



Database Management Project

Spring 2023

Final Report of Group-18

Name	ID
Monjurul Hoque Sharnav	1821422
Ikterab Yazda	1910970
Ashir Abdal Ravee	2020646
S.M. Tanzim Tuhin	2022302
Tushar Basak	2022315
Md. Golam Saqlain	2030404
Shahriar Amin Ronok	2031361

CONTENTS

1 INTRODUCTION 2

- 1.1 Background of the Organization 2
 - 1.2 Background of the Project 2
 - 1.3 Objectives of the Project 3
 - 1.4 Scope of the Project 3
-

2 REQUIREMENT ANALYSIS 4

- 2.1 Rich Picture -> Existing (AS-IS) System 4
 - 2.2 Six Element Analysis -> Existing (AS-IS) System 5
 - 2.3 Identification and Analysis of problem 14
 - 2.4 Rich Picture -> Proposed (TO-BE) System 15
 - 2.5 Proposed process and Six Element Analysis 16
-

3 LOGICAL SYSTEM DESIGN 31

- 3.1 Business Rules 31
 - 3.2 Entity Relationship Diagram (ERD) 34
 - 3.3 Relational Schema 35
 - 3.4 Normalization 36
 - 3.5 Data Dictionary 43
-

4 PHYSICAL SYSTEM DESIGN 63

- 4.1 Input Forms 63
 - 4.2 Output Query and Reports 67
-

5 CONCLUSION 72

- 5.1 Problems and Solutions 72
- 5.2 Additional features and future development 72

1

INTRODUCTION

1.1 Background of the Organization

Independent University, Bangladesh (IUB) is a private university located in Dhaka, the capital city of Bangladesh. It was founded in 1993 to offer a top-notch education on par with global standards. The university has many departments, including the schools of business, engineering, computer science, liberal arts and social sciences, environmental science and management, and life sciences. In addition, each department provides undergraduate and graduate degree programs in various subjects' institute for Education and Research, the Center for Sustainable Development, the Institute for Business, Law, and Technology, and the Bangladesh Enterprise Institute are just a few of the research centers and institutions that IUB has to offer in addition to its academic programs.

1.2 Background of the Project

SpmsV4 system is a software application designed to track and manage the academic progress of students. The system is typically used by teachers, administrators, department Heads, and deans to monitor student performance, identify areas of concern, and track progress over time. Depending on the particular demands and objectives, the project's backdrop student performance monitoring system may change. A student performance monitoring system can assist in finding students who are having academic difficulty by keeping track of their grades, attendance, and other indicators. This enables instructors and administrators to step in early and offer the assistance the student needs to develop. A performance monitoring system may automate administrative processes like grade reporting and tracking, which lightens the strain on instructors and administrators and enables them to devote more time to education. A student performance monitoring system's overall objective is to guarantee that each student receives the assistance they require to succeed academically.

1.3 Objectives of the Project

A student performance monitoring system's major goal is to provide a thorough platform for tracking and evaluating students' performance, development, and growth across time. Teachers should be able to enter and record grades for various tasks, tests, and examinations on the system, and students should be able to see their development over time. This system allows instructors and administrators to monitor student attendance to spot trends of absence and take appropriate action. This feature allows instructors and administrators to monitor student attendance to spot trends of absence and take appropriate action.

SpmsV4 keeps relevant documents and data in the database to evaluate the performance of the stakeholders, such as all exam question papers, answer scripts, course outlines, and marks of the examinations and assessments concerning the students' Course Outcomes (CO), Program Learning Outcomes (PLO), and Program Outcomes (PO). Therefore, Students can quantitatively track their progress. Giving them access to a variety of analytical data based on student, course instructor, department, school, and program performance SpmsV4 also gives Higher Authorities chances to draw conclusions and make additional changes.

1.4 Scope of the Project

While analysis of the existing system (SPMS 3.0), identified some issues in the business processes and ERD which can cause a loss of information in the database. SPMS 3.0 has many functions missing in this project. The proposed solution is to overcome those issues and create a more improved version of the existing system. We are calling this new version SpmsV4 which use a RDBMS (Relational Database Management System) to store, update and retrieve necessary data and generate documents such as Course Outlines, Exam Question and Answer Scripts as well as other necessary required data to monitor student performance and produce other OBE (Outcome-Based Education) reports. It's important to design interface that are intuitive, user-friendly and tailored to each user's specific needs. There are lots of consideration while using for interface design techniques, such as wireframes, mockups and prototypes. The interface is also needed to get feedback from users and iterate designs for interface usability, functionality

and collaboration, to ensure that all users can easily access and manipulate the data they need to effectively manage their courses and programs.

2

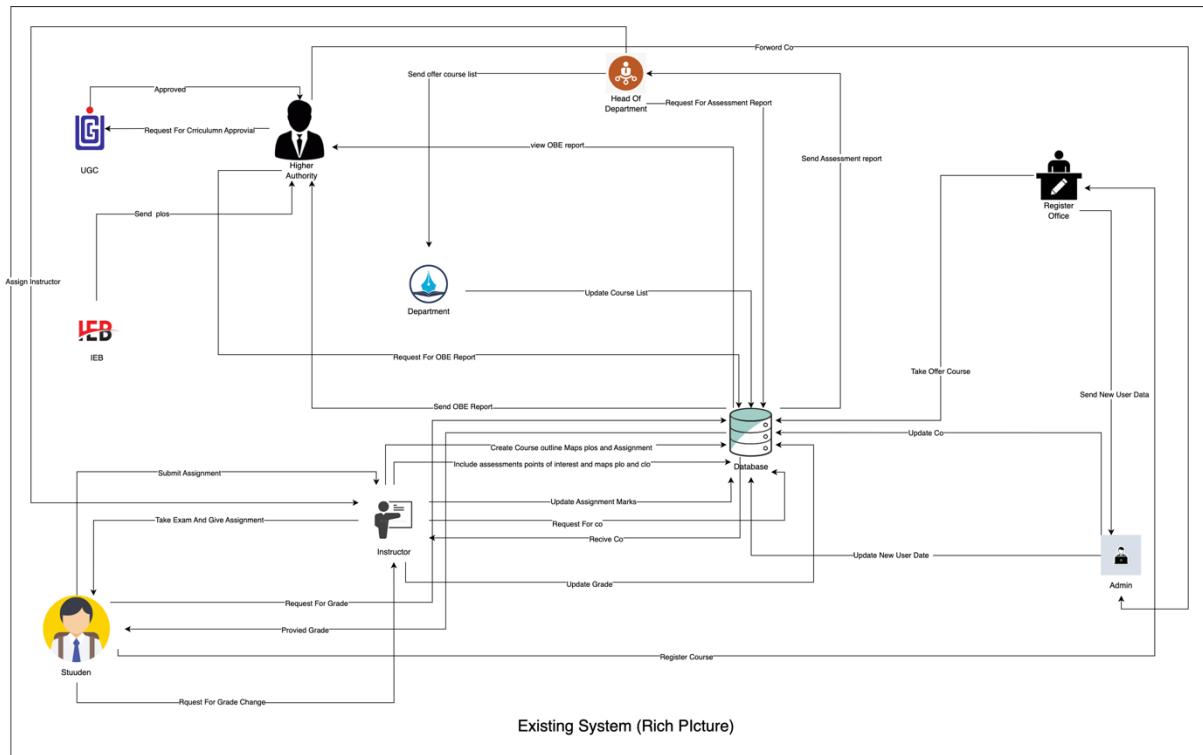
REQUIREMENT ANALYSIS

The act of obtaining, comprehending and capturing the needs and expectations of stakeholders for a certain system or piece of software is known as requirement analysis. It entails determining the system's operational or commercial goals, outlining its functional and non-functional needs and creating a set of requirements that the system must follow. The basis for all ensuing activities, including design, development, testing, and deployment, is laid by the requirement analysis phase of the software development lifecycle. The basic requirement analysis starts from the process of capturing requirements in a thorough. Before moving on to the following stage, it is important to check the criteria and get there from all parties.

Examining Requirements: Examining the gathered requirements to make sure they are thorough, consistent, and practical. Requirement analysis assists in ensuring that the finished product satisfies the needs and expectations of the stakeholders and provides commercial value. Early requirement identification and comprehension can assist to prevent costly errors, rework, and delays later in the development process.

2.1 Rich Picture -> Existing (AS-IS) System

A rich picture is a diagrammatic depiction of a complicated circumstance or issue that aids in understanding the different elements and interconnections at play. It is a technique that's frequently used in organizational analysis and systems thinking to help in understanding and communicating complicated problems.



In this Rich picture stakeholders are:

1. UGC,
2. IBE,
3. Higher Authority,
4. Department,
5. Register Office,
6. Admin,
7. Department Head,
8. Instructor and
9. Student.

The main storage is:

1. Database

2.2 Six Element Analysis -> Existing (AS-IS) System

From Rich picture we identified 7 key processes:

1. Calculate Course Outcomes (COs) and Program Learning Outcomes (PLOs),
2. Record Student Evaluation Data,
3. Generate OBE Marksheets and Course Evaluation Report,

4. View result and download Transcript,
5. Create Student/Faculty/Staff accounts and insert/adjust necessary data,
6. Analyze student's records, OBE Marksheets and Course evaluation reports over time to see student performance pattern and
7. Review and grade change request.

