



Topic: Student Performance Monitoring System

Final Report

Section:01

Faculty Name: Md. Abu Sayed

NAME	ID
Md Ashikuzzaman Esti	1720086
Shahin Sikder	1720559
Md Shahim Uddin Saba	1720586
Md. Tuhin Mia	1721379

Table of Contents

Chapter 1: Introduction	04
Section 1.1: Background of the Organization	04
Section 1.2: Background of the Project.....	04
Section 1.3: Objectives of the Project	04
Section 1.4: Scope of the Project	05
Chapter 2: Requirement Analysis	06
Section 2.1: Rich Picture (Existing).....	06
Section 2.2: Six Elements System Analysis (Existing).....	07
Section 2.3: Process Diagram (Existing).....	12
Section 2.4: Existing Problems & Analysis of the Problem	13
Section 2.5: Rich Picture (Proposed)	15
Section 2.6: Six Element System Analysis (Proposed).....	16
Section 2.7: Process Diagram (Proposed)	20
Chapter 3: Logical System Design	21
Section 3.1: Business Rules	21
Section 3.2: ERD.....	23
Section 3.3: ERD to Relations.....	24
Section 3.4: Normalization.....	25
Section 3.5: Data Dictionary	26
Chapter 4: Physical System Design	30
Section 4.1.1: Add New User - Input Forms.....	30
Purpose	30
Related SQL Used	30
Section 4.1.2: Add Course with CO - Input Forms.....	30
Purpose	30
Related SQL Used	31
Section 4.1.3: Add New Program with PLO – Input Forms	31
Purpose	31
Related SQL Used	31
Section 4.1.4: Enter Marks - Input Forms.....	32
Purpose	32
Related SQL Used	32
Section 4.2.1: Progress View - Output Query and Reports.....	33
Purpose and Use	33
Related SQL Used	33

Section 4.2.2: PLO Achievement - Output Query and Reports	34
Purpose and Use	34
Related SQL Used	34
Section 4.2.3: Student Result - Output Query and Reports.....	35
Purpose and Use	35
Related SQL Used	35
Section 4.3: System Design Architecture.....	36
Chapter 5: Conclusion.....	39
Section 5.1: Problem & Solution.....	39
Section 5.2: Additional Features and Future Development	39
Section 5.3: Conclusion & Recommendation	39
Contribution of Each Member	40
References.....	41
Figure 1: Existing System Rich Picture	06
Figure 2: Existing System BPMN.....	12
Figure 3: Proposed System Rich Picture	15
Figure 4: Proposed System BPMN	20
Figure 5: Entity Relationship Diagram	23
Figure 6: Relational Schema Diagram	24
Figure 7: Normalization Diagram	25
Figure 8: Add New User UI.....	36
Figure 9: Add New Course with CO UI	36
Figure 10: Add New Program with PLO UI.....	37
Figure 11: Enter Marks UI.....	37
Figure 12: PLO Achievement View UI	38
Figure 13: Student Result.....	38

CHAPTER 1: INTRODUCTION

SECTION 1.1: BACKGROUND OF THE INSTITUTION

Independent University Bangladesh or IUB is a private university in Bangladesh. It was established in 1993 under the Private University Act, 1992. with an explicit focus on Research and Global partnerships. IUB has an enrolment of 7,378 students, 11,556 alumni and 401 faculty members (of which 38% have PhD's mostly from North America). The student population is projected to grow at 10% annually. IUB has three academic terms: Spring, Summer, and Autumn. The university is committed to research and global partnerships. Therefore, students are encouraged to engage in research projects, alongside conventional classroom-based learning. The Independent University has academic research collaborations with several prestigious universities including Harvard University, Stanford University, and the University of Colorado at Boulder. [1]

IUB currently have six academic schools:

- School of Business
- School of Engineering and Computer Science
- School of Environmental Sciences and Management
- School of Liberal Arts and Social Sciences
- School of Life Sciences
- School of Public Health

SECTION 1.2: BACKGROUND OF THE PROJECT

The aim of our project is to design, build and deliver a software that we believe will help universities everywhere to promote a more productive and effective way of evaluating students. Measuring the output of students, faculties, departments, and their respective courses to measure their productivity regarding the outcome relevance of the course activities. Basically, to provide a range of tools and data intended to help universities and education authorities such as IEB, UGC, as well as other stakeholders to evaluate the performance of students and inform strategies for improvements. Developing a national framework for Outcome-Based Education while at the same time leaving considerable freedom to universities in implementing local approaches.

SECTION 1.3: OBJECTIVES OF THE PROJECT

Our project intends to create an interactive, user-friendly software that will act as a platform for students, faculties, and other members of the university to help improve the quality of education and revolutionize the way we integrate technology into our education. We believe the data we have collected, evaluated, and arranged will unlock opportunities for massive advancements in our educational sector and will also contribute significantly to the field of Computer Science. Such being the case, SPM system will enhance the project scope so that it will bring about benefits to all the departments. And one of the goals of this Project is to provide insight about how learning might improve in each program-whether it be online, in a classroom, or happening in another context. To provide insight into what students are learning in relation to the big ideas of the courses and the program they aim to complete. To automate the process of monitoring student performance to reduce the manual processing involved in it. To analyze how student populations are learning inside of their programs so that the departments can focus more strategically on equity and success.

SECTION 1.4: SCOPE OF THE PROJECT

Our solution is to create a web application using a Database Management System to store, edit, add, and update necessary data for monitoring student performance and producing and storing related OBE data, reports, and documents. So we will produce potential users for the web based system and speculated how they would be using the system and the necessary information and data they would need access to. Since the problems can arise from many points of all business processes, we will make custom user interfaces and login capabilities for all stakeholders who will also be the users of this system. Since we use a Database Management System for data storage, retrieving necessary files, tabular data, page layouts and reports becomes incredibly easy and allows us to interact with the necessary data to occur real-time. We also create interfaces for all users to easily access these data and use them to generate and download reports. We will also build an interface for faculties to be able to collaborate with each other on developing course outlines, course reports, marksheets, assessments, mapping assessments to CO's and PLOs for PLO achievements, and record assessments of students throughout the semester for all their courses.

CHAPTER 2: REQUIREMENT ANALYSIS

SECTION 2.1: RICH PICTURE (EXISTING)

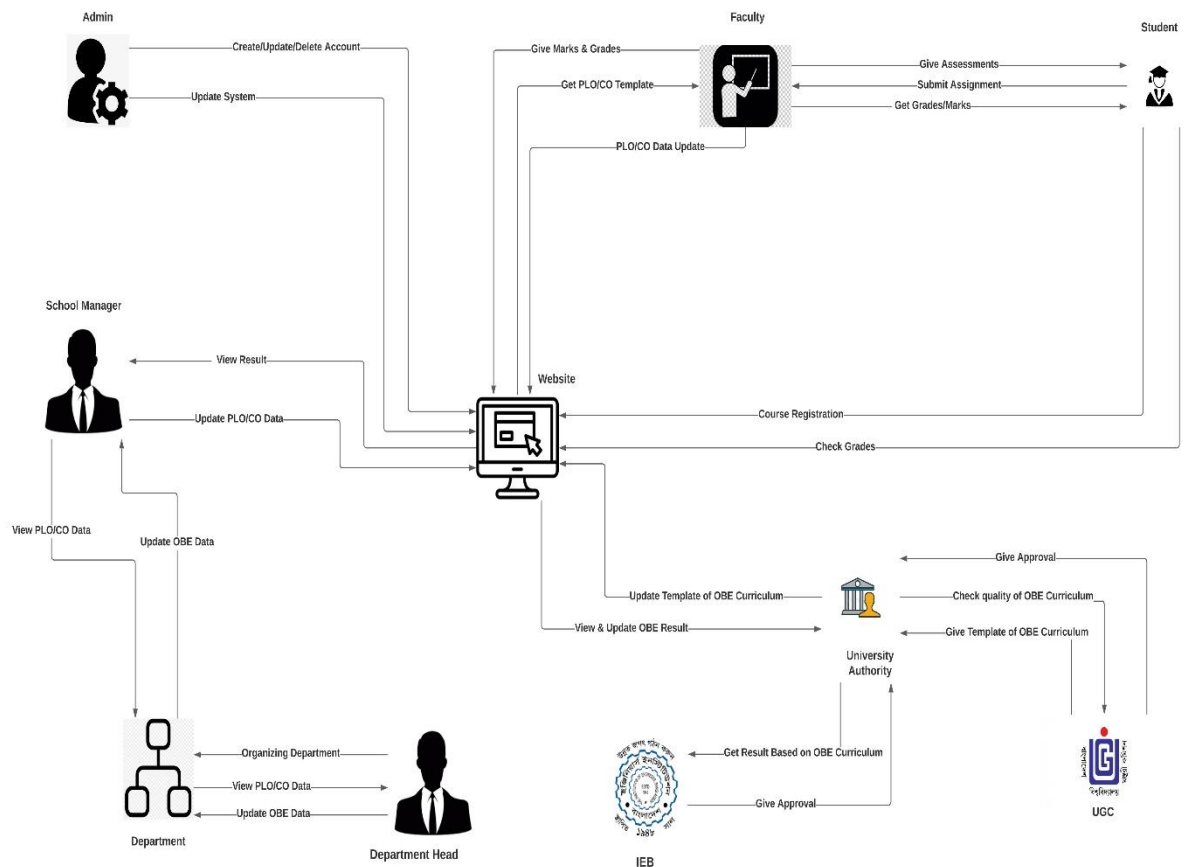


Figure 1: Existing System Rich Picture

Admin creates accounts for new users of the system in the existing business system by gathering the user's name, DOB, and assigning a unique id and password. IEB/UGC sends the university an accreditation manual with defined PLOs, which the university then distributes to all departments, who then collect CO from the respective PLO. If the course content is not provided in the course outline, the department produces a list of course content and a list of course outcomes, then maps the CO to the course content. The department then assigns the received COs to specific PLOs, and hence assigns the COs to specific mid-term, final, and project questions. The instructor sends the course overview to the online classroom in the present online system, and the course outline is printed and distributed among the students in the physical courses. The instructor gets COs that must be earned in a certain course and administers exams to evaluate students on various course outcomes. As a result, the teacher translates the overall marks and totals all of the CO's. The CO's grades are sent to the department by the instructor. CO percentages are calculated by the department, and if the percentage is greater than or equal to 40%, a student passes that specific PLO; otherwise, the student fails. The related COs are mapped to the PLOs, and the PLO accomplishments are documented. To create a report, the department pulls student PLO achievement data from a previously generated OBE mark sheet and creates a report based on UGC/ IEB/ Higher Management criteria.

