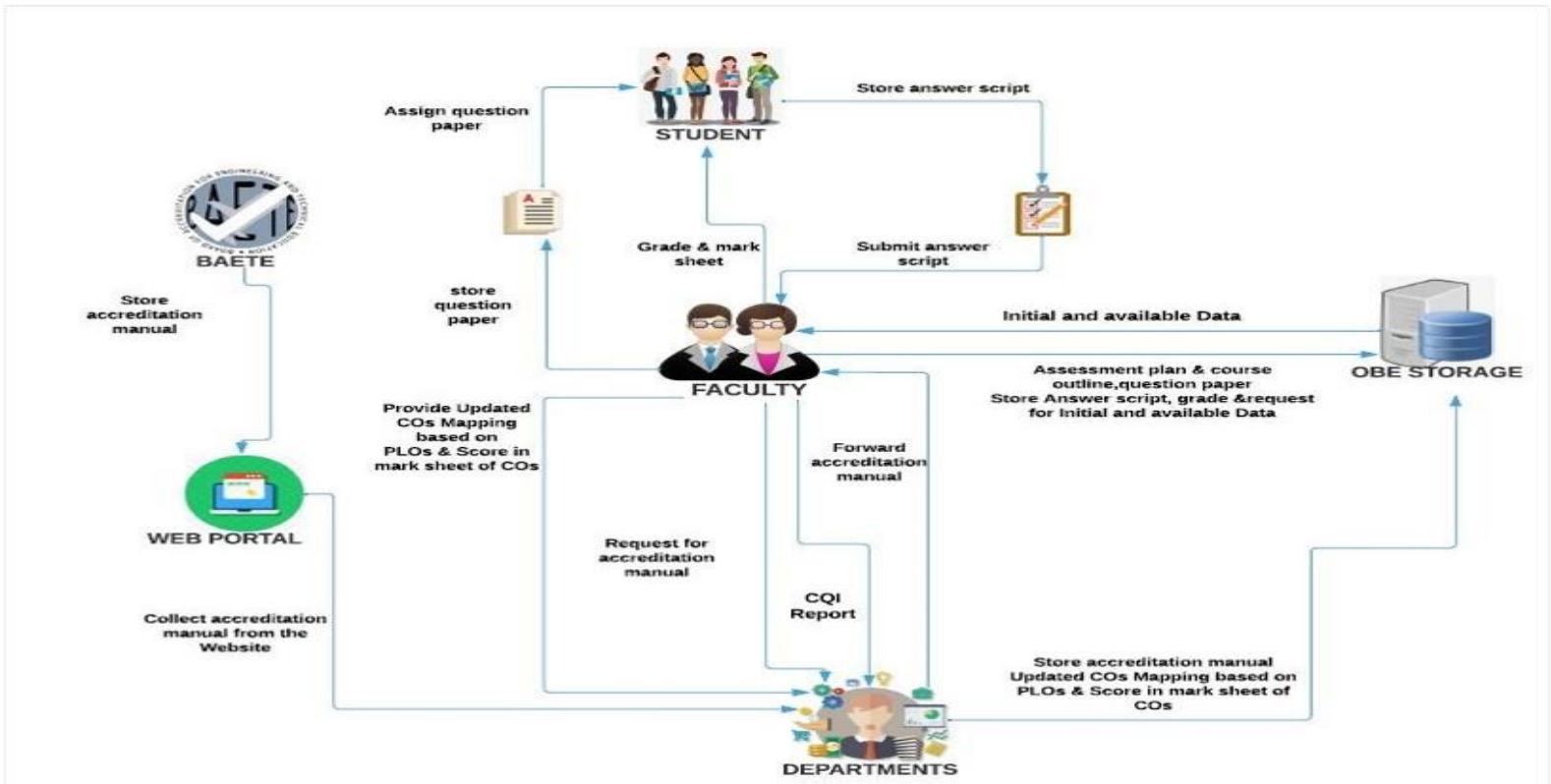
The project will be a replacement to the existing system or manual system. Student assessment monitoring systems will take the process online in a more secure and organized form. The data flow from one user to another will be seamless. Reports for appropriate users and appropriate data will be generated in graph, bar chart and other visual forms for easy reading and understanding of data. Furthermore, there will be scope for further development and integration by the admins.

The scope is to assist in the efficient and effective implementation of the project through the following tasks:

- Facilitate the implementation, including planning and management.
- Support for review and improvement of the project implementation
- Project initiation
- Data Collection
- Potential Modeling
- Program Analysis
- Reporting
- Project management

Chapter 2: Requirement Analysis

DESCRIBING EXISTING BUSINESS SYSTEM (WITH RICH PICTURE



PROCESS ALONG WITH SIX ELEMENT ANALYSIS (AS IS)

AS IS						
Process	**Human**	**Non-Computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Network & Communication**

1. Planning	**BAETE:** 1.1) Store Accreditation Manual with defined PLOs to the web portal. **Department:** 1.1)Department Collect Accreditation Manual with defined PLOs & store it. 1.2) Identifies which courses needs to map. Department will identify only those courses which are offered and always available according to their curriculum select those courses. 1.3)make an initial planning of mapping PLO & Cos. Department list down COs While list down of Cos Department will keep in mind the Domain level of the particular course. Department will check in which domain level of its bloom taxonomy match with the mapped CO/PLO. Then list down the mapped CO from its PLO according to Domain Level. 1.4department a sent initial mapping of COs/PLO, pervious course out line and assessment plan to faculty. 1.5 collect updated PLO& course out line and assessment plan. 1.6 Store updated PLO& course out line and assessment plan. **Faculty:** 1.1Received Accreditation Manual with defined PLOs & initial planning of mapping PLO/CO from the Department.	**Pen & Paper:** 1.1) Department use it for noting down all PLOs & store it as hard copy. 1.2) Department use it to list down available courses. 1.3)department use it in initial mapping of PLO/CO. 1.1)faculty use it to print hard copy of PLOs from department on the paper. 1.6)faculty use it to provide hard copy of Assessment plan & course outline to the student.	**Computer:** 1.1)BEATE use it to upload Accreditation manual to their website. 1.1)Department use computer to Collect Accreditation Manual which is stored in the website of BAETE. 1.3)department use it to create initial mapping of PLO/CO. 1.1)Faculty member use it to collect a soft copy of Accreditation Manual with defined PLOs. 1.2) Faculty use it to collect data and previous course outline and assessment plan from the department 1.4) Faculty use it to create initial mapping of CO, course outline &assessment plan **Printer:** 1.1)Department use it to print hard copy of accreditation manual 1.6)Faculty use it to print out of PLOs, course outline and assessment plan as hard copy.	**Excel sheet:** 1.1)Department use Excel sheet to collect & save Accreditation Manual. **MS Word:** 1.4) faculty use it to create assessment plan, course outline. **PDF Viewer:** 1.1) Department use it to view PDF file of PLOs. **Goggle classroom:** 1.6)faculty use it to upload course outline and assessment plan. **Gmail:** 1.5)Department use it collect updated mapping. 1.2) Faculty use it to collect Accreditation manual from department & sent updated mapping to the department.	**MS Excel** 1.3) Faculty use excel to store mapped COs. **Department OBE Storage:** 1.1)Department use it to store Accreditation Manual with defined PLOs. 1.3)Faculty use it to Store mapped COs from PLOs based on Syllabus.	**Internet:** 1)BAETE use to store their manual to their website. 2)Department use it to collect manual from the web portal. 3)department its use for Communication with BAETE & faculty member via email. 4)department use it to store updated CO mapping to the OBE storage. 5)Faculty use internet to email for communicating to the department and requesting for PLOs. 6) faculty Download PLOs as PDF **Others:** 1) faculty use phone or physical means with department member or with other faculty from same department to discuss about Course outline and COs mapping.

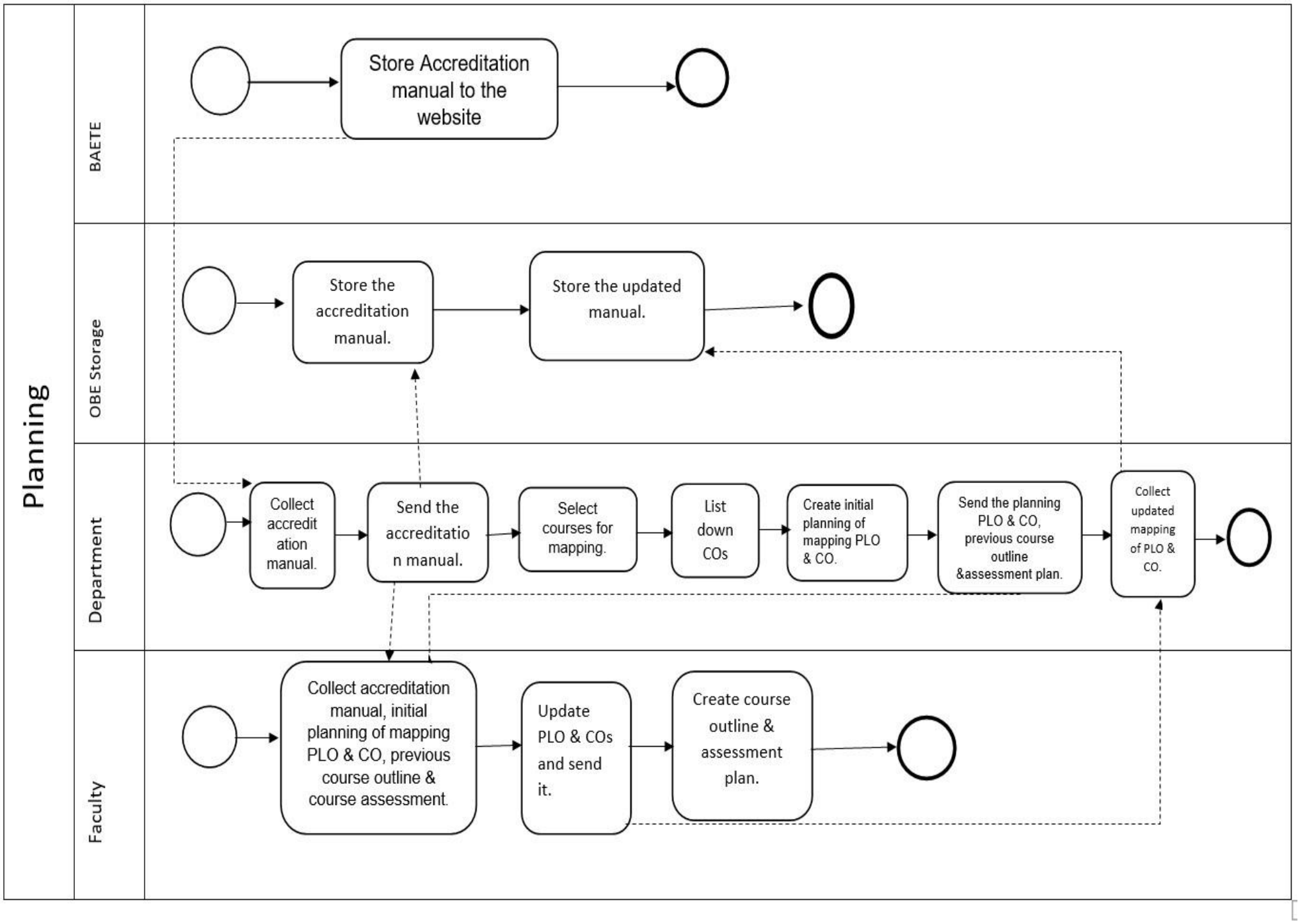
	1.2collect the data of making of assessment plan and course out line from the OBE storage also from the meeting with other faculty and from the previous assessment plan & course outline. 1.3Faculty member list Cos from course description & maps COs from PLOs based on the syllabus. 1.4Based on COs from PLOs faculty create assessment plan & course outline. 1.5 sent to updated mapping of PLO/CO & assessment plan & course outline to the department. 1.6 provide hard & soft copy of assessment plan and course out line to the student.					

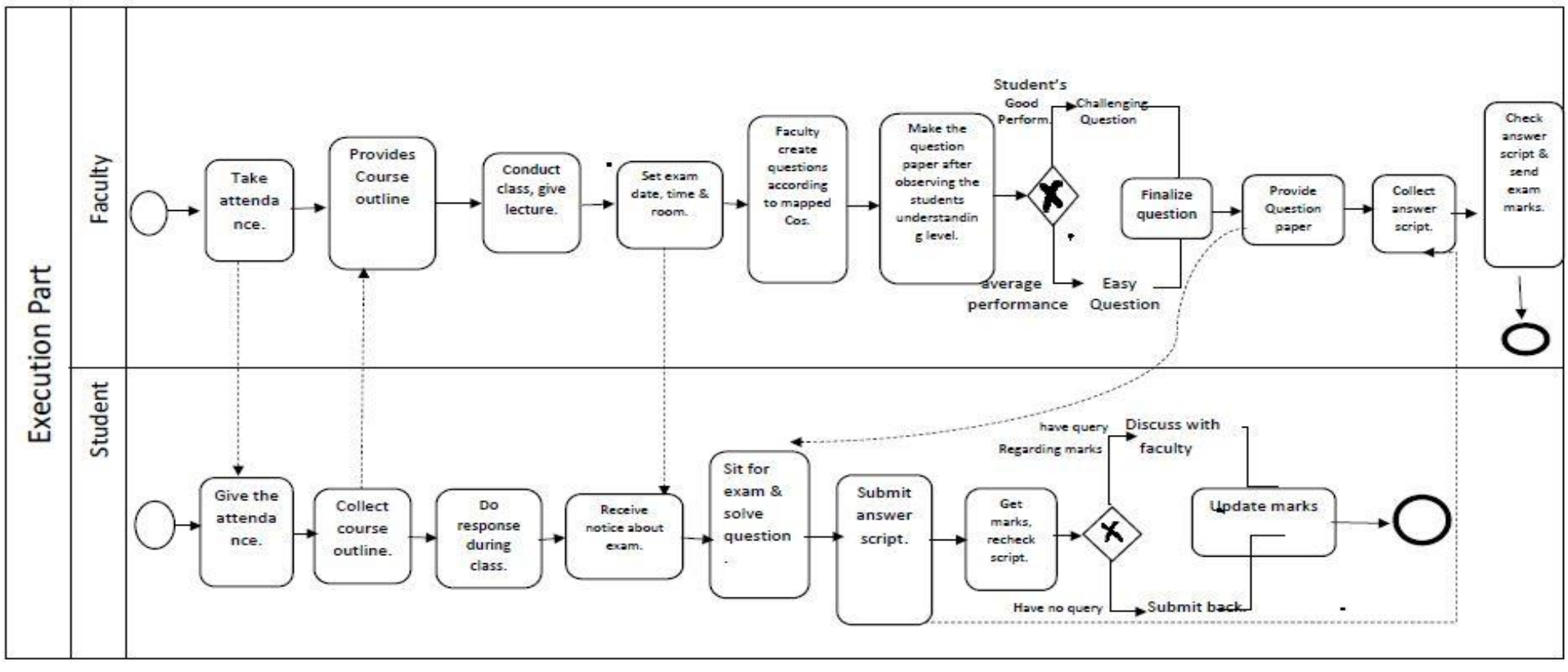
EXECUATION PART	**Faculty** 2.1)Faculty takes attendance manually of which students are enrolled in course. 2.2) Faculty provides course outline (Syllabus) to students. 2.3) Faculty conduct classes & give lecture according to syllabus. 2.4) Faculty observes students' performance 2.5)Faculty evaluate their understanding level & responses and make the questions easy or challenging based on that. 2.6) Faculty creates questions according to mapped COs on respective question based on Bloom taxonomy for assessment tools. 2.7) Then give a particular time, date for an exam, and manage a classroom. 2.8) Faculty invigilates during exam time. 2.9) After getting the script faculty will ensure standard of that answer fulfill the requirement Of that particular Cos. 2.10) Then faculty checks script and give marks. **Students** 2.1) Students enroll in a particular course. 2.2) Students give the attendance in class. 2.3) Collect the course outline (Course description, marks distribution, syllabus &	**Marker & Whiteboard** 2.3) Faculty use marker to deliver his lecture during class time. **Duster** 2.3) To remove writing on whiteboard, faculty use duster. **Pen & paper** 2.4) Faculty use pen to solve problem in class when students show him their exercise problem in theirs note. 2.6) Faculty use pen and paper to make draft question. 2.8) During exam time, Faculty use pen and paper to sign on answer script 2.10) Faculty use pen & paper when he checks answer script. 2.4) Students use pen and paper in class to note important lecture's part. 2.6) During examination, Students use pen and paper in their answer script to write answers. **Compass, Ruler and other stationary** are used for drawing diagrams. **Chair and table** To attend in class and exam faculty &	**Computer and Laptop** 2.6) Typing the question require a computer for coding or open book exam **Calculator** 2.6) Some exams require of use it to calculation **Printer and photocopy Machine** 2.6) Faculty use it for printing the question	**Microsoft word** • Faculty use it to make class task for student. • Student use it to do assignment **Operating system** Any OS windows, Mac. **Adobe Reader** For Viewing the lecture slide And question paper format.	**Microsoft excel.** 2.10) Faculty use to list down student assessment marks. 2.8) Students use excel to open test's marks.	**Internet** • Faculty use internet to communicate with students anytime besides class time. • Faculty use internet to send assignment or homework via online. • Students use internet to view assignments. • Students use it to do group work via online for project purpose or group study