We are using six-element analysis to analyze the impact of six elements in a process. And the six elements are:

1. Human,
2. Non-Computing Hardware,
3. Computing Hardware,
4. Software,
5. Database and
6. Network and Communications.

Process	System Roles					
	Human	Non-Computing Hardware	Computing Hardware	Software	Database	Network and Communication
1. Calculate Course Outcomes (COs) and Program Learning Outcomes (PLOs)	A. IBE/UGC/ Higher Authority: 1. Submit an accreditation requirement containing defined PLOs to the Department office. B. Department Office: 1. Receive an accreditation requirement from IEB. 2. Send the accreditation	A. Pen and Paper: 1. Used to write down preliminary thoughts and ideas.	A. Computer: 1. Used to make softcopies of a course outline, CO. B. Printer: 1. Used to make hard copies of documents such as Course outcomes (CO).	A. MS Word: 1. Used to create detailed lesson plans and course assessment reports, mapping course outcome (CO) to program learning outcome (PLO).		A. Internet and Email: 1. Used to communicate with UGC / IBE or other topic-related partners to discuss key issues related to mapping Course outcome (CO) and Program learning outcome (PLO). B. Others: 1. For communication with other partners use a

	<p>requirement containing defined PLOs to Faculty Member / Instructor and direct them to design and coordinate courses as instructed.</p> <p>C. Faculty Member / Instructor:</p> <ol style="list-style-type: none"> 1. List course content and COs. 2. Link CO to PLO and specific questions for examinations and projects. 3. Design a course evaluation report using course content, course outline, and CO. 		<p>B. Excel Sheet:</p> <ol style="list-style-type: none"> 1. Used to outline specific questions in the exam and project to specific Course outcome (CO). 		<p>phone and physical media to discuss important topics for mapping Course outcome (CO) and Program learning outcome (PLO).</p>
2. Record Student Evaluation Data	<p>A. Faculty Member / Instructor:</p> <ol style="list-style-type: none"> 1. Assign project work and assignments and also take tests and exams all through the semester. 2. Record student evaluation data throughout the student's semester for each task 	<p>A. Pen and Paper:</p> <ol style="list-style-type: none"> 1. Used to write down data for efficient work. 	<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used to make softcopies of all assessment data for specific courses. 	<p>A. Excel Sheet:</p> <ol style="list-style-type: none"> 1. Record the required evaluation data and grades in an Excel Sheet. <p>B. IRAS:</p> <ol style="list-style-type: none"> 1. Upload the student's final grade to IRAS for the student 	<p>A. Department Storage:</p> <ol style="list-style-type: none"> 1. The record hard copies of student evaluation data and grades retained by the Department and Register's office for future use. <p>A. Internet:</p> <ol style="list-style-type: none"> 1. Used to access IRAS and upload grades of students.

	<p>(assignment, tests, mid and final exams, project) in both soft and hard copy. And also record marks for each specific question for both mid and final exams.</p> <p>3. Calculate total marks for assignments, tests, and mid and final exams, and assign final grades to each student for specific courses.</p> <p>4. Convert midterm and final exam grades and put all of each student's grades for a course into one transcript to classify students and upload student final grades to IRAS.</p> <p>5. Submit the transcript to the Department office and the Register's office.</p>		<p>or Register's office to view.</p>	<p>B. IRAS Database Server:</p> <p>1. IRAS records and maintain students' grade in the database.</p>	
--	--	--	--------------------------------------	---	--

<p>3. Generate OBE Marksheets and Course Evaluation Report</p>	<p>A. Faculty Member / Instructor:</p> <ol style="list-style-type: none"> 1. Calculate marks from each task (assignment, tests, mid and final exams, project) to calculate the total mark received for CO. 2. Calculate the total percentage obtained for all CO on the OBE transcript. 3. Report if the student achieves yes or no which is greater or equal to 40% and if a student receives a PLO for the CO involved. 4. Prepare a statement and analysis of the number of students who can receive certain CO, PLO, and other documents containing the required information and data. 5. Design a course evaluation report according 	<p>A. Pen and Paper:</p> <ol style="list-style-type: none"> 1. OBE transcript save on paper and additional marking may be required for further clarifications. 	<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used to create soft copies of OBE transcript and course evaluation reports. <p>B. Printer:</p> <ol style="list-style-type: none"> 1. Used to print hard copies of the final version of OBE transcripts and course evaluation reports. 	<p>A. Modified Excel Sheet:</p> <ol style="list-style-type: none"> 1. Used to automatically calculate PLO success / failure for students by Faculty Members / Instructors. <p>B. MS Word:</p> <ol style="list-style-type: none"> 1. Used to make course evaluation report soft copies. 	<p>A. Department Storage:</p> <ol style="list-style-type: none"> 1. Hard copies of student evaluation data records and final grads will be kept in the Department for future use. <p>B. Register's Office Storage:</p> <ol style="list-style-type: none"> 1. Hard copies of OBE transcript, Course evaluation report and other documents submitted by Department office will be retained for future use. 	<p>A. Internet:</p> <ol style="list-style-type: none"> 1. An online platform such as google docs, google sheet can be used to process the OBE evaluation data.
---	---	--	---	--	--	--

	<p>to the Course outline, Course content, and COs.</p> <p>6. Submit the final version of the OBE transcript to the Department Office.</p> <p>B. Department Office:</p> <ol style="list-style-type: none"> 1. Send OBE transcripts, course evaluation reports, and other documents to the Register's office. 2. Keep OBE transcripts and evaluation reports in the department. <p>C. Register's Office:</p> <ol style="list-style-type: none"> 1. Keep OBE transcripts and evaluation reports and other documents in the Register's office. 				
4. View result and download Transcript	<p>A. Student:</p> <ol style="list-style-type: none"> 1. Login to IRAS. 2. Search semester wise result for expected semester and see grades. 	<p>A. Pen and Paper:</p> <ol style="list-style-type: none"> 1. Transcript can be printed in paper and maybe required for additional marking. 	<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used for accessing IRAS. <p>B. Printer:</p> <ol style="list-style-type: none"> 1. Used for print transcript. 	<p>A. IRAS:</p> <ol style="list-style-type: none"> 1. Keep grade for each course and gives the online client interface for seeing 	<p>A. Register's Office Storage:</p> <ol style="list-style-type: none"> 1. Student information is kept in hardcopies for future use. <p>A. Internet and Email:</p> <ol style="list-style-type: none"> 1. Used to connect with IRAS and mailing softcopies of documents.

	<p>3. Download the transcript via browser into drive.</p> <p>B. Register's Office:</p> <ol style="list-style-type: none"> 1. Access IRAS and view students' grades if needed. 2. Download the transcript via browser into drive. 			grades and transcripts.	<p>B. IRAS Database Server:</p> <ol style="list-style-type: none"> 1. Used to keep, maintain, receive and modify student grade and other information. <p>C. Remote Database Server:</p> <ol style="list-style-type: none"> 1. For accessing user interface and web pages saved by remote web server. 	
5. Create Student/ Faculty/ Staff accounts and insert / adjust necessary data	<p>A. Admin:</p> <ol style="list-style-type: none"> 1. Information about new students is collected from the registration process. 2. Information about new faculty member is collected from HR. 3. Create accounts for students, faculty and also able to adjust account details as needed. 	<p>A. Pen and Paper:</p> <ol style="list-style-type: none"> 1. Can be used to copy / write important credentials to create an account. 	<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used to access and add / modify data in IRAS. 	<p>A. IRAS:</p> <ol style="list-style-type: none"> 1. Used as client interface to connect with student/faculty information. 	<p>A. Register's Office Storage:</p> <ol style="list-style-type: none"> 1. Student / Faculty information is kept in hard copy for future use. <p>B. IRAS Database Server:</p> <ol style="list-style-type: none"> 1. Used to keep, maintain, receive, and modify student and faculty information. 	<p>A. Internet:</p> <ol style="list-style-type: none"> 1. Used to connect with IRAS and remote database server.

					C. Remote Database Server: 1. For accessing user interface and web pages saved by the remote web server.	
6. Analyze student's records, OBE Marksheets and Course evaluation reports over time to see student performance pattern	<p>A. IBE / UGC: 1. Notify university administrators of a due date inside which OBE transcript, course evaluation report and other documents are required for quality review to make the necessary improvements to any programs and if the government officials will be visiting the campus.</p> <p>2. Visit any relevant department in the university to obtain required documents and reports.</p> <p>B. Department: 1. Department will collect the required documents, OBE transcript, course</p>	<p>A. Pen and Paper: 1. Used to record / note key points from the report and hard copies of the report is also used.</p>	<p>A. Computer: 1. Used to view and share soft copies of OBE transcript, course evaluation report and other required documents.</p>		<p>A. Department Storage: 1. Used to keep, maintain and to retrieve hard copies of OBE transcript, course evaluation report, student performance pattern and other required documents.</p>	<p>A. Internet and Email: 1. An online platform such as google docs, google sheet can be used to view reports and email can be used for sharing reports.</p>

	<p>evaluation reports within the specified time by IBE / UGC.</p> <p>2. Give all the essential documents to government authorities (IBE / UGC) and VC of the university.</p> <p>3. Find the issue based on student performance pattern where instructional resources can be modified or improved.</p> <p>B. Office of VC:</p> <p>1. Based on request, review OBE transcript, course evaluation reports and student performance pattern.</p>					
7. Review and grade change request	<p>A. Student:</p> <p>1. Request for review and change of grade to the faculty member / instructor.</p> <p>B. Faculty Member / Instructor:</p> <p>1. Review answer sheets, other tasks upon request and if</p>	<p>A. Pen and Paper:</p> <p>Used to record / note key points and marking in the hard copy of the answer sheet.</p>	<p>A. Computer:</p> <p>1. Used for communication between student, faculty and admin.</p>	<p>A. IRAS:</p> <p>1. Admin uses IRAS for grade change.</p>	<p>A. IRAS Database Server:</p> <p>1. Used to update new grade.</p> <p>B. Department Storage:</p> <p>1. Used to update new grade in hard copy.</p>	<p>A. Internet, Email and Phone:</p> <p>1. Email and phone is basically used for communication.</p>