SECTION 2.2: SIX ELEMENTS SYSTEM ANALYSIS (EXISTING)

Process	System Roles					
	Human	Non-Computing Hardware	Computing Hardware	Software	Database	Network & Communication
Account Generation	<p>Admin:</p> <ol style="list-style-type: none"> Stores new students information into a database and as per the data generates default accounts for Students. Generates account for faculties by receiving information from department head. Update student /Faculty accounts based on their need. <p>IT Team:</p> <ol style="list-style-type: none"> Maintains the entire system. Creates the 	<p>Pen and Paper:</p> <ol style="list-style-type: none"> Can be used to write log in data of students or any employee manually. 	<p>Computer:</p> <ol style="list-style-type: none"> Used for accessing & adding/editing account details. 	<p>MS Excel:</p> <ol style="list-style-type: none"> All related information are stored. 	<p>SQL:</p> <ol style="list-style-type: none"> Student/Faculty information is kept on SQL based Database. <p>Other Sources:</p> <ol style="list-style-type: none"> All related information are stored in the specific location. 	<p>Internet:</p> <ol style="list-style-type: none"> Used so that the system can function Properly. User interface and website pages are served using internet access.

	entire UI for all the stakeholders					
Basic Data Entry	<p>Admin:</p> <ol style="list-style-type: none"> 1. Generates user accounts providing default ID and password. <p>School Manager:</p> <ol style="list-style-type: none"> 1. Inputs and Update students PLO/CO data into the system. 2. Receives data PLO/CO template from the system. <p>Department Head:</p> <ol style="list-style-type: none"> 1. Organizes department Meeting through the server. 2. Send faculties PLO/CO template through email <p>Faculty:</p> <ol style="list-style-type: none"> 1. Types course outline Based on university CO data. 2. Inputs data regarding student attendance into the system. 3. Takes exams throughout the semester. 	<p>Pen and Paper:</p> <ol style="list-style-type: none"> 1. Pen and paper can be used to note down additional information. <p>File Holder:</p> <ol style="list-style-type: none"> 1. Any sort of file holder can be used to store data physically. 	<p>Computer</p> <ol style="list-style-type: none"> 1. Use computers to make softcopies of the PLO/CO data. <p>Printer:</p> <ol style="list-style-type: none"> 1. Print hardcopies of final versions of the student assessment report. 	<p>MS Word:</p> <ol style="list-style-type: none"> 1. Faculty can use MS Word to make a course outline for each course. <p>Excel Sheet:</p> <ol style="list-style-type: none"> 1. Excel can be used to store student information-n. 	<p>SQL:</p> <ol style="list-style-type: none"> 1. SQL based database To store all the data into the system. <p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information are stored in the specific location. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. Used so that the system can function Properly and all related data are searched through the internet.

	<p>4. Input student grades into the system at the end of the semester.</p> <p>Student :</p> <p>1. Submit assignments into the system.</p> <p>University authority:</p> <p>1.Store OBE curriculum into the system.</p>					
Course Registration	<p>Student:</p> <p>1. Load the web page /user interface.</p> <p>2. Log in using user id and password</p> <p>3. Load list of causes</p> <p>4.Check mark necessary courses which students want to register for.</p> <p>5. Click register course and complete registration.</p> <p>6. Get the pdf format of the course bill for the current semester.</p> <p>IT Team:</p> <p>1. Maintain the entire course registration process and monitor any bugs .</p>	<p>Pen and Paper:</p> <p>1. Can be used to write a list of courses each student wants to register for course registration.</p>	<p>Computer:</p> <p>1. Used to view the UI and go for further course registration process.</p>	<p>Microsoft Word :</p> <p>1. Can be used to list courses which students want to register for the current semester.</p>	<p>SQL:</p> <p>1. Any sort of SQL based database can be used to store information.</p> <p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	<p>Internet:</p> <p>1. The Internet is required to run the entire system.</p>

	2. Generate Student course bill manually if needed.					
View and validate OBE, Course Assessment report	<p>IEB/ UGC:</p> <ol style="list-style-type: none"> 1. Notify respective university about deadline of OBE Report Submission. 2. Provide appointment schedules for respective university employees. 3. Receive OBE and course assessment report from respective university for better validation purpose. 4. Provide necessary documents and templates for the improvement of the educational system. based on received reports. <p>School Authority:</p> <ol style="list-style-type: none"> 1. View OBE and Course Assessment report. <p>Departmental Head:</p> <ol style="list-style-type: none"> 1. Store OBE and course assessment report into the 	<p>Pen & Paper:</p> <ol style="list-style-type: none"> 1. Written exams conducted by the faculty to evaluate the students. 2. Used for writing any necessary information manually. 	<p>Computer:</p> <ol style="list-style-type: none"> 1. Used to display OBE and Course Assessment Report's softcopies. 2. Transfer OBE and Course Assessment Reports to other computers or any secured computing devices 	<p>MS Excel:</p> <ol style="list-style-type: none"> 1. All related information are stored. <p>Microsoft Word :</p> <ol style="list-style-type: none"> 1. Can be used to store OBE , course assessment report or any necessary data into software. <p>Adobe Acrobat:</p> <ol style="list-style-type: none"> 1. Used to convert any file type to pdf form /view pdf files. 	<p>SQL:</p> <ol style="list-style-type: none"> 1. OBE , course assessment data's need to be stored inside any SQL based database <p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information are stored in the specific location. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. OBE and course assessment reports may need to be transferred / mailed online.

	<p>system for further validation.</p> <p>2. Provide all the necessary documents to School authorities.</p>					
Store and update student performance data	<p>Faculty:</p> <p>1. Input individual student performance data after performance assessment into excel.</p> <p>2. Input individual student performance data into the university database.</p> <p>3. Update student performance data if change required.</p> <p>School authority:</p> <p>1. Grants permission if change of grade required.</p>	<p>Pen and Paper:</p> <p>1. Pen and paper can be used to note down student performance data manually.</p> <p>File holder:</p> <p>1. Any sort of file holder can be used to store data physically.</p>	<p>Computer:</p> <p>1. Use computers to make softcopies of the student performance data.</p> <p>Printer:</p> <p>1. Print hardcopies of final versions of the student performance report.</p>	<p>MS Word:</p> <p>1. Faculty can use MS word to input individual student performance data</p> <p>Excel Sheet:</p> <p>1. Excel can be used to store student performance data.</p>	<p>SQL:</p> <p>1. SQL: Based database To store all the data into the system.</p> <p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	<p>Internet:</p> <p>1. All related data are provided through the internet.</p>

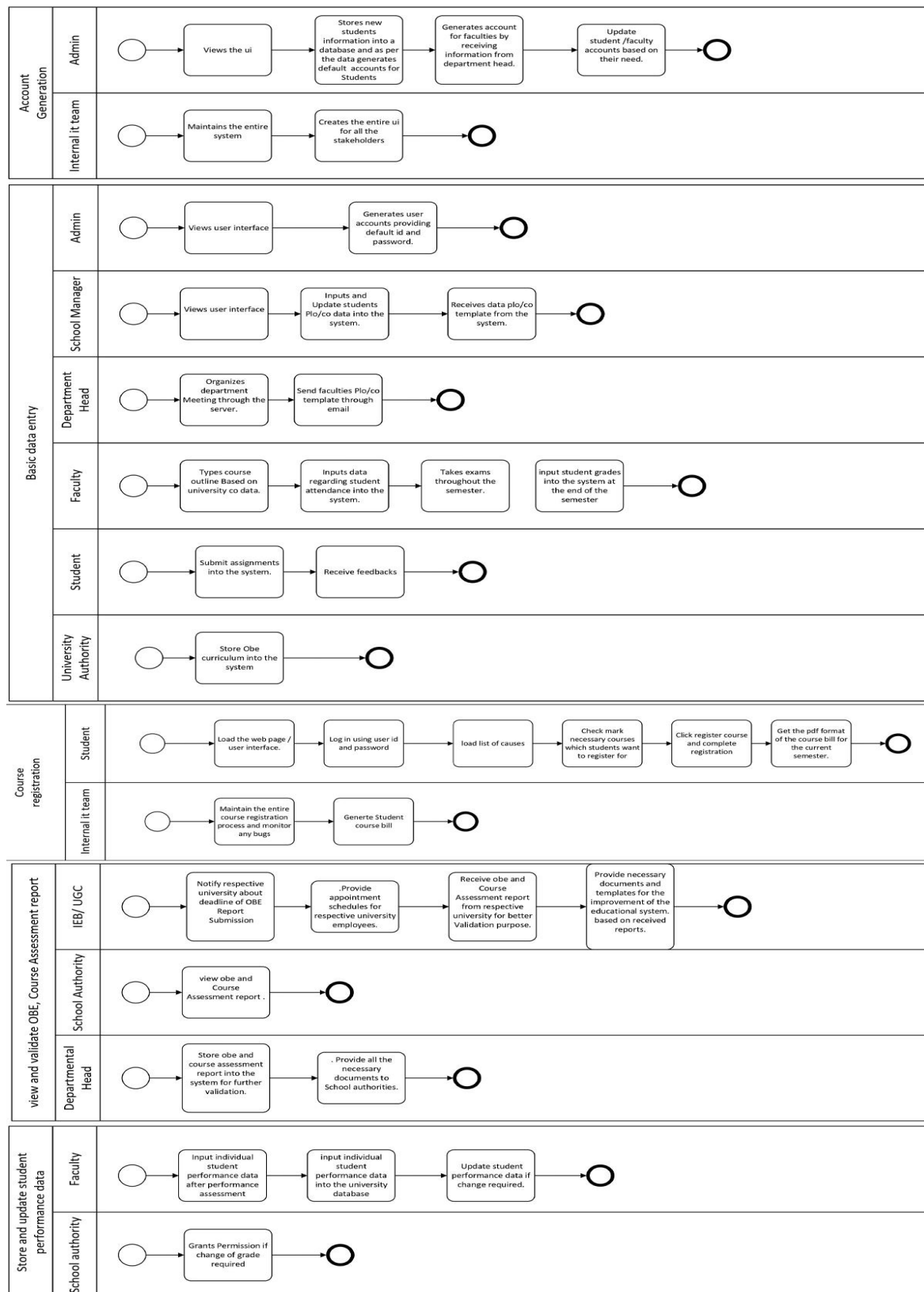
SECTION 2.3: PROCESS DIAGRAM (EXISTING)

Figure 2: Existing System BPMN

SECTION 2.4: PROBLEM ANALYSIS OF EXISTING BUSINESS MODULE

Process Name	Stakeholders	Problems Found	Analysis (Reason of the Problems)	Proposed Solution
Account Generation	Admin	Different individuals with the same username will create flaws.	There might be several persons with the same name.	Including username and Date of birth, users also input their email Id since email id are unique and all this information is different for each user.
Map course Objective with Program Language Outcome	Department	<ol style="list-style-type: none"> 1. Types course content into word file and checks if it already exists in the course outline. 2. Types course outcomes into word file. 3. Manually maps the COs with the course content. 4. Manually mapping CO's to PLO's. 	The total process for Mapping course objective with program language outcome is done by human not a system so the efficiency of the process becomes less.	Using a built-in web-based application which will gather the course content and course outcome from the department and then map the COs with the course content and hence mapping the received COs to specific PLO's.
Course Registration	Student	<ol style="list-style-type: none"> 1. less course enrolment capacity cannot fulfil the target number of students. 2. Server overflow. 	1. Less course enrolment capacity causes havoc as students cannot enrol required courses in current semester thus also creates server load making the entire process slow.	Increasing the capacity of courses automatically as soon as it gets filled so that there is no delay in the registration process. There must be an internal it team who handles these matters.

