Project Report

Database Management System

STUDENT PERFORMANCE MONITOR

Submitted BY

Name	ID
Partho Protim Saha	1730016
Ahnaf Tazwar Araf	1731462
Nusrat Zahan	1730739
Md. Riyad Hossain	1731407
Afra Hossain	1811011

Table of Contents

CHAPTER 1

INTRODUCTION

›› BACKGROUND OF THE PROJECT

›› OBJECTIVE OF THE PROJECT

›› SCOPE OF THE PROJECT

SECTION 1: INTRODUCTION

BACKGROUND OF THE PROJECT

The purpose of our project is to design, and implement a software that will help universities everywhere to promote a more productive and effective way of evaluating students and we did this through the idea of Course Outcomes (COs) and Program Learning Outcomes (PLOs), where each CO is mapped to a PLO and each PLO represents a specific skill that the students are expected to gain at the end of that course, such as problem analysis, design, implementation, etc. 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 faculty members get the PLOs from BEATE website and will then input the COs for each of their students so that the system can map the COs to PLO accordingly. Through the implementation of this project, it was found that the efficiency did not only save time but also improve quality of education. It will help the faculty members to identify the specific section in which a student needs improvement also it will help the student to gain the most skills out of a course - students can keep track of their progress in each sector and pin-point the areas that need self-improvement and self-growth. In addition, our software hopes to benefit the administrative bodies and departmental bodies as well –to track progress of students, departmental performance and help them distribute and allocate resources better.

OBJECTIVE OF THE PROJECT

The main objective of our project is to ensure a user-friendly software that will help both the institutional bodies and students to increase the quality of education. We hope that our software will bring massive advancements in our education and will also contribute significantly not only to the field of Computer Science but also all sectors of education.

SCOPE OF THE PROJECT

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

1. Facilitate the implementation, including planning and management

2. Conduct monitoring of the project

3. Support for review and improvement of the project implementation

4.Project initiation

5. Data Collection

6. Potential Modeling

7.Program Analysis

8. Reporting

7. Project management

CHAPTER 2

REQUIREMENT ANALYSIS

›› **RICH PICTURE AS-IS**

›› **SIX ELEMENT ANALYSIS AS-IS**

›› **PROCESS DIAGRAM AS-IS**

›› **PROBLEM ANALYSIS**

›› **RICH PICTURE TO-BE**

›› **SIX ELEMENT TO-BE**

›› **PROCESS DIAGRAM TO-BE**

SECTION 2: REQUIREMENT ANALYSIS

RICH PICTURE AS-IS

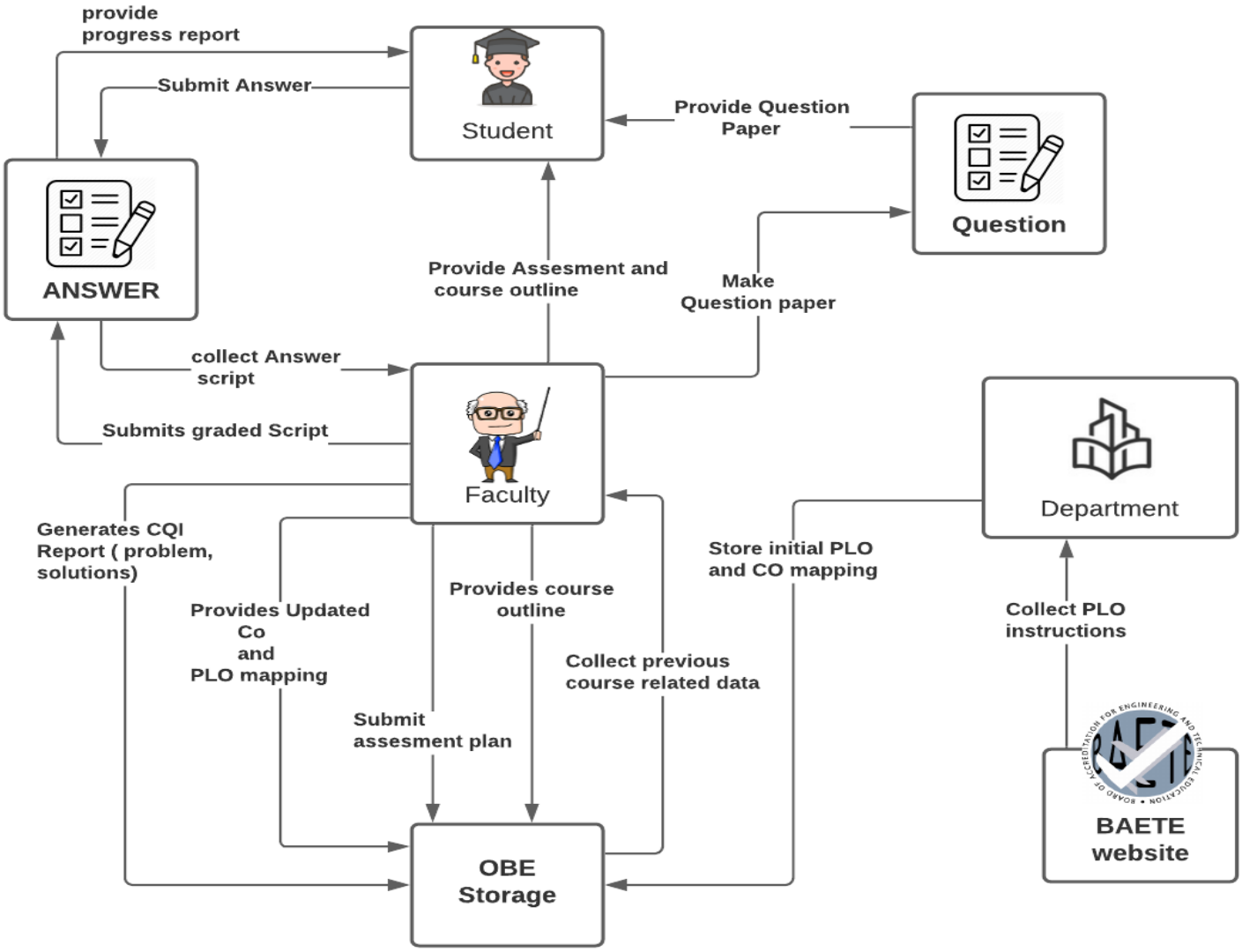


FIGURE 2.1- Rich Picture As-Is

SIX ELEMENT ANALYSIS AS-IS

Process Name	System Roles					
	Human	**Non-Computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Network & Comminati on**
1. Collect the PLO and store the initial mapping	**Department Head**: 1. Go to BAETE's website 2. Collect the BAETE accreditation manual. 3. Read the PLO instructions from the manual. 4. They have to follow the instruction which is provided by the accreditation manual for graduation from educational Criteria. 5. Before starting mapping, he/she must set the number of PLO for each course. Example: we have 12 PLO (According to the latest manual, 05.03.2019). and we have to map those PLO with 18 courses. So that each	**Pen and Paper:** CO and PLO mapping with courses are mapped in the paper using pen.	**Computer** 1) Computer is used to go to the website of BAETE to collect the accreditation manual 2) Computers are also used for making softcopies of PLO and CO mapping. **Printer:** To print the accreditation manual paper and softcopy of CO and PLO paper.	**Browsing:** To browse like Samsung browser, chrome, Mozilla Firefox Safari etc. **Drive:** Used to store all information to OBE storage Head/Faculty. **Microsoft Word:** Preparing the mapping of PLO and CO papers, in that case, they can use word files. **PDF viewer:** To view the BAETE accreditation manual which is collected from BAETE website.	**Microsoft Excel Files:** access the data to see or edit the Curriculum of PLO & CO's initial mapping.	**Internet:** Online platforms such as- google may be used for entering the BAETE website. Collect accreditation manual from BAETE website about PLO. and also collect information related to courses which is found in IUB website.

	course will get at least 5 or 6 PLO for mapping and each PLO will get at least 7 courses for mapping roughly so we have to choose the most important and specific ones. 6. For doing so faculty member will-- a. Collect the previous outline, find out the main objectives of a single course b. Mapped those objectives with the level of learning Taxonomy. c. Then set the program learning outcome based on Course outcome. d. Also set the assessment initial planning with course outcome. 7. Accreditation Manual paper and initial mapping paper will be stored in the OBE storage. 8. Faculty collect all information from OBE storage which is stored by the department.			**Operating System:** Any OS may be used. e.g. Windows, MacOS.		
2. Update mapping (if needed) generating assessment	**Faculty:** 1. Collect all information from OBE storage	**Pen and Paper:** Instructions of Course outline	**Computer:** Computers are used to prepare	**Microsoft Word:** Typing the course outline	**Docx/pdf Files:** To edit all kinds of information	**Internet:** Online platforms such as- google docx

plan and course outline	which is stored by the department. 2. Now If the faculty wants to change something in PLO vs CO initial mapping, then s/he can change that mapping and store it in OBE storage. 3. If faculty wants to check the previous course outline, then they have to collect it from OBE storage. 4. Then if needed they have to prepare the new course outline with course outcomes. 5. Now identify for each course main objectives. 6. Then create mapping Co's with PLO like: a) Here, CO1 is considered as a "first level (remembering)" (That's provided "Level of learning bloom Taxonomy") of this course. But in bloom taxonomy: (remembering, understanding, Applying, Analyzing,	and course assessment planning as CO and PLO basis details printed on paper.	course outline and assessment planning using current CO and PLO mapping, also for making softcopies of course outline and Assessment planning. **Printer:** To print the softcopy of course outline and assessment planning.	and assessment planning and generating a printable pdf. **Operating System:** Any OS can be used. e.g. Windows, MacOS. **Adobe Acrobat Reader:** For viewing the assessment planning paper and course outline in pdf format.	like: course outline, assessment planning papers are stored in the docx/pdf file. **Department Storage:** A hardcopy of OBE course outline docs/pdf file is stored in the department storage.	may be used to prepare docx files for course outline and assessment paper.

	Evaluation, and lastly creating) b) Now in PLO, which is more appropriate for the first level of CO. c) After this, then write down the reason. 7.Then they have to find the course outline of a single course, mapped with the course outline to PLO and store it to OBE storage. 8. Now, Faculty members will create the course assessment based on CO and PLO updated mapping like: a) In this course the first quiz will be very fundamental theory. There are 4 CO's (CO1, CO2, CO3, CO4). Here CO1 Mapped with PLO (Engineering knowledge) which is a very basic thing. And also used in (Quiz-2, MID, Quiz-4, Final) . b) In the 2nd Exam: this is a little more difficult. Read out the problems, formulated the					

	problems using the first principle of mathematics. so this CO2 is actually represented by the "Quiz-2" and "MID term" exam. In this way faculty can create assessment planning and also update course outline. 9. Store course assessments and course outline information to OBE storage.					
3. Course progress	**Faculty:** 1. Faculty member designing the question paper based on the current CO and PLO mapping. 2. Creates and connects individual questions with the COs based on the course's assessment table. 3.The question paper is sent to the other faculties who have the same course to check the following question paper. 4. Those faculties will verify the question paper and	**Stationery:** 1) Paper is used to Print the question papers. 2) Supply pen, pencil, eraser, pencil sharpener, stapler, ruler and other necessities that are required during the examination. **Chairs and Table** For using during the exam. **Room** Designated room for examination.	**Computer/ Laptop:** 1. For preparing the question paper, a computer is needed. 2. Some courses require a computer for coding or an open book exam. 3.To prepare the question paper they use the docs/pdf file. 4. For printing question papers.	**Microsoft Word:** Being used by the faculty for typing and preparing the questions and generating a docs. **Operating System** Any OS may be used. e.g. Windows, MacOS. **Adobe Acrobat Reader** For viewing the question paper in pdf format **Google Classroom**	**Docx/pdf Files:** To edit all kinds of information like: CO and PLO mapped updated course outline , question papers are stored in the docx/pdf file format. **Department Storage:** A hardcopy of course outline, question papers for every course will be stored in the department storage (OBE)	**Internet:** 1.Used by students during open world exam 2.Online platforms such as- google docs may be used to prepare question papers for examination 3. From the examinee to confirm for exam date, time and room no.to send this information, in that time maybe they use the internet.

	check for any errors and correct them. 5.Prepare SODs and invigilators for the exam. 6. Return the question paper to the designated faculty. 7. Contacts with the exam committee to manage exam date, time and place by providing the necessary information about the exam i.e. total no. of students and total exam duration. 8. Receives confirmation and schedule about the exam that was requested. 9. Publish information about the exam to the students i.e. exam date, time and place and syllabus to the students. **Examination committee:** After getting notification from the faculty members about the exam, they need to		**Scientific Calculators:** Some exams require the use of scientific calculators like mathematics, circuit, discrete math etc. **Printers & photocopy machine:** Instructors use it for printing question papers.	Used by faculties and students during examinations.		

	fix a particular date, time and place for the examination and confirm the faculty member about the date of examination, time, room number and other things. Student: 1. Receive information about the examination from the faculty, that is, the syllabus for that upcoming exam, the date and time and where the exam will be held. 2. Attend the exam at the correct time and place and give the exam. 3. When done with the exam, submit the answer scripts to the SODs or faculty and leave the examination hall.					
4. Checking Scripts and generating progress report	**Faculty:** 1. Faculty members must retrieve all response scripts from the answer bank after taking the assessment. 2. Faculty members have to mark the answer sheets after checking properly.	**Stationary:** 1)Pen and paper for Check answer scripts for evaluating. And also, for creating grade sheets (manually).	**Computer/ Laptop** To prepare the excel file of the grade sheet **Calculators:** Some exams require the use of calculators like mathematics, circuit,	**Microsoft Excel:** Typing the id and marks generates a printable excel file. **Operating System:** Any OS may be used. e.g. Windows, MacOS.	**Microsoft Excel:** Used for storing exam marks and calculating final grade through ID wise.	**Internet:** Used by faculty members to create online excel files and also for sharing excel files.

	3. After checking all scripts, distribute them to students for rechecking and wait for their feedback. 4. After resolving problems with students about exam papers, faculty members will create sample solution papers for the given questions in exams and store those to the OBE storage. 5. Faculty members have to prepare grade sheets as per course outline and store the data in OBE storage. 6. These reports must be sent to the department head by faculty members. 7. If department head have any query about marks or grade sheets, faculty members will give feedback . **Student:** 1. Students will collect the scripts from faculty		discrete math etc. **Printers & photocopy machine:** Faculty members Can use it for printing grade sheets.			

	members and check their marks . 2. If any query then requests them to recheck the answer scripts and wait for the feedback **Department Head:** 1.Check the final grades with marks in excel files which are provided by the faculty members 2. If there is any query for the marks or grade sheet, department heads will contact that faculty who is taking that course.					
5.Generating CQI Report	**Faculty Member:** 1. Get all assessment reports of that semester from OBE 2. Check the percentage under a specific PO. 3. Identify the lowest percentage 4. Figure out the reason for the lowest percentages.	**Stationary: Pen and** Paper: Used if a faculty member wants to print the report or write something on the report. **Marker Pen:** Used if a faculty member needs to mark something on the report.	**Computer** 1) Computers are used to prepare the CQI report with a printable format. 2. Used to edit the Excel file. **Printer** To print the assessment report or CQI report.	**Operating System** Any OS Can be used. e.g. Windows, MacOS. **Adobe Acrobat Reader:** For viewing the report in pdf format, Adobe reader is needed.	**Docx/pdf Files:** To view the CQI report which is prepared in the docx or pdf file. **Department Storage** A hardcopy of CQI reports will be stored in the department storage.	**Internet** Online platforms such as- google docs can be used to prepare docx files for CQI reports.

	5. Solve those problems. For that faculties can do the following: a) change course outline and assessment planning b) update the CO PLO mapping. 6. Add suggestions about how the performances can be improved. 7. Compile all the data and generate a report 8. Store the report in the OBE storage. **OBE Storage:** Send an assessment report to the faculty member.			**Web Browser:** To send and receive the report through email.		

PROCESS DIAGRAM AS-IS

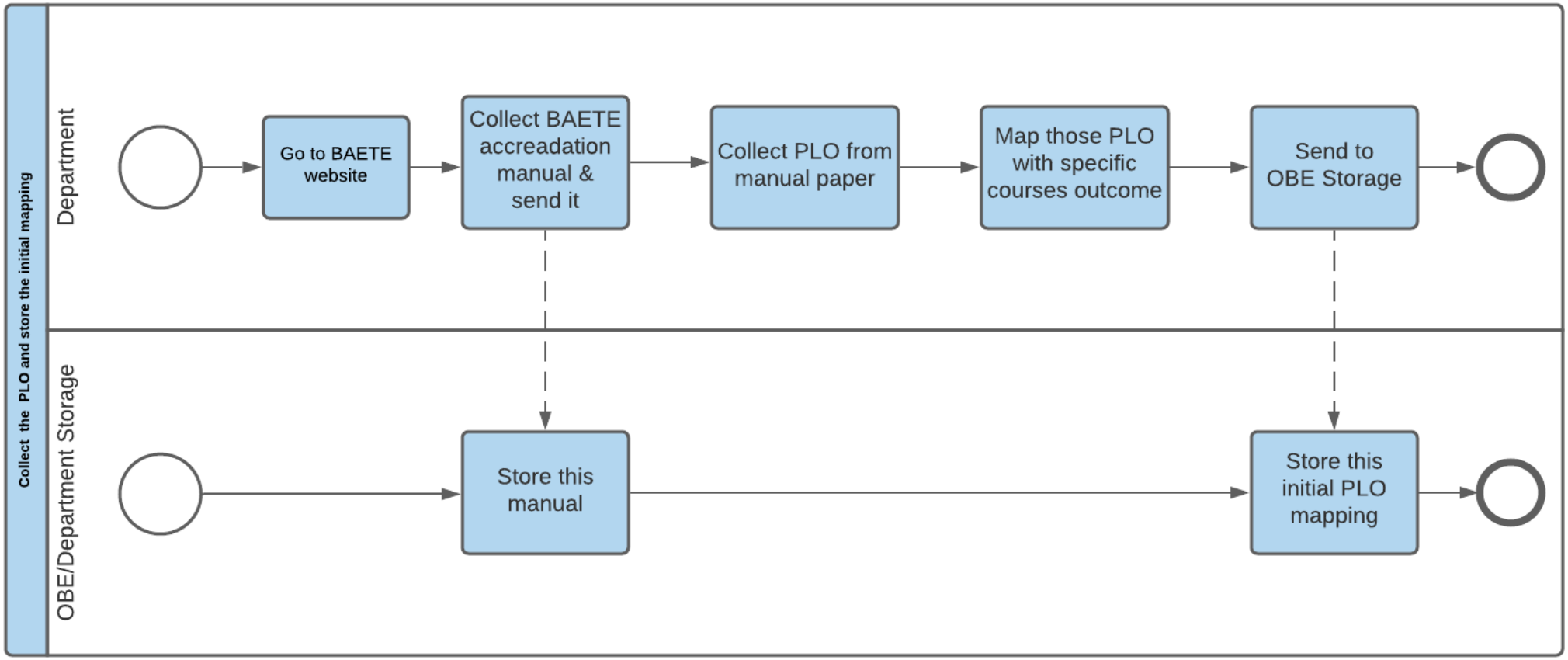


FIGURE 2.2- Process diagram of collect the PLO and store the initial mapping (as-is)

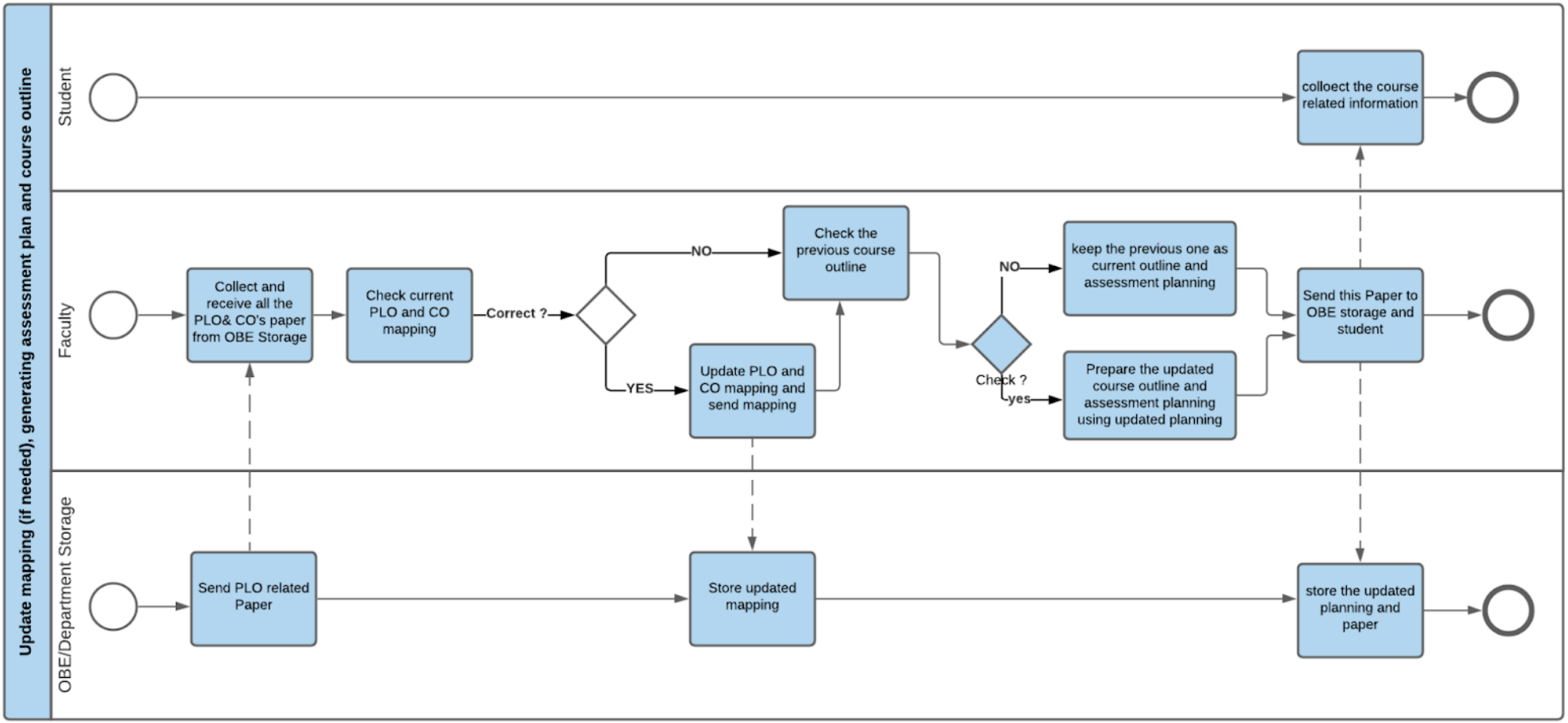


FIGURE 2.3- Process diagram of Update mapping (if needed) generating assessment plan and course outline (as-is)