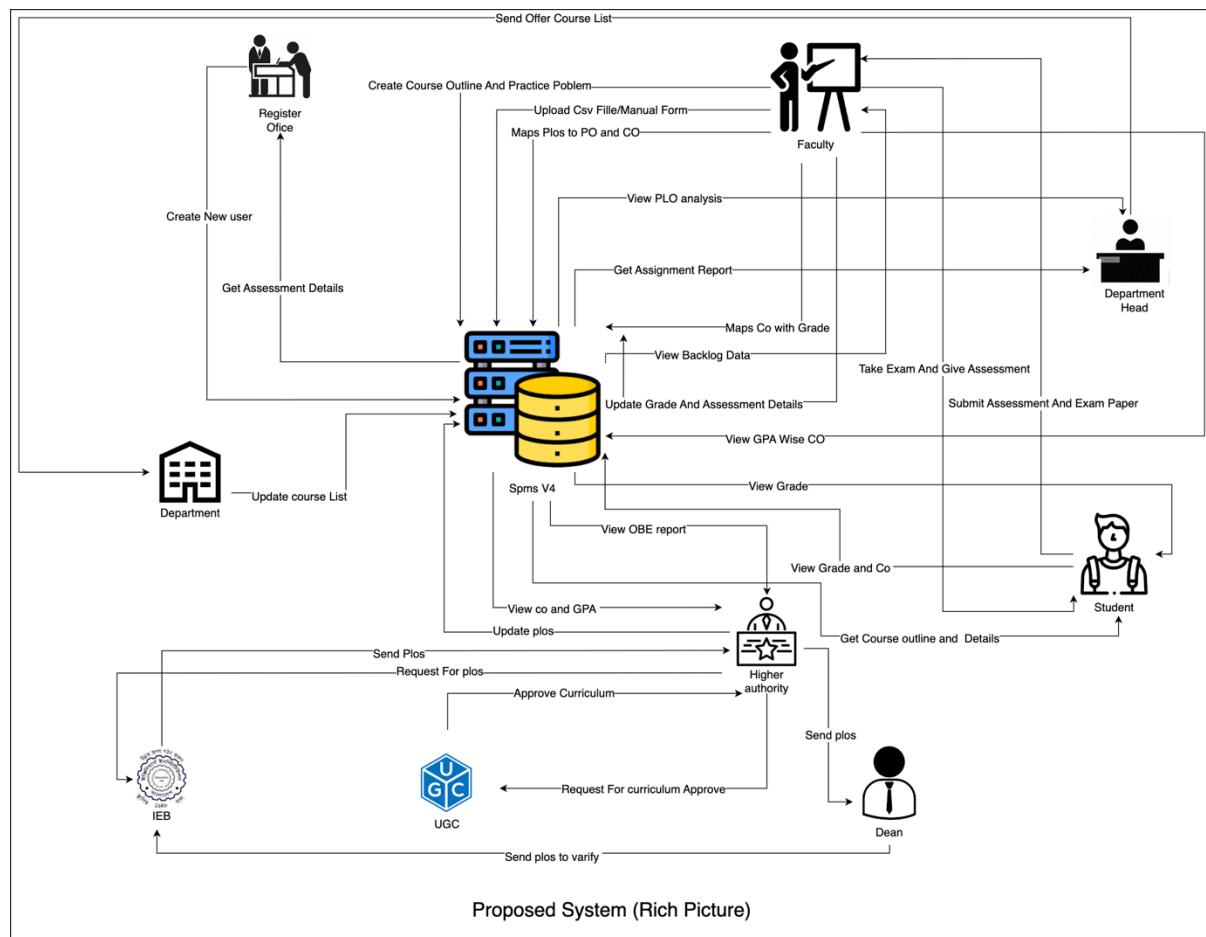
	<p>request is valid then send a request to admin for grade change but if request is not valid then no request to admin is needed.</p> <p>C. Admin:</p> <ol style="list-style-type: none"> 1. Modify student grade based on instructor request. 				
--	--	--	--	--	--

2.3 Identification and Analysis of problem

Process Name	Stockholders	Problems	Analysis	Proposed Solution
Record student evaluation data	<ol style="list-style-type: none"> 1. Admin 2. Faculty member 3. Student 	<ol style="list-style-type: none"> 1. Project work, assignments, question papers, and answer scripts condition. 2. Giving and receiving process. 3. Storage problem. 	<ol style="list-style-type: none"> 1. The hard copies of answer scripts, project reports, and question papers that are being stored physically may get damaged or lost. 2. The process of checking the answer script or receiving it and then recording it makes the process slow. 3. Physical storage may be a problem for hard copies. 	<p>The answer scripts and question paper are stored in the database, so it will be easy for him to find and search by student id. Which will be time reducing. Online submission of answer scripts saves time. Submitting online will be a good option for reducing storage problem.</p>

Generate OBE marksheet and course evaluation report	1. Admin 2. Faculty member 3. Student	1. Storing hard copies and soft copies becomes hard to manage.	1. Storing softcopies and hardcopies can become extremely difficult to manage when the organization has been operating for years. It also gets increasingly tedious to track documents to study student performance trends for a certain timeline. Updating information for a specific document would require tracking them, which in turn would make it harder to retrieve them and would require personnel to update various copies.	These problems can be fixed by maintaining these data tables in our database and giving necessary departments, offices, and outside parties (IEB) access via a user interface to view and print them.
--	---	--	--	---

2.4 Rich Picture -> Proposed (TO-BE) System



In this Rich picture stakeholders are:

1. UGC,
2. IBE,
3. Higher Authority,
4. Department,
5. Register Office,
6. Dean,
7. Department Head,
8. Faculty and
9. Student.

The main storage is:

2. Spms V4.

2.5 Proposed process and Six Element Analysis

From Rich picture we identified 10 key processes:

1. Make and deliver course outlines and lesson plans,
2. Statistics of student enrollment under VC / Dean / Head of Department supervision,
3. Include question sheets of quizzes, assignments and exams and evaluate replays of answer sheet,
4. Statistics of student performance patterns by GPA within any course,
5. Statistics of student performance patterns by GPA under faculty supervision,
6. Statistics of student performance patterns by GPA under VC / Dean / Head of Department supervision,
7. Course Outcome (CO) and Program Outcome (PO),
8. Program Learning Outcome (PLO),
9. Statistics of Courses, Programs, Departments and Schools CLO and PLO and
10. Statistics of average CLO / PLO of students under Department Supervisions of total number of attempts.

We are using six-element analysis to analyze the impact of six elements in a process. And the six elements are:

1. Human,
2. Non-Computing Hardware,
3. Computing Hardware,
4. Software,
5. Database and
6. Network and Communications.

process	System Roles					
	Human	Non-Computing Hardware	Computing Hardware	Software	Database	Network and Communication
1. Make and deliver course outlines and lesson plans	A. Faculty: 1. Log in to SpmsV4 with username and password. 2. Select option for course outline and lesson plan. 3. Select option for upload and will submit necessary data. 4. After clicking submit button, inserted data will be uploaded to database.		A. Computer: 1. Used to log in to SpmsV4. B. Printer: 1. Used to print hard copies of course outline on demand.	A. SpmsV4: 1. Used as an interface to work with data in the SpmsV4 database. B. Printer: 1. Used to print hard copies of course outline on demand.	A. SpmsV4 database server: 1. Used to keep, maintain, modify and use student grade and other information by system admin.	A. Internet: 1. Used to access SpmsV4.

	B. Student: 1. Log in to SpmsV4 with username and password. 2. Select option for course outline and lesson plan. 3. Select option for course and course outline and lesson plan will be shown with a download option.				
2. Statistics of student enrollment under VC / Dean / Head of Department supervision	A. VC: 1. Log in to SpmsV4 with username and password. 2. Select option for enrollment. 3. Select option for semester and educational year and a submit button. 4. After clicking submit		A. Computer: 1. Used to log in to SpmsV4. B. Printer: 1. Used to print hard copies of student enrollment data on demand.	A. SpmsV4: 1. Used as an interface to work with data in the SpmsV4 database. B. Printer: 1. Used to print hard copies of student enrollment data on demand.	A. SpmsV4 database server: 1. Used to keep, maintain, modify and use student grade and other information by system admin. A. Internet: 1. Used to access SpmsV4.

	<p>button, statistics of data will be shown.</p> <p>B. Dean:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for enrollment. 3. Select option for semester and educational year and a submit button. 4. After clicking submit button, statistics of data will be shown. <p>C. Head of Department:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for enrollment. 				
--	---	--	--	--	--

	<p>3. Select option for semester and educational year and a submit button.</p> <p>4. After clicking submit button, statistics of data will be shown.</p>				
3. Include question sheets of quizzes, assignments and exams and evaluate replays of answer sheet	<p>A. Faculty:</p> <p>1. Log in to SpmsV4 with username and password.</p> <p>2. Select option for question sheets.</p> <p>3. Select option for course and section and add question sheet and click assign button.</p> <p>4. And for gradings select option for evaluation and submit marks.</p>		<p>A. Computer:</p> <p>1. Used to log in to SpmsV4.</p> <p>B. Printer:</p> <p>1. Used to print hard copies of question sheets and grade sheets on demand.</p>	<p>A. SpmsV4:</p> <p>1. Used as an interface to work with data in the SpmsV4 database.</p> <p>B. Printer:</p> <p>1. Used to print hard copies of question sheets and grade sheets on demand.</p>	<p>A. SpmsV4 database server:</p> <p>1. Used to keep, maintain, modify and use student grade and other information by system admin.</p> <p>A. Internet:</p> <p>1. Used to access SpmsV4.</p>

	B. Student: 1. Log in to SpmsV4 with username and password. 2. Select option for assigned task. 3. Select any task and upload answer and click submit button. 4. And for checking gradings select option for grade and also select course and exams. Grade will be shown.				
4. Statistics of student performance patterns by GPA within any course	A. VC / Dean / Head of Department: 1. Log in to SpmsV4 with username and password. 2. Select option for course student performance . All the	A. Computer: 1. Used to log in to SpmsV4. B. Printer: 1. Used to print hard copies of student performance data sheets of any course on demand.	A. SpmsV4: 1. Used as an interface to work with data in the SpmsV4 database.	A. SpmsV4 database server: 1. Used to keep, maintain, modify and use student grade and other information by system admin.	A. Internet: 1. Used to access SpmsV4.