Collects Student Marks	Instructor	<p>1. Instructor collects student marks by taking assessments and converts the total marks manually by using MS Excel.</p> <p>2. Using MS Excel to find out the total course outcome and check if it matches with the requirements.</p>	<p>The entire process for collecting and calculating student marks is very less efficient and time consuming</p> <p>There is high chance of manual type error by the instructor.</p>	A system can be applied so that it can perform all these tasks automatically.
Add and Update Grade	Instructor	<p>1. Adding grade is done manually by checking the marks.</p> <p>2. Manually checking if a student passed or failed.</p> <p>3. Manually recording PLO achievement from the CO received.</p>	<p>The entire process is done manually more time consuming and less efficient.</p>	<p>There needs to be a system which automatically adds grade by scanning through the marksheet and updates the grade.</p> <p>Also, which can calculate CO percentages automatically with the marks provided as an input. The system will show by what percentages the PLOs and COs have been achieved and all the COs and PLOs that the student failed to achieve.</p>
Report generation	Department	Manually typing the report data into the system, which finally becomes the report.	OBE data submitted by the instructor needs to be collected manually into the report template, which is less efficient.	This factor can be solved by introducing a mechanism which will enable the department to automatically generate the report by clicking into a certain button.

SECTION 2.5: RICH PICTURE (PROPOSED)

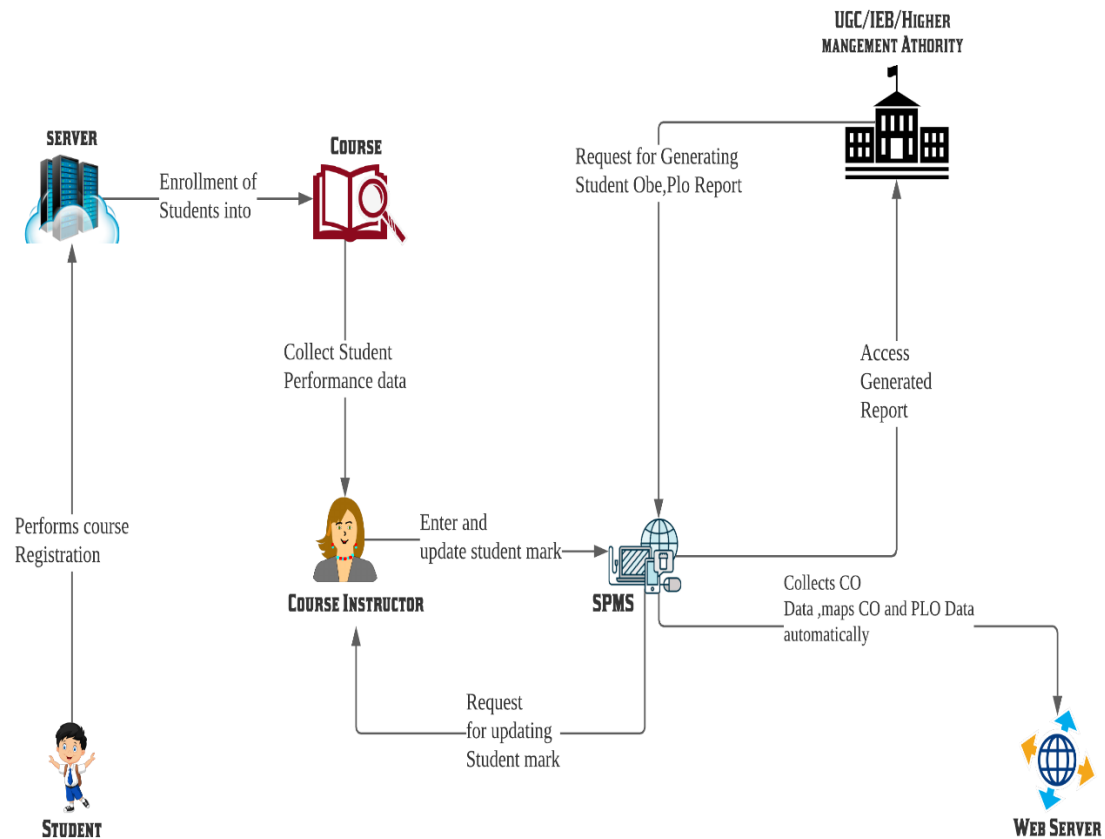


Figure 3: Proposed System Rich Picture

The CO data is gathered by the system from the website / IEB handbook. The system then generates a list of course content and a list of course outcomes, allowing the COs to be mapped to the course content. The COs are subsequently assigned to specific PLOs. The instructor receives COs that must be earned in a certain course and administers tests to evaluate the students in that course. After that, the instructor transforms the total marks and computes the total for all CO's. The CO markings are entered into the system, which then stores them. The system subsequently generates reports for the UGC/ IEB/ Higher Management to analyse based on the stored data.

SECTION 2.6: SIX ELEMENT SYSTEM ANALYSIS (PROPOSED)

Process	System Roles					
	Human	Non-Computing Hardware	Computing Hardware	Software	Database	Network & Communication
Mapping COs with PLOs	<p>Department:</p> <ol style="list-style-type: none"> 1. IEB/UGC provides with effective accreditation manual with PLO's efficient for each department. 2. Each Department Collects information regarding CO from the website. 3. If the course content exists in the CO, review the course content automatically, else create a required list of the course content. 4. Creates a list of the course outcomes. 5. Login to the system with user ID and password. Map the CO with the course content. 6. Map the received CO's to specific PLOs. 	<p>Pen and Paper:</p> <ol style="list-style-type: none"> 1. Department might print and get the hard copy of the accreditation manual. 	<p>Computer:</p> <ol style="list-style-type: none"> 1. All of this data is stored and processed using a computer. 2. Computer is used to work with the system. 	<p>MS Excel:</p> <ol style="list-style-type: none"> 1. To store all related information. <p>Microsoft Word :</p> <ol style="list-style-type: none"> 1. Used to update/create related data. <p>Adobe Acrobat:</p> <ol style="list-style-type: none"> 1. Used to view pdf files. 	<p>Other Sources:</p> <ol style="list-style-type: none"> 1. Databases like SQL are used to store and display any sort of data table or chart. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. All related data is searched through the internet. 2. Internet is also used to send or receive any data.

Course Registration	<p>Student:</p> <ol style="list-style-type: none"> 1. Log in to SPMS with Id and password. 2. Click to the registration page, load courses, and select courses. 3. Check course capacity and prerequisites. 4. Click the submit button and complete registration. 5. Display "Registration Successfully Done". 	<p>Pen & Paper:</p> <ol style="list-style-type: none"> 1. Sometimes course registration is done manually by writing into a paper. 2. Students might also use pen and paper to prepare a plan for course registration. 	<p>Computer:</p> <ol style="list-style-type: none"> 1. Computer is used to run the entire system. 	<p>SPMS:</p> <ol style="list-style-type: none"> 1. Web based application like SPMS is used to complete course registration. 	<p>SQL Server:</p> <ol style="list-style-type: none"> 1. SPMS is integrated with MS SQL Server. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. SPMS is a web-based application and requires internet access to run.
Collects Student Marks	<p>Instructor:</p> <ol style="list-style-type: none"> 1. Receives CO's to be achieved in that particular course. 2. Takes exams such as mid-term, final & project to assess students in various course outcomes. 3. Gather marks for each student instance. 4. Convert the total mark. 5. Calculate totals for all the CO's. 	<p>Pen & Paper:</p> <ol style="list-style-type: none"> 1. Written exams conducted by the faculty to evaluate the students. 	<p>Computer:</p> <ol style="list-style-type: none"> 1. The entire system and all the data accessed using a computer. 	<p>MS Excel:</p> <ol style="list-style-type: none"> 1. For each student instance the entire information is stored in MS excel. <p>Microsoft Word :</p> <ol style="list-style-type: none"> 1. Used to update/create related data. <p>Adobe Acrobat:</p> <ol style="list-style-type: none"> 1. Used to view pdf files. 	<p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information is stored in the specific location. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. Need the internet to store and publish student marks.

Add Student Marks	<p>Instructor:</p> <ol style="list-style-type: none"> 1. Login to the system with ID and password. Enter students' total CO marks achieved in all CO's in mid-term, final & project into the system including student id, course id, section, semester. 2. System stores students' marks achieved in all CO's based on Course assessments. 3. Calculates CO percentages. 4. If $\geq 40\%$ CO's have been achieved; a student passes that certain CO otherwise fails. 5. The corresponding CO's are mapped against PLO's and PLO achievements are recorded. 	<p>Pen & Paper:</p> <ol style="list-style-type: none"> 1. Student might apply for grade change manually by writing an application 	<p>Computer:</p> <ol style="list-style-type: none"> 1. All related data is searched and stored using a computer. 2. The entire system and all the data accessed using a computer. 	<p>MS Excel:</p> <ol style="list-style-type: none"> 1. All related information is stored. <p>Microsoft Word :</p> <ol style="list-style-type: none"> 1. Used to update/create related data. <p>Adobe Acrobat:</p> <ol style="list-style-type: none"> 1. Used to view pdf files. 	<p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information is stored in the specific location. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. All related data is provided through the internet.
Generate Report	<p>Department:</p> <ol style="list-style-type: none"> 1. Users will login to the system and retrieve the student information of PLO achievement from previously 	None.	<p>Computer:</p> <ol style="list-style-type: none"> 1. All related data is searched and stored using a computer. 	<p>MS Excel:</p> <ol style="list-style-type: none"> 1. All related information are stored and calculated. 	<p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information are stored in the specific location. 	<p>Internet:</p> <ol style="list-style-type: none"> 1. All related data is provided through the internet.