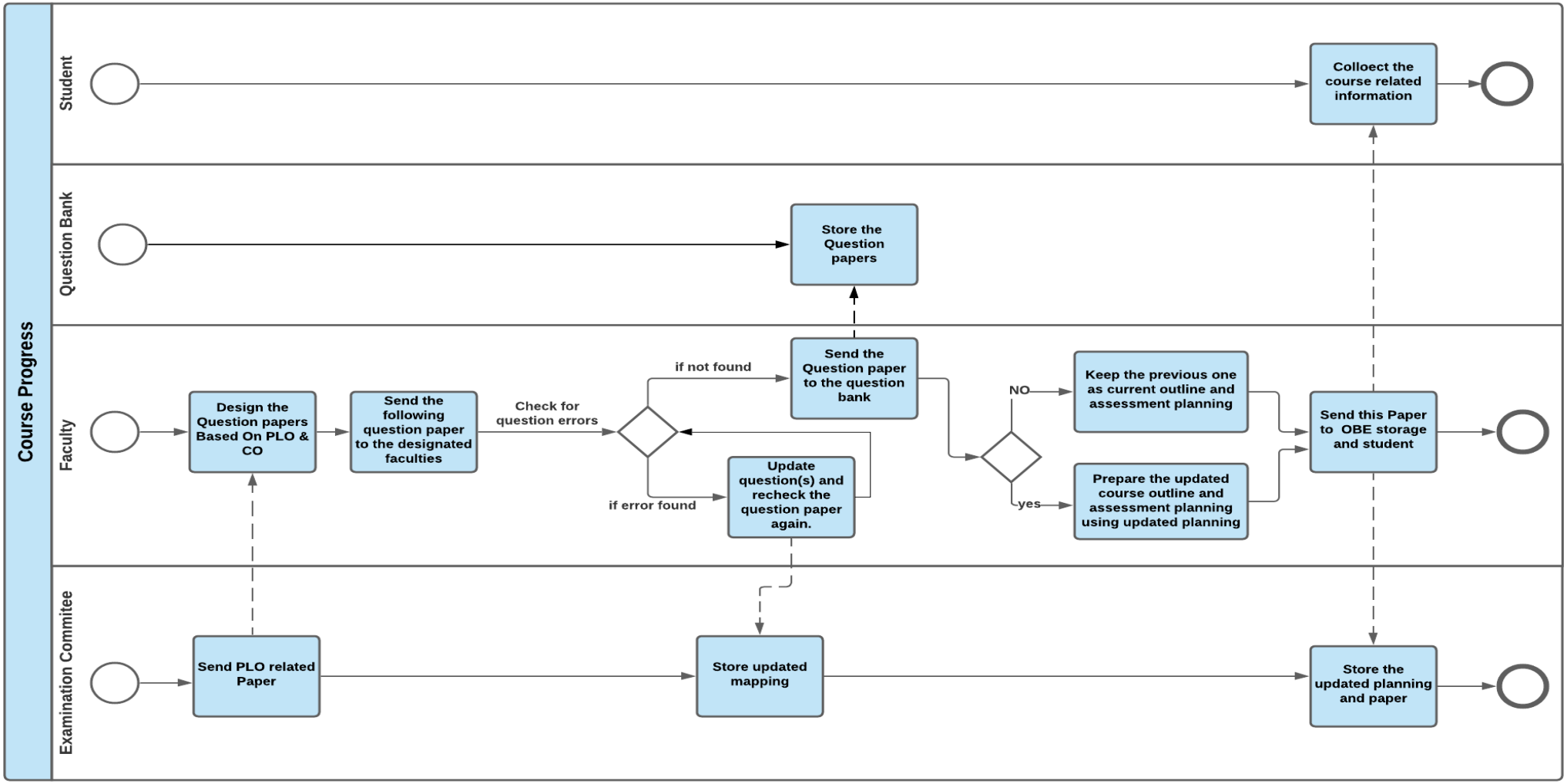


FIGURE 2.10 - Process diagram of Course progress (to-be)

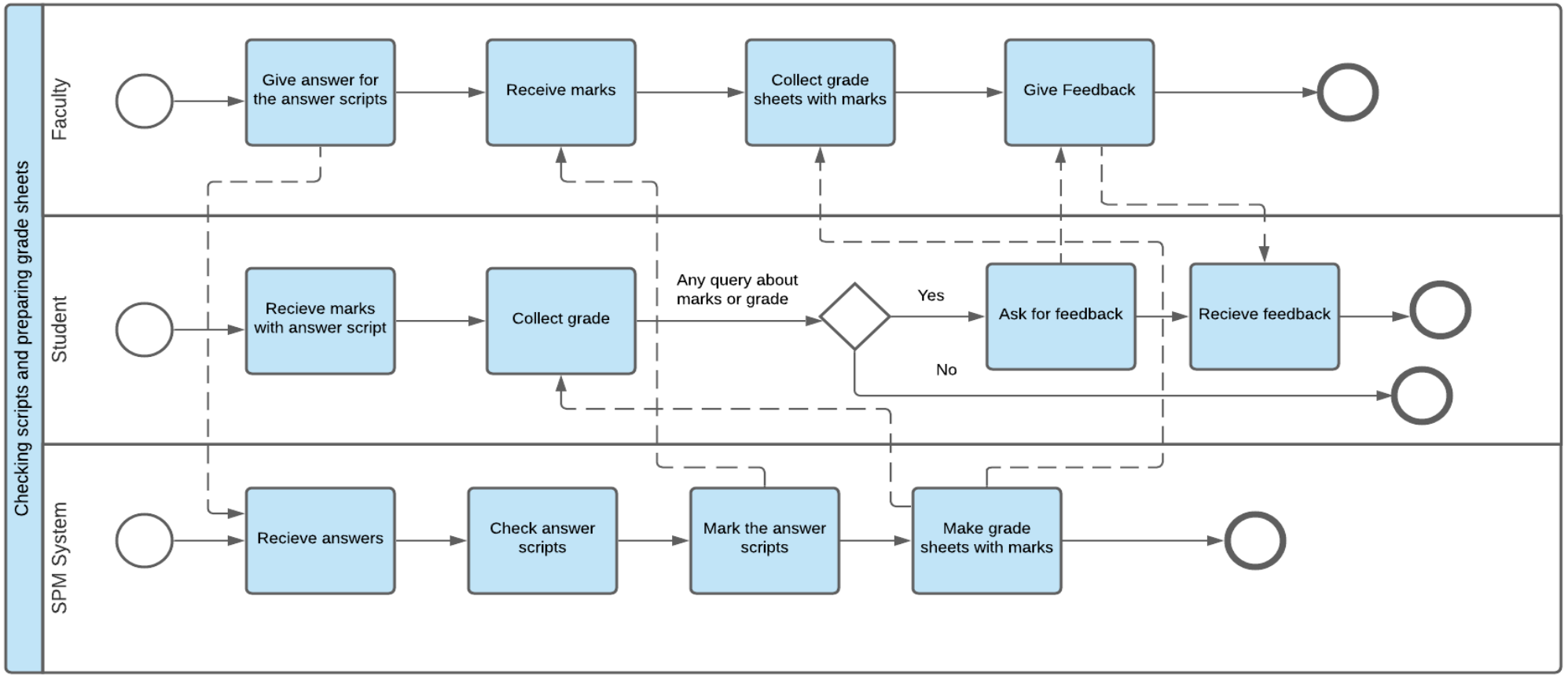


FIGURE 2.5- Process diagram of Checking scripts and preparing grade sheets (as-is)

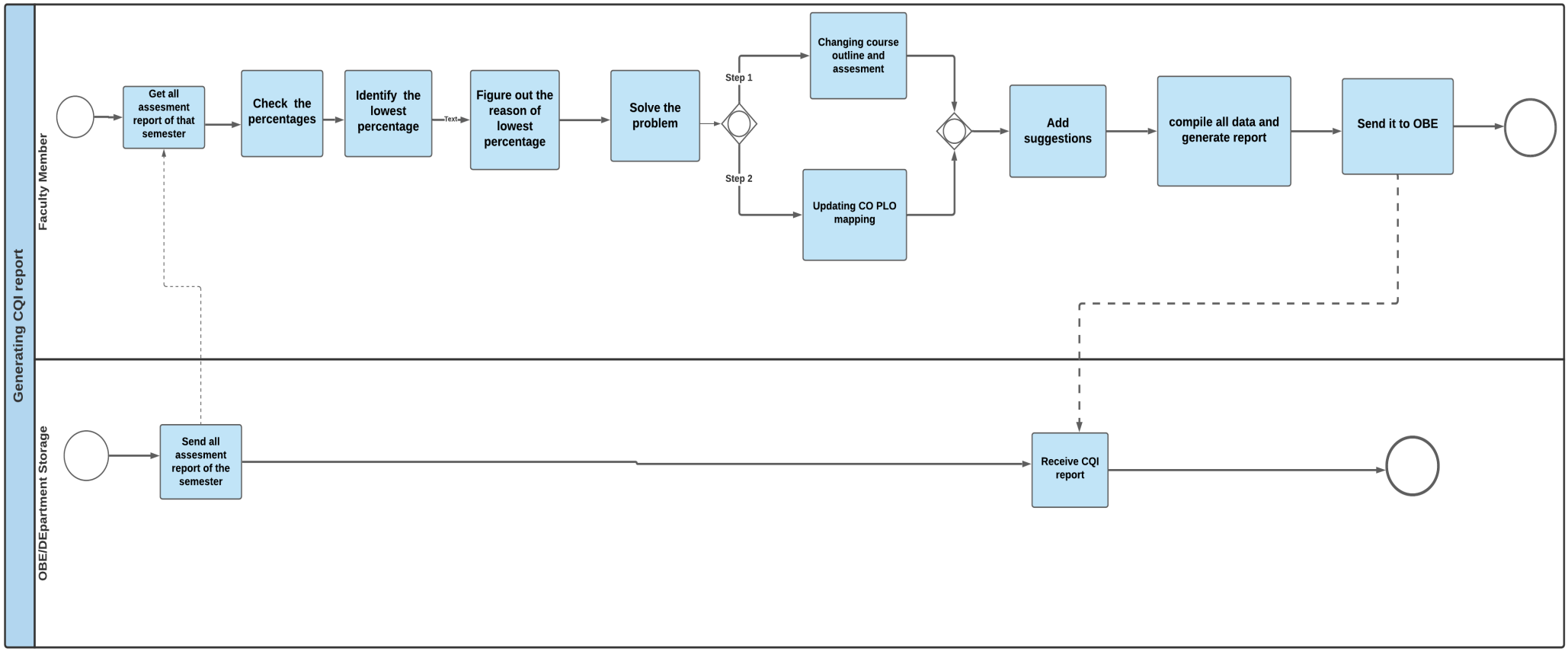


FIGURE 2.6- Process diagram of Generating CQI Report (as-is)

PROBLEM ANALYSIS

Process Name	Stakeholders	Concerns (problems)	Analysis (Reason of the Problem)	Proposed Solution
Update the Initial Mapping of PLO (course wise)	Department	Department has to do that mapping of a single course using pen and paper which is manual. In that case, if there is needed any updated solution, then the department has to prepare the mapping again manually. This is problematic.	In the manual system, the department has to prepare the mapping manually on hand and that is so time consuming and hassle.	By using this point we can solve this problem. 1. Weight or level would be assigned to each PLO and courses. 2. Based on the weight of PLO and Courses are mapped. The matrixed would be generated and the initial mapping would be done. 3. The admin can change the mapping based on the number of instances of courses having PLO and PLO having courses.
Automated mapping for a specific course and prepare course assessment planning	Faculty	When faculty members want to prepare the course planning, they have to collect the previous PLO or initial mapping of PLO. Which cannot always be perfect. Also it is time consuming. When they are preparing a mapping of PLO and CO they have to remember the number of PLO they are mapping. It is also another problem.	In the manual system, faculty members have to prepare these mapping all by themselves. So there might arise some problems when they will be mapping PLO and CO.	In our system, course labels and PLO labels will be predefined. System will suggest the number of CO and number of assessments, if faculty agree with that then a table will be shown to them for mapping the co and assessment.
Suggest questions from the question bank while making the question paper.	Faculty	Making questions takes more time and effort. Every time a faculty designs a question, he/she will have to map the COs with e questions and create the question paper manually.	The previous question papers are not possible to recycle manually and as a result, the question papers are dumped as soon as the exam ends. It is almost impossible for the faculty to retrieve the past papers of an exam.	Our system will store soft copies of all verified question papers and will show them when the faculty is going to make a question paper. For example: A faculty if going to make a mid term question paper of a course. While making the question paper, the faculty will be suggested to browse all the past mid term papers of that following course. Thus shortening the time and effort for making the mid term question paper.

Checking scripts and preparing grade sheets.	Faculty members	1. Faculty members check answer scripts individually and mark them manually. 2. Faculty members have to write all marks in grade sheets and grade them manually.	It is a long time process to check answer scripts, mark them and make grade sheets manually. It is time consuming and hassle as well.	In our system there are some automated features. 1. The system will automatically check scripts and supply the grade sheet. When it is MCQ exam only. For example: If the exam is taken in quiz format, then our system shows the marks of the MCQs and provides the answer scripts as well. 2. For the CQ exam, faculty members have to check the answer papers, mark them and give grade manually. Faculty members will submit the marks and grade to the system. Students will get the marks and grades from the system.
Generate CQI report	Faculty Members OBE storage	**1.**Faculty members collect the progress report from the OBE **2.**Faculty members check the percentage and identify the lowest percentage against each and every PLO of a student. **3.**Faculty members compile all the data and prepare the report.	The process is time-consuming since it takes time for the report to be passed from the OBE to the faculty. Faculty members might make mistakes in identifying the lowest percentage. During compilation, faculties might write some inputs wrong. Besides that, it is a hassle to identify problems and find solutions as they might be subjected to change each semester depending on the students' performance	Our system will generate the following things in our CQI report------- 1. A graph showing how many students have enrolled in each department with respect to a given period of time/semesters. 2. Course-wise student performance trend based on GPA with respect to a given period of time/semesters. 3. Instructor-wise student performance trend based on the GPA of the students with respect to a given period of time/semesters. 4. Instructor-wise student performance trend for a chosen course with respect to a given period of time/semesters. 5. Calculate the percentage of PLO corresponding to the specific course. identify the lowest percentage of each PO for every student, show the possible solutions or suggestions about improving students' performance. 6. Comparison of PLO-achieved percentage versus PLO-attempted percentage 7. Comparison of a course's expected PLO-achievement versus actual with respect to a given period of time/semesters.

RICH PICTURE TO-BE

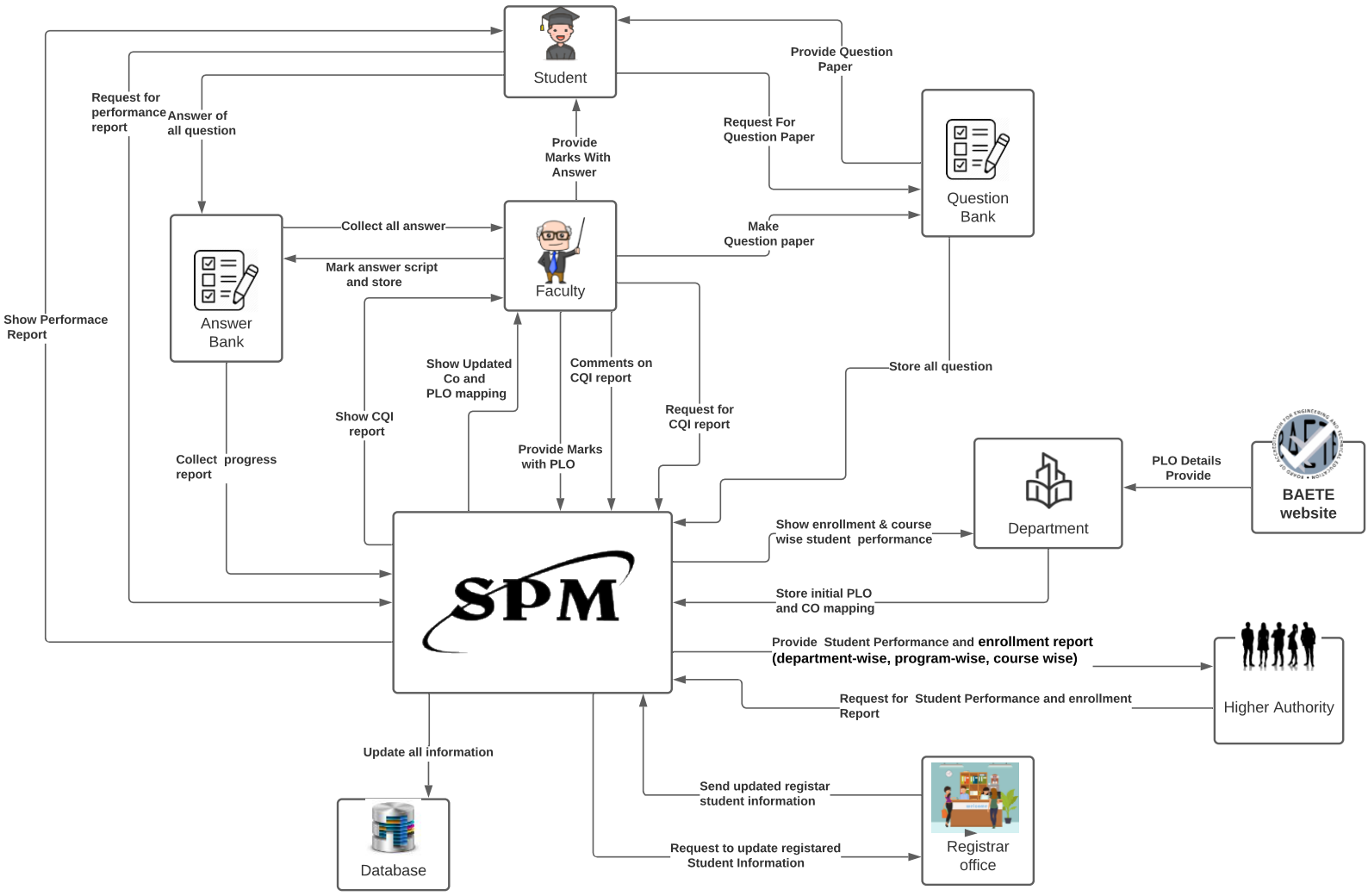


FIGURE 2.7- Rich Picture To-Be

SIX ELEMENT ANALYSIS TO-BE

Process Name	System Roles					
	Human	**Non-Computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Network & Commination**
1. Update the Initial Mapping of PLO (course wise)	**Department:** 1.Department have to go to our system's website to map the PLO and course 2. Then click on mapping to map and a new table shown. 3.In that table there are lists of PLO and all courses. And also mention that a PLO can be mapped with how many courses and one course can be mapped with how many PLO. 4. In our system PLO and courses have a label and PLO will be shown based on level. 5. That will be helpful for the department to identify the PLO mapping. 6. Then the department has to click for mapping	**Pen & Paper:** Mapped with PLO and course in paper using a pen for rough.	**Computer:** computer is needed to use our website to update the PLO and Course mapping. Also, they can view the course outline and course details.	**SPM:** To update PLO and Course mapping, SPM is needed. **Operating System:** Any OS can be used by the users, Windows, Mac	**SPM Database:** For storing the mapped Course and PLO	**Internet:** This website is being used through Online. So the Internet is needed to upload the PLO and Course planning. .

	and store it in our database.					
2. Automated mapping for a specific course and prepare course assessment planning	**Faculty** 1.Department already mapped the initial mapping of PLO and Course. 2.Faculty members have to go to our website and login their id. 3.After that, they can see their course which is assigned for them. Also can view the PLO for that course which is suggested by the department. 4.Faculty members can view the CQI report if a report is available and if needed, then faculty members can update the PLO mapping. 5.System will give faculty members some suggestions like number of course outcomes and PLO with the label wise courses and specific PLO then show the number of assessments. Then faculty members want to change something then they have to select the number of CO and	**Stationary:** Instructions of course outline and course assessment planning as CO and PLO basis details printed on paper.	**Computer**: Laptop or computer is important to use our website to select the PLO and CO mapping, assessment planning and mapping Also they can view the course CQI report and course. Based on the CQI report, if they want to change the mapping then they have to use a computer.	**SPM:** To update PLO and course outcome mapping, assessment and course outcome mapping, SPM is needed to store that information to the database. **Operating System:** Any OS used by the users, Windows, Mac	**SPM Database:** For storing the mapped Course and PLO	**Internet:** This website is being used through Online. So the Internet is needed to update the PLO and Course outcome mapping & assessment planning.

	mapped those CO with PLO. 6. After doing this, they have to select the number of assessments and also mapping with CO and assessment. 7. After updating all the mapping then click on save.					
3. Course Progress	**Faculty:** 1. Successfully log in to the faculty 2. Go to the create question paper section. 3. Select exam type and total marks of that examination. 4. Select question number. 5. Make that particular question by browsing through the past papers of that same exam type. 6. Set marks of that particular question 7. Make another question, if needed, by following the same process from 4 - 6. 8. Click on "Save" to save that question	**Pen and Paper:** Students may have to answer some part of the questions in a paper and upload it. **Calculator, ruler, pencil, eraser, sharpener.** All the necessary tools that might be needed to answer during the exam.	**Computer/ Laptop/ Smartphone:** Both the students and the faculty need a computer to attend and conduct the examination successfully.	**Internet Browser:** Any internet browsing software will be suitable. For example: Google chrome, Firefox, etc.	**SPM database:** To store faculty and student users.	**Internet:** It is used by the faculty members and students to access the SPM software and the database.

	paper for further processes. 9. Also add information about the exam, for example, syllabus for that exam, time etc. **Student:** 1. Log in to the website with the correct ID and password. 2. Go to the exam section which contains exam history and upcoming exams of all the enrolled courses of the ongoing semester. 3. Click on the upcoming exam to view the syllabus. 4. During the exam, follow part 1 and 2. Click on that upcoming exam, and view the information question paper i.e. exam syllabus and timing. 5. Upload the answer scripts (soft copy) on that exam section. 6. Rate the particular exam. (optional)					

4. Checking scripts and preparing grade sheets.	**Faculty:** 1. Faculty members have to give the answer to our system and the system will check the answer script and give marks. For example: If the exam is taken in quiz format, then the website shows the marks of the MCQs and provides the answer scripts as well. 2. Faculty members can collect grade sheets with marks in excel files from the website which is prepared by system. **Student:** 1. Students can see their marks with answer scripts from the website. 2. Students can collect their grade from the website.	**Paper:** It is used when faculty members need to print mark sheets for grade sheets.	**Computer/ Laptop:** It is needed for faculty members and students to log in SPM and check the marks and grades. **Printer:** It is used when faculty members need to print anything.	**SPM:** It is needed for checking scripts, marks and grade sheets.	**SPM Database:** It is needed to store checking scripts, marks and grade sheets.	**Internet:** It is used by the faculty members and students to access the SPM software and the database. .
5. Generate CQI report	**Faculty:** 1. Go to the website. 2. Select course. 3.Click on student performance. system will show all activities. 4. Now click on the CQI report button that will show PLO percentage.	**Paper:** It is used if a faculty wants to print something	**Computer:** It is used by faculty members to login to SPM and do their respective work. **Database Server:** Used by the faculty members to	**SPM:** It is used to generate the report. **Operating System:** Any OS can be used e.g. Windows, Mac.	**SPM Database:** It is used to store the updated report.	**Internet:** It is used by the faculty to access the SPM software and the database.

	5. Below a specific percentage of PLO faculty write down the reason. 7.Faculty members will also add some suggestions about how the performances can be improved. **Department:** 1. Go to the website. 2. Click on student performance. System will show all activities for a specific course. 3. If click on instructor wise then it will show instructor wise course performance.		access and store or update the database.			

6. Update Student enrollment information in SPM	**Registrar Office:** 1. Get notification from SPM to update student information. 2. Send the updated student enrolment report. **Higher Authority:** 1. Request to see the student enrolment report. 2. View student enrollment report as a graph.	**Paper:** It is used to print something **Pen:** It is used to write something on the report.	**Computer:** It is used by higher authority members and registrar office members to login to SPM and do their respective work. **Database Server:** Sending data from the registrar office to store or update the database.	**SPM:** It is used to update student enrollment information. **Operating System:** Any OS can be used e.g. Windows, Mac.	**SPM Database:** It is used to store the updated student enrollment information.	**Internet:** It is used by the registrar office members and higher authority to access the SPM software and the database.

