



Student Performance Monitor System 3.0

Database Management System Project Report

STUDENT PERFORMANCE MONITORING SYSTEM 3.0

CSE303, SECTION 2, GROUP 1

Project Completed By:

Farzana Tabassum	1811014
Hasibul Haque	1821876
Rafi Khan Khadem	1830411
Md. Tariqul Islam	1830444
Md. Rahad Bin Islam	1820755
Mahfuzur Rahman	1811077

Course Co-instructed By:

Ms. Sadita Ahmed
Mr. Abu Sayed

CONTENTS

STUDENT PERFORMANCE MONITORING SYSTEM 3.0.....	1
<i>cse303, section 2, group 1</i>	<i>1</i>
CONTENTS	2
CHAPTER 1: INTRODUCTION	3
BACKGROUND OF THE ORGANIZATION	3
BACKGROUND OF THE PROJECT	3
OBJECTIVE OF THE PROJECT	4
SCOPE OF THE PROJECT.....	4
CHAPTER 2: REQUIREMENT ANALYSIS.....	5
EXISTING SYSTEM (WITH RICH PICTURE)	5
SIX ELEMENT ANALYSIS (AS IS).....	7
BUSINESS PROCESS DIAGRAM (AS IS)	18
PROBLEM ANALYSIS	21
PROPOSED SYSTEM (WITH RICH PICTURE)	24
SIX ELEMENT ANALYSIS (TO BE)	25
BUSINESS PROCESS DIAGRAM (TO BE)	50
CHAPTER 3: LOGICAL SYSTEM DESIGN	56
BUSINESS RULE	56
ERD	57
RELATIONAL SCHEMA	58
NORMALIZATION	59
DATA DICTIONARY	60
CHAPTER 4: PHYSICAL SYSTEM DESIGN	67
INPUT FORM:.....	67
OUTPUT FORM:	69
CHAPTER 5: CONCLUSION	89
PROBLEM:.....	89
SOLUTION:	89

CHAPTER 1: INTRODUCTION

The Independent University, Bangladesh currently has five academic schools:

- Business & Entrepreneurship
- Engineering, Technology & Science
- Environment and Life Science
- Liberal Arts & Social Science
- Pharmacy and Public Health

The university has robust and versatile departments under these academic schools

The mission of this university is to deliver a high-quality education and it has been an active participant in the education sector of Bangladesh. [1]

IUB have achieved this by closely working with government educational institutions

Such as University Grants Commission (UGC), Ministry of Education, The Institute of Engineers, Bangladesh (IEB) and other necessary educational institutions for each of the schools and departments under them, based on Outcome-Based Education (OBE) regularly updating the curriculum of the courses. According to the standards set by UGC and the Bangladesh Government, IUB is constantly tracking student performance for every semester and mainly using OBE for monitoring student performance and setting university curriculum.

The focus of this report is to study the current existing system of IUB and other universities. The aim of this report is to propose a new and better improved system that will improve the shortcomings, reduce the errors and make necessary improvements to provide better scholars.

BACKGROUND OF THE ORGANIZATION

Independent University, Bangladesh was established in 1993. It is one of the oldest private universities in Bangladesh, currently has more than an estimated 7,048 undergraduate and graduate students and over 10,455 alumni. This student population is mostly predicted to grow at 10% annually. [2]

IUB uses smart and new methods of education which are robust and is committed to producing graduates who will be equipped to provide new leadership through skilled employment. [2]

IEB provides a set of standards in an Accreditation Manual in the form of Program Educational Objectives (PEO) and Program Learning Outcomes (PLO) for specific departments [1]. These PLO and PEO are mapped to specific courses by the relevant Course Instructor. This will help the stakeholders to keep track of the performance of faculties, courses, departments and schools and provides valuable insight for making necessary improvements.

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. We measure the output of students, faculties, departments, and their respective courses in order to measure their productivity in regard to the outcome relevance of the course activities. We provide a range of tools and data intended to help universities and education authorities such as IEB and other stakeholders to evaluate the performance of students and inform strategies for improvements.

OBJECTIVE OF THE PROJECT

Our goal for this project is to create an open and collaborative environment through which everyone has complete insight of the performance of every student. Every student will be able to track their progress; university officials will be able to track student progress for which they are responsible. Government officials and other universities, who are willing to collaborate, can also view other university performance in the system to constructively improve the education system. Parents and guardians will be able to view the performance report of their child(ren). And, we intend to achieve all these objectives with remote accessibility and operability as the primary way of usage. This will allow uninterrupted monitoring and development of educational quality, and increase productivity of the users.

SCOPE OF THE PROJECT

Our goal for this project is to create an open and collaborative environment through which everyone has complete insight of the performance of every student. Every student will be able to track their progress; university officials will be able to track student progress for which they are responsible. Government officials and other universities, who are willing to collaborate, can also view other university performance in the system to constructively improve the education system. Parents and guardians will be able to view the performance report of their child(ren). And, we intend to achieve all these objectives with remote accessibility and operability as the primary way of usage. This will allow uninterrupted monitoring and development of educational quality, and increase productivity of the users.

CHAPTER 2: REQUIREMENT ANALYSIS

Requirement Analysis is the process of defining user expectations for a new or modified product. [3]. By doing the Analysis, we can see how the stakeholders interact with each other. We have also got the idea of the work process of the stakeholders in monitoring the student performance. Since different universities are involved in this project, we have also analyzed the way of monitoring student performance by observing different universities websites. We have also analyzed the previous reports done on this project and by talking with different stakeholders we have done our analysis on monitoring student performance.

EXISTING SYSTEM (WITH RICH PICTURE)

Rich picture provides a way to explore, acknowledge and define a business process and express it through diagrams to create a preliminary mental model.[4]

After gathering and analyzing all the information we have realized that in our existing system there is no system through which student PLO performance can be monitored. In our existing system rich picture, we have seen that OBE documents are prepared manually and we have also seen that if UGC ask for OBE documents from a University the Department/ Registrar Office have to manually collect the OBE documents and have to manually compile the documents to send it to UGC. While analyzing the documents we have seen that our existing system can't calculate the PLO and faculty have to manually calculate the PLO through excel sheet. Students can't track their own PLOs achievement and failure. Even the Department has to manually compile and collect the OBE marksheets and course assessment report from faculties. So, we have seen that most of the work in our system has been done manually.

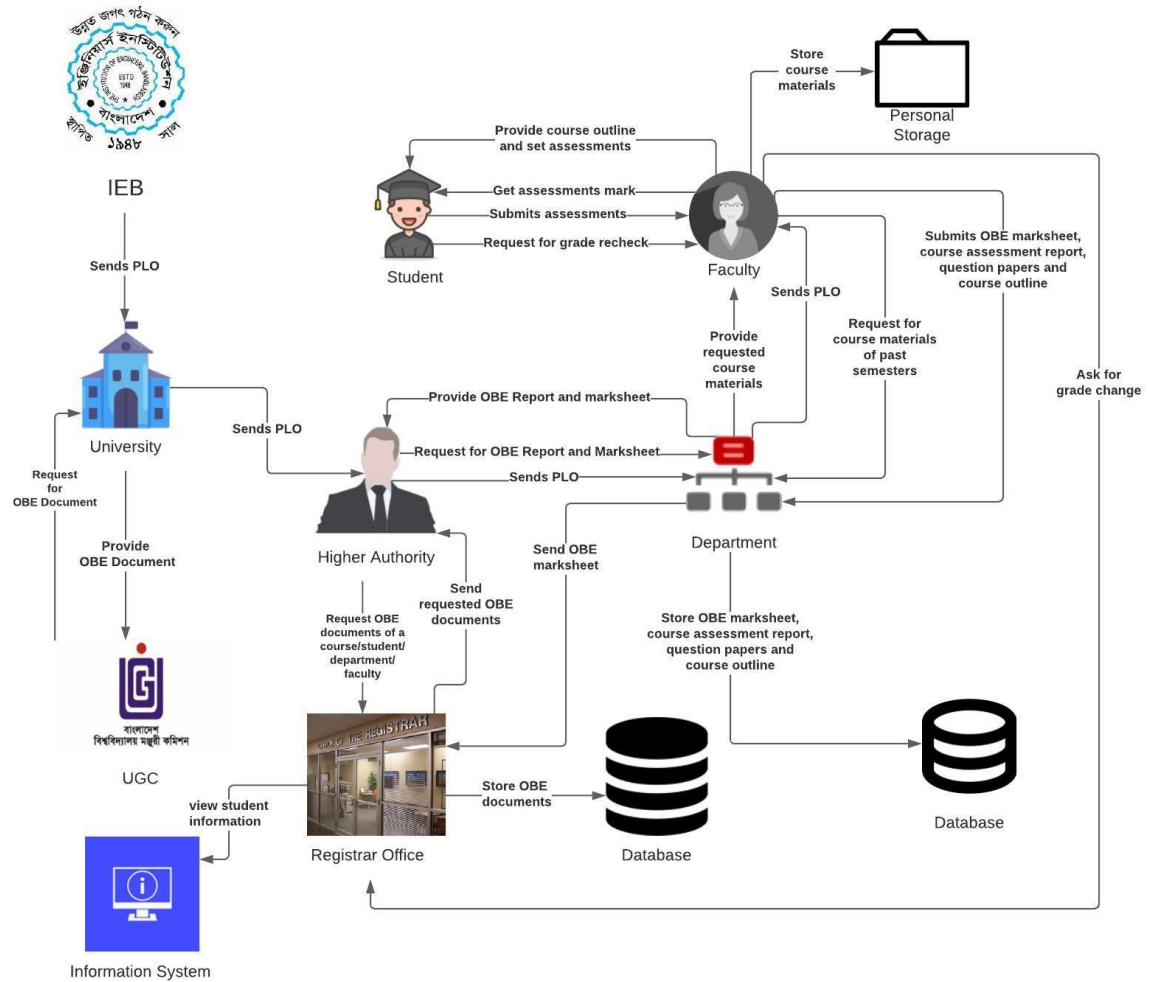


Figure2.1: Rich Picture (As Is)

SIX ELEMENT ANALYSIS (AS IS)

Process	System Roles					
	Human	Non-comp. Hardware	Computing Hardware	Software	Database	Network & Communication
Course outcomes (COs) and Program Learning Outcomes (PLOs) mapping	<p>IEB/UGC:</p> <p>1) Send Accreditation manual with PLOs defined to the Head of Department / Dean of School.</p> <p>Head of Department / Dean of School:</p> <p>1) Receive Accreditation manual from IEB.</p> <p>2) Send the accreditation manual to the Department.</p> <p>3) Direct the Department to tell the faculties to design the course outline, course assessment report and assessments and monitor the student performance according to the PLOs defined in the accreditation</p>	<p>Pen and Paper:</p> <p>To note down the necessary Information/Ideas regarding CO-PLO mapping.</p>	<p>Computer:</p> <p>1) Faculty/ Course Coordinator use computer to make soft copies of COs and Course Outline of the courses they are assigned for teaching.</p> <p>Printer:</p> <p>1) To print out hard copies of Course outline</p>	<p>MS Word:</p> <p>Faculty/ Course Coordinator use it to make a course outline and mapping of COs with PLOs.</p> <p>MS Excel:</p> <p>Faculty/ Course Coordinator use it to map COs with the individual assessments and check whether individual students have achieved PLO or not.</p>	<p>Personal Storage:</p> <p>Used By Faculty to store/access the mapping of COs with PLOs in cloud or physical storages.</p> <p>File Cabinet:</p> <p>Used by Faculty / Course Coordinator to maintain the necessary course related documents in hard copy.</p>	<p>Internet & Email:</p> <p>Used it to send mail to the stakeholders to discuss information regarding CO-PLO mapping.</p>

	<p>manual provided by IEB.</p> <p>Department:</p> <p>1) Receive from the Accreditation manual from Head of Department / Dean of School.</p> <p>2) Send the accreditation manual to the faculties.</p> <p>Faculty:</p> <p>1) Gathers information regarding Course Outcomes from IEB accreditation manual.</p> <p>2) Create a list of Course Content if it doesn't exist in course outline.</p> <p>3) Create list of COs from Course Content.</p> <p>4) Maps COs with PLOs.</p> <p>5) Maps COs to individual questions of Quiz, Midterm, Final term, Assignments, Project work</p>					
--	--	--	--	--	--	--

	<p>i.e. the assessments through which the student performance can be monitored.</p> <p>6) Send Course outline to students.</p> <p>7) Start to design Course assessment report using course outline and CO's.</p> <p>8) Store course materials to file cabinet/Personal storage.</p> <p>9) Submit the Course Outline to the Department.</p>					
Track Student's Assessment	<p>Faculty:</p> <p>1) Provides students with course outline.</p> <p>2) Provides students with course materials and course assessments throughout the semester.</p> <p>3) Receives assessment responses</p>	<p>Pen and paper:</p> <p>Required preparing for course outline, questions, course notes and OBE documents.</p>	<p>Computer:</p> <p>1) Used by faculty to prepare course outline, course materials and OBE documents. Used by students to prepare their assignments and attempt online assessments.</p>	<p>Document editor:</p> <p>1) Used by faculties to create or update soft-copies of the course materials and course assessments.</p> <p>2) Used by students to</p> <p>Spreadsheet software:</p>	<p>Local hard-disk and cloud storage:</p> <p>Used for storing soft-copies of course materials, course outlines and OBE documents</p> <p>Files, Folders</p>	<p>Internet:</p> <p>1) Used by faculties to store and retrieve course materials from online storage and share with students.</p> <p>2) Used for online classes and exams</p> <p>3) Used for online communication</p>

	<p>from the students</p> <p>4) Checks each of the responses and awards a mark to each of the assessments.</p> <p>5) Returns marks to students for each of their assessments.</p> <p>6) Stores all the course materials and assessment marks either in their computer hard-disk, personal cloud storage or local file cabinet.</p> <p>Student:</p> <p>1) Receives the course outline from the faculty.</p> <p>2) Receives the course materials and course assessments from the faculty.</p> <p>3) Attempts the problems of assessments and submits them to faculty</p> <p>4) Receives marks for the</p>		<p>2) In online classes, used by both faculties and students to conduct their class.</p> <p>Printers:</p> <p>Used by both faculties and students to print necessary documents.</p>	<p>Used by faculties to keep track of marks received on each questions, COs and PLOs by each student.</p> <p>Google Classroom and Meet:</p> <p>Used by faculties and student to conduct classes, exams and assignments online.</p> <p>Email Client:</p> <p>Used by both faculties and students for online communication.</p>	<p>and Cabinet:</p> <p>Required for storing hard-copy of the course materials, course outline and OBE documents</p>	<p>ions between students and faculties.</p>
--	---	--	---	--	--	---

	assessments from the faculty.					
Produce and submit OBE marksheet and course assessment report, and submit course outlines	Faculty: 1) Update student marks achieved in each questions of assessments in the OBE marksheet. 2) Update percentage of marks achieved in each COs mapped with each questions. 3) Find whether a student has achieved a particular CO or not. (40% or more) 4) Find whether a student has achieved a PLO for the corresponding COs achieved from the particular course. 5) Analyze how many students were able to achieve a particular CO with corresponding PLO by	Pen and paper: May be used to prepare hard-copy of OBE marksheet and assessment report.	Calculator: Used to calculate CO percentage mark achieved. Computer: To create and/or update softcopy of OBE marksheet and assessment report. Printer: May be used to print hard-copies of OBE marksheet and assessment report.	Operation System: Used by users to operate their computers. Office Suite: To create or modify OBE marksheet and assessment reports. Email Client: May be used to transfer OBE marksheet and assessment reports from faculties to the department and from department to registrar's office.	Database: 1) Local hard-disk, cloud storage and/or local cabinet used by faculties to store OBE marksheet, assessment reports and course outlines. 2) Department's central database and registrar's office's central database to store OBE marksheet, assessment reports and course outlines.	Internet: Used to access cloud storage and central databases, exchange emails and online office suites.