	<p>calculated OBE marksheet.</p> <p>2. Generate reports after comparing results of multiple students.</p> <p>3. Percentage of successfully passed or failed to achieve are calculated based on the total number of students.</p>			<p>Adobe Acrobat:</p> <p>1. Used to view pdf files.</p>		
Request for report generation	<p>UGC/ IEB/ Higher Management:</p> <p>Login to the system with user ID and password.</p> <p>Asking to generate report for:</p> <p>1a. Viewing the number of students passing or failing in a certain PLO.</p> <p>1b. Viewing progression of students with charts and graphs.</p>	<p>Pen & Paper:</p> <p>1. Paper is required for the printing purpose.</p>	<p>Printer:</p> <p>1. Print the report based on users' request.</p> <p>Computer:</p> <p>1. All related data is searched and stored using a computer.</p>	<p>Adobe Acrobat:</p> <p>1. Used to view pdf files.</p>	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	<p>Internet:</p> <p>1. All related data is provided through the internet.</p>

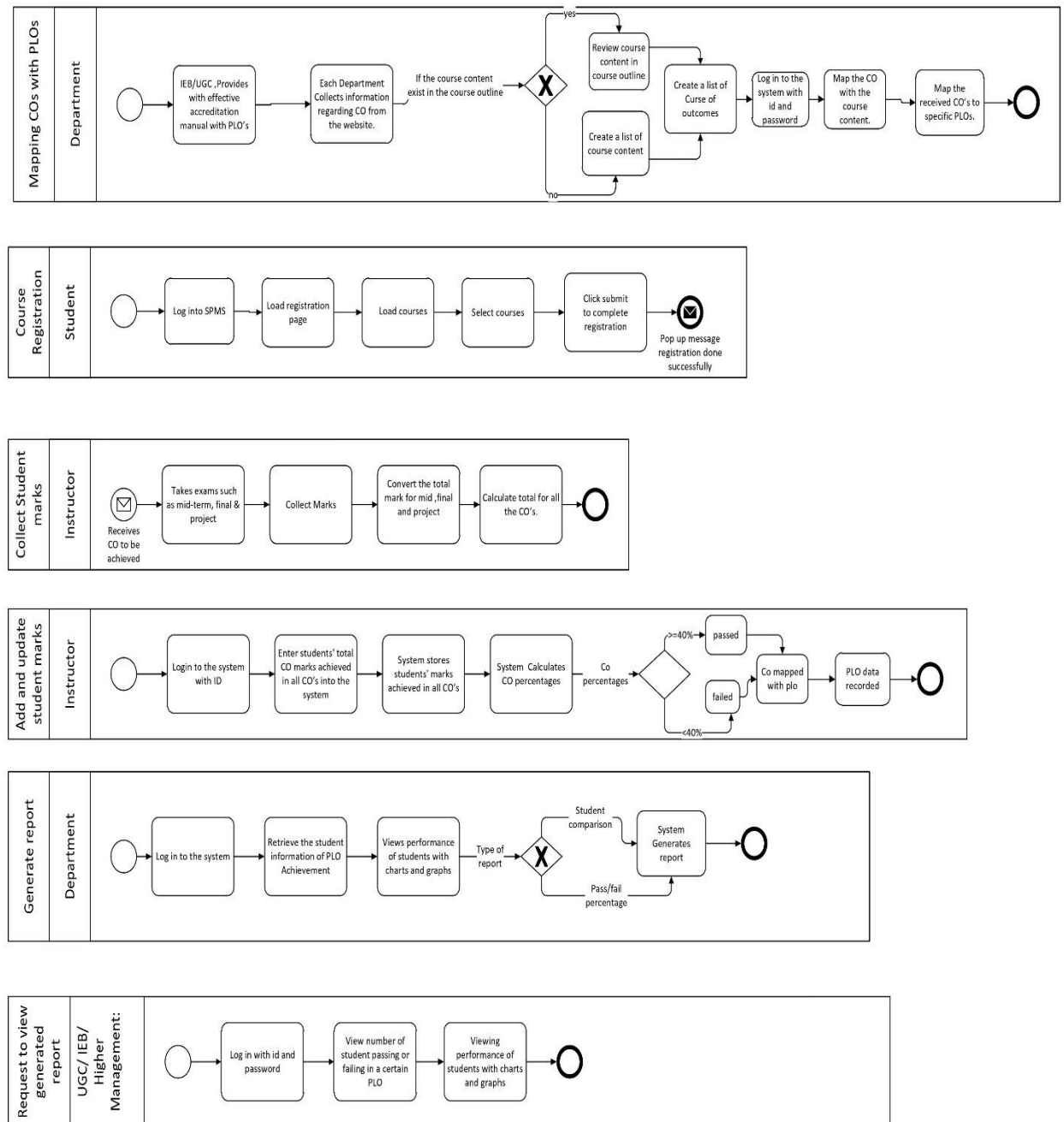
SECTION 2.7: PROCESS DIAGRAM (PROPOSED)

Figure 4: Proposed System BPMN

CHAPTER 3: LOGICAL SYSTEM DESIGN

SECTION 3.1: BUSINESS RULES

1. The IEB sends departments an accreditation manual with PLOs defined. The information is then gathered by each department from the IEB manual/website.
2. Each of the Degree Programs into which a student is accepted is part of a department. A Program only has one department. Each program is made up of many courses, and each course belongs to only one program.
3. Each program will have a set of programs learning outcomes (PLO) under the OBE model. A PLO has a PLOID, as well as a name and a description.
4. A Department can have many students, but each student can only have one department. Each department has a name and a phone number. Each department is led by a specific faculty member. The department offers many courses, but a course is only offered by one department.
5. Courses have a set of course outcomes (COs) that are mapped with the degree program's PLOs to evaluate students in each course. A CO must be mapped to a single PLO. A PLO can be associated with one or more CO's. COs are measured using various assessment techniques (e.g., quiz, mid, final, project, presentation). A CO has a COID, as well as a CO name and description.
6. A CO is mapped with exactly one assessment, and an assessment is mapped with one or more assessments. An AssessmentID is used to identify each of the assessments. Assessments have a name (e.g., Mid Q1, Mid Q2, Final Q1, Final Q3, etc.), a CourseID, a COID, a Section Number, and the total marks that can be obtained in that particular assessment. There is only one section in an assessment. One or more assessments are required for each section.
7. Faculties assess the COs and mapped PLOs attained by each student in a course. An evaluation is carried out by one or more faculties, and each faculty is required to carry out at least one evaluation. There is a deadline for faculty evaluations. Each evaluation is uniquely identified by a StudentID and an AssessmentID. The total marks obtained by a student in an assessment are also included in evaluation. An assessment may have one or more evaluations, but each evaluation is completed for only one assessment. A student is assigned to one or more evaluations, but each evaluation is assigned to only one student.
8. Accounts are kept for two types of system users: students and faculty. An account has an ID, a name (first and last name), a birth date, a gender, an email address, a phone number, and an address. To register for a course, a student must first log in to SPMS.
9. Academic qualifications (highest degree certificate gained so far), area/s of specialization, job position (Lecturer, Professor), and salary are all attributes of faculties. A student must have an enrolment date and have completed the entire course. A faculty has only one department, while a department has several faculties.
10. A CourseID is a unique identifier for the course. A CourseTitle is also assigned to a course. A course may be a prerequisite for one or more other courses, and a course itself may have one or more prerequisites. A course can be mapped to many other courses, and multiple courses can be mapped to the same course.

11. Each course must have at least one faculty member teaching it. A Faculty member may teach more than one course. Every faculty member has a teaching schedule, which includes teaching days and teaching hours for each course. In each semester, there may be multiple sections for each course, but each section teaches only one course.
12. A section has a section number; however, sections with the same number may be assigned to different courses. As a result, a semester and CourseID, as well as the section number, are required to identify a section. A section has a schedule as well as a maximum capacity. A student may enroll in one or more sections, with each section requiring at least one student. Each semester, a student's courses have a registration deadline. If no students are enrolled in a course, all of its sections, as well as the course itself, are removed.