PROCESS DIAGRAM TO-BE

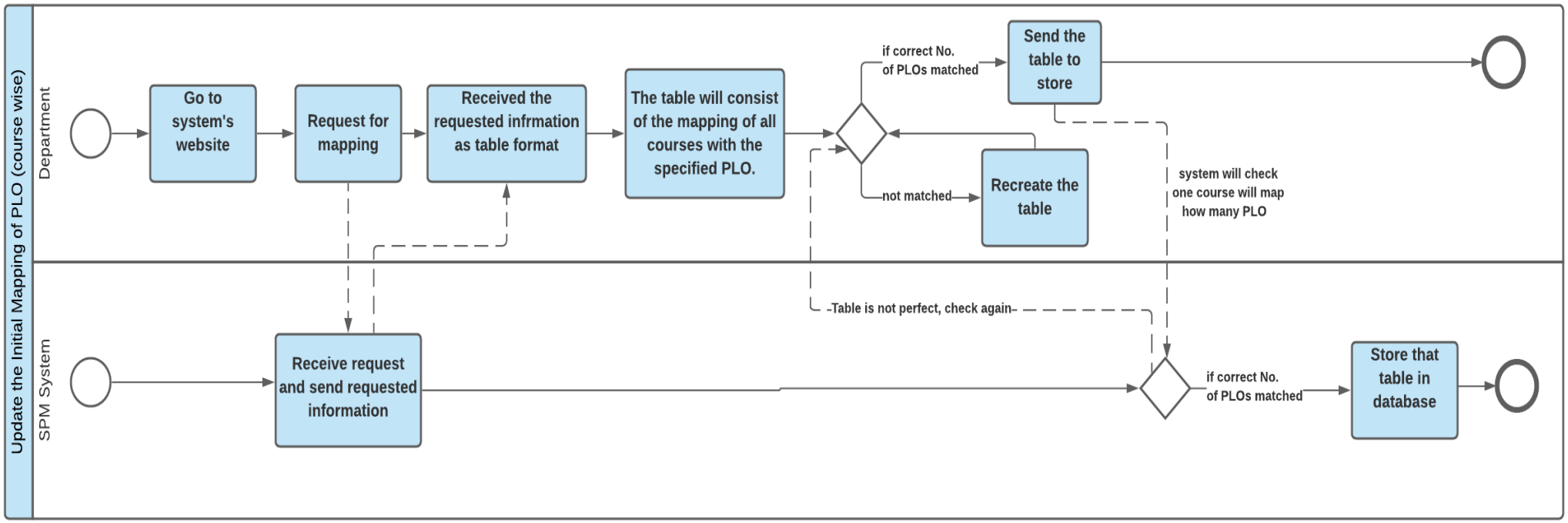


FIGURE 2.8 - Process diagram of Update initial mapping of PLO (course wise) (to-be)

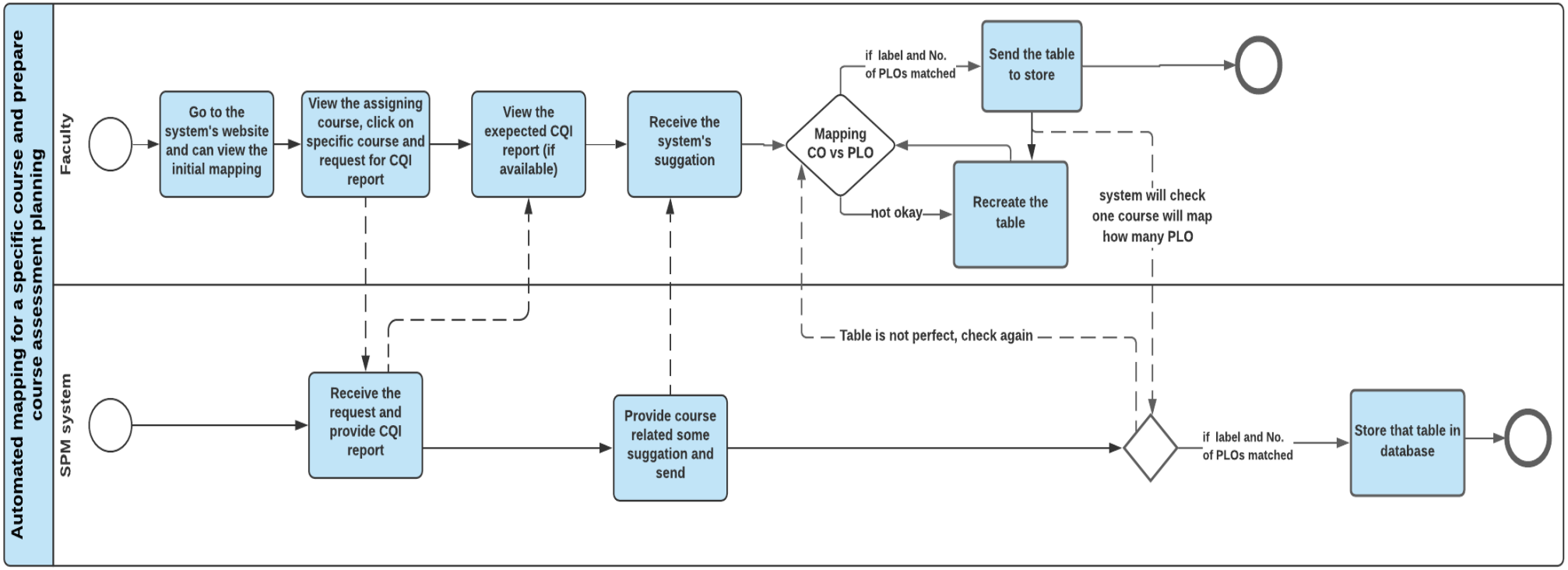


FIGURE 2.9 - Process diagram of Automated mapping for a specific course and prepare course assessment planning (to-be)

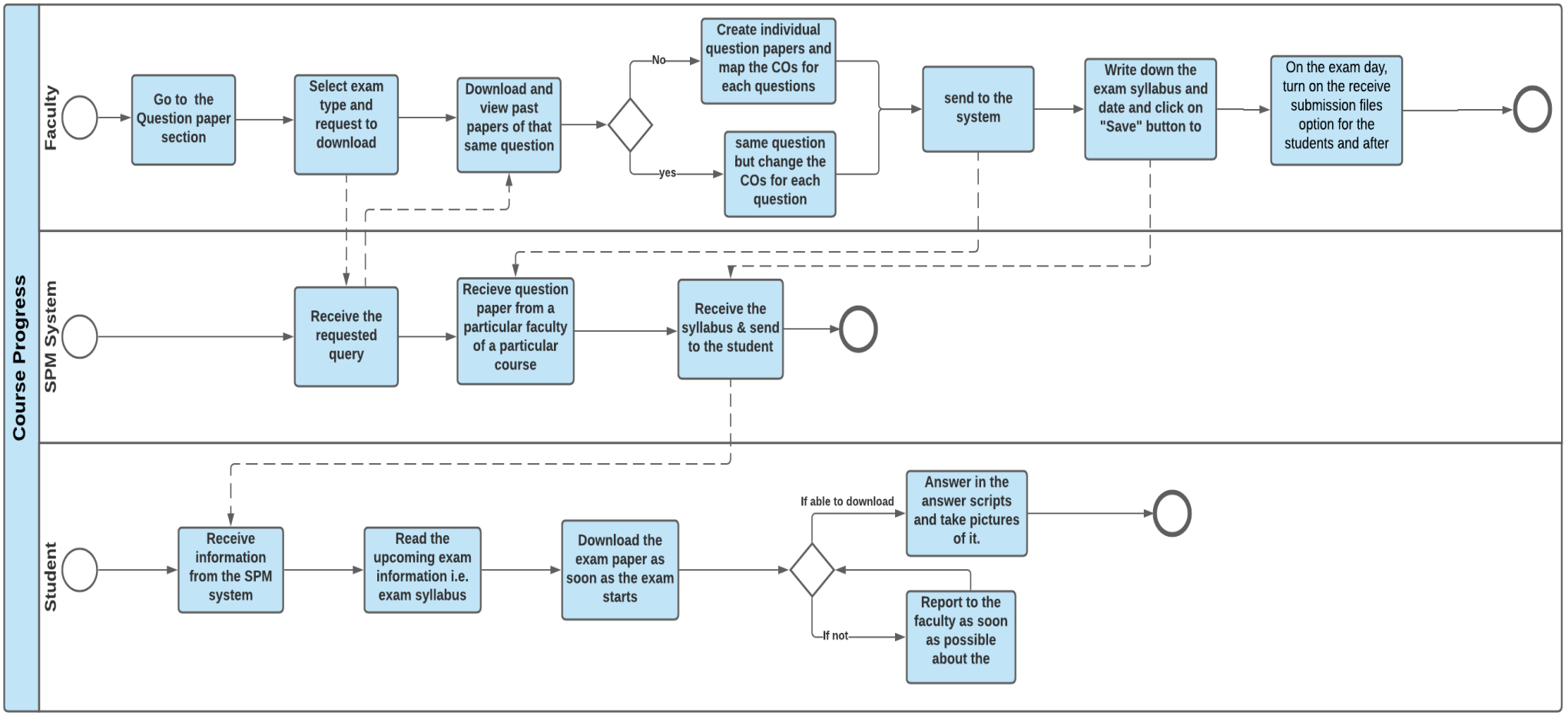


FIGURE 2.10 - Process diagram of Course progress (to-be)

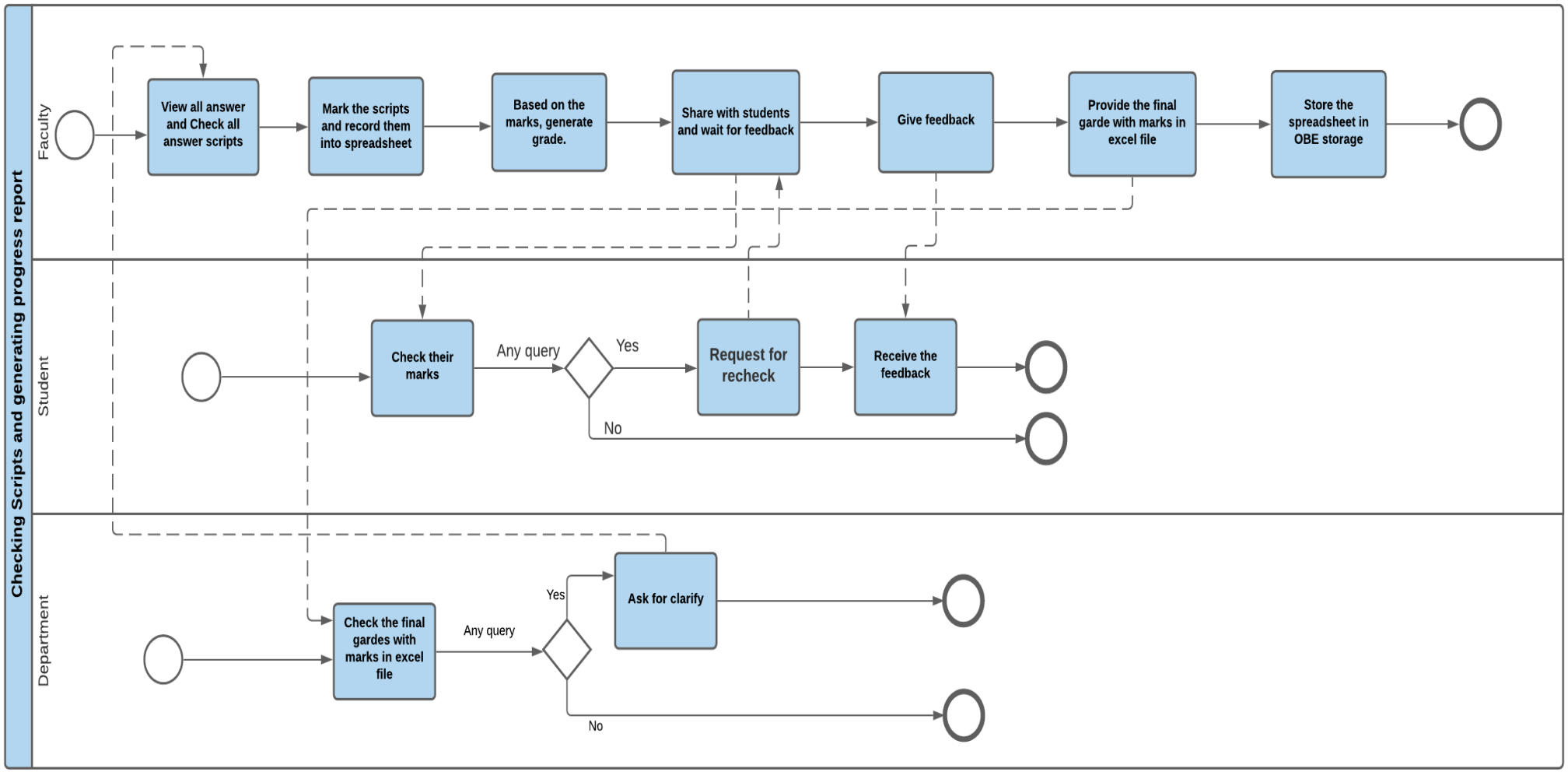


FIGURE 2.11 - Process diagram of Checking Scripts and generating progress report (to-be)

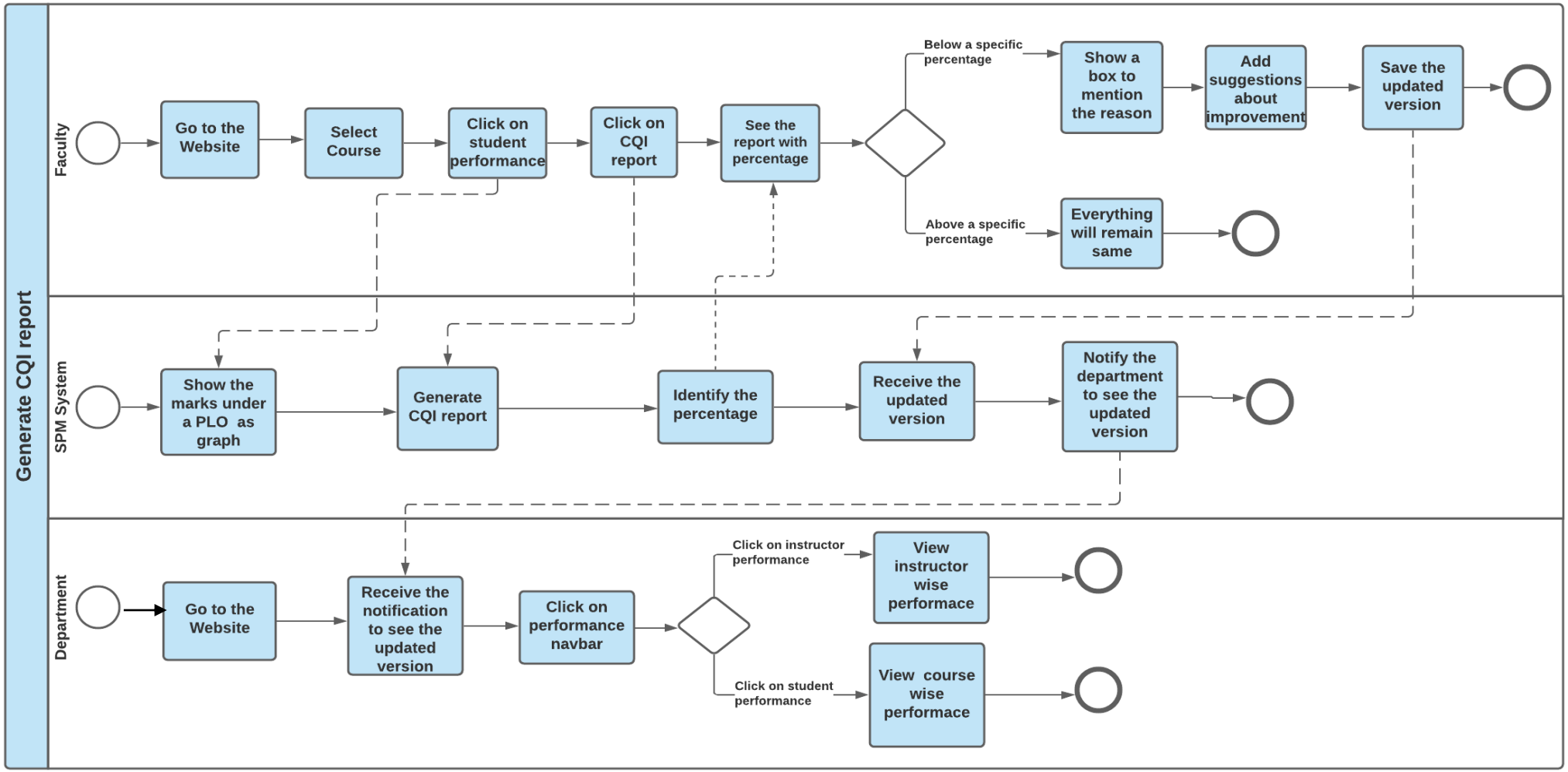


FIGURE 2.12 - Process diagram of Generate CQI report (to-be)

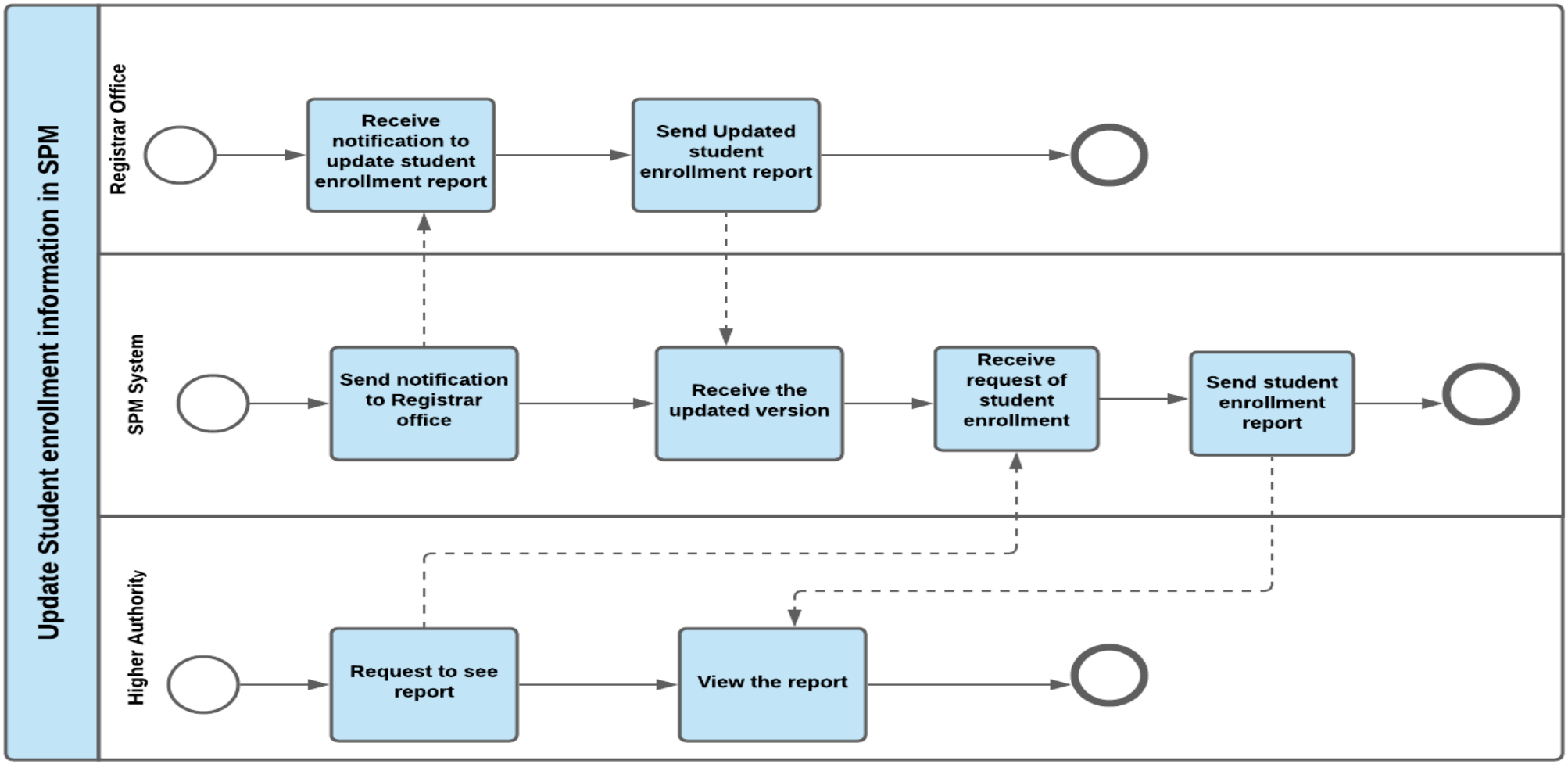


FIGURE 2.13 - Process diagram of Update student enrollment information in SPM (to-be)

CHAPTER 3

LOGICAL SYSTEM DESIGN

›› **BUSINESS RULES**

›› **ERD**

›› **ERD TO RELATION**

›› **NORMALIZATIOM**

›› **DATA DICTIONAERY**

SECTION 3: LOGICAL SYSTEM DESIGN

BUSINESS RULES

The goal of the software is to increase efficiency in monitoring students' performance. The SPM system contains all the PLO (Program Learning Outcome) and CO (Course Outcome), initial mapping of PLO and CO, course outline of every course, assessment planning, relation with assessment and course outcome, comments on CQI report based on PLO percentage, past question papers and their answers of all subjects.

Every university has a name, unique id and location. There are different schools in a university which has a name and id. Under a school there are different programs which has name and id. There are many departments as well. But we are working only with the CSE department.

Department has a name, some faculty members and a particular faculty member as head of the department. Department heads usually do the initial mapping of Course with PLO in the system. Courses have a course id (unique), course Name against which it will be mapped with PLO. PLO has an id (unique), name, details, level. Departments offer multiple courses, each of which has a name, unique course id, credit hour, and a course description. A course has multiple course outcomes. A single CO can be mapped with multiple PLO. Every course outcome has an id(unique), description, level.

