

STUDENT PERFORMANCE MONITORING SYSTEM

Final Report

Section: 2

Faculty Name: Ms. Sadita Ahmed

NAME	ID
Md Navidul Islam	1821952
Tiasha Swarnaker	1820874
Abdullah Al-Noman	1630081
Sumiya Afrin	1711012
Sabrina Yasmin	1821172
Md. Faysal Chowdhury	1721071

Table of Contents

Chapter 1: Introduction	4
Section 1.1: Background of the Organization	4
Section 1.2: Background of the Project.....	4
Section 1.3: Objectives of the Project	4
Section 1.4: Scope of the Project	5
Chapter 2: Requirement Analysis	6
Section 2.1: Rich Picture (Existing).....	6
Section 2.2: Six Elements System Analysis (Existing).....	7
Section 2.3: Process Diagram (Existing).....	13
Section 2.4: Existing Problems & Analysis of the Problem	14
Section 2.5: Rich Picture (Proposed)	16
Section 2.6: Six Element System Analysis (Proposed).....	17
Section 2.7: Process Diagram (Proposed)	23
Chapter 3: Logical System Design	24
Section 3.1: Business Rules	24
Section 3.2: ERD.....	26
Section 3.3: ERD to Relations.....	27
Section 3.4: Normalization.....	28
Section 3.5: Data Dictionary	29
Chapter 4: Physical System Design	34
Section 4.1.1: Input Forms – Add User.....	34
Purpose	34
Related SQL Used	34
Section 4.1.2: Input Forms – Add Program and PLO	34
Purpose	34
Related SQL Used	35
Section 4.1.3: Input Forms – Add Course with CO	35
Purpose	35
Related SQL Used	35
Section 4.1.4: Input Forms – Add Marks Individual.....	36
Purpose	36
Related SQL Used	36
Section 4.1.5: Input Forms – Add Marks Mass	37
Purpose	37
Related SQL Used	37
Section 4.2.1: Output Query and Reports – Higher Management Dashboard	39

Purpose and Use	39
Description along with SQL.....	39
Section 4.2.2: Output Query and Reports – PLO Achievement.....	39
Purpose and Use	39
Description along with SQL	39
Section 4.2.3: Output Query and Reports – Progress View	41
Purpose and Use	41
Description along with SQL.....	41
Section 4.2.4: Output Query and Reports – Student Result.....	45
Purpose and Use	45
Description along with SQL.....	45
Section 4.3: System Design Architecture.....	46
Chapter 5: Conclusion.....	52
Section 5.1: Problem & Solution.....	52
Section 5.2: Additional Features and Future Development	52
Section 5.3: Conclusion & Recommendation	52
Contribution of Each Member	53
References.....	54
 Figure 1: Existing System Rich Picture	6
Figure 2: Existing System BPMN.....	13
Figure 3: Proposed System Rich Picture	17
Figure 4: Proposed System BPMN	23
Figure 5: Entity Relationship Diagram	26
Figure 6: Relational Schema Diagram	27
Figure 7: Normalization Diagram	28
Figure 8: Add User.....	46
Figure 9: Add Program and PLO	47
Figure 10: Add Course with CO	47
Figure 11: Add Marks	48
Figure 12: Higher Management Dashboard.....	48
Figure 13: PLO Achievement View 1	49
Figure 14: PLO Achievement View 2	49
Figure 15: PLO Achievement View 3	50
Figure 16: Progress View 1.....	50
Figure 17: Progress View 2.....	51
Figure 18: Student Result.....	51

CHAPTER 1: INTRODUCTION

SECTION 1.1: BACKGROUND OF THE ORGANIZATION

Independent University, Bangladesh (IUB) established in 1993 is the leading private university in Bangladesh with an explicit focus on Research and Global partnerships. IUB's mission is to achieve the goals of higher education and of sustainable economic growth in the country through a two-way relationship between community and university. Its goals are to produce graduates of international standards within the local environment, with knowledge and relevant skills to provide leadership in enterprise, public service and welfare; encourage and support useful research; create knowledge; and provide further learning opportunities.

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

IUB is rapidly expanding its portfolio and is in the process of introducing Architecture and Biotechnology. The University curriculum and course of study are progressively revised and adjusted on the basis of their relevance to national needs and the global market demand. [1]

SECTION 1.2: BACKGROUND OF THE PROJECT

The Student Performance Monitoring System focuses on performance monitoring of student's continuous assessment (tests) and examination scores in order to predict their final achievement status upon graduation. The main idea is to evaluate the COs achieved and mapped PLOs achieved by each student in each of the enrolled courses as that would be necessary for monitoring the student performance.

SECTION 1.3: OBJECTIVES OF THE PROJECT

- One of the goals of this Project is to provide insight about how learning might improve in a given program-whether it be online, in a classroom, or happening in another context.
- To provide insight into what students are actually 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 so as 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

Scope of the project is a necessity to ensure the success of a project. As we are changing an existing system, we have to ensure that the proposed system will be more effective than the existing one. The proposed system would include evaluating the COs achieved, mapping the COs with the PLOs achieved and storing them as records, all of these were done manually in the existing system. The records can also be used to generate reports for analysis purpose. The system can be accessed by the instructors, students, UGC, IEB and Higher Management (VC, Deans, Heads). It is very inefficient to maintain detailed records of student's performance, and therefore there is a need of an improved and automated student performance monitoring system. Primarily we focused on IUB as the organization for which we are doing this project but the project has the potential of being useful to UGC/IEB as well, and future prospects seems likely to also cater to all universities that conform to OBE regulations.

CHAPTER 2: REQUIREMENT ANALYSIS

SECTION 2.1: RICH PICTURE (EXISTING)

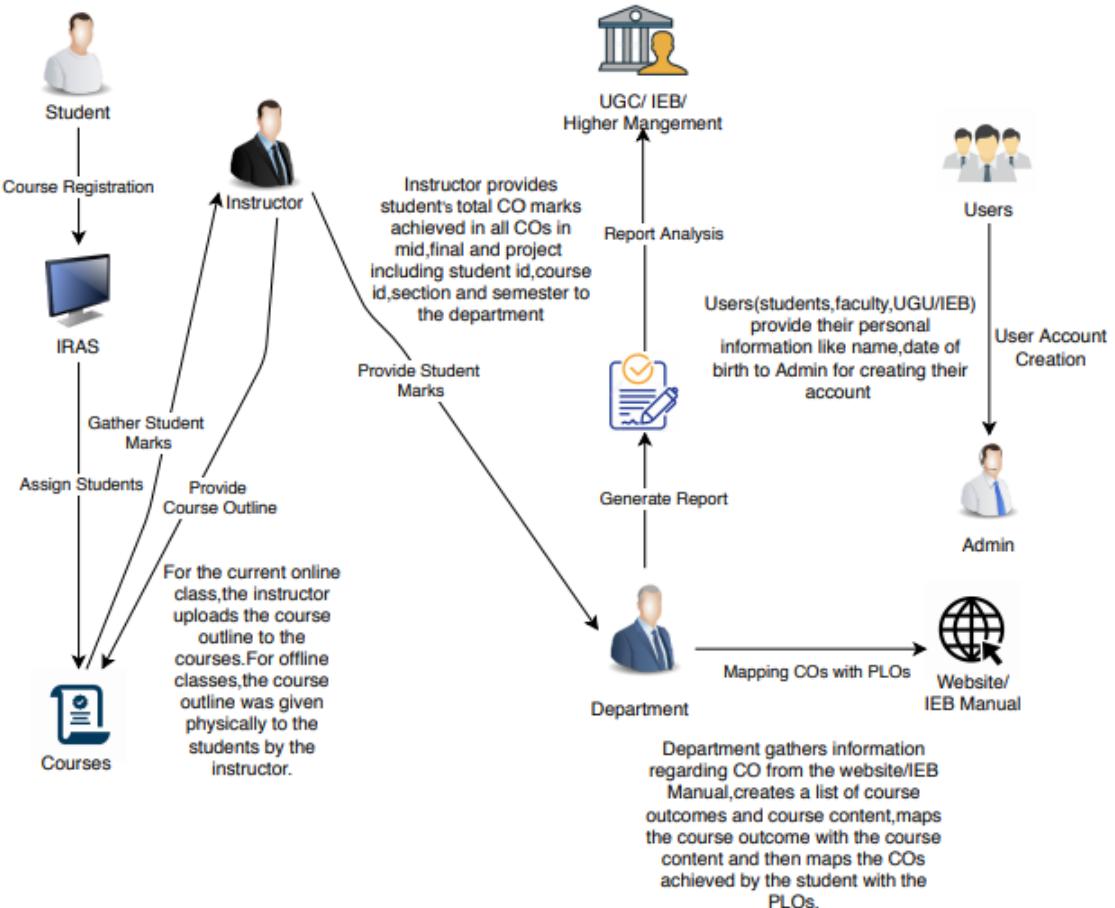


Figure 1: Existing System Rich Picture

In the existing business system, users create account for new users of the system by collecting user's name, DOB and assigns a certain id and password. IEB/UGC sends accreditation manual with PLO's defined to the department and then the department gathers CO from the corresponding PLO. Department creates a list of the course content if it is not available in the course outline and a list of course outcomes and maps the CO with the course content. The department then maps the received CO's to specific PLO's and hence mapping the CO's to specific questions of mid-term, final & project. For the current online system, the instructor provides the course outline to online classroom and for the physical classes, course outline is printed and distributed among the students. The instructor receives CO's to be achieved in a particular course and takes exams for assessing students in various course outcomes. Hence, the instructor converts the total marks and calculate total for all the CO's. The instructor sends the CO's marks to the department. The department calculates CO percentages and if the percentage is greater than or equal to 40% a student passes that specific PLO and hence otherwise fails. The corresponding CO's are mapped

against PLO's and PLO achievements are recorded. In order to generate a report, the department retrieves the student information of PLO achievement from previously calculated OBE mark sheet and makes report based on the requirements of UGC/ IEB/ Higher management.

SECTION 2.2: SIX ELEMENTS SYSTEM ANALYSIS (EXISTING)

Process	System Roles					
	Human	Non-Computing Hardware	Computing Hardware	Software	Database	Network & Communication
User Account Creation	<p>Admin:</p> <ul style="list-style-type: none"> 1. Can create new users for the system. 2. Collect user information such as name, DOB based on their specific role. 3. Assigns a certain id and password for each individual user type. <p>Users:</p> <ul style="list-style-type: none"> 1. Provide personal information such as name, DOB to the admin for their specific role. 	<p>Pen & Papers:</p> <ul style="list-style-type: none"> 1. The information sometimes are recorded manually. 	<p>Computer:</p> <ul style="list-style-type: none"> 1. All related data is searched and stored using computer. 	<p>MS Excel:</p> <ul style="list-style-type: none"> 1. All related information are stored. 	<p>Other Sources:</p> <ul style="list-style-type: none"> 1. All related information are stored in the specific location. 	None.
Mapping COs with PLOs	<p>Department:</p> <ul style="list-style-type: none"> 1. IEB/UGC sends accreditation manual with PLO's defined to the department 	None.	<p>Computer:</p> <ul style="list-style-type: none"> 1. All related data is searched and stored using computer. 	None.	<p>Other Sources:</p> <ul style="list-style-type: none"> 1. All related information are stored in the 	All related data are searched through the internet.

	<p>2. Department then gathers information regarding Course Outcome from the website/ IEB manual.</p> <p>3. If the course content exist in the course outline, review the course content else create a list of the course content.</p> <p>4. Creates a list of the course outcomes.</p> <p>5. Mapping the CO with the course content.</p> <p>6. Mapping the received CO's to specific PLOs.</p> <p>7. Mapping CO's to specific questions of mid-term, final & project.</p>				specific location.	
Provide Course Outline	<p>Instructor:</p> <p>1. For the current online class, Course Outline is uploaded to the online classroom (Google Classroom).</p> <p>2. For offline classes, Course Outline</p>	None.	<p>Computer:</p> <p>1. All related data is searched and stored using computer.</p>	None.	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	None.