	<p>inputting them in an excel file/spreadsheet.</p> <p>6) Produce verdict in the OBE marksheet.</p> <p>7) Make Course Assessment Reports by using the course outline.</p> <p>8) Send the course outline, OBE marksheet and course assessment to the Department.</p> <p>Department:</p> <p>1) Receives course outline, OBE marksheet and course assessment report from faculties.</p> <p>2) Sends course outline, OBE marksheet and course assessment report to Registrar's office.</p> <p>Registrar's Office:</p>					
--	---	--	--	--	--	--

	1) Receives course outline, OBE marksheet and course assessment report from department.					
Request for grade change	<p>STUDENTS:</p> <p>Requests faculties for assessment script recheck and change grade if necessary.</p> <p>Faculty:</p> <p>1) Receives request for script recheck.</p> <p>2) Rechecks the assessment scripts and assignment of the student who requested for recheck.</p> <p>3) After recheck if changing marks needs to made, then faculty need to check whether CO's corresponding PLO change or not, for a specific student else end the process.</p>	<p>Pen and paper:</p> <p>1) May be used to note down any marks related information on the student assessment.</p> <p>2) May be used by both students to send written request for script recheck and grade change to faculties and faculties for grade change to registrar's office.</p>	<p>Computer or phone:</p> <p>1) Used by faculties, students and registrar's office to communicate with each other.</p> <p>2) Used by registrar's office access their database and change grade if necessary.</p>	<p>Operating system:</p> <p>Any OS used by the users on their computers and phones.</p> <p>An Information system:</p> <p>Used by registrar's office to access their database and make grade amendment.</p> <p>Email Client:</p> <p>Used by students, faculties and registrar's office to communicate between each other.</p>	<p>Registrar's office database:</p> <p>To store student grade data.</p> <p>Local hard-disk or cloud storage:</p> <p>Used by faculties to store student assessments, assignments and marksheets.</p>	<p>Internet:</p> <p>1) Used by registrar's office staffs to access Registrar's office information system.</p> <p>2) To send email for communications between students and faculties and faculties and registrar's office.</p>

	<p>4) If CO's corresponding PLO is changed, faculty update OBE marksheet then faculty asked for a grade change to registrar office else end the process.</p> <p>Registrar's office:</p> <p>1) Receives grade change request from a faculty for a specific student on a specific course.</p> <p>2) Changes grade based on faculty request for a specific student on a specific course.</p>					
Inspect and analyze student's performance, course assessment report over a range of semesters	<p>VC/Dean of school/ head of department:</p> <p>1) Request to the registrar office to send the records of OBE marksheets, assessment reports to analyze students' performance</p>	<p>Pen and Paper:</p> <p>1) Used for noting down some important points if needed.</p> <p>2) Used for any hard-copy of the</p>	<p>Computer:</p> <p>1) Used to create, update and display different reports like OBE marksheet, course assessment reports.</p> <p>Printer:</p>	<p>Operating System:</p> <p>1) Used by the users to operate their computers.</p> <p>Information System:</p> <p>1) Used by the registrar office for getting any student</p>	<p>Registrar Office's records:</p> <p>1) Retrieval of OBE marksheets and course assessment reports and other necessary documents</p>	<p>Internet:</p> <p>1) Used to send OBE marksheets and course assessment reports through email.</p>

	<p>trends and analyze the quality of the courses.</p> <p>2) Receive OBE marksheets, assessment reports from registrar's office.</p> <p>3) Analyze all the important data which came from the registrar office.</p> <p>Registrar's Office:</p> <p>1) Gather all the data such as OBE reports, assessment reports and other necessary documents.</p> <p>2) Send all the necessary documents to the higher authority.</p>	reports if needed.	Used to print any reports if govt. official visit the university and want to see in hard-copy.	information if needed	when needed.	
Provide OBE documents to UGC/IEB for inspection	<p>IEB/UGC:</p> <p>1) UGC will inform the VC of the university to submit the OBE report, Course assessment report and other required reports within</p>	<p>Pen and Paper:</p> <p>1) Used for noting down some important points if needed.</p> <p>2) Used for any hard-copy</p>	<p>Computer:</p> <p>1) Used to create, update and display different reports like OBE marksheet, course assessment reports.</p>	<p>Operating System:</p> <p>Used by the users to operate their computers.</p>	<p>Department records:</p> <p>1) Retrieval of OBE marksheets and course assessment reports</p>	<p>Internet:</p> <p>1) Used to OBE marksheets and course assessment reports through email.</p> <p>2) Online platforms like</p>

	<p>a specified time to check the quality of the degrees.</p> <p>2) Inform the VC of the university if any govt. official will visit the university for inspection.</p> <p>3) Visit university to receive the necessary reports for inquiry.</p> <p>VC:</p> <p>Request to the department heads to send the records of OBE marksheets, assessment reports to analyze students' performance trends and analyze the quality of the courses.</p> <p>Department head/Dean of School:</p> <p>1) Request to the department staff to gather the data like OBE marksheet,</p>	of the reports if needed.	<p>Printer:</p> <p>1) Used to print any reports if govt. official visit the university and want to see in hard-copy.</p>		<p>when needed.</p> <p>2) Store the feedback of the govt. official.</p>	<p>Microsoft Office Online to display reports of soft-copies.</p>
--	---	---------------------------	---	--	---	---

	<p>course assessment reports and other necessary data and return back all the data.</p> <p>2) Monitor all the important data which came from the department staff.</p> <p>3) Give feedback to the dept. staff if any change needed in the data.</p> <p>Department Staff:</p> <p>1) Gather necessary OBE reports, assessment reports and other necessary documents.</p> <p>2) Send all the necessary documents to the govt. official.</p>					
--	---	--	--	--	--	--

Table2.1: Six Element Analysis (As Is)

BUSINESS PROCESS DIAGRAM (AS IS)

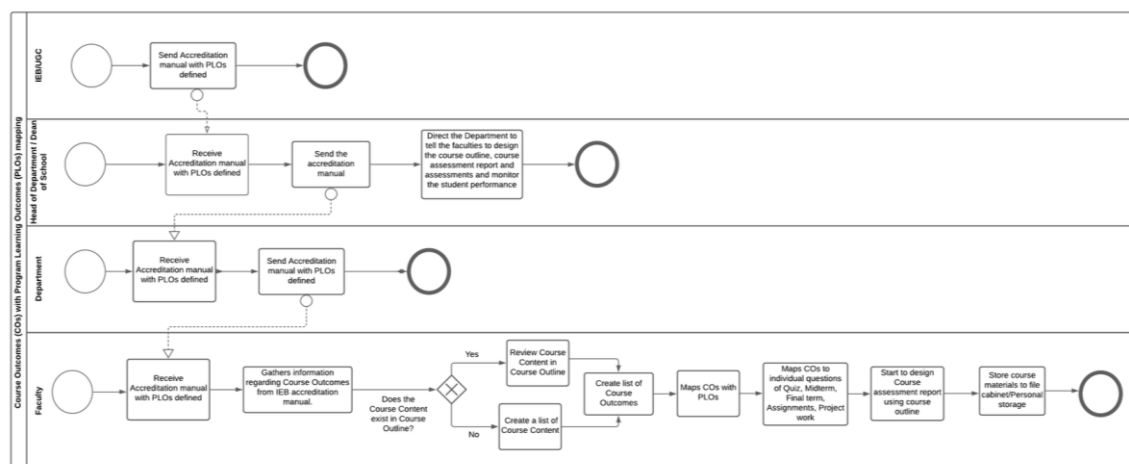


Figure 2.1: Course outcomes (COs) and Program Learning Outcomes (PLOs) mapping

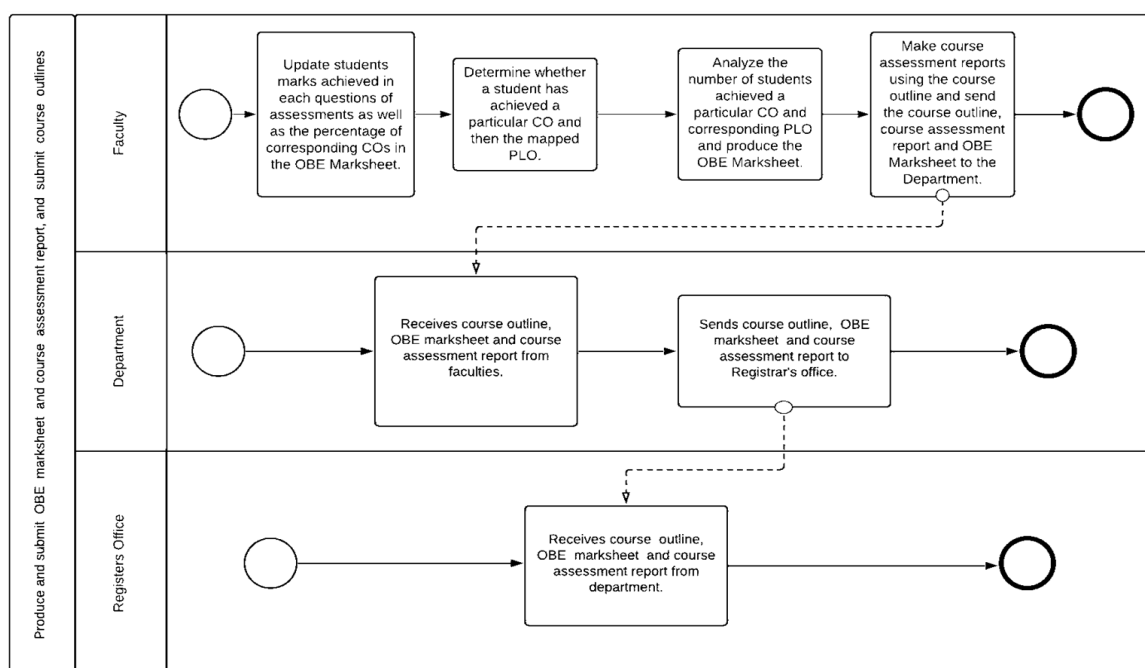


Figure 2.2: Produce and submit OBE marksheet and course assessment report, and submit course outlines

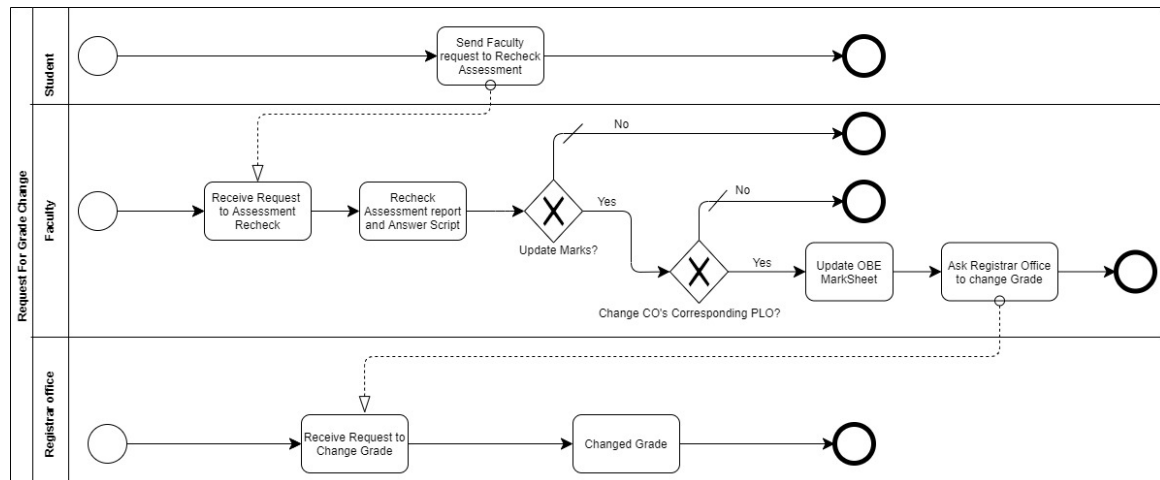


Figure 2.3: Request for Grade Change

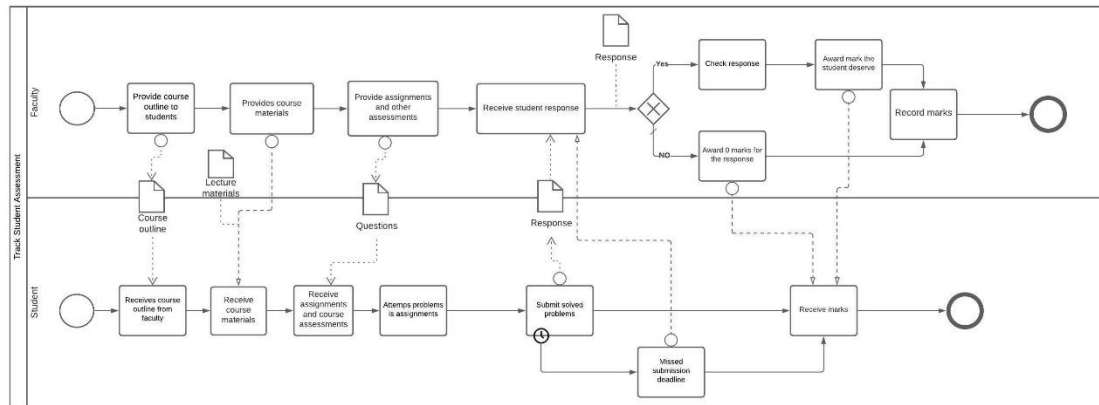


Figure 2.4: Track Student Assessment

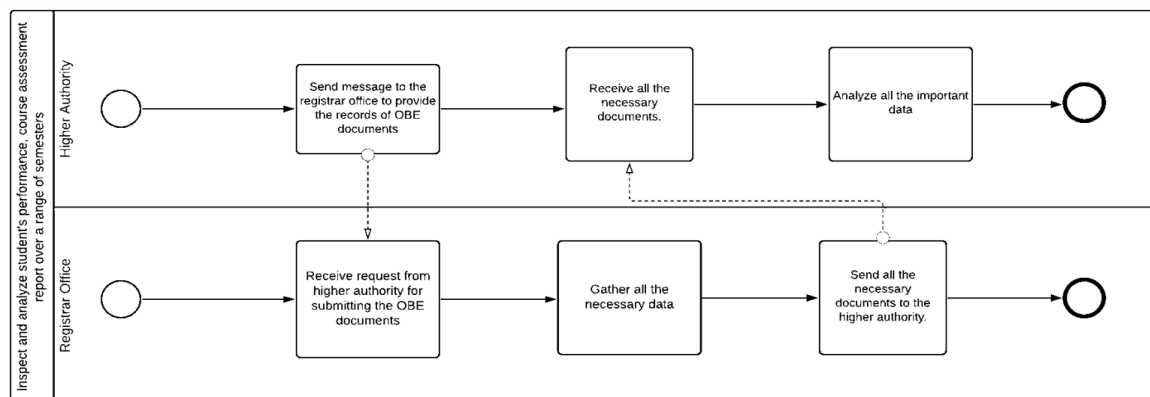


Figure 2.5: Inspect and analyze student's performance, course assessment report over a range of semesters

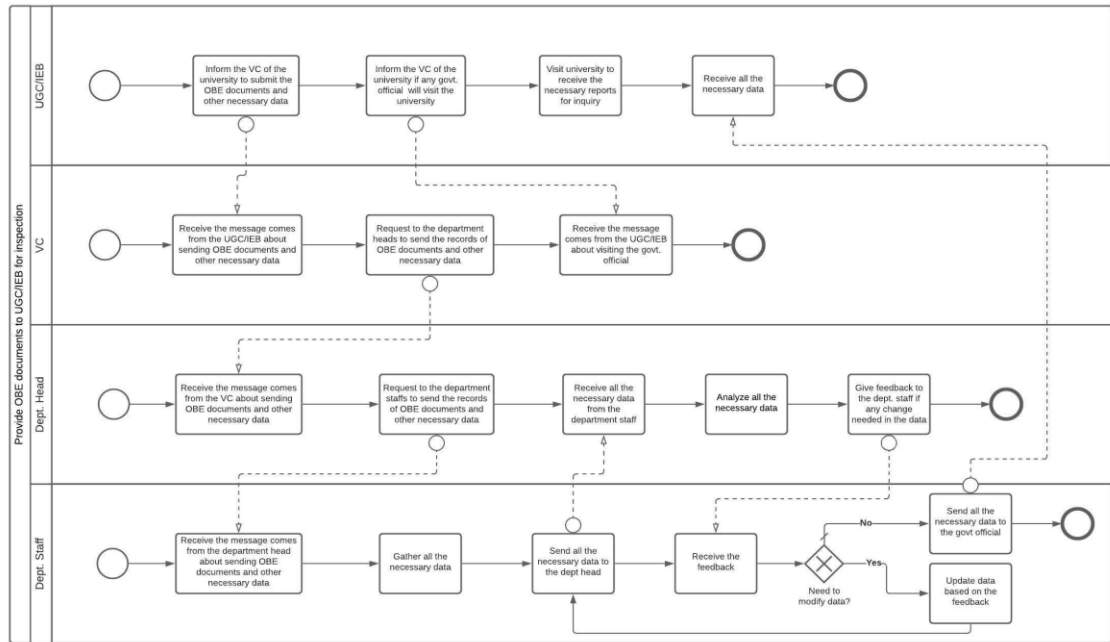


Figure 2.6: Provide OBE documents to UGC/IEB for inspection

PROBLEM ANALYSIS

Process	Stakeholders	Concerns (Problems)	Analysis(Reason of the problems)	Proposed Solution
Course Outcomes (COs) with Program Learning Outcomes (PLOs) mapping	1) Faculty	This process is lengthy, time consuming, consists of trivial tasks and error prone.	<p>1) Faculties have to go through different documents, like course content, accreditation manual and course outlines.</p> <p>2) Faculties and department stuffs have to maintain bundles of documents and going through them one by one is hard, time consuming and prone to error.</p> <p>3) CO-PLO mapping and question to CO mappings have to be done manually.</p>	<p>1) Our system will allow the Faculty to input the course code and its course outcomes and then the system will map the received CO's with the PLO's for that particular course.</p> <p>2) Our system will also map the received CO's with individual questions of the Quiz, Midterm, and Final Term etc. In this way, it will be more efficient.</p>
Produce and submit OBE marksheet and course assessment report, and submit course outlines	<p>1) Faculty</p> <p>2) Department</p>	This process is lengthy, time consuming, consists of trivial tasks and error prone. Also, there is cost factor associated with it.	<p>1) Faculties have to produce OBE marksheet using spreadsheet software which is expensive, lot of different features that might not required for OBE marksheet and not so user friendly for this purpose.</p> <p>2) When submitting the marksheets, either they have to print and submit it in person to the department or email a softcopy of it. That means they have been switching different software and services to achieve the purpose. Also,</p>	1) In our system, faculties can input the marks for each question for individual students. The software will automatically calculate the CO and PLO achievement percentages.

			there are chances that any of the documents getting lost during the process.	
Request for grade change	1) Faculty 2) Registrar's office	Time consuming and requires too many human staffs for this purpose.	1) Faculties have to request the registrar's office for the grade change. There could be too many requests for grade change to the registrar's office which means they might overlook some of the request and may delay or not change the grades for some students by mistake.	1) In our system, faculty will be able to enter marks of each of the students for each of the courses to our database and modify marks when needed without any interference from the registrar's office.
Inspect and analyze student's performance, course assessment report over a range of semesters	1) Higher authority 2) Registrar's office 3) Department	Time consuming, error-prone, cumbersome and involvement of too many staffs.	1) Since, we do not have any centralized repository of raw data, faculties and higher authority have to go through different OBE marksheets to understand trends of student performance. This hard, time consuming and errors may occur during the process. 2) Registrar's office and departments have to maintain too many documents which are hard and require too many workforces.	1) In the proposed system, users will not have to go through so many marksheets and documents to understand student PLO performance trends. 2) They will be able to view software generated charts and PLO performance trends for many different queries.
Provide OBE documents to UGC/IEB for inspection	1) Department Head 2) Department Staffs 3) UGC/IEB	Time consuming, strenuous, error-prone.	1) In order to view student PLO performance trends for a university, UGC/IEB has to wait for OBE document from that university to	1) Our system will generate necessary charts for UGC/IEB to view necessary performance trends without needing to rely

			<p>arrive. This could be time consuming and easily get lost.</p> <p>2) Also, they have to go through every single document to understand them.</p> <p>3) Also, compiling every single OBE document and sending them to UGC/IEB is also very time consuming and strenuous for university staffs, department staffs and head of departments as well.</p>	solely on OBE documents.
View student's PLO performance information	<p>1) Students</p> <p>2) Guardians</p>	No access for either students or guardians to view PLO and PLO performance trends of individual courses.	<p>1) Since there is no access for viewing PLO performance, students or guardians may not be able to get better insight about the students' performance.</p>	<p>1) Each student can view their PLO achievements through their SPMS 3.0 dashboard.</p> <p>2) Parents will also be able to see their child(ren)'s PLO performance through the dashboard.</p>

Table 2.2: Problem Analysis

PROPOSED SYSTEM (WITH RICH PICTURE)

In our proposed system, we are going to make the whole system efficient by reducing the human involvement. It will have a central SPMS system that will host all the necessary data. Users will be able to upload necessary data to the SPMS and other users like students, faculties, higher authorities, guardians and UGC/IEB can view them to monitor students and faculty performance. Faculties can also store their assessment documents for easier access by other authorized university staffs and input or update student PLO marks. UGC/IEB will be able to add PLOs to the system and faculty will be able to map questions to COs and map COs to PLOs. We are expecting this newer system will improve time efficiency vastly and reduce material resource requirement.