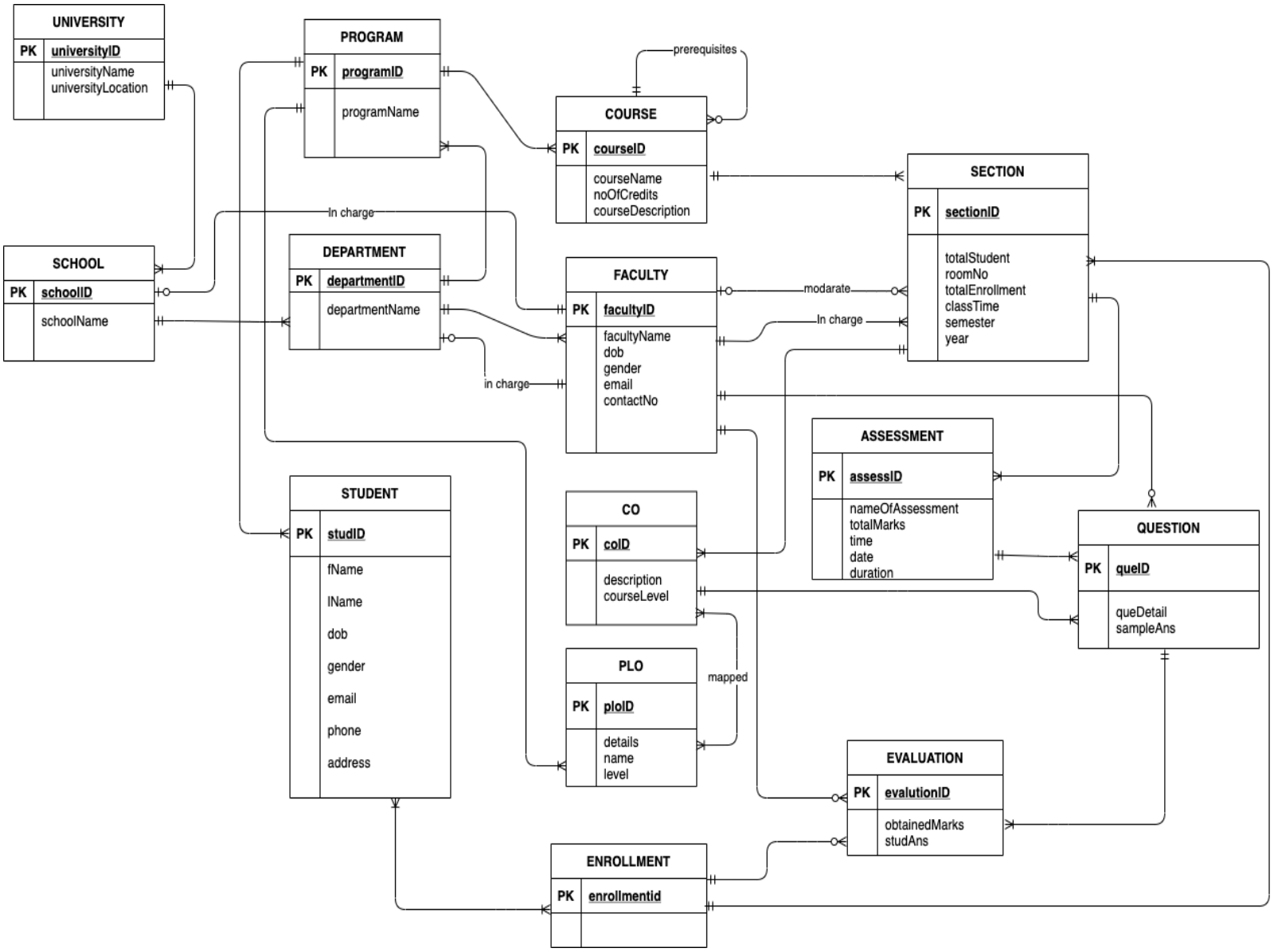
In a university the most important role is played by its students. Students have id, name (fast name, last name), date of birth, gender, email, phone number, address. Each semester students enroll in one or many courses which has an enrollment id. Each course might have multiple sections. Sections have id, room No, total number of students capacity, total number of enrollments, class time, semester and year. Every course has at least one section.

At the beginning of the semester faculty members are assigned to the courses in different sections. One faculty member can conduct multiple sections of a single course. Faculty members have a unique identification number, name (first name, last name), date of birth, gender, email and contact number. In every section there are multiple assessments. In assessment there are assessment id, name of assessment, total marks, time, date and duration.

In assessment, there are multiple questions. In the question section there are question id, question details, sample answer. Students will sit for the exam and Faculty members will evaluate them accordingly. Evaluation has unique id, obtained marks and students answer.

Faculty members don't have to check the answers manually as the SPM system will automatically check the scripts through the help of the sample answer and make evaluation reports using those marks. This way our system will add remarkable changes in our education system.

ENTITY RELATIONSHIP DIAGRAM



FIRURE 3.1 – Entity Relationship Diagram of SPM

ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA

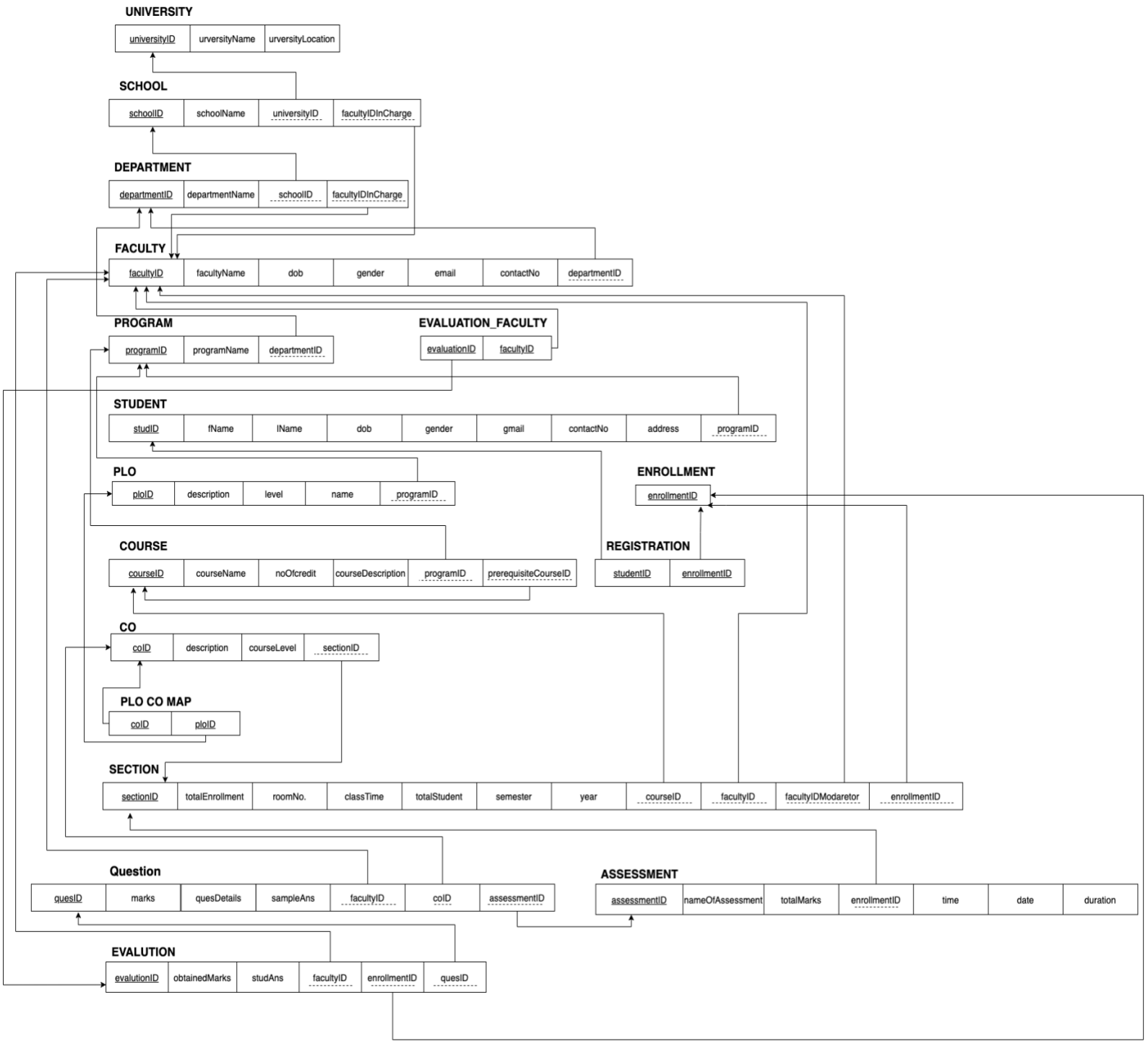


FIGURE 3.2 – Relational Schema Diagram of SPM

NORMALIZATION

As we made a relational schema from ERD based on theory that's why we think normalization is not needed here.

However, we tested normalization on every relation. Those perfectly fulfill the requirement of normalization.

DATA-DICTIONARY

University

Name	Type	Size	Remark
universityID	INTEGER		This is the primary key of University. example: for IUB university ID will be 1.
universityName	VARCHAR	100	This is the primary key of University. Example: "Independent University, Bangladesh"
universityLocation	TEXT		Here will store the location of the university.

School

Name	Type	Size	Remark
schoolID	VARCHAR	6	This is the Primary Key of School. Example: “SETS”.
schoolName	VARCHAR	45	This is the name of the School. Example: “School of Engineering, Technology and Science”

Student

Name	Data Type	Size	Remark
studentID	VARCHAR	7	This is the primary key of this relationship. This contains the ID of the student. Example : “1731407”
fname	VARCHAR	30	This is the first name of the student. Example : “Nusrat”
lname	VARCHAR	15	This is the last name of the student. Example : “Zahan”
dob	DATE	DD-MM-YYYY	This the Date of Birth of the Student. Example: “12-06-1997”
gander	VARCHAR	6	This is the gender of the Student. Example: “Female” or “Male” or “Other”
email	VARCHAR	30	This is the email address of the Student. Example: “1730129@iub.edu.bd”
phone	VARCHAR	14	This is the phone number of the Student. Example:”+8801711234567”
address	TINYTEXT		This is the address of the Student. Example: “house No: 129, Road-4, A block, Bashundhara R/A, Dhaka, 1299”

Department

Name	Data Type	Size	Remark
departmenShortName	VARCHAR	7	This is the Primary Key of the Department. Example: “CSE”
departmentName	VARCHAR	35	This contains the course title of a particular course. Example: ‘Computer Science and Engineering

Program

Name	Data Type	Size	Remark
programID	VARCHAR	7	This is the Primary Key for a Program Example: ”B.Sc”.
programName	VARCHAR	30	This is the name of the Degree Program. Example: “Bachelor of Science”

Course

Name	Data Type	Size	Remark
courseID	VARCHAR	9	This is the Primary Key for the Course. ”Example: “CSE203”
courseName	VARCHAR	50	This is the name of the Course. Example: ”Database Management”
noOfCredits	INTEGER		This is the credit for the Course. Example: ”3”
course description	TEXT		The TEXT data type can hold up to 64 KB. This is the description of the Course.

CO

Name	Data Type	Size	Remark
coID	VARCHAR	27	This is the Primary Key for Course Outcome. Example:1
description	TEXT		This is the details of the course outcome.
courseLevel	INTEGER		Level of CO. For example: 400 level courses like cse437.

PLO

Name	Data Type	Size	Remark
ploID	VARCHAR	**5**	This is the Primary Key for the PLO. ”Example: “2”
name	VARCHAR	35	This is the name of PLO. Example: “Design/development of solutions”
details	TEXT		This is the details of the Program Learning Outcome.
level	INTEGER		Level of PLO . Example: 1

Enrollment

Name	Data Type	Size	Remark
enrollmentid	VARCHAR	**25**	This is the Primary Key for the enrollment id.”Example: “2”

Section

Name	Data Type	Size	Remark
sectionID	INTEGER		This is the Primary Key for Section. Example : “2”
totalStudent	INTEGER		This is the total no of student of section

			Example : “40”
roomNo	VARCHAR	7	This is the room no of the section. Example : “BC4010”
totalEnrollment	INTEGER		This is the total no of enrollment of the section.Example : “50”
classTime	VARCHAR	17	This is the class time of the section Example : “MW 8:00 AM - 9:30 AM”
semester	VARCHAR	6	This is the name of the semester's section. Example : “Autumn”
year	DATE		This is the year of the section Example : “2021”
Day	VARCHAR	2	

Assessment

Name	**Data Type**	**Size**	**Remark**
assessmentID	VARCHAR	29	This is the Primary Key for the assessment id. Format: “courseID_section_semester_year_typeofAssess” Example: “CSE101_01_2_2021_01”
nameOfAssessment	VARCHAR	8	This entity stores the name of assessment. Example: “Final”
totalMarks	INTEGER		Here will store the total marks of assessment. Example: 40
time	VARCHAR	**9**	
date	DATA		

duration	VARCHAR	5	

Evaluation

Name	Data Type	Size	Remark
evaluationID	VARCHAR	33	This is the Primary Key for the evaluation id. Format: "courseID_section_semester_year_typeofAssess_studID" Example: "1730016_CSE101+L_01_summer_2021_01_"
obtainedMarks	FLOAT		Here will store the total obtained marks of a student. Example: 30.5
studAns	LONGTEXT		This entry will store the student's answer.

Question

Name	Data Type	Size	Remark
queID	VARCHAR	29	This is the Primary Key for the assessment id. Format: "courseID_section_semester_year_typeofAssess_questionNumber" Example: "CSE101_01_summer_2021_01_001,".
queDetail	TEXT		This is the details of the question.
sampleAns	TEXT		This is the sample answer of the question.
marks	INT		Each question mark.

CHAPTER 4

PHYSICAL SYSTEM DESIGN

›› INPUT FORMS

- **Purpose**
- **Controls and flow controls of the form**
- **Related SQL Used**

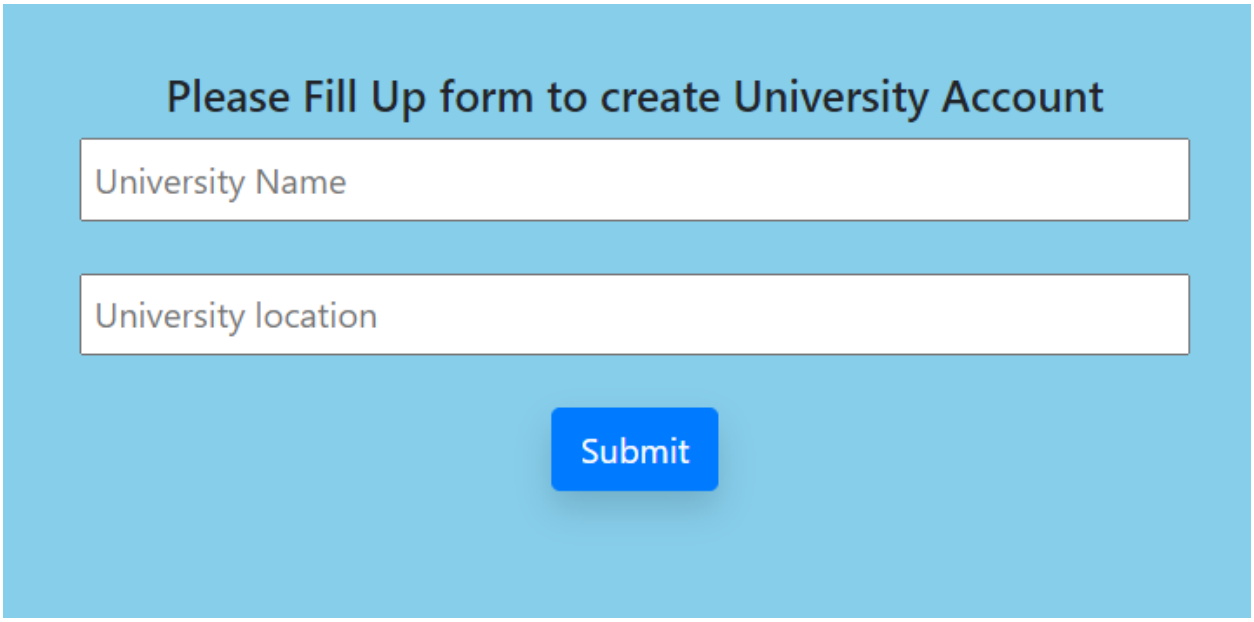
›› OUTPUT QUERY AND REPORTS

- **Purpose and use**
- **Controls and flow of controls**
- **Description along with SQL**

SECTION 4: PHYSICAL SYSTEM DESIGN

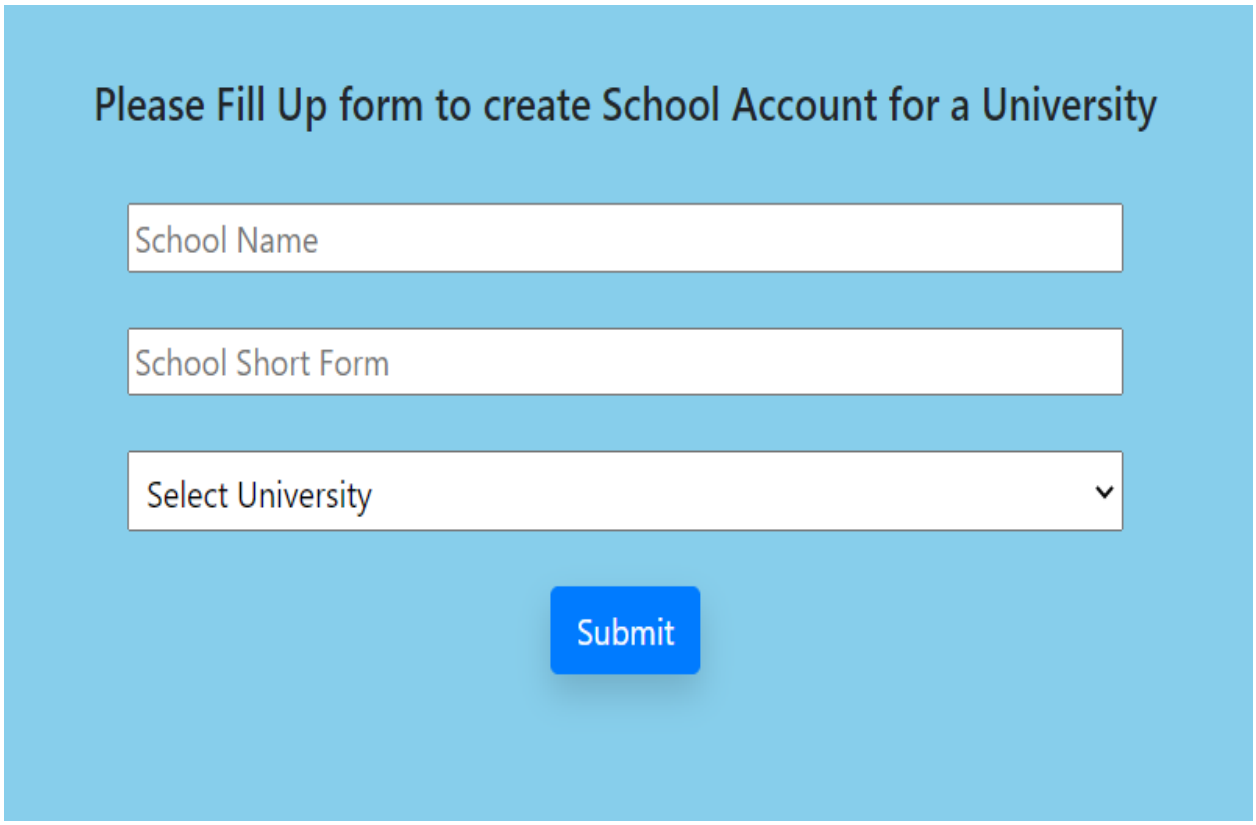
INPUT FORMS

University Account Create Form:



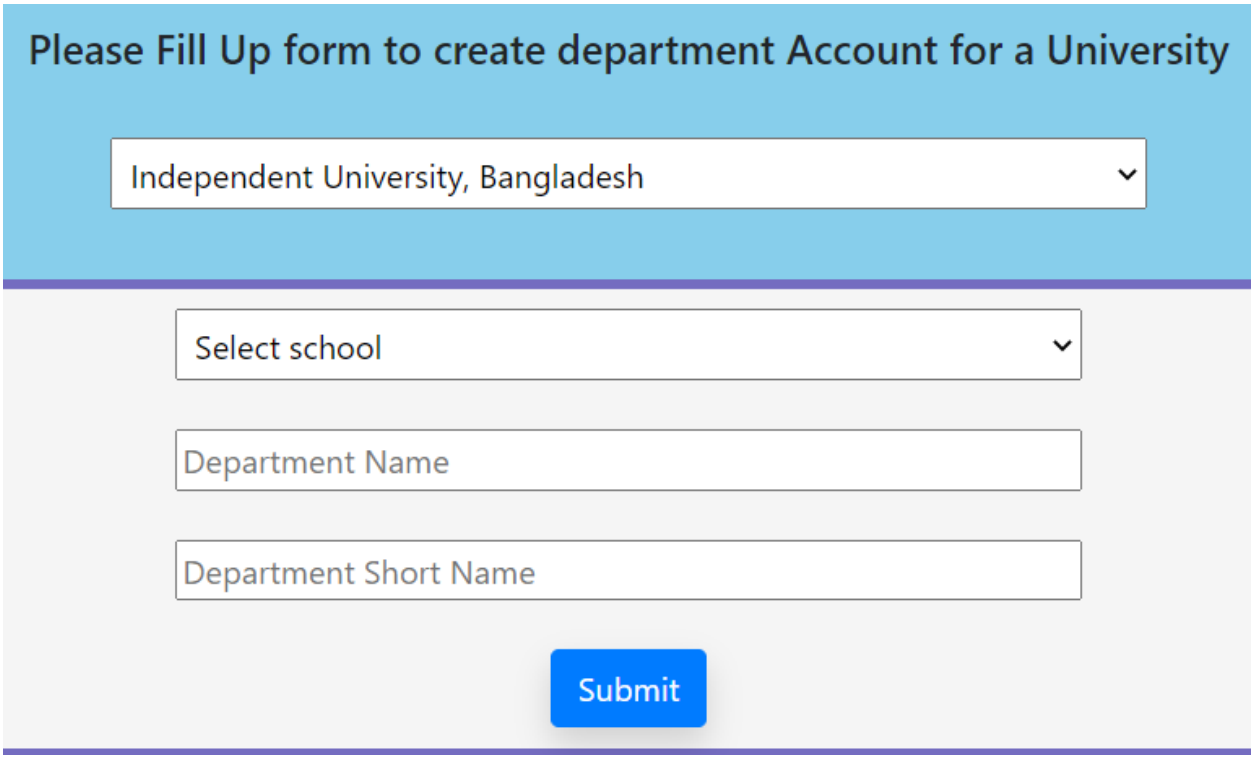

```
<?php
   $con = mysqli_connect('localhost','root');
   mysqli_select_db($con, 'spm');

   $uName=null;
   $uLoc=null;
   $uName = $_POST['uniName'];

   $uLoc = $_POST['location'];

   $insert = "INSERT INTO university(universityName,universityLocation)
values('$uName', '$uLoc')";
   mysqli_query($con, $insert);
   echo $uName;
?>
```

School Account Create Form:




```
<?php
   $con = mysqli_connect('localhost','root');
   mysqli_select_db($con, 'spm');

   $name = $_POST['schoolName'];
   $shortname = $_POST['sShortName'];
   $uniid = $_POST['uni'];

   $insert = "INSERT INTO school(schoolID ,schoolName, universityID)
              values('$shortname', '$name','$uniid')";
   mysqli_query($con, $insert);
 ?>
```

Department Account Create Form:



Query - 1: For University Selection

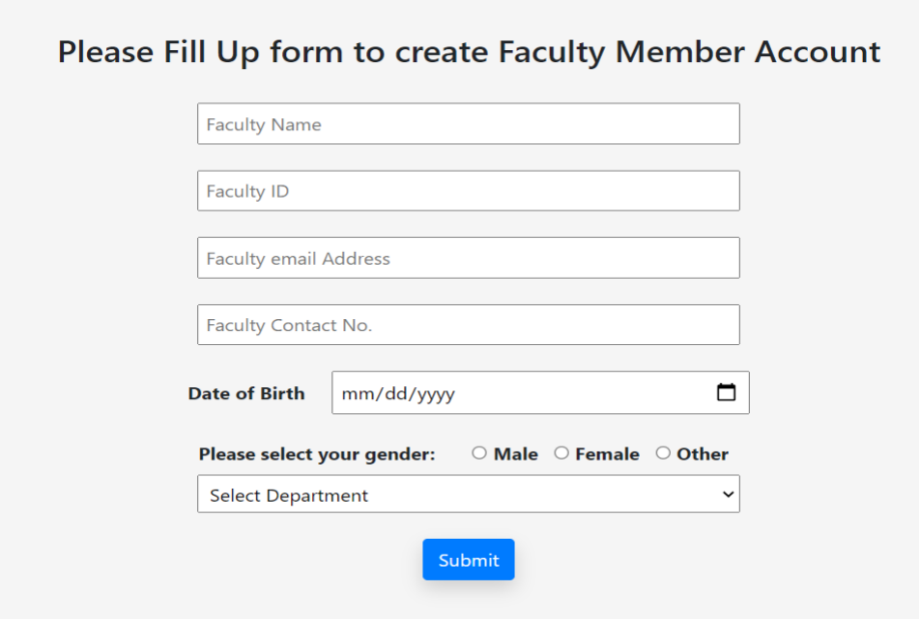
```
<?php

            $uniName = null;
            $q = "SELECT * from university";
            $query = mysqli_query($con, $q);
            while($travese = mysqli_fetch_array($query)){
                echo "<option value = ".$travese['universityID']
."">".$travese['universityName']."</option>";
            }
        ?>
```