	<p>is printed and distributed among the students.</p> <p>3. At the end of the semester, Course Outline is submitted to the department.</p>					
Course Registration	<p>Student:</p> <ol style="list-style-type: none"> 1. On the reserved day for registration log in to IRAS with their id and password. 2. Go to registration page, load courses and select courses. Can only select courses of which prerequisites have been completed and the course capacity is not exceeded. 3. Click on 'DONE' button to complete the registration process. 4. A message prompts saying "Registration Successfully Done". 	<p>Pen & Paper:</p> <ol style="list-style-type: none"> 1. Sometimes course registration is done manually. 	<p>Computer:</p> <ol style="list-style-type: none"> 1. All related data is searched and stored using computer. 	<p>IRAS:</p> <ol style="list-style-type: none"> 1. IRAS has an interface for doing registration. 	<p>MS SQL Server:</p> <ol style="list-style-type: none"> 1. IRAS is integrated with MS SQL Server. <p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information are stored in the specific location. 	<p>Internet:</p> <p>IRAS is a web-based application and requires internet to access.</p>
Gather Student Marks from	<p>Instructor:</p> <ol style="list-style-type: none"> 1. Receives CO's to be 	<p>Pen & Paper:</p> <ol style="list-style-type: none"> 1. 	<p>Computer:</p> <ol style="list-style-type: none"> 1. All related data 	<p>MS Excel:</p> <ol style="list-style-type: none"> 1. All related 	<p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related 	None.

Courses they have enrolled	<p>achieved in that particular course.</p> <p>2. Takes exams such as mid-term, final & project to assess students in various course outcomes.</p> <p>3. Gather marks for different assessment including student id, course id, section, semester.</p> <p>4. Convert the total marks for mid-term, final.</p> <p>5. Calculate total for all the CO's.</p>	<p>1. Written exams conducted by the faculty to evaluate the students.</p>	<p>such as student name, id is searched and stored using computer.</p>	<p>information are stored.</p>	<p>information are stored in the specific location.</p>	
Provide Student Marks	<p>Instructor:</p> <p>1. Provides students total CO marks achieved in all CO's in mid-term, final & project to the department including student id, course id, section, semester.</p> <p>Department:</p> <p>1. Gather students marks in all CO's achieved based</p>	<p>None.</p>	<p>Computer:</p> <p>1. All related data is searched and stored using computer.</p>	<p>MS Excel:</p> <p>1. All related information are stored.</p>	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	<p>All related data are provided through the internet.</p>

	<p>on mid-term, final & project with student id, course id, section, semester.</p> <p>2. Calculates CO percentages.</p> <p>3. If greater than or equal to 40% CO's have been achieved, a student passes that certain CO otherwise fails.</p> <p>4. The corresponding CO's are mapped against PLO's and PLO achievements are recorded.</p>					
Generate Report	<p>Department:</p> <p>1. Retrieve the student information of PLO achievement from previously calculated OBE marksheet.</p> <p>2. Make reports after comparing results of multiple students.</p> <p>3. Percentage of successfully passed or failed to</p>	None.	<p>Computer:</p> <p>1. All related data is searched and stored using computer.</p>	<p>MS Excel:</p> <p>1. All related information are stored and calculated.</p>	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	None.

	achieve are calculated based on the total number of students.					
Report Analysis	UGC/ IEB/ Higher Management: Does analysis by 1a. Viewing number of students passing or failing in a certain PLO. 1b. Viewing progression of students with charts and graphs.	Paper: 1. Paper is required for the printing purpose.	Printer: 1. Print the report based on users' request.	None.	Other Sources: 1. All related information are stored in the specific location.	None.

SECTION 2.3: PROCESS DIAGRAM (EXISTING)

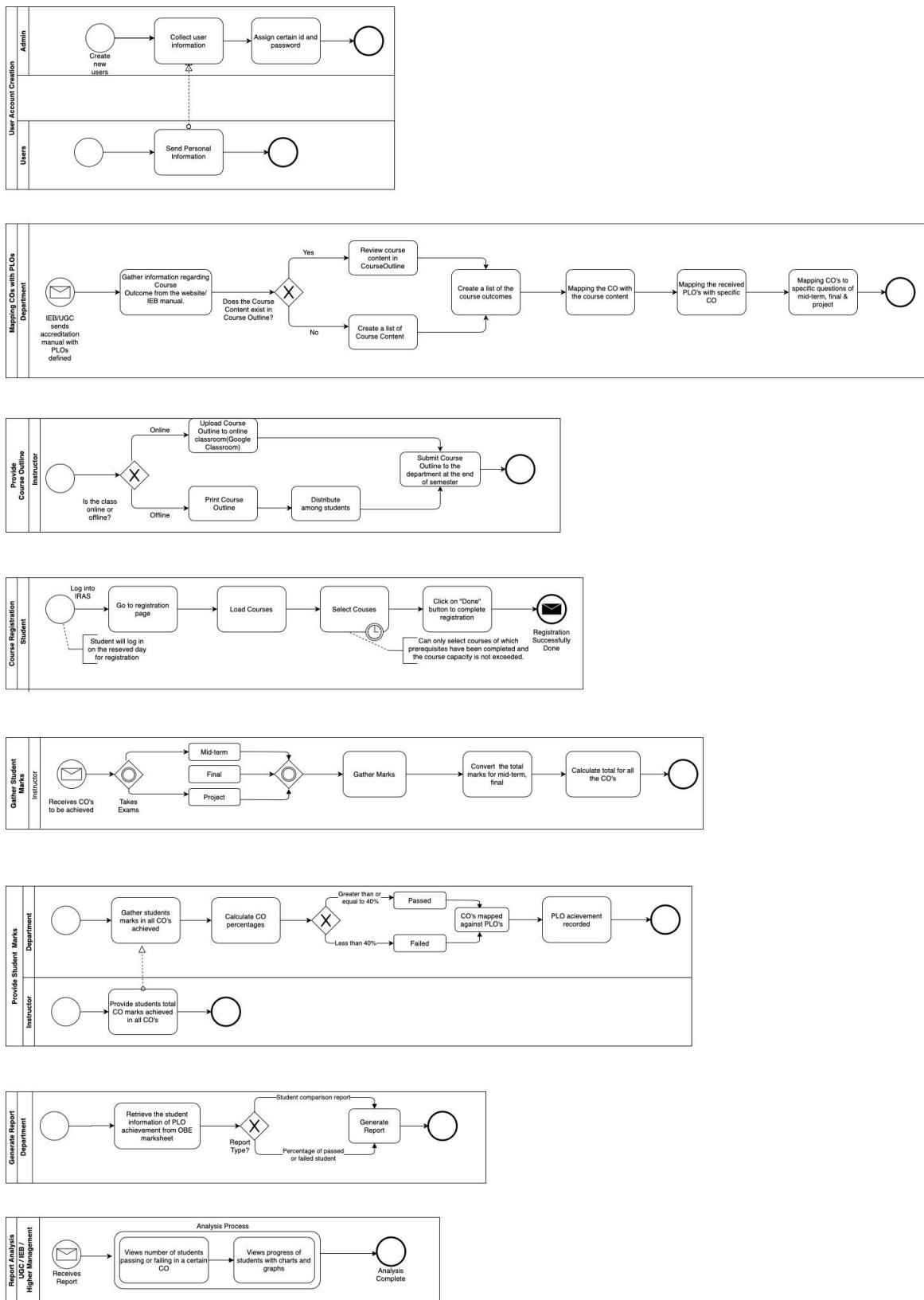


Figure 2: Existing System BPMN

SECTION 2.4: EXISTING PROBLEMS & ANALYSIS OF THE PROBLEM

Process Name	Stakeholders	Concerns (Issues/Problems)	Analysis (Reason of the Problems)	Proposed Solution
User Account Creation	Users	The users name might be similar.	There might be several persons with the same name.	In addition to the name and DOB the user provides for creating an account, they must also provide their contact numbers since contact numbers are unique and all these information must be stored in a system.
Mapping CO's with PLO's	Department	1. Manually creates a list of the course content if it does not exist in course outline. 2. Manually creates a list of the course outcomes. 3. Manually mapped the CO's with the course content. 4. Manually mapping CO's to PLO's.	All the processes for mapping CO's with PLO's are manually done by the department since there is no such system which can do the work without any manual task.	There can be a system which will gather the course content and course outcome from the department and then map the CO's with the course content and hence mapping the received CO's to specific PLO's.
Course Registration	Student	1. Course capacity gets filled up due to which students cannot register on the day they intended to. 2. Server gets jammed.	There is a specific capacity for any particular course and if the capacity exceeds then students cannot enroll in that course. During registration	Increasing the capacity of courses automatically as soon as it gets filled so that there is no delay in the registration process. There can be an information system where the load distribution is

			system cannot take huge load so it creates delay in the process.	handled properly when there is extensive load.
Gather Student Marks from courses they have enrolled	Instructor	<p>1. Instructor gathers students marks from courses they have enrolled by taking assessments (mid, final, project) and converts the total marks for mid-term, final, project manually using MS Excel.</p> <p>2. Using MS Excel calculate total for all the CO's.</p>	<p>The process of converting marks for each student in each of the courses is very inefficient and takes a lot of time.</p> <p>There is a chance of manual error as well during input or when calculating the marks.</p>	A system can be introduced which can take total marks for mid-term, final as an input and convert it and hence automatically calculate total for all the CO's.
Provide Student Marks	Instructor	<p>.Student marks that are provided to the department by the instructor are calculated manually. CO percentages are calculated manually using MS Excel.</p> <p>2. Manually checking if a student passed or failed.</p> <p>3. Manually recording PLO achievement from the CO received.</p>	There is no such specific system which can automatically calculate CO percentages and determine whether the COs and PLOs have been achieved or not.	A system can be introduced which can calculate CO percentages automatically with the marks provided as an input .The system will show by what percentages the PLOs and COs have been achieved and also all the COs and PLOs that the student failed to achieve.
Generate Report	Department	Manually retrieving	The OBE marksheet	This issue can be resolved by

		<p>information from MS Excel.</p> <p>submitted by the instructor to the department for generating report is stored in MS Excel so manually retrieving every information from it might be very inefficient and can cause manual errors.</p>	<p>introducing a system in which the faculty provides the OBE marksheet and through that system the department can automatically generate a report.</p>
--	--	--	---

SECTION 2.5: RICH PICTURE (PROPOSED)