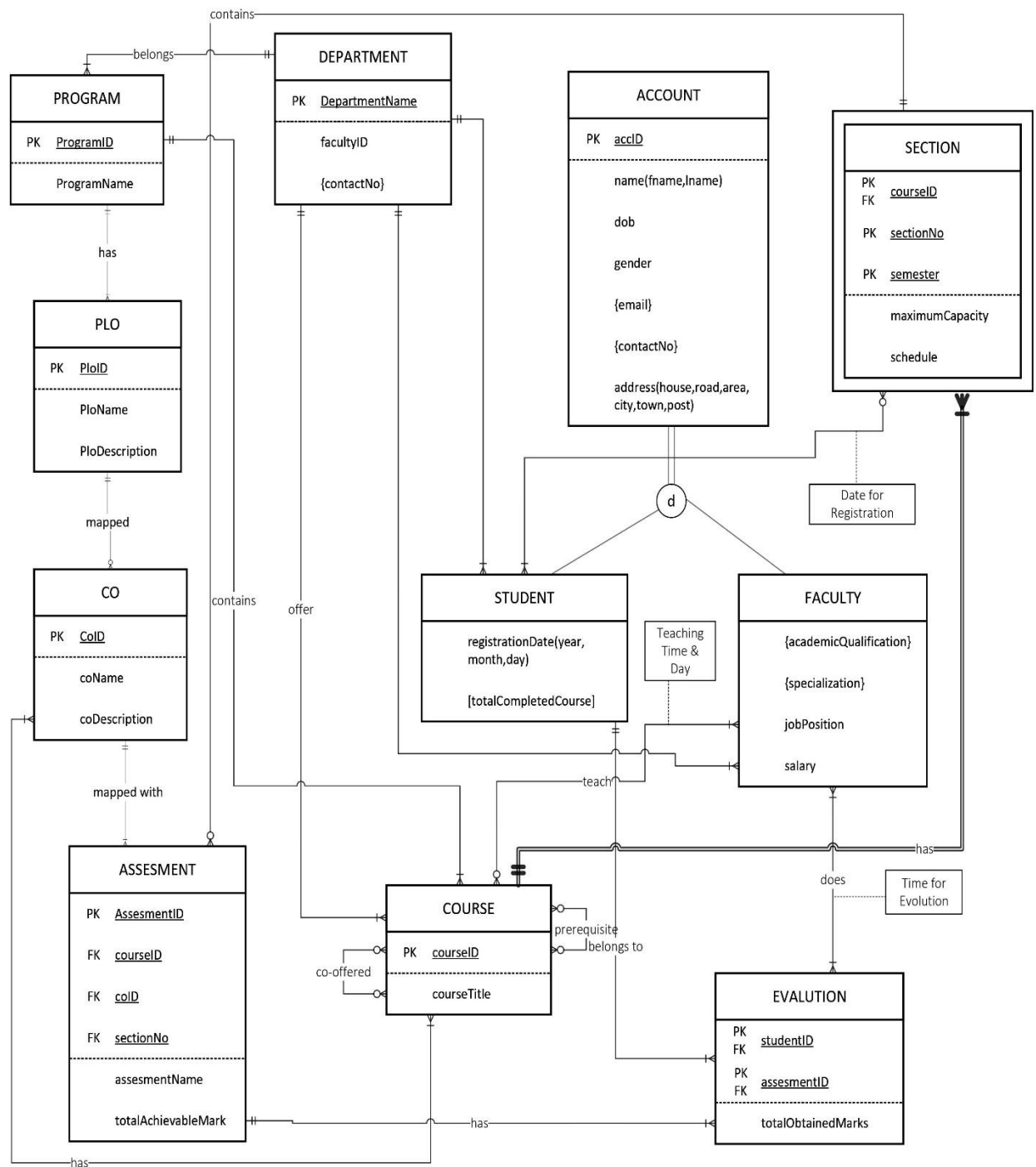
SECTION 3.2: ERD

Figure 5: Entity Relationship Diagram

SECTION 3.3: ERD TO RELATIONS

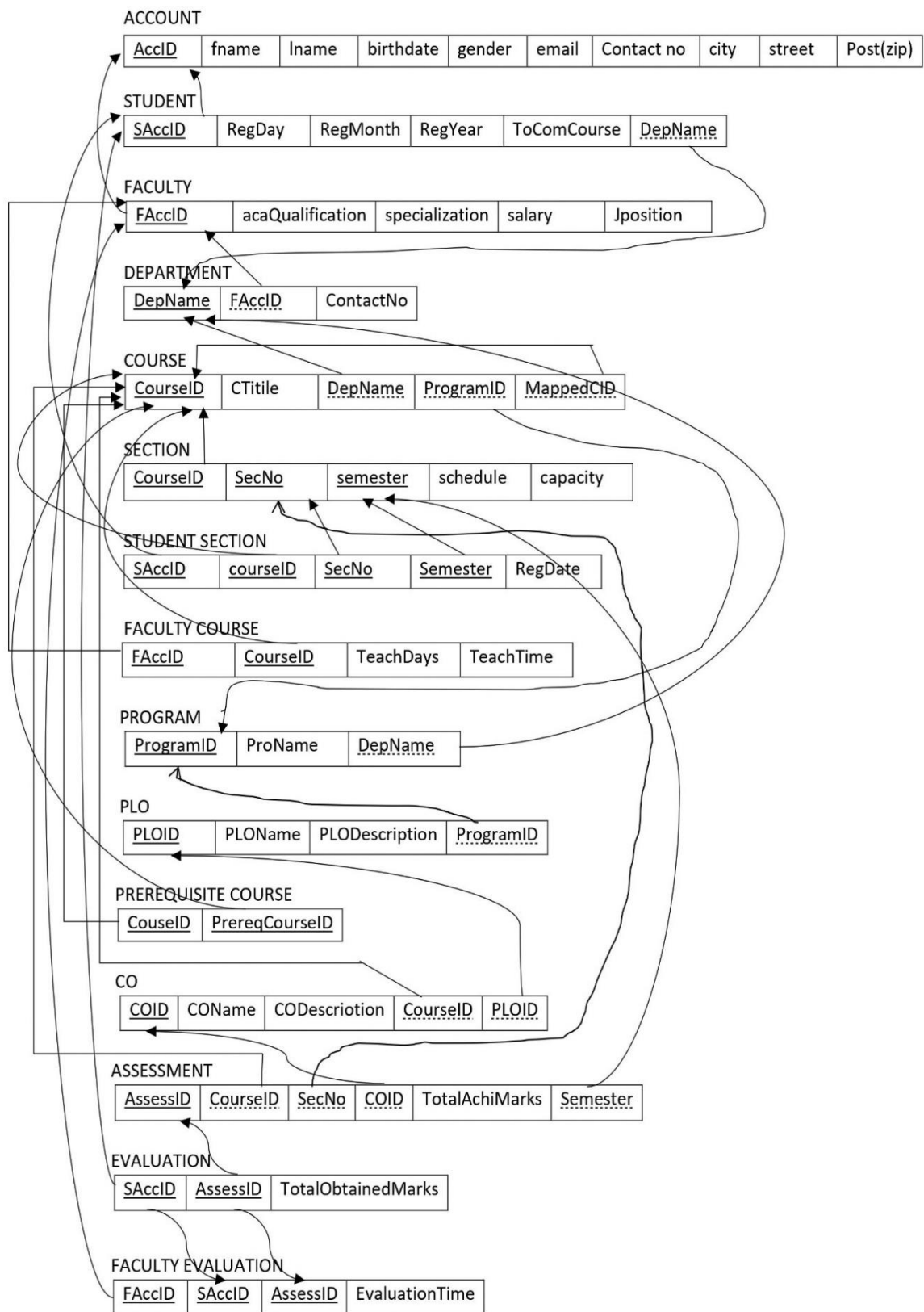


Figure 6: Relational Schema Diagram

SECTION 3.4: NORMALIZATION

AccountID (a) → fname (a1), lname (a2), dob (a3), gender (a4), email (a5), contactno (a6), address (a7), regdate (a8), totalcompletedcourse (a9), departmentname (c), academicqualification (a10), specialization (a11), jobposition (a12), salary (a13)

CourseID (b) → coursetitle (b1), mappedcourseid (b2), programid (d), departmentname (c), prerequisitecourseid(p)

CourseID (b), SectionNumber (s), Semester (t) → schedule (u), maxcapacity(v)

Saccountid (aa), CourseID (b), SectionNo (s), Semester (t) → registrationdate(r)

Departmentname (c) → Faccountid (ab), contactno (c1)

CourseID (b), Faccountid (ac) → teachdays (q), teachtime (w)

ProgramID (d) → programname (d1), DepartmentName (c)

PLOID (e) → PLOname (c1), PLOdescription (c2), ProgramID (d)

COID (f) → COname (f1), COdescription (f2), PLOID (e), CourseID (b)

AssessmentID (g) → CourseID (b), sectionno (s), COID (f), totalachievablemarks (g1), semester (t)

AssessmentID (g), SaccountID (ad) → totalobtainedmarks (m)

AssessmentID (g), S.accountID (ad), FaccountID (ac) → evaluationtime (n)

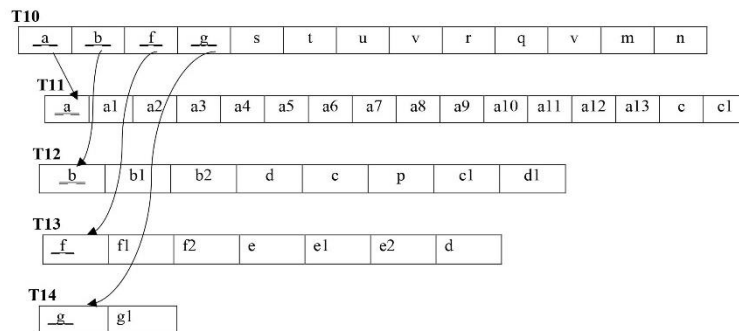
Note: Removed all derived attribute. And SAccountID (aa), FAccountID (ab), S.accountID (ad), Faccountid (ac) are all AccountID (a)

1NF:

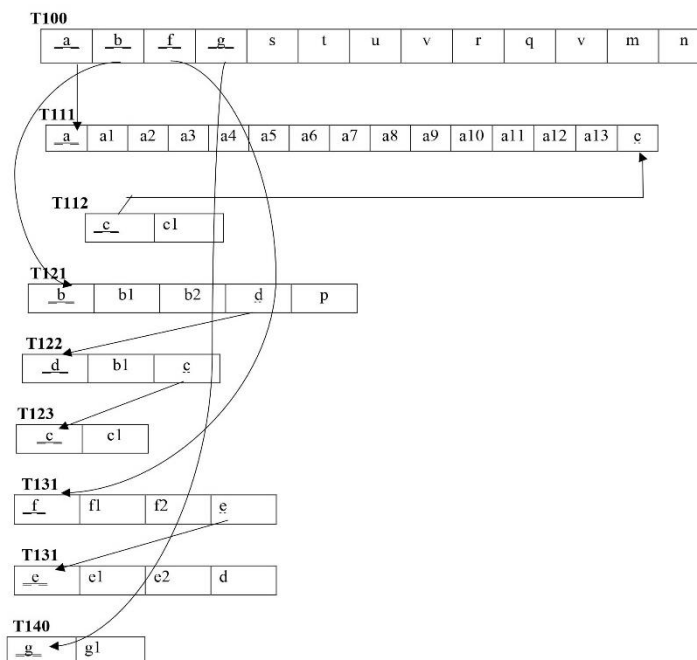
T1

<u>a</u>	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a11	a12	a13	m	r	<u>b</u>	b1	b2	d	c	p
s	t	u	v	c1	n	q	w	d1	e	e1
e2	<u>f</u>	f1	f2	<u>g</u>	g1					

2NF:



3NF:



BCNF: No Non-Key can identify any primary key or part of the primary key. Therefore we have all relation upto BCNF.

Figure 7: Normalization Diagram

SECTION 3.5: DATA DICTIONARYAccount_T:

Name	Data Type	Size	Remark
ID	varchar	7	This is the primary key to understanding this relationship. This contains the student and instructor IDs. For instance, '1720586' for a student and '5533' for a faculty member.
firstname	varchar	15	This is the students' or instructors' first name. As an example, consider the word 'Tuhin'
lastname	varchar	15	This is the students' or faculty's last name. As an example, consider the word 'Mia'
birthdate	Date	"dd/mm/yyyy"	This section contains the students' or faculty's birth dates. As an example, consider the date '01/01/2001'
gender	varchar	6	This section includes information about the gender of the students or instructors. For instance, 'Male' or 'Female.'
email	varchar	25	This is the students' or faculty's email address.
contactnumber	varchar	11	This is the student or faculty contact information. As an example, consider the number '01777222456'
address	varchar	40	This is the students' or faculty's home address.

Department_T:

Name	Data Type	Size	Remark
dptname	varchar	40	This is the primary key to understanding this relationship. This contains the name of the department.
facultyID	varchar	4	This contains the department head's name.
contactnumber	varchar	11	This contains the department's phone number.

Student_T:

Name	Data Type	Size	Remark
SaccountID	varchar	7	This is the primary key in this relationship. This contains the students' account id. As an example, consider the number '1720586'.
enrollDate	Date	"dd/mm/yyyy"	This section contains the students' enrollment dates.
totalcompletedcourse	Number	3	This includes the entire course that the students completed.
dptName	varchar	40	This is a foreign key from DEPARTMENT's table.