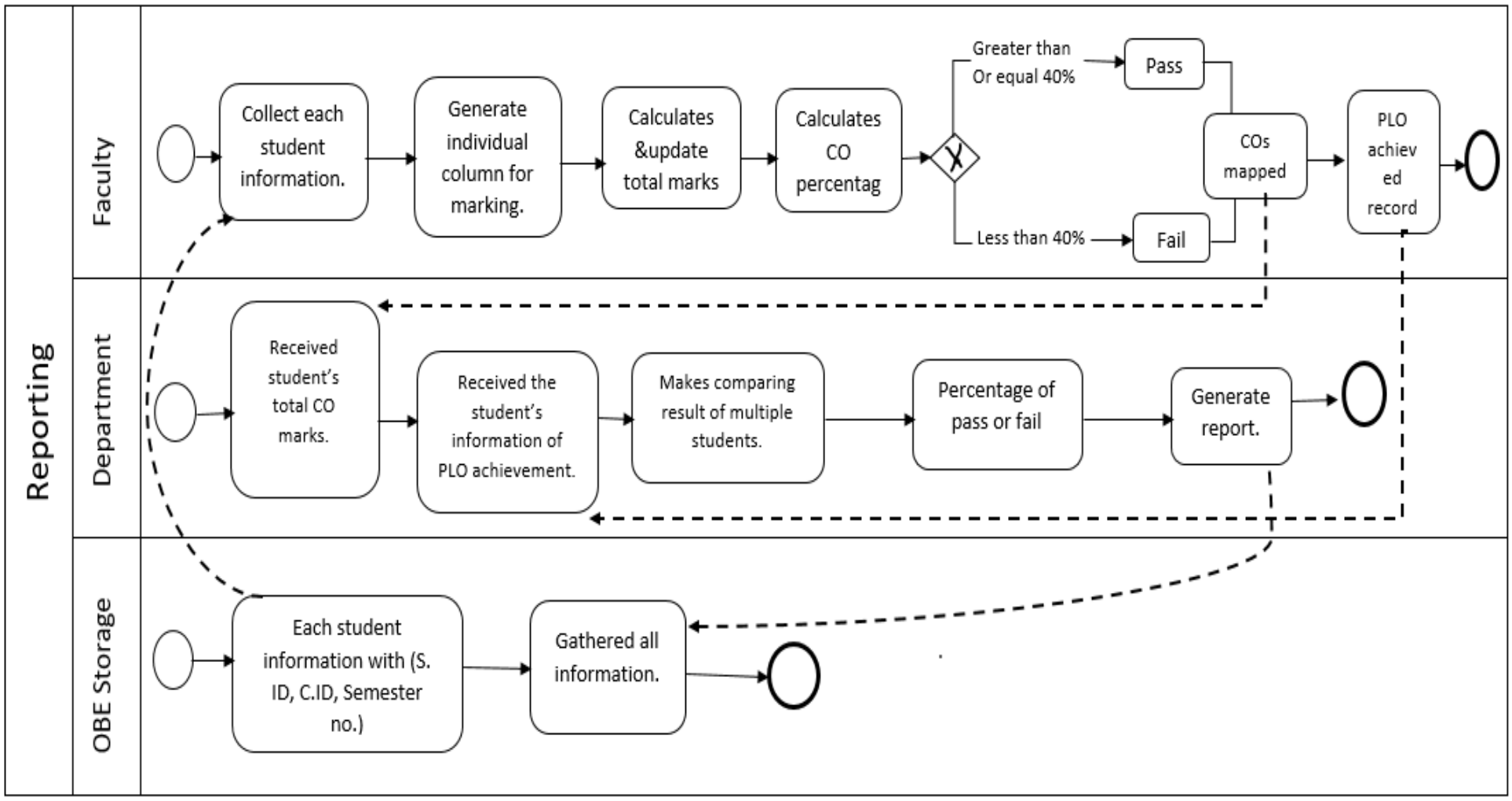
	assessment date etc.) from faculty. 2.4) Students do response during class time and ask questions if they face any issues. 2.5) Students sit for exam. 2.6) Solve the questions. 2.7) Submit the answer script back to the faculty. 2.8) Students can see their scripts marks and recheck if they have any query regarding their marks.	students use these. **Classroom** Faculty and students use classroom for a space for conducting class and exams. **Stapler** For attaching all the extra paper, rough work and answer.				
Reporting	**Faculty :** 1. Faculty collects each student information(student's Id, Course ID, Section, current Semester) from OBE storage 2. Faculty generate individual columns for quiz, Assessment, midterm, final ,project, total marks and grade based on the COs and the Questions mapping. 3. The faculty calculates and updates the total mid, final and project marks. then store the total and individual marks based on COs for each Student. 4. Calculates CO percentage. If the marks greater than or equal to 40% CO's have been achieved, a student passes otherwise fails.	none	**Computer:** 1. All related data is searched and stored using computer	**MS Excel:** 1. Create a table 2. Create columns for quiz, Assessment, midterm, final ,project, total marks and grade. Google Mail a)Faculty sends the CQI report to the department.	**Other Sources:** 1. All related information are stored in the specific location.	**Internet:** 1. All related data are provided through the internet

	5. Faculty will Provide students total CO marks achieved in all CO's in mid-term, final & project to the department/OBE storage including student id, course id, section, semester Department 1. Department makes reports after comparing results of multiple students. Percentage of successfully passed or failed to achieve are calculated based on the total number of students Department: 1.Create report for the student information of PLOs achievement based on COs. 2. All information store to the OBE storage.					

PROCESS DIAGRAM (AS IS)


Planning
BAETE
Store Accreditation manual to the website
OBE Storage
Store the accreditation manual.
Store the updated manual.
Department
Collect accredit ation manual.
Send the accreditatio n manual.
Select courses for mapping.
List down COs
Create initial planning of mapping PLO & CO.
Send the planning PLO & CO, previous course outline &assessment plan.
Collect updated mapping of PLO & CO.
Faculty
Collect accreditation manual, initial planning of mapping PLO & CO, previous course outline & course assessment.
Update PLO & COs and send it.
Create course outline & assessment plan.


Execution Part
Faculty
Take attenda nce.
Provides Course outline
Conduct class, give lecture.
Set exam date, time & room.
Faculty create questions according to mapped Cos.
Make the question paper after observing the students understandin g level.
Student's Good Perform.
Challenging Question
average performance
Easy Question
Finalize question
Provide Question paper
Collect answer script.
Check answer script & send exam marks.
Student
Give the attenda nce.
Collect course outline.
Do response during class.
Receive notice about exam.
Sit for exam & solve question .
Submit answer script.
Get marks, recheck script.
have query Regarding marks
Discuss with faculty
Update marks
Have no query
Submit back



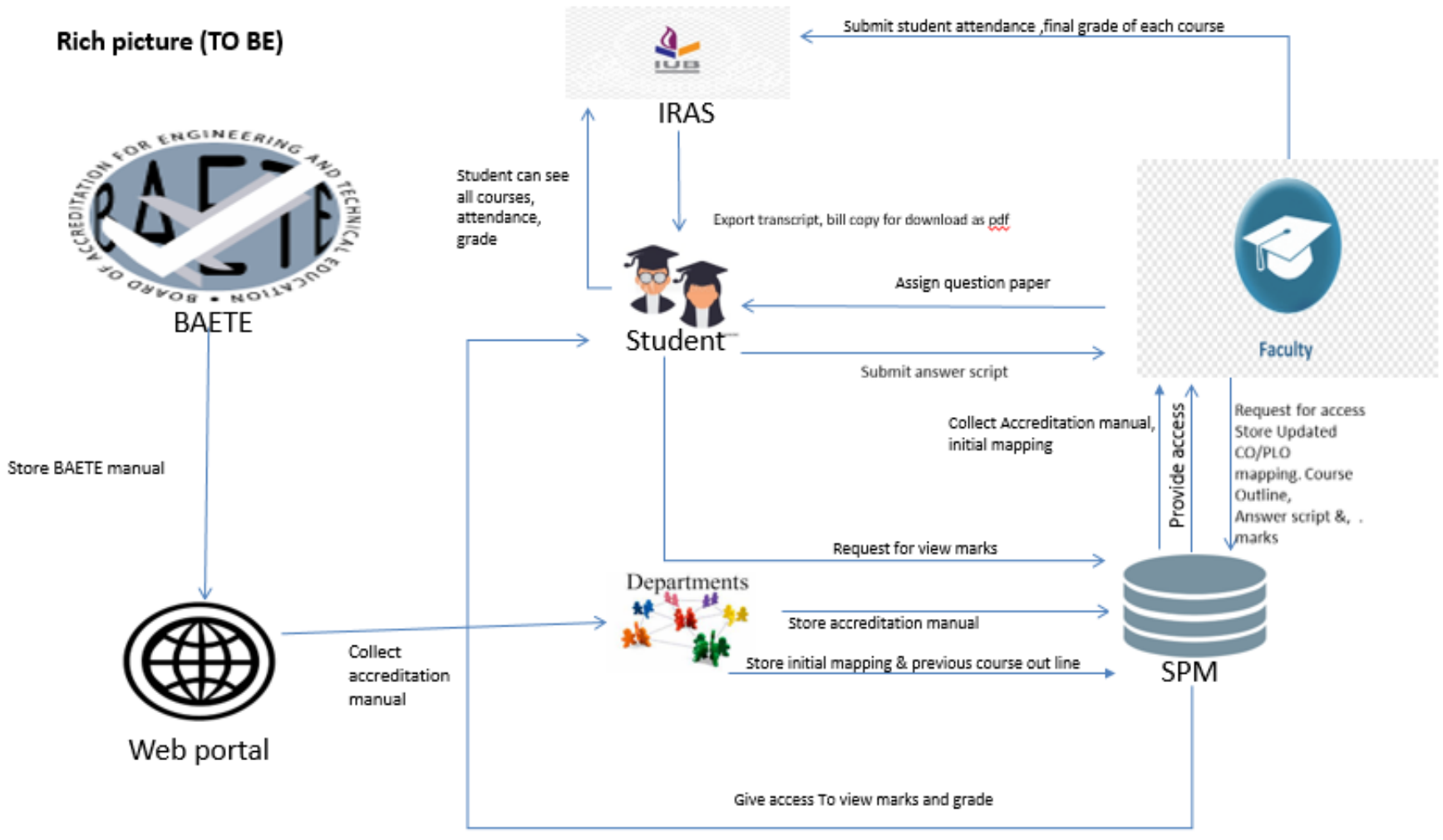
PROBLEM ANALYSIS:

Process Name	Stakeholders	Concerns (Problems)	Analysis (reason of the problem)	Proposed solution
1.Planning	1.1) Department 1.2) Faculty	1.1 Department member has to do initial mapping manually. 1.2 Faculty has to collect manually all the require information from department like initial mapping of PLO/CO to update CO/PLO. 1.3 Faculty has to create the assessment report & course outline and map	1.1 Collecting all the PLO & Map PLO with course manually which is very difficult, and this process is very time consuming. 1.2 Faculty has to request & wait until the department provide the accreditation manual & initial planning of mapping. 1.3 Faculty sent assessment report & course outline then wait for the Department's approval	We will create an automated system to address concern 1.1 • Provide weight to PLO & course. • As per weight of PLO & Course automatically map the Course with the PLO. • Adjust the map based on the number of PLO for a course or number of COs for a PLO. We will create a system to address concern 1.2 • By providing faculties direct access to database for collecting the Accreditation manual & and the initial mapping of CO/PLO. We will create an automated system to address concern 1.3 • Faculty will store the prepared course outline and

		every question with CO then they have to send it to department manually.	which is very time consuming for both faculty and department.	assessment report to the system where department can view and change it and give feedback.
2.Executing	2)Faculty	2.1) Faculty take attendance manually and submit the hard copy to the Register office Then register office store the Data, so time consumption and delay are prime limitation. 2.2) Calculating each students attendance manually process is time consuming as well as Faculty must focus on Automatically withdraw issue. 2.3) Faculty request for previous semester's question paper and other assessment to Department to create standard question and give an idea to students about question paper. As this whole process happens manually, so this is a lengthy process and time consumption is also an issue. 2.4) Faculty check the answer script manually, so it takes time. During checking answer script Faculty can do mistake unconsciously.	2.1) Here, as sending hard copy to the register office involve multiple process so there is a possibility of losing it. 2.2) As there is a rule that a student of below 70% attendance will automatically get withdrawal, therefore, when faculty take attendance manually, it will be hassle and time-consuming to calculate each student present. 2.3) In this process, Department need to store each semester's question paper and it is hard to collect hard copy of every semester's question paper and other assessment. It will not be informative because faculty and student cannot collect it easily. 2.4) Faculty check answer script manually, Hence, it is time consuming and has probability to do mistake	2.1) We can create a system (IRAS) to take attendance in online so that it can automatically store attendance database. 2.2) Take attendance in automated generate systems and it automatically store and calculate attendance give withdraw issue. 2.3) Create a question bank and upload every semester's question paper in that data base so that faculty and students can easily get it. 2.3a) Faculty upload questions. 2.3b) Faculty get previous questions. 2.3c) Students can get questions paper. 2.4) So, create an automated generate system which can evaluate answer script automatically. 2. 4.a) If faculty does mistake to choose right option in automated generate system, he can reassign right answer.

Reporting	1.Faculty 2.Department	1. faculty Calculates the Students marks manually 2. COs percentage are calculated manually by using MS excel 3. Manually checking student pass or fail 4. Manually Recording PLOs achievement 5. Manually creating reports for the student performance achievement based on COs	1. There have no specific system which can automatically calculate CO percentages and determine whether the COs and PLOs have been achieved or not. 2. The process is time consuming since it takes time for the manually calculates students marks, manually checking students pass or fail, manually recording PLOs achievement and manually creating overall report	The features that are available to: 1)A faculty can login into the system and perform any of the operation options -can view student information - can view Co percentage. - can see PLO achievement for each student. 2) Department can login into the system and perform any of the operation options -can search each student by using their student id. -can edit student information. . The system prompts for the student detail from one the above keys. . The students details are displayed on the screen.