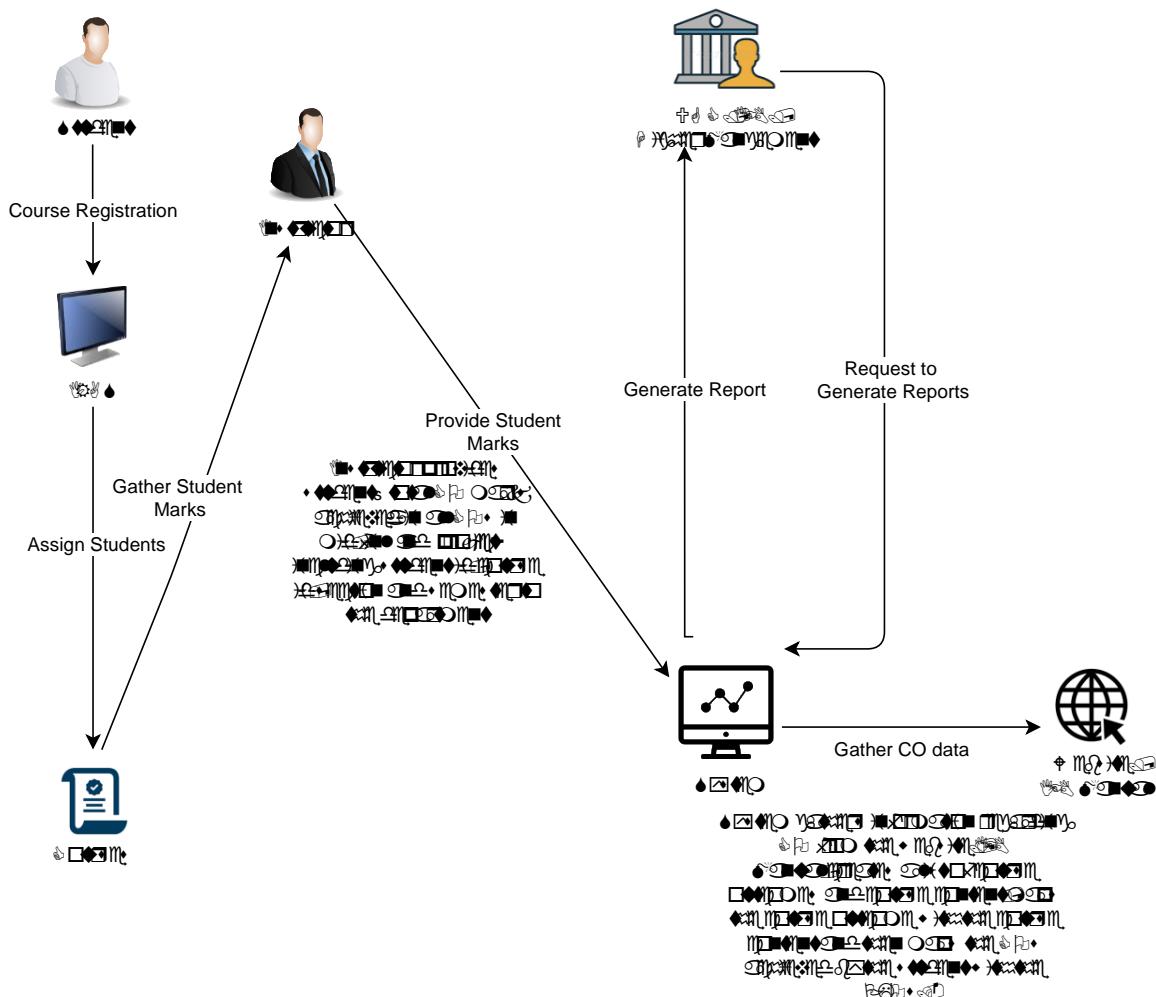


Figure 3: Proposed System Rich Picture

The system gathers the CO data from the website / IEB manual. Then the system creates a list of the course content and a list of the course outcomes and hence mapping the CO's with the course content. The CO's received are then mapped to specific PLO. The instructor receives CO's to be achieved in a particular course and takes exams for assessing the students in that course. The instructor then converts the total marks and calculates total for all CO's. The CO marks are provided to the system and the system stores the marks. Based on the stored information, the system then creates reports for analysis by the UGC/ IEB/ Higher management.

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 sends accreditation manual with PLO's defined to the department 2. Department then gathers information regarding Course Outcome from the website/ IEB manual. 3. If the course content exist in the course outline, review the course content else 	None.	<p>Computer:</p> <ol style="list-style-type: none"> 1. All related data is searched and stored using computer. 	None.	<p>Other Sources:</p> <ol style="list-style-type: none"> 1. All related information are stored in the specific location. 	All related data are searched through the internet.

	<p>create a list of the course content.</p> <p>4. Creates a list of the course outcomes.</p> <p>5. Login to the system with user ID and password.</p> <p>Map the CO with the course content.</p> <p>6. Map the received CO's to specific PLOs.</p> <p>7. Map CO's to specific questions of mid-term, final & project.</p>				
Course Registration	<p>Student:</p> <p>1. On the reserved day for registration log in to IRAS with their id and password.</p> <p>2. Go to registration page, load courses and select courses.</p> <p>Can only select courses of which</p>	<p>Pen & Paper:</p> <p>1. Sometimes course registration is done manually.</p>	<p>Computer:</p> <p>1. All related data is searched and stored using computer.</p>	<p>IRAS:</p> <p>1. IRAS has an interface for doing registration.</p>	<p>MS SQL Server:</p> <p>1. IRAS is integrated with MS SQL Server.</p> <p>Internet:</p> <p>IRAS is a web-based application and requires internet to access.</p> <p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>

	<p>prerequisites have been completed and the course capacity is not exceeded.</p> <p>3. Click on 'DONE' button to complete the registration process.</p> <p>4. A message prompts saying "Registration Successfully Done".</p>					
Gather Student Marks from Courses they have enrolled	<p>Instructor:</p> <p>1. Receives CO's to be achieved in that particular course.</p> <p>2. Takes exams such as mid-term, final & project to assess students in various course outcomes.</p> <p>3. Gather marks for different assessment including student id, course id, section, semester.</p>	<p>Pen & Paper:</p> <p>1. Written exams conducted by the faculty to evaluate the students.</p>	<p>Computer:</p> <p>1. All related data such as student name, id is searched and stored using computer.</p>	<p>MS Excel:</p> <p>1. All related information are stored.</p>	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	None.

	<p>4. Convert the total marks for mid-term, final.</p> <p>5. Calculate total for all the CO's.</p>					
Enter Student Marks	<p>Instructor:</p> <p>1. Login to the system with ID and password.</p> <p>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.</p> <p>2. System stores students marks achieved in all CO's based on mid-term, final & project with student id, course id, section, semester.</p> <p>3. Calculates CO percentages.</p> <p>4. If greater than or equal to 40% CO's</p>	<p>None.</p>	<p>Computer:</p> <p>1. All related data is searched and stored using computer.</p>	<p>MS Excel:</p> <p>1. All related information are stored.</p>	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>	<p>All related data are provided through the internet.</p>

	<p>have been achieved, a student passes that certain CO otherwise fails.</p> <p>5. The corresponding CO's are mapped against PLO's and PLO achievement s are recorded.</p>				
Generate Report	<p>Department:</p> <p>1. User will login to the system and retrieve the student information of PLO achievement from previously calculated OBE marksheets.</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>None.</p>	<p>Computer:</p> <p>1. All related data is searched and stored using computer.</p>	<p>MS Excel:</p> <p>1. All related information are stored and calculated.</p>	<p>Other Sources:</p> <p>1. All related information are stored in the specific location.</p>

Request to Generate Report	UGC/ IEB/ Higher Management : Login to the system with user ID and password. Asking to generate report for: 1a. Viewing number of students passing or failing in a certain PLO. 1b. Viewing progression of students with charts and graphs.	Paper: 1. Paper is required for the printing purpose.	Printer: 1. Print the report based on users' request.	None.	Other Sources: 1. All related information are stored in the specific location.	None.
----------------------------	--	--	--	-------	---	-------

SECTION 2.7: PROCESS DIAGRAM (PROPOSED)

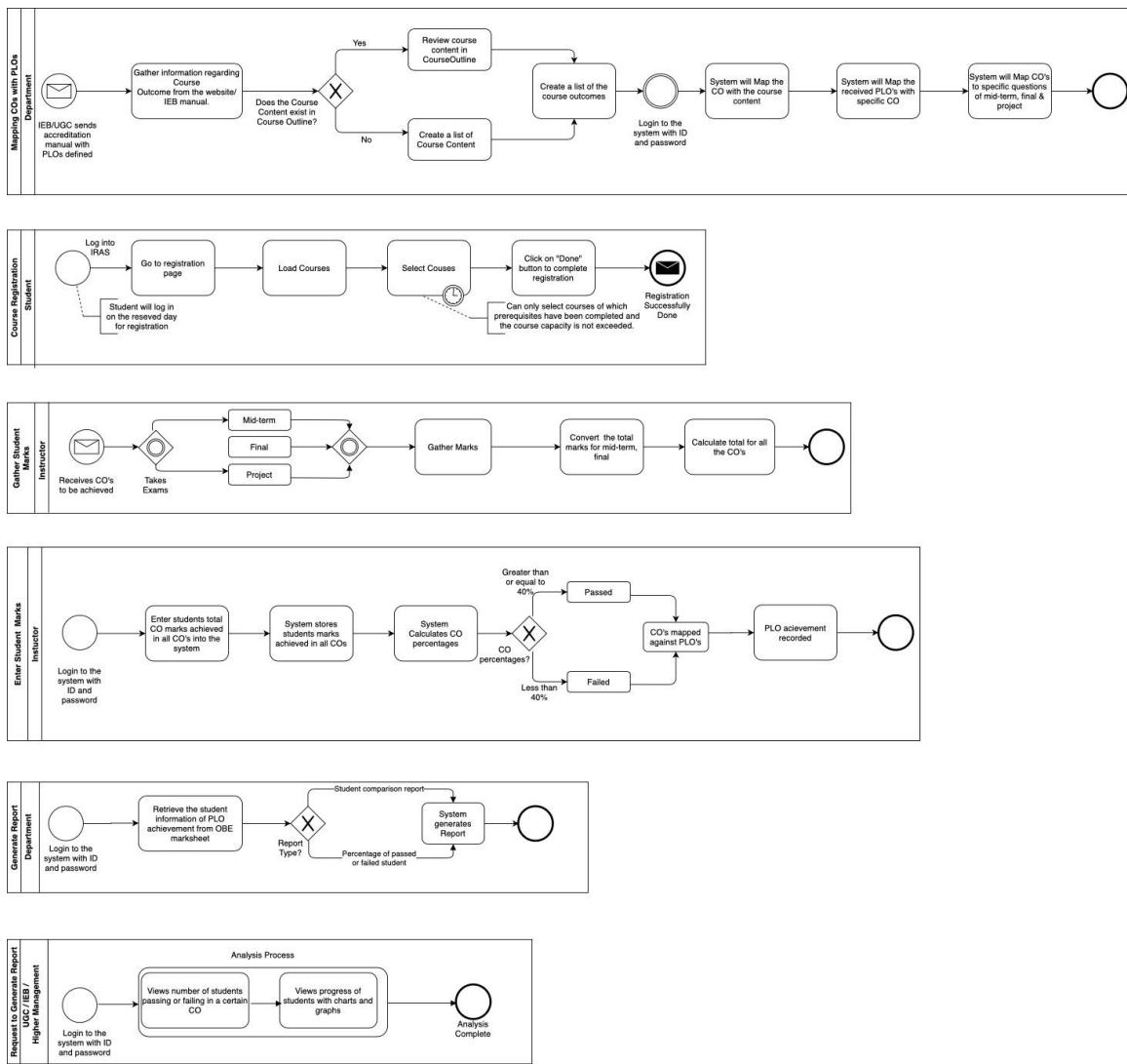


Figure 4: Proposed System BPMN

CHAPTER 3: LOGICAL SYSTEM DESIGN

SECTION 3.1: BUSINESS RULES

1. IEB sends accreditation manual with PLO's defined to the departments. Each department then gathers information from the IEB manual/ website.
2. Each of the Degree Programs under which a student gets admitted belongs to a department. A Program has exactly one department. Each program consists of many courses and a course belongs to exactly one program.
3. Under the OBE model for each program there will be a set of program learning outcomes (PLO). A PLO has a PLOId, PLO name and description.
4. A Department has many students but a student must have exactly one department. Each of the departments has a name and a phone. Each department has a particular faculty as a head. Department offers many courses but a course is offered by exactly one department.
5. To evaluate the students in each course, courses have a set of course outcomes(COs) that are mapped with the PLOs of the degree program. A CO must be mapped with exactly one PLO. A PLO may be mapped with one or more COs. The COs are measured through different assessment techniques(e.g., quiz, mid, final, project, presentation). A CO has a COId, CO name and description.
6. An assessment is mapped with exactly one CO and a CO is mapped with one or more assessments. Each of the assessments are identified uniquely using an AssessmentID. Assessment has an assessment name (eg.Mid Q1, Mid Q2, Final Q1, Final Q3 etc), CourseID, COID, Section Number and the total marks that is achievable in that particular assessment. An assessment contains exactly one section. A section must have one or more assessments.
7. Faculties evaluate the COs achieved and mapped PLOs achieved by each student in a course. An Evaluation is done by one or more faculties and a faculty must do atleast one evaluation. There is a date of evaluations done by a faculty. Each of the evaluations are identified uniquely using a StudentID and an AssessmentID. Evaluation also has the total marks obtained by a student in an assessment. An assessment has one or more evaluation but an evaluation is done for exactly one assessment. A student belongs to one or more evaluation but an evaluation belongs to exactly one student.
8. Accounts are maintained for two kinds of users of the system: students and faculties. An account has an ID, name (first name, last name), birth date, gender, email address, contact number, address. A student must log in to Iras for registration of course.
9. Faculties have academic qualifications (i.e. highest degree certificate so far), area/s of specialization, job position (e.g. Lecturer, Professor), salary. A student must have an enroll date and the total course completed. A faculty has exactly one department and a department has multiple faculties.