Query - 2: Department ID

```
<?php
    $universityID = $_SESSION['value'];
    $schoolID = $_POST['school'];
    $deptName = $_POST['deptName'];
    $deptShortName = $_POST['deptShortName'];
```

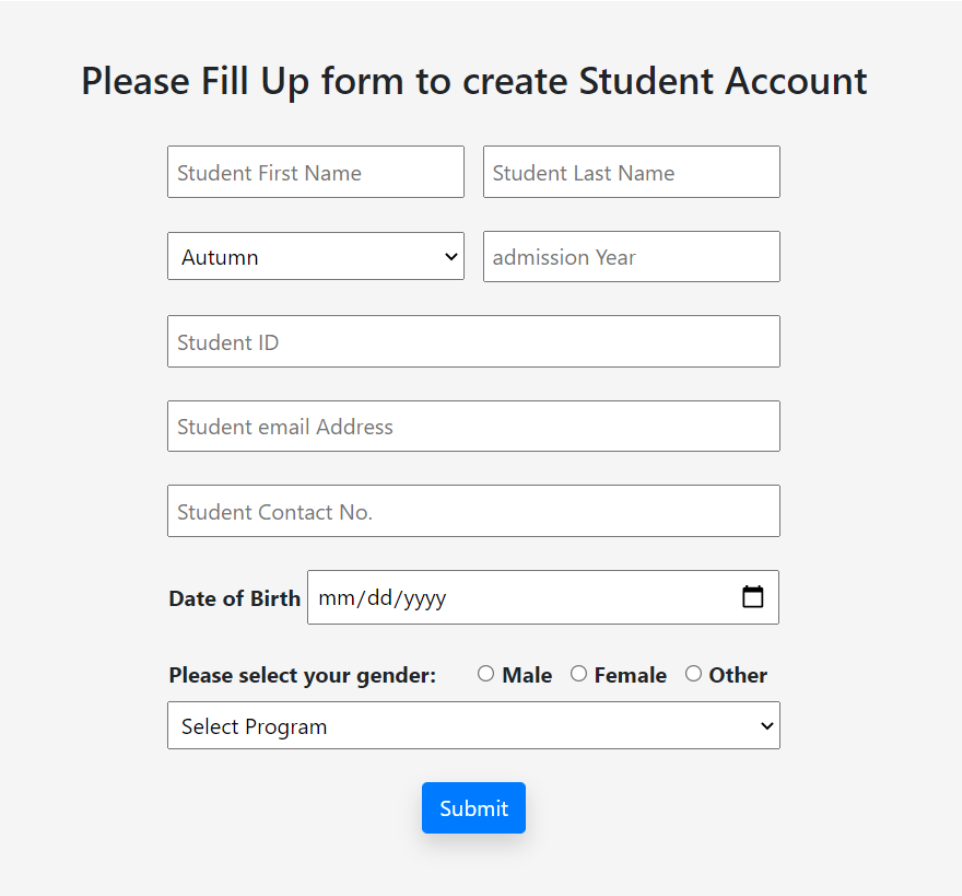
```
    $insert = "INSERT INTO department(deptShortName,deptName,schoolID,universityID)
values('$deptShortName', '$deptName','$schoolID','$universityID')";
    mysqli_query($con, $insert); ?>
```

Faculty Member Account Form:



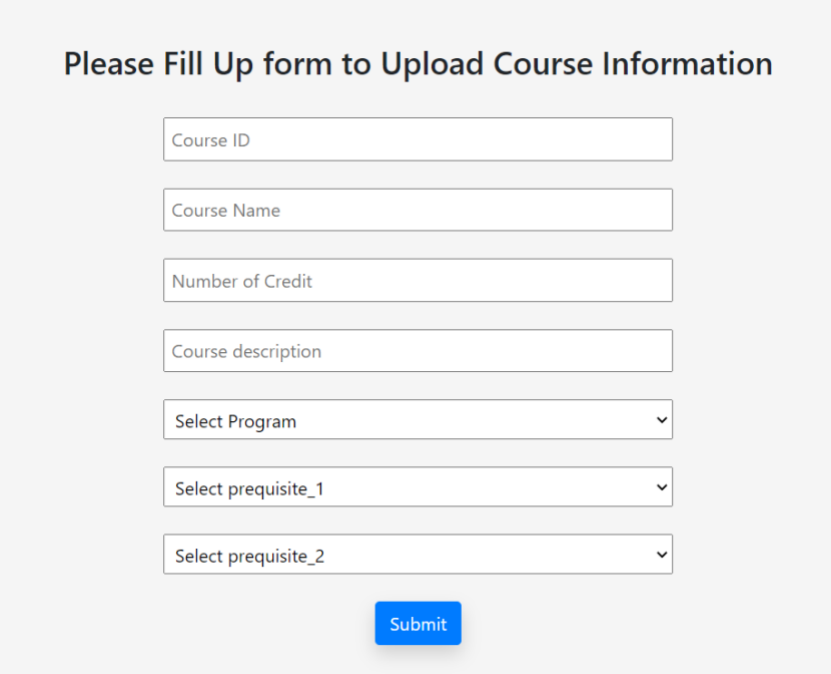

```
<?php
    include '../../connection.php';
    $name =  $_POST['name'];
    $id =  $_POST['fid'];
    $email = $_POST['email'];
    $contact = $_POST['phoneNo'];
    $dob = $_POST['dob'];
    $gender = $_POST['gender'];
    $dept = $_POST['dept'];



    $insert = "INSERT INTO faculty(facultyID, facultyName, dob, gender, email,
contactNo, deptShortName)
                values('$id', '$name','$dob', '$gender','$email','$contact','$dept')";
    mysqli_query($con, $insert);
?>
```

Student Account Create Form:



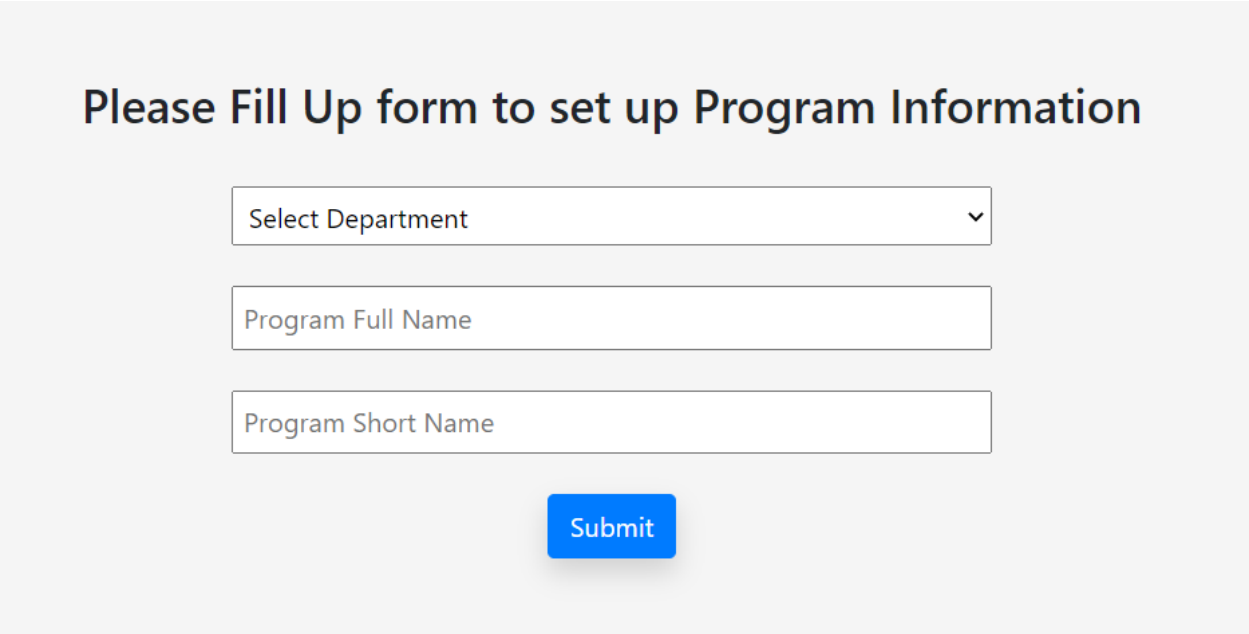

```
<?php
   include '../../connection.php';

   $courseID =  $_POST['courseID'];
   $name =  $_POST['name'];
   $noOfCredit = $_POST['noOfCredit'];
   $description = $_POST['description'];
   $program = $_POST['program'];
   $pro = $_POST['pro'];
   $pri = $_POST['pri'];

   $insert = "INSERT INTO course(courseID, courseName, noOfCredit, courseDescription,
programID, prequisiteCourseID1, prequisiteCourseID2)
            values('$courseID','$name', '$noOfCredit','$description','$program','$pro','$pri')";
   mysqli_query($con, $insert); ?>
```

Upload Course Info Form:



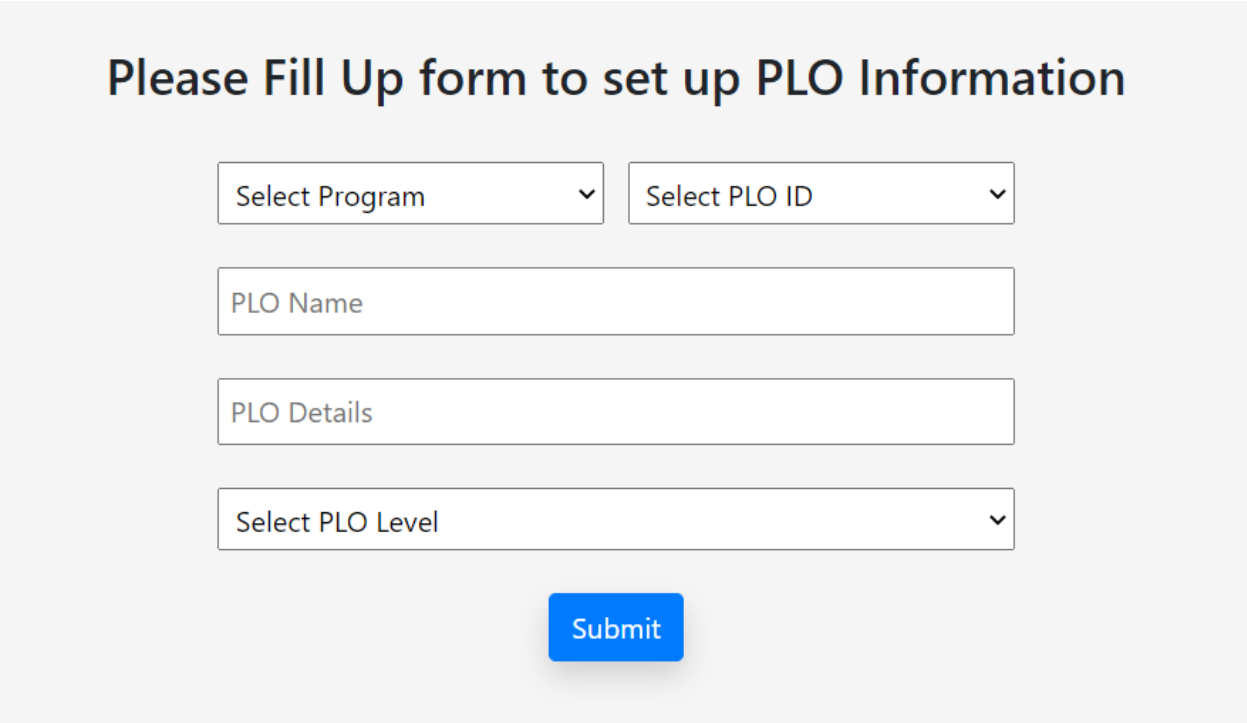

```
<?php
   include '../../connection.php';

   $courseID =  $_POST['courseID'];
   $name =  $_POST['name'];
   $noOfCredit = $_POST['noOfCredit'];
   $description = $_POST['description'];
   $program = $_POST['program'];
   $pro = $_POST['pro'];
   $pri = $_POST['pri'];

   $insert = "INSERT INTO course(courseID, courseName, noOfCredit, courseDescription,
programID, prequisiteCourseID1, prequisiteCourseID2)
            values('$courseID','$name', '$noOfCredit','$description','$program','$pro','$pri')";
   mysqli_query($con, $insert);

?>
```

Setup Program Info Form:



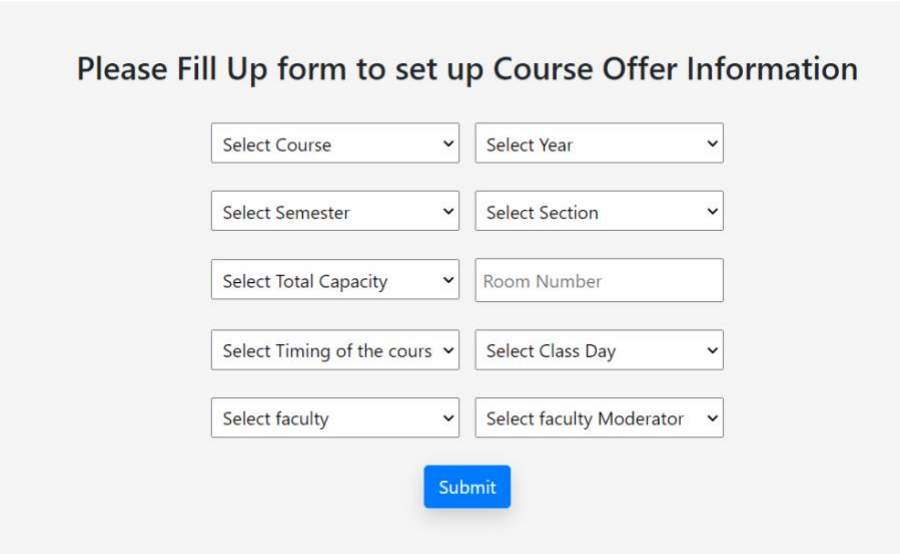

```
<?php
    include '../../connection.php';
    $fullname =  $_POST['fullname'];
    $shortname =  $_POST['shortname'];
    $dept = $_POST['dept'];

    $insert = "INSERT INTO program(programID,programName,deptShortName)
values('$shortname','$fullname', '$dept')";
    mysqli_query($con, $insert);

?>
```

PLO Info Input Form:




```
<?php
   include '../../connection.php';

   $program =  $_POST['program'];
   $plo =  $_POST['plo'];
   $name =  $_POST['name'];
   $details =  $_POST['details'];
   $ploLevel =  $_POST['ploLevel'];


   $insert = "INSERT INTO plo(ploID,name,deatails,lavel,programID)
            values('$plo','$name','$details','$ploLevel','$program')";
   mysqli_query($con, $insert);

?>
```

Input form of Course Offer Information:



For course:

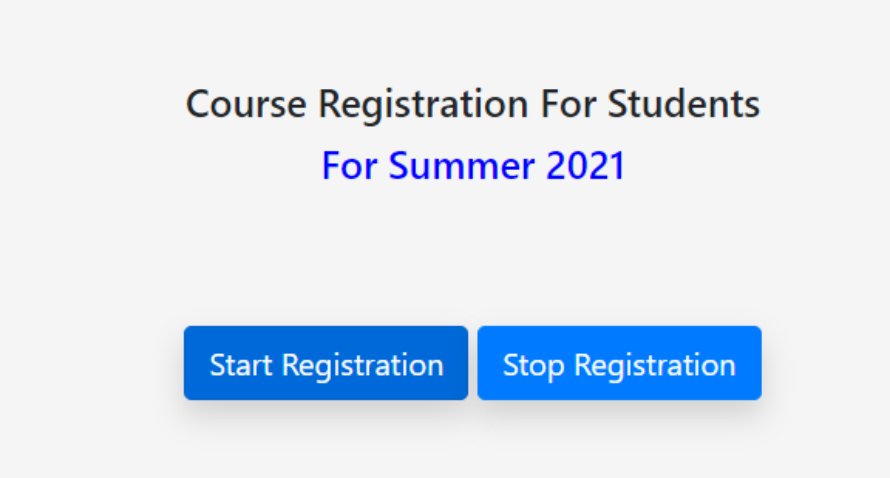
```
<?php

            $q = "SELECT * from course";
            $query = mysqli_query($con, $q);
            while ($travese = mysqli_fetch_array($query)) {
               echo "<option value = " . $travese['courseID'] . ">" . $travese['courseName'] .
"</option>";
            }
            ?>
```

Course offer:

```
<?php
   include '../../connection.php';

   $courseID =  $_POST['course'];
   $year =  $_POST['year'];
   $sem =  $_POST['sem'];
   $section =  $_POST['section'];
   $totalEnrollment =  $_POST['totalEnrollment'];
   $room =  $_POST['room'];
   $timing =  $_POST['timing'];
   $day =  $_POST['day'];
```

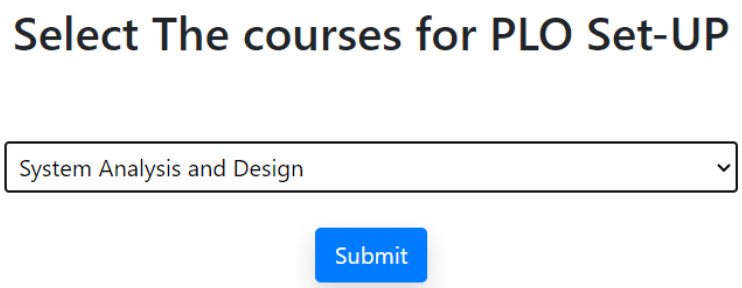
```
    $faculty = $_POST['faculty'];
    $invisilatorfaculty = $_POST['invisilatorfaculty'];
    $totalNumberOfStdent = 0;
    $enrollmentID = $courseID."_".$section."_".$sem."_".$year ;


    $insert = "INSERT INTO
section(sectionID,courseID,year,semester,totalEnrollment,roomNo,classTime,day,
                    totalNumberOfStudent,facultyID,facultyIDModaretor,enrollmentID)
            values('$section','$courseID','$year','$sem','$totalEnrollment','$room','$timing','$day',
                    '$totalNumberOfStdent','$faculty','$invisilatorfaculty','$enrollmentID')";
    mysqli_query($con, $insert);
    echo $enrollmentID;
  ?>
```

Course Registration Input Form:



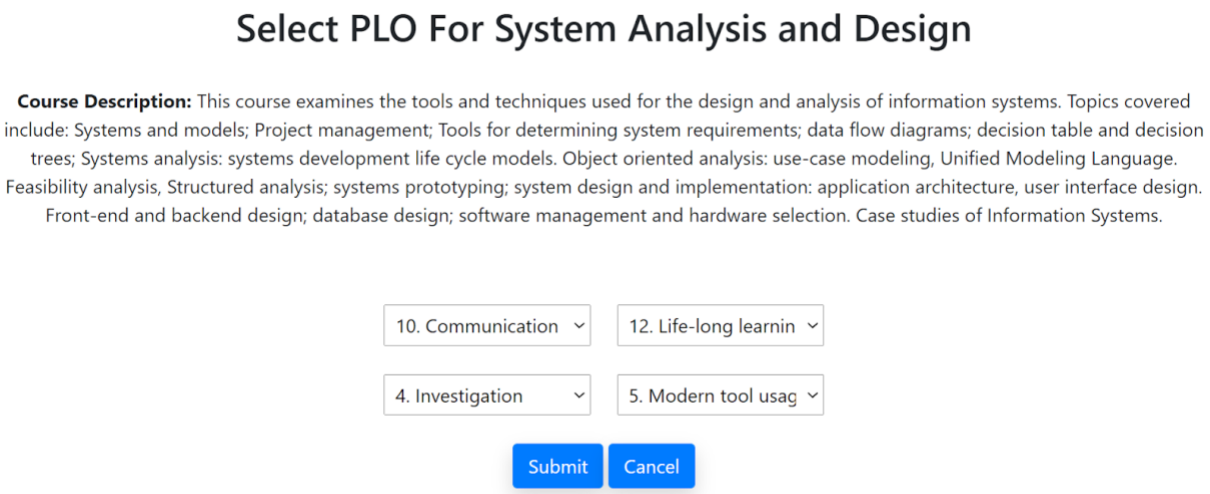

```
<?php
include '../../connection.php';
$r = $_POST['status'];
$registrion = "UPDATE activeness
            SET status = '$r'
            WHERE name = 'registration'";
mysqli_query($con, $registrion);

?>
```

PLO Set up for a course (by department Head):




```
<?php
        $q = "SELECT distinct c.courseID AS course, c.courseName AS nam FROM course
AS c
            WHERE c.courseID NOT IN (SELECT distinct p.courseID FROM
ploinitialmapping AS p)";

        $query = mysqli_query($con, $q);
        while($travese = mysqli_fetch_array($query)){
           echo "<option value = ".$travese['course'] .">".$travese['nam']."</option>";
        }
    ?>
```

Select PLO of a Course:




```
<form action="ploMappingInsert.php" id="form" method="POST" align="center">
      <?php
         $q = "SELECT * FROM course AS c WHERE c.courseID = '$c'";
         $query = mysqli_query($con, $q);
         $travese = mysqli_fetch_array($query);
         $cName = $travese['courseName'];
         $courseDescription = $travese['courseDescription'];
      ?>

      <h2>Select PLO For <?php echo $cName?></h2> <br>
      <p><b>Course Description: </b><?php echo $courseDescription?></p> <br> <br>

      <select name="plo1" style="width:17%; padding:5px">
         <option selected disabled>Select PLO </option>
 <?php
         $q = "SELECT * FROM plo AS p WHERE p.lavel <= '$label'";
         $query = mysqli_query($con, $q);
```