	<p>courses under users' supervision will be shown.</p> <p>3. Select any course or all courses and click submit button.</p> <p>4. Statistics of selected courses will be displayed graphically.</p> <p>B. Faculty:</p> <p>1. Log in to SpmsV4 with username and password.</p> <p>2. Select option for course student performance . All the courses under users' supervision will be shown.</p> <p>3. Select any course and click submit button.</p> <p>4. Statistics of selected course will be displayed graphically.</p>				
--	--	--	--	--	--

<p>5. Statistics of student performance patterns by GPA under faculty supervision</p>	<p>A. VC / Dean / Head of Department:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for faculty student performance . All the courses under selected faculties supervision will be shown. 3. Select any course or all courses and click submit button. 4. Statistics of selected courses will be displayed graphically. <p>B. Faculty:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for faculty 		<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used to log in to SpmsV4. <p>B. Printer:</p> <ol style="list-style-type: none"> 1. Used to print hard copies of student performance data sheets under faculty supervision on demand. 	<p>A. SpmsV4:</p> <ol style="list-style-type: none"> 1. Used as an interface to work with data in the SpmsV4 database. 	<p>A. SpmsV4 database server:</p> <ol style="list-style-type: none"> 1. Used to keep, maintain, modify and use student grade and other information by system admin. 	<p>A. Internet:</p> <ol style="list-style-type: none"> 1. Used to access SpmsV4.
--	--	--	--	--	---	--

	<p>student performance . All the courses under users' supervision will be shown.</p> <p>3. Select any course and click submit button.</p> <p>4. Statistics of selected course will be displayed graphically.</p>				
6. Statistics of student performance patterns by GPA under VC / Dean / Head of Department supervision	A. VC / Dean / Head of Department: 1. Log in to SpmsV4 with username and password. 2. Select option for student performance . All the courses under users' supervision will be shown. 3. Select any course or all courses and click submit button.	A. Computer: 1. Used to log in to SpmsV4. B. Printer: 1. Used to print hard copies of student performance data sheets under VC / Dean / Head of Department supervision on demand.	A. SpmsV4: 1. Used as an interface to work with data in the SpmsV4 database. A. SpmsV4 database server: 1. Used to keep, maintain, modify and use student grade and other information by system admin.	A. Internet: 1. Used to access SpmsV4.	

	4. Statistics of selected courses will be displayed graphically.				
7. Course Outcome (CO) and Program Outcome (PO)	<p>A. VC / Dean / Head of Department:</p> <p>1. Log in to SpmsV4 with username and password.</p> <p>2. Select option for CO / PO and select, semester, course / program and click submit button.</p> <p>3. CO / PO earned by students from that course / program will be shown graphically.</p> <p>B. Faculty:</p> <p>1. Log in to SpmsV4 with username and password.</p> <p>2. Select option for CO / PO and all the</p>		<p>A. Computer:</p> <p>1. Used to log in to SpmsV4.</p> <p>B. Printer:</p> <p>1. Used to print hard copies of student CO / PO reports on demand.</p>	<p>A. SpmsV4:</p> <p>1. Used as an interface to work with data in the SpmsV4 database.</p> <p>A. SpmsV4 database server:</p> <p>1. Used to keep, maintain, modify and use student CO / PO's atomic data and other information by system admin.</p>	<p>A. Internet:</p> <p>1. Used to access SpmsV4.</p>

	<p>courses of user supervision will be shown course / program and click submit button.</p> <p>3. CO / PO earned by students from that course / program will be shown graphically.</p> <p>C. Student:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for CO / PO. 3. All CO / PO that student achieved will be displayed graphically. 				
8. Program Learning Outcome (PLO)	<p>A. VC / Dean / Head of Department:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username 	<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used to log in to SpmsV4. <p>B. Printer:</p> <ol style="list-style-type: none"> 1. Used to print hard copies of 	<p>A. SpmsV4:</p> <ol style="list-style-type: none"> 1. Used as an interface to work with data in the SpmsV4 database. 	<p>A. SpmsV4 database server:</p> <ol style="list-style-type: none"> 1. Used to keep, maintain, modify and use student PLO's atomic data and 	<p>A. Internet:</p> <ol style="list-style-type: none"> 1. Used to access SpmsV4.

	<p>and password.</p> <p>2. Select option for PLO and select, semester, course / program and click submit button.</p> <p>3. PLO earned by students from that course / program will be shown graphically.</p> <p>B. Faculty:</p> <p>1. Log in to SpmsV4 with username and password.</p> <p>2. Select option for PLO and all the courses of user supervision will be shown course / program and click submit button.</p> <p>3. PLO earned by students from that course / program will</p>		student PLO reports on demand.		other information by system admin.	
--	---	--	--------------------------------	--	------------------------------------	--

	<p>be shown graphically.</p> <p>C. Student:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for PLO. 3. All PLO that student achieved will be displayed graphically. 					
<p>9. Statistics of Courses, Programs, Departments and Schools CLO and PLO</p>	<p>A. VC / Dean / Head of Department:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for CLO / PLO and select, course or program or department or school and click submit button. 3. Statistics of selected options data 		<p>A. Computer:</p> <ol style="list-style-type: none"> 1. Used to log in to SpmsV4. <p>B. Printer:</p> <ol style="list-style-type: none"> 1. Used to print hard copies of reports on demand. 	<p>A. SpmsV4:</p> <ol style="list-style-type: none"> 1. Used as an interface to work with data in the SpmsV4 database. 	<p>A. SpmsV4 database server:</p> <ol style="list-style-type: none"> 1. Used to keep, maintain, modify and use student CLO / PLO's atomic data and other information by system admin. 	<p>A. Internet:</p> <ol style="list-style-type: none"> 1. Used to access SpmsV4.

	<p>will be displayed graphically.</p> <p>B. Faculty:</p> <ol style="list-style-type: none"> Log in to SpmsV4 with username and password. Select option for CLO / PLO and all the courses of user supervision will be shown course / program and click submit button. 				
<p>10. Statistics of average CLO / PLO of students under Department Supervision s of total number of attempts</p>	<p>A. VC / Dean / Head of Department:</p> <ol style="list-style-type: none"> Log in to SpmsV4 with username and password. Select option for department wise CLO / PLO and select semester, department and click submit button. 	<p>A. Computer:</p> <ol style="list-style-type: none"> Used to log in to SpmsV4. <p>B. Printer:</p> <ol style="list-style-type: none"> Used to print hard copies of student CLO / PLO reports on demand. 	<p>A. SpmsV4:</p> <ol style="list-style-type: none"> Used as an interface to work with data in the SpmsV4 database. 	<p>A. SpmsV4 database server:</p> <ol style="list-style-type: none"> Used to keep, maintain, modify and use student CLO / PLO's atomic data and other information by system admin. 	<p>A. Internet:</p> <ol style="list-style-type: none"> Used to access SpmsV4.

	<p>3. Statistics of selected semester of a department will be displayed graphically.</p> <p>B. Faculty:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for department wise CLO / PLO and select semester and click submit button. 3. Statistics of selected semester of a department will be displayed graphically. <p>C. Student:</p> <ol style="list-style-type: none"> 1. Log in to SpmsV4 with username and password. 2. Select option for department 				
--	---	--	--	--	--

	<p>wise CLO / PLO and select semester and click submit button.</p> <p>3. Statistics of selected semester of a department will be displayed graphically.</p>				
--	---	--	--	--	--

3

LOGICAL SYSTEM DESIGN

3.1 Business Rules – SpmsV4

1. A student must have one department. A STUDENT has studentID, name consisting of fname and lname, dateOfBirth, gender, email, phone, address, deptID, programID, enrollmentYear, enrollmentSemester, and password. A department must have one or many students.
2. A student may perform many registrations. A REGISTRATION includes registrationID, sectionID, studentID, enrolled_course, enrolled_section, educational_semester, educational_year. Registration must be performed by exactly one student.
3. A student must complete exactly one program. A PROGRAM has programID, programName, deptID. A program must be completed by one or many students.