10. A Course have a CourseID that uniquely identifies the course. Course also has a CourseTitle. A course can be a prerequisite of one or more courses and a course may have one or more prerequisites. A course may be mapped with many other courses and multiple courses may be mapped with exactly one course.
11. Each course must be taught by atleast one faculty. A Faculty may teach multiple courses. Every faculty has a teaching schedule i.e. teaching days and teaching time for a course. There may be multiple sections for each course in a particular semester but a particular section teaches exactly one course.
12. A section has a section number; however, different courses may have sections with the same number assigned to them. Hence, a semester and CourseID are required along with the section number to identify a section. A section also has a schedule and a maximum capacity. A student may enroll in one or more sections and a section must have atleast one student. There is a registration date for the courses taken by a student in a semester. If a course has no student enrolled, then all of it's sections along with the course are removed.

SECTION 3.2: ERD

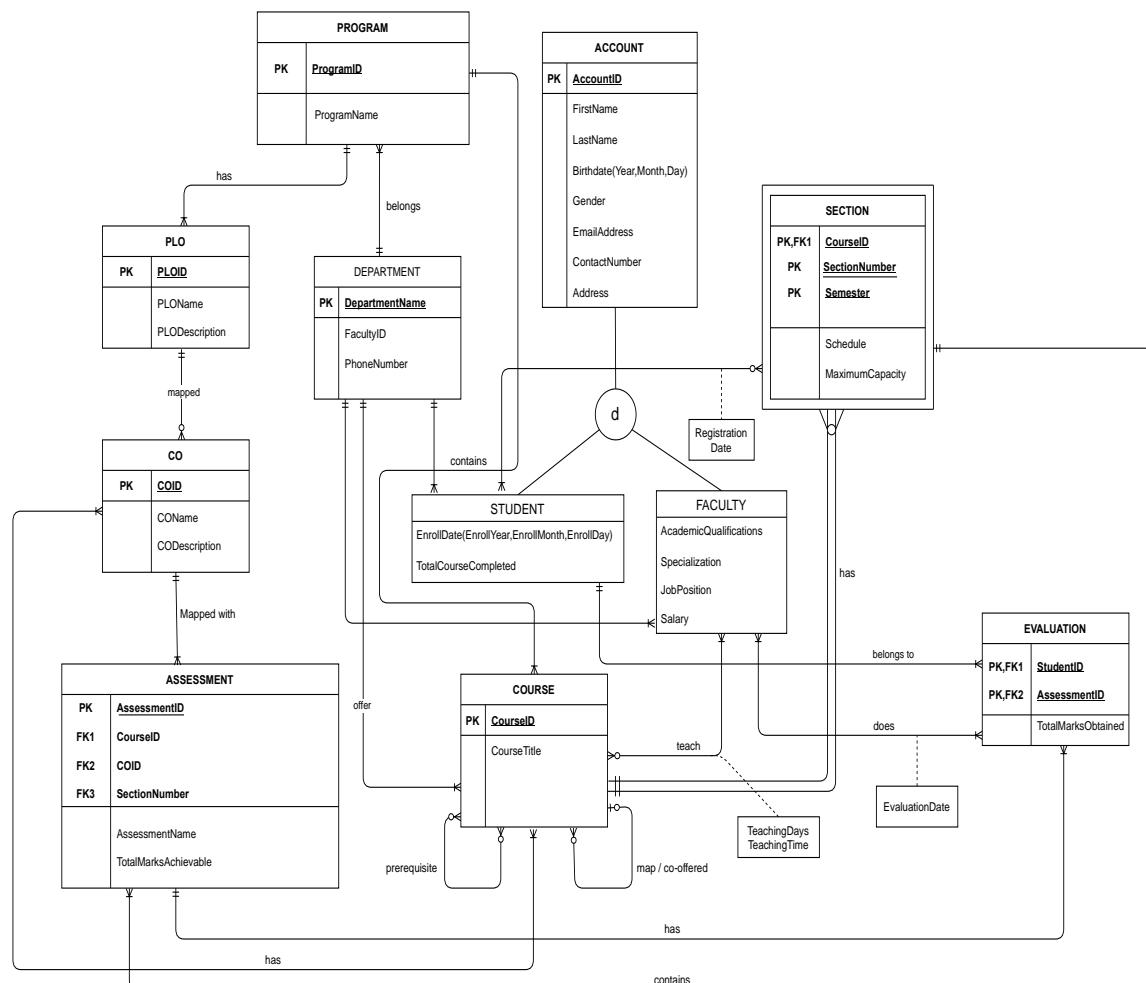


Figure 5: Entity Relationship Diagram

SECTION 3.3: ERD TO RELATIONS

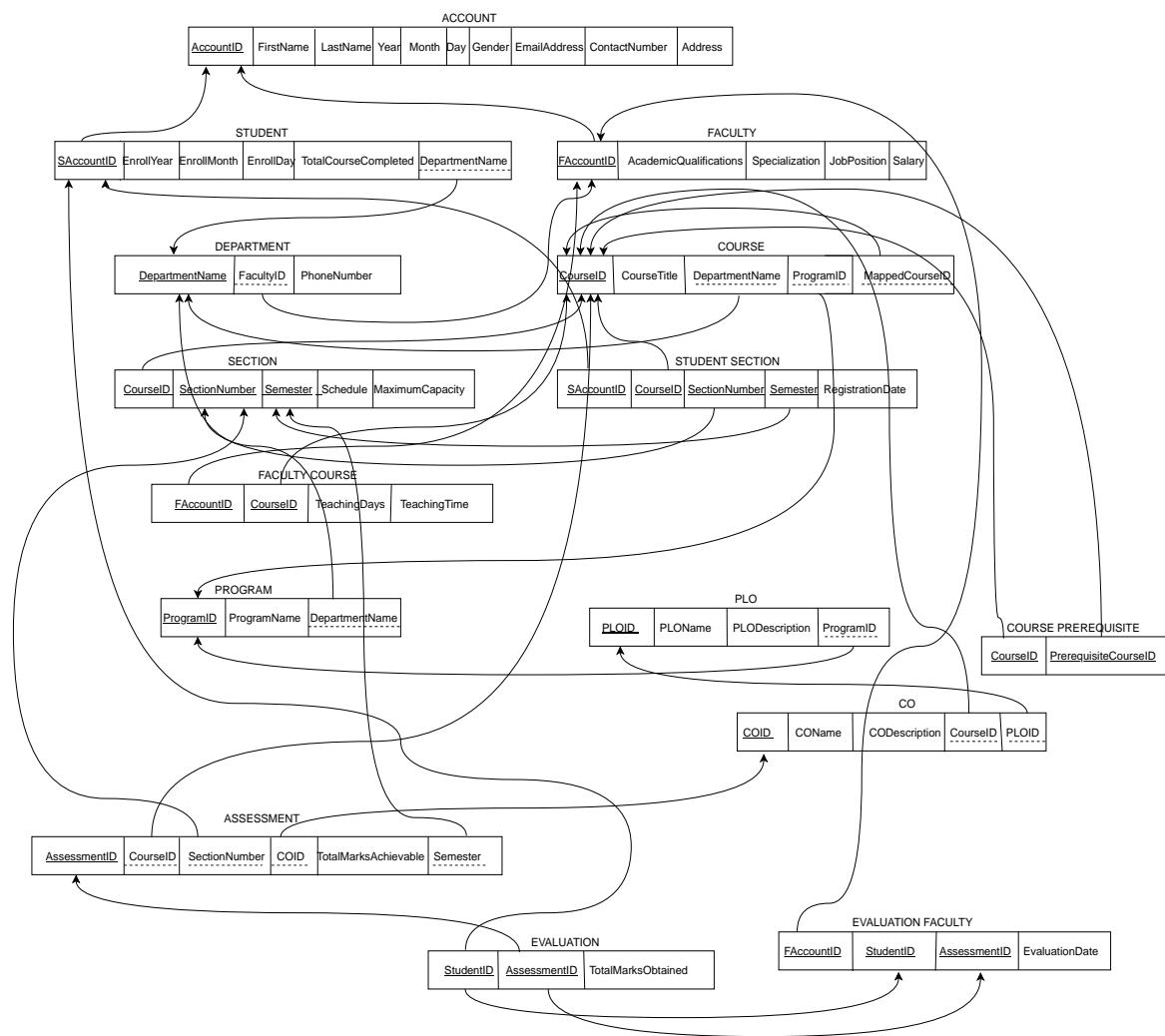


Figure 6: Relational Schema Diagram

SECTION 3.4: NORMALIZATION

Functional Dependencies

AccountID → FirstName, LastName, Year, Month, Day, Gender, EmailAddress, ContactNumber, Address, EnrollYear, EnrollMonth, EnrollDay, TotalCourseCompleted,
CourseID → CourseTitle, DepartmentName, ProgramID, MappedCourseID, PrerequisiteCourseID
CourseID, SectionNumber, Semester → Schedule, MaximumCapacity
SAccountID, CourseID, SectionNumber, Semester → RegistrationDate
DepartmentName → FacultyID, PhoneNumber
FacultyID, CourseID → TeachingDays, TeachingTime
ProgramID → ProgramName, DepartmentName
PLOID → PLOName, PLODescription, ProgramID
COID → COName, CODescription, CourseID, PLOID
AssessmentID → CourseID, SectionNumber, COID, TotalMarksAchievable, Semester
StudentID, AssessmentID → TotalMarksObtained
AccountID, StudentID, AssessmentID → EvaluationDate

1 NF

RELATION

AccountID	FirstName	LastName	Year	Month	Day	Gender	EmailAddress	ContactNumber	Address	EnrollYear	EnrollMonth	EnrollDay	TotalCourseCompleted
PrerequisiteCourseID	DepartmentName	AcademicQualifications	Specialization	JobPosition	Salary	PhoneNumber	Courses	CourseTitle	SectionNumber				
ProgramID	MappedCourseID	Semester	Schedule	MaximumCapacity	RegistrationDate	TeachingDays	TeachingTime	ProgramName	PLOID	PLOName			

2 NF

RELATION1

AccountID	FirstName	LastName	Year	Month	Day	Gender	EmailAddress	ContactNumber	Address	EnrollYear	EnrollMonth	EnrollDay	TotalCourseCompleted
EnrollDay	TotalCourseCompleted	DepartmentName	AcademicQualifications	Specialization	JobPosition	Salary	PhoneNumber						

RELATION2

CourseID	CourseTitle	DepartmentName	ProgramID	MappedCourseID	PrerequisiteCourseID	PhoneNumber	ProgramName
----------	-------------	----------------	-----------	----------------	----------------------	-------------	-------------

RELATION3

COID	COName	CODescription	PLOID	PLOName	PLODescription	ProgramID
------	--------	---------------	-------	---------	----------------	-----------

RELATION4

AssessmentID	TotalMarksAchievable
--------------	----------------------

RELATIONS

AccountID	CourseID	SectionNumber	Semester	COID	AssessmentID	Schedule	MaximumCapacity	RegistrationDate	TeachingDays	TeachingTime	TotalMarksObtained	EvaluationDate
-----------	----------	---------------	----------	------	--------------	----------	-----------------	------------------	--------------	--------------	--------------------	----------------

3 NF

RELATION11

AccountID	FirstName	LastName	Year	Month	Day	Gender	EmailAddress	ContactNumber	Address	EnrollYear	EnrollMonth
EnrollDay	TotalCourseCompleted	DepartmentName	AcademicQualifications	Specialization	JobPosition	Salary	PhoneNumber				

RELATION12

DepartmentName	PhoneNumber
----------------	-------------

RELATION21

CourseID	CourseTitle	CourseOutcome	ProgramID	MappedCourseID	PrerequisiteCourseID
----------	-------------	---------------	-----------	----------------	----------------------

RELATION22

ProgramID	ProgramName	DepartmentName
-----------	-------------	----------------

RELATION23

DepartmentName	PhoneNumber
----------------	-------------

RELATION31

COID	COName	CODescription	PLOID
------	--------	---------------	-------

RELATION32

PLOID	PLOName	PLODescription	ProgramID
-------	---------	----------------	-----------

RELATION4

AssessmentID	TotalMarksAchievable
--------------	----------------------

RELATIONS

AccountID	CourseID	SectionNumber	Semester	COID	AssessmentID	Schedule	MaximumCapacity	RegistrationDate	TeachingDays	TeachingTime	TotalMarksObtained	EvaluationDate
-----------	----------	---------------	----------	------	--------------	----------	-----------------	------------------	--------------	--------------	--------------------	----------------

BCNF

All relations are already in BCNF.