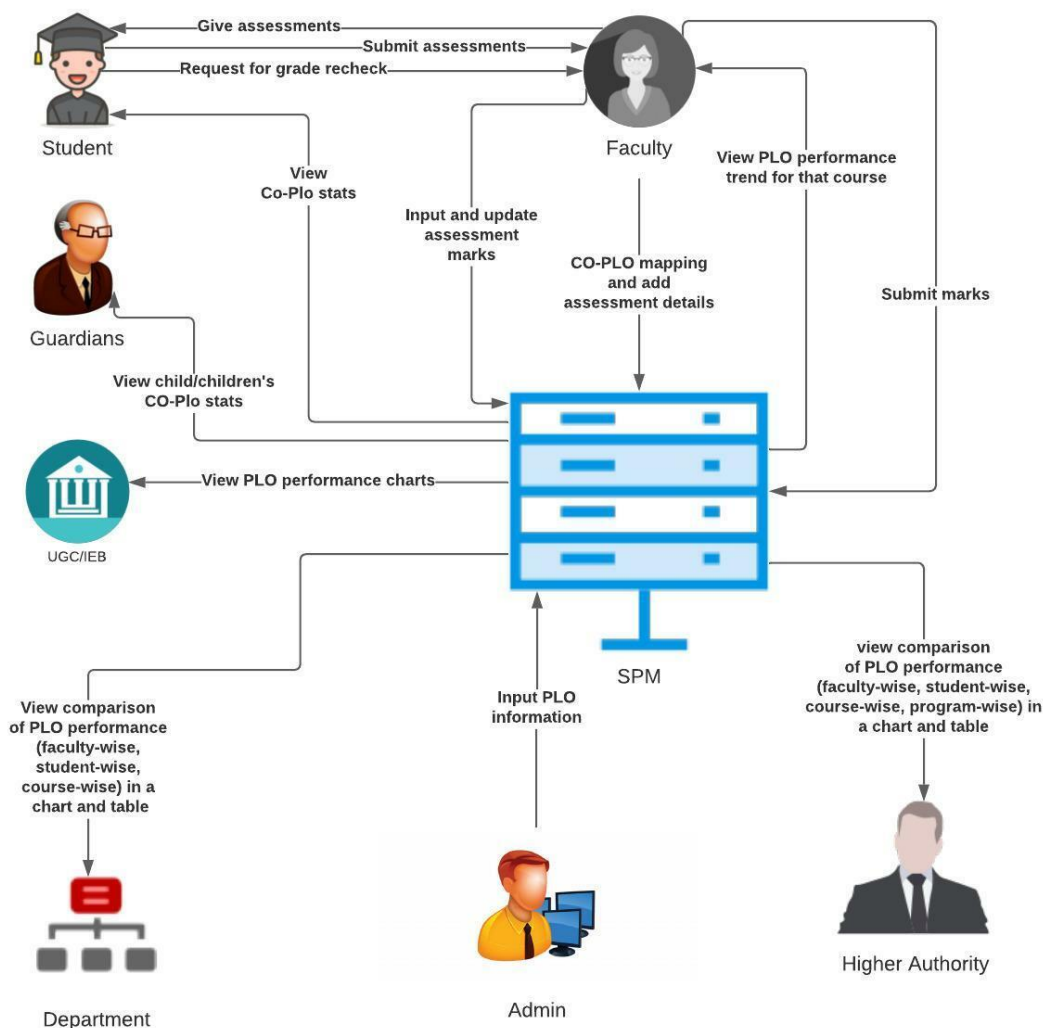


Figure 2.8: Rich Picture (To Be)

SIX ELEMENT ANALYSIS (TO BE)

Process	System Roles					
	Human	Non-Computing Hardware	Computing Hardware	Software	Database	Network & Connection
Course Outcomes (COs) with Program Learning Outcomes (PLOs) mapping	<p>Admin:</p> <ol style="list-style-type: none"> 1) Login in to the system with correct user ID and password. 2) Head over to the designated section. 3) Enter PLO title and description in the designated textboxes for each of the PLOs. 4) Press submit button to submit the new PLOs received from the UGC/IEB into the SPMS 3.0 system. <p>Faculty:</p> <ol style="list-style-type: none"> 1. Login to the SPMS 3.0 with correct user ID and password. 2) Head over to the designated section. 3) Select a course from the drop down menu. 	<p>Pen and Paper:</p> <ol style="list-style-type: none"> 1) Used for noting down some important points if needed. 	<p>Computer:</p> <ol style="list-style-type: none"> 1)All related data is searched and stored using a computer. 	<p>SPMS:</p> <ol style="list-style-type: none"> 1)Used to map the CO's with the PLO's by the input taken from faculty. 2)Used to map theCO's with the individual questions of the assessment by the faculty of that particular course. 	<p>SPMS Database:</p> <ol style="list-style-type: none"> 1) Used to store the mapping of theCO's with the PLO's. 2) Used to store the mapping of theCO's with the individual questions of the assessment by the faculty of that particular course. 	<p>Internet:</p> <ol style="list-style-type: none"> 1) All related data are searched through the internet. 2) Required to login and access the SPMS 3.0

	<p>4) New PLOs will be shown and existing CO-PLO mapping from the previous semester will be shown.</p> <p>5) In order to modify the CO-PLO mapping, click on the modify button.</p> <p>6) A form will be displayed where users can select new PLOs from a drop down menu for each of the COs.</p> <p>7) Click on update button to submit the CO-PLO mapping.</p>					
Map Course Assessments with COs for a Selected Course	<p>Faculty:</p> <p>1) Login to SPMS 3.0 with correct user ID and password.</p> <p>2) Head over to the designated section.</p> <p>3) Select a course by entering course ID in a text field and press submit button to initiate the process.</p> <p>4) For each of the assessment, click on add button.</p> <p>5) A form will appear where</p>		<p>Computer:</p> <p>1) Used to access the SPMS 3.0</p>	<p>SPMS 3.0:</p> <p>1) Enter assessments marks and mapping with COs.</p> <p>2) Track student marks.</p>	<p>SPMS 3.0 DB:</p> <p>1) Store assessment marks, CO of a course.</p>	<p>Internet:</p> <p>1) Access SPMS 3.0</p>

	<p>faculty can input assessment title and total achievable mark in a textbox for each of the questions.</p> <p>6) Select CO from a dropdown menu.</p> <p>7) Click submit button after entering all the assessment details and mapping with CO to submit the questions, achievable marks.</p>					
Submit Student Marks to SPMS 3.0	<p>Faculty:</p> <ol style="list-style-type: none"> 1. Log into the system using userID and password. 2) Head over to the designated section. 3) Select a course by entering course ID in a text field and press submit button to initiate the process. 4) Select a student by entering student ID into a textbox. 5) Press submit button to select the student. 6) Input marks for each of the questions of 	<p>Pen and Paper:</p> <ol style="list-style-type: none"> 1) Used for student answer scripts, assignments and/or paper spreadsheet where he or she might have recorded student marks previously. 	<p>Computer/ Mobile:</p> <ol style="list-style-type: none"> 1. Used to access the SPMS 3.0 software. <p>Printer:</p> <ol style="list-style-type: none"> 1. Can be used to print out reports if needed. 	<p>SPMS 3.0:</p> <ol style="list-style-type: none"> 1. The software will be used to upload the mark. <p>MS Office:</p> <ol style="list-style-type: none"> 1. Store students marks in excel/spread sheets throughout the semester. 	<p>SPMS 3.0 Database:</p> <ol style="list-style-type: none"> 1. Store student marks, courses details and students details. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. Required to login and access the SPMS 3.0

	<p>each of the assessments.</p> <p>7) Click save button to save the marks in the SPMS 3.0.</p>					
Request for grade change	<p>Faculty:</p> <ol style="list-style-type: none"> 1) Recheck the assessment scripts and assignment of the student who requested for recheck. 2) Login to the SPMS 3.0 using correct user ID and password. 3) In case of mark change, the faculty has to move to the designated section. 4) Enter student ID in a textbox for whom the mark will be changed. 5) Press submit button to select the student. 6) Every single question in every single assessment along with the text fields containing the marks achieved will be shown. 7) Faculty will be able to enter new mark in the required text field. 	<p>Pen and Paper:</p> <p>Used for noting down some important points if needed.</p>	<p>Computer/ Mobile:</p> <ol style="list-style-type: none"> 1. Used to access the SPMS software. 2. To check email. 3. To update the OBE marksheet if needed. 	<p>SPMS :</p> <p>The software will be used to update OBE marksheet and submit the modified marksheet.</p>	<p>SPMS Database:</p> <p>To store student assessments, assignments and marksheets.</p>	<p>Internet:</p> <ol style="list-style-type: none"> 1. Required to login and access the SPMS. 2. To send email for communications between students and faculties. 3. Used the internet to update the OBE marksheet and submit the modified marksheet in SPMS.

	8) Upon clicking on save button, new changes will be saved in the SPMS 3.0.					
Compare Individual Student PLO Performance against Average PLO Score for a Selected Course	<p>Student:</p> <ol style="list-style-type: none"> 1) Login to SPMS 3.0 with valid user ID and password. 2) Head over to the course-wise PLO comparison section to view the chart. 3) Select a course from a drop down menu 4) Click show button to display the chart. 5) Chart will be displayed showing bar chart of PLO score achieved in each of the PLOs of the course against the course average. <p>Faculty/VC/Head of Department/Dean of School:</p> <ol style="list-style-type: none"> 1) Login to SPMS 3.0 with valid user ID and password. 2) Head over to the course-wise PLO comparison 	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view PLO comparison of an individual student against the average for a course.</p> <p>SPMS 3.0 Server:</p> <ol style="list-style-type: none"> 1) Used for receiving and responding the users' request for showing individual student PLO comparison against the average for a course. 2) It retrieves the PLO score data for a course and calculates the average. 	<p>SPMS 3.0:</p> <p>Used to view PLO achievements.</p> <p>Operating System:</p> <p>Used by the users to operate their computer.</p>	<p>SPMS 3.0 DB:</p> <ol style="list-style-type: none"> 1) It stores login credentials for all the users. 2) Used to store records for PLO achievements. 3) It stores parent-children relation for all the guardians. 	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>section and select a course.</p> <p>3) VC, Dean of School, Head of department and faculty have to enter student ID to select a student.</p> <p>4) Click on show button to generate and show the chart.</p> <p>5) Chart showing bar chart of PLO score achieved in each of the PLOs of the course against the course average will be displayed.</p>					
Compare Individual Student PLO Performance against Average PLO Score for a Selected Program	<p>Student:</p> <p>1) Login to SPMS 3.0 with valid user ID and password.</p> <p>2) Head over to program-wise PLO comparison section to view the chart.</p> <p>3) Click on show button to generate and display the chart.</p> <p>4) Chart will be displayed</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view PLO comparison of an individual student against the average for a selected program.</p>	<p>SPMS 3.0:</p> <p>Used to view PLO achievements.</p> <p>Operating System:</p> <p>Used by the users to operate their computer.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for PLO achievements.</p> <p>3) It stores parent-children relation for</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>showing individual student PLO comparison against the average PLO scores a selected program.</p> <p>VC/Faculty/Head of Department/Dean of School:</p> <p>1) Login to SPMS 3.0 with valid user ID and password.</p> <p>2) Head over to the program-wise PLO comparison section and select a program.</p> <p>3) VC, Dean of School, Head of department and faculty have to enter student ID to select a student.</p> <p>4) Click on show button to generate and show the chart.</p> <p>5) Chart will be displayed showing individual student PLO comparison against the average PLO scores a selected program.</p>		<p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing individual student PLO comparison against the average for a program.</p> <p>2) It retrieves the PLO score data for a program and calculates the average.</p>		all the guardians.	
--	--	--	--	--	--------------------	--

View Individual Student PLO achievement and Failed to Achieve Table Including the Received Score	<p>Student:</p> <ol style="list-style-type: none"> 1) Login to SPMS 3.0 with valid user ID and password. 2) Head over to the designated section to view a table containing record. 3) Click on the show button to generate and show the table. <p>VC/Faculty/Head of Department/Dean of School/Guardians:</p> <ol style="list-style-type: none"> 1) Login to SPMS 3.0 with valid user ID and password. 2) Head over to the designated section to view the table 3) Faculties, Head of Department, Dean of School and VC have to enter the student ID in the text box to select the student. 4) Guardians will be able select the student recorded as a child for those guardians from a drop down menu. 	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view PLO achievement table for individual student</p> <p>SPMS 3.0 Server:</p> <ol style="list-style-type: none"> 1) Used for receiving and responding the users' request for showing individual student PLO achievement table. 2) It retrieves the PLO score data for individual students. 	<p>SPMS 3.0:</p> <p>Used to view PLO achievement table.</p> <p>Operating System:</p> <p>Used by the users to operate their computer.</p>	<p>SPMS 3.0 DB:</p> <ol style="list-style-type: none"> 1) It stores login credentials for all the users. 2) Used to store records for PLO achievements. 3) It stores parent-child relation for all the guardians. 	<p>Internet:</p> <p>Used to access the SPMS server.</p>
--	--	--	---	--	---	--

	<p>5) Click show button to generate and show the table.</p> <p>6) A table containing records for PLO achievement VS failed to achieve for the student for each of the courses he or she has completed so far along with the achieved score.</p>					
<p>PLO Performance Comparison for all Instructors who have Taught a Selected Course</p>	<p>VC/ Dean of School/Head of Department:</p> <p>1) Login to SPMS 3.0 to correct user ID and password.</p> <p>2) Head over to the designated section to view the chart.</p> <p>3) Select the course for which the comparison to make from a drop down menu.</p> <p>4) Press generate button to generate and display the chart.</p> <p>5) An average PLO score-PLO graph will be shown for all the instructors of the course.</p> <p>Faculty:</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view PLO achievement for the instructors who taught the selected course.</p> <p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing instructor-wise PLO performance for a course.</p> <p>2) It retrieves the PLO score for all</p>	<p>SPMS 3.0:</p> <p>Used to view instructor-wise PLO performance graph.</p> <p>Operating System:</p> <p>Used by the users to operate their computer.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for PLO achievements, course-instructor records.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>1) Login to SPMS 3.0 with correct user ID and password.</p> <p>2) Head over to the designated section to view the chart.</p> <p>3) Select the course from all the courses that he or she has instructed for which the comparison to make from a drop down menu.</p> <p>4) Press generate button to generate and display the chart.</p> <p>5) An average PLO score-PLO graph will be shown for that individual faculty, along with average for the other faculties.</p>		students in the selected course and calculates their average.			
View PLO Comparison for Courses that have Selected PLO(s)	<p>VC/Dean of School/Head of Department:</p> <p>1) Login to the SPMS 3.0 with correct user ID and password</p> <p>2) Head over to the designated section to view the chart.</p> <p>3) Select any number of PLOs</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view PLO achievement for the courses those have selected PLOs.</p>	<p>SPMS 3.0:</p> <p>Used to view PLO performance for the courses with the selected courses.</p> <p>Operating System:</p> <p>Used by the users to</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for PLO achievements, course-PLO records.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>from a drop down menu.</p> <p>4) Select the upper bound and lower bound for range of semester using drop down menus.</p> <p>5) Click generate button to generate and display the chart.</p> <p>6) PLO comparison for all the courses will be shown that contains the selected PLOs.</p>		<p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing PLO performance for all the courses those have the selected PLOs.</p> <p>2) It retrieves the PLO score for all students in the courses those have the selected PLO(s) and calculates their average.</p>	operate their computer.		
--	--	--	---	-------------------------	--	--

<p>Chart Showing the Percentage of students who achieved and failed to achieve each of the PLOs in a range of semesters for a selected course</p>	<p>VC/Dean of School/Head of Department/Fa culty:</p> <p>1) Login to SPMS 3.0 with correct user ID and password</p> <p>2) Head over to the designated section to view the chart.</p> <p>3) Enter upper bound and lower bound of the semester range from a drop down menus.</p> <p>4) Select a course from a drop down menu. [VC will be able to select any course from the university, Dean of School will be able to select only the courses under his or her school, Head of Department will be able to select only the course under his or her department and faculties will be able to select only the course he or she has instructed.]</p> <p>5) Press generate button to generate and display the chart.</p> <p>6) A chart of percentage for the selected</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view PLO achieved VS failed within a range of semesters for a selected course.</p> <p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing PLO achieved VS failed for all the PLOs in the given semester range.</p> <p>2) It retrieves the PLO score for all students in the all the courses in the provided semester range and calculates their average.</p>	<p>SPMS 3.0:</p> <p>Used to view percentage of PLO achieved VS failed for all the PLOs in the given semester range.</p> <p>Operating System:</p> <p>Used by the users to operate their computer.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for PLO achievement s, course- PLO and semester- wise course records.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>
---	---	--	---	--	---	--

	course of students who achieved and failed to achieve in each of the PLOs in the course within the range of the semester will be shown.					
Chart Showing Count of Students who Achieved each PLOs VS Attempted for a range of semesters for a Selected Program	VC/Dean of School/Head of Department: 1) Login to SPMS 3.0 with correct user ID and password 2) Head over to the designated section to view the chart. 3) Enter the upper bound and lower bound of the semester range from a drop down menus.	Pen & Paper: May be used to take any PLO achievement related notes.	Computer: Used by the users to access SPMS 3.0 and view count of students of PLO achieved VS attempted in each of PLOs within a range of semesters for the selected program.	SPMS 3.0: Used to view count of student who a PLO achieved VS attempted for all the PLOs in the given semester range. Operating System: Used by the users to operate their computer.	SPMS 3.0 DB: 1) It stores login credentials for all the users. 2) Used to store records for student PLO achievements.	Internet: Used to access the SPMS server.