RICH PICTURE (TO BE)



PROCESS ALONG WITH SIX ELEMENT ANALYSIS (TO BE)

TO BE						
Process	**Human**	**Non-Computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Network & Communication**
1.Planning	**BAETE:** 1.1) Store Accreditation Manual with defined PLOs to the web portal. **Department:** 1.1) Department Collect Accreditation manual & store it to SPM. 1.2) Department will make an initial planning of the mapping department will use the system where an automated mapping system which is already designed. Department will use the system to make initial planning of CO/PLO. 1.3) department sent initial mapping of Cos/PLO sent previous course outline and assessment plan to the SPM 1.4) department will collect updated PLO & course out line and assessment plan from SPM. **Faculty:** 1.1) Received Accreditation Manual with defined PLOs & initial planning of mapping	**Pen & Paper:** 1.1) Department prints a hard copy of PLOs on the paper as back up. 1.2) Faculty print a hard copy of Assessment plan & course outline.	**Computer:** 1.1) BEATE uses it to upload the Accreditation manual to their website. 1.1) Department use to Collect Accreditation Manual which is stored on the website by the BAETE. 1.2) Department use it to create initial mapping of PLO/CO. 1.3) Department use it to store accreditation manual & initial mapping to the SPM 1.1) Faculty use it to collect a soft copy of Accreditation Manual with defined PLOs from SPM. 1.2) Faculty use it to collect data and previous course outline and assessment plan from SPM 1.4) Faculty use it to update initial mapping & to create course outline & assessment plan	**SPM:** 1.1) • Provide weight to PLO & course. • As per the weight of PLO/CO, automatically map the course with the PLO. • Adjust the map based on the number of PLO for a course or number of COs for a PLO. 1.1) To give faculty direct access to the database for collecting the Accreditation manual & and the initial mapping of CO/PLO. 1.3) By providing faculties direct access to the database so that he can directly update the mapping there. The department will log in from his account and update it there and time will be saved. 1.5) to store the prepared course outline and assessment report to the system where the department can view and change it and give feedback. **MS Word:**	**SPM database:** 1.1) Department use it to store Accreditation Manual with defined PLOs. 1.3) Department use it to Store mapped CO from PLOs based on Syllabus. 1.5) faculty use it to store assessment plan and course outline	**internet:** 1)BAETE used to store their manuals on their website. 2)Department use to collect manual from the web portal. 3)department its use for Communication with BAETE & faculty member via email. 4)To download Accreditation Manual as PDF. 5)to store updated Co mapping to the SPM. 6) Faculty use internet to collect every information to create course out line & updated PLO/COs 7) Download PLOs as PDF.

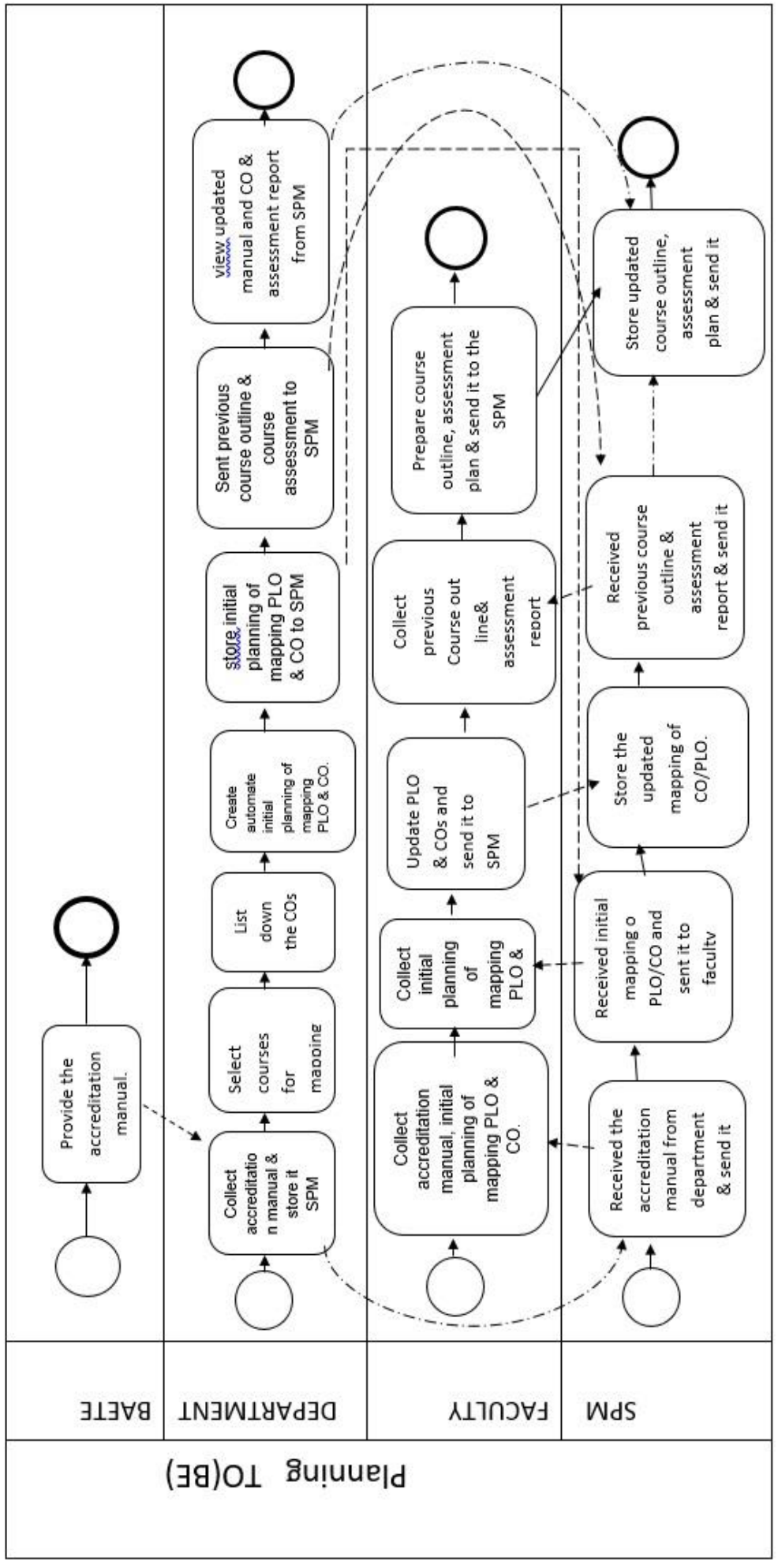
	PLO/CO from SPM. 1.2) faculty also received previous assessment plan & course outline data of making of assessment plan and course outline from the SPM. 1.3) Faculty member list Cos from course description & maps COs from PLOs based on the syllabus. 1.4) Based on COs from PLOs faculty create assessment plan & course outline. 1.5) sent an updated mapping of PLO/CO & assessment plan & course outline to the SPM.		**Printer:** 1.1) Department use it to print hard copy of accreditation manual. 1.4) Faculty use it to print out PLOs, course outline and assessment plan as hard copy.	1.4) Faculty to create assessment plan, course outline. **PDF Viewer:** 1.3) faculty used to view PDF file of PLOs. **Google classroom:** 1.4) Faculty use it to upload course outline and assessment plans.		
2.EXECUATION PART	**Faculty** 2.1) Faculty takes automated attendance of which students are enrolled in course. 2.2) Faculty upload course outline (Syllabus) in database. 2.3) Faculty conduct classes & give lectures according to syllabus & upload lecture slide-in system. 2.4) Faculty observes students' performance.	**Marker & Whiteboard** 2.3) Faculty use markers to deliver his lecture during class time. **Duster** 2.3) To remove writing on whiteboard, faculty use duster. **Pen & paper** 2.4) Faculty use pen to solve problems in class when students show him their exercise problem in their notes.	**Computer and Laptop** 2.6) Typing the question require a computer for coding or open book exam **Calculator** 2.6) Some exams require of use it to calculation **Printer and photocopy Machine** 2.8) Faculty use it for printing the question	**SPM** 2.1) Attendance can be taken via IRAS and store it in SPM. Attendance reports of each student will be stored in SPM. 2.7) Faculty upload question papers in every semester & get previous question papers from SPM. Students get question papers from SPM. 2.12) Faculty submit student's exam marks in SPM. **Microsoft word** 2.3) Faculty use it to make class task for students. 2.4) Students use to do assignments.	**Microsoft excel**. 2.12) Faculty use Microsoft excel to show students their assessment marks. 2.8) Students use excel to open marks	**Internet** • Faculty use the internet to communicate with students anytime besides class time. • Faculty use the internet to send assignments or homework in online. • Students use internet to view assignments. • Students use it to do group work via online for project purpose or group study

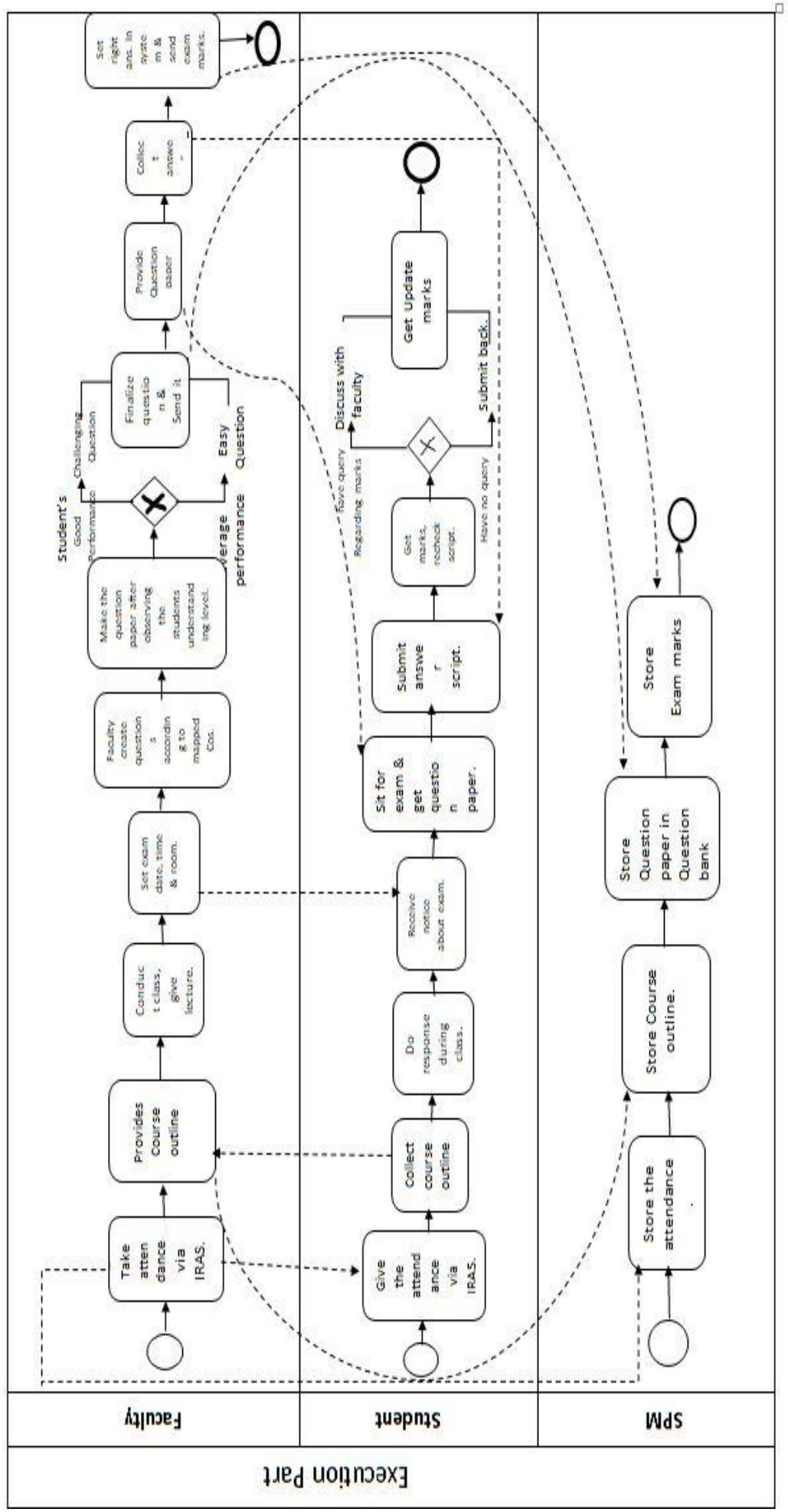
	2.5) Faculty evaluates their understanding level & responses and makes the questions easy or challenging based on that. 2.6) Faculty creates questions according to mapped COs on respective questions based on Bloom taxonomy for assessment tools. 2.7) Faculty store questions in database. 2.8) Then give a particular time and date for an exam and manage a classroom. 2.9) Faculty invigilates during exam time. 2.10) After getting the script faculty will ensure the standard of that answer fulfill the requirement Of that COs and set the right answer in system. 2.11) Faculty collect exam marks from automating generate evaluate answer script system . 2.12) Faculty show students their exam marks and then submit updated marks in the system. **Students** 2.1) Students enroll in a particular course.	2.6) Faculty use pen and paper to make draft questions. 2.9) During exam time, Faculty use pen and paper to sign on answer script 2.4) Students use pen and paper in class to note important lecture's part. 2.6) During examination, Students use pen and paper in their answer script to write answers. **Compass, Ruler and other stationery** are used for drawing diagrams in class or examination. **Chair and table** To attend in-class and exam faculty & students use these. **Classroom** Faculty and students use the classroom as a space for conducting class and exams. **Stapler** For attaching all the extra paper, rough work and answers.		**Operating system** Any OS maybe windows, Mac. **Adobe Reader** For Viewing the lecture slide And question paper format.		

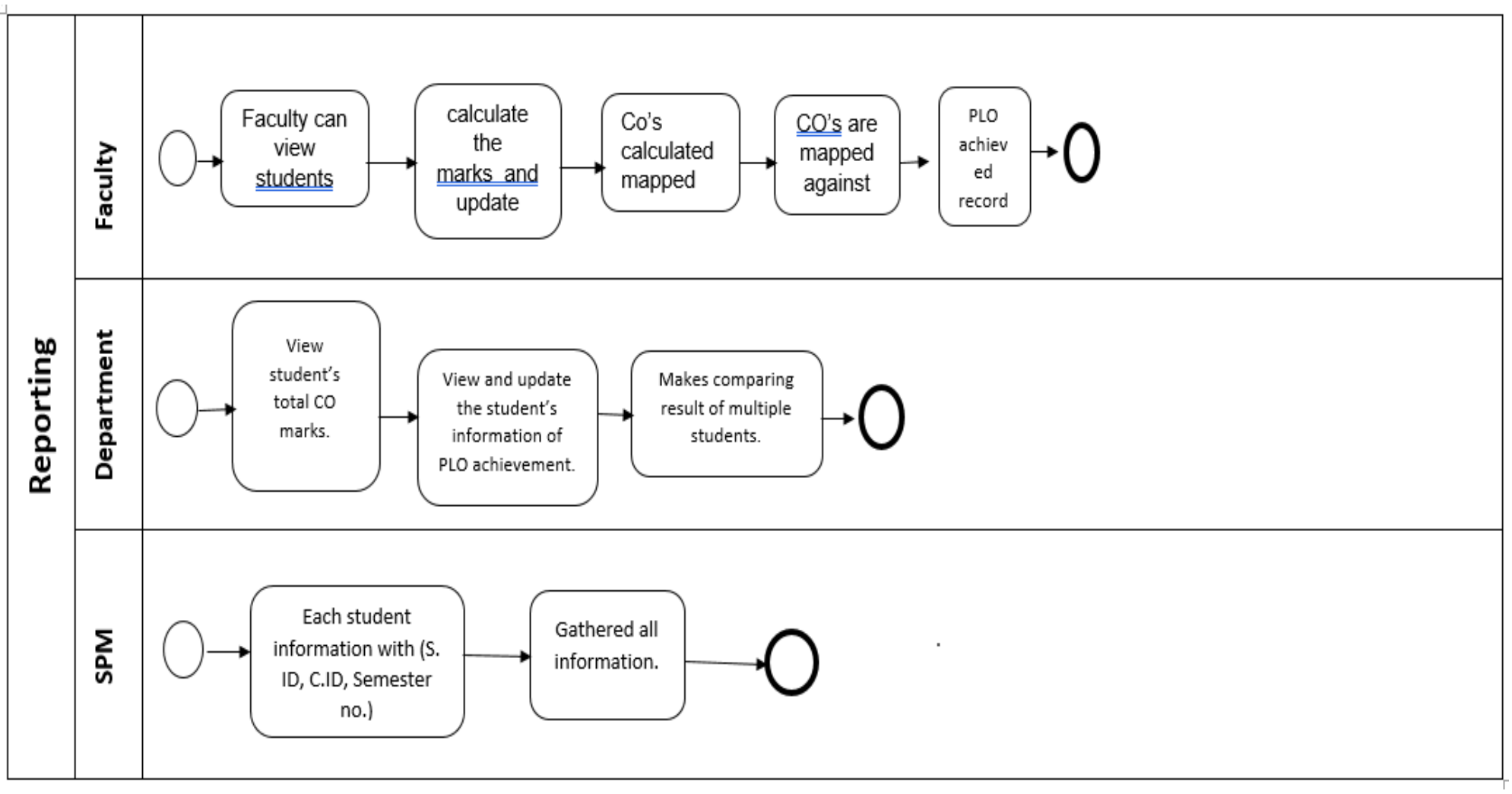
	2.2) Students give attendance in class via IRAS. 2.3) Collect the course outline (Course description, marks distribution, syllabus & assessment date, etc.) from the system. 2.4) Students do respond during class time and ask questions if they face any issues. 2.5) Students sit for exams. 2.6) Solve the questions. 2.7) Submit the answer script back to the faculty. 2.8) Students can see their script marks and recheck if they have any queries regarding their marks.					
3.Reporting	**Faculty** 3.1) Faculty can view students information by using students id, course id, section, semester. 3.2) Then faculty will calculate the marks and update grades of each student. 3.3) COs percentage will calculate and checking If greater than or equal to 40% CO's have been achieved, a	none	**Computer:** a) Computer Used to store plo and co scores ,grade of the student. b) All related data is searched and stored using computer a) All related data is searched and stored using compute	**MS Excel:** a. Create a table b. Create columns for quiz, Assessment, midterm, final ,project, total marks and grade Google Mail a. Department received the CQI report from the faculty .	**SPM Database** a) Will contain all data regarding COs and PLOs for individual students. b) A system can be introduced which can calculate CO percentages automatically with the marks provided as an input .The system will show by what percentages the PLOs and COs have been achieved and also all the COs and PLOs that the student failed to achieve.	**Internet:** 1. All related data are provided through the internet

	student passes that certain CO otherwise fails. 3.4)CO's are mapped against PLO's and PLO achievements are recorded. **Department**: 3.1)can view the student information of PLO achievement . 3.2)make reports after comparing results of multiple students 3.3) Percentage of successfully passed or failed to achieve are calculated based on the total number of students and store to the SPM				Other Sources: 1. All related information are stored in the specific location	