Figure 7: Normalization Diagram

SECTION 3.5: DATA DICTIONARY

tblaccount:

Name	Data Type	Size	Remark
naccountid	Number	7	This is the primary key of this relation. This contain the ID of students and instructors. Example: ‘1821952’ for student and ‘4321’ for faculty.
cffirstname	Text		This is the first name of the students or instructors. Example: ‘Navid’.
clastname	Text		This is the last name of the students or faculty. Example: ‘Islam’.
dbirthdate	Datetime	“dd/mm/yy”	This contains date of birth of the students or faculty. Example: ‘26/07/97’.
cgender	Text	6	This contains the gender of the students or instructors. Example: ‘Male’ or ‘Female’.
cemailaddress	Text		This is the email address of the students or faculty.
ncontactnumber	Number	11	This is the contact number of the students or faculty. Example: ‘01712345678’.
caddress	Text		This is the home address of the students or faculty.

tblfaculty:

Name	Data Type	Size	Remark
nfaccountid	Number	4	This is the primary key of this relation. This contain the ID of faculty. Example: ‘4321’.
cacademicqualifications	Text		This is the latest academic qualifications of the faculty.
cspecialization	Text		This is the specialization of the faculty.
cjobposition	Text	9	This is the job position of the faculty. Example: ‘Professor’ or ‘Lecturer’.
nsalary	Number		This is the monthly salary of the faculty.

tbldepartment:

Name	Data Type	Size	Remark

cdepartmentname	Text		This is the primary key of this relation. This contains the department name.
nfacultyid	Number	4	This contains the name of the department head.
nphonenumbers	Number	11	This contains the phone number of the department.

tblsection:

Name	Data Type	Size	Remark
ccourseid	Text	7	This is the composite key of this relation. This contains the course id. Example: 'CSE303'.
nsectionnumber	Number	2	This is the composite key of this relation. This contains the section number of a particular course in a particular semester.
csemester	Text	6	This is the composite key of this relation. This contains the semester name. Example: 'Spring', 'Summer', 'Autumn'.
nschedule	Text		This is the schedule of a section which is the timing of a course.
nmaximumcapacity	Number	2	This is the maximum capacity a section can hold.

tblstudentsection:

Name	Data Type	Size	Remark
nsaccountid	Number	7	This is the composite key of this relation. This contains the account id of the students. Example: '1821952'.
ccourseid	Text	7	This is the composite key of this relation. This contains the course id. Example: 'CSE303'.
nsectionnumber	Number	2	This is the composite key of this relation. This contains the section number of a particular course in a particular semester.
csemester	Text	6	This is the composite key of this relation. This contains the semester name. Example: 'Spring', 'Summer', 'Autumn'.
dregistrationdate	Datetime	"dd/mm/yy"	This contains registration date of the students in a particular course.

tblfacultycourse:

Name	Data Type	Size	Remark
nfaccountid	Number	4	This is the composite key of this relation. This contains the account id of the instructors. Example: '4321'.
ccourseid	Text	7	This is the composite key of this relation. This contains the course id. Example: 'CSE303'.
cteachingdays	Text	9	This contains the days a particular faculty teaches.
nteachingtime	Number		This contains the teaching time of a faculty.

tblcourseoutcome:

Name	Data Type	Size	Remark
ccourseid	Text	7	This is the composite key of this relation. This contains the course id of a particular course.
ccourseoutcome	Text		This is the composite key of this relation. This contains the course outcome from a particular course.

tblstudent:

Name	Data Type	Size	Remark
nsaccountid	Number	7	This is the primary key of this relation. This contains the account id of the students. Example: '1821952'.
denrolldate	Datetime	"dd/mm/yy"	This contains the enroll date of the students.
ntotalcoursecompleted	Number	2	This contains the total course completed by the students.
cdepartmentname	Text		This is a foreign key from Table DEPARTMENT.

tblcourse:

Name	Data Type	Size	Remark
ccourseid	Text	7	This is the primary key of this relation. This contains the course id. Example: 'CSE303'.

ccourseid	Text		This contains the course title of a particular course. Example: ‘Database Management’.
-----------	------	--	--

tblprogram:

Name	Data Type	Size	Remark
nprogramid	Number	1	This is the primary key of this relation. This contains the program id.
ccprogramname	Text		This contains the program name.
cdepartmentname	Text		This is a foreign key from Table DEPARTMENT.

tplo:

Name	Data Type	Size	Remark
cploid	Number	1	This is the primary key of this relation. This contains the plo id.
cploname	Text		This contains the plo name of a particular course. Example: ‘Database Management’.
cplodescription	Text		This is the plo description.
nprogramid	Number	1	This is a foreign key from Table PROGRAM.

tco:

Name	Data Type	Size	Remark
ccoid	Number	1	This is the primary key of this relation. This contains the co id.
cconame	Text		This contains the co name.
ccodescription	Text		This contains the description of co.
ccourseid	Number	7	This is a foreign key from Table COURSE.
cploid	Number	1	This is a foreign key from Table PLO.

tcourseprerequisite:

Name	Data Type	Size	Remark
ccourseid	Text	7	This is the composite key of this relation. This contains the course id. Example: ‘CSE303’.

ccprerequisitecourseid	Text	7	This is the composite key of this relation. This contains the course id of the prerequisite course.
------------------------	------	---	---

tassessment:

Name	Data Type	Size	Remark
nassessmentid	Number		This is the primary key of this relation. This contains the assessment id.
ccourseid	Text	7	This is a foreign key from Table COURSE.
nsectionnumber	Number	2	This is a foreign key from Table SECTION.
ncoid	Number	1	This is a foreign key from Table CO.
nassessmenttype	Number	2	This contains the assessment marks.
ntotalmarksachievable	Number	3	This contains the total marks achievable.

tevaluation:

Name	Data Type	Size	Remark
nstudentid	Number	7	This is the composite key of this relation. This contains the student id.
nassessmentid	Number		This is the composite key of this relation. This contains the assessment id.
ntotalmarksobtained	Number	3	This contains the total marks achieved.

tevaluationfaculty:

Name	Data Type	Size	Remark
nfaccountid	Number	4	This is the composite key of this relation. This contains the faculty id.
nstudentid	Number	7	This is the composite key of this relation. This contains the student id.
nassessmentid	Number		This is the composite key of this relation. This contains the assessment id.
devaluationdate	Datetime	“dd/mm/yy”	This contains the evaluation date.

CHAPTER 4: PHYSICAL SYSTEM DESIGN

SECTION 4.1.1: INPUT FORMS – ADD USER

PURPOSE

The purpose of adding user is that these users will be able to use the system. So, without adding any kinds of users this system would not exist as this system depends on storing and retrieving data.

RELATED SQL USED

```

1. <?php
2.     require 'mysql.php';
3.     // getting post requests
4.     $id = $_POST['id'];
5.     $fName = $_POST['fName'];
6.     $lName = $_POST['lName'];
7.     $prog = $_POST['prog'];
8.     $email = $_POST['email'];
9.     $password = $_POST['password'];
10.    $role = " target=\" blank"
11.        title="http://www.php.net/strtolower">">http://www.php.net/strtolower">st
12.        rtolower($_POST['role']);
13.
14.    $store = "INSERT INTO user (id, first_name, last_name, program_id,
15.        email, password, role) VALUES
16.            ('$id', '$fName', '$lName', '$prog', '$email', '$password'
17.            , '$role')";
18.
19.    if($mysql->query($store)){
20.
21.        " target=\" blank"
22.        title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
23.        cation: ../admin/add-user.php");
24.    }else{
25.
26.        " target=\" blank"
27.        title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
28.        cation: ../admin/add-user.php");
29.    }
30.
31.
32.    ?>

```

SECTION 4.1.2: INPUT FORMS – ADD PROGRAM AND PLO

PURPOSE

Since this system tracks PLO achievement of students of any courses in any program, so we must input the PLO data with their corresponding program. The system cannot collect

the data automatically, so manually inputting the programs with their PLO's are done using input form.

RELATED SQL USED

```

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',
8.     '$program_name', '$school')";
9.     //echo $sql;
10.    if($mysql->query($sql) == FALSE) {
11.        "target=" blank
12.        title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
13.        cation: ../admin/add-program.php?failed=1");
14.    }
15.    while(" target=" blank"
16.        title="http://www.php.net/isset">">http://www.php.net/isset">isset($_POST
17.        ['title'].$i)) {
18.            $name = $_POST['title'].$i];
19.            $sql = "INSERT INTO plo(program_id, plo_no, plo_name) VALUES
20.                ('$id', $i, '$name')";
21.            if($mysql->query($sql) == FALSE) {
22.                "target=" blank
23.                title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
24.                cation: ../admin/add-program.php?failed=1");
25.            }
26.            $i++;
27.        }
28.        "target=" blank
29.        title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
30.        cation: ../admin/add-program.php?success=1");
31.
32.    ?>
```

SECTION 4.1.3: INPUT FORMS – ADD COURSE WITH CO

PURPOSE

Each individual course must be added to the system with their respective CO's. In order to do that, an input form must be used to give inputs to the system.

RELATED SQL USED

```

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, $total_co)";
12.
13.    if($mysql->query($sql) == FALSE) {
14.        "target=" blank"
15.        title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
cation: ../admin/add-course.php?failed=1");
16.
17.    for($i=1; $i<=15; $i++) {
18.        if(" target=" blank"
19.            title="http://www.php.net/isset">">http://www.php.net/isset">isset($_POST
["plo-co".$i]) {
20.                $sql = "SELECT sl FROM plo WHERE program_id = '$program_id'
      AND plo_no = $i";
21.                $plo_id = $mysql->query($sql)->fetch_assoc()['sl'];
22.                $data = $_POST["plo-co".$i];
23.                $field = ""; $val = "";
24.                foreach($data as $co) {
25.                    $field .= 'co'. $co . ', ';
26.                    $val .= '1, ';
27.                }
28.                $sql = "INSERT INTO co (course_id, plo_id, ". "
      target=" blank"
29.                title="http://www.php.net/substr">">http://www.php.net/substr">substr($fi
eld, 0, -2).") VALUES ('$id', $plo_id, ".$_ "target=" blank"
30.                title="http://www.php.net/substr">">http://www.php.net/substr">substr($va
l, 0, -2).")";
31.                if($mysql->query($sql) == FALSE) {
32.                    "target=" blank"
33.                    title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
cation: ../admin/add-course.php?failed=1");
34.                }
35.            }
36.        }
37.    }
38.    "target=" blank"
39.    title="http://www.php.net/header">">http://www.php.net/header">header ("Lo
cation: ../admin/add-course.php?success=1");
40.
41. ?>

```

SECTION 4.1.4: INPUT FORMS – ADD MARKS INDIVIDUAL

PURPOSE

This input form requires marks entry process with which a students' data can be entered based on a course in a particular semester. This is then stored in the database which can be retrieved when required. Marks entry form can input all the assessments of a particular student.