	<p>4) Select a program from a drop down menu. [VC will be able to select any program in the university, Dean of School will be able to select any program under his or her school and Head of Department will be able to select any program in the department they belong to.]</p> <p>5) Press generate button to generate and display the chart.</p> <p>6) A chart of count of students who achieved and attempted in each of the PLOs for that program within the range of the semester will be shown.</p>		<p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing the count of students who PLO achieved VS attempted in each of PLOs within a range of semesters for the selected program.</p> <p>2) It retrieves the count of students who achieved a PLO VS attempted for each of PLOs within a range of semesters.</p>			
A Chart Showing Count of Students who Achieved each PLO for a Range of Semesters for a Selected Program	<p>VC/Dean of School/Head of Department:</p> <p>1) Login to SPMS 3.0 with correct user ID and password</p> <p>2) Head over to the designated section to view the chart.</p> <p>3) Enter upper bound and lower bound of the</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view the chart for count of students who achieve a PLO for each of the PLOs within a range of semesters for</p>	<p>SPMS 3.0:</p> <p>Used to view the chart for the count of students achieved a PLO for all the PLOs in the given semester range for a selected program.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for student PLO achievements and</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>semester range from a drop down menus.</p> <p>4) Select a program from a drop down menu. [VC will be able to select any program in the university; Dean of school will be able to select only the programs under his or her school. Head of Department will be able to select only the programs under his or her department.]</p> <p>5) Press generate button to generate and display the charts.</p> <p>6) A chart showing count of students who achieved each of the PLOs of a range of semester.</p> <p>7) Upon clicking on any of the bar, a pie chart will be displayed to show count of students who achieved that PLO in courses that have the PLO.</p>		<p>a selected program.</p> <p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing the count of students who PLO achieved in each of PLOs within a range of semesters for a selected program.</p> <p>2) It retrieves the count of students who achieved a PLO for each of PLOs within a range of semesters.</p>	<p>Operating System:</p> <p>Used by the users to operate their computer.</p>	course-PLO relation.	
A Chart to Show PLO Achieved	UGC/IEB/VC/Dean of	Pen & Paper:	Computer: Used by the users to	SPMS 3.0: Used to view the chart	SPMS 3.0 DB:	Internet:

Count Comparison for each PLOs for a Selected Program for a Selected Universities	<p>School/Head of Department:</p> <p>1) Login to SPMS 3.0 with correct user ID and password</p> <p>2) Head over the designated section to view the charts.</p> <p>3) UGC/IEB has to select universities from a dropdown menu (VCs, Deans of School, Heads of Department will only view comparison for university they belong and will not have do the selection) for which comparison will be shown.</p> <p>4) Select a program from a dropdown menu (VCs will be able to select any program in the university they belong, Deans of school will be able to select any program under their school, Heads of Department will be to select any program under his or her department) for comparison chart and range of semesters.</p>	May be used to take any PLO achievement related notes.	<p>access SPMS 3.0 and view the chart for count of students who achieved a PLO for each of the PLOs within a range of semesters in a selected program.</p> <p>SPMS 3.0 Server:</p> <p>1) Used for receiving and responding the users' request for showing the count of students who PLO achieved in each of PLOs within a range of semesters in a selected program.</p> <p>2) It retrieves the count of students who achieved a PLO for each of PLOs within a range of semesters.</p>	<p>for the count of students achieved a PLO for all the PLOs in the given semester range for a selected program.</p> <p>Operating System:</p> <p>Used by the users to operate their computer.</p>	<p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for student PLO achievement and record for course-PLO, university and program.</p>	Used to access the SPMS server.
---	---	--	--	--	---	---------------------------------

	<p>5) Click generate button to generate and display the chart.</p> <p>6) A chart will be shown for count of students who achieved a PLO for each of the PLOs in the selected program within that semester range.</p>					
<p>Chart Showing Percentage of Student/Graduates who Achieved all the PLOs in a Selected Program</p>	<p>UGC/IEB/VC/Dean of School/Head of Department:</p> <p>1) Login to SPMS 3.0 with correct user ID and password</p> <p>2) Head over to the designated section to view the chart.</p> <p>3) UGC/IEB has to select universities from a dropdown menu (VCs, Deans of School, Heads of Department will only view comparison for university they belong and will not have to select any university) for which comparison will be shown.</p> <p>4) Select a program from a dropdown menu</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view the comparison</p> <p>SPMS 3.0 Server:</p> <p>1) It handles user request for view comparison of students/graduates who achieved all the PLOs.</p> <p>2) It generates necessary charts and comparison from the data retrieved from the database.</p>	<p>Operating System:</p> <p>Used by the users to operate their computer.</p> <p>SPMS 3.0:</p> <p>Produce and shows the comparison for percentage of students/graduates who achieved all the PLOs.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for student PLO achievements and record for course-PLO, university, schools, departments and program.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>(VCs will be able to select any program in the university they belong, Deans of school will be able to select any program under their school, Heads of Department will be to select any program under his or her department) for comparison chart.</p> <p>5) VC, Dean of School and Head of Department, but not UGC/IEB, will get to select whether to view comparison for graduates or students. UGC/IEB will only see comparison for graduates from a dropdown menu.</p> <p>6) Click on generate button to generate and display the charts</p> <p>7) A chart will be shown for percentage of students/graduates who achieved all the PLOs in the selected program.</p>					
--	---	--	--	--	--	--

<p>A Comparison to Show Percentage of Students who Achieved Selected PLOs</p>	<p>UGC/IEB/VC/Dean of School/Head of Department:</p> <ol style="list-style-type: none"> 1) Login to SPMS 3.0 with correct user ID and password. 2) Head over to the designated section to view the chart. 3) UGC/IEB has to select a university from a dropdown menu for which they want to view the comparison and VC, Dean of School, Head of Department will only view comparison for the university they belong to. 4) Select PLOs from a dropdown menu. UGC/IEB will be able view comparison for the whole university they have selected, VC will be able to view comparison for the whole university they belong, Dean of School will be able to see comparison for the school they belong, and Head of department will be able to view comparison for 	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view the comparison</p> <p>SPMS 3.0 Server:</p> <ol style="list-style-type: none"> 1) It handles user request to view comparison of percentage of students who achieved the selected PLOs. 2) It generates necessary charts and comparison from the data retrieved from the database. 	<p>Operating System:</p> <p>Used by the users to operate their computer.</p> <p>SPMS 3.0:</p> <p>Produce and shows the comparison of percentage of students who achieved the selected PLOs.</p>	<p>SPMS 3.0 DB:</p> <ol style="list-style-type: none"> 1) It stores login credentials for all the users. 2) Used to store records for student PLO achievements and record for course-PLO, university, school and department. 	<p>Internet:</p> <p>Used to access the SPMS server.</p>
---	---	--	--	---	---	--

	<p>the department they belong.</p> <p>5) Upon clicking on generate button, a chart will be generated and displayed to the users.</p> <p>6) A chart will be displayed that contains the percentage of student who achieved the selected PLOs.</p>					
<p>Chart Showing Percentage of Students who Achieved all the PLOs they Attempted</p>	<p>VC/Dean of School:</p> <p>1) Login to SPMS 3.0 with correct user ID and password.</p> <p>2) Head over to the designated section to view the charts.</p> <p>3) Enter upper bound and lower bound of the semester range from a drop down menus.</p> <p>4) VCs and Dean of Schools will have first select School-wise or Department-wise from a drop down menu.</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view the comparison</p> <p>SPMS 3.0 Server:</p> <p>1) It handles user request to view the comparison of the percentage of students who achieved all the PLOs they attempted in the selected schools or</p>	<p>Operating System:</p> <p>Used by the users to operate their computer.</p> <p>SPMS 3.0:</p> <p>Produce and shows the comparison of the percentage of students who achieved all the PLOs they attempted in the selected schools or departments.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for student PLO achievements and record for university, schools and departments.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>5) VCs have to select a range of departments or schools from a drop down menu. Dean of School has to select a range of departments under his school.</p> <p>6) The users have click generate button to generate and display the chart.</p> <p>7) A chart showing percentage of students who achieved all the PLO they attempted for the selected schools or departments will be displayed.</p>		<p>departments .</p> <p>2) It generates necessary charts and comparison from the data retrieved from the database.</p>			
<p>Chart Showing Average Count of PLOs Achieved for a Selected School or Department</p>	<p>VC/Dean of School:</p> <p>1) Login to SPMS 3.0 with correct user ID and password.</p> <p>2) Head over to the designated section to view the charts.</p> <p>3) Enter upper bound and lower bound of the semester range from a drop down menus.</p> <p>4) VCs will have first select School-wise or</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view the comparison</p> <p>SPMS 3.0 Server:</p> <p>1) It handles user request to view the comparison of average number of PLOs achieved by a student in</p>	<p>Operating System:</p> <p>Used by the users to operate their computer.</p> <p>SPMS 3.0:</p> <p>Produce and shows the comparison of average number of PLOs achieved by a student in the selected schools or departments.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for student PLO achievements and record for course-PLO, university, school and department.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>Department from a drop down menu.</p> <p>5) VCs will have to select either any schools or any departments from a dropdown menu. Deans of School will be able to select only departments under their school.</p> <p>6) A chart will be displayed to show the average count of PLOs a student has achieved in the selected schools or departments.</p>		<p>the selected schools or departments</p> <p>2) It generates necessary charts and comparison from the data retrieved from the database.</p>			
Instructor-wise PLO Performance Comparison	<p>VC/Dean of School/Head of Department/Faculty:</p> <p>1) Login to SPMS 3.0 with correct user ID and password.</p> <p>2) Head over to the designated section to view instructor wise PLO performance.</p> <p>2) Select a range of semester for the comparison to make by entering the upper bound and lower bound</p>	<p>Pen & Paper:</p> <p>May be used to take any PLO achievement related notes.</p>	<p>Computer:</p> <p>Used by the users to access SPMS 3.0 and view the comparison</p> <p>SPMS 3.0 Server:</p> <p>1) It handles user request to view PLO performance of individual faculty and compare with others.</p> <p>2) It generates necessary charts and comparison from the</p>	<p>Operating System:</p> <p>Used by the users to operate their computer.</p> <p>SPMS 3.0:</p> <p>Produce and shows the PLO performance of individual faculty and compare with others.</p>	<p>SPMS 3.0 DB:</p> <p>1) It stores login credentials for all the users.</p> <p>2) Used to store records for student PLO achievements and record for course-PLO, university, school and department.</p>	<p>Internet:</p> <p>Used to access the SPMS server.</p>

	<p>in a dropdown menu.</p> <p>VC:</p> <p>1) Select a faculty among all the faculties in the university by entering the faculties ID in a text box</p> <p>2) Press submit button to generate and display the charts.</p> <p>3) A chart will be displayed that displays the percentage of students out of all the students the selected faculty has instructed who achieved each of the PLOs.</p> <p>4) Upon selecting any of PLO, they will be able to compare with other faculties achievement for that PLO in the entire university.</p> <p>Dean of School:</p> <p>1) Select a faculty among all the faculties in his or her school by entering the faculty ID in a textbox.</p>		data retrieved from the database.			
--	---	--	-----------------------------------	--	--	--

	<p>2) Press submit button to generate and display the charts.</p> <p>3) A chart will be displayed that displays the percentage of students out of all the students the selected faculty has instructed who achieved each of the PLOs.</p> <p>4) Upon selecting any of PLO, they will be able to compare with other faculties achievement for that PLO in his or her school.</p> <p>Head of Department:</p> <p>1) Select a faculty among all the faculties in his or her department by entering the faculty ID in a text box.</p> <p>2) Press submit button to generate and display the charts.</p> <p>3) A chart will be displayed that displays the percentage of students out of all the students the selected</p>					
--	---	--	--	--	--	--

	<p>faculty has instructed who achieved each of the PLOs.</p> <p>4) Upon selecting any of PLO, they will be able to compare with other faculties achievement for that PLO in his or her department.</p>					
--	--	--	--	--	--	--

Table 2.3: Six Element Analysis (To Be)

BUSINESS PROCESS DIAGRAM (TO BE)

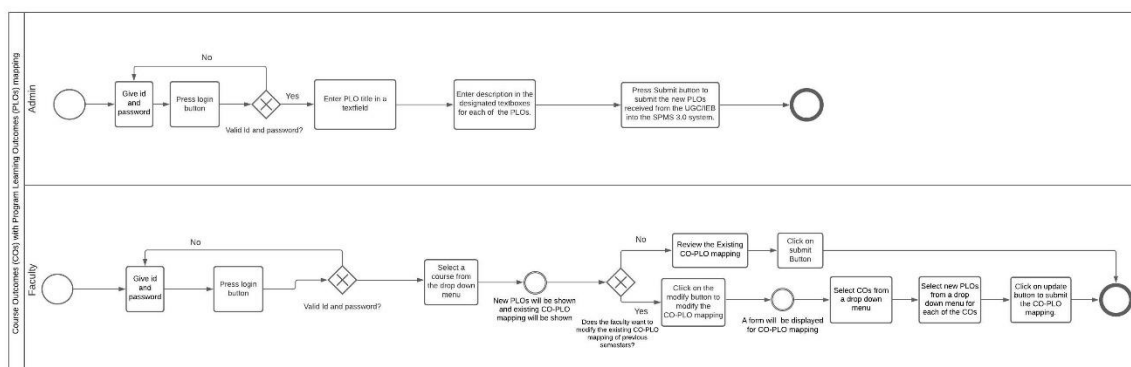


Figure 2.9: Course Outcomes (COs) with Program Learning Outcomes (PLOs) mapping

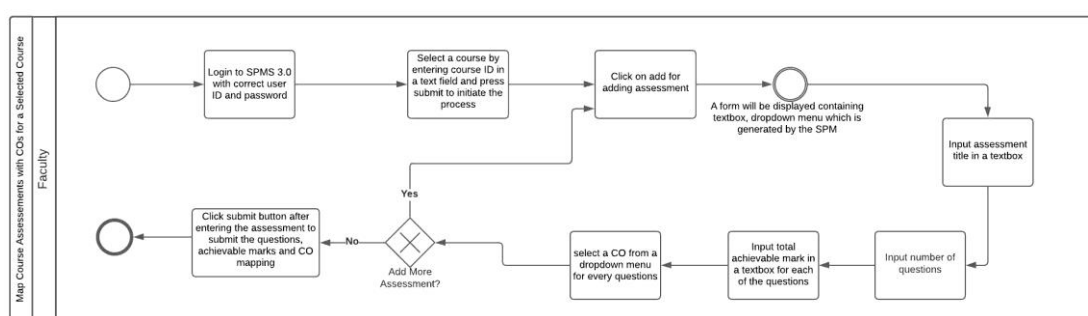


Figure 2.10: Map Course Assessments with COs for a Selected Course

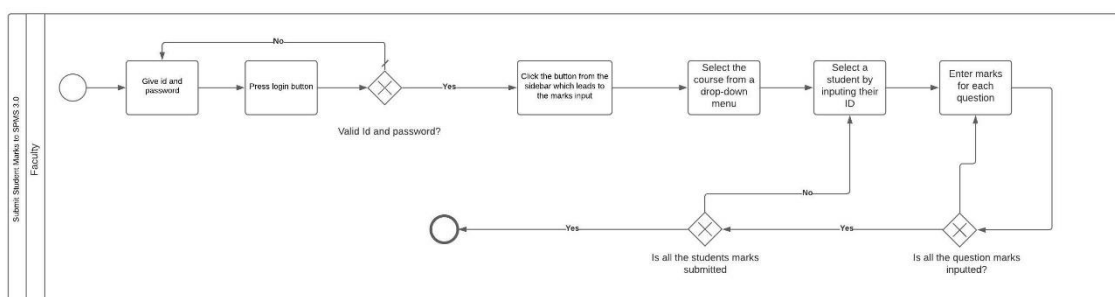


Figure 2.11: Submit Student Marks to SPMS 3.0

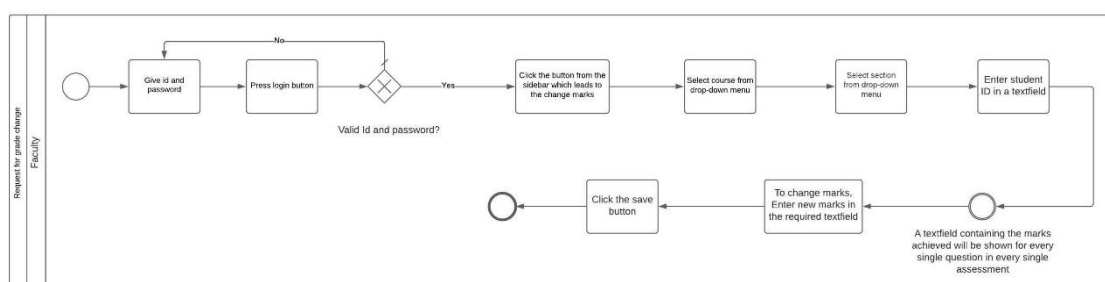


Figure 2.12: Request for grade change

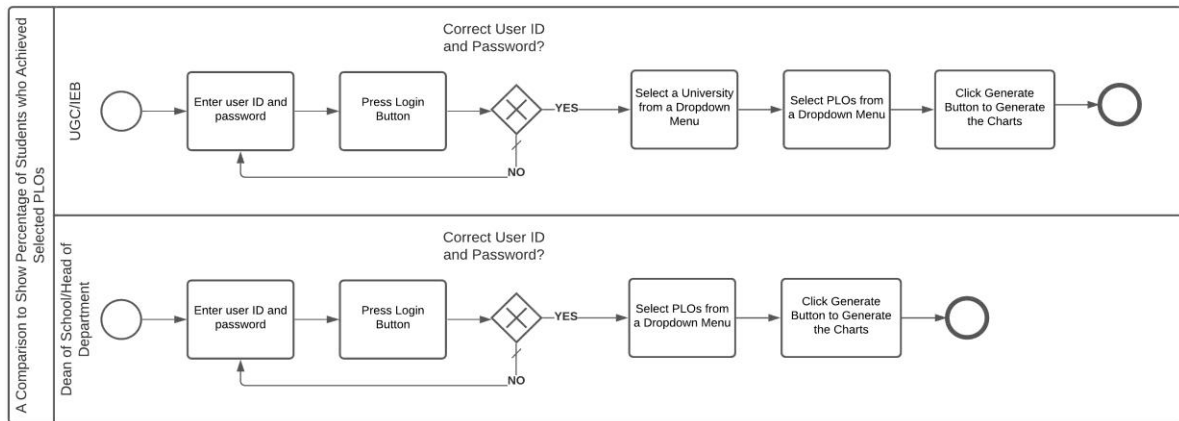


Figure 2.13: A Comparison to Show Percentage of Students who Achieved Selected PLOs