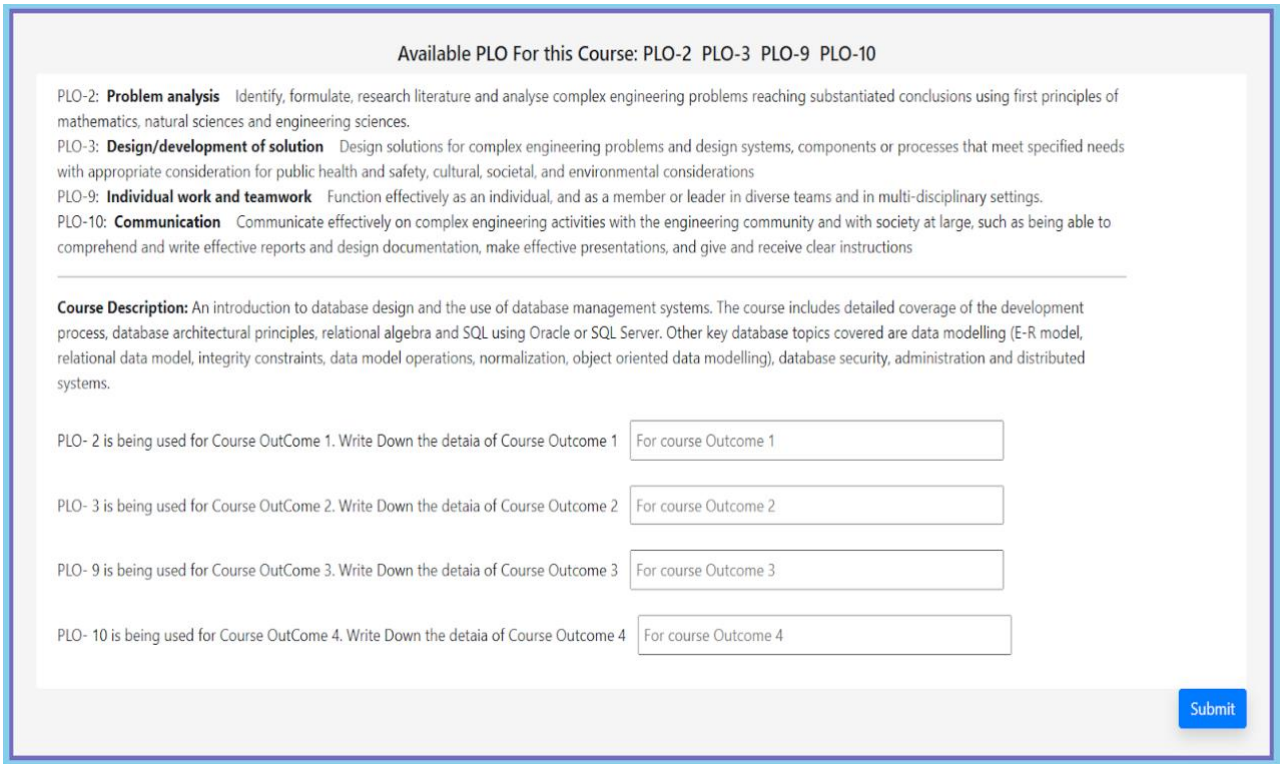
```
            while($travese = mysqli_fetch_array($query)){
                echo "<option value = ".$travese['ploID'] .">".$travese['ploID']. ". ".
$travese['name']."</option>";
            }
        ?>
        </select>        

        <select name="plo2" style="width:17%; padding:5px">
            <option selected disabled>Select PLO </option>

        <?php           $q = "SELECT * FROM plo AS p WHERE p.lavel <= '$label'";
            $query = mysqli_query($con, $q);
            while($travese = mysqli_fetch_array($query)){
                echo "<option value = ".$travese['ploID'] .">".$travese['ploID']. ". ".
$travese['name']."</option>";
            }
        ?>
        </select> <br> <br>
        <select name="plo3" style="width:17%; padding:5px">
            <option selected disabled>Select PLO </option>

        <?php
            $q = "SELECT * FROM plo AS p WHERE p.lavel <= '$label'";
            $query = mysqli_query($con, $q);
            while($travese = mysqli_fetch_array($query)){
                echo "<option value = ".$travese['ploID'] .">".$travese['ploID']. ". ".
$travese['name']."</option>";
            }
        ?>
        </select>        

        <select name="plo4" style="width:17%; padding:5px">
            <option selected disabled>Select PLO </option>

        <?php
            $q = "SELECT * FROM plo AS p WHERE p.lavel <= '$label'";
            $query = mysqli_query($con, $q);
            while($travese = mysqli_fetch_array($query)){
      echo "<option value = ".$travese['ploID'] .">".$travese['ploID']. ". ".
$travese['name']."</option>";
            }
        ?>
        </select> <br> <br>


        <div class="align-center">
```

```
        <input type="submit" id="form1" name="insert" class="btn but1 btn-primary text-white
shadow" value="Submit" onclick="setValue()"/>
        <a href="dashboard_head.php" class="btn but1 btn-primary text-white
shadow">Cancel</a>
      </div>
   </form>
```

For Submit Button:

```
<?php
   include '../../connection.php';

   $c =  $_SESSION['c'];
   $plo1 = $_POST['plo1'];
   $plo2 = $_POST['plo2'];
   $plo3 = $_POST['plo3'];
   $plo4 = $_POST['plo4'];
   $insert1 = "INSERT INTO ploinitialmapping(ploID,courseID) values('$plo1','$c')";
   $insert2 = "INSERT INTO ploinitialmapping(ploID,courseID) values('$plo2','$c')";
   $insert3 = "INSERT INTO ploinitialmapping(ploID,courseID) values('$plo3','$c')";
   $insert4 = "INSERT INTO ploinitialmapping(ploID,courseID) values('$plo4','$c')";
   mysqli_query($con, $insert1);
   mysqli_query($con, $insert2);
   mysqli_query($con, $insert3);
   mysqli_query($con, $insert4);
?>
```

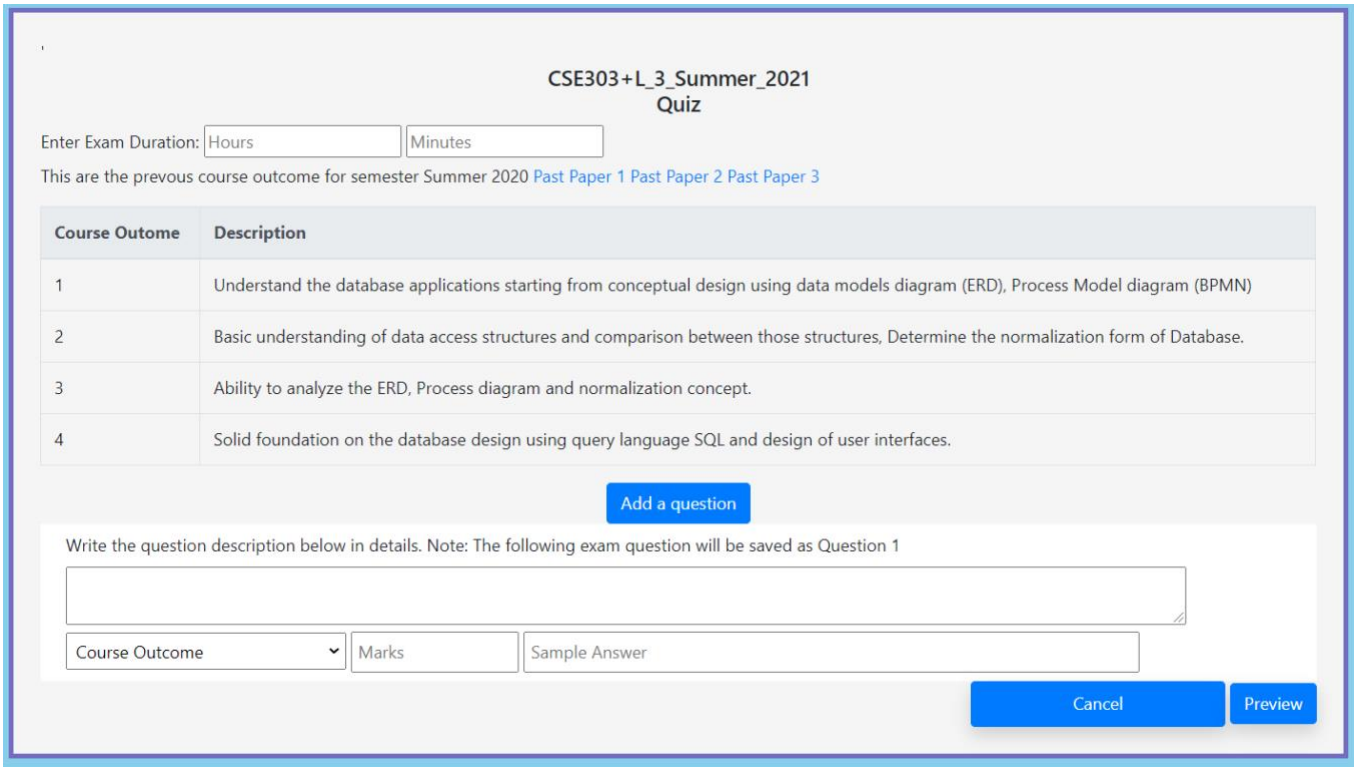
CO PLO Mapping:



```
<?php
                $cou = $travese['courseID'];
                $q = "SELECT * FROM ploinitialmapping WHERE courseID = '$cou'";
                $query = mysqli_query($con, $q);
                $arr = array();

                $i = 0;
                while ($travese = mysqli_fetch_array($query)) {
                    $arr[$i] = $travese['ploID'];
                    echo "PLO-" . $travese['ploID'] . "  ";
                    $i++;
                }
                $arr[4] = $enrollment;
                echo '<input type ="hidden" name="enroll" value='.$enrollment.' >';
                $_SESSION['array'] = $arr;
                ?>
            </h5>

            <div class="input_fields_wrap">
                <?php
                $q1 = "SELECT * FROM course WHERE courseID = '$cou'";
                $query1 = mysqli_query($con, $q1);

                $travese1 = mysqli_fetch_array($query1);
                $description = $travese1['courseDescription'];
                ?>
                <div class="row CLO_Div">
                    <div class="col-md-11">
                        <?php

                        $query = mysqli_query($con, $q);
                        while ($travese = mysqli_fetch_array($query)) {
                            echo "PLO-" . $travese['ploID'] . ":  ";
                            $p = $travese['ploID'];
                            $a = "SELECT * FROM plo WHERE ploID = '$p'";
                            $qqq = mysqli_query($con, $a);
                            $ttt = mysqli_fetch_array($qqq);
                            echo "<b>" . $ttt['name'] . "</b>    " . $ttt['details'];
                            echo "<br>";
                        }
                        ?>
                        <hr>
```

```
<label for="course_description">
   <p><b>Course Description: </b> <?php echo $description; ?> </p>
</label>

<?php
$query = mysqli_query($con, $q);
$travese = mysqli_fetch_array($query);
$plo1 = $travese['ploID'];
$travese = mysqli_fetch_array($query);
$plo2 = $travese['ploID'];
$travese = mysqli_fetch_array($query);
$plo3 = $travese['ploID'];
$travese = mysqli_fetch_array($query);
$plo4 = $travese['ploID'];
```

Make Question Input Form:

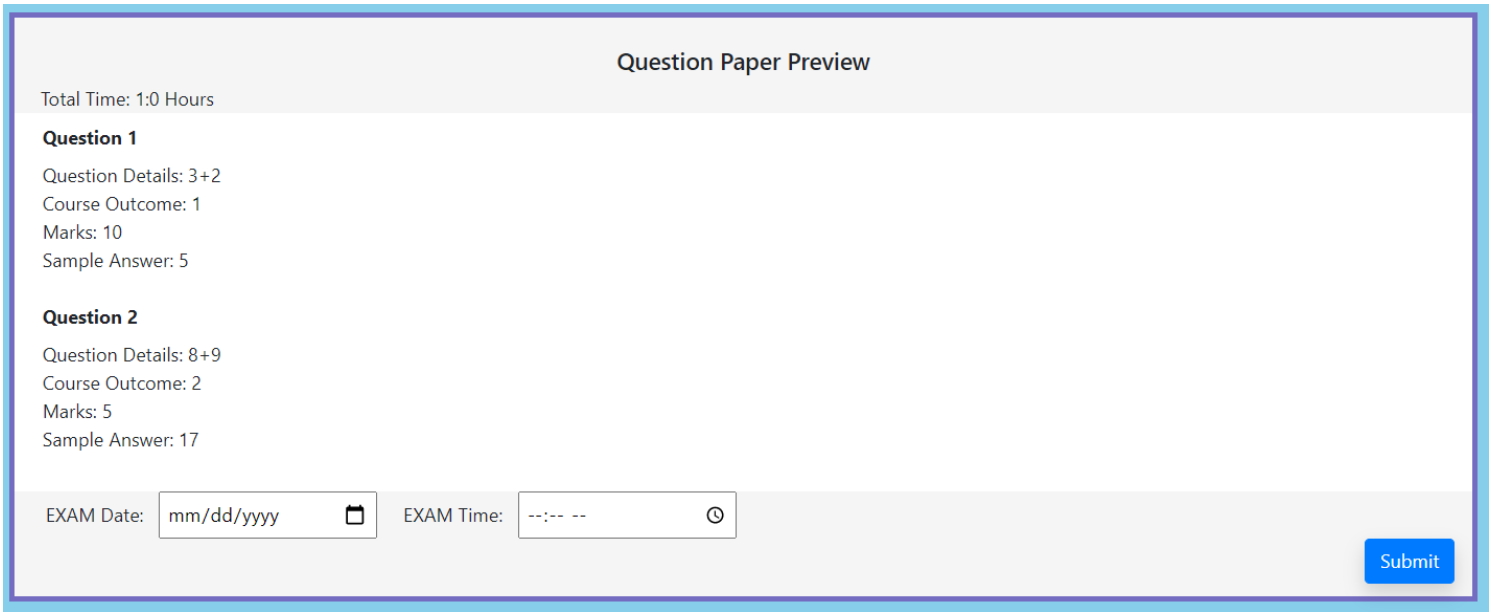


After click Submit Button:

```
<?php
include '../../connection.php';
```

```
$sem = $_SESSION['semester'];
$y = $_SESSION['year'];
$arr =  $_SESSION['array'];
$co1 = $_POST['name1'];
$co2 = $_POST['name2'];
$co3 = $_POST['name3'];
$co4 = $_POST['name4'];
$course = $_POST['enroll'];
echo $course;
$c1 = "1_$course";
$c2 = "2_"."$course";
$c3 = "3_"."$course";
$c4 = "4_"."$course";
$c = substr($course, 3,1);
$insert = "INSERT INTO co (coID, description,courseLevel,enrollmentID) VALUES ('$c1',
'$co1', '$c', '$course')";
mysqli_query($con, $insert);
$insert = "INSERT INTO co (coID, description,courseLevel,enrollmentID) VALUES ('$c2',
'$co2', '$c', '$course')";
mysqli_query($con, $insert);
$insert = "INSERT INTO co (coID, description,courseLevel,enrollmentID) VALUES ('$c3',
'$co3', '$c', '$course')";
mysqli_query($con, $insert);
$insert = "INSERT INTO co (coID, description,courseLevel,enrollmentID) VALUES ('$c4',
'$co4', '$c', '$course')";
mysqli_query($con, $insert);

$insert = "INSERT INTO plocomapping (ploID, coID) VALUES ('$arr[0]', '$c1')";
mysqli_query($con, $insert);
$insert = "INSERT INTO plocomapping (ploID, coID) VALUES ('$arr[1]', '$c2')";
mysqli_query($con, $insert);
$insert = "INSERT INTO plocomapping (ploID, coID) VALUES ('$arr[2]', '$c3')";
mysqli_query($con, $insert);
$insert = "INSERT INTO plocomapping (ploID, coID) VALUES ('$arr[3]', '$c4')";
mysqli_query($con, $insert);

?>
```

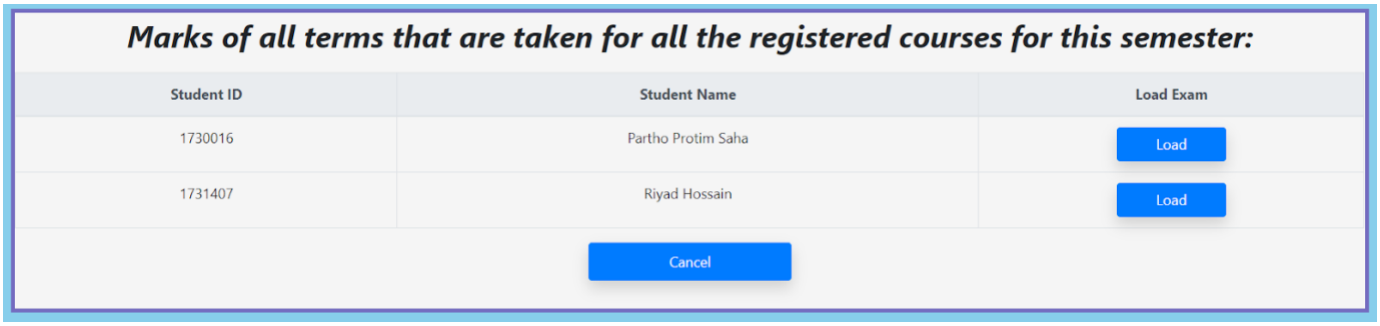
Question Paper Review:




```
<?php
include '../../connection.php';

$id = $_SESSION['id'];

$questionArr = $_POST['questionDescription'];
$marks = $_POST['total'];
$sampleAns = $_POST['sampleans'];
$co = $_POST['co'];
$hours = $_POST['hours'];
$minutes = $_POST['minutes'];
$dueration = $hours.": ".$minutes;
$type = $_POST['type'];
$t1 = substr($type,0,1);
$time = $_POST['time'];
$date = $_POST['date'];
$enrollmentID = $_SESSION['course'];

$a = "SELECT count(assessmentID) AS t FROM assessment WHERE nameOfAss = '$type'
AND enrollmentID ='$enrollmentID' GROUP BY nameOfAss";
$qqq = mysqli_query($con, $a);
$ttt = mysqli_fetch_array($qqq);
$c = isset($ttt['t']) ? $ttt['t'] : 0;
$c = (int)$c + 1;

$assessmentID = $_SESSION['course']."_".$t1.$c ;
$q;
```

```
// echo "<br>The Questions Are: <br><br><br>";
$totalMarks = 0;
if (!empty($questionArr)) {
   for ($y = 0; $y < count($questionArr); $y++) {
      $totalMarks += (int)$marks[$y];
   }

   $insert = "INSERT INTO assessment (assessmentID, totalMarks, enrollmentID,nameOfAss,
time, date, duration)
                  VALUES ('$assessmentID', '$totalMarks', '$enrollmentID', '$type', '$time', '$date',
'$dueration')";
   mysqli_query($con, $insert);
   $y =0 ;
   $q; $qu; $ma; $ans; $cccccc;
   while($y < count($questionArr)) {
      echo "Question Details: ";
      echo $questionArr[$y];
      $qu =  $questionArr[$y];
      echo "<br>Course Outcome: " . $co[$y];
      $cccccc = $co[$y]."_".$enrollmentID;;
      echo "<br>";
      echo "Marks: " . $marks[$y];
      $ma = $marks[$y];
      echo "<br>";
      echo "Sample Answer: " . $sampleAns[$y];
      $ans =  $sampleAns[$y];
      echo "<br> <br>";
      $q = $assessmentID."_".($y+1);
      $totalMarks += (int)$marks[$y];

      // $iii = "INSERT INTO question(quesID, quesDetails, marks, sampleAns, facultyID, coID,
assessmentID)
      //           VALUES ('$q', '$qu', '$ma', '$ans', '$id', '$cccccc', '$assessmentID')";
      // mysqli_query($con, $iii);
      $insert = "INSERT INTO question (quesID, details, mark,sampleAns, facultyID, coID,
assessmentID)
                  VALUES ('$q', '$qu', '$ma', '$ans', '$id', '$cccccc', '$assessmentID')";
      mysqli_query($con, $insert);
      $y++;
   }
```

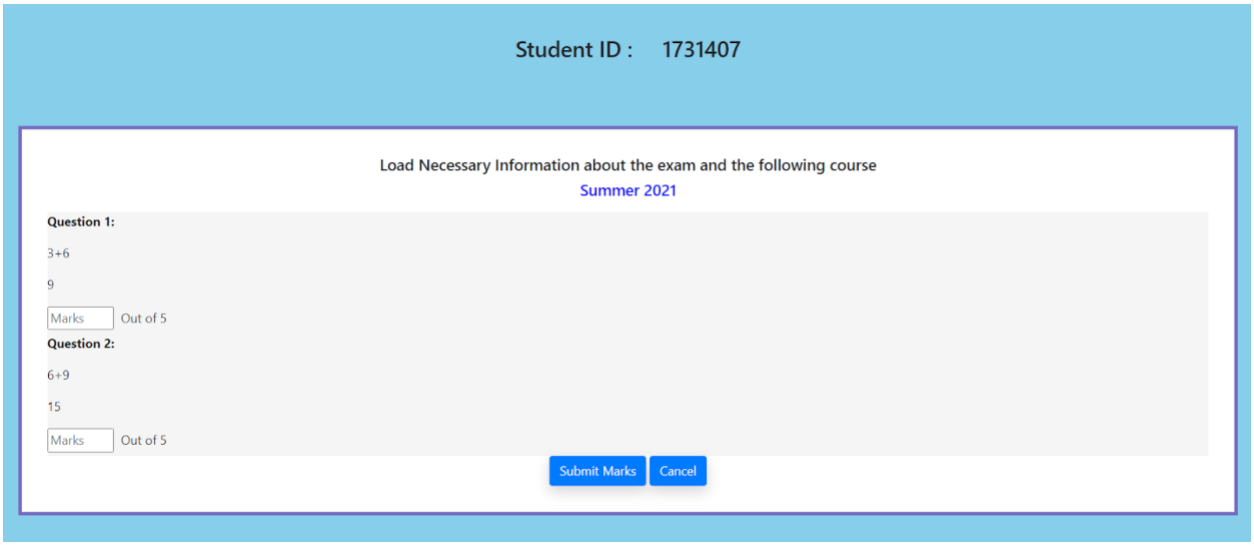
For Student Evaluation: Based on Student's performance, this dashboard will be loaded.




```
<?php
        $value = "-9999";
        $q = "SELECT DISTINCT s.fname AS fname , s.lname AS lname, s.studentID AS id
        FROM student AS s, assessment AS a, question AS q, evaluation AS e
        WHERE e.quesID=q.quesID AND q.assessmentID = a.assessmentID AND
a.enrollmentID = '$c' AND a.nameOfAss = '$examID' AND e.studentID = s.studentID AND
e.evaluationID NOT IN (SELECT evaluationID FROM evaluation_faculty)";


        $qu = mysqli_query($con, $q);
        $i = 0;
        while ($t = mysqli_fetch_assoc($qu)) {
echo "<tr>
            <td>" . $t['id'] . "</td>
            <td>" . $t['fname'] . " " . $t['lname'] . "</td>
            <td>
              <form action=\"mark_Exam3.php\" method=\"POST\">
              <input type=\"hidden\" name = \"student\" value = " . $t['id'] . ">
              <input type=\"hidden\" name = \"examID\" value = " . $examID . ">
              <a href = \"#\" onclick=\"document.forms[" . $i . "].submit();return false;\" style =
\"text-decoration: none; width: 30%;\"
              class=\"btn btn-primary text-white shadow\">Load</a>
              </form>
            </td>

        </tr>";
            $i++;
        }
        ?>
```

```
</tbody>        </table>
```

After clicking any load Button, the student's answer with his or her id will be loaded and faculty members can check those and marks then based on the given marks for that question.