Process Diagram (TO BE)


Planning TO(BE)
BAETE
DEPARTMENT
FACULTY
SPM
Provide the accreditation manual.
Collect accreditation manual & store it SPM
Select courses for mapping
List down the COs
Create automate initial planning of mapping PLO & CO.
store initial planning of mapping PLO & CO to SPM
Sent previous course outline & course assessment to SPM
view updated manual and CO & assessment report from SPM
Collect accreditation manual, initial planning of mapping PLO & CO.
Collect initial planning of mapping PLO & CO
Update PLO & COs and send it to SPM
Collect previous Course out line& assessment report
Prepare course outline, assessment plan & send it to the SPM
Received the accreditation manual from department & send it
Received initial mapping o PLO/CO and sent it to faculty
Store the updated mapping of CO/PLO.
Received previous course outline & assessment report & send it
Store updated course outline, assessment plan & send it


Execution Part
Faculty
Student
SPM
Take atten dance via IRAS.
Provides Course outline
Conduc t class, give lecture.
Set exam date, time & room.
Collec t answe r
Set right ans. in syste m & send exam marks.
Give the attend ance Via IRAS.
Collect course outline
Do response during class.
Receive notice about exam.
Sit for exam & get questio n paper.
Submit answe r script.
Get marks, recheck script.
Have no query
Have query Regarding marks
Discuss with faculty
Submit back.
Get Update marks
Store the attendance .
Store Course outline.
Store Question paper in Question bank
Store Exam marks
Provide Question paper
Finalize questio n & Send it
Student's Good performance
Challenging Question
Average performance
Easy Question
Make the question paper after observing the students understand ing level.
Faculty create question s accordin g to mapped Cos.


Reporting
Faculty
Faculty can view students
calculate the marks and update
Co's calculated mapped
CO's are mapped against
PLO achieved record
Department
View student's total CO marks.
View and update the student's information of PLO achievement.
Makes comparing result of multiple students.
SPM
Each student information with (S. ID, C.ID, Semester no.)
Gathered all information.

Chapter 3: Logical System Design

Business Rule

A university must have one or many School. A School is belongs to exactly one university.

A school must have one or many department. A department belongs to exactly one school.

A department must have many faculty. A faculty is managed by exactly one department A faculty may be a Department head or not but a department head must be a faculty.

A program is enrolls many student. Each student is enrolled by exactly one program. A PLO is contain by exactly one program. A program must consist of many PLOs.

A faculty must have many marksheet to evaluate the students. A marksheet must be evaluate by exactly one faculty. A enrollment belongs to exactly one marksheet. A marksheet exactly have one enrollment.

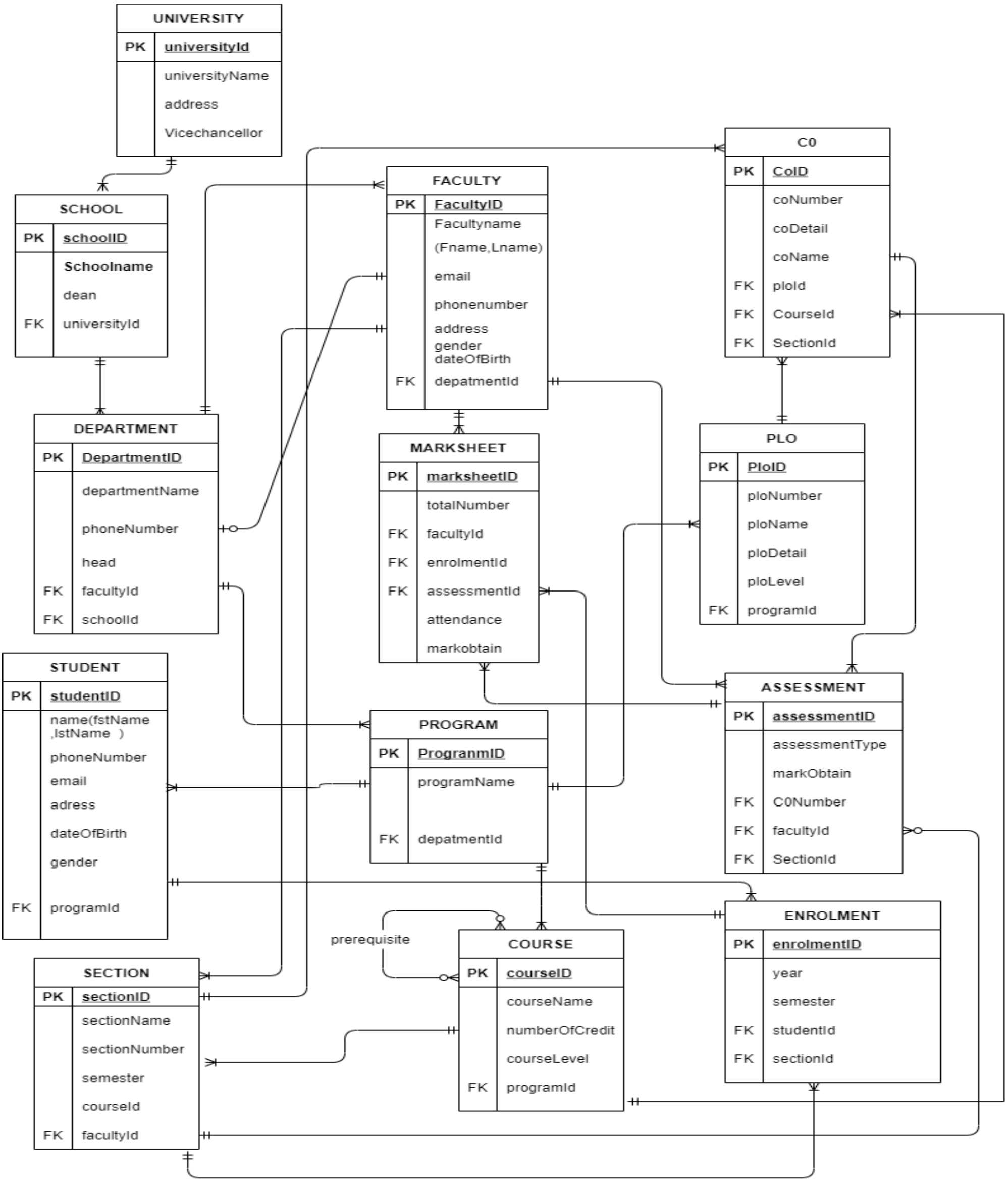
A faculty is assign to many section. A section must have exactly one faculty. A course is assign to many section. A section must contains exactly one course. A section may assign many assessment. Each assessment is assign by exactly one section. A section must have many cos. Cos must be under exactly one section.

A student have many enrollment. A enrollment must have exactly one student. A section must have many enrollment. A enrollment must have many section.

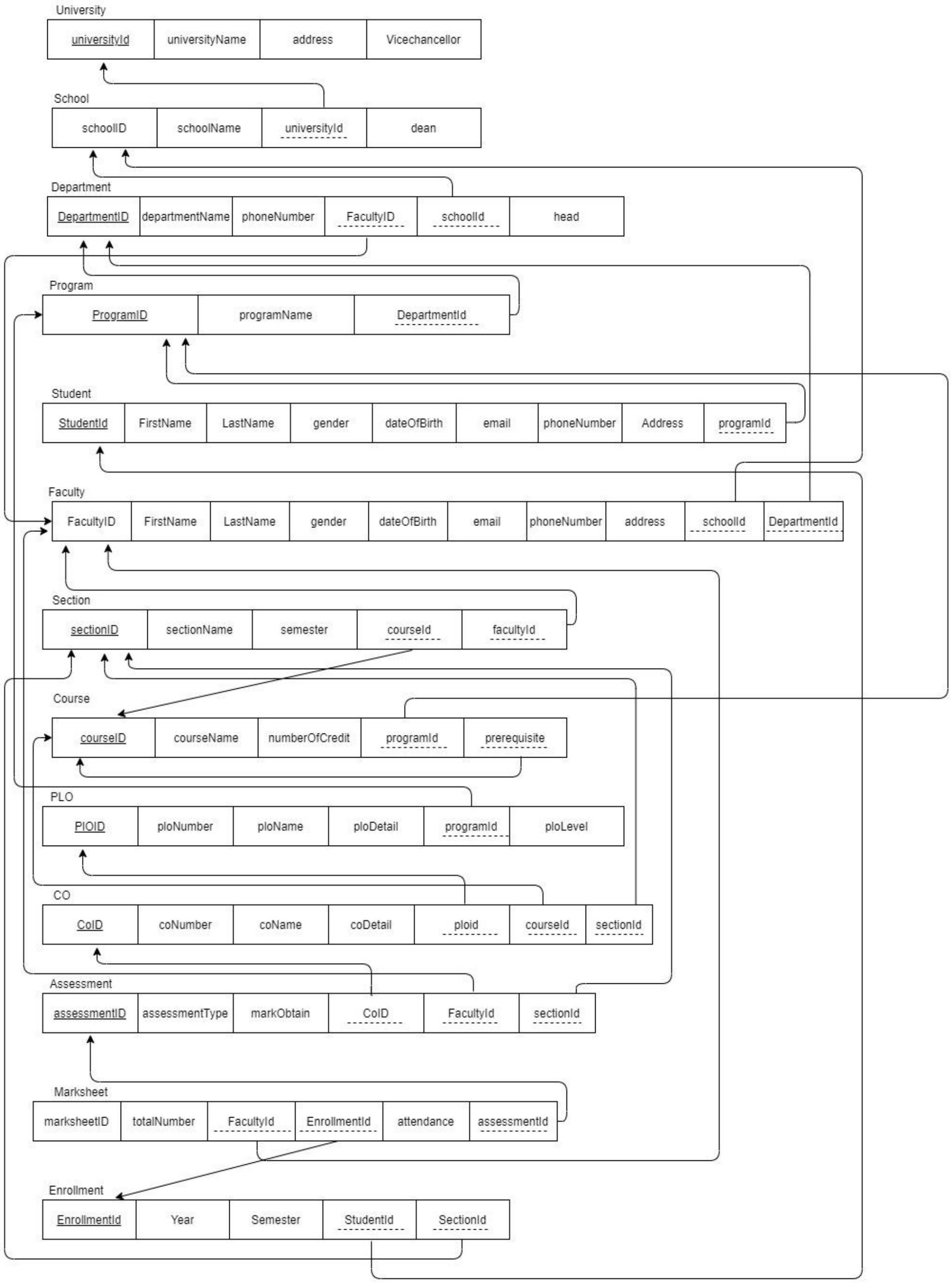
A assessment is assign to many student. Each student must receive many assessment. A assessment must have multiple Cos. A CO is belongs to exactly one assessment.

A CO is belongs to exactly one course. A course must have many Cos. A CO is belongs to exactly one PLO. A PLO must have many COs.

Entity Relationship Diagram



Entity Relationship Diagram to Relational Schema


University
universityId
universityName
address
Vicechancellor
School
schoolID
schoolName
universityId
dean
Department
DepartmentID
departmentName
phoneNumber
FacultyID
schoolId
head
Program
ProgramID
programName
DepartmentId
Student
StudentId
FirstName
LastName
gender
dateOfBirth
email
phoneNumber
Address
programId
Faculty
FacultyID
FirstName
LastName
gender
dateOfBirth
email
phoneNumber
address
schoolId
DepartmentId
Section
sectionID
sectionName
semester
courseId
facultyId
Course
courseID
courseName
numberOfCredit
programId
prerequisite
PLO
PlOID
ploNumber
ploName
ploDetail
programId
ploLevel
CO
CoID
coNumber
coName
coDetail
ploid
courseId
sectionId
Assessment
assessmentID
assessmentType
markObtain
CoID
FacultyId
sectionId
Marksheet
marksheetID
totalNumber
FacultyId
EnrollmentId
attendance
assessmentId
Enrollment
EnrollmentId
Year
Semester
StudentId
SectionId

Normalization:

Since we made Relations from ERD based on the theory. In that case, we think we don't need of any kind of normalization here. However, we have tested each of the relation separately to test normalization and those perfectly fulfill the requirement of normalization. That's why relations we are received in 1NF, 2NF and 3NF all are in normal form.

Data Dictionary

UNIVERSITY_T

Name	Data type	Size	Remark
universityId	VARCHAR	5	This is the primary key of the University. Example: IUB
universityName	VARCHAR	50	This is the name of the University. Example: "Independent University, Bangladesh.
Vice chancellor	VARCHAR	50	This is the name of the vice chancellor.
Address	VARCHAR	100	This is the permanent address of the University. example: Plot 16, Block B Aftabuddin Ahmed Road Bashundhara R/A, Dhaka-1212, Bangladesh

DEPARTMENT_T

Name	Data type	Size	Remark
departmentId	VARCHAR	5	This is the primary key of the Department. Example: CSE
departmentName	VARCHAR	200	This is the name of the Department. Example: "Computer Science and Engineering"
phoneNumber	VARCHAR	15	This is the Number of the Department
Head	VARCHAR	150	This is the part of the Head.
schoolId	VARCHAR	5	This is the Foreign Key of the School table Example: "SETS"
facultyId	VARCHAR	10	This is the foreign key of the Faculty Table example : 156

PROGRAM_T

Name	Data type	Size	Remark
programID	VARCHAR	5	This is the Primary Key for a Program Example: "CSE".
programName	VARCHAR	30	This is the name of Program. Example: Bachelor Of Science

departmentID	VARCHAR	5	This is the Foreign Key from the Department table. Example: “CSE”

Student_T

Name	Data type	Size	Remark
studentId	VARCHAR	7	This is the primary key of the Student Table : 1710000
name(fname,lname)	VARCHAR	100	This is the first & last name of the student. example: muhammad asif Hasan
phoneNumber	VARCHAR	15	This is the phone number of the student. Example:01875050
email	VARCHAR	200	This is the email address of the student. Example: “1710000@iub.edu.bd”
Password	VARCHAR	250	This is the Password of the student.
address	VARCHAR	40	This is the permanent address of the student. example: H:1, road 7 Block A Basundhara R/A
dateOfBirth	DATETIME	DD-MM-YYYY	This is the date of birth of the student. Example: “01-01-1998”
gender	VARCHAR	10	This is the gender of the student. Example: “M”
programId	VARCHAR	5	This is the Foreign Key from Program table Example: ”CSE”.