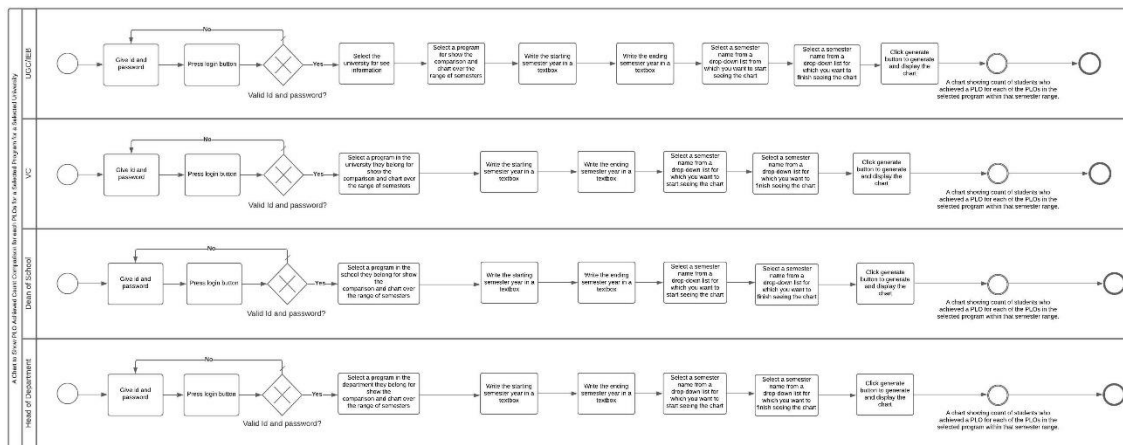


Figure 2.14: A Chart to Show PLO Achieved Count Comparison for each PLOs for A Selected Program for a Selected University

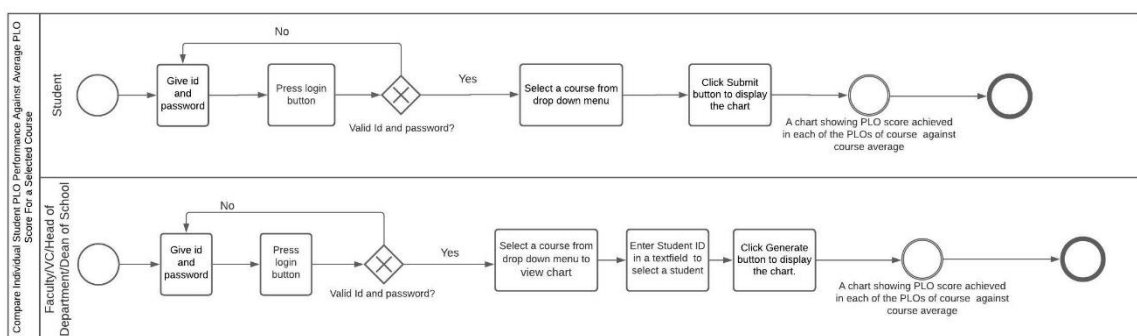


Figure 2.15: Compare Individual Student PLO Performance Against Average PLO Score for a Selected Course

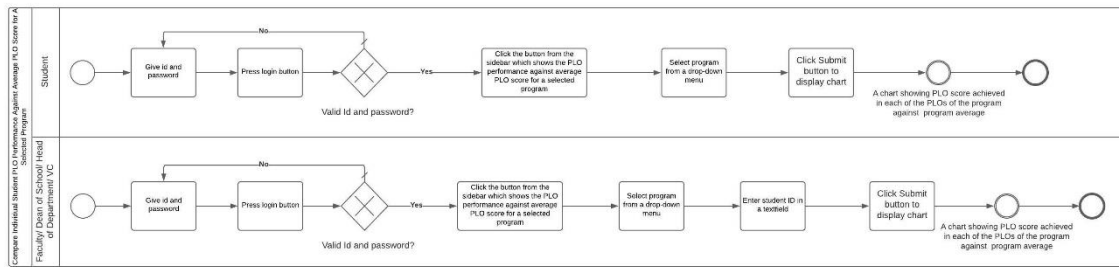


Figure 2.16: Compare Individual Student PLO Performance Against Average PLO Score for A Selected Program

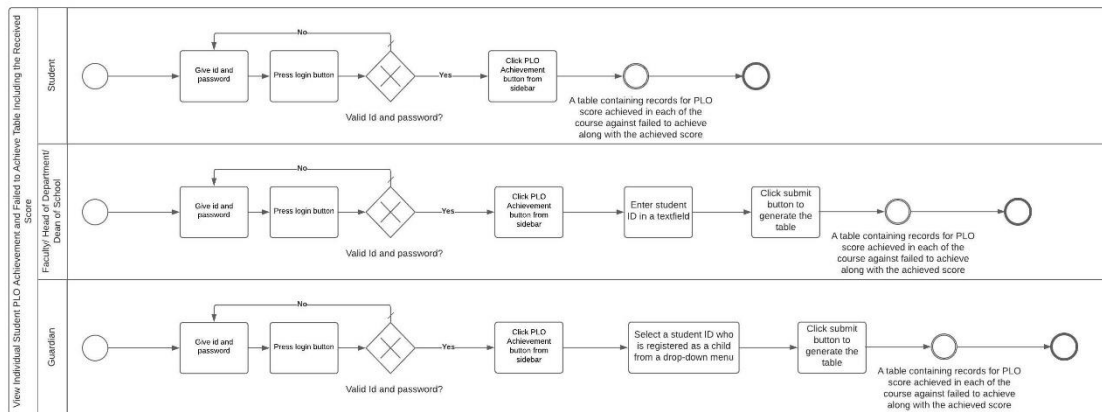


Figure 2.17: View Individual Student PLO Achievement and Failed to Achieve Table Including the Received Score

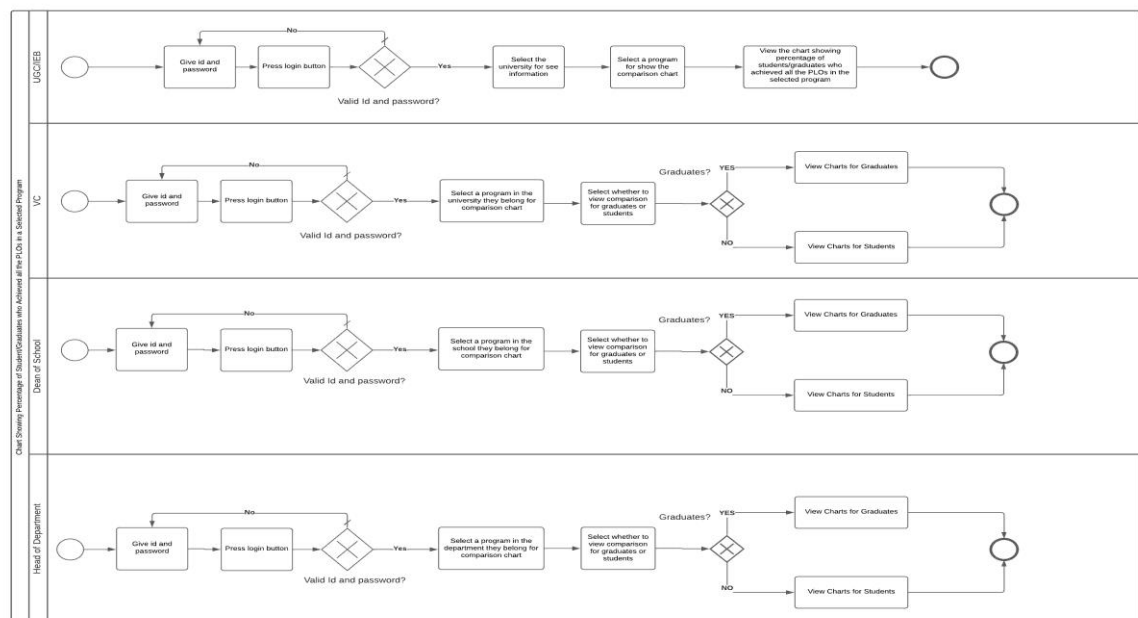


Figure 2.18: Chart Showing Percentage of Students/Graduates who Achieved all the PLOs in a Selected Program

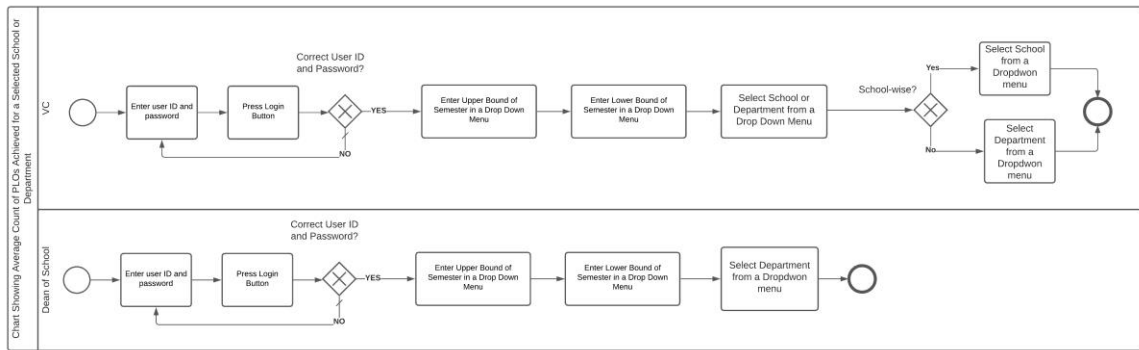


Figure 2.19: Chart Showing Average Count of PLOs Achieved for Selected Schools or Departments

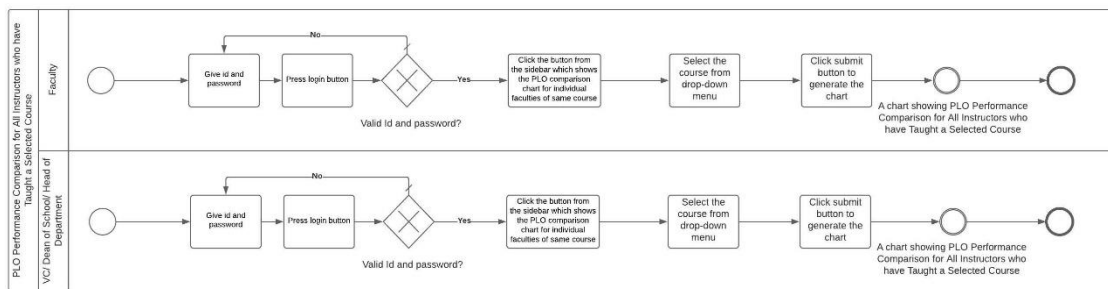


Figure 2.20: PLO Performance Comparison for All Instructors who have Taught a Selected Course

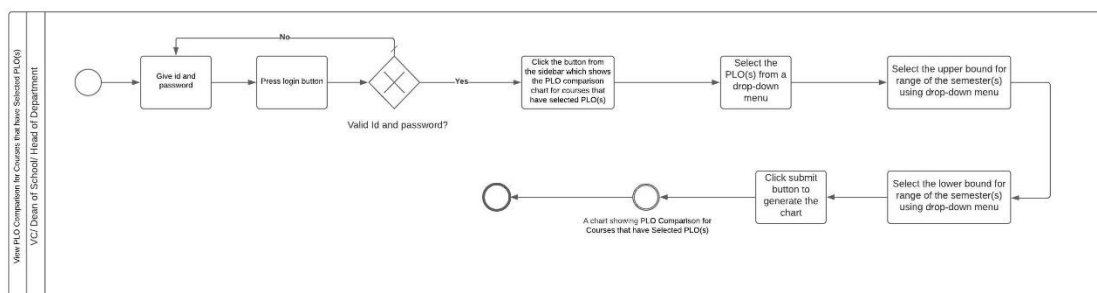


Figure 2.21: View PLO Comparison for Courses that have Selected PLO(s)

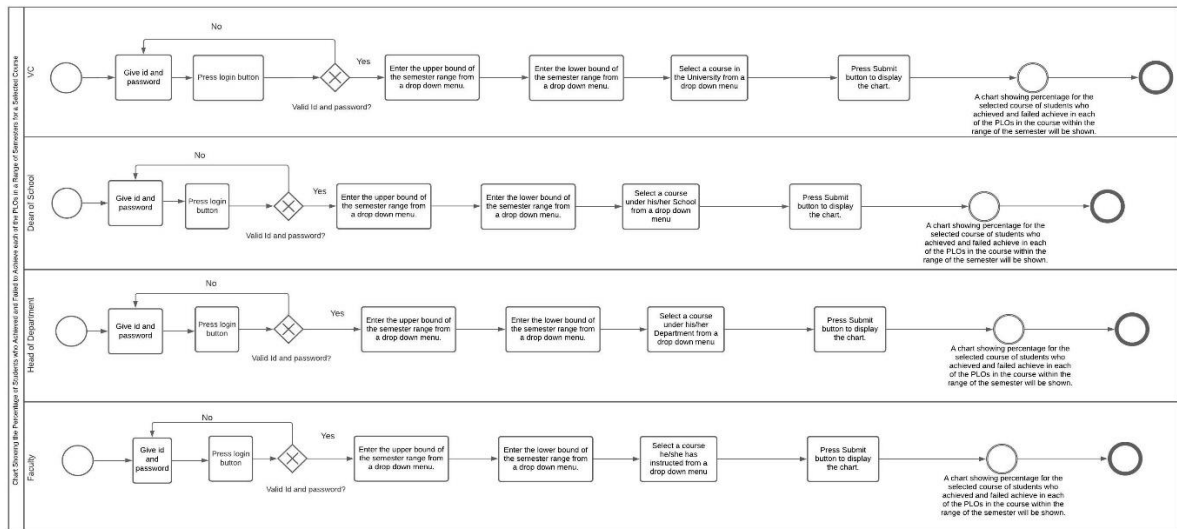


Figure 2.22: Chart Showing the Percentage of Students who Achieved and Failed to Achieve each of the PLOs in a Range of Semesters for a Selected Course

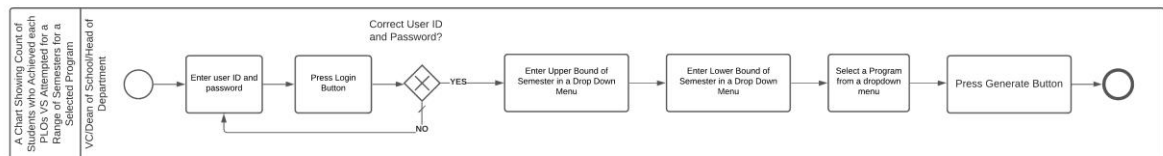


Figure 2.23: A Chart Showing Count of Students who Achieved each PLOs VS Attempted for a Range of Semesters for a Selected Program

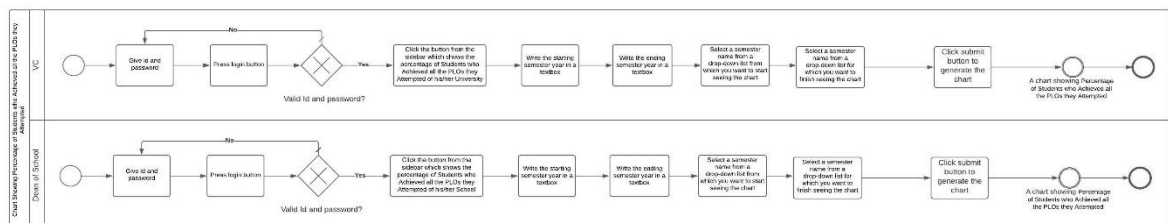


Figure 2.24: Chart Showing Percentage of Students who Achieved all the PLOs they Attempted

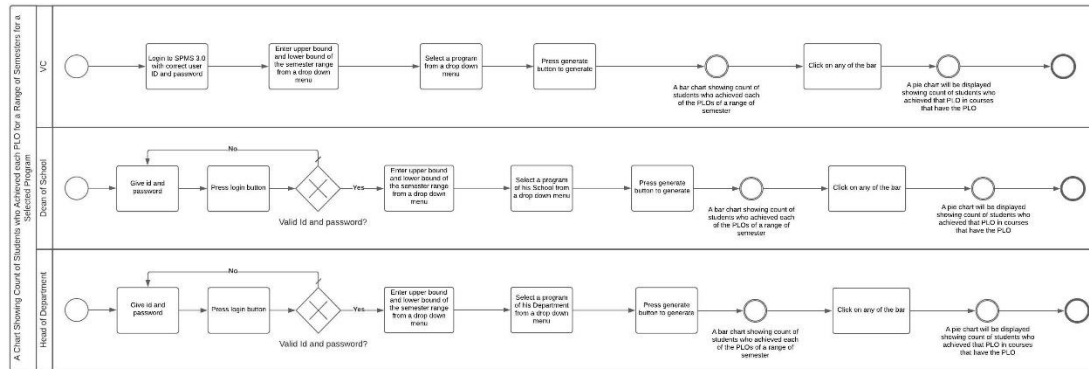


Figure 2.25: A Chart Showing Count of Students who Achieved each PLOs for a Range of Semesters for Selected Programs

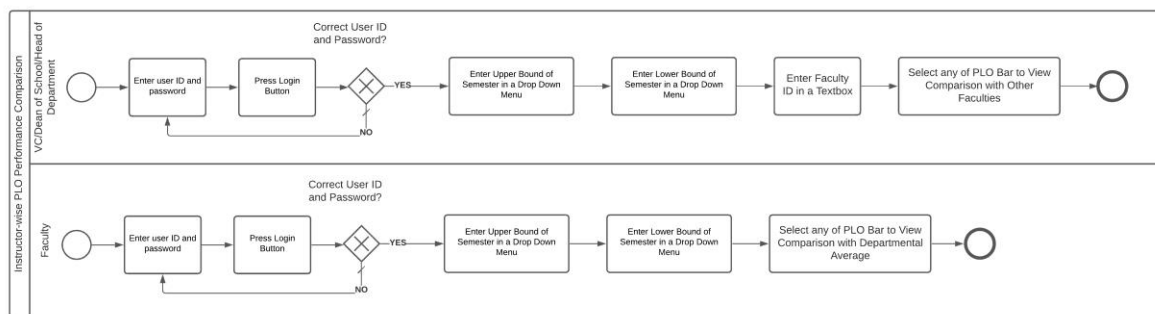


Figure 2.26: Instructor-wise PLO Performance Comparison

CHAPTER 3: LOGICAL SYSTEM DESIGN

BUSINESS RULE

In our system, university must be consisting of one or more schools. And, a school must belong to exactly one university. A school must be consisting of one or more departments. And, a department must belong to exactly one school. A department must offer one or more programs. And, a program must be offered by exactly one department.