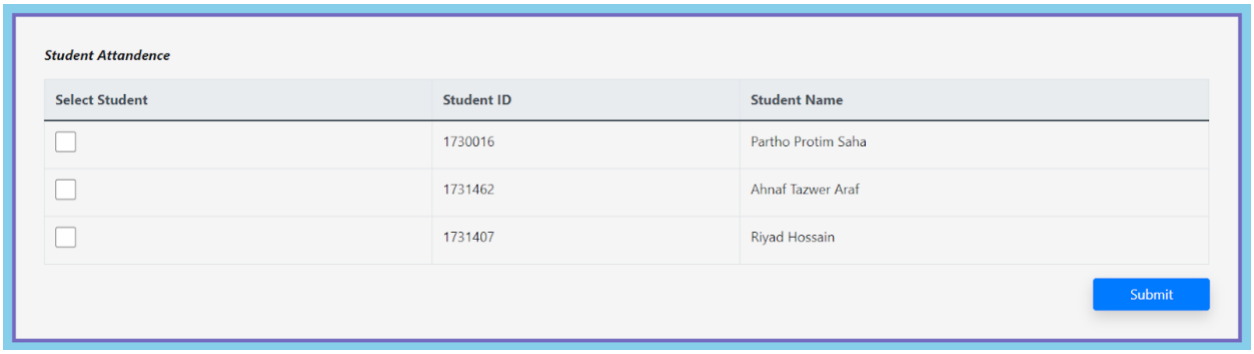

```
<form action="updateMarking.php" class="tableStyle" method="POST">
        <h5 style="text-align: center;">Load Necessary Information about the exam and the following course</h5>
        <h5 style="text-align: center; color: blue;">Summer 2021</h5>
        <p></p>
        <input type="hidden" name="examID" value="<?php echo $examID;?>">

        <?php

        $q = "SELECT DISTINCT e.evaluationID as eva, q.quesID AS ques, q.details AS question, e.studentAns as answer, q.mark as mark FROM student AS s,
            assessment AS a, question AS q, evaluation AS e WHERE e.quesID=q.quesID AND q.assessmentID = a.assessmentID
            AND  e.obatinMarks = '-9999' AND a.enrollmentID = '$c' AND a.nameOfAss = '$examID' AND e.studentID = s.studentID AND e.studentID = '$studentID'";
        $query = mysqli_query($con, $q);
        $i = 0;
        while($t = mysqli_fetch_assoc($query)){
            echo "<div style = \"background-color: whitesmoke;\"><p><b>Question " . ($i + 1) . ":
</b></p>
            <input type=\"hidden\" name = \"student\" value = " . $studentID . ">
            <input type=\"hidden\" name = \"ques[]\" value = " . $t['ques'] . ">
            <input type=\"hidden\" name = \"eva[]\" value = " . $t['eva'] . ">
```

```
        <p>".$t['question']."</p>
        <p>".$t['answer']."</p>
        <input type=\"number\" step=\"0.01\" min=\"0\" max=".$t['mark']." name = \"marks[]\"
placeholder=\"Marks\" required><label>   Out of ".$t['mark']."</label></div>";
        $i++;       } ?>
```

After press submit This SQL will be execute:

```
<?php

include '../../connection.php';
$sem = $_SESSION['semester'];
$y = $_SESSION['year'];
$id = $_SESSION['id']; // faculty id
$arr =  $_SESSION['array'];
if (isset($_POST['course'])) {
   $c = $_POST['course'];
} else {
   if (isset($arr[4])) {
      $c = $arr[4];
   } else {
      $c = $_SESSION['course'];
   }
}
// there is course id in $c
$_SESSION['course'] = $c;

$studentID = isset($_POST['student']) ? $_POST['student'] : null;
$ques = isset($_POST['ques']) ? $_POST['ques'] : null;
$marks = isset($_POST['marks']) ? $_POST['marks'] : null;
$eva = isset($_POST['eva']) ? $_POST['eva'] : null;
$y =0;
while($y<count($ques)){
   echo $id." ".$studentID." ".$ques[$y]." ".$marks[$y]." ".$eva[$y]."<br>";

   $q = "UPDATE evaluation
        SET obatinMarks = '$marks[$y]'
        WHERE evaluationID = '$eva[$y]'";
   mysqli_query($con, $q);
   $q = "INSERT INTO evaluation_faculty (evaluationID, facultyID) VALUES ('$eva[$y]',
'$id')";
   mysqli_query($con, $q);
   $y++;
}
?>
```

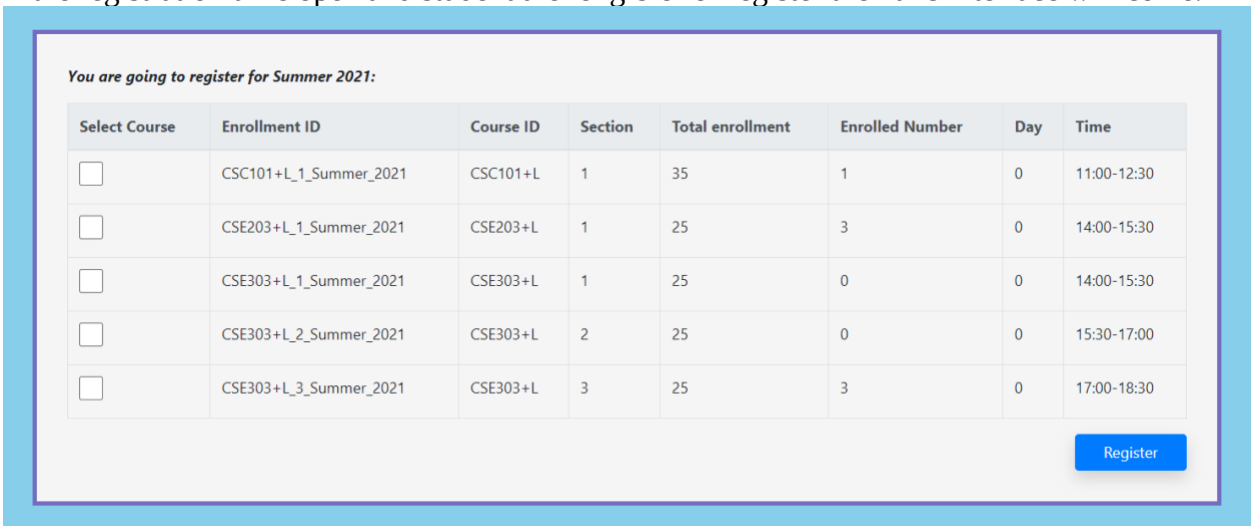
Student Attendance interface:



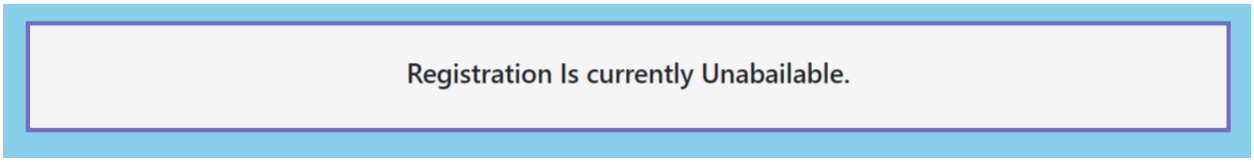

```
<tbody>
                <?php
                $q = "SELECT * FROM student AS s, registration AS r WHERE r.enrollmentID
= '$c' AND r.studentID = s.studentID";
                $query = mysqli_query($con, $q);
                $i = 0;
                while ($t = $query->fetch_assoc()) {
                   echo '<td><input type="checkbox" name = "studentAttandence[]" value = ' . $i
. ' style = "width: 3vh; height: 3vh;"></td>
                <input type="hidden" name = "enrollmentID[]" value=' . $t['studentID'] . '>
                <td>' . $t['studentID'] . '</td>
                <td>' . $t['fname']." ".$t['lname']. '</td>
                </tr>';
                }
                ?>
              </tbody>
```

This interface will for Student registration:

If the registration time open and student are eligible for register then this interface will come.



Otherwise this interface will show.



All those execution, this SQL and html part is needed.

```
<?php
        $q = "SELECT status FROM activeness WHERE name = 'registration'";
        $query = mysqli_query($con, $q);
        $t = $query->fetch_assoc();
        $abc = $sem . "_" . $y;
        $qq = "SELECT * FROM registration WHERE enrollmentID LIKE '%$abc%' AND
studentID = '$id'";
        $query1 = mysqli_query($con, $qq);
        $t1 = $query1->fetch_assoc();

        if ($t['status'] == 'true' && $t1 == null) {
        ?>
           <form action="courseRegistration.php" method="POST">
              <p><strong><i>You are going to register for <?php echo $sem . " " . $y;
?>:</i></strong></p>
              <table class="table table-bordered">
                 <thead class="thead-light">
                    <tr>
                       <th>Select Course</th>
                       <th>Enrollment ID</th>
                       <th>Course ID</th>
                       <th>Section</th>
                       <th>Total enrollment</th>
                       <th>Enrolled Number</th>
                       <th>Day</th>
                       <th>Time</th>
                    </tr>
                 </thead>
                 <tbody>
                    <?php
                    $q = "SELECT * FROM section WHERE semester = '$sem' AND year = '$y'";
                    $query = mysqli_query($con, $q);
                    $i = 0;
                    while ($t = $query->fetch_assoc()) {
                       echo '<td><input type="checkbox" name = "registeredCourses[]" value = ' .
$t['enrollmentID'] . ' style = "width: 3vh; height: 3vh;"></td>
                 <input type="hidden" name = "enrollmentID[]" value=' . $t['enrollmentID'] . '>
                 <td>' . $t['enrollmentID'] . '</td>
```

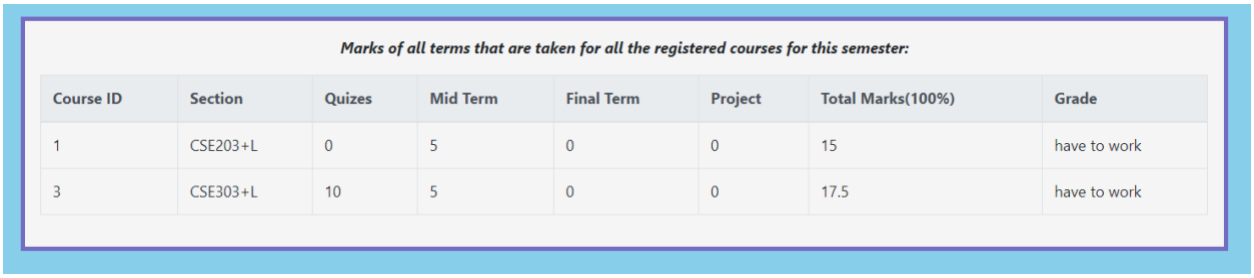
```
                    <td>' . $t['courseID'] . '</td>
                    <td>' . $t['sectionID'] . '</td>
                    <td>' . $t['totalEnrollment'] . '</td>
                    <td>' . $t['totalNumberOfStudent'] . '</td>
                    <td>' . $t['day'] . '</td>
                    <td>' . $t['classTime'] . '</td>
                    </tr>';
                        $i++;
                    }
                    echo '<input type="hidden" name="e" value = ' . $i . '>';
                    ?>
                </tbody>
            </table>
            <div align="right">
                <input type="submit" style="width: 10%;" name="insert" class="btn btn-primary
text-white shadow" value="Register" />
            </div>
        </form>
    <?php
```

After clicking submit Button this sql part will be execute :

```
<?php
include '../../connection.php';
$sem = $_SESSION['semester'];
$y = $_SESSION['year'];
$id = $_SESSION['id'];
$e = $_POST['e'];
$registeredCourses = $_POST['registeredCourses'];
$enrollmentID = $_POST['enrollmentID'];
for ($y = 0; $y < $e; $y++) {
   for ($y1 = 0; $y1 < count($registeredCourses); $y1++) {
      if ($registeredCourses[$y1] == $enrollmentID[$y]) {

         $insert = "INSERT INTO registration(studentID, enrollmentID) VALUES ('$id',
'$enrollmentID[$y]')";
         mysqli_query($con, $insert);
         $insert = "UPDATE section SET totalNumberOfStudent = totalNumberOfStudent+1
WHERE enrollmentID='$enrollmentID[$y]'";
         mysqli_query($con,$insert);
         break;
      }
   }
}
```

?>

For the student dashboard, student courses and all marks of those courses will be shown. All this information will be stored in the database.



Marks of all terms that are taken for all the registered courses for this semester:

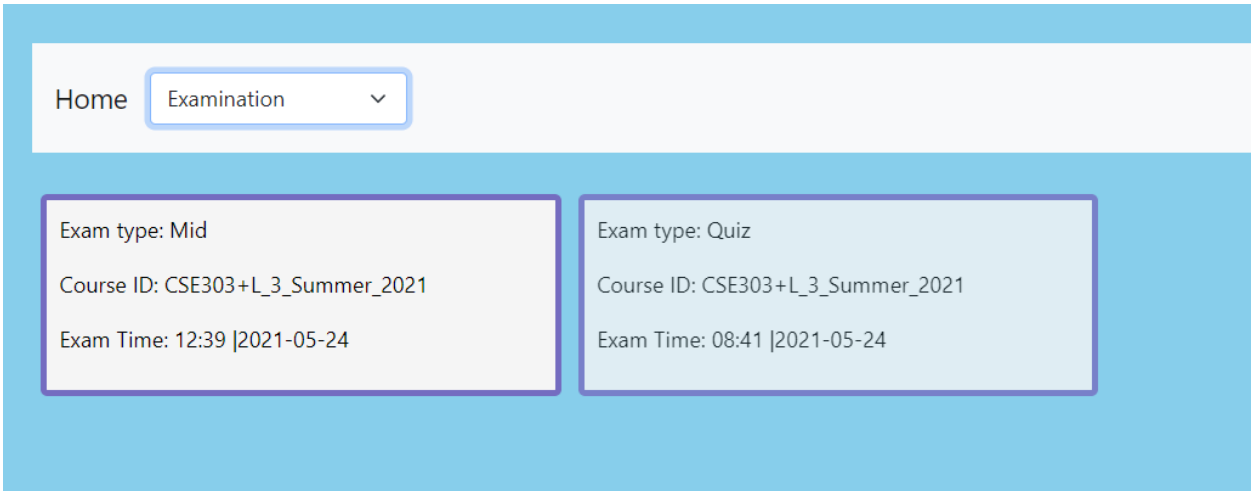
Course ID	Section	Quizes	Mid Term	Final Term	Project	Total Marks(100%)	Grade
1	CSE203+L	0	5	0	0	15	have to work
3	CSE303+L	10	5	0	0	17.5	have to work

```
<?php
            $q = "SELECT s.sectionID AS sec, s.courseID AS cou, s.enrollmentID as enr
FROM registration AS r, section AS s
                WHERE r.studentID = '$id' AND r.enrollmentID = s.enrollmentID";
            $query = mysqli_query($con, $q);
            while ($t = $query->fetch_assoc()) {
               $sec = $t['sec'];
               $course = $t['cou'];
               $enr = $t['enr'];
               $mid;
               $project;
               $projectt;
               $quiz;
               $final;
               $midt;
               $quizt;
               $finalt;
            ?>
               <tr>
                  <td><?php echo $sec; ?></td>
                  <td><?php echo $course; ?></td>
                  <td>
                     <?php
                     $qqq = "SELECT SUM(obatinMarks) as mark, SUM(mark) as m FROM
evaluation AS e, question AS q, assessment AS a
      WHERE e.evaluationID LIKE '$id%'AND a.enrollmentID = '$enr' AND e.quesID =
q.quesID AND q.assessmentID = a.assessmentID AND a.nameOfAss = 'Quiz'  AND
e.obatinMarks != '-9999' GROUP BY q.assessmentID";
                     $qqq = mysqli_query($con, $qqq);
```

```
                    $qqq = $qqq->fetch_assoc();
                    $quiz = isset($qqq['mark']) ? $qqq['mark'] : 0;
                    $quizt = isset($qqq['m']) ? $qqq['m'] : 0;
                    echo $quiz;
                    ?>
                </td>
                <td><?php
                    $qqq = "SELECT SUM(obatinMarks) as mark , SUM(mark) as m FROM
evaluation AS e, question AS q, assessment AS a
                        WHERE e.evaluationID LIKE '$id%' AND e.quesID = q.quesID AND
q.assessmentID = a.assessmentID AND a.nameOfAss = 'Mid'  AND e.obatinMarks != '-9999'
GROUP BY q.assessmentID";
                    $qqq = mysqli_query($con, $qqq);
                    $qqq = $qqq->fetch_assoc();
                $mid = isset($qqq['mark']) ? $qqq['mark'] : 0;
                    $midt = isset($qqq['m']) ? $qqq['m'] : 0;
                    echo $mid;
                    ?></td>
                <td><?php
                    $qqq = "SELECT SUM(obatinMarks) as mark , SUM(mark) as m FROM
evaluation AS e, question AS q, assessment AS a
                        WHERE e.evaluationID LIKE '$id%' AND e.quesID = q.quesID AND
q.assessmentID = a.assessmentID AND a.nameOfAss = 'Final'  AND e.obatinMarks != '-9999'
GROUP BY q.assessmentID";
                    $qqq = mysqli_query($con, $qqq);
                    $qqq = $qqq->fetch_assoc();
              $final = isset($qqq['mark']) ? $qqq['mark'] : 0;
                $finalt = isset($qqq['m']) ? $qqq['m'] : 0;
                    echo $final;
                    ?></td>
                <td>
                    <?php $qqq = "SELECT SUM(obatinMarks) as mark , SUM(mark) as m
FROM evaluation AS e, question AS q, assessment AS a
                        WHERE e.evaluationID LIKE '$id%' AND e.quesID = q.quesID AND
q.assessmentID = a.assessmentID AND a.nameOfAss = 'Project'  AND e.obatinMarks != '-9999'
GROUP BY q.assessmentID";
                    $qqq = mysqli_query($con, $qqq);
                    $qqq = $qqq->fetch_assoc();
                    $project = isset($qqq['mark']) ? $qqq['mark'] : 0;
                    $projectt = isset($qqq['m']) ? $qqq['m'] : 0;
                    echo $project;

                    ?></td>
                <td>
                <?php
```

```
                $t = (($mid * 30 + $quiz * 20 + $final * 30  + $project * 20) / ($quizt + $midt + $finalt + $projectt));
                    echo $t;
                    $quee = "UPDATE registration
                    SET totalMakrs = '$t'
                    WHERE studentID = '$id' AND enrollmentID = '$enr'";
                    mysqli_query($con,$quee);
                    ?></td>
```

The Student Exam will be shown in the part of examination which is not submitted by student:

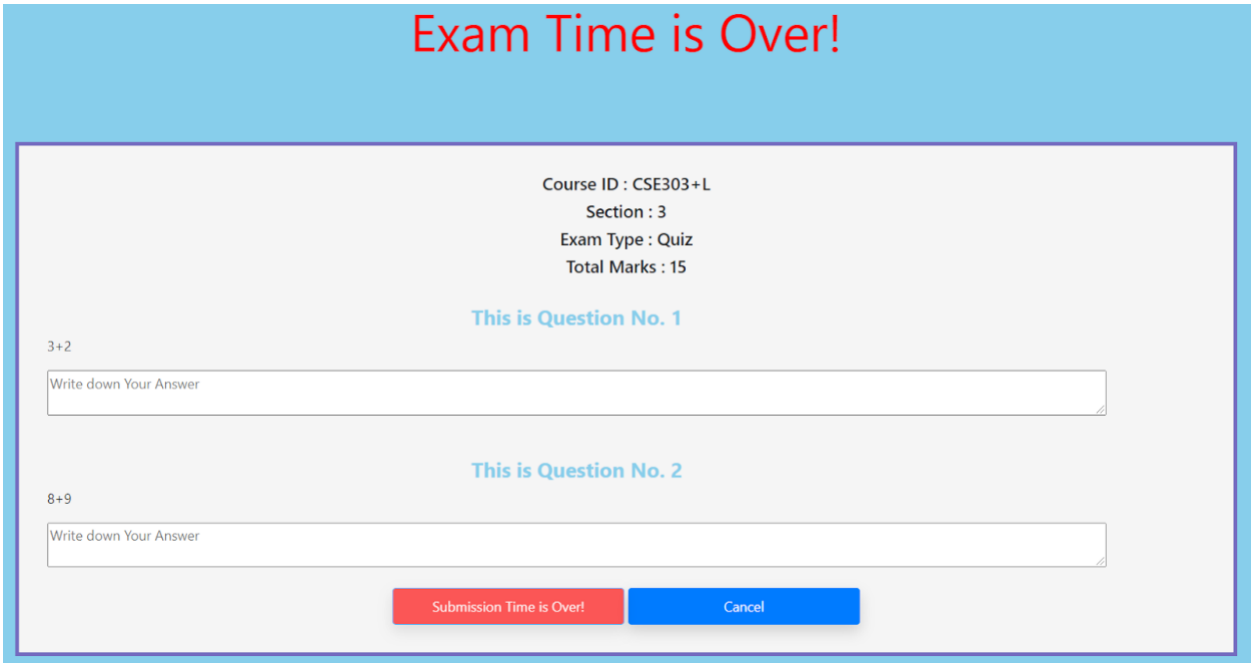



```
$q = "SELECT * FROM assessment AS a, registration AS r WHERE r.studentID = '$id' AND a.enrollmentID = r.enrollmentID";
            // }
            $query = mysqli_query($con, $q);
            $i = 0;
            while ($t = $query->fetch_assoc()) {
                $ass = $t['assessmentID'];
                $q1 = "SELECT * FROM assessment AS a, registration AS r, evaluation AS e, question AS q WHERE a.assessmentID = '$ass'  AND q.assessmentID = a.assessmentID AND e.quesID = q.quesID AND e.studentID = '$id'";
                $query1 = mysqli_query($con, $q1);
                if ($query1->fetch_assoc() == null) {
                    echo "<form  method=\"POST\" action=\"giveExam.php\">
                <input type=\"hidden\" name=\"assessmentID\" value = " . $t['assessmentID'] . ">
                <a href = \"#\" onclick=\"document.forms[" . $i . "].submit();\" style = \"text-decoration: none;\">
                <div class=\"loadExam\">
                    <p>Exam type: " . $t['nameOfAss'] . "</p>
```

```
            <p>Course ID: " . $t['enrollmentID'] . "</p>
            <p>Exam Time: " . $t['time'] . " |" . $t['date'] . "</p>

            </div>
            </a>
            </form>";
            $i++;
        }            }

?>
```

Click any Examination: all information will be shown related with that particular exam:

If the time passes then the student cannot submit that exam paper



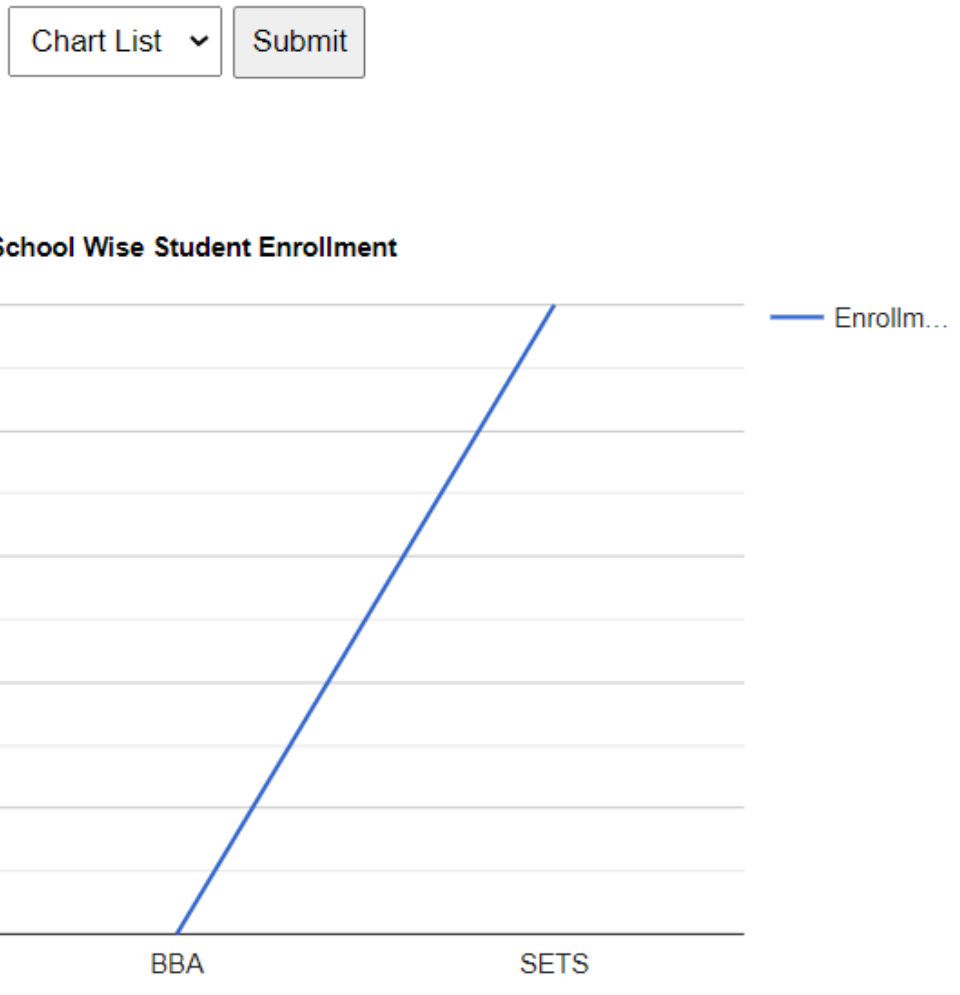
After clicking submit button:

```
<?php
include '../../connection.php';
$sem = $_SESSION['semester']; // current semester
$y = $_SESSION['year']; // current year
$id = $_SESSION['id']; //student id



$file = isset($_POST['myfile']) ? $_POST['myfile'] : null;
$quesID = isset($_POST['quesID']) ? $_POST['quesID'] : null;
$enrollmentID = isset($_POST['enrollmentID']) ? $_POST['enrollmentID'] : null;
$evaluationID;
for ($i = 0; $i < count($file); $i++) {
    echo $i + 1 . " File Name<b>::</b> " . $file[$i] . " "; // this is per question answer
    echo "Question ID " . $quesID[$i] . " "; // this is question id
    $evaluationID = $id . "_" . $quesID[$i];  // evaluation id set up
    echo $evaluationID . "<br>";
```

```
   $q= "INSERT INTO evaluation (evaluationID,
studentAns,enrollmentID,quesID,studentID,obatinMarks)
      VALUES ('$evaluationID', '$file[$i]', '$enrollmentID', '$quesID[$i]','$id', '-9999')";
   mysqli_query($con,$q);
} ?>
```