Faculty_T:

Name	Data Type	Size	Remark
facultyID	varchar	4	This is the primary key to understanding this relationship. This contains the ID of the faculty member. As an example, consider the number '5533'
acqualifications	varchar	60	This is the faculty's most recent academic credentials.
specialization	varchar	60	This is the faculty's area of expertise.
jobposition	varchar	9	This is the faculty's job description. For instance, 'Professor' or 'Lecturer'.
salary	Number	7	This is the faculty's monthly salary.

Program_T:

Name	Data Type	Size	Remark
programID	varchar	4	This is the primary key to understanding this relationship. This field contains the program's id.
programName	varchar	15	This contains the name of the program.
dptName	varchar	40	This is a foreign key from DEPARTMENT's table.

Course_T:

Name	Data Type	Size	Remark
courseID	varchar	7	This is the primary key to understanding this relationship. The course id is contained in this. As an example, consider the term 'CSE303.'
courseTitle	varchar	20	This section contains the course title for a specific course. As an example, consider the term 'Database Management.'

Facultycourse_T:

Name	Data Type	Size	Remark
FaccountID	varchar	4	This is the composite key of this relation. This contains the account id of the instructors. Example: '5533'.
courseID	varchar	7	This is the composite key of this relation. This contains the course id. Example: 'CSE303'.
teachDays	varchar	15	This contains the days a particular faculty teaches.
teachTime	time	"hh:mm:ss"	This contains the teaching time of a faculty.

Section_T:

Name	Data Type	Size	Remark
courseID	varchar	7	This is the relation's composite key. The course id is contained in this. As an example, consider the term 'CSE303.'

sectionNo	Number	2	This is the relation's composite key. This contains the section number of a specific course in a specific semester.
semester	varchar	6	This is the relation's composite key. This includes the semester name. For instance, 'Spring,' 'Summer,' and 'Autumn.'
schedule	varchar	15	This is a section's schedule, which is the timing of a course.
maxcapacity	Number	2	This is the maximum capacity of a section.

Studentsection_T:

Name	Data Type	Size	Remark
SaccountID	varchar	7	This is the composite key for this relationship. This contains the students' account id. As an example, consider the number '1720586'.
courseID	varchar	7	This is the relation's composite key. The course id is contained in this. As an example, consider the term 'CSE303.'
sectionNo	Number	2	This is the relation's composite key. This contains the section number of a specific course in a specific semester.
semester	varchar	6	This is the relation's composite key. This includes the semester name. For instance, 'Spring,' 'Summer,' and 'Autumn.'
regDate	Date	“dd/mm/yy yy”	This section contains the registration dates for students enrolled in a specific course.

PLO_T:

Name	Data Type	Size	Remark
PLOID	varchar	4	This is the primary key to understanding this relationship. The PLO ID is contained in this.
PLOName	varchar	15	This section contains the PLO name of a specific course. As an example, consider the term 'Database Management.'
PLODescription	varchar	100	This is the PLO description.
programID	varchar	4	This is a foreign key from PROGRAM's table.

CO_T:

Name	Data Type	Size	Remark
COID	varchar	4	This is the primary key to understanding this relationship. The CO ID is contained in this.
COName	varchar	15	This includes the CO Name.
CODescription	varchar	100	This section contains a description of CO.
courseID	varchar	7	This is a foreign key from COURSE's table.
PLOID	varchar	4	This is a foreign key from PLO Table.

Prerequisitecourse_T:

Name	Data Type	Size	Remark
courseID	varchar	7	This is the relation's composite key. The course id is contained in this. As an example, consider the term 'CSE303.'
prerequisitecourseID	varchar	7	This is the relation's composite key. This contains the prerequisite course's course id.

Assessment_T:

Name	Data Type	Size	Remark
assesID	varchar	7	This is the primary key to understanding this relationship. The assessment id is contained in this field.
courseID	varchar	7	This is a foreign key from COURSE's table.
sectionNo	Number	2	This is a foreign key from SECTION's table.
COID	varchar	4	This is a foreign key from CO's table.
assessType	varchar	10	This section includes the assessment types.
totalachievablemarks	Number	3	This section contains the total number of marks that can be earned.

Evaluation_T:

Name	Data Type	Size	Remark
StudentID	varchar	7	This is the relation's composite key. This field contains the student's ID.
assesID	varchar	7	This is the relation's composite key. The assessment ID is contained in this field.
totalobtainedmarks	FLOAT	3	This section contains the total number of marks earned.

Evaluationfaculty_T:

Name	Data Type	Size	Remark
FaccountID	varchar	4	This is the relation's composite key. This section contains the faculty ID.
StudentId	varchar	7	This is the relation's composite key. This field contains the student's ID.
assesID	varchar	7	This is the relation's composite key. The assessment id is contained in this field.
evaluationTime	DateTime	“dd-mm-yyyy, hh:mm:ss”	This section includes the evaluation time.

CHAPTER 4: PHYSICAL SYSTEM DESIGN

SECTION 4.1.1: ADD NEW USER - INPUT FORMS

PURPOSE

The purpose of adding users is to allow them to use the system. As a result, without the addition of any types of users, this system would not exist because it is dependent on storing and retrieving data.

RELATED SQL USED

```
php > add-user.php
1  <?php
2      require 'mysql.php';
3      // geting post requests
4      $id = $_POST['id'];
5      $fName = $_POST['fName'];
6      $lName = $_POST['lName'];
7      $program = $_POST['program'];
8      $email = $_POST['email'];
9      $password = $_POST['password'];
10     $role = strtolower($_POST['role']);
11
12     $store = "INSERT INTO user (id, f_name, l_name, program_id, email, password, role) VALUES
13         ('$id', '$fName', '$lName', '$prog', '$email', '$password', '$role)';
14
15     if($mysql->query($store)){
16         header("Location: ../admin/add-user.php");
17     }
18     else{
19         header("Location: ../admin/add-user.php");
20     }
21
22
23  ?>
```

SECTION 4.1.2: ADD NEW COURSE WITH CO - INPUT FORMS

PURPOSE

Each course must be added to the system along with their respective CO's. To do so, an input form must be used to provide input to the system.

RELATED SQL USED

```

php > add-course.php
1 <?php
2     require 'mysql.php';
3
4     $id = $_POST['course_id'];
5     $program_id = $_POST['program_id'];
6     $credit = $_POST['credit'];
7     $total_co = $_POST['total-co'];
8     $title = $_POST['course_title'];
9
10    $sql = "INSERT INTO course (id, program_id, title, credit, total_co) VALUES
11           ('$id', '$program_id', '$title', '$credit', '$total_co)";
12    if($mysql->query($sql) == FALSE){
13        header("Location: ../admin/add-course.php?failed=1");
14    }
15
16    for($i=1; $i<=15; $i++){
17        if(isset($_POST["plo-co".$i])){
18            $sql = "SELECT sl FROM plo WHERE program_id = '$program_id' AND plo_no = $i";
19            $plo_id = $mysql->query($sql)->fetch_assoc()['sl'];
20            $data = $_POST["plo-co".$i];
21            $field = ""; $val = "";
22            foreach($data as $co){
23                $field .= 'co'. $co . ', ';
24                $val .= '1, ';
25            }
26            $sql = "INSERT INTO co (course_id, plo_id, ".substr($field, 0, -2).") VALUES ('$id', '$plo_id, ".substr($val, 0, -2).")";
27            if($mysql->query($sql) == FALSE){
28                header("Location: ../admin/add-course.php?failed=1");
29            }
30        }
31    }
32
33    header("Location: ../admin/add-course.php?success=1");
34
35    ?>

```

SECTION 4.1.3: ADD NEW PROGRAM WITH PLO - INPUT FORMS

PURPOSE

Because this system tracks PLO achievement of students in any course in any program, we must associate the PLO data with the appropriate program. Because the system cannot collect data automatically, the programs with their PLOs are manually entered using an input form.

RELATED SQL USED

```

php > add-program.php
1 <?php
2     require 'mysql.php';
3
4     $id = $_POST['program_id'];
5     $program_name = $_POST['program_name'];
6     $school = $_POST['school'];
7     $sql = "INSERT INTO program(id, program_name, school) VALUES ('$id', '$program_name', '$school)";
8     //echo $sql;
9     if($mysql->query($sql) == FALSE){
10         header("Location: ../admin/add-program.php?failed=1");
11     }
12
13     $i = 1;
14
15     while(isset($_POST['title'.$i])){
16         $name = $_POST['title'.$i];
17         $sql = "INSERT INTO plo(program_id, plo_no, plo_name) VALUES
18              ('$id', $i, '$name)";
19         if($mysql->query($sql) == FALSE){
20             header("Location: ../admin/add-program.php?failed=1");
21         }
22         $i++;
23     }
24
25     header("Location: ../admin/add-program.php?success=1");
26
27    ?>

```

SECTION 4.1.4: ENTER MARKS - INPUT FORMS

PURPOSE

This input form necessitates a marks entry process in order for a student's data to be entered based on a course in a specific semester. This is then saved in the database and can be retrieved as needed. The Marks Entry Form allows you to enter all of a student's assessments.

RELATED SQL USED

```
php > add-marks.php
1  <?php
2      require 'mysql.php';
3
4      $student_id = $_POST['student_id'];
5      $course_id = $_POST['course_id'];
6      $exam_name = $_POST['exam_name'];
7      $semester = $_POST['semester'];
8      $section = $_POST['section'];
9
10     $field = "";
11     $val = "";
12     $i=1;
13
14     while(isset($_POST['co'.$i])){
15         $field .= 'q'.$i.'_mark, ' . 'q'.$i.'_co, ' . 'q'.$i.'_max, ' ;
16         $val .= $_POST['mark'.$i].', ' . $_POST['co'.$i].', ' . $_POST['max'.$i].', ' ;
17         $i++;
18     }
19
20     $sql = "INSERT INTO marks(student_id, course_id, exam_name, semester, section, ".substr($field, 0, -2).") VALUES
21         ('$student_id', '$course_id', '$exam_name', '$semester', '$section', ".substr($val, 0, -2).")";
22
23     echo $sql . '<br>';
24     $mysql->query($sql);
25
26
27     header("Location: ../faculty/entry-marks.php?success=1");
28
29  ?>
```


SECTION 4.2.1: PROGRESS VIEW - OUTPUT QUERY AND REPORTS

PURPOSE AND USE

It includes views of the student and course progress. It displays the number of PLOs expected to be achieved and the number of PLOs actually achieved for each student at the end of each semester. When you select a course, it displays the number of students enrolled in that course as well as the percentages of CO achieved and failed.