A university has UniversityID and UniversityName. A school has SchoolID and SchoolName. A department has DepartmentID and DepartmentName. A program has ProgramID and ProgramName.

A program must be consisting of one or more PLOs. And, each PLO belongs to exactly one program. A PLO must be mapped with one or more COs. And a CO is mapped to exactly one PLO. A course is based on one or more COs. A CO is the basis of exactly one course. A program offers many courses. And, a course is offered under exactly one program. A course may have many prerequisite courses. A course may be prerequisite for many courses. A course may have many co-offered courses. A course has CourseID, CourseName and NumOfCredits.

A PLO has PloID, PloName and Details. A CO has CoID and CoNum. A course has CourseID, CourseName and NumOfCredits.

A department hires many faculties. And, a faculty must be hired by one or more departments. A student enrolls to exactly one program. And, a program is enrolled by many students.

A student has StudentID, StudentName, Gender, PhoneNumber, Address and DateofBirth. A faculty has FacultyID, FacultyName, FacultyContactNum, Email and address.

A course is taught by one or more sections. And, a section is teaches exactly one course. A section is taught by one faculty. And a faculty may teach in many sections. A student may register to a section. And, a section is registered by many students. A section takes many assessments. And, an assessment is taken by exactly one section. An assessment maps with exactly one CO. And, a CO is mapped to exactly many assessments. A student may submit many evaluations. And, an evaluation is submitted by exactly one student. An assessment contains many evaluations. And, each evaluation belongs to exactly one assessment.

A section has CourseID, SectionNum, SemesterName(Semester, Year) and MaximumCapacity. An assessment has AssessmentID, CourseID, SectionNum, SemesterName(Semester, Year), AssessmentType, TotalMarks and QuestionNumber. An evaluation has StudentID, AssessmentID and TotalMarksObtained.

ERD

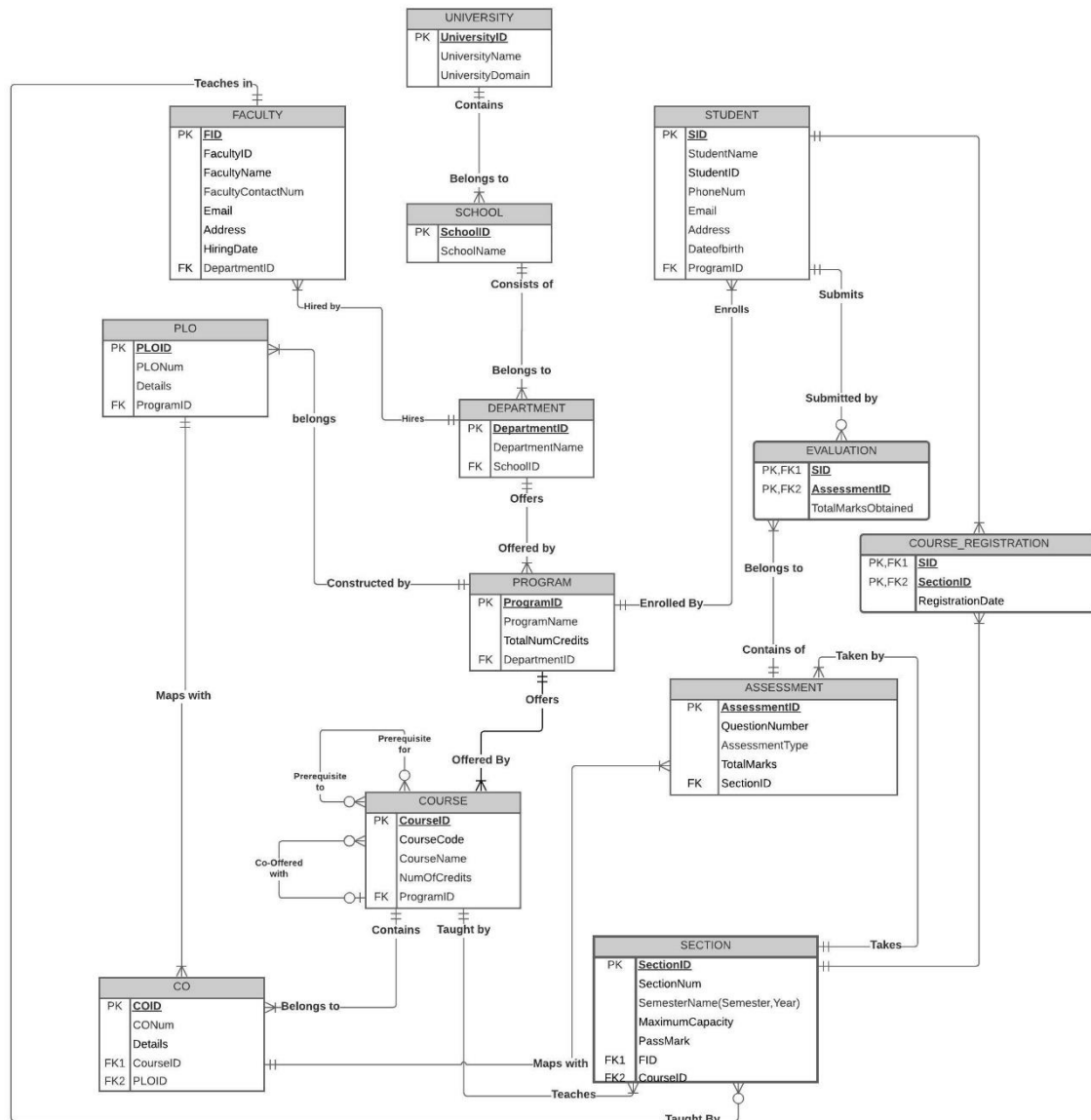


Figure 3.1: Enhanced ERD

RELATIONAL SCHEMA

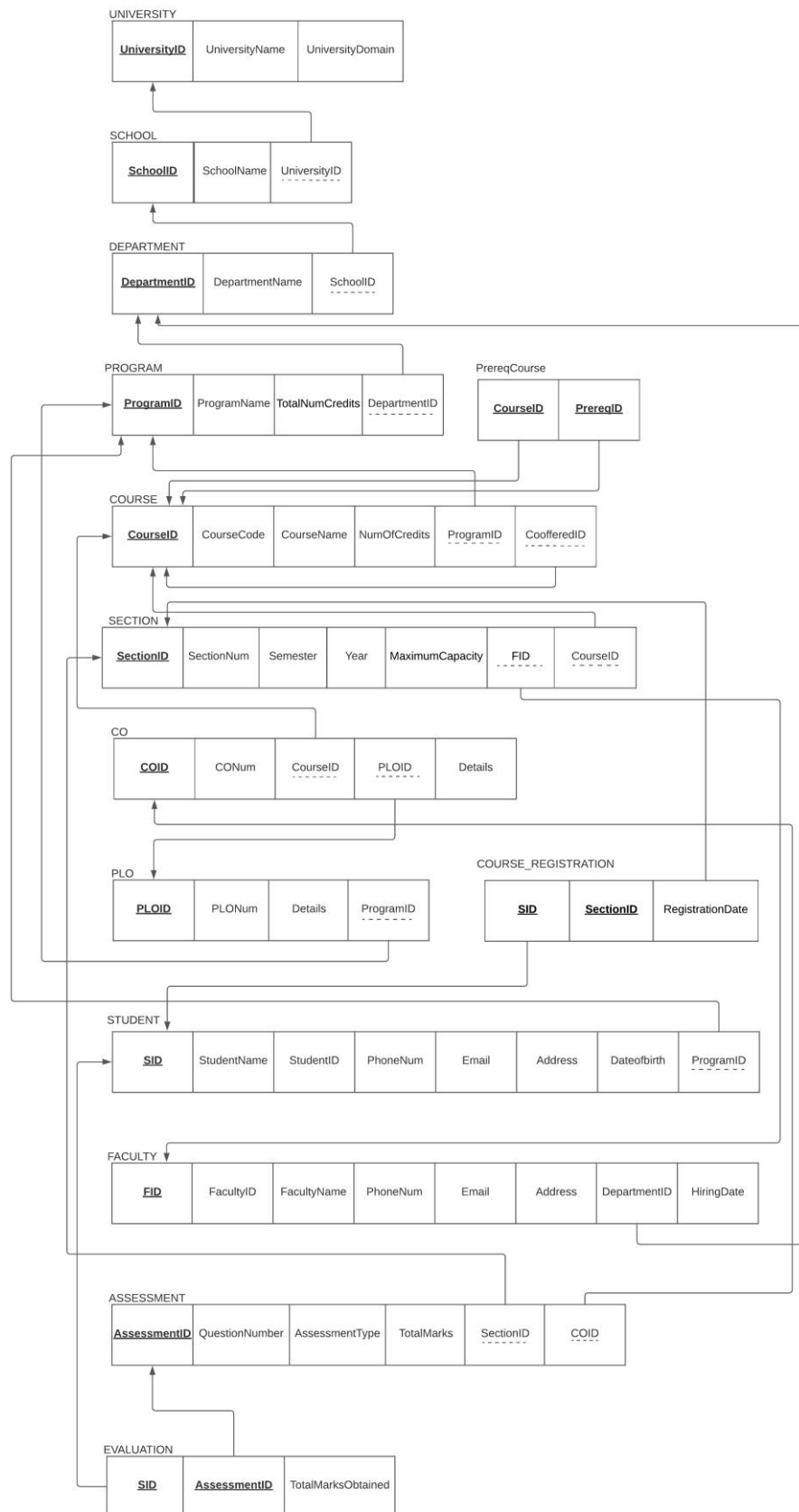


Figure 3.2: Relational Schema

NORMALIZATION

1NF: In 1NF, the schema must be arranged in a way where each relation has a primary key and in which there are no repeating groups and multivalued attributes.

In our schema, we see that in every relation, there has a primary key or composite primary key and there are no repeating groups and multivalued attributes. So, in our schema, all relations are in 1NF.

2NF: In 2NF, every relation of the schema must not have any non-key attributes that are partially dependent on the primary key. In our schema, we see that in every relation, no non-key is partially dependent on the primary key. All non-keys are fully dependent on the primary key or composite primary key. So, in our schema, all relations are in 2NF.

3NF: In 3NF, every relation of the schema must not have any transitive dependency. In our schema, we see that in every relation, there is no transitive dependency. So, in our schema, all relations are in 3NF.

BCNF: In BCNF, every relation of the schema, a non-key must not identify a candidate key or part of it. In every relation of our schema, there is no non-key that is identify any candidate key and part of it. So, in our schema, all relations are in BCNF.

DATA DICTIONARY

Department_T

Name	Data Type	Size	Remarks
nDepartmentID	INTEGER		This is the system generated primary key for the Department table. This must not be null.
cDepartmentName	VARCHAR	255	This is the name of the department. Example: "Computer Science and Engineering". This must not be nullable.
nSchoolID	INTEGER		This is the foreign key to the department table. This must not be nullable.

Program_T

Name	Data Type	Size	Remarks
nProgramID	INTEGER		This is the system generated primary key for the program table. For Example: 1. This must not be null.
cProgramName	VARCHAR	255	This is the name of the program. Example: "Bachelor of Science". This must not be nullable.
nTotalNumCredits	INTEGER		This is the number of credits for a program. For example: 143. This must not be nullable.
nDepartmentID	INTEGER		This is the foreign key to the department table. This must not be nullable.

PLO_T

Name	Data Type	Size	Remarks
nPloID	INTEGER		This is the system generated primary key to the Program Learning Outcome table. For example: 1. This must not be nullable.
cPloName	VARCHAR	6	This is the name of the PLO. For example: "PLO-01". This must not be null.
cDetails	VARCHAR	255	This is the details for PLO. For example: "An ability to select and apply the knowledge, skills, and modern tools of the computer science and engineering discipline". This must not be null.
nProgramID	INTEGER		This is the foreign key to the Program table. This must not be nullable.

School_T

Name	Data Type	Size	Remarks
nSchoolID	INTEGER		This is the system generated primary key for the School table. This must not be null.
cSchoolName	VARCHAR	255	This is the name of the School. Example: "Computer Science and Engineering". This must not be nullable.
nUniversityID	INTEGER		This is the foreign key to the University table. This must not be nullable.

Student_T

Name	Data Type	Size	Remarks
nSID	INTEGER		This is the system generated primary key for the Student table. This must not be nullable.
nStudentID	INTEGER		This is the organization ID the student. E.g.: 1821876. This must not be nullable.
cStudentName	VARCHAR	255	This is the name of the student. E.g.: "Hasibul Haque". This must not be nullable.
cPhoneNum	VARCHAR	14	This is the phone number of the student. E.g.: "+8801XXXXXXXXXX". This can be nullable.
cEmail	VARCHAR	50	This is the organizational email of the student. E.g.: "1821876@iub.edu.bd". This must not be nullable.
cAddress	VARCHAR	255	This is the address of the student. E.g.: "House 10, Road 7, Block B, Bashundhara RA". This can be nullable.
dDateofBirth	DATE	DD-MM-YYYY	This is the birth date of the student. E.g.: "23-06-2000". This can be nullable.
nProgramID	INTEGER		This is the foreign key to the Program table. This must not be nullable.

Assessment_T

Name	Data Type	Size	Remarks
nAssessmentID	INTEGER		This is the system generated primary key for the Assessment table. This must not be nullable.
nQuestionNumber	INTEGER		This is the question number for assessment. E.g.: 1, 2, 3... This must not be nullable.
cAssessmentType	VARCHAR	50	This is the name of the assessment. E.g.: "Mid". This must not be nullable.
nTotalMarks	INTEGER		This is the total achievable marks on the assessment. E.g.: 10. This must not be nullable.
nSectionID	INTEGER		This is the foreign key to the section table. E.g.: 1. This must not be nullable.
nCoID	INTEGER		This is the foreign key to the CO table. This must not be nullable.

University_T

Name	Data Type	Size	Remarks
nUniversityID	INTEGER		This is the system generated primary key for the university table. This must not be nullable.
cUniversityName	VARCHAR	255	This is the name of the university. E.g.: "Independent University, Bangladesh". This must not be nullable.
cUniversityDomain	VARCHAR	255	This is the university domain. E.g.: "iub.edu.bd". This is not nullable.

CO_T

Name	Data Type	Size	Remarks
nCoID	INTEGER		This is the system generated primary key for the CO (Course outline) table. This must not be nullable.
cCoName	VARCHAR	5	This is the CO name. E.g.: "CO-02", "CO-01", etc. This must not be nullable.
nCourseID	INTEGER		This is the foreign key to the course table. This must not be nullable.
nPloID	INTEGER		This is the foreign key to the PLO (Program Learning Outcome) table. This must not be nullable.
Details	VARCHAR	255	This is the details of the CO-PLO mapping. It may be null.

Section_T

Name	Data Type	Size	Remarks
nSectionID	INTEGER		This is the system generated primary key for the section table. E.g.: 1, 2, 3... This must not be nullable.
nSectionNum	INTEGER		This is the section number. E.g.: 1, 2, 3... This must not be nullable.
cSemester	VARCHAR	6	This is the semester name. E.g.: "Summer", "Spring", etc. This must not be nullable.
nYear	INTEGER		This is the year. E.g.: 2020. This must not be nullable.
nMaximumCapacity	INTEGER		This is the maximum student capacity for the section. E.g.: 40. This must not be nullable.
nFID	INTEGER		This is the foreign key to the Faculty table. This must not be nullable.
nCourseID	INTEGER		This is the foreign key to the course table. This must not be nullable.

Course_Registration_T

Name	Data Type	Size	Remarks
nSID	INTEGER		This is the part of the primary key and a foreign key to the student table. This must not be nullable.
nSectionID	INTEGER		This is the part of the primary and a foreign key to the section table. This must not be nullable.
dRegistrationDate	DATE	DD-MM-YYYY	This is the date of registration. E.g.: 19-01-2020. This must not be nullable.

Evaluation_T

Name	Data Type	Size	Remarks
nSID	INTEGER		This is both a part of the composite primary key and a foreign key that references a student. It must not be nullable.
nAssessmentID	INTEGER		This is both a part of the composite primary key and a foreign key that references a assessment. This must not be nullable.
nTotalObtainedMark	INTEGER		This is the mark a student has obtained in the assessment. This must not be nullable.