Faculty_T

Name	Data type	Size	Remark
facultyId	VARCHAR	10	This is the primary key of the Faculty Table example : 156
facultyName(Fname,Lname)	VARCHAR	100	This is the first & last name of the Faculty. example: muhammad Nasir hossain
email	VARCHAR	200	This is the email address of the Faculty. Example: “Mnasir@iub.edu.bd”
phoneNumber	VARCHAR	15	This is the phone number of the Faculty. Example:01715345673
address	VARCHAR	50	This is the permanent address of the Faculty.example: H:6, road:10 Block:D Basundhara R/A
gender	VARCHAR	10	This is the gender of theFaculty. Example: “M”
dateOfBirth	DATETIME	DD-MM-YYYY.	This is the date of birth of the Faculty

			Example: “01-01-1988”
depatmentId	VARCHAR	10	This is the foreign key From the Department Table. Example:”CSE”

CO_T

Name	Data type	Size	Remark
CoId	INTEGER	5	This is the Primary Key for Course Outcome. Example:Co1
CoNumber	INTEGER	5	This is the number of the Course Outcome. Example: ”1”
coDetail	VARCHAR	50	This is the detail of Co.
coName	VARCHAR	100	This is the name of the Co.
ploId	VARCHAR	5	This is the foreign key from the Program Learning Outcome table. Example: “PLO1”
sectionId	INTERGER	5	This is the foreign Key for Section
courseId	INTERGER	5	This is the foreign key of the Course.

PLO_T

Name	Data type	Size	Remark
ploId	VARCHAR	5	This is the primary key for Program Learning Outcome. Example: “PLO1”
ploNumber	VARCHAR	5	This is the number of Program Learning Outcome. Example: 2
ploName	VARCHAR	100	This is the name of the Plo. Example:
ploDetail	VARCHAR	50	This is the details of the Program Learning Outcome. Example: knowledge, techniques, skills.
ploLevel	VARCHAR	200	This is the plolevel of the Plo.
programId	VARCHAR	5	This is the foreign key from Program table. Ex: Bsc

Marksheet_T

Name	Data type	Size	Remark
marksheetID	VARCHAR	5	This is the primary key of the Marksheet.

			.
totalNumber	FLOAT	5	This is the total number of the student. Example: 65.5
facultyId	VARCHAR	10	This is the foreign key from Faculty table
enrollmentId	INTEGER	5	This is the foreign key from enrolment table
assessmentId	INTEGER	5	This is the foreign key from assessment table.
attendance	INTEGER	26	This is the attendance from mark sheet table.

Course_T

Name	Data type	Size	Remark
courseId	VARCHAR	5	This is the primary key of the Course.
courseName	VARCHAR	200	This is the name of the course. Examle:
numberOfCredit	INTEGER	5	This the credit of thr course. Example:3
programId	VARCHAR	5	This is the Foreign Key from Program table Example: ”CSE”.
perquisite	VARCHAR	20	This is the unary relationship.

Assessment_T

Name	Data type	Size	Remark
assessmentID	INTEGER	5	This is the primary key of the Assessment.
assessmentType	VARCHAR	5	This is the type of the assessment. Example: quiz,mid,final
obtainMark	FLOAT	5	This is the obtain mark in the assessment.example:mid : 29 Final:50
C0_ID	INTEGER	5	This is the forigen key from the CO table.
faculty_Id	VARCHAR		This is the forigen key from thefaculty table.

Section_T

Name	Data type	Size	Remark

assessmentID	INTEGER	5	This is the primary key of the Assessment.
assessmentType	VARCHAR	20	This is the type of the assessment. Example: quiz,mid,final
obtainMark	FLOAT	5	This is the obtain mark in the assessment.example:mid : 29 Final:50
coId	INTEGER	5	This is the forigen key from the CO table.
facultyId	VARCHAR	10	This is the forigen key from the faculty table.
SectionId	INTEGER	5	This is the Foreign Key for Section

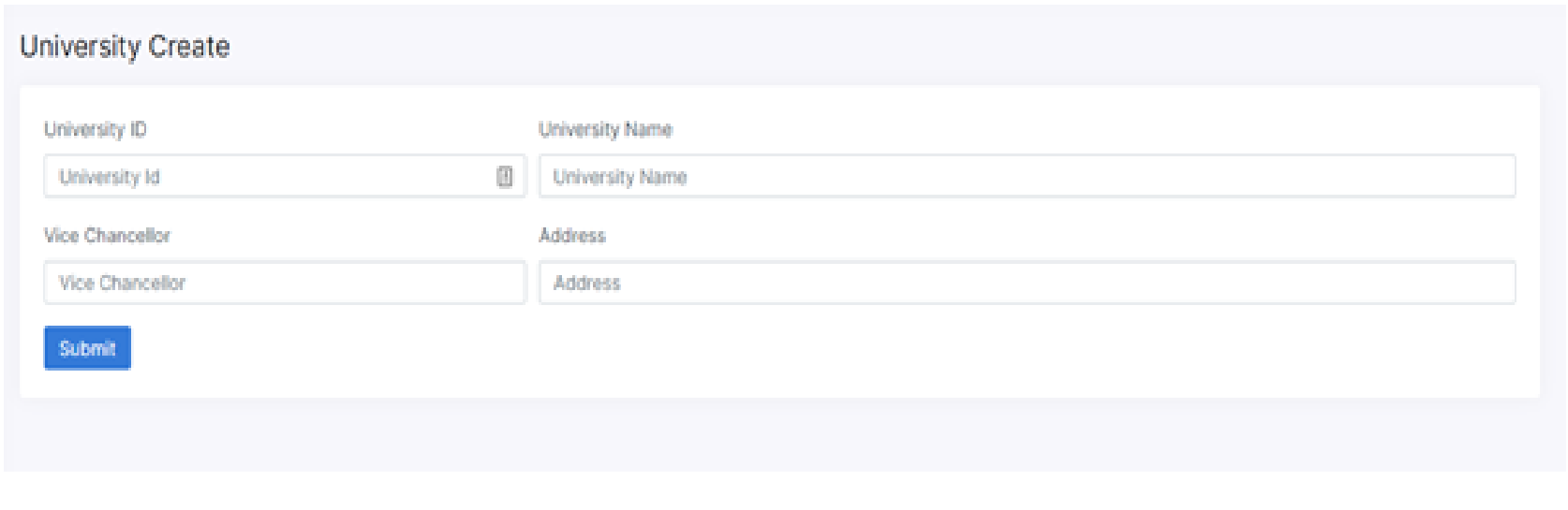
Enrollment_T

Name	Data type	Size	Remark
enrollmentID	VARCHAR	5	This is the Primary Key for Enrolment.
year	YEAR	YYYY	This is the year of the Enrolment.
semester	VARCHAR	20	This is the semester of the enrolment.
Student_Id	INTEGER	7	This is the foreign key from the student table.
Section_Id	INTEGER	5	This is the foreign key from the section table.

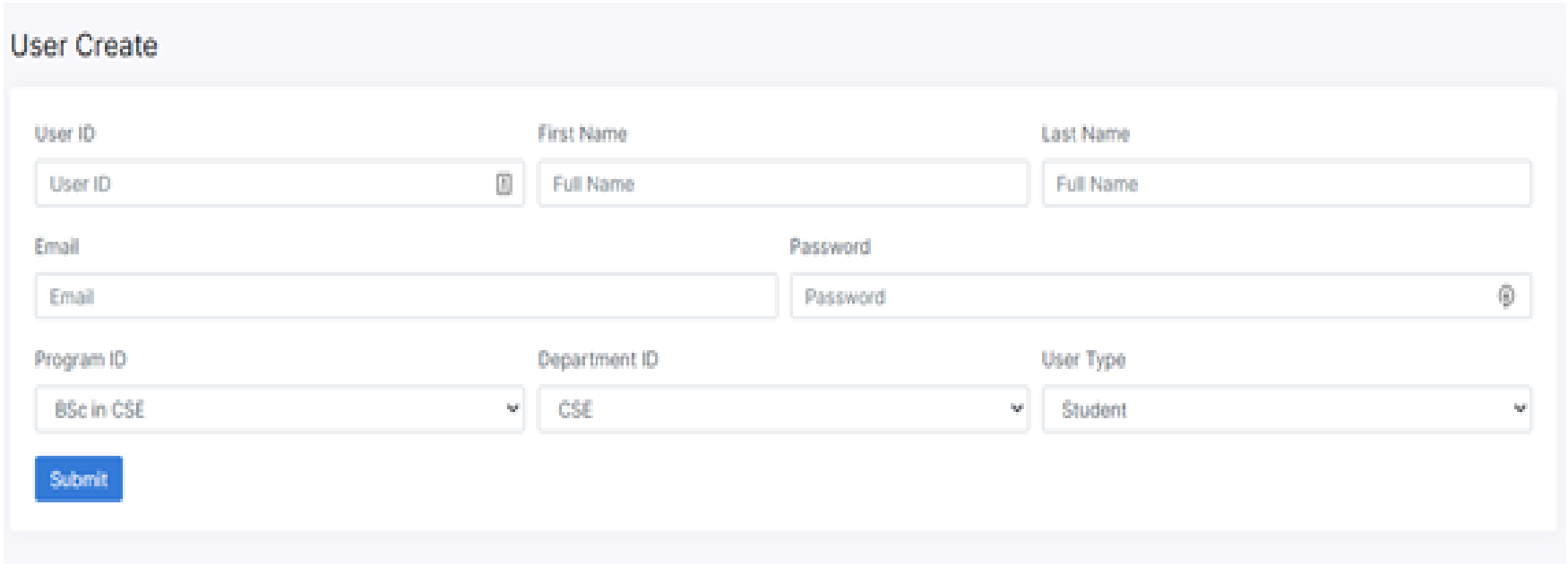
Chapter 4

Physical System Design

Input Forms

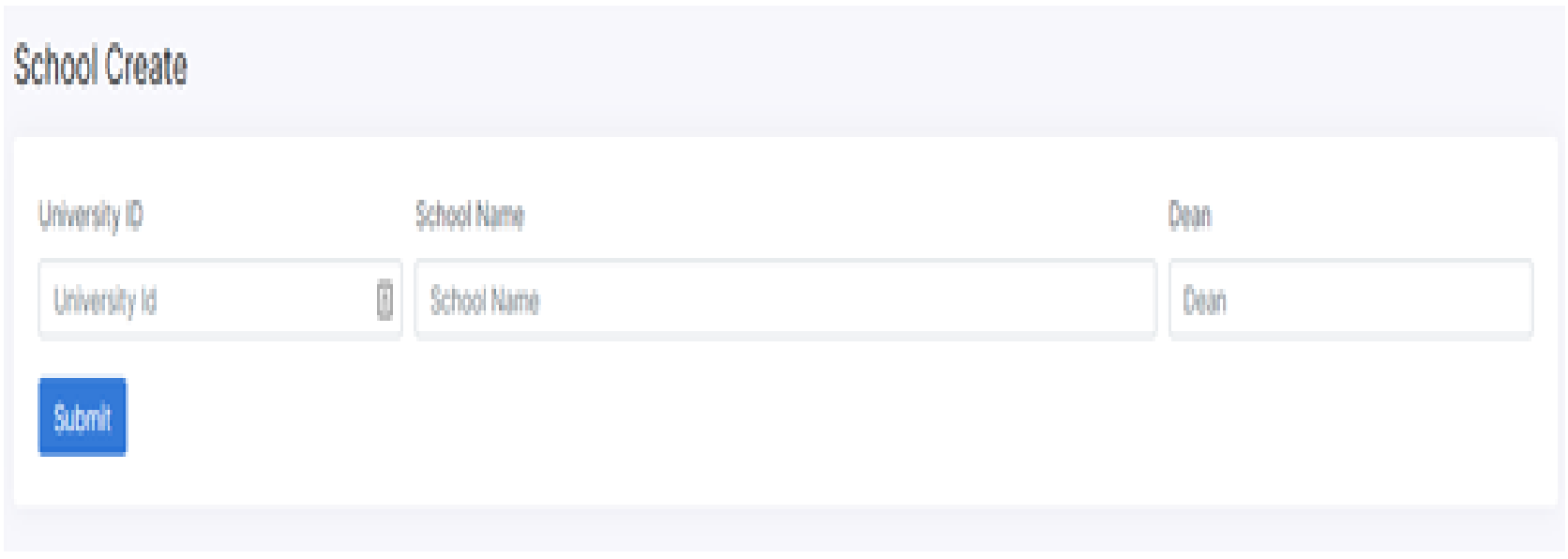



```
<?php
    require '../../database/mysql.php';
    $university_id = strtoupper($_POST['university_id']);
    $university_name = $_POST['university_name'];
    $vice = $_POST['vice'];
    $address = $_POST['address'];

    $sql = "INSERT INTO university (university_id, university_name, vice, address)
            VALUES('$university_id', '$university_name', '$vice', '$address')";

    if($mysql->query($sql)){
        header("Location: ../universities.php");
    }else{
        header("Location: ../university-create.php");
    }
?>
```



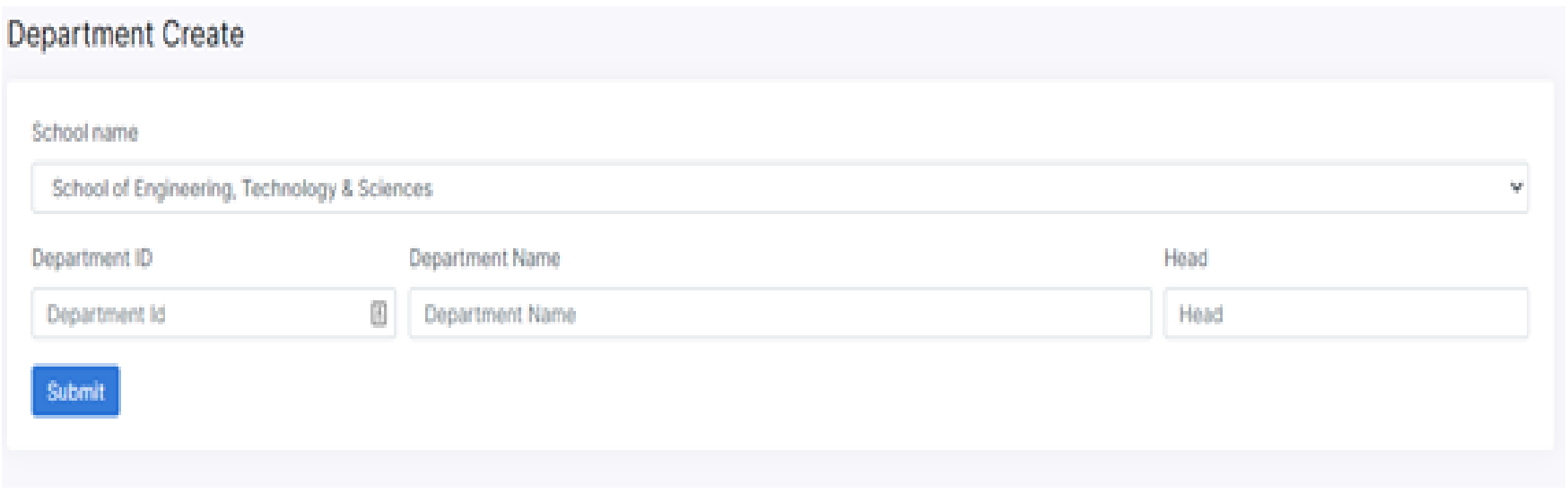
<?php

```
require '../../database/mysql.php';
$user_id = $_POST['user_id'];
$first_name = $_POST['first_name'];
$last_name = $_POST['last_name'];
$email = $_POST['email'];
$password = $_POST['password'];

$program_id = $_POST['program_id'];
$department_id = $_POST['department_id'];

$user_type = $_POST['user_type'];

if($user_type == "student"){
   $sql = "INSERT INTO student (student_id, first_name, last_name, email, password, program_id)
         VALUES($user_id, '$first_name', '$last_name', '$email', '$password', '$program_id')";
      $mysql->query($sql);
   }else{
      $sql = "INSERT INTO faculty (faculty_id, first_name, last_name, email, password, department_id)
            VALUES($user_id, '$first_name', '$last_name', '$email', '$password', '$department_id')";
      $mysql->query($sql);
   }

   header("Location: ../users.php");
?>
```