RELATED SQL USED

```
php > progress-views.php
1  <?php
2  require 'mysql.php';
3  if(isset($_GET['id'])){
4      $id = $_GET['id'];
5      $sql = "SELECT DISTINCT semester FROM marks WHERE student_id = $id";
6      $ssem = $mysql->query($sql);
7
8      $ploProg = array();
9      foreach($ssem as $sem){
10         $res = seeker($sem['semester'], $id);
11         $ploProg[$sem['semester']]['total'] = $res['p'];
12         $ploProg[$sem['semester']]['completed'] = $res['t'];
13     }
14
15 }if(isset($_GET['c'])){
16     $crs = $_GET['c'];
17     $sql = "SELECT DISTINCT student_id FROM marks WHERE course_id = '$crs'";
18     $sList = $mysql->query($sql);
19     $totalS = $sList->num_rows;
20     $report = array();
21     foreach($sList as $s){
22         $usr = $s['student_id'];
23         $ret = seeker2($crs, $usr);
24         foreach($ret['co'] as $i => $j){
25             if($j==1){
26                 if(isset($report["co"][$i])){
27                     $report["co"][$i]++;
28                 }else{
29                     $report["co"][$i] = 1;
30                 }
31             }
32         }
33         foreach($ret['plo'] as $i => $j){
34             if($j==1){
35                 if(isset($report["plo"][$i])){
36                     $report["plo"][$i]++;
37                 }else{
38                     $report["plo"][$i] = 1;
39                 }
40             }
41         }
42     }
43     ksort($report["co"]);
44     ksort($report["plo"]);
45 }
46
47 function seeker($sem, $uid){
48     require 'mysql.php';
49     $sql;
50     if($sem!="null"){
51         $sql = "SELECT * FROM marks WHERE student_id = $uid AND semester = '$sem'";
52     }else{
53         $sql = "SELECT * FROM marks WHERE student_id = $uid";
54     }
55
56     $sMarks = $mysql->query($sql);
57
58     $cMarks = array();
59     $cTotal = array();
60 }
```

SECTION 4.2.2: PLO ACHIEVEMENT – OUTPUT QUERY AND REPORTS

PURPOSE AND USE

It is used to display student-specific PLO analysis, which includes the PLO total percentage score for each PLO calculated from the scores achieved in each CO associated with the corresponding PLO across all courses completed by the student. After entering a student ID, a course-by-course PLO analysis can be viewed. A tabular view of PLO achievement by student is also available.

RELATED SQL USED

```

php > plo-achievement.php
1  <?php
2  require 'mysql.php';
3  if(isset($_GET['id'])){
4      $id = $_GET['id'];
5      $sql = "SELECT * FROM marks WHERE student_id = $id";
6      $sMarks = $mysql->query($sql);
7
8      //course based total co marks
9      $cMarks = array();
10     $cTotal = array();
11     foreach($sMarks as $marks){
12         $course = $marks['course_id'];
13         for($i=1; $i<=10; $i++){
14             if(isset($marks["q".$i."_co"]) && $marks["q".$i."_co"]!=0){
15                 $sco = $marks["q".$i."_co"];
16                 if(isset($cMarks[$course][$sco])){
17                     $cMarks[$course][$sco] += $marks["q".$i."_mark"];
18                     $cTotal[$course][$sco] += $marks["q".$i."_max"];
19                 }else{
20                     $cMarks[$course][$sco] = $marks["q".$i."_mark"];
21                     $cTotal[$course][$sco] = $marks["q".$i."_max"];
22                 }
23             }
24         }
25     }
26
27     $pMarks = array();
28     $pTotal = array();
29     foreach($cMarks as $c => $v){
30
31         $sql = "SELECT * FROM co WHERE course_id = '$c'";
32         $plos = $mysql->query($sql);
33         foreach($plos as $plo){
34             $pId = $plo['plo_id'];
35             for($i=1; $i<=10; $i++){
36                 if(isset($plo["co".$i]) && $plo["co".$i]==1){
37                     if(isset($pMarks[$c][$pId])){
38                         $pMarks[$c][$pId] += $cMarks[$c][$i];
39                         $pTotal[$c][$pId] += $cTotal[$c][$i];
40                     }else{
41                         $pMarks[$c][$pId] = $cMarks[$c][$i];
42                         $pTotal[$c][$pId] = $cTotal[$c][$i];
43                     }
44                 }
45             }
46         }
47     }
48
49     //total marks in plo
50     $pfMarks = array();
51     $pfTotal = array();
52     foreach($pMarks as $c => $v){
53         foreach($v as $i => $j){
54             if(isset($pfMarks[$i])){
55                 $pfMarks[$i] += $j;
56                 $pfTotal[$i] += $pTotal[$c][$i];
57             }else{
58                 $pfMarks[$i] = $j;
59                 $pfTotal[$i] = $pTotal[$c][$i];
60             }

```

SECTION 4.2.3: STUDENT RESULT - OUTPUT QUERY AND REPORTS

PURPOSE AND USE

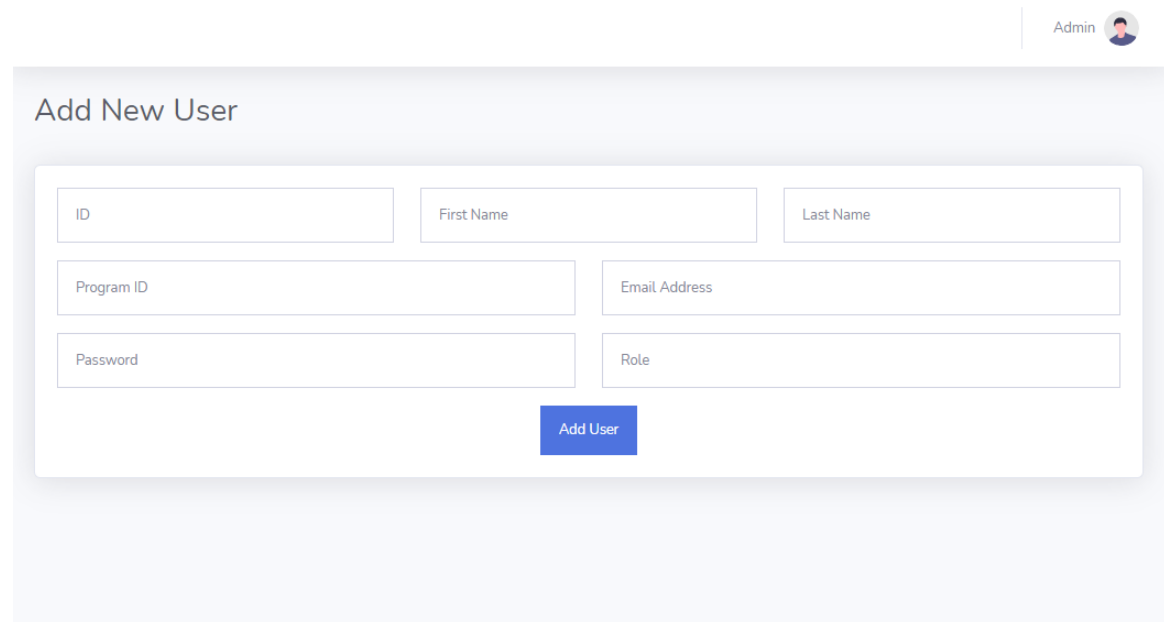
When the student's id is entered, it displays the student's PLO achievement results in a pie chart for all of the courses that student has completed.

RELATED SQL USED

```

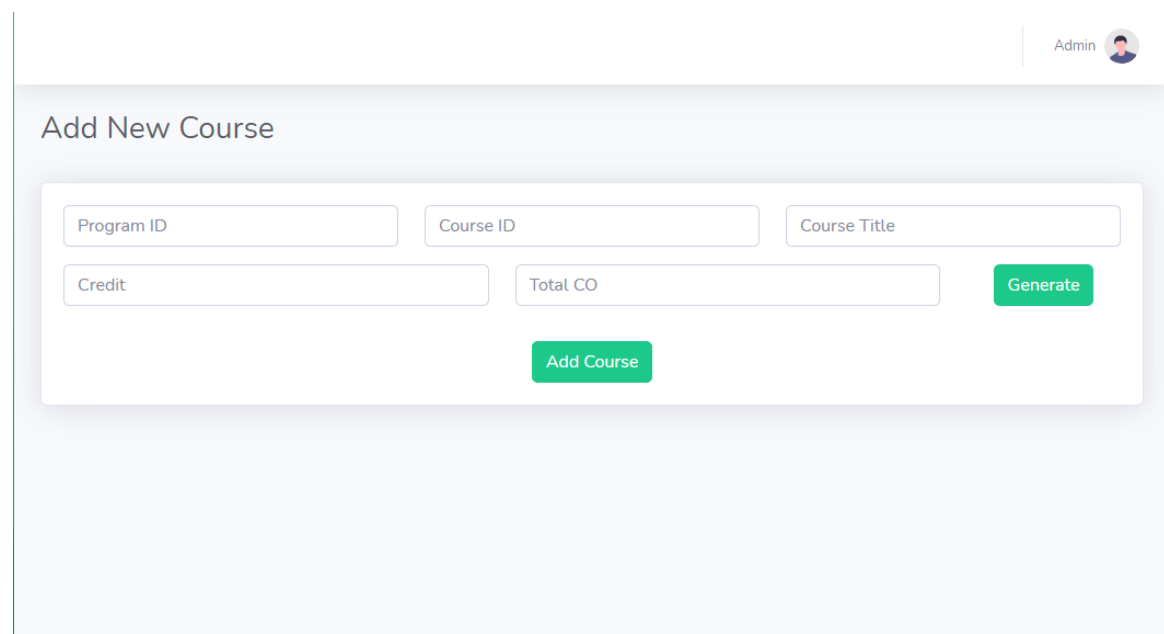
php > result.php
1  <?php
2      require '../php/mysql.php';
3
4      $color = ["", "#1FE7C4", "#E45C17", "#06B97B", "#8CE026", "#E1CCFF", "#5BA2CC", "#0A2E82", "#957107", "#80CF18"];
5      // session_start();
6      $id = $_SESSION['user_id'];
7      $sql = "SELECT * FROM user WHERE id = $id";
8      $uInfo = $mysql->query($sql)->fetch_assoc();
9
10
11     $sql = "SELECT * FROM plo WHERE program_id = '". $uInfo['program_id']."'";
12     $totalPlo = $mysql->query($sql)->num_rows;
13
14     $sql = "SELECT * FROM marks WHERE student_id = $id";
15     $sMarks = $mysql->query($sql);
16
17     //course based total co marks
18     $cMarks = array();
19     $cTotal = array();
20     foreach($sMarks as $marks){
21         $course = $marks['course_id'];
22         for($i=1; $i<=10; $i++){
23             if(isset($marks["q".$i."_co"]) && $marks["q".$i."_co"]!=0){
24                 $co = $marks["q".$i."_co"];
25                 if(isset($cMarks[$course][$co])){
26                     $cMarks[$course][$co] += $marks["q".$i."_mark"];
27                     $cTotal[$course][$co] += $marks["q".$i."_max"];
28                 }else{
29                     $cMarks[$course][$co] = $marks["q".$i."_mark"];
30                     $cTotal[$course][$co] = $marks["q".$i."_max"];
31                 }
32             }
33         }
34     }
35
36     $pMarks = array();
37     $pTotal = array();
38     foreach($cMarks as $c => $v){
39
40         $sql = "SELECT * FROM co WHERE course_id = '$c'";
41         $plos = $mysql->query($sql);
42         foreach($plos as $plo){
43             $pId = $plo['plo_id'];
44             for($i=1; $i<=10; $i++){
45                 if(isset($plo["co".$i]) && $plo["co".$i]==1){
46                     if(isset($pMarks[$c][$pId])){
47                         $pMarks[$c][$pId] += $cMarks[$c][$i];
48                         $pTotal[$c][$pId] += $cTotal[$c][$i];
49                     }else{
50                         $pMarks[$c][$pId] = $cMarks[$c][$i];
51                         $pTotal[$c][$pId] = $cTotal[$c][$i];
52                     }
53                 }
54             }
55         }
56     }
57
58     ?>