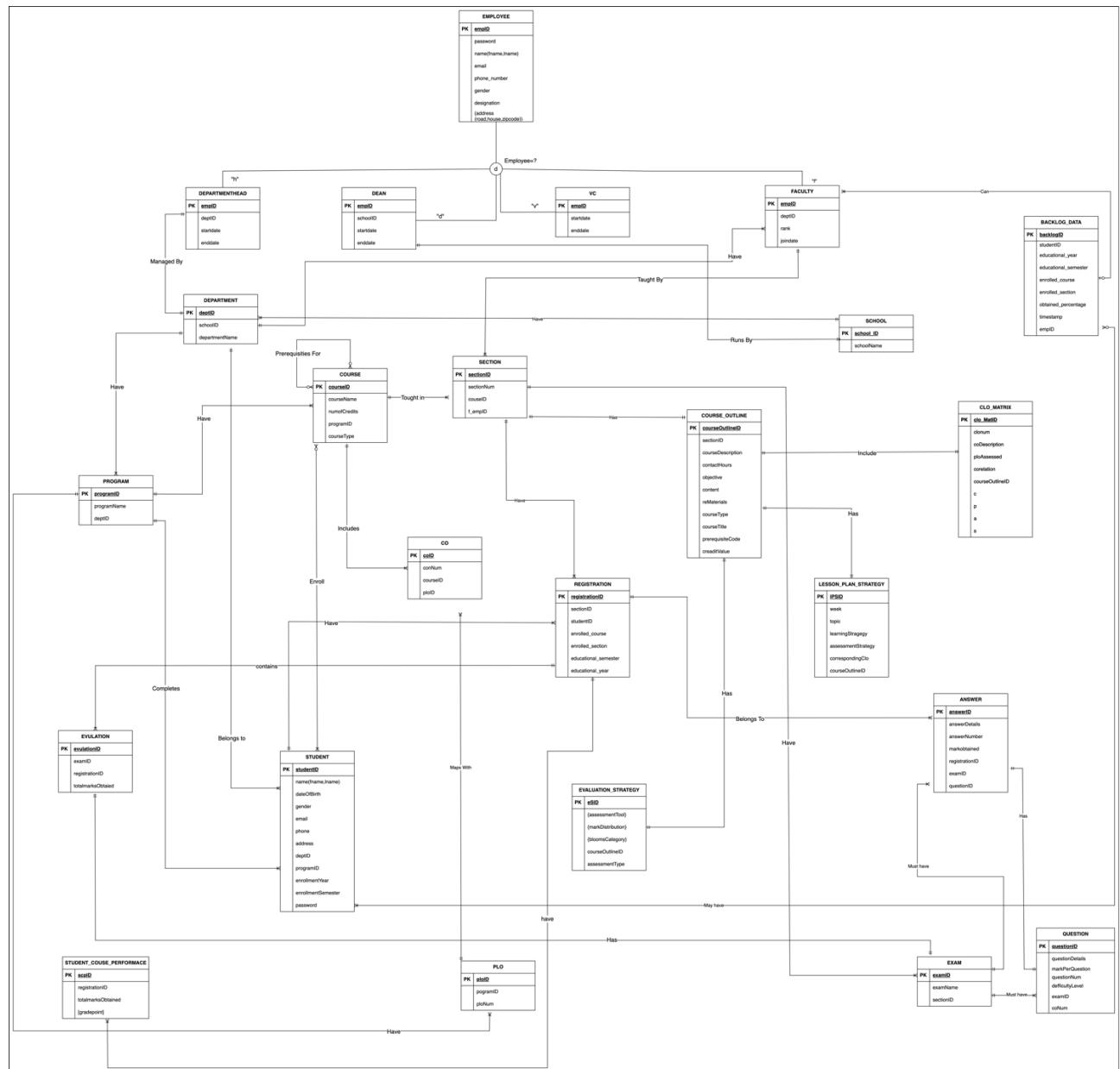
4. A section mandatorily has many registrations. A registration must have at least one section. A Section includes sectionID, sectionNum, courseID, f_empID.
5. A course must have one program. A program must contain one or many courses. A COURSE contains courseID, courseName, numofCredits, courseType, programID. Many courses are perquisites for many courses. A course must be taught in one or many sections. A section must be taught by exactly one course.
6. A course must include one or many CO. A CO has coID, coNum, courseID, ploID. A CO must be included in exactly one course.
7. A registration may belong to many evaluations. An evaluation mandatorily belongs to one registration. An EVALUATION contains evaluationID, examID, registrationID, totalmarksObtained.
8. A CO must map with one PLO. A PLO's must map with one or many CO's. PLO includes ploID, ploNum, programID. A PLO must contain one program but a program must contain one or many PLO's.
9. A program must belong to one department. A department must belong to one or many programs. A DEPARTMENT contains deptID, departmentName, schoolID.
10. A department must contain one school. A school must contain one or many departments. A SCHOOL includes school_ID, schoolName.
11. An employee has four sub-types with Department Head, Dean, VC, Faculty. An EMPLOYEE includes empID, password, name consist of fname and lname, email, phone_number, gender, designation, addresses with road, house and zip code number.
12. A school must be run by exactly one. A dean must run exactly one school. A DEAN has empID, schoolID, startdate, enddate.
13. A Department must be run by exactly one Department head. A department head must manage exactly one department. A DEPARTMENTHEAD includes empID, deptID, startdate, enddate.

14. A Faculty must have exactly one Department. A department must have one or many Faculties. A FACULTY includes empID, deptID, rank, joindate. A faculty may teach many sections. A section must be taught by exactly one faculty.
15. A course outline belongs to exactly one section. A section must have exactly one course outline. A COURSE_OUTLINE includes courseOutlineID, sectionID, contactHours, courseDescription, objective, content, refMaterials, courseType, courseTitle, prerequisiteCode, creditValue.
16. A course outline must have exactly one CLO Matrix. A CLO matrix belongs to exactly one course outline. A CLO_MATRIX includes clo_MatID, cloNum, coDescription, ploAssessed, correlation, courseOutlineID, c, p, a, s.
17. A Lesson Plan Strategy must have exactly one Evaluation strategy. An Evaluation strategy must have exactly one Lesson Plan Strategy. A LESSON_PLAN_STRATEGY includes IPSID, week, topic, learning Strategy, assessmentStrategy, correspondingClo, courseOutlineID.
18. An exam has exactly one evaluation. An Evaluation for an exam is done exactly once. An EXAM includes examID, examName, sectionID. A section must have one or many exams. An exam must be held in one section.
19. An exam must have one or many questions. Every question must belong to exactly one exam. A QUESTION includes questionID, questionDetails, markPerQuestion, questionNum, difficultlyLevel, examID, coNum. A Question is answered exactly once. An answer has exactly one question.
20. An answer must have one exam. An ANSWER includes answerID, answerDetails, answerNumber, markobtained, registrationID, examID, questionID. An exam must have multiple answers. A registration must have many answers but an answer has one registration.
21. A student course performance evaluation is done for registration exactly once. A registration must have many students course performance evaluation. A

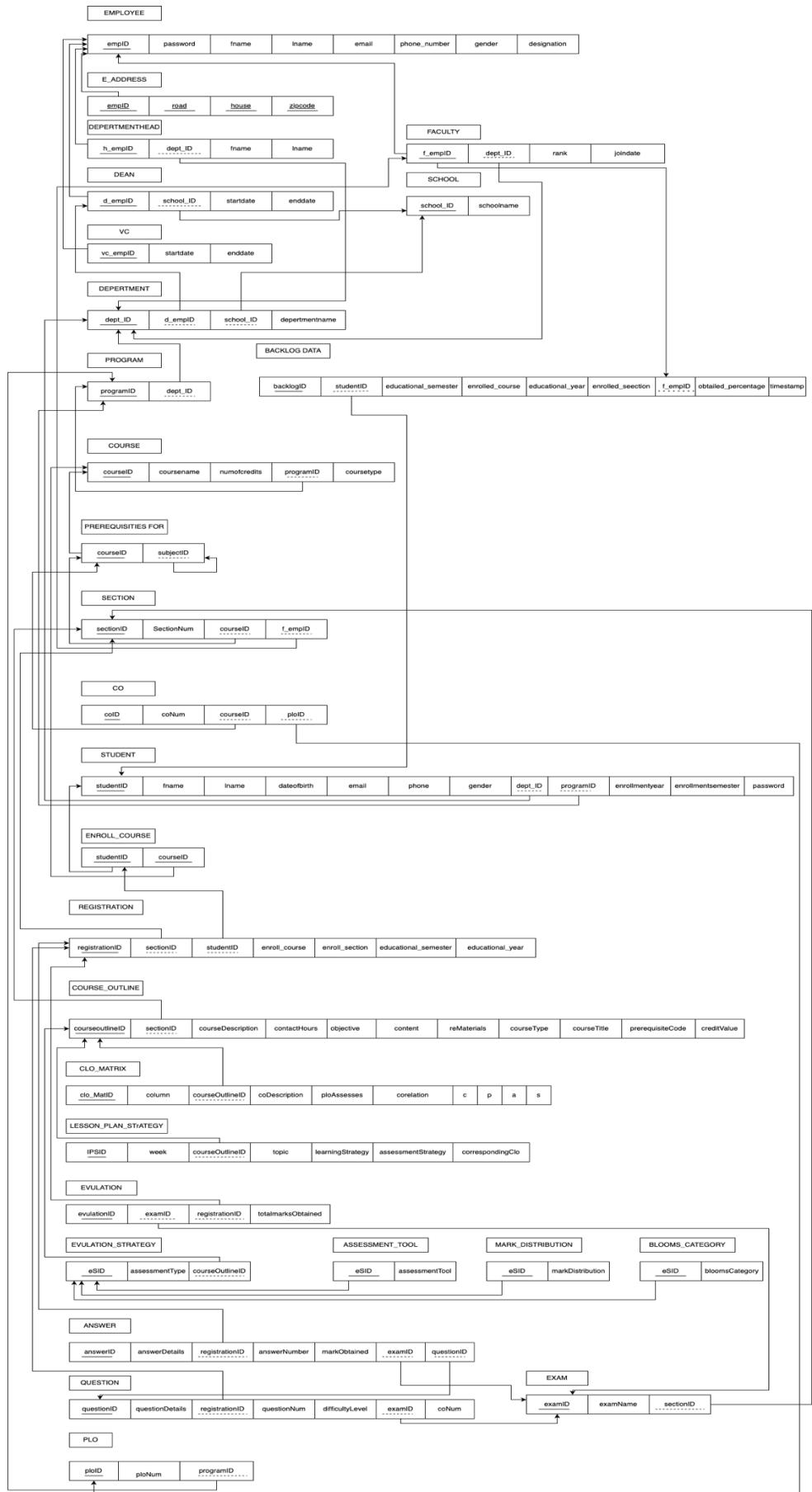
STUDENT_COURSE_PERFORMANCE includes scpID, registrationID, totalmarksObtained and gradepoint that is calculated from grade.

22. A faculty may enter backlog data only once. A backlog data must be entered by many faculties. A BACKLOG_DATA includes backlogID, studentID, educational_year, educational_semester, enrolled_course, enrolled_section, obtained_percentage, timestamp, empID. Every backlog data must need one or many students. A student may have one or many backlog data.