RELATED SQL USED

```

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.     $i=1;
13.
14.     while(" target="" blank"
15.           title="http://www.php.net/isset">>http://www.php.net/isset">isset($_POST
16.           ['co'.'.$i])){
17.         $field .= 'q'.'$i.'_mark, ' . 'q'.'$i.'_co, ' . 'q'.'$i.'_max, ';
18.         $val .= $_POST['mark'.'$i'].'.', '.$_POST['co'.'$i'].'.,
19.           '.$_POST['max'.'$i'].'.', ',';
20.         $i++;
21.     }
22.
23.     $sql = "INSERT INTO marks(student_id, course_id, exam_name, semester,
24.     section, ". target="" blank"
25.       title="http://www.php.net/substr">>http://www.php.net/substr">substr($fi
26.       eld, 0, -2).") VALUES
27.           ('$student_id', '$course_id', '$exam_name', '$semester',
28.             '$section', ". target="" blank"
29.               title="http://www.php.net/substr">>http://www.php.net/substr">substr($va
1.   l, 0, -2).");
22.
23.   echo $sql . '<br>';
24.   $mysql->query($sql);
25.
26.
27.   ". target="" blank"
28.   title="http://www.php.net/header">>http://www.php.net/header">header("Lo
29.   cation: ../faculty/entry-marks.php?success=1");
28.
29. ?>

```

SECTION 4.1.5: INPUT FORMS – ADD MARKS MASS

PURPOSE

This input form requires marks entry process with which many students data can be entered on any course in any semester. This is then stored in the database which can be retrieved when required. Marks entry form can input all the assessments of all the students using a .csv format file.

RELATED SQL USED

```

1. <?php
2. require 'mysql.php';
3. $course_id = $_POST['course_id'];
4. $exam_name = $_POST['exam_name'];
5. $semester = $_POST['semester'];
6. $section = $_POST['section'];
7.
8. $file = ". target="" blank"
9. title="http://www.php.net/fopen">>http://www.php.net/fopen">fopen($_FILE
S['file']['tmp_name'], "r");

```

```

10.      "_target=\" blank\""
        title="http://www.php.net/fgetcsv">">http://www.php.net/fgetcsv">fgetcsv(
        $file,200); // dumping header
11.      $co = "_target=\" blank\""
        title="http://www.php.net/array">">http://www.php.net/array">array();
12.      $f = 1; $i=1;
13.      foreach("_target=\" blank\""
        title="http://www.php.net/fgetcsv">">http://www.php.net/fgetcsv">fgetcsv(
        $file, 200) as $c){
14.          if($f == 1){
15.              $f = 0;
16.              continue;
17.          }
18.          $co[$i] = $c;
19.          $i++;
20.      }
21.      $max = "_target=\" blank\""
        title="http://www.php.net/array">">http://www.php.net/array">array();
22.      $f = 1; $i=1;
23.      foreach("_target=\" blank\""
        title="http://www.php.net/fgetcsv">">http://www.php.net/fgetcsv">fgetcsv(
        $file, 200) as $m){
24.          if($f == 1){
25.              $f = 0;
26.              continue;
27.          }
28.          $max[$i] = $m;
29.          $i++;
30.      }
31.
32.      while($marks = "_target=\" blank\""
        title="http://www.php.net/fgetcsv">">http://www.php.net/fgetcsv">fgetcsv(
        $file, 200)){
33.
34.          $student_id;
35.          $field = "";
36.          $val = "";
37.          $f = 1; $i=1;
38.
39.          foreach($marks as $m){
40.              if($f==1){
41.                  $student_id = $m;
42.                  $f=0;
43.                  continue;
44.              }
45.              $field .= 'q' . $i . '_mark, ' . 'q' . $i . '_co, ' . 'q' . $i . '_max,
        ';
46.              $val .= $m . ', ' . $co[$i] . ', ' . $max[$i] . ', ';
47.              $i++;
48.          }
49.
50.          $sql = "INSERT INTO marks(student_id, course_id, exam_name,
        semester, section, "_target=\" blank\""
        title="http://www.php.net/substr">">http://www.php.net/substr">substr($fi
        eld, 0, -2).") VALUES
51.          ('$student_id', '$course_id', '$exam_name', '$semester',
        '$section', "_target=\" blank\""
        title="http://www.php.net/substr">">http://www.php.net/substr">substr($va
        l, 0, -2).')";
52.
53.          $mysql->query($sql);
54.      }
55.
56.      "_target=\" blank\""
        title="http://www.php.net/header">">http://www.php.net/header">header("Lo
        cation: ../faculty/entry-marks-mass.php?success=1");
57. ?>

```

SECTION 4.2.1: OUTPUT QUERY AND REPORTS – HIGHER MANAGEMENT DASHBOARD

PURPOSE AND USE

It gives the overall summary of the system which includes the total number of courses that uses OBE model, total students who are evaluated with OBE model, number of faculty evaluated using OBD, total number of PLO set in the program. It can be used to gather information about an institution by the higher management.

DESCRIPTION ALONG WITH SQL

```

1. <?php
2.     require 'php/mysql.php';
3.     $sql = "SELECT * FROM course";
4.     $course = $mysql->query($sql)->num_rows;
5.     $sql = "SELECT * FROM user WHERE role='student'";
6.     $student = $mysql->query($sql)->num_rows;
7.     $sql = "SELECT * FROM user WHERE role='faculty'";
8.     $faculty = $mysql->query($sql)->num_rows;
9.     $sql = "SELECT * FROM plo";
10.    $plo = $mysql->query($sql)->num_rows;
11. ?>
```

SECTION 4.2.2: OUTPUT QUERY AND REPORTS – PLO ACHIEVEMENT

PURPOSE AND USE

It is used to show student wise PLO analysis which includes PLO total percentage score for each PLO calculated from the scores achieved in each CO associated with the corresponding PLO among all the courses the student has done so far. Upon entering a student id course wise PLO analysis can be viewed. Also, a tabular view of student wise PLO achievement can be viewed.

DESCRIPTION ALONG WITH SQL

```

1. <?php
2.     require 'mysql.php';
3.     if("_target=" blank"
4.         title="http://www.php.net/isset">">http://www.php.net/isset">isset($_GET[
5.         'id']){
6.             $id = $_GET['id'];
7.             $sql = "SELECT * FROM marks WHERE student_id = $id";
8.             $marks = $mysql->query($sql);
9.             //course based total co marks
10.            $cMarks = "_target=" blank"
11.            title="http://www.php.net/array">">http://www.php.net/array">array();
12.            $cTotal = "_target=" blank"
13.            title="http://www.php.net/array">">http://www.php.net/array">array();
14.            foreach($marks as $marks) {
```

```

12.         $course = $marks['course_id'];
13.         for($i=1; $i<=10; $i++) {
14.             if("_target=" blank"
15.                 title="http://www.php.net/isset">">http://www.php.net/isset">isset($marks
16. ["q".$i."_co"]) &&$marks["q".$i."_co"]!=0) {
17.                 $co = $marks["q".$i."_co"];
18.                 if("_target=" blank"
19.                     title="http://www.php.net/isset">">http://www.php.net/isset">isset($cMark
20. s[$course][$co])) {
21.                         $cMarks[$course][$co] += $marks["q".$i."_mark"];
22.                         $cTotal[$course][$co] += $marks["q".$i."_max"];
23.                     }
24.                 }
25.             }
26.
27.             //course base total plo marks
28.             $pMarks = "_target=" blank"
29.             title="http://www.php.net/array">">http://www.php.net/array">array();
30.             $pTotal = "_target=" blank"
31.             title="http://www.php.net/array">">http://www.php.net/array">array();
32.             foreach($cMarks as $c => $v) {
33.
34.                 $sql = "SELECT * FROM co WHERE course_id = '$c'";
35.                 $plos = $mysql->query($sql);
36.                 foreach($plos as $plo){
37.                     $pId = $plo['plo_id'];
38.                     for($i=1; $i<=10; $i++) {
39.                         if("_target=" blank"
40.                             title="http://www.php.net/isset">">http://www.php.net/isset">isset($plo["co".$i]) && $plo["co".$i]==1)
41.                             if("_target=" blank"
42.                                 title="http://www.php.net/isset">">http://www.php.net/isset">isset($pMakr
43. s[$c][$pId])) {
44.                                     $pMarks[$c][$pId] += $cMarks[$c][$i];
45.                                     $pTotal[$c][$pId] += $cTotal[$c][$i];
46.                                 }
47.                             }
48.                         }
49.
50.             //total marks in plo
51.             $pfMarks = "_target=" blank"
52.             title="http://www.php.net/array">">http://www.php.net/array">array();
53.             $pfTotal = "_target=" blank"
54.             title="http://www.php.net/array">">http://www.php.net/array">array();
55.             foreach($pMarks as $c => $v) {
56.                 foreach($v as $i => $j) {
57.                     if("_target=" blank"
58.                         title="http://www.php.net/isset">">http://www.php.net/isset">isset($pfMar
59. ks[$i])) {
60.                             $pfMarks[$i] += $j;
61.                             $pfTotal[$i] += $pTotal[$c][$i];
62.                         }
63.                     }
64.                 }
65.             //student info

```

```

66.      $sql = "SELECT * FROM user WHERE id = $id";
67.      $student = $mysql->query($sql)->fetch_assoc();
68.      //total plo
69.      $sql = "SELECT * FROM plo WHERE program_id =
    '$student['program_id']."' ";
70.      $ploNum = $mysql->query($sql)->num_rows;
71.
72.      $color = ["", "#1FE7C4", "#E45C17", "#06B97B", "#8CE026", "#E1CCF
    F", "#5BA2CC", "#0A2E82", "#957107", "#80CF18"];
73.  }
74. ?>

```

SECTION 4.2.3: OUTPUT QUERY AND REPORTS – PROGRESS VIEW

PURPOSE AND USE

It contains student and course progress views. For a given student, it shows the count of PLO's expected to be achieved and the counts actually achieved at the end of each semester. Upon selecting a certain course, it shows the number of students in that particular course with the percentages of CO achieved or failed.

DESCRIPTION ALONG WITH SQL

```

1. <?php
2.     require 'mysql.php';
3.     if(" target=_blank"
    title="http://www.php.net/isset">>http://www.php.net/isset">isset($_GET[
    'id'])) {
4.         $id = $_GET['id'];
5.         $sql = "SELECT DISTINCT semester FROM marks WHERE student_id
    = $id";
6.         $sems = $mysql->query($sql);
7.
8.         $ploProg = " target=_blank"
    title="http://www.php.net/array">>http://www.php.net/array">array();
9.         foreach($sems as $sem) {
10.             $res = seeker($sem['semester'], $id);
11.             $ploProg[$sem['semester']]['total'] = $res['p'];
12.             $ploProg[$sem['semester']]['com'] = $res['t'];
13.         }
14.
15.     }if(" target=_blank"
    title="http://www.php.net/isset">>http://www.php.net/isset">isset($_GET[
    'c'])) {
16.         $crs = $_GET['c'];
17.         $sql = "SELECT DISTINCT student_id FROM marks WHERE course_id =
    '$crs'";
18.         $uList = $mysql->query($sql);
19.         $totalsS = $uList->num_rows;
20.         $report = " target=_blank"
    title="http://www.php.net/array">>http://www.php.net/array">array();
21.         foreach($uList as $u) {
22.             $usr = $u['student_id'];
23.             $ret = seeker2($crs, $usr);
24.             foreach($ret["co"] as $i => $j) {
25.                 if($j==1) {
26.                     if(" target=_blank"
    title="http://www.php.net/isset">>http://www.php.net/isset">isset($rep
    ort["co"][$i])){
27.                         $report["co"][$i]++;
}