```

SECTION 4.3: SYSTEM DESIGN ARCHITECTURE


The 'Add New User' form is located in the top right corner of the page, indicated by a user profile icon and the name 'Admin'. The form is titled 'Add New User' and contains several input fields: 'ID', 'First Name', 'Last Name', 'Program ID', 'Email Address', 'Password', and 'Role'. A blue 'Add User' button is positioned at the bottom center of the form.

Figure 8: Add New User UI



The 'Add New Course' form is located in the top right corner of the page, indicated by a user profile icon and the name 'Admin'. The form is titled 'Add New Course' and contains several input fields: 'Program ID', 'Course ID', 'Course Title', 'Credit', and 'Total CO'. A green 'Generate' button is positioned to the right of the 'Total CO' field. A green 'Add Course' button is positioned at the bottom center of the form.

Figure 9: Add New Course with CO UI

Admin 

Add New Program

PLO 1





Figure 10: Add New Program with PLO UI

Faculty 

Entry Marks




Figure 11: Enter Marks UI

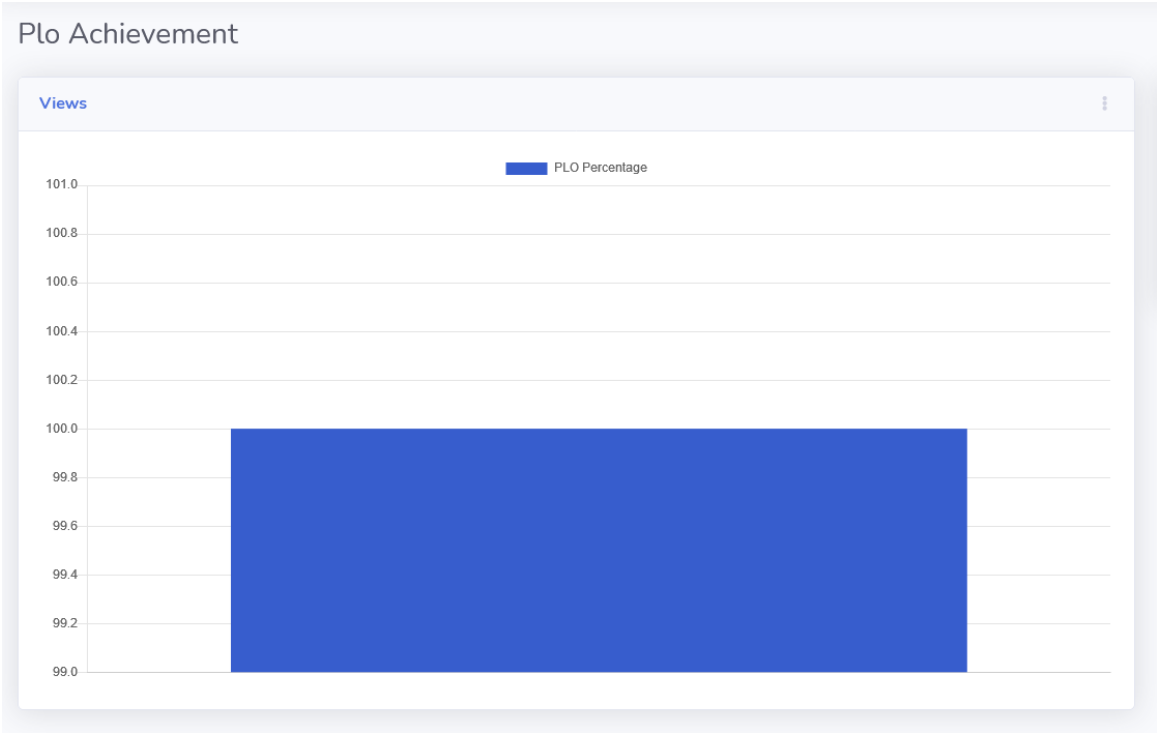


Figure 12: PLO Achievement View UI

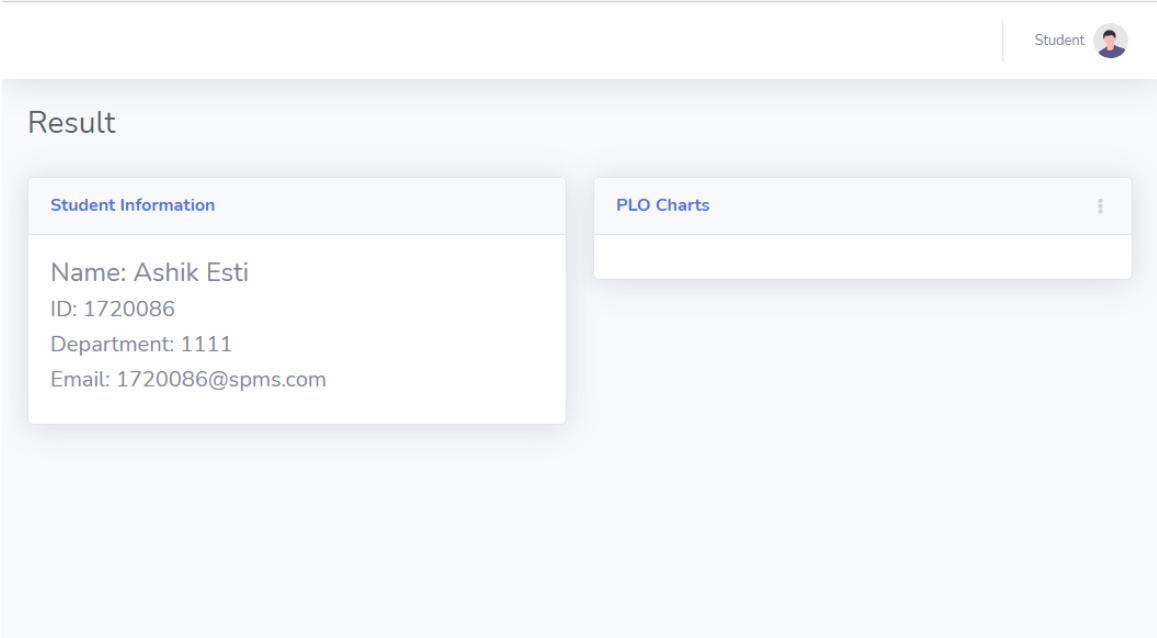


Figure 13: Student Result UI

CHAPTER 5: CONCLUSION

SECTION 5.1: PROBLEM & SOLUTION

There were some problems that we have faced while creating the Student Performance Monitoring System. The major issue was we had lack of knowledge on the languages such as (PHP, CSS, JAVASCRIPT, HTML, SQL) that we must use while creating the system. We came across this problem by seeking help from our faculty members, YouTube, Google searches and friends through email, using meet or by watching video sessions. We had issues on using GitHub which was a new platform for us. So, we tried doing some internet searches and gaining information ourselves and using it properly. We are facing technical difficulties to showing char in the project. We might overcome if we got more day. Hopefully on upcoming days we will find solution for this technical problem.

SECTION 5.2: ADDITIONAL FEATURES AND FUTURE DEVELOPMENT

The current proposed device does now no longer includes a system with which we are able to track a failed PLO that had been formerly accomplished in a selected course. Moreover, the device may be made greater steady through including two-aspect authentication in order that most effective precise customers can get admission to the statistics saved withinside the application. An extra AI characteristic may be covered to make it less difficult in inputting the statistics through giving voice input in place of typing.

SECTION 5.3: CONCLUSION & RECOMMENDATION

After concluding requirement analysis, logical system design we took some analytical decision and made certain documentation at first then finally we developed the entire structure of the project using physical design and thus came up with a suitable Student performance monitoring system (SPMS) project.

CONTRIBUTION OF EACH MEMBER

	Md Ashikuzzama n Esti	Shahin Sikder	Md. Shahim Uddin Saba	Md. Tuhin Mia
Cover Page			✓	
Table of Contents			✓	
List of Figures			✓	
Background of the Organization	✓	✓	✓	✓
Background of the Project	✓	✓	✓	✓
Objectives of the Project	✓	✓	✓	✓
Scope of the Project	✓	✓	✓	✓
Existing Rich Picture	✓	✓	✓	✓
Existing Six Element System Analysis	✓	✓	✓	✓
Existing BPMN	✓	✓	✓	
Existing Problems and Analysis of the Problems	✓	✓	✓	✓
Proposed Rich Picture	✓	✓	✓	✓
Proposed Six Element System Analysis	✓	✓	✓	✓
Proposed BPMN	✓	✓	✓	✓
Business Rules	✓	✓	✓	✓
ERD	✓	✓	✓	
Relational Schema	✓	✓	✓	✓
Normalization	✓	✓	✓	✓

Data Dictionary	✓	✓	✓	✓
Input Forms - Purpose	✓	✓	✓	
Output Query and Reports – Purpose and Use	✓	✓	✓	
Input Forms – Related SQL Used	✓	✓	✓	
Output Query and Reports – Description along with SQL	✓	✓	✓	
System Design Architecture	✓	✓	✓	✓
Problem & Solution	✓	✓	✓	✓
Additional Features and Future Development	✓	✓	✓	✓
Conclusion and Recommendation	✓	✓	✓	✓

REFERENCES

- [1] Wikipedia – Independent University, Bangladesh
Available: https://en.wikipedia.org/wiki/Independent_University,_Bangladesh
- [2] “W3Schools”
Available: <https://www.w3schools.com>