3.2 Entity Relationship Diagram (ERD)



3.3 Relational Schema



3.4 Normalization

EMPLOYEE (e)	empID	e1
	password	e2
	name	e3
	email	e4
	phone_number	e5
	gender	e6
	designation	e7
	address	e8
VC (v)	vc_empID	v1
	startdate	v2
	enddate	v3
DEAN (n)	d_empID	n1
	schoolID	sc1
	startdate	n2
	enddate	n3
DEPARTMENTHEAD (h)	h_empID	h1
	dept_ID	d1
	startdate	h2
	enddate	h3
FACULTY (f)	f_empID	f1
	dept_ID	d1
	rank	f2

	joindate	f3
SCHOOL (sc)	school_ID	sc1
	schoolname	sc2
DEPARTMENT (d)	dept_ID	d1
	schoolID	sc1
	departmentname	d2
PROGRAM (p)	programID	p1
	programName	p2
	dept_ID	d1
COURSE (c)	courseID	c1
	courseName	c2
	numofCredits	c3
	programID	p1
	courseType	c4
SECTION (st)	sectionID	st1
	sectionNum	st2
	courseID	c1
	f_empID	f1
REGISTRATION (r)	registrationID	r1
	sectionID	st1
	studentID	s1
	enrolled_course	r2
	enrolled_section	r3
	educational_semester	r4

	educational_year	r5
STUDENT (s)	studentID	s1
	name	s2
	dateOfBirth	s3
	gender	s4
	email	s5
	phone	s6
	address	s7
	deptID	d1
	programID	p1
	enrollmentYear	s8
COURSE_OUTLINE (cu)	enrollmentSemester	s9
	password	s10
	courseOutlineID	cu1
	sectionID	st1
	courseDescription	cu2
	contactHours	cu3
	objective	cu4
	content	cu5
	reMaterials	cu6
	courseType	cu7
	courseTitle	cu8
	prerequisiteCode	cu9
	craditValue	cu10

LESSON_PLAN_STRATEGY (l)	lPSID	l1
	week	l2
	topic	l3
	learningStragegy	l4
	assessmentStragegy	l5
	corrospondingClo	l6
	courseOutlineID	cu1
EXAM (ex)	examID	ex1
	examName	ex2
	sectionID	st1
QUESTIION (q)	questionID	q1
	questionDetails	q2
	markPerQuestion	q3
	questionNum	q4
	deficultyLevel	q5
	examID	ex1
	coNum	q6
ANSWER (a)	answerID	a1
	answerDetails	a2
	answerNumber	a3
	markobtained	a4
	registrationID	r1
	questionID	q1
	examID	ex1

EVALUATION_STRATEGY (es)	eSID	es1
	assessmentTool	es2
	markDistribution	es3
	bloomsCategory	es4
	courseOutlineID	cu1
	assessmentType	es5
EVALUATION (e)	evaluationID	e1
	examID	ex1
	registrationID	r1
	totalmarksObtained	e2
CLO_MATRIX (cm)	clo_MatID	cm1
	clonum	cm2
	coDescription	cm3
	ploAssessed	cm4
	corelation	cm5
	courseOutlineID	cu1
	c	cm6
	p	cm7
	a	cm8
CO (co)	s	cm9
	coID	co1
	coNum	co2
	courseID	c1
	ploid	p1

PLO (pl)	ploID	pl1
	programID	p1
	ploNum	pl2
STUDENT_COURSE_PERFORMANCE (sp)	scpID	sp1
	registrationID	r1
	totalmarksObtained	sp2
BACKLOG_DATA (b)	backlogID	b1
	studentID	s1
	enrolled_course	r2
	enrolled_section	r3
	educational_semester	r4
	educational_year	r5
	timestamp	b2
	f_empID	f1

1NF:

1. Every entity has at least one primary key and
2. There is no repetition on rows.

<u>e1</u>	e2	e3	e4	e5	e6	e7	e8	<u>v1</u>	v2
-----------	----	----	----	----	----	----	----	-----------	----

v3	<u>n1</u>	n2	n3	<u>h1</u>	h2	h3	<u>f1</u>	f2	f3
----	-----------	----	----	-----------	----	----	-----------	----	----

<u>sc1</u>	sc2	<u>d1</u>	d2	<u>p1</u>	p2	<u>c1</u>	c2	c3	c4
------------	-----	-----------	----	-----------	----	-----------	----	----	----

<u>st1</u>	st2	<u>r1</u>	r2	r3	r4	r5	<u>s1</u>	s2	s3
------------	-----	-----------	----	----	----	----	-----------	----	----

s4	s5	s6	s7	s8	s9	s10	<u>cu1</u>	cu2	cu3
----	----	----	----	----	----	-----	------------	-----	-----

cu4	cu5	cu6	cu7	cu8	cu9	cu10	<u>l1</u>	l2	l3
-----	-----	-----	-----	-----	-----	------	-----------	----	----

l4	l5	l6	<u>ex1</u>	ex2	<u>q1</u>	q2	q3	q4	q5
----	----	----	------------	-----	-----------	----	----	----	----

q6	<u>a1</u>	a2	a3	a4	<u>es1</u>	es2	es3	es4	es5
----	-----------	----	----	----	------------	-----	-----	-----	-----

<u>e1</u>	e2	<u>cm1</u>	cm2	cm3	cm4	cm5	cm6	cm7	cm8
-----------	----	------------	-----	-----	-----	-----	-----	-----	-----

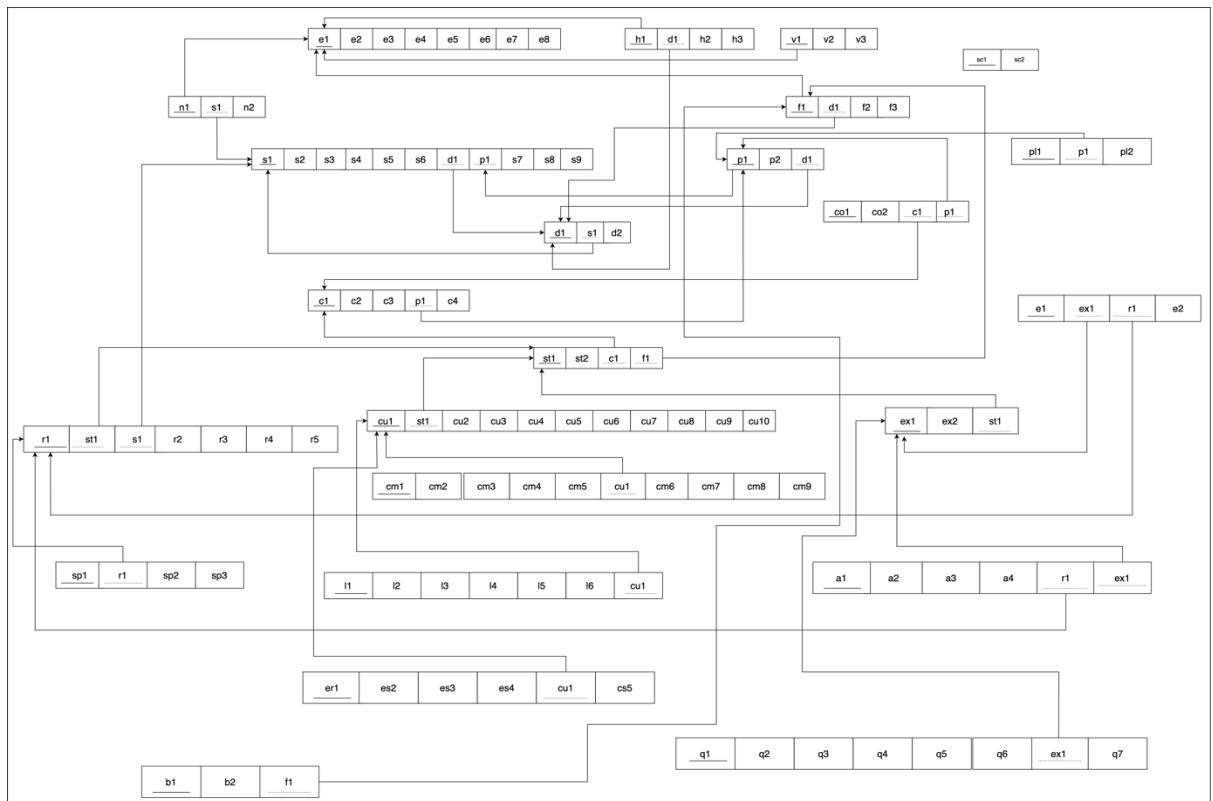
cm9	<u>co1</u>	co2	<u>pl1</u>	pl2	<u>sp1</u>	sp2	b1	b2
-----	------------	-----	------------	-----	------------	-----	----	----

2NF:

1. There is no partial dependency and
2. All relations are already in 2NF.

3NF:

- All relations are already in 3NF.



3.5 Data Dictionary

Employee_T

Name	Data Type	Size	Remark
nemployeeID	INTEGER	11	This is the primary key for Employee table. E.g.: "1801"
cpassword	VARCHAR	10	This is the password of the employee
cfirstname	VARCHAR	50	This is the first name of the faculty. E.g.: "Noor"
clastname	VARCHAR	50	This is the last name of the faculty. E.g.: "Sadman"
cemail	VARCHAR	30	This is the email of the employee. E.g.: "arnoyk123sets@iub.edu.bd."
cphone_number	VARCHAR	11	This is the phone number of the employee. E.g.: "01XXXXXXXX".
cgender	VARCHAR	6	This is the gender of the employee. E.g.: "Male".
cdesignation	VARCHAR	20	Employee hold an office or post. E.g.: "an employee can be a department head, dean, vc, faculty".
caddress	VARCHAR	50	This is the address of the employee. E.g: "House 12, Road 1, Block F, Bashundhara RA

VC_T

Name	Data Type	Size	Remark
nv_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4250"
dstartDate	DATE	DD MM YYYY	This is starting date for the VC. E.g.: "01-06-2020"
dendDate	DATE	DD MM YYYY	This is the date VC retire from his post. E.g.: "01-03-2023"

SCHOOL_T

Name	Data Type	Size	Remark
cschoolID	VARCHAR	5	This is the primary key of School. E.g: "SETS"
cschoolName	VARCHAR	50	This is the name of the school. E.g.: "School of Engineering, Technology & Science".

DEPARTMENT_T

Name	Data Type	Size	Remark
cdepartmentID	VARCHAR	3	This is the primary key for the Department table. E.g: "CSE"
cdepartmentName	VARCHAR	50	This is the name of the department. E.g: "Computer Science and Engineering".
cSchoolID	VARCHAR	5	This is a foreign key from the school table. E.g: "SETS".