```

```

28.          } else{
29.              $report["co"][$si] = 1;
30.          }
31.      }
32.  }
33. foreach($ret["plo"] as $i => $j) {
34.     if($j==1){
35.         if(" target=_blank"
36.             title="http://www.php.net/isset">">http://www.php.net/isset">isset($repor
37. t["plo"][$i])){
38.             $report["plo"][$i]++;
39.         }
40.     }
41.   }
42. }
43. " target=_blank"
44. title="http://www.php.net/ksort">">http://www.php.net/ksort">ksort($repor
45. t["co"]);
46. " target=_blank"
47. title="http://www.php.net/ksort">">http://www.php.net/ksort">ksort($repor
48. t["plo"]);
49.
50. function seeker($sem, $uid) {
51.     require 'mysql.php';
52.     $sql;
53.     if($sem!="null") {
54.         $sql = "SELECT * FROM marks WHERE student_id = $uid AND
55. semester = '$sem'";
56.     } else{
57.         $sql = "SELECT * FROM marks WHERE student_id = $uid";
58.     }
59.     $sMarks = $mysql->query($sql);
60.
61.     $cMarks = " target=_blank"
62.     title="http://www.php.net/array">">http://www.php.net/array">array();
63.     $cTotal = " target=_blank"
64.     title="http://www.php.net/array">">http://www.php.net/array">array();
65.     foreach($sMarks as $marks){
66.         $course = $marks['course_id'];
67.         for($i=1; $i<=10; $i++){
68.             if(" target=_blank"
69.                 title="http://www.php.net/isset">">http://www.php.net/isset">isset($marks
70. ["q"][$i."_co"]) &&$marks["q"][$i."_co"]!=0){
71.                 $co = $marks["q"][$i."_co"];
72.                 if(" target=_blank"
73.                     title="http://www.php.net/isset">">http://www.php.net/isset">isset($cMark
74. s[$course][$co])){
75.                         $cMarks[$course][$co] += $marks["q"][$i."_mark"];
76.                         $cTotal[$course][$co] += $marks["q"][$i."_max"];
77.                     } else{
78.                         $cMarks[$course][$co] = $marks["q"][$i."_mark"];
79.                         $cTotal[$course][$co] = $marks["q"][$i."_max"];
80.                     }
81.                 }
82.             }
83.         }
84.     }
85. }
86.
87. $pMarks = " target=_blank"
88. title="http://www.php.net/array">">http://www.php.net/array">array();
89. $pTotal = " target=_blank"
90. title="http://www.php.net/array">">http://www.php.net/array">array();
91. foreach($cMarks as $c => $v){
92.     $sql = "SELECT * FROM co WHERE course_id = '$c'";
93. }
```

```

81.         $plos = $mysql->query($sql);
82.         foreach($plos as $plo){
83.             $pId = $plo['plo_id'];
84.             for($i=1; $i<=10; $i++){
85.                 if(" target=_blank"
86.                     title="http://www.php.net/isset">">http://www.php.net/isset">isset($plo["co".$i]) && $plo["co".$i]==1) {
87.                         if(" target=_blank"
88.                             title="http://www.php.net/isset">">http://www.php.net/isset">isset($pMakr
89.                             s[$c][$pId])) {
90.                                 $pMarks[$c][$pId] += $cMarks[$c][$i];
91.                                 $pTotal[$c][$pId] += $cTotal[$c][$i];
92.                             } else{
93.                                 $pMarks[$c][$pId] = $cMarks[$c][$i];
94.                                 $pTotal[$c][$pId] = $cTotal[$c][$i];
95.                             }
96.                         }
97.                     }
98.             $res = " target=_blank"
99.             title="http://www.php.net/array">">http://www.php.net/array">array();
100.            $res['t'] = 0;
101.
102.            $pTrack = " target=_blank"
103.            title="http://www.php.net/array">">http://www.php.net/array">array();
104.            foreach($pMarks as $c => $v) {
105.                foreach($v as $i => $j) {
106.                    $pTrack[$i]=1;
107.                    if($j * 100 / $pTotal[$c][$i]>=40) {
108.                        $res['t']++;
109.                    }
110.                }
111.            }
112.            $res['p'] = " target=_blank"
113.            title="http://www.php.net/count">">http://www.php.net/count">count($pTrac
114.            k);
115.            return $res;
116.        }
117.        function seeker2 ($crs, $uid) {
118.            require 'mysql.php';
119.            $sql = "SELECT * FROM marks WHERE student_id = $uid AND
course_id = '$crs'";
120.            $sMarks = $mysql->query($sql);
121.
122.            $cMarks = " target=_blank"
123.            title="http://www.php.net/array">">http://www.php.net/array">array();
124.            $cTotal = " target=_blank"
125.            title="http://www.php.net/array">">http://www.php.net/array">array();
126.            foreach($sMarks as $marks) {
127.                $course = $marks['course_id'];
128.                for($i=1; $i<=10; $i++) {
129.                    if(" target=_blank"
130.                        title="http://www.php.net/isset">">http://www.php.net/isset">isset($marks
["q".$i."_co"]) &&$marks["q".$i."_co"]!=0) {
131.                            $co = $marks["q".$i."_co"];
132.                            if(" target=_blank"
133.                                title="http://www.php.net/isset">">http://www.php.net/isset">isset($cMark
s[$course][$co])) {
134.                                    $cMarks[$course][$co] += $marks["q".$i."_m
ark"];
135.                                }
136.                            }
137.                        }
138.                    }
139.                }
140.            }
141.        }
142.    }
143.}

```

```

132.                               $cTotal[$course][$co] += $marks["q".$i."_m
133.                               ax"];
134.                               }else{
135.                               $cMarks[$course][$co] = $marks["q".$i."_ma
136.                               rk"];
137.                               }
138.                           }
139.                       }
140.
141.           $pMarks = " target=_blank"
142.           title="http://www.php.net/array">">http://www.php.net/array">array();
143.           $pTotal = " target=_blank"
144.           title="http://www.php.net/array">">http://www.php.net/array">array();
145.           foreach($cMarks as $c => $v) {
146.               $sql = "SELECT * FROM co WHERE course_id = '$c'";
147.               $plos = $mysql->query($sql);
148.               foreach($plos as $plo){
149.                   $pId = $plo['plo_id'];
150.                   for($i=1; $i<=10; $i++){
151.                       if(" target=_blank"
152.                           title="http://www.php.net/isset">">http://www.php.net/isset">isset($plo["
153.                           co".$i]) && $plo["co".$i]==1){
154.                               if(" target=_blank"
155.                                   $pMarks[$c][$pId] += $cMarks[$c][$i];
156.                                   $pTotal[$c][$pId] += $cTotal[$c][$i];
157.                               }else{
158.                                   $pMarks[$c][$pId] = $cMarks[$c][$i];
159.                                   $pTotal[$c][$pId] = $cTotal[$c][$i];
160.                               }
161.
162.           $stats = " target=_blank"
163.           title="http://www.php.net/array">">http://www.php.net/array">array();
164.           foreach($cMarks as $c => $v) {
165.               foreach($v as $i => $m) {
166.                   if(($m * 100 / $cTotal[$c][$i]) >=40 ) {
167.                       $stats["co"][$i] = 1;
168.                   }else{
169.                       $stats["co"][$i] = 0;
170.                   }
171.               }
172.
173.               foreach($pMarks as $p => $v) {
174.                   foreach($v as $i => $m) {
175.                       if(($m * 100 / $pTotal[$p][$i]) >=40 ) {
176.                           $stats["plo"][$i] = 1;
177.                       }else{
178.                           $stats["plo"][$i] = 0;
179.                       }
180.                   }
181.               }
182.
183.           return $stats;
184.       }
185.
186.   ?>

```

SECTION 4.2.4: OUTPUT QUERY AND REPORTS – STUDENT RESULT

PURPOSE AND USE

It is used to show the students result of PLO achievement in a pie chart for all the courses that student has completed upon entering the students id.

DESCRIPTION ALONG WITH SQL

```

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

```

44.         for($i=1; $i<=10; $i++) {
45.             if("target=" blank"
46.                 title="http://www.php.net/isset">">http://www.php.net/isset">isset($plo["co".$i]) && $plo["co".$i]==1) {
47.                     if(" target=" blank"
48.                         title="http://www.php.net/isset">">http://www.php.net/isset">isset($pMakr
49.                             s[$c][$pId])) {
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

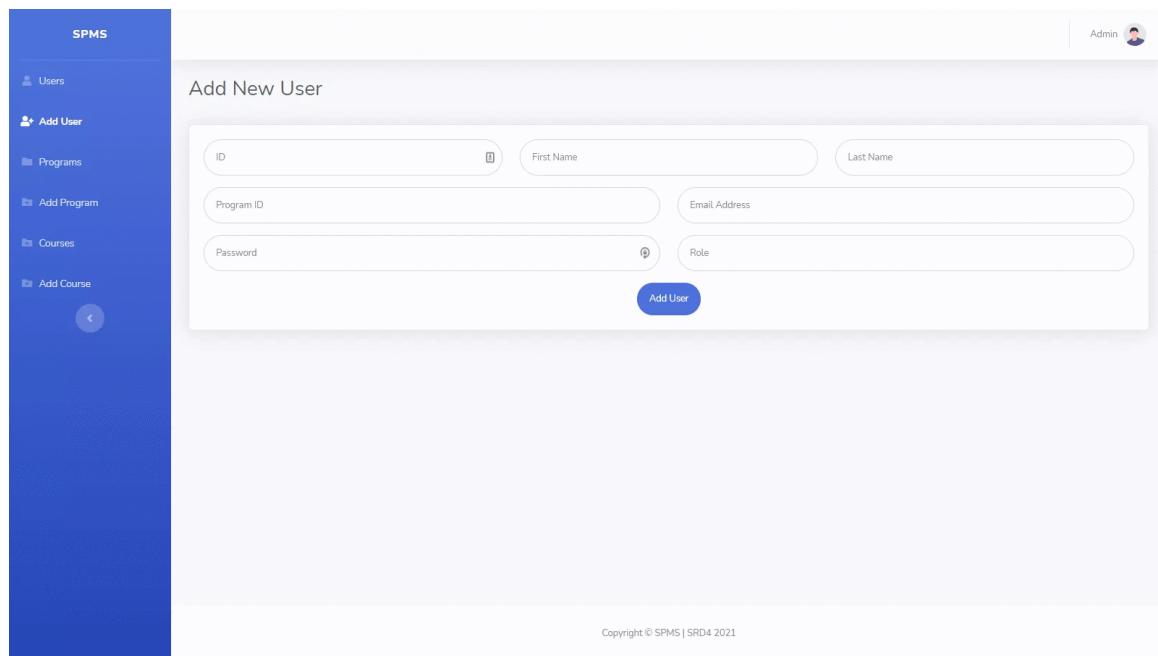


Figure 8: Add User

Add New Program

Program ID	Program name	School
PLO 1	PLO Name	
PLO2	PLO Name	
PLO3	PLO Name	
PLO4	PLO Name	
PLO5	PLO Name	

Add Program

Copyright © SPMS | SRD4 2021

Figure 9: Add Program and PLO

Add New Course

Program ID	Course ID	Course Title
Credit	4	Generate

#	CO1	CO2	CO3	CO4
PLO1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLO15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add Course

Copyright © SPMS | SRD4 2021

Figure 10: Add Course with CO

The screenshot shows the 'Marks Entry - Mass' section of the SPMS interface. On the left sidebar, there are links for 'Marks Entry - Individual' and 'Marks Entry - Mass'. The main area is titled 'Entry Marks - Mass' and contains fields for 'Semester', 'Course ID', 'Section', and 'Exam Name'. Below these is a 'Browse Data File' input field with a 'Choose File' button and a message 'No file chosen'. At the bottom right is a blue 'Submit Mark' button. The footer of the page includes the copyright notice 'Copyright © SPMS | SRD4 2021'.