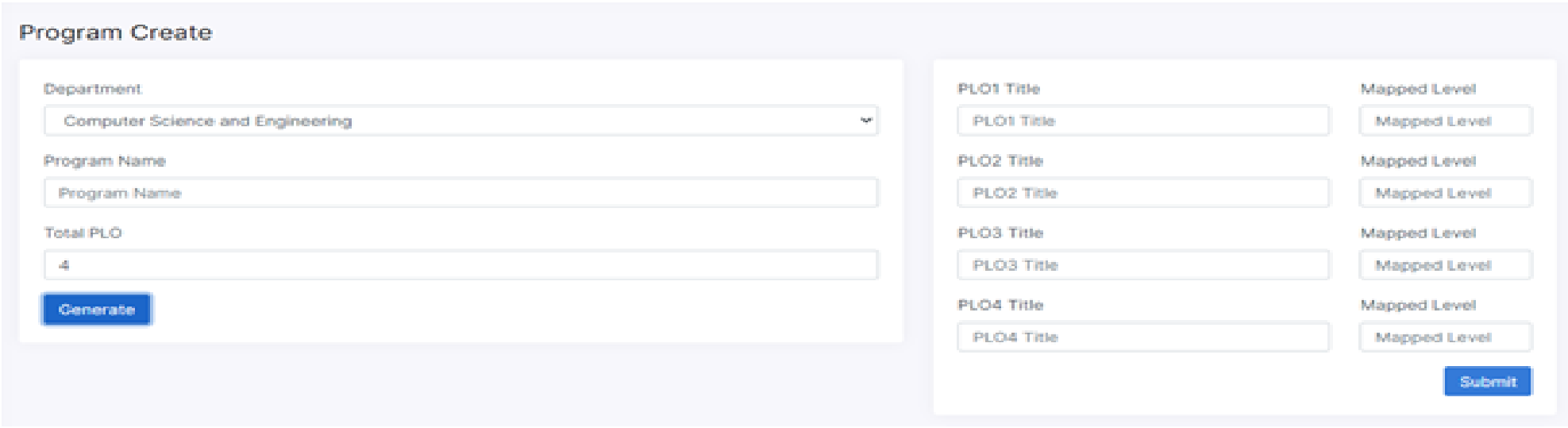



```
<?php
   require '../../database/mysql.php';

   $university_id = strtoupper($_POST['university_id']);
   $school_name = $_POST['school_name'];
   $dean = $_POST['dean'];

   $sql = "INSERT INTO school (school_name, dean, university_id)
         VALUES('$school_name', '$dean', '$university_id')";

   if($mysql->query($sql)){
```

```
        header("Location: ../schools.php");
    }else{
        header("Location: ../school-create.php");
    }
?>
```

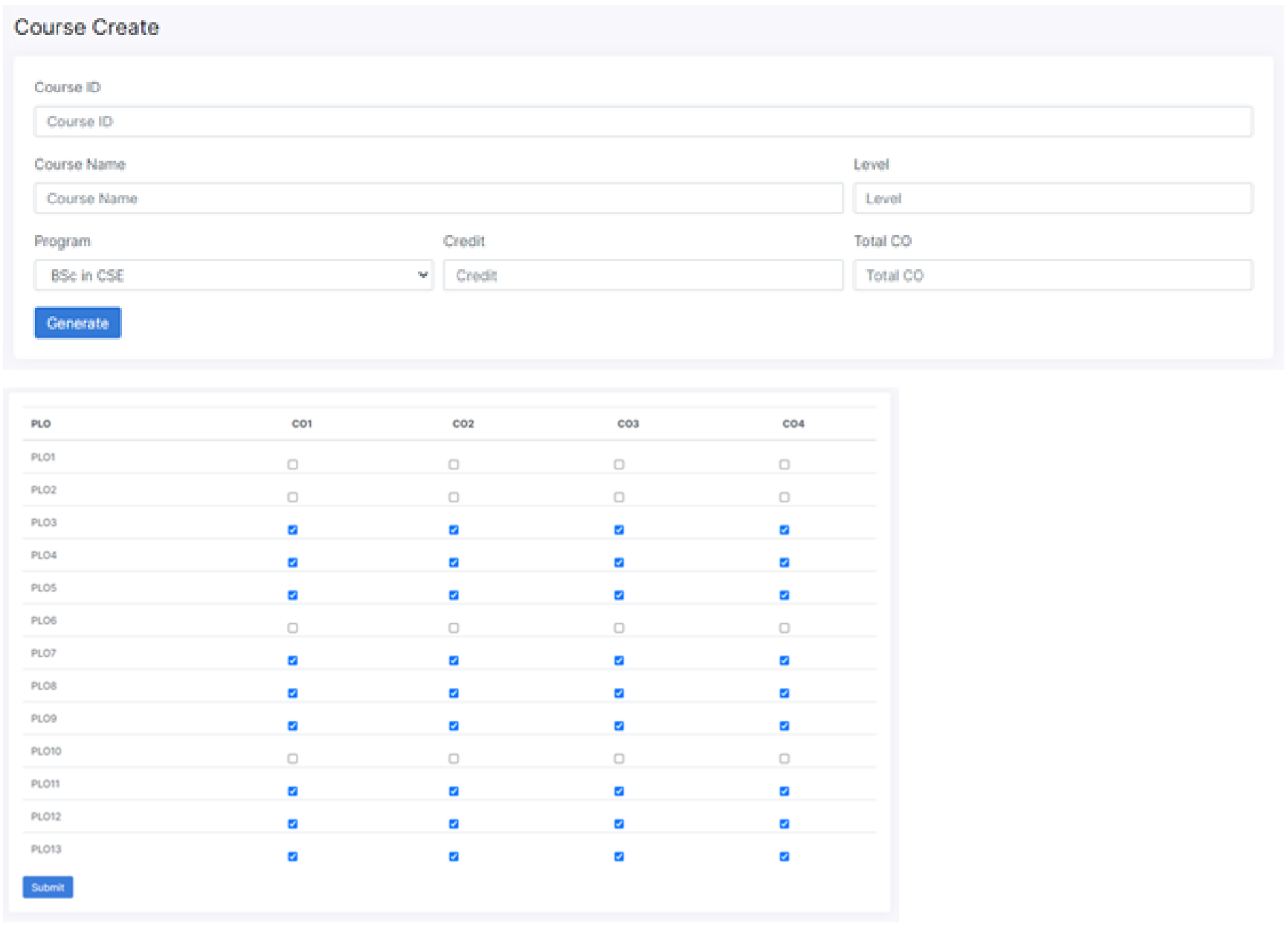



```
<?php
    require '../../database/mysql.php';
    $school_id = strtoupper($_POST['school_id']);
    $department_id = $_POST['department_id'];
    $department_name = $_POST['department_name'];
    $head = $_POST['head'];

    $sql = "INSERT INTO department (department_id, department_name, head, school_id)
            VALUES('$department_id', '$department_name', '$head', '$school_id')";

    if($mysql->query($sql)){
        header("Location: ../departments.php");
    }else{
        header("Location: ../department-create.php");
    }
?>
```



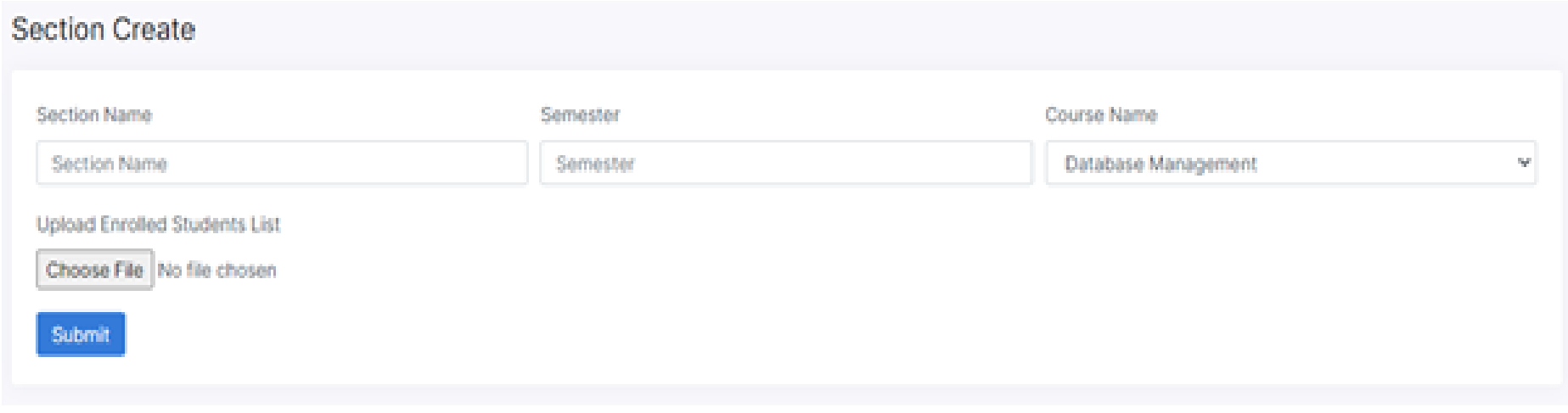

```
<?php
    require '../../database/mysql.php';
    $department_id = strtoupper($_POST['department_id']);
    $program_name = $_POST['program_name'];
```

```
$sql = "INSERT INTO program (program_name, department_id)
        VALUES('$program_name', '$department_id')";

$mysql->query($sql);

$program_id = $mysql->insert_id;
$plo_count = $_POST['plo_count'];

for($plo=1; $plo <= $plo_count; $plo++){
    $plo_name = $_POST["plo".$plo];
    $level = $_POST["map".$plo];
    $sql = "INSERT INTO plo (plo_number, plo_name, plo_level, program_id)
        VALUES ($plo, '$plo_name',  '$level', $program_id)";
    $mysql->query($sql);
}

    header("Location: ../programs.php");

?>
```



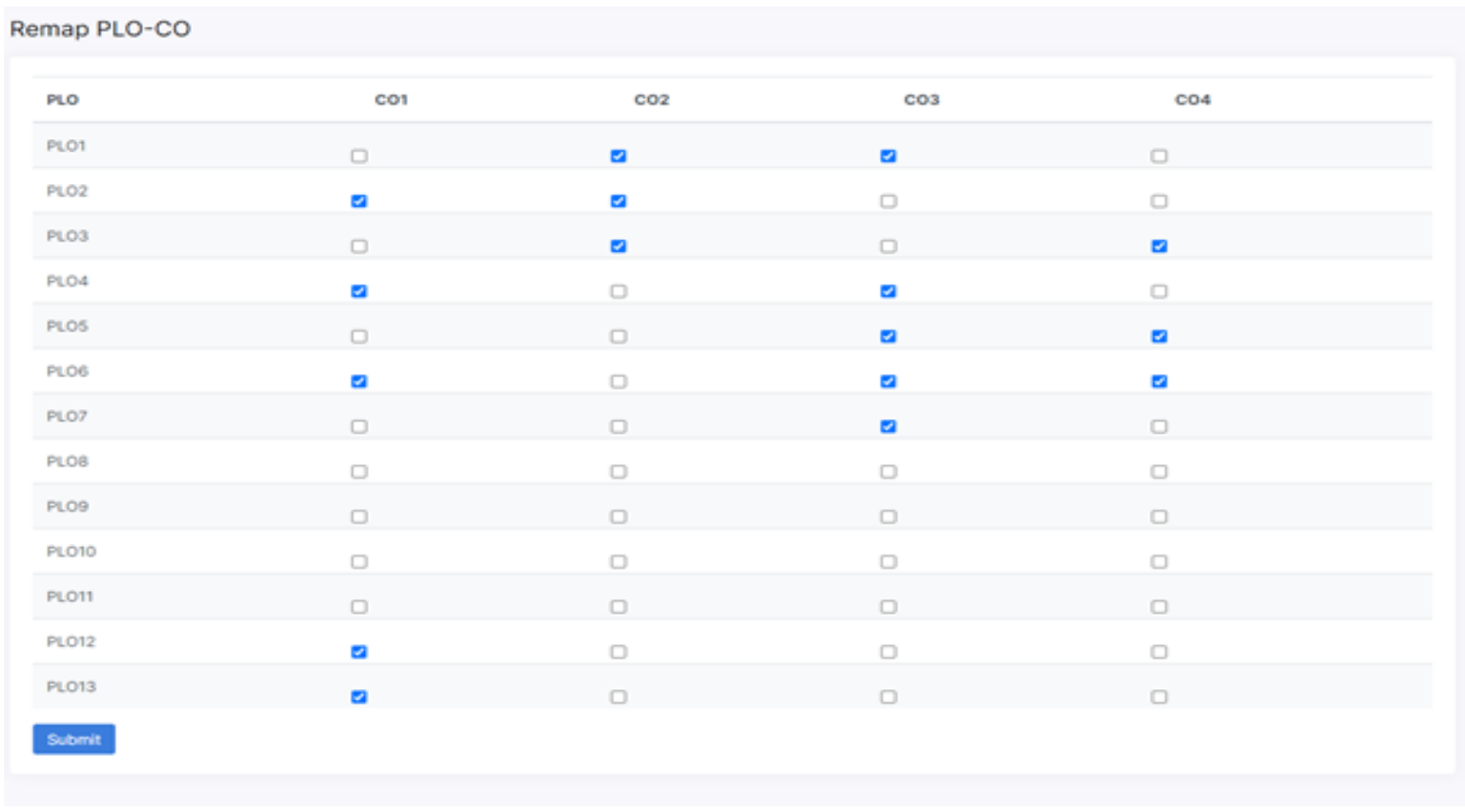

```
<?php
    require '../../database/mysql.php';
    $course_id = strtoupper($_POST['course_id']);
    $course_name = $_POST['course_name'];
```

```
$course_level = $_POST['course_level'];
$credits = $_POST['credits'];
$program_id = $_POST['program_id'];

$sql = "INSERT INTO course (course_id, course_name, course_level, credits, program_id)
      VALUES ('$course_id', '$course_name', $course_level, $credits, $program_id)";
$mysql->query($sql);
$total_plo = $_POST['total_plo'];
for($i=1; $i<$total_plo; $i++){
   if(isset($_POST["plo".$i])){
      $sql = "SELECT plo_id FROM plo WHERE plo_number = $i AND program_id = $program_id";
      $plo_id = $mysql->query($sql)->fetch_row()[0];
      for($j=0; $j<sizeof(($_POST["plo".$i])); $j++){
         $co = $_POST["plo".$i][$j];
         $sql = "INSERT INTO co (co_number, plo_id, course_id) VALUES($co, $plo_id, '$course_id')";
         $mysql->query($sql);
        }

      }
   }
   header("Location: ../courses.php");

?>
```