Course_T

Name	Data Type	Size	Remarks
nCourseID	INTEGER		This is a system generated primary key for courses.
nCourseCode	VARCHAR	8	This is the course title for a course. It is not nullable. For instance, CSE303 or MATH1115.
CCourseName	VARCHAR	255	This is the name of the course. It is not nullable. For instance, "Database Management System".
nNumOfCredits	INTEGER		This is the number of credits of a course. It must not be null.
nProgramID	INTEGER		This is the foreign key that references the primary key of a program. It must not be null.
CoofferedID	INTEGER		This is the foreign key that points to the CourseID of the Course_T. This must not be null.

Faculty_T

Name	Data Type	Size	Remarks
nFID	INTEGER		This is the system generated primary key for the faculties. This is not nullable.
nFacultyID	INTEGER		This is the faculty ID in his or her University. This is not null.
cFacultyName	VARCHAR	255	This is the name of the faculty. E.g.: "MsSadita Ahmed". This must not be nullable.
cPhoneNum	VARCHAR	14	This is the phone number of the faculty. E.g.: "+8801XXXXXXXXXX". This can be nullable.
cEmail	VARCHAR	50	This is the organizational email of the faculty. E.g.: "sadita.ahsets@iub.edu.bd". This must not be nullable.
cAddress	VARCHAR	255	This is the address of the faculty. It is optional.
nDepartmentID	INTEGER		This is a foreign key that references the department he or she belongs to. It must not be null.
HiringDate	DATE		This could be null.

PrereqCourse_T

Name	Data Type	Size	Remarks
CourseID	INTEGER		This is the part of the primary key and a foreign key to the course table. This must not be nullable.
PrereqID	INTEGER		This is the part of the primary key and a foreign key to the course table. This must not be nullable.

Cooffered_T

Name	Data Type	Size	Remarks
CourseID	INTEGER		This is the part of the primary key and a foreign key to the course table. This must not be nullable.
CoofferedID	INTEGER		This is the part of the primary key and a foreign key to the course table. This must not be nullable.

CHAPTER 4: PHYSICAL SYSTEM DESIGN

INPUT FORM:

```
[HttpPost("/faculty/iad/")]
```

```
public IActionResult InputAssessmentDetails(int selectedCourse)
{
    InputAssessmentDetailsViewModel viewModel = new InputAssessmentDetailsViewModel()
    {
        TopbarViewModel = new TopbarViewModel()
        {
            Name = activeFaculty.FullName,
            ID = activeFaculty.FacultyID
        },
        Courses = ds.courses.Where(c => c.Program.Department == activeFaculty.Department
        && c.CoofferedCourse == null).ToList()
    };

    viewModel.SelectedCourse = ds.courses.Find(c => c.CourseID == selectedCourse);

    var sections = ds.sections.Where(s => s.Course == viewModel.SelectedCourse &&
    s.Semester.CompareTo(activeFaculty.Department.School.University.CurrentSemester) == 0);

    viewModel.Assessments = ds.assessments.Where(a =>
    sections.Contains(a.Section)).ToList();

    viewModel.CourseList = ds.courses.Where(c => c == viewModel.SelectedCourse ||
    c.CoofferedCourse == viewModel.SelectedCourse).ToList();

    return View(viewModel);
}
```

```
internal void AddAssessment(Assessment assessment)
```

```
{
    assessments.Add(assessment);

    connection = new
    MySqlConnection("server=localhost;database=spmsdb;userid=spms;password=");
    connection.Open();
}
```

```

MySQLCommand cmd = new MySQLCommand(
    "Insert into Assessment_T (AssessmentID, QuestionNumber, AssessmentType,
    TotalMarks, SectionID, CoID) values" +
    $"({assessment.AssessmentID}, {assessment.QuestionNumber}, " +
    $"'{assessment.AssessmentType}', {assessment.TotalMark},
    {assessment.Section.SectionID}," +
    $"'{assessment.CourseOutcome.CoID});", connection);
cmd.ExecuteNonQuery();
connection.Close();
}

```

Select a course: Selelct a Course Submit

Type	Number	Mark	Section	Program	CoID		
Mid Term Exam	1	10	1	Electrical and Electronic Engineering	CO-01	Modify	Delete
Mid Term Exam	2	10	1	Electrical and Electronic Engineering	CO-02	Modify	Delete
Final Exam	3	10	1	Electrical and Electronic Engineering	CO-03	Modify	Delete
Final Exam	4	10	1	Electrical and Electronic Engineering	CO-04	Modify	Delete
Mid Term Exam	1	10	1	Electrical and Telecommunication Engineering	CO-01	Modify	Delete
Mid Term Exam	2	10	1	Electrical and Telecommunication Engineering	CO-02	Modify	Delete
Final Exam	3	10	1	Electrical and Telecommunication Engineering	CO-03	Modify	Delete
Final Exam	4	10	1	Electrical and Telecommunication Engineering	CO-04	Modify	Delete
Mid Term Exam	1	10	2	Electrical and Electronic Engineering	CO-01	Modify	Delete
Mid Term Exam	2	10	2	Electrical and Electronic Engineering	CO-02	Modify	Delete
Final Exam	3	10	2	Electrical and Electronic Engineering	CO-03	Modify	Delete
Final Exam	4	10	2	Electrical and Electronic Engineering	CO-04	Modify	Delete
Mid Term Exam	1	10	2	Electrical and Telecommunication Engineering	CO-01	Modify	Delete
Mid Term Exam	2	10	2	Electrical and Telecommunication Engineering	CO-02	Modify	Delete
Final Exam	3	10	2	Electrical and Telecommunication Engineering	CO-03	Modify	Delete
Final Exam	4	10	2	Electrical and Telecommunication Engineering	CO-04	Modify	Delete

Selelct a Course Add New Assessment

Question Number: 0 Question Type: Enter Question Type

Total Mark: 0 Section: Selelct a Section CO Name: Selelct a CO

Submit

OUTPUT FORM:

```
[HttpPost("/faculty/cpafc")]
```

```
public IActionResult ComparisonPloAchievedFailedSelectCourses([FromBody]
ComparisonPloAchievedFailedSelectCoursesViewModel viewModel) // 6 continued...
```

```
{
    var evaluations = ds.evaluations.Where(ev =>
viewModel.SelectedCourses.Contains(ev.Assessment.Section.Course.CourseID))
        .Where(ev =>
viewModel.SelectedSemesters.Contains(ev.Assessment.Section.Semester));

    var evaluationsPloGroups = evaluations.GroupBy(ev =>
ev.Assessment.CourseOutcome.PLO.PloName);

    var ploNameList = new List<string>();
    var achievedList = new List<float>();
    var failedList = new List<float>();

    foreach (var evGroup in evaluationsPloGroups)
    {
        ploNameList.Add(evGroup.Key);

        var passedCount = 0;

        foreach (var ev in evGroup)
        {
            var percent = ev.TotalObtainedMark / ev.Assessment.TotalMark * 100;

            if (percent > ev.Assessment.Section.PassMark)
            {
                passedCount++;
            }
        }

        var passPercent = (float)passedCount / evGroup.Count() * 100;
        achievedList.Add(passPercent);
        failedList.Add(100 - passPercent);
    }

    var myData = new {label = ploNameList, passData = achievedList, failData = failedList};

    return Json(myData);
}
```

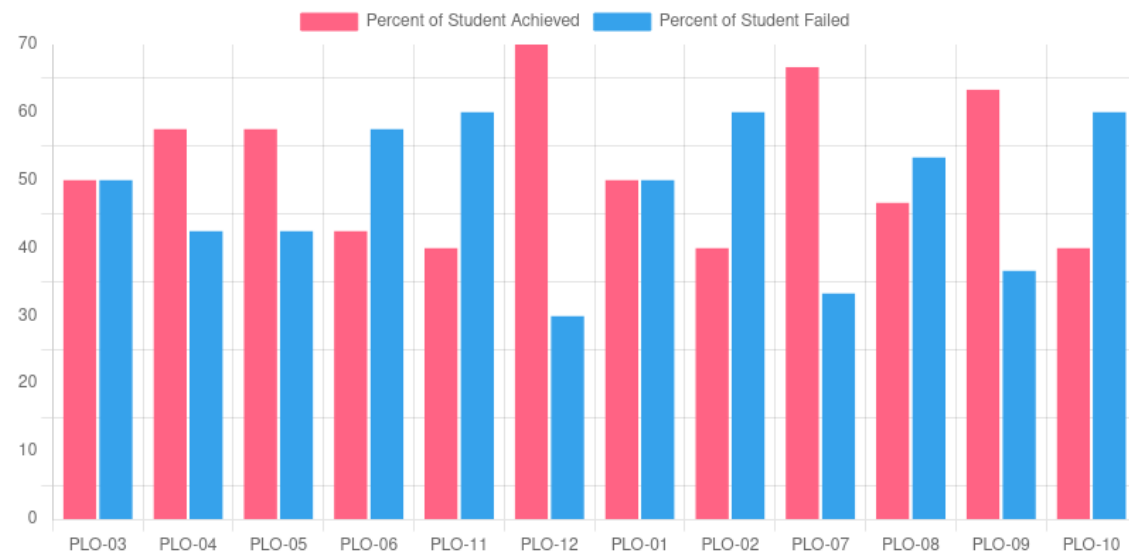
}

Select Course(s)

Selected Courses: PHY111, EEE211

Select Semester(s)

Selected Semesters: Summer 2001, Summer 2002, Summer 2003

```
[HttpPost("/ugc/upasp/")]
```

```
public IActionResult UniversityPloAchievementSelectProgram([FromBody]
UniversityPloAchievementSelectProgram viewModel)
{
    viewModel.TopbarViewModel = new TopbarViewModel()
    {
        Name = ActiveUgcieb.FullName,
        ID = ActiveUgcieb.UgciebID
    };
    viewModel.Programs = ds.programs;
    var program = ds.programs.Find(p => p.ProgramID == viewModel.SelectedProgram);
    var programEval = ds.evaluations.Where(ev =>
ev.Assessment.CourseOutcome.PLO.Program == program);
    var evalPloGroups = programEval.GroupBy(ev =>
ev.Assessment.CourseOutcome.PLO.PloName);
    List<string> labels = new List<string>();
    List<int> passCount = new List<int>();
    foreach (var evalPloGroup in evalPloGroups)
    {
        labels.Add(evalPloGroup.Key);
        int passed = 0;
        foreach (var eval in evalPloGroup)
        {
            float percent = (float)eval.TotalObtainedMark / eval.Assessment.TotalMark * 100;
            if (percent >= eval.Assessment.Section.PassMark)
            {
                passed++;
            }
        }
        passCount.Add(passed);
    }
}
```



```
Dataset dset = new Dataset()

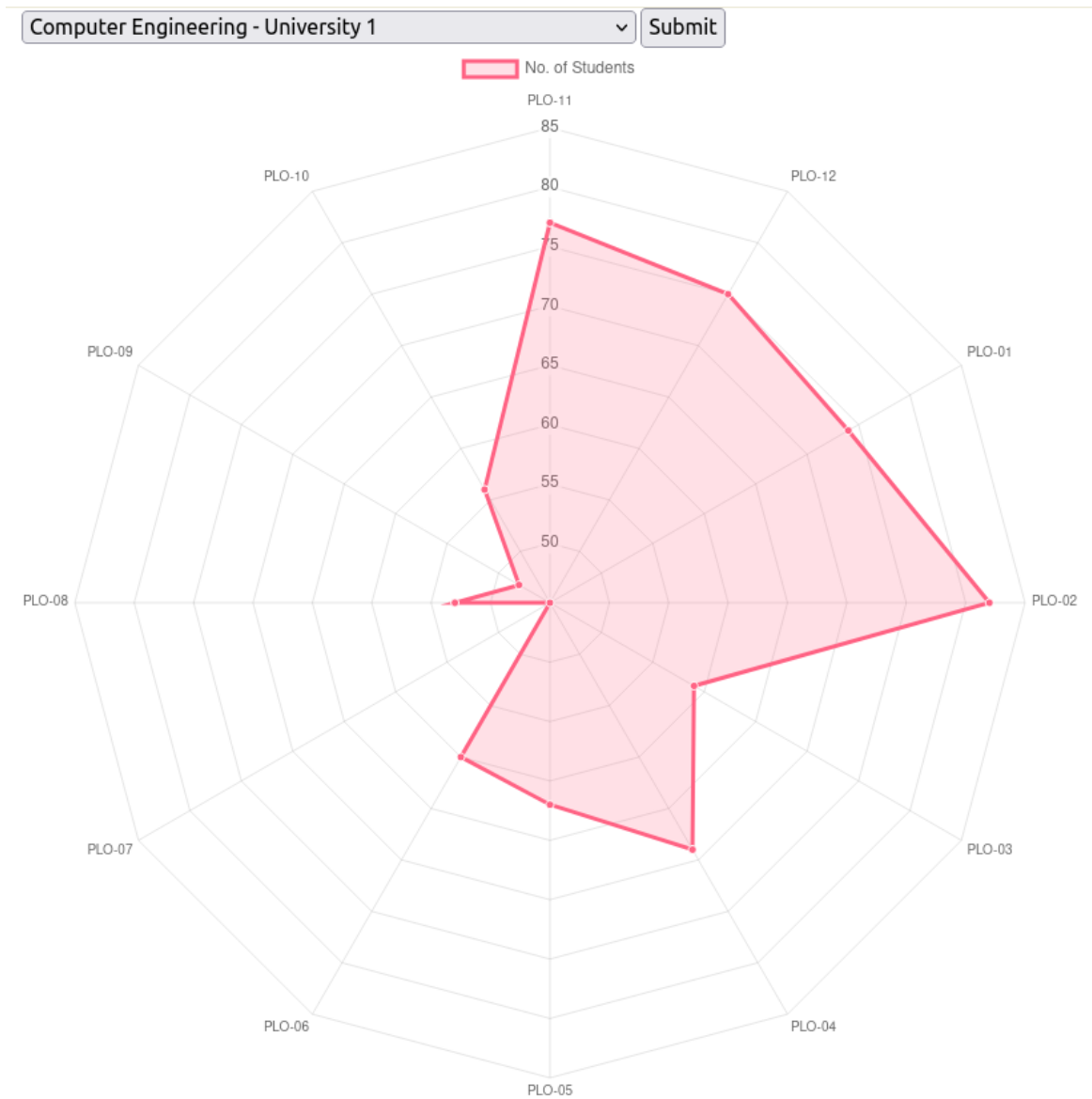
{
    Data = passCount,
    Label = "No. of Students",
    BackgroundColor = "rgba(255, 99, 132, 0.2)",
    BorderColor = "rgb(255, 99, 132)",
    PointBackgroundColor = "rgb(255, 99, 132)",
    PointBorderColor = "#fff",
    PointHoverBackgroundColor = "#fff",
    PointHoverBorderColor = "rgb(255, 99, 132)"
};

ViewModels.Data data = new ViewModels.Data()

{
    Labels = labels,
    Datasets = new List<Dataset>() { dset }
};

viewModel.Data = data;

return Json(viewModel);
}
```



```
[HttpGet("/faculty/ispscc/{studentID}/{courseID}")]
```

```
public IActionResult IndividualStudentPloScoreComparisonCourse(int studentID, int courseID)
```

```
{
```

```
    var student = ds.students.Find(s => s.StudentID == studentID && s.Program.Department == activeFaculty.Department);
```

```
    var course = ds.courses.Find(c => c.CourseID == courseID);
```

```
    if (student == null) return NotFound();
```

```
    var courseRegistrations = ds.courseRegistrations.Where(cr => cr.Student == student && cr.Section.Course == course).ToList();
```

```
    if (courseRegistrations.Count <= 0) return NotFound();
```

```
    var sections = courseRegistrations.Select((cr, idx) => cr.Section).ToList();
```

```
    var section = sections.First();
```

```
    for (int i = 1; i < sections.Count; i++)
```

```
    {
```

```
        if (section.Semester.CompareTo(sections[i].Semester) > 0)
```

```
        {
```

```
            section = sections[i];
```

```
        }
```

```
    }
```

```
    var semester = section.Semester;
```

```
    var studentEvaluation = ds.evaluations.Where(e => e.Assessment.Section == section && e.Student == student);
```

```

var studentEvalGroup = studentEvaluation.GroupBy(se =>
se.Assessment.CourseOutcome.PLO.PloName);

```

```

List<string> studentPloList = new List<string>();

```

```

List<float> studentScoreList = new List<float>();

```

```

foreach (var evalGroup in studentEvalGroup)

```

```

{

```

```

    studentPloList.Add(evalGroup.Key);

```

```

    float score = 0;

```

```

    foreach (var eval in evalGroup)

```

```

    {

```

```

        score += eval.TotalObtainedMark;

```

```

    }

```

```

    studentScoreList.Add(score);

```

```

}

```

```

var courseEval = ds.evaluations.Where(ev =>
ev.Assessment.Section.Semester.Equals(semester) && ev.Assessment.Section.Course == course
&& ev.Assessment.CourseOutcome.PLO.Program == student.Program);

```

```

var courseEvalGroupbyPlo = courseEval.GroupBy(ce =>
ce.Assessment.CourseOutcome.PLO.PloName);

```

```

var coursePloList = new List<string>();

```

```

var courseAvgScoreList = new List<float>();

```

```

foreach (var evalGroup in courseEvalGroupbyPlo)