Figure 11: Add Marks

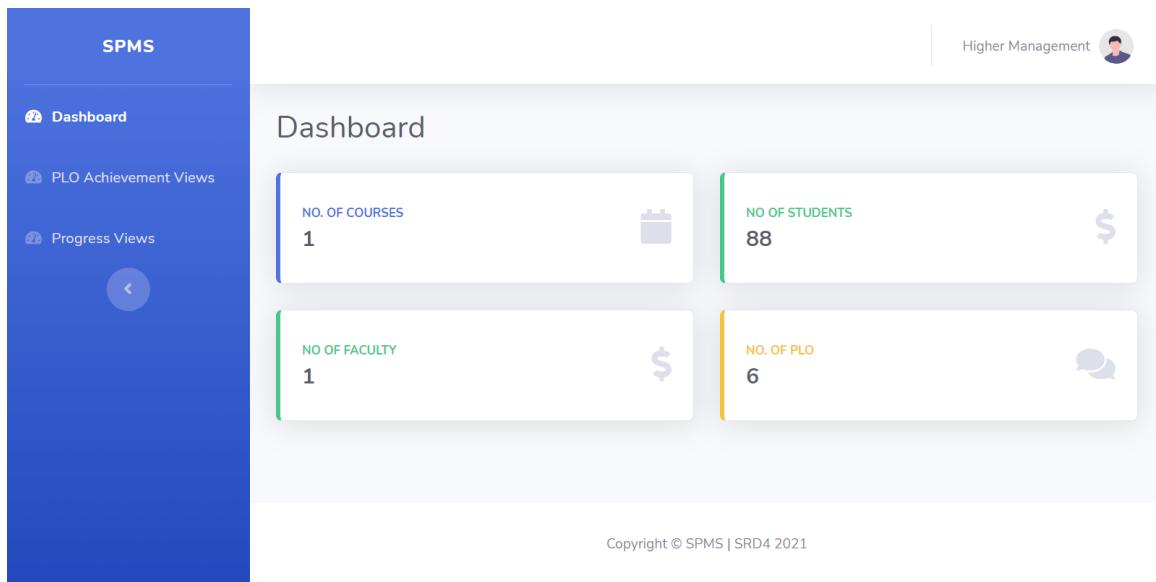


Figure 12: Higher Management Dashboard

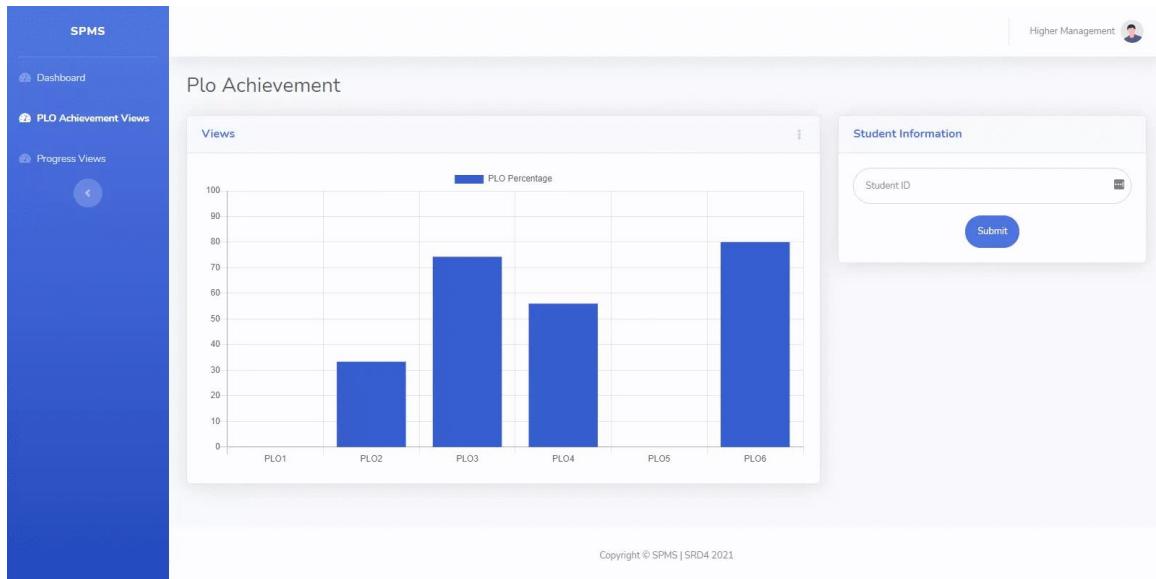


Figure 13: PLO Achievement View 1

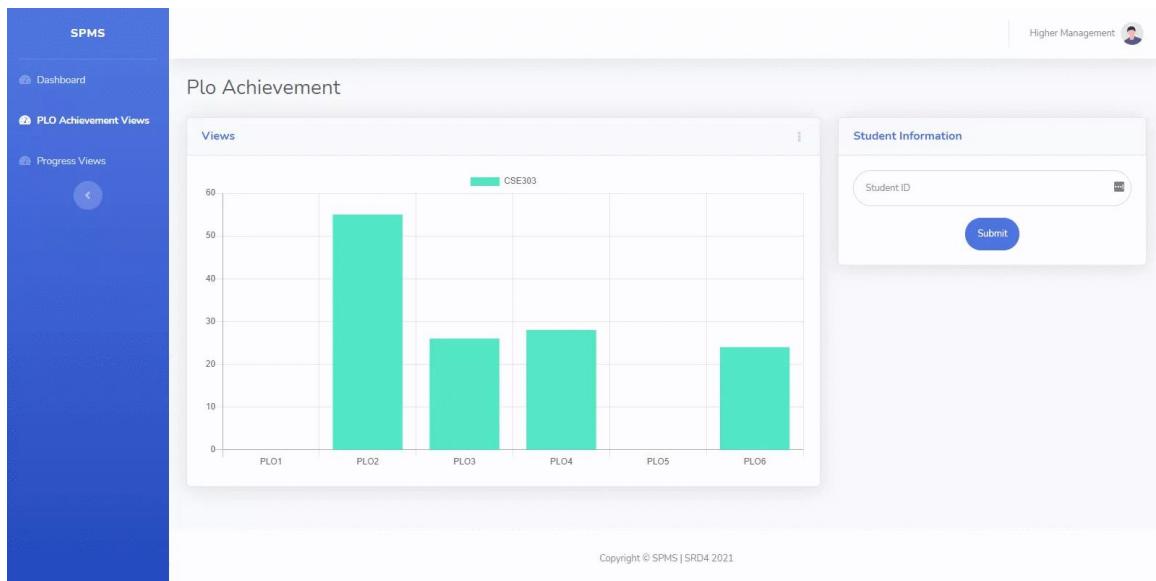


Figure 14: PLO Achievement View 2

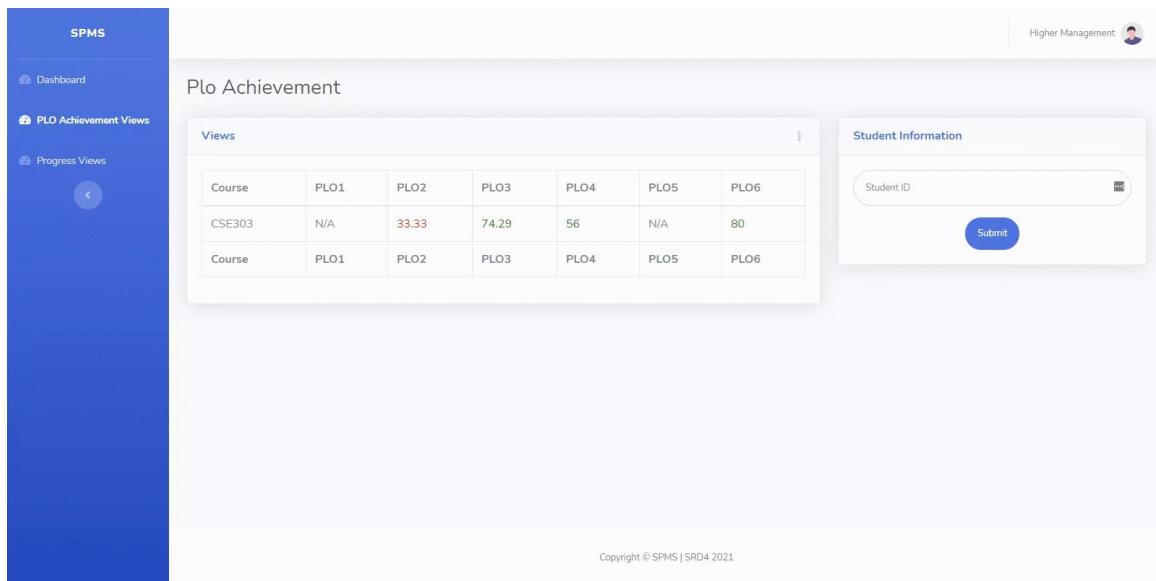


Figure 15: PLO Achievement View 3

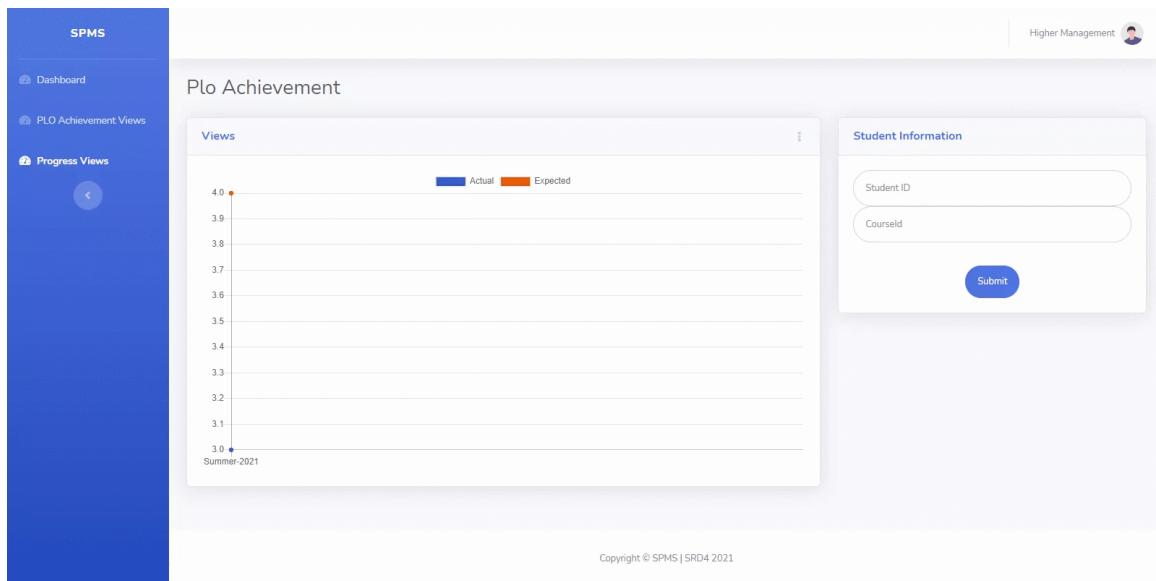


Figure 16: Progress View 1

The screenshot shows the SPMS interface with a blue sidebar on the left containing navigation links: Dashboard, PLO Achievement Views, and Progress Views. The main content area is titled "Plo Achievement" and displays a table titled "Views" with the total student count as 88. The table has columns for CO, Successfully Achieved, Successfully Achieved (%), Failed to Achieve, and Failed to Achieve (%). Data rows include CO1 through CO6 and PLO2 through PLO6. A "Student Information" section on the right contains fields for Student ID and CourseID, with a "Submit" button. The footer copyright notice is "Copyright © SPMS | SRD4 2021".

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

Figure 17: Progress View 2

The screenshot shows the SPMS interface with a blue sidebar on the left containing navigation links: Result. The main content area is titled "Result" and displays a "Student Information" box with details: Name: Student 1, ID: 1416455, Department: CSE, Email: 1416455@iub.edu.bd. To the right is a "PLO Charts" section featuring a pie chart with one segment labeled "CSE303" and "CSE303.55". The footer copyright notice is "Copyright © SPMS | SRD4 2021".

Figure 18: Student Result

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 through email or by doing problem solving sessions, who were experienced enough to guide us in creating the system.

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.

SECTION 5.2: ADDITIONAL FEATURES AND FUTURE DEVELOPMENT

The current proposed system does not include a process with which we can track a failed PLO that were previously achieved in a particular course. Moreover, the system can be made more secured by adding two-factor authentication so that only specific users can access the data stored in the application. An additional AI feature can be included to make it more ease in inputting the data by giving voice input rather than typing.

SECTION 5.3: CONCLUSION & RECOMMENDATION

This Student Performance Monitoring System would provide an insight about how learning might improve in a given program. We have created the system through which a user can automatically store and retrieve data that were previously done manually. It is more user friendly as gathering and collecting data manually was a tiresome task and it required more manpower. Now, these things can be done with ease. Primarily, we have focused on IUB as the organization for which we have done this project but the project has the potential of being useful to other universities as well.

CONTRIBUTION OF EACH MEMBER

	Md Navidul Islam	Tiasha Swarnaker	Sabrina Yasmin	Sumiya Afrin	Md Faysal Chowdhury	Abdullah Al- Noman
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] “IUB at a glance,” [Online].
- [2] “W3Schools,” [Online]. Available: <https://www.w3schools.com/default.asp>.
- [3] J. Duckett, HTML & CSS: Designing and Building Web Sites and Javascript & jQuery: Interactive Front-End Development (1st Edition), Wiley, 2014.
- [4] L. Welling and L. Thomson, PHP and MySQL Web Development (4th Edition), Indianapolis, IN: Sams, 2009.