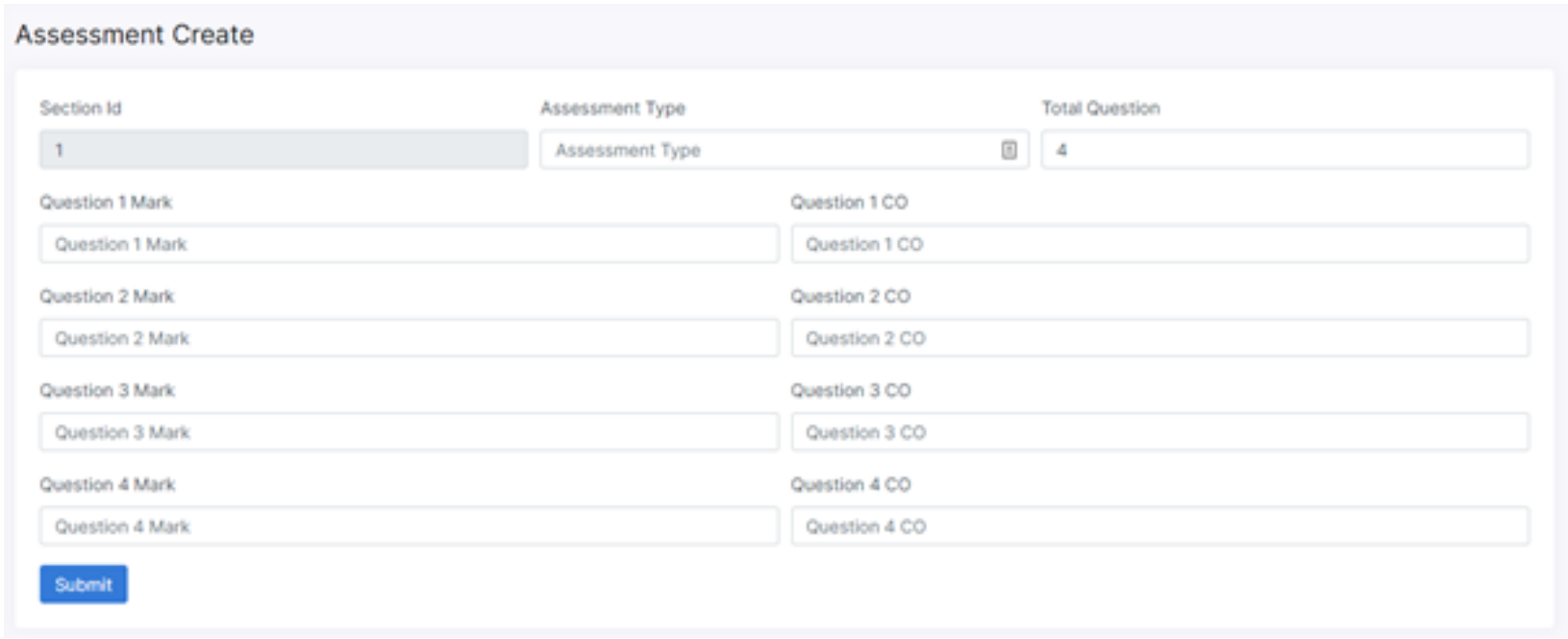

```
<?php
   require '../../database/mysql.php';
   session_start();
   $section_name = strtoupper($_POST['section_name']);
   $semester = $_POST['semester'];
   $course_id = $_POST['course_id'];
   $faculty_id = $_SESSION['user_id'];

   $sql = "SELECT section_id FROM section WHERE semester = '$semester' AND section_name = '$section_name' AND course_id = '$course_id' AND
faculty_id = '$faculty_id'";
   $result = $mysql->query($sql)->fetch_row();

   if($result){
```

```
    $section_id = $result[0];
}else{
    $sql = "INSERT INTO section (section_name, semester, course_id, faculty_id)
        VALUES('$section_name', '$semester', '$course_id', '$faculty_id')";
    $mysql->query($sql);
    $section_id = $mysql->insert_id;
}

$file = fopen($_FILES['students']['tmp_name'], "r");

fgetcsv($file);

while($d = fgetcsv($file)){
    $year = substr($semester, -4);
    $student_id = $d[0];
        $sql = "INSERT INTO enrollment (year, semester, student_id, section_id)
            VALUES ($year, '$semester', $student_id, $section_id)";
        $mysql->query($sql);
        echo $mysql->error;
    }

    header("Location: ../section-co-create.php?section_id=$section_id&course_id=$course_id");

?>
```



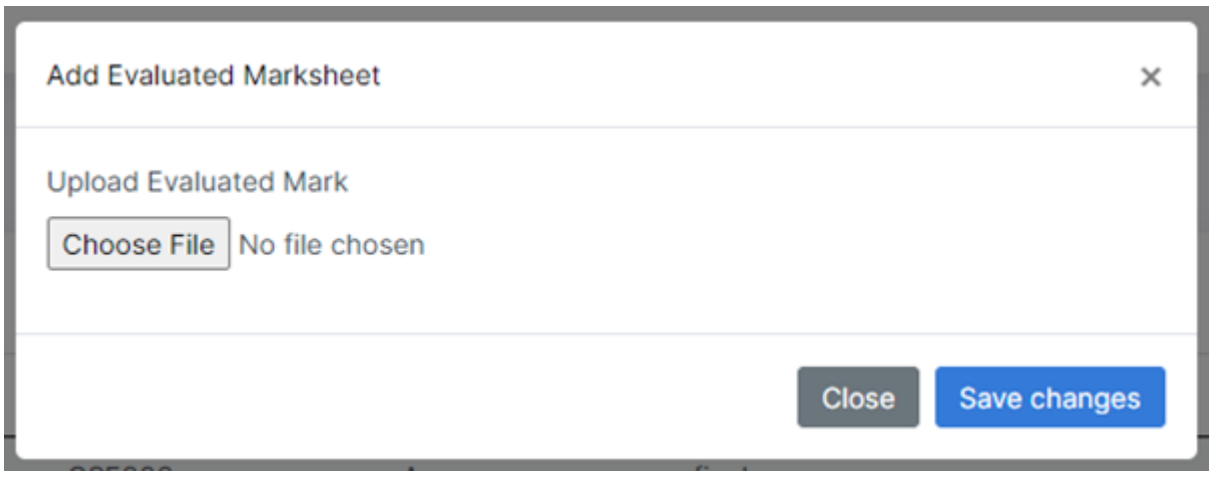

```
<?php
    require '../../database/mysql.php';

    $course_id = $_POST['course_id'];
    $section_id = $_POST['section_id'];
    $total_plo = $_POST['total_plo'];
```

```
$sql = "SELECT program_id FROM course WHERE course_id = '$course_id'";
$program_id = $mysql->query($sql)->fetch_row()[0];

for($plo=1; $plo<=$total_plo; $plo++){
   if(isset($_POST['plo'.$plo])){
      for($j=0; $j<sizeof($_POST['plo'.$plo]); $j++){
         $co = $_POST['plo'.$plo][$j];
         $sql = "SELECT plo_id FROM plo WHERE program_id = $program_id AND plo_number = $plo";
         $plo_id = $mysql->query($sql)->fetch_row()[0];
         $sql = "INSERT INTO co (co_number, plo_id, course_id, section_id)
               VALUES ($co, $plo_id, '$course_id', $section_id)";
         $mysql->query($sql);
      }
   }
}

   header("Location: ../sections.php");

?>
```




```
<?php
   require '../../database/mysql.php';

   $section_id = $_POST['section_id'];
   $assessment_type = strtolower($_POST['assessment_type']);
   $total_q = $_POST['total_q'];

   for($i=1; $i<=$total_q; $i++){
      $mark = $_POST['mark'.$i];

      $sql = "SELECT course_id FROM section WHERE section_id = $section_id";
      $course_id = $mysql->query($sql)->fetch_row()[0];

      $co = $_POST['co'.$i];
```

```
    $sql = "INSERT INTO assessment (assessment_type, question_number, co, mark, section_id)
            VALUES('$assessment_type', $i, $co, $mark, $section_id)";
    $mysql->query($sql);
  }

  header("Location: ../assessments.php?section_id=$section_id");

?>
```




```
<?php
  require '../../database/mysql.php';

  $section_id = $_POST['section_id'];
  $assessment_type = strtolower($_POST['type']);

  $file = fopen($_FILES['evaluation']['tmp_name'], "r");
  fgetcsv($file);

  while($d = fgetcsv($file)){
    $student_id = $d[0];
    $sql = "SELECT enrollment_id FROM enrollment WHERE student_id = $student_id AND section_id = $section_id";
    $enrollment_id = $mysql->query($sql)->fetch_assoc()['enrollment_id'];
    for($i=1; $i<sizeof($d); $i++){
      $mark_obtains = $d[$i];
      $sql = "SELECT assessment_id FROM assessment WHERE section_id = $section_id AND assessment_type = '$assessment_type' AND
question_number = $i";
      $assessment_id = $mysql->query($sql)->fetch_assoc()['assessment_id'];
      $sql = "SELECT * FROM marksheet WHERE assessment_id = $assessment_id AND enrollment_id = $enrollment_id";
      if($mysql->query($sql)->num_rows == 0){
        $sql = "INSERT INTO marksheet (assessment_id, enrollment_id, mark_obtains)
          VALUES ($assessment_id, $enrollment_id, $mark_obtains)";
        $mysql->query($sql);
      }else{
        $sql = "UPDATE marksheet SET mark_obtains = $mark_obtains WHERE assessment_id = $assessment_id AND enrollment_id = $enrollment_id";
        $mysql->query($sql);
      }

    }
```

```
    }
    header("Location: ../assessments.php?section_id=$section_id");
?>
```

System Output

SELECT school.school_name, COUNT(DISTINCT(enrollment.student_id)) as 'students' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN enrollment

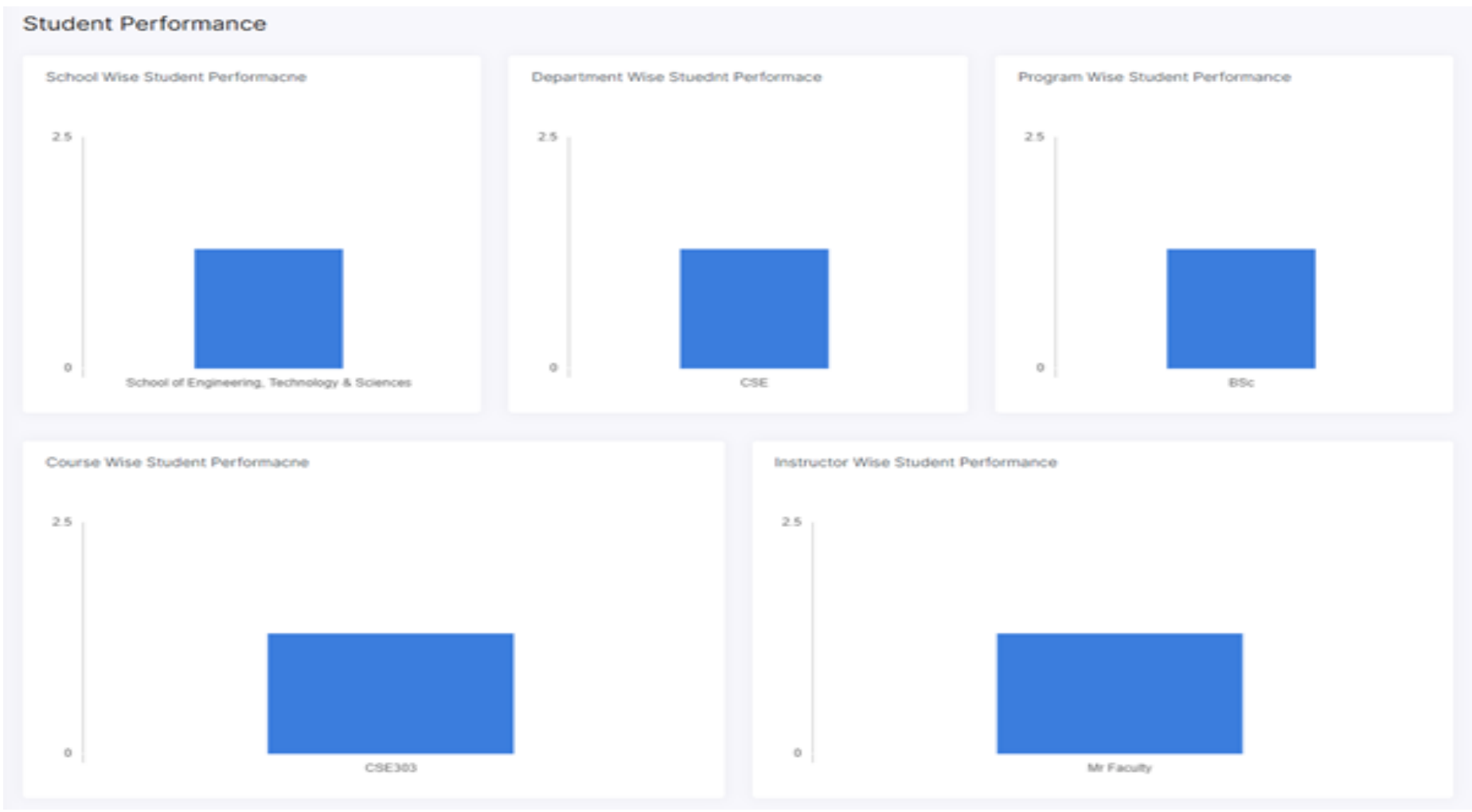
WHERE section.semester = '$semester' GROUP BY school.school_name

SELECT department.department_id, COUNT(DISTINCT(enrollment.student_id)) as 'students' FROM department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN enrollment

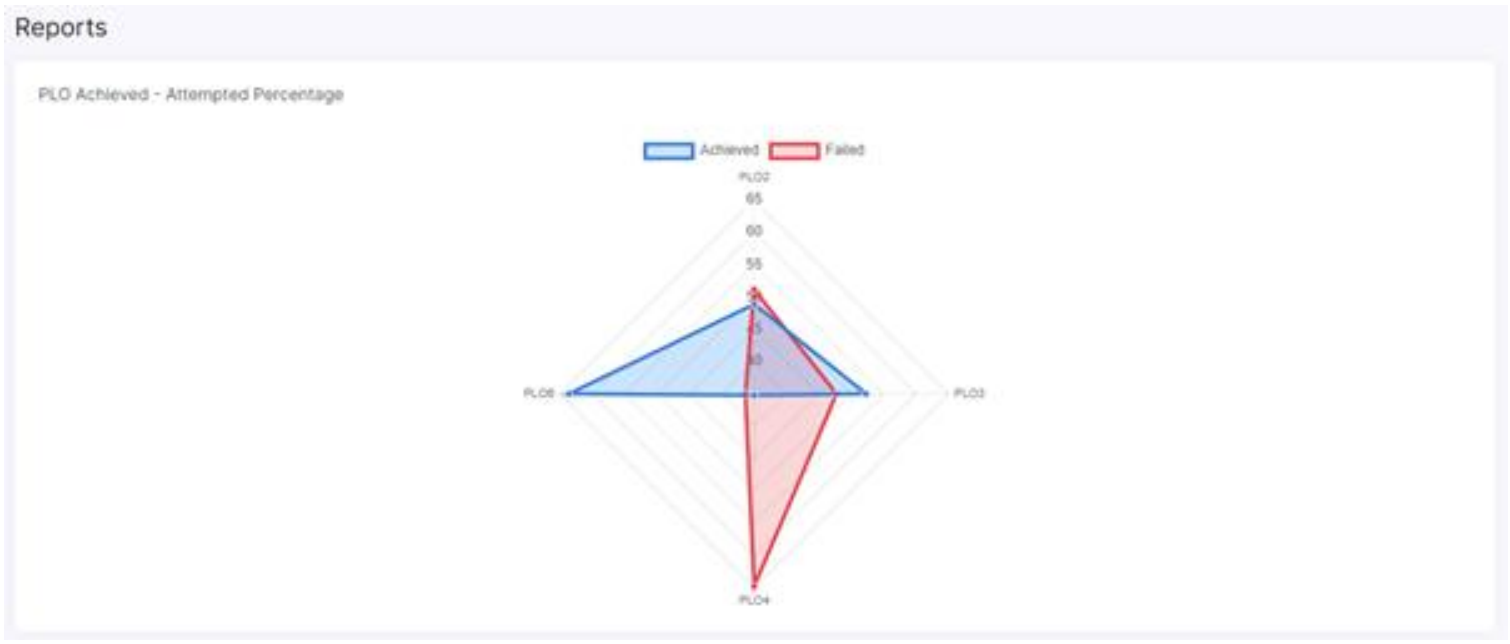
WHERE section.semester = '$semester' GROUP BY department.department_id

SELECT program.program_name, program.department_id, COUNT(DISTINCT(enrollment.student_id)) as 'students' FROM program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN enrollment