OUTPUT FORM

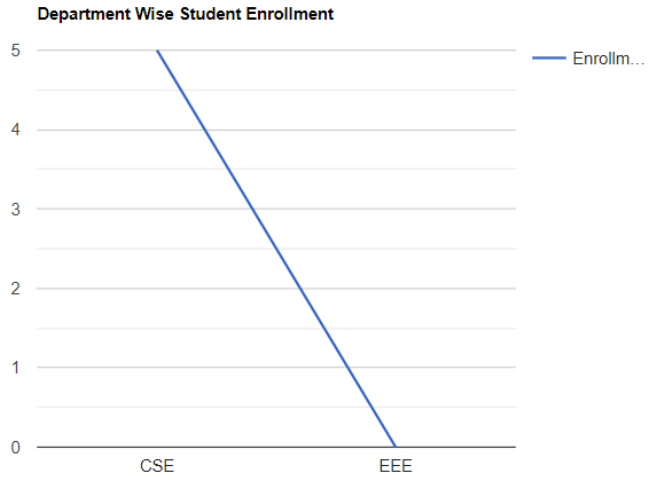
School wise Student Enrollment:




```
<?php
```

```
    $ycpArr = array();
    $ycpArr1 = array();
    $i=0;
    $q = "SELECT sc.schoolID AS School , COUNT(s.studentID) AS total
          FROM school AS sc
          LEFT JOIN department AS d
          ON sc.schoolID = d.schoolID
          LEFT JOIN  program AS p
          ON d.deptShortName = p.deptShortName
          LEFT JOIN student AS s
          ON s.programID = p.programID
          GROUP BY sc.schoolID";

    $query = mysqli_query($con, $q);
    while($travese = $query->fetch_assoc()){
       $ycpArr[$i][0]=$travese['School'];
       $ycpArr[$i][1]=$travese['total'];
       $i++;
    }

?>
```

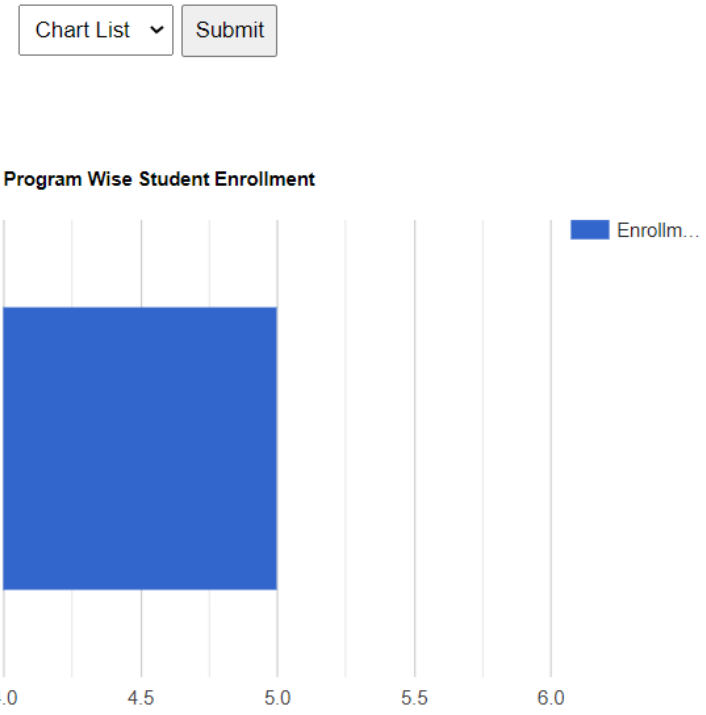
Department wise student enrollment:




```
<?php
    $ycpArr = array();
    $ycpArr1 = array();
    $i=0;
```

```
    $q = "SELECT d.deptShortName AS School , COUNT(s.studentID) AS total
          FROM  department AS d
          LEFT JOIN  program AS p
          ON d.deptShortName = p.deptShortName
          LEFT JOIN student AS s
          ON s.programID = p.programID
          GROUP BY d.deptShortName";

    $query = mysqli_query($con, $q);
    while($travese = $query->fetch_assoc()){
       $ycpArr[$i][0]=$travese['School'];
       $ycpArr[$i][1]=$travese['total'];
       $i++;
    }

?>
```

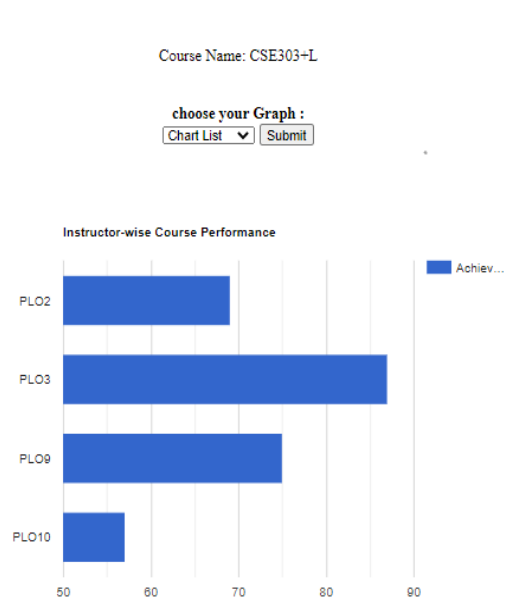
Program wise Student Enrollment:




```
<?php
    $ycpArr = array();
    $ycpArr1 = array();
```

```
    $i=0;
    $q = "SELECT programID, COUNT(s.studentID) AS total FROM student AS s GROUP BY
programID";
    $query = mysqli_query($con, $q);
    while($travese = $query->fetch_assoc()){
        $ycpArr[$i][0]=$travese['programID'];
        $ycpArr[$i][1]=$travese['total'];
        $i++;
    }

?>
```

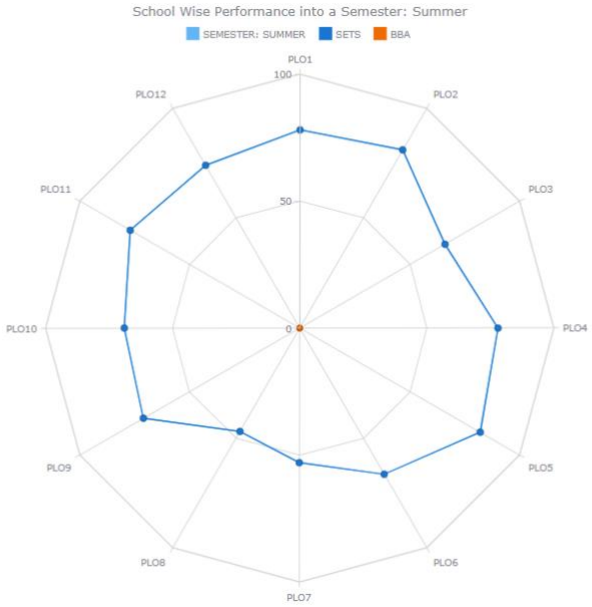
Instructor-wise Course Performance:




```
<?php
    include 'connection.php';
    $ycpArr = Array();
    $structure = isset($_POST['select']) ? $_POST['select'] : null;
    for($i=0;$i<=4;$i++){
        $select = "SELECT SUM(obatinMarks)*100 / SUM(q.mark) AS plo
                FROM evaluation AS e, question AS q,student AS s
```

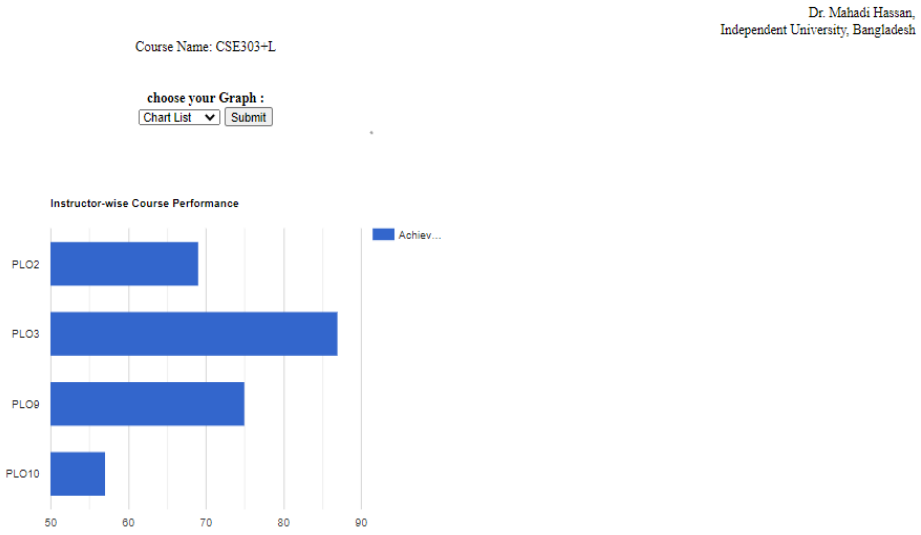
```
            WHERE e.studenID=s.studentID AND e.quesID = q.quesID AND q.coID LIKE
'{$i}%'
                    AND q.assessmentID LIKE '%CSE203+L_1_Summer_2021%'";
      $result = mysqli_query($con,$select);
      while($travese = $result->fetch_assoc()){
        // array_push($ycpArr, $travese['total']);
        // array_push($ycpArr1, $travese['programID']);
        $ycpArr[$i][0]="co".$i;
        $ycpArr[$i][1]=$travese['plo'];
        $i++;

     }
   }
   ?>
```

School Wise Performance



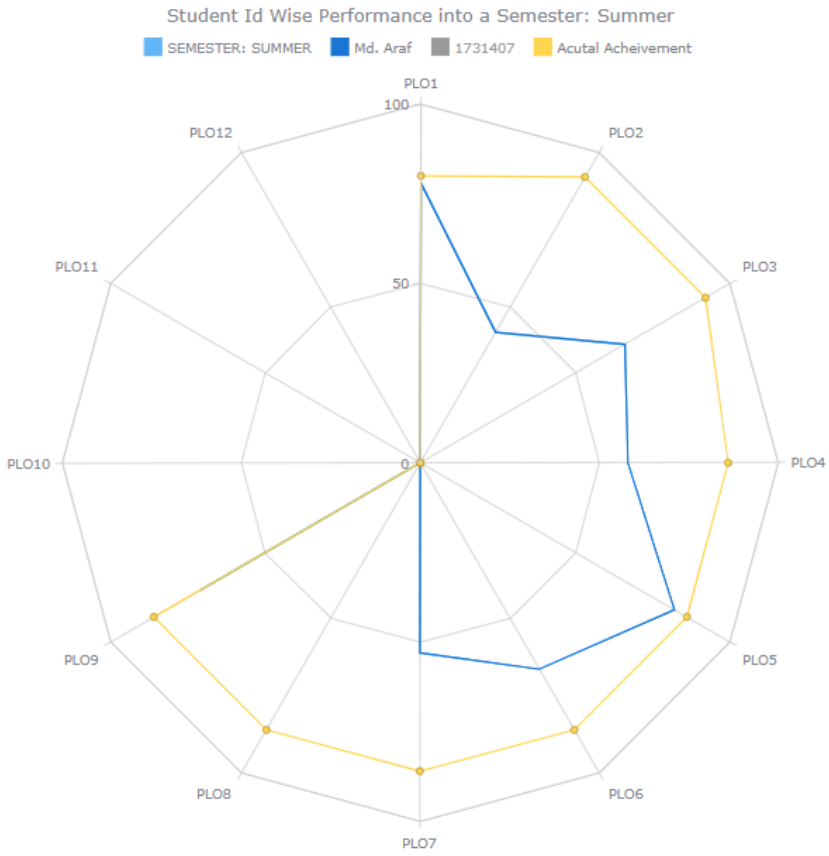

```
$ycpArr = Array();
   $structure = isset($_POST['select']) ? $_POST['select'] : null;
   for($i=0;$i<=4;$i++){
      // $select = "SELECT SUM(obatinMarks)*100 / SUM(q.mark) AS plo
      //          FROM evaluation AS e, question AS q,student AS s
      //          WHERE e.studenID=s.studentID AND e.quesID = q.quesID AND q.coID LIKE
'{$i}%'
      //               AND q.assessmentID LIKE '%CSE203+L_1_Summer_2021%' ";
      $select = "SELECT SUM(obatinMarks)*100 / SUM(q.mark) AS plo, sc.schoolID
               FROM school AS sc, department AS d, program AS p, student AS s, evaluation AS
e, question AS q
               WHERE sc.schoolID=d.schoolID AND d.deptShortName=p.deptShortName
               AND p.programID=s.programID AND e.studentID=s.studentID AND e.quesID =
q.quesID AND q.coID LIKE '{$i}%'
               GROUP BY sc.schoolID";
      $result = mysqli_query($con,$select);
      while($travese = $result->fetch_assoc()){
        // array_push($ycpArr, $travese['total']);
        // array_push($ycpArr1, $travese['programID']);
        $ycpArr[$i][0]="co".$i;
        $ycpArr[$i][1]=$travese['plo'];
        $i++;

     }
   }
```

Course Wise Performance:




```
<?php
   include 'connection.php';
   $ycpArr = Array();
   $structure = isset($_POST['select']) ? $_POST['select'] : null;
   for($i=0;$i<=4;$i++){
      $select = "SELECT SUM(obatinMarks)*100 / SUM(q.mark) AS plo
               FROM evaluation AS e, question AS q,student AS s
               WHERE e.studenID=s.studentID AND e.quesID = q.quesID AND q.coID LIKE
'{$i}%'
                     AND q.assessmentID LIKE '%CSE203+L_1_Summer_2021%'";
      $result = mysqli_query($con,$select);
      while($travese = $result->fetch_assoc()){
        // array_push($ycpArr, $travese['total']);
        // array_push($ycpArr1, $travese['programID']);
        $ycpArr[$i][0]="co".$i;
        $ycpArr[$i][1]=$travese['plo'];
        $i++;

     }
   }
   ?>
```

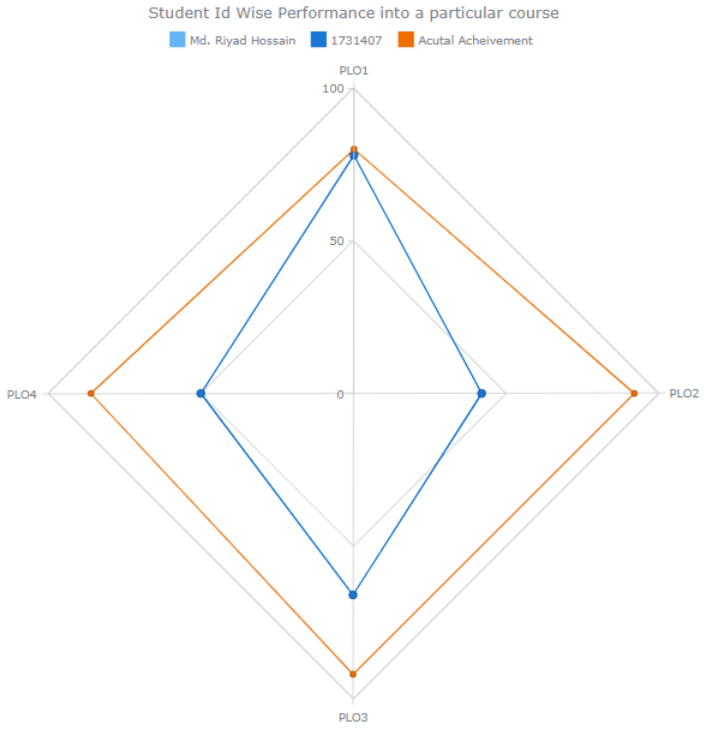
Student wise Performance (Seen by student):




```
<!-- <?php
   include 'connection.php';
   $ycpArr = Array();
   $structure = isset($_POST['select']) ? $_POST['select'] : null;
   for($i=0;$i<=4;$i++){
      // $select = "SELECT SUM(obatinMarks)*100 / SUM(q.mark) AS plo
      //          FROM evaluation AS e, question AS q,student AS s
      //          WHERE e.studenID=s.studentID AND e.quesID = q.quesID AND q.coID LIKE
'{$i}%'
      //                AND q.assessmentID LIKE '%CSE203+L_1_Summer_2021%' AND
e.enrollmentID=s.enrollment And s.semester='summer'" ;

      $result = mysqli_query($con,$select);
      while($travese = $result->fetch_assoc()){
        // array_push($ycpArr, $travese['total']);
        // array_push($ycpArr1, $travese['programID']);
        $ycpArr[$i][0]="co".$i;
        $ycpArr[$i][1]=$travese['plo'];
        $i++;
```

```
  }
}
?>
```

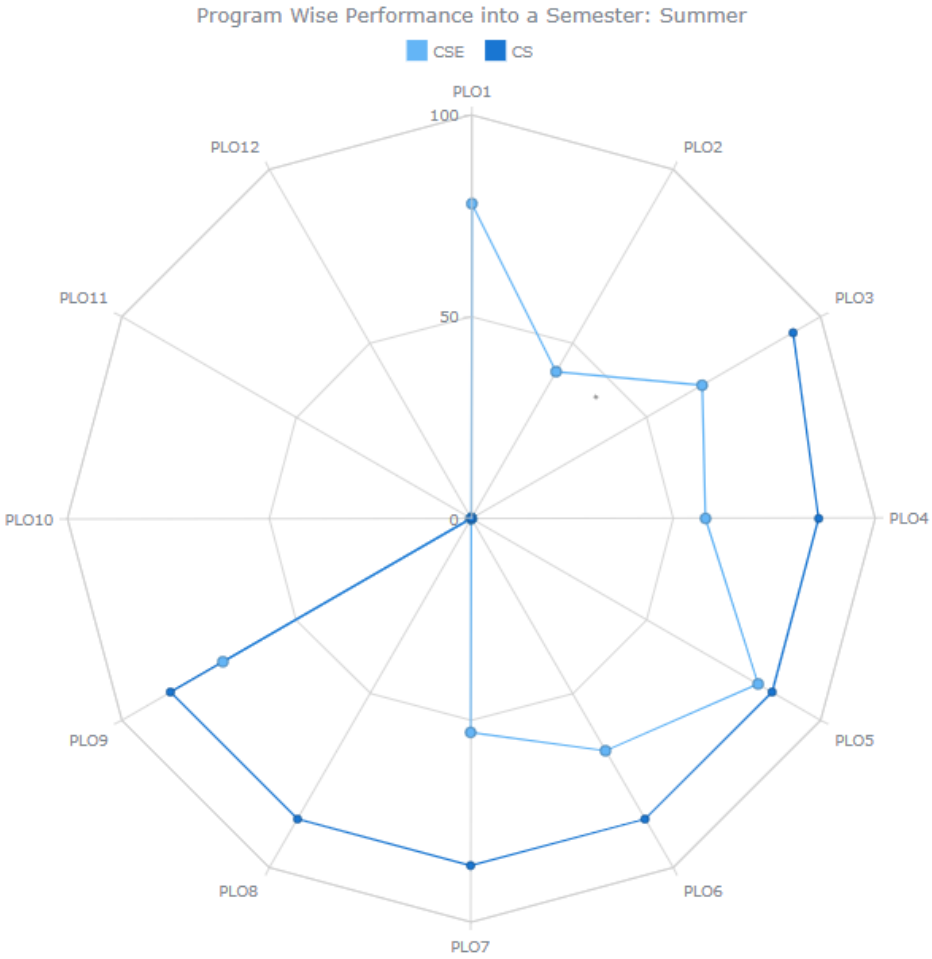
Student id Wise performance into his/her specific course(Seen by Faculty):




```
<?php echo json_encode($ycpArr); ?>

   <script type="text/javascript">
   //var obj = <?php echo json_encode($ycpArr); ?>;
   <!-- <?php
   include 'connection.php';
   $ycpArr = Array();
   $structure = isset($_POST['select']) ? $_POST['select'] : null;
   for($i=0;$i<=4;$i++){
      $select = "SELECT SUM(obatinMarks)*100 / SUM(q.mark) AS plo
            FROM evaluation AS e, question AS q,student AS s
            WHERE e.studenID=s.studentID AND e.quesID = q.quesID AND q.coID LIKE
'{$i}%'
                  AND q.assessmentID LIKE '%CSE203+L_1_Summer_2021%'" ;

      $result = mysqli_query($con,$select);
```

```
    while($travese = $result->fetch_assoc()){
      // array_push($ycpArr, $travese['total']);
      // array_push($ycpArr1, $travese['programID']);
      $ycpArr[$i][0]="co".$i;
      $ycpArr[$i][1]=$travese['plo'];
      $i++;



  }
}
?>
```

Program wise performance:




```
<?php
   include 'connection.php';
   $ycpArr = Array();
   $structure = isset($_POST['select']) ? $_POST['select'] : null;
   for($i=0;$i<=4;$i++){
      $select = "SELECT s.programID AS prog,  SUM(obatinMarks)*100 / SUM(q.mark) AS plo
                FROM evaluation AS e, question AS q,student AS s
               WHERE e.studenID=s.studentID AND e.quesID = q.quesID AND q.coID LIKE
'{$i}%'
                  AND q.assessmentID LIKE '%CSE203+L_1_Summer_2021%'
ANDp.e.enrollmentID=s.enrollment And s.semester='summer'  GROUP BY s.programID" ;
      $result = mysqli_query($con,$select);
      while($travese = $result->fetch_assoc()){
```

```
    // array_push($ycpArr, $travese['total']);
    // array_push($ycpArr1, $travese['programID']);
$pro = $travese[‘prog’];
    $ycpArr[$i][0]="co".$i;
    $ycpArr[$i][1]=$travese['plo'];
    $i++;



  }
}
?>
```

CHAPTER 5

CONCLUSION

›› PROBLEM AND SOLUTION

›› ADDITIONAL FEATURE AND FUTURE DEVELOPMENT

SECTION 5: CONCLUSION

PROBLEM AND SOLUTION

We tried our best to implement the best possible software in the bounded timeframe of the semester and limited amount of marksheet and info provided on students and faculty members. However, our system lacks an automated marking feature and multiple answering. It is not added in our system.

If provided with more resources and data to work with, we believe we could have achieved much more reliable and accurate results, representations and predictions

ADDITIONAL FEATURE AND FUTURE DEVELOPMENT

In future we would like to add the following features:

1. SPM will mark the answer script automatically.
2. A compiler will be added so that coding part can be marked easily.
3. We would like to give the access to the Higher Authority so that they can directly check the course curriculum and make changes if they want.
4. We would like to add course material where faculty members will be able to see how much time a student is spending on every course material.