```

```

{

```

```

coursePloList.Add(evalGroup.Key);

var evalGroupbySt = evalGroup.GroupBy(eg => eg.Student);
var stCount = evalGroupbySt.Count();

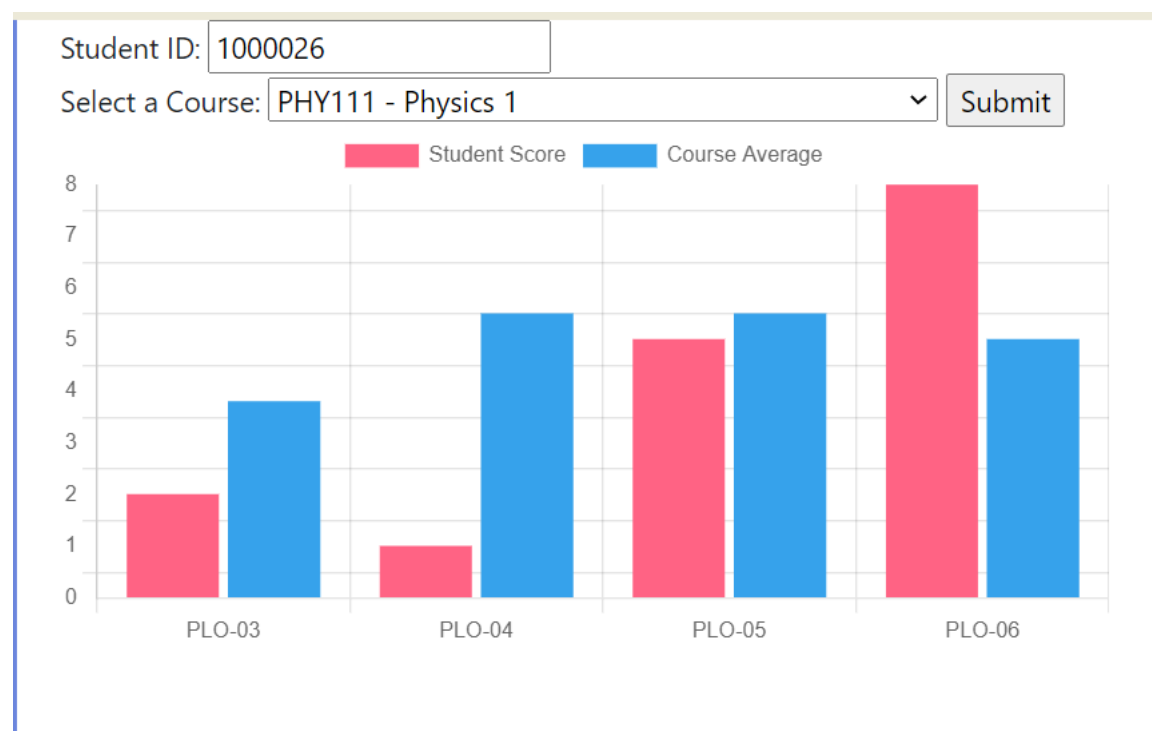
float score = 0;

foreach (var eval in evalGroup)
{
    score += eval.TotalObtainedMark;
}

courseAvgScoreList.Add(score / stCount);
}

return Json(new { StData = studentScoreList, StLabel = studentPloList, CourseData =
courseAvgScoreList });
}

```



```
[HttpGet("/faculty/ispscp/{studentID}/{programID}")]
```

```
public IActionResult IndividualStudentPloScoreComparisonProgram(int studentID, int
programID)
```

```
{
```

```
    var student = ds.students.Find(s => s.StudentID == studentID);
```

```
    if (student == null) return NotFound();
```

```
    var program = ds.programs.Find(p => p.ProgramID == programID);
```

```
    var programPlos = ds.plos.Where(plo => plo.Program == program);
```

```
    var evaluations = ds.evaluations.Where(ev =>
ev.Assessment.CourseOutcome.PLO.Program == program);
```

```
    List<float> programScores = new List<float>();
```

```
    List<float> studentScores = new List<float>();
```

```
    foreach (var plo in programPlos)
```

```
    {
```

```
        var ploScoreProgram = evaluations.Where(ev => ev.Assessment.CourseOutcome.PLO
== plo);
```

```
        var stCount = ploScoreProgram.GroupBy(psp => psp.Student).Count();
```

```
        // var ploScoreStudent = ploScoreProgram.Where(ev => ev.Student == student);
```

```
        float programScore = 0;
```

```
        float studentScore = 0;
```

```
        foreach (var p in ploScoreProgram)
```

```
        {
```

```
            programScore += p.TotalObtainedMark;
```

```

        if (p.Student == student)
        {
            studentScore += p.TotalObtainedMark;
        }
    }

    programScores.Add(programScore / stCount);

    studentScores.Add(studentScore);
}

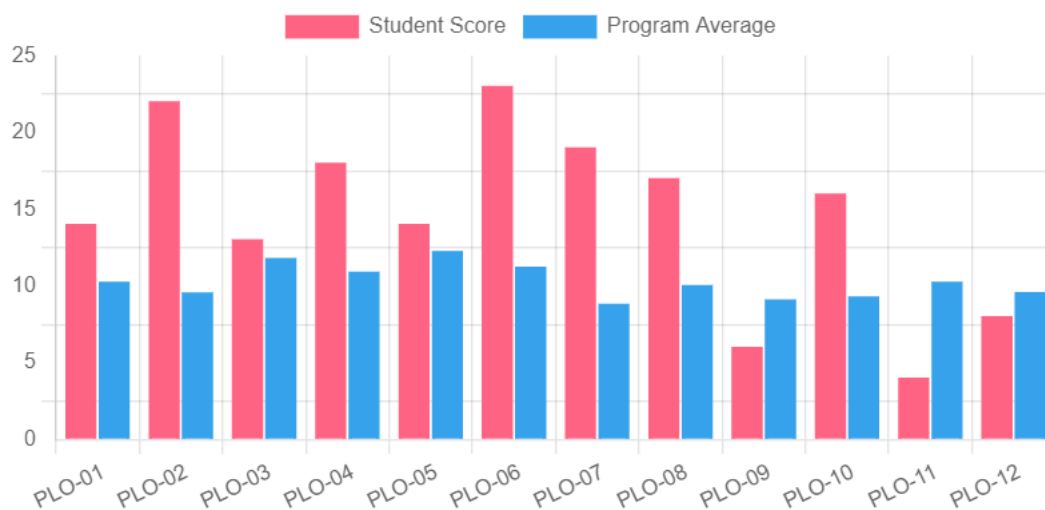
var data = new { PloList = programPlos.Select(p => p.PloName), StudentScores =
studentScores, ProgramScores = programScores };

return Json(data);
}

```

Student ID:

Select a Program:



```
[HttpGet("/faculty/pat/{studentID}")]
```

```
public IActionResult PloAchievementTable(int studentID) // 3 continued...
{
    var student = ds.students.Find(s => s.StudentID == studentID);

    if (student == null) return Json(null);

    var _data = ds.PloAchievementTableData(student);

    if (_data.Count <= 0) return Json(null);

    var plos = ds.plos.Where(o => o.Program == student.Program);

    if (plos.Count() <= 0) return Json(null);

    var mydata = new { studentName = student.FullName, ploList = plos, data = _data };

    return Json(mydata);
}
```


Student PLO Achievement Table

Enter a Student ID

Name : Sadie Chandler

ID : 1000026

Course Code	PLO-01	PLO-02	PLO-03	PLO-04	PLO-05	PLO-06	PLO-07	PLO-08	PLO-09	PLO-10	PLO-11	PLO-12
PHY111	N/A	N/A	20	10	50	80	N/A	N/A	N/A	N/A	N/A	N/A
PHY121	N/A	N/A	N/A	N/A	N/A	N/A	50	30	0	50	N/A	N/A
CSC121	10	20	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	10	10
EEE131	N/A	N/A	80	60	30	70	N/A	N/A	N/A	N/A	N/A	N/A
EEE132	N/A	N/A	N/A	N/A	N/A	N/A	50	90	30	10	N/A	N/A
EEE211	80	100	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	20	40
EEE221	N/A	N/A	10	10	50	30	N/A	N/A	N/A	N/A	N/A	N/A
EEE225	N/A	N/A	N/A	N/A	N/A	N/A	90	50	30	100	N/A	N/A
EEE231	50	100	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	10	30
EEE232	N/A	N/A	20	100	10	50	N/A	N/A	N/A	N/A	N/A	N/A

```
[HttpGet("/faculty/ippsc/{selectedCourse}/{startSemester}/{startYear}/{endSemester}/{endYear}")]
```

```
public IActionResult InstructorwisePLOPerformanceSelectCourses(int selectedCourse, int
startSemester, int startYear, int endSemester, int endYear) // 4 continued
```

```
{
    var course = ds.courses.Find(c => c.CourseID == selectedCourse);

    var start = new Semester(startSemester, startYear);

    var end = new Semester(endSemester, endYear);

    var sections = ds.sections.Where(s => (s.Course == course || s.Course ==
course.CoofferedCourse) && s.Semester.CompareTo(start) >= 0 &&
s.Semester.CompareTo(end) <= 0)

        .GroupBy(s => s.Faculty);

    // Dictionary<Faculty, List<float>> scores = new Dictionary<Faculty, List<float>>();

    var scores = new ArrayList();

    var plos = ds.plos.Where(plo => plo.Program == course.Program);

    foreach (var facultySection in sections)
    {
        List<float> _scores = new List<float>();

        foreach (var plo in plos)
        {
            float count = 0;

            int total = 0;

            foreach (var section in facultySection)
            {
                var evaluations = ds.evaluations.Where(ev => ev.Assessment.Section == section
&& ev.Assessment.CourseOutcome.PLO == plo);

                total += evaluations.Count();
            }
        }
    }
}
```

```

        foreach (var eval in evaluations)
        {
            var percent = eval.TotalObtainedMark / eval.Assessment.TotalMark * 100;

            count = percent > eval.Assessment.Section.PassMark ? count + 1 : count;
        }
    }

    if (total > 0) _scores.Add(count / total * 100);
    else _scores.Add(0);
}

var d = new
{
    faculty = facultySection.Key,
    data = _scores
};

scores.Add(d);
}

var myData = new
{
    Course = course,
    Data = scores,
    ploList = plos.Select(p => p.PloName)
};

return Json(myData);
}

```



```
[HttpPost("/faculty/pccsp/{startSemester}/{startYear}/{endSemester}/{endYear}")]
```

```
    public IActionResult PloComparisonCourseWithSelectPlos([FromBody]
    PloComparisonCourseWithSelectPlosViewModel viewModel, int startSemester, int startYear, int
    endSemester, int endYear) // 5 continued
```

```
{
```

```
    var start = new Semester(startSemester, startYear);
```

```
    var end = new Semester(endSemester, endYear);
```

```
        // var courses = ds.courses.Where(c =>
    viewModel.SelectedCoursesID.Contains(c.CourseID)).ToList();
```

```
        // var plos = ds.plos.Where(plo => plo.Program.Department == activeFaculty.Department
    && viewModel.SelectedPlosName.Contains(plo.PloName)).ToList();
```

```
    List<float> scores = new List<float>();
```

```
    List<string> ploNames = new List<string>();
```

```
    List<Course> courses = new List<Course>();
```

```
    foreach (var course in ds.courses)
```

```
{
```

```
    if (viewModel.SelectedCoursesID.Contains(course.CourseID))
```

```
{
```

```
        bool available = true;
```

```
        var plos = ds.cos.Where(co => co.Course == course).Select(co =>
    co.PLO.PloName);
```

```
        foreach (var ploName in viewModel.SelectedPlosName)
```

```
{
```

```
            if (!plos.Contains(ploName))
```

```

        available = false;
    }

    if (available) courses.Add(course);
}
}

var evaluations = ds.evaluations.Where(ev =>
courses.Contains(ev.Assessment.Section.Course)

                                &&
viewModel.SelectedPlosName.Contains(ev.Assessment.CourseOutcome.PLO.PloName)

                                && ev.Assessment.Section.Semester.CompareTo(start) >= 0

                                && ev.Assessment.Section.Semester.CompareTo(end) <= 0)
                                .GroupBy(ev => ev.Assessment.CourseOutcome.PLO.PloName);

foreach (var evg in evaluations)
{
    ploNames.Add(evg.Key);

    int count = 0;

    foreach (var ev in evg)
    {
        var percent = ev.TotalObtainedMark / ev.Assessment.TotalMark * 100;

        if (percent >= ev.Assessment.Section.PassMark)
        {
            count++;
        }
    }

    scores.Add((float)count / evg.Count() * 100);
}

```

```
var myData = new { labels = ploNames, data = scores };  
  
return Json(myData);  
}
```

Select Starting Semester:

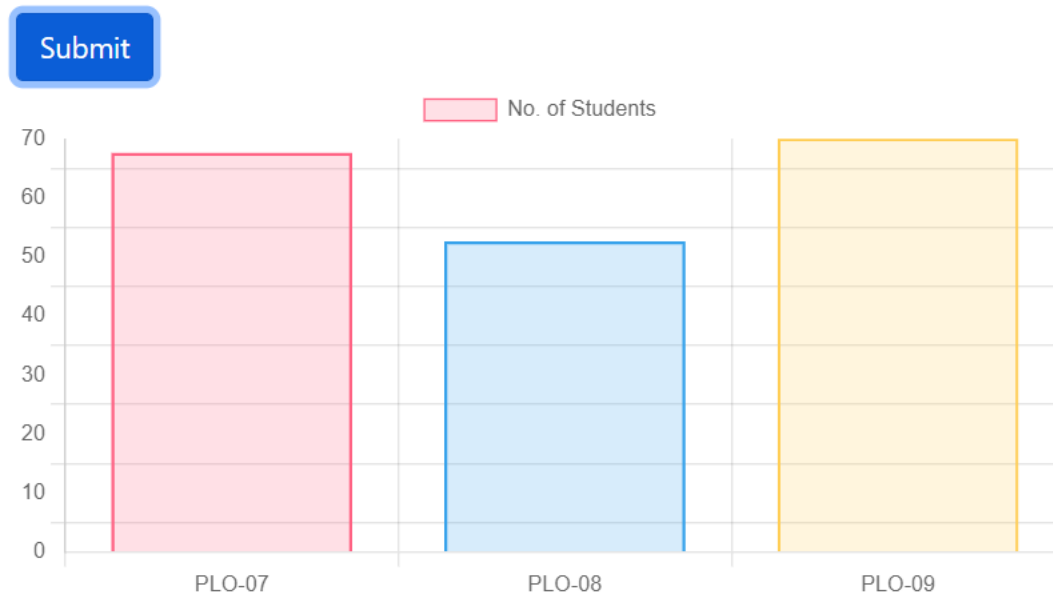
Select Ending Semester:

Select Course(s):

Selected Course(s): Physics 1 (EEE)

Select PLO(s):

Selected PLO(s): PLO-07, PLO-08, PLO-09



```
[HttpPost("/faculty/cpaap")]
```

```
    public IActionResult ComparisonPloAchievedAttemptedSelectPrograms([FromBody]
    ComparisonPloAchievedAttemptedSelectProgramsViewModel viewModel) // 7 continued...
```

```
    {
        var evaluations = ds.evaluations.Where(e =>
        viewModel.SelectedPrograms.Contains(e.Assessment.CourseOutcome.PLO.Program.ProgramID
        ))

            .Where(e =>
        viewModel.SelectedSemesters.Contains(e.Assessment.Section.Semester));

        var evaluationsPloGroups = evaluations.GroupBy(ev =>
        ev.Assessment.CourseOutcome.PLO.PloName);

        var ploNameList = new List<string>();
        var achievedList = new List<float>();
        var attemptedList = new List<float>();

        foreach (var evGroup in evaluationsPloGroups)
        {
            ploNameList.Add(evGroup.Key);

            var passedCount = 0;

            foreach (var ev in evGroup)
            {
                var percent = ev.TotalObtainedMark / ev.Assessment.TotalMark * 100;

                if (percent > ev.Assessment.Section.PassMark)
                {
                    passedCount++;
                }
            }
        }
    }
```



```

        achievedList.Add(passedCount);

        attemptedList.Add(evGroup.Count());

    }

    var myData = new {label = ploNameList, passData = achievedList, attemptData =
attemptedList};

    return Json(myData);
}

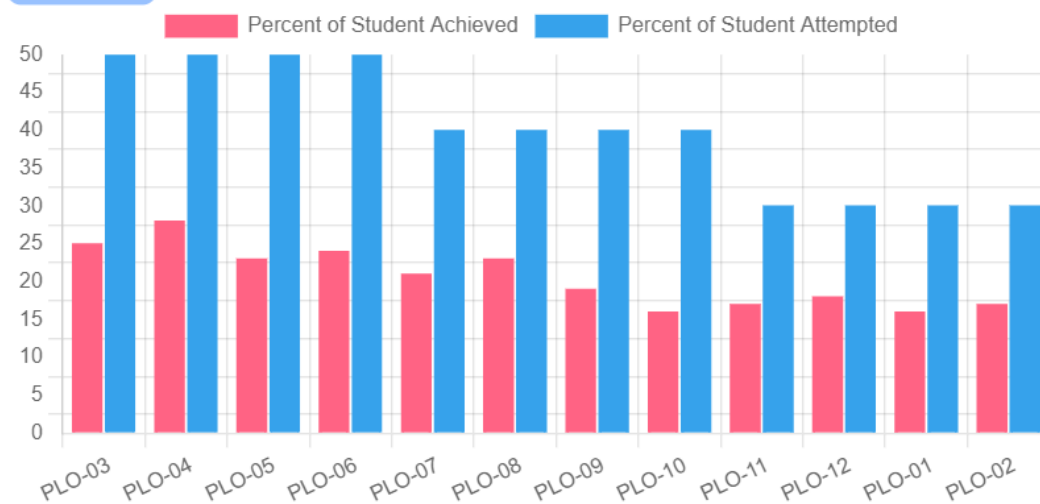
```

Select Program(s)

Selected Programs: ETE

Select Semeser(s)

Selected Semesters: Summer 2003, Summer 2002, Summer 2001



CHAPTER 5: CONCLUSION

PROBLEM:

1. Though we are a part of the system still it was very difficult for us to gather all the information regarding the existing system.
2. While implementing the charts for our proposed system we faced a lot of difficulties as we are not skilled in the tech language, we are using to implement the charts.

SOLUTION:

1. Because of the faculties i.e our stakeholders we have been able to solve all our queries and problems regarding the existing system.
 2. We should skill ourselves with the programming language we are thinking about implement the proposed system with.
- Additional features & future development:
1. Through our system guardians are able to monitor their child/children PLO achievement table.
 2. Our system is not able to generate course assessment report. So, we are planning to work on this part.

REFERENCES:

- [1] Independent University - Bangladesh, "Curriculum for B Sc. in Computer Science and Engineering (Version 2.2)," Independent University - Bangladesh, March 19, 2017.
- [2] Independent University - Bangladesh, "www.iub.edu.bd," 2020.[Online]. Available: www.iub.edu.bd
- [3] T.Contributor, March 2007.[Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/requirements-analysis>.
- [4] A.Pain, 2008.[Online]. Available: <https://sswm.info/train-trainers/facilitation/rich-pictures>.