DEPARTMENTHEAD_T

Name	Data Type	Size	Remark
nh_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4228"
cdepartmentID	VARCHAR	3	This is the departmentID of the department HEAD manages. E.g.: "CSE"

dstartdate	DATE	DD MM YYYY	This is starting date. E.g.: "01-07-2021"
denddate	DATE	DD MM YYYY	This is the date HEAD retire from his post. E.g.: "01-03-2024"

DEAN_T

Name	Data Type	Size	Remark
nd_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g: "4250"
cschoolID	VARCHAR	5	This is the SchoolID of the school DEAN manages. E.g: "SETS"
dstartdate	DATE	DD MM YYYY	This is starting date. E.g: "11-03-2020"
denddate	DATE	DD MM YYYY	This is the date DEAN retire from his post. E.g: "12-03-2024"

FACULTY_T

Name	Data Type	Size	Remark
nf_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4450"
cdepartmentID	VARCHAR	3	This is the DepartmentID of the department faculty belongs to. E.g.: "CSE"
crank	VARCHAR	30	This is the rank of the faculty. E.g: "Assistant Professor"
djoindate	DATE	DD MM YYYY	This is starting date. E.g: "01-03-2022"

PROGRAM_T

Name	Data Type	Size	Remark
nprogramID	INTEGER	11	This is the primary key for a program. E.g.: "1"
cprogramName	VARCHAR	50	This is the name of the program. E.g: "Bachelor of Science"

cdeptID	VARCHAR	3	This is the foreign key from the Department table. E.g: "CSE"
---------	---------	---	--

COURSE_T

Name	Data Type	Size	Remark
ccourseID	VARCHAR	6	This is the Primary Key for the Course. E.g: "CSE203"
ccourseName	VARCHAR	40	This is the name of the Course. E.g: "Database Management"
nnumofCredits	INTEGER	11	This is the number of credits for the Course. E.g: "3"
ccourseType	VARCHAR	10	This is the type of the Course. E.g: "Core"
nprogramID	INTEGER	11	This is the foreign key from the program table. E.g: "1"

PRE_REQ_COURSE_T

Name	Data Type	Size	Remark
npreReqID	INTEGER	11	This is the primary key for this table. E.g.: "1"
cpreReqCourseID	VARCHAR	6	This is the id of prereqcourse. E.g: " CSE101"
ccourseID	VARCHAR	6	This is the foreign key from the course table. E.g: "CSE203"

SECTION_T

Name	Data Type	Size	Remark
nsectionID	INTEGER	11	This is the Primary Key for Section. E.g: "1"
nsectionNum	INTEGER	11	This is the section number. E.g: "1"
nf_empID	INTEGER	11	This is the foreign key from Faculty table. E.g: "1801"

ccourseID	VARCHAR	6	This is the foreign key from the Course table. E.g: "CSE101"
-----------	---------	---	---

COURSE_OUTLINE_T

Name	Data Type	Size	Remark
ncourseOutlineID	INTEGER	11	This is the primary key for this table
nSectionID	INTEGER	11	This is the foreign key from the section table
cSemester	VARCHAR	30	This is the semester name. E.g.: spring 2021
nsectionNum	INTEGER	11	
mtcourseDescription	MEDIUMTEXT		This is the description of the course
mtobjective	MEDIUMTEXT		This is the objective of the course
mtcontent	MEDIUMTEXT		This is the content of the course
mtrefMaterials	MEDIUMTEXT		This is the reference material
ccourseTitle	VARCHAR	1000	This is the title of the course

cprerequisiteCode	VARCHAR	6	This is the prerequisite course code
ncreditValue	INTEGER	11	This is the credit value of the course

CLO_MATRIX_T

Name	Data Type	Size	Remark
nclo_MatID	INTEGER	11	This is the primary key for this table
nclonum	INTEGER	11	This is the clo number
mtcoDescription	MEDIUMTEXT		This is the co-description
cploAssessed	VARCHAR	10	This is the name of the plo assessed
ncorrelation	INTEGER	11	This is the correlation value or number
ncourseOutlineID	INTEGER	11	This is the foreign key from the course outline table
nc	INTEGER	11	This is the bloom's category level

np	INTEGER	11	This is the bloom's category level
na	INTEGER	11	This is the bloom's category level
ns	INTEGER	11	This is the bloom's category level

CO_T

Name	Data Type	Size	Remark
ncolID	INTEGER	11	This is the primary key for the CO table. E.g: "CO1".
ncoNum	INTEGER	11	This is the CO number. E.g: 1,2 etc.
ccourseID	VARCHAR	6	This is the foreign key from the Course table. E.g: "CSE303"
cploID	VARCHAR	5	This is the foreign key from the PLO table. E.g: "PLO1"

STUDENT_T

Name	Data Type	Size	Remark
nstudentID	INTEGER	11	This is the primary key for the Student table. E.g: "2022315".
cffirstname	VARCHAR	30	This is the first name of the student. E.g: "Imran".
clastname	VARCHAR	30	This is the last name of the student. E.g: "Hasan".
ddateOfBirth	DATE	DD MM YYYY	This is the birth date of the student. E.g: "21-12-2001".
cgender	VARCHAR	6	This is the gender of the student. E.g: "Male".
cemail	VARCHAR	30	This is the email of the student. E.g: "2022315@iub.edu.bd"
cphone	VARCHAR	11	This is the phone of the student. E.g: "01XXXXXXXXX".
caddress	VARCHAR	50	This is the address of the student. E.g: "House 12, Road 1, Block F, Bashundhara RA

cdepartmentID	VARCHAR	3	This is the foreign key from the Department table. E.g: "CSE"
nprogramID	INTEGER	11	This is the foreign key from the Program table. E.g: "1"
cenrollmentSemester	VARCHAR	6	This is the enrollment semester of the student. E.g: "Summer"
yenrollmentYear	YEAR	YYYY	This is enrollment year of the student. E.g: "2020"
cpassword	VARCHAR	10	This is the password of the student.

REGISTRATION_T

Name	Data Type	Size	Remark
nregistrationID	INTEGER	11	This is the Primary Key for Registration. E.g: "0101010101"
nsectionID	INTEGER	11	This is the foreign key from section table
nstudentID	INTEGER	11	This is the foreign key from student table

nEnrolled_course	INTEGER	11	This is the number of course a student can enroll
nEnrolled_section	INTEGER	11	This is the number of sections a student can enroll.
yEducationa_year	YEAR	YYYY	This is the educational year. E.g: 2022
cEducational_semester	VARCHAR	6	This is the educational semester of the student. E.g: "Summer"

EXAM_T

Name	Data Type	Size	Remark
nexamID	INTEGER	11	This is the primary key for this table
cexamName	VARCHAR	30	This is the name of the exam
nsectionID	INTEGER	11	This is the foreign key from section table

EVALUATION_T

Name	Data Type	Size	Remark
nevaluationID	INTEGER	11	This is the primary key for this table
nregistrationID	INTEGER	11	This is the foreign key from registration table
ntotalmarks	INTEGER	11	This is the total marks achieved by the student in a specific exam
nexamID	INTEGER	11	This is the foreign key from exam table

EVALUATION_STRATEGY_T

Name	Data Type	Size	Remark
neSID	INTEGER	11	This is the primary key for this table
ncourseOutlineID	INTEGER	11	This is the foreign key from course outline table

ASSESSMENT_TOOL_T

Name	Data Type	Size	Remark
nasID	INTEGER	11	This is the primary key for this table
neSID	INTEGER	11	This is the foreign key from evaluation strategy table

MARK_DISTRIBUTION_T

Name	Data Type	Size	Remark
nmdID	INTEGER	11	This is the primary key for this table
neSID	INTEGER	11	This is the foreign key from evaluation strategy table

BLOOMS_CATEGORY_T

Name	Data Type	Size	Remark
nbcID	INTEGER	11	This is the primary key for this table
neSID	INTEGER	11	This is the foreign key from evaluation strategy table

LESSON_PLAN_STRATEGY_T

Name	Data Type	Size	Remark
nlpSID	INTEGER	11	This is the primary key of the table
nweek	INTEGER	11	This is the week number
mttopic	MEDIUMTEXT		This is the topic name
mtlearningStrategy	MEDIUMTEXT		This is the lesson plan strategy of that topic
mtassessmentStrategy	MEDIUMTEXT		This is the assessment strategy of that topic
ncourseOutlineID	INTEGER	11	This is the foreign key from course outline table
ccorrespondingClo	VARCHAR	10	This is the corresponding clo For the corresponding course.

PLO_T

Name	Data Type	Size	Remark
nploid	INTEGER	11	This is the primary key for Program Learning Outcome. E.g: "PLO2"
nplonum	INTEGER	11	This is the PLO number. E.g: "2"
nprogramID	INTEGER	11	This is a foreign key from Program table. E.g: "2"

STUDENT_COURSE_PERFORMANCE_T

Name	Data Type	Size	Remark
nscpID	INTEGER	11	This is the primary key for this table
nregistrationID	INTEGER	11	This is the foreign key from registration table
ntotalmarksObtained	INTEGER	11	This is the total marks obtained by the student
fgradePoint	FLOAT		This is the grade point achieved by the student

QUESTION_T

Name	Data Type	Size	Remark
nquestionID	INTEGER	11	This is the primary key of this table
mtquestionDetails	MEDIUMTEXT		This is the question
nmarkPerQuestion	INTEGER	11	This is the mark each question contains
nquestionNum	INTEGER	11	This is the number of the question
ndifficultyLevel	INTEGER	11	This is the difficulty level of the question
nexamID	INTEGER	11	This is the foreign key from exam table
ncoNum	INTEGER	11	This is the CO number of the question
ccourseID	VARCHAR	6	This is the foreign Key for this table. E.g: "CSE203"

ANSWER_T

Name	Data Type	Size	Remark
nanswerID	INTEGER	11	This is the primary key for this table
mtanswerDetails	MEDIUMTEXT		This is the answer details
nanswerNumber	INTEGER	11	This is the number of the answer
nMarkObtained	INTEGER	11	This is the mark obtained by the student for each answer
nRegistrationID	INTEGER	11	This is the foreign key from registration table
nexamID	INTEGER	11	This is the foreign key from the exam table
nquestionID	INTEGER	11	This is the foreign Key from question table