WHERE section.semester = '$semester' GROUP BY program.program_id



SELECT school, department, program, semester, faculty, course, student, credits, assessment, SUM(obmrk) as 'obmark' FROM (SELECT school.school_name as 'school', UPPER(department.department_id) as 'department', program.program_name as 'program', section.semester as 'semester', CONCAT(faculty.first_name, ' ', faculty.last_name) as 'faculty', course.course_id as 'course', enrollment.student_id as 'student', course.credits, assessment.assessment_type as 'assessment', IF(assessment.assessment_type = 'final', (SUM(marksheet.mark_obtains) / SUM(assessment.mark)) * 40, (SUM(marksheet.mark_obtains) / SUM(assessment.mark)) * 30) as 'obmrk' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN faculty NATURAL LEFT JOIN assessment NATURAL LEFT JOIN marksheet NATURAL LEFT JOIN enrollment LEFT JOIN co ON assessment.co = co.co_number AND section.section_id = co.section_id LEFT JOIN plo on co.plo_id = plo.plo_id WHERE section.semester = LOWER('$semester') GROUP BY course.course_id, enrollment.enrollment_id, assessment.assessment_type, course.course_id) as mySql GROUP BY semester, student, course



SELECT section.course_id as 'course', section.semester, assessment.co, plo.plo_number as 'plo',

IF(SUM(marksheet.mark_obtains) / SUM(assessment.mark) >=0.40, 1, 0) as 'pof'

FROM section NATURAL LEFT JOIN assessment

NATURAL LEFT JOIN marksheet

NATURAL LEFT JOIN enrollment

LEFT JOIN co ON assessment.co = co.co_number AND section.section_id = co.section_id

LEFT JOIN plo ON co.plo_id = plo.plo_id

GROUP BY enrollment.student_id, section.section_id, plo.plo_id

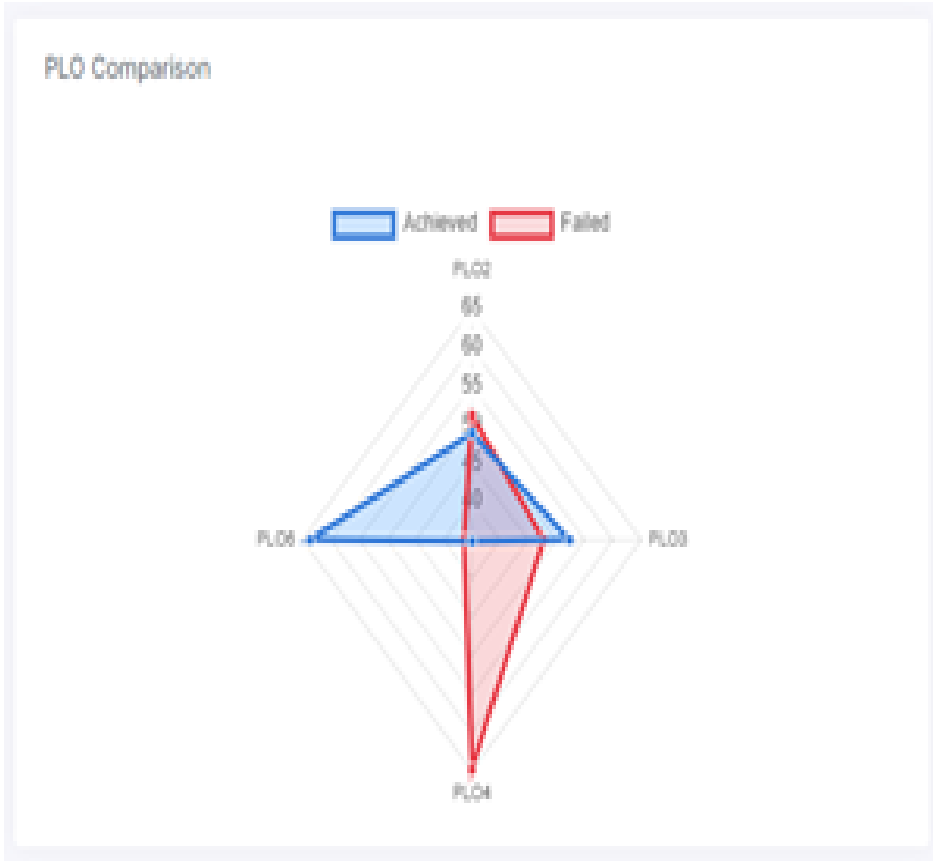
ORDER BY pof DESC, plo ASC

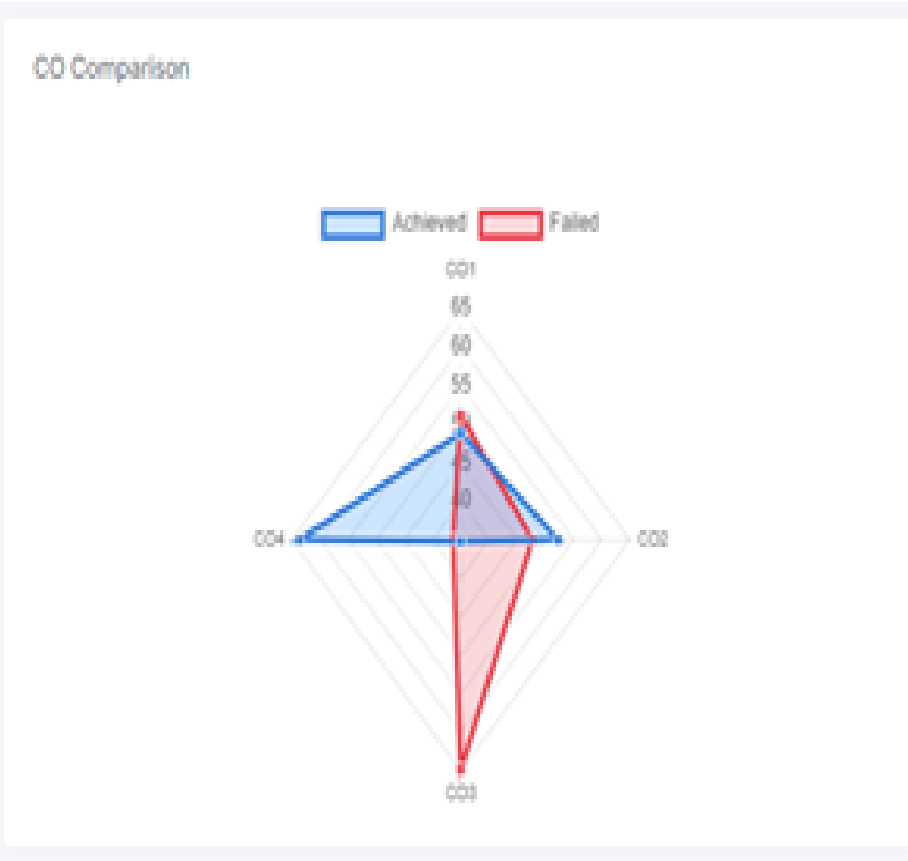
PLO CO Verdict

#	Achieved	Achieved (%)	Failed	Failed (%)	Total
PLO2	43	48.86	45	51.14	88
PLO3	46	52.27	42	47.73	88
PLO4	31	35.23	57	64.77	88
PLO6	56	63.64	32	36.36	88
CO1	43	48.86	45	51.14	88
CO2	46	52.27	42	47.73	88
CO3	31	35.23	57	64.77	88
CO4	56	63.64	32	36.36	88

SELECT school, department, program, plo, COUNT(res) as 'res' FROM (SELECT school.school_name as 'school', program.department_id as 'department', program.program_name as 'program', plo.plo_number as 'plo', IF(SUM(marksheet.mark_obtains)/SUM(assessment.mark)>=0.40, 1, 0) AS 'res' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN assessment NATURAL LEFT JOIN marksheet NATURAL LEFT JOIN enrollment LEFT JOIN co ON assessment.co = co.co_number AND assessment.section_id = co.section_id LEFT JOIN PLO ON co.plo_id = plo.plo_id WHERE school.school_name LIKE '%engineer%' GROUP BY enrollment.student_id, section.course_id, plo.plo_number ORDER BY res DESC, plo ASC) as testQ WHERE res=1 GROUP BY plo

SELECT school, department, program, plo, COUNT(plo) as 'res' FROM (SELECT school.school_name as 'school', program.department_id as 'department', program.program_name as 'program', plo.plo_number as 'plo', IF(SUM(marksheet.mark_obtains)/SUM(assessment.mark)>=0.40, 1, 0) AS 'res' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN assessment NATURAL LEFT JOIN marksheet NATURAL LEFT JOIN enrollment LEFT JOIN co ON assessment.co = co.co_number AND assessment.section_id = co.section_id LEFT JOIN PLO ON co.plo_id = plo.plo_id WHERE school.school_name LIKE '%engineer%' GROUP BY enrollment.student_id, section.course_id, plo.plo_number ORDER BY res DESC, plo ASC) as testQ WHERE plo <> 0 GROUP BY plo

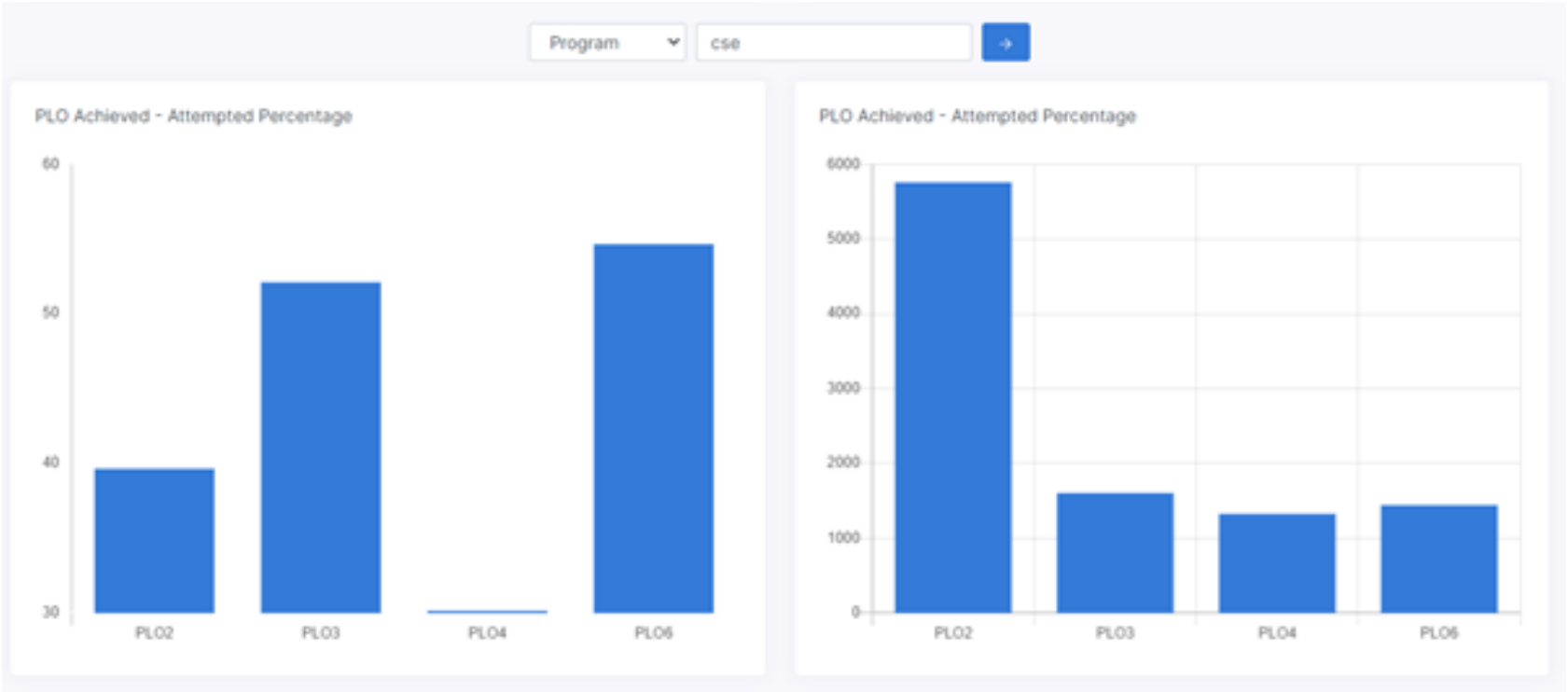




SELECT school, department, program, co, COUNT(res) as 'res' FROM (SELECT school.school_name as 'school', program.department_id as 'department', program.program_name as 'program', co.co_number as 'co', IF(SUM(marksheet.mark_obtains)/SUM(assessment.mark)>=0.40, 1, 0) AS 'res' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN assessment NATURAL LEFT JOIN marksheet NATURAL LEFT JOIN enrollment LEFT JOIN co ON assessment.co = co.co_number AND assessment.section_id = co.section_id

LEFT JOIN PLO ON co.plo_id = plo.plo_id WHERE school.school_name LIKE '%engineer%' GROUP BY enrollment.student_id, section.course_id, co.co_number ORDER BY res DESC, co ASC) as testQ WHERE res=1 GROUP BY co

SELECT school, department, program, co, COUNT(co) as 'res' FROM (SELECT school.school_name as 'school', program.department_id as 'department', program.program_name as 'program', co.co_number as 'co', IF(SUM(marksheet.mark_obtains)/SUM(assessment.mark)>=0.40, 1, 0) AS 'res' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN assessment NATURAL LEFT JOIN marksheet NATURAL LEFT JOIN enrollment LEFT JOIN co ON assessment.co = co.co_number AND assessment.section_id = co.section_id LEFT JOIN PLO ON co.plo_id = plo.plo_id WHERE school.school_name LIKE '%engineer%' GROUP BY enrollment.student_id, section.course_id, co.co_number ORDER BY res DESC, co ASC) as testQ WHERE co <> 0 GROUP BY co



SELECT school.school_name as'school', UPPER(program.department_id) as 'department', CONCAT(program.program_name, '(', UPPER(program.department_id), ')') as 'program', course.course_id as 'course', plo.plo_number as 'plo', co.co_number as 'co', SUM(assessment.mark) as 'total', SUM(marksheet.mark_obtains) as 'mark', COUNT(DISTINCT(enrollment.student_id)) as 'student' FROM school NATURAL LEFT JOIN department NATURAL LEFT JOIN program NATURAL LEFT JOIN course NATURAL LEFT JOIN section NATURAL LEFT JOIN assessment NATURAL LEFT JOIN marksheet LEFT JOIN co ON assessment.section_id = co.section_id AND assessment.co = co.co_number LEFT JOIN plo ON co.plo_id = plo.plo_id LEFT JOIN enrollment ON marksheet.enrollment_id = enrollment.enrollment_id WHERE department.department_id LIKE '%cse%' GROUP BY department.department_id, plo.plo_number, co.co_number ORDER by plo.plo_number, co.co_number

Conclusion

- ## Problem & Solution

There were many problems that we have come across while creating the SPM System.

1. The data given to us were not detailed enough for what action shall it be taken for.
2. The data we had did not have enough information of what output will it give us.
3. There were errors that came across our coding process.
4. We weren't experience on the languages (PHP, CSS, JAVASCRIPT, HTML, SQL) that we must use to create the application.

So, even though we did face some issues while creating the system, but we learned to fix it too. We took long time to learn the languages and we divided in our group individually and discussed all together on which part we do and then worked together in the coding part. We manually check each line for the coding to make sure to find the error and fix it.

- ## feature & Future Development

Further development that we will be working on is getting to receive more large number of data to work on.in future we will try to add more feature.

- The addition of A question Bank page where faculty can upload there question and collect the previous question paper from there.
- An addition of a automated result page where student can see there result after a semester end.

- ## Conclusion & Recommendation

We have created a SPM a system through which a user can take inputs and outputs. We have tried to design, built and tried to implement the best quality for our SPM system. We believe this system will help the Department member and faculty to save much more time to work on and improve the quality of education. This system allows the user to Collect the resources they need and can store all the available information and analysis. we believe this system will help the faculty member to keep track their student performance and also will help to improve their performance