BACKLOG_DATA_T

Name	Data Type	Size	Remark
nbacklogID	INTEGER	11	This is the Primary Key for this table. E.g: "11108297"
nstudentID	INTEGER	11	This is the ID of student. E.g: "2022315".
yEducationa_year	YEAR	YYYY	This is the educational year. E.g: 2022.
cEducational_semester	VARCHAR	6	This is the educational semester of the student. E.g: "Summer"
nEnrolled_course	INTEGER	11	This is the number of course a student can enroll
nEnrolled_section	INTEGER	11	This is the number of sections a student can enroll.
fobtained_percentage	FLOAT		This is the obtained Percentage for a student
dtimestamp	DATE	DD MM YYYY	This is the timestamp for backlog data. E.g.: "21-12-2022"
nempID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4450"

4

PHYSICAL SYSTEM DESIGN

4.1 Input Forms

The image shows a mobile application interface titled "Login Form". The main screen displays a "User Type" selection with two options: "USER TYPE" (selected), "Employee", and "Student". Below this are fields for "User ID" and "Password", and buttons for "Login" and "Sign Up". The background of the app shows a large, classical-style building under a sunset sky. At the bottom of the image is a code editor window showing the PHP code for "login.php".

```
login.php
1  <?php
2  require_once(__DIR__."/testfunction.php");
3  require_once(__DIR__."/connect.php");
4  session_start();
5  $invalid=0;
6  if($_SERVER['REQUEST_METHOD']=='POST'){
7
8      echo pre();
9
10     $userType=$_POST['userType'];
11     $ID=$_POST['ID'];
12     $password=$_POST['password'];
13
14     if($userType=='faculty'){
15         $sql="SELECT * from employee_t where employeeID='$ID' and password='$password'";
16         var_dump($_POST);
17         $result=mysqli_query($con,$sql);
18         if($result){
19             $num=mysqli_num_rows($result);
20             if($num>0){
21                 $invalid=0;
22                 $_SESSION['ID']=$ID;
23                 $_SESSION['userType']='faculty';
24                 header('location:employee_dashboard.php');
25             }
26         }
27     }elseif($userType=='student'){
28         var_dump($_POST);
29         $sql="SELECT * from student_t where studentID='$ID' and password='$password'";
30         $result=mysqli_query($con,$sql);
31         if($result){
32             $num=mysqli_num_rows($result);
33             if($num>0){
34                 $invalid=0;
35                 $_SESSION['ID']=$ID;
36                 $_SESSION['userType']='student';
37                 header('location:estudent_dashboard.php');
38             }
39         }
40     }else{
41         $invalid=1;
42     }
43 }
44 ?>
```

Student Dashboard

- Search...
- [Dashboard](#)
- [PLO Analysis](#)
- [PLO Achievement](#)
- [Spider Chart A](#)
- [Data Entry](#)
- [View Course Offerings](#)
- [Enrollment Status](#)
- [GPA Analysis](#)
- [Logout](#)

Light mode

Dashboard Sidebar

Dashboard

First Name: Md.Abdul Moin
studentID: 1531176
 gender: male
 Email: 1531176@iub.edu.bd
 Phone: 123456
 enrollmentSemester: spring
 enrollmentYear: 2021

[click to view](#)

estudent_dashboard.php

```

1
2
3  <?php
4  require_once(__DIR__."/testfunction.php");
5  require_once(__DIR__."/connect.php");
6  require_once(__DIR__."/user_header.php");
7  // $_SESSION['dashboard_referrer'] = 'estudent_dashboard.php';
8
9  // if (!empty($_SERVER['HTTP_REFERER'])) {
10 //   $_SESSION['dashboard_referrer'] = $_SERVER['HTTP_REFERER'];
11 // }
12
13 //echo pre($_SESSION);
14 if (isset($_POST['submit'])) {
15   $studentID = $_POST['studentID'];
16 } elseif (isset($_SESSION['ID'])) {
17   $studentID = $_SESSION['ID'];
18 }
19
20 // Fetch employee data
21 $student_data = null;
22 if (isset($studentID)) {
23   $query = "SELECT * FROM student_t WHERE studentID = ?";
24   $stmt = $con->prepare($query);
25   $stmt->bind_param("s", $studentID);
26   $stmt->execute();
27   $result = $stmt->get_result();
28   if ($result->num_rows > 0) {
29     $student_data = $result->fetch_assoc();
30   }
31 }
32
33
34 ?>
```

Faculty Dashboard

- Search...
- [Dashboard](#)
- [PLO Analysis](#)
- [PLO Achievement](#)
- [Spider Chart A](#)
- [Data Entry](#)
- [View Course Overview](#)
- [Enrollment Status](#)
- [GPA Analysis](#)
- [Logout](#)

Dark mode

Dashboard Sidebar

Dashboard

First Name: Mainuddin
 Last Name: Chowdhury
 Employee ID: 2259
 Gender:
 Email:
 Phone:
 Designation:

[click to view](#)

employee_dashboard.php

```

1
2
3 <?php
4 require_once(__DIR__."/testfunction.php");
5 require_once(__DIR__."/connect.php");
6 require_once(__DIR__."/user_header.php");
7 //echo pre($_SESSION);
8
9 if (isset($_POST['submit'])) {
10     $employeeID = $_POST['employeeID'];
11 } elseif (isset($_SESSION['ID'])) {
12     $employeeID = $_SESSION['ID'];
13 }
14
15 // Fetch employee data
16 $employee_data = null;
17 if (isset($employeeID)) {
18     $query = "SELECT * FROM employee_t WHERE employeeID = $ID?";
19     $stmt = $con->prepare($query);
20     $stmt->bind_param("s", $employeeID);
21     $stmt->execute();
22     $result = $stmt->get_result();
23     if ($result->num_rows > 0) {
24         $employee_data = $result->fetch_assoc();
25     }
26 }
27
28
29 ?>
```

```
🐘 connect.php
1  <?php
2
3  $HOSTNAME='localhost';
4  $USERNAME='root';
5  $PASSWORD=' ';
6  $DATABASE='spms';
7
8  $con=mysqli_connect($HOSTNAME,$USERNAME,$PASSWORD,$DATABASE);
9
10 <if(!$con){
11   |   die(mysqli_error($con));
12 }
13
14 ?>
```

```
🐘 logout.php
1  <?php
2  session_start();
3  session_unset();
4  session_destroy();
5  header('location:login.php');
6  ?>
7
```

4.2 Output Query and Reports

Dashboard [Dashboard](#) [PLO Analysis With Department/Program/School Average](#) [PLO Analysis \(Overall, CO Wise, Course Wise\)](#) [Upload CSV file](#) [Logout](#)

Dashboard

PLO ANALYSIS WITH DEPARTMENT AVERAGE **PLO ANALYSIS WITH PROGRAM AVERAGE** **PLO ANALYSIS WITH SCHOOL AVERAGE**

PLO Analysis with School Average

The chart displays the percentage of individual achievement versus school average for five specific PLO numbers. The Y-axis represents the Percentage from 0 to 100, and the X-axis lists the PLO numbers. The legend indicates that light grey bars represent 'Individual' and dark teal bars represent 'School Average'.

PLO Number	Individual (%)	School Average (%)
PL01	~85	~75
PL02	~100	~75
PL03	~100	~80
PL07	~100	~80
PL011	~100	~75

```
<?php
require_once( __DIR__ ."/testfunction.php");
require_once( __DIR__ ."/connect.php");
require_once( __DIR__ ."/user_header.php");

if (!empty($_SERVER['HTTP_REFERER'])) {
    || $_SESSION['dashboard_referrer'] = $_SERVER['HTTP_REFERER'];
}

// Set the dashboard link based on the previous page URL
if (isset($_SESSION['dashboard_referrer']) && strpos($_SESSION['dashboard_referrer'], 'estudent_dashboard.php') !== false) {
    $dashboardLink = 'estudent_dashboard.php';
} else {
    $dashboardLink = 'employee_dashboard.php';
}

echo "dashboardLink = " . $dashboardLink;

// current page
$dashboardLink = isset($_SESSION['dashboard_referrer']) && strpos($_SESSION['dashboard_referrer'], 'estudent_dashboard.php') !== false ?
'estudent_dashboard.php' : 'employee_dashboard.php';
echo '<a href=' . $dashboardLink . '>';

// Set the dashboard link based on the previous page URL
// if (isset($_SESSION['dashboard_referrer']) && strpos($_SESSION['dashboard_referrer'], 'estudent_dashboard.php') !== false) {
//     $dashboardLink = 'estudent_dashboard.php';
// } else {
//     $dashboardLink = 'employee_dashboard.php';
// }

//echo pre($_SESSION);
if (isset($_POST['submit'])) {
    $studentID = $_POST['studentID'];
} elseif (isset($_SESSION['ID'])) {
    $studentID = $_SESSION['ID'];
}

?>
```

Dashboard PLO Analysis With Department/Program/School Average PLO Analysis (Overall, CO Wise, Course Wise) Co analysis Logout

OVERALL PLO

CO WISE PLO

Overall PLO Analysis

PLO Number	PLO Percentage
PLO1	85
PLO2	100
PLO3	100
PLO7	100
PLO11	100

CO Achieved

CO	Achievement Level
CO1	0.9
CO2	0.6
CO3	0.5
CO4	0.4
CO5	0.3
CO6	0.2
CO7	0.1
CO8	0.1

```

spiderChart.php
184     <?php
185     if (isset($_POST['submit'])) {
186         $studentID = $_POST['studentID'];
187     } elseif (isset($_SESSION['ID'])) {
188         $studentID = $_SESSION['ID'];
189     }
190     ?>
191
192
193     <script>
194         function poView() {
195             <?php
196                 $sql = "SELECT po.poNum AS poNum,
197                     AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
198                     FROM registration_t AS r, answer_t AS ans, question_t AS q,
199                     co_t AS co, po_t AS po
200                     WHERE r.registrationID=ans.registrationID
201                     AND ans.examID=q.examID
202                     AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
203                     AND q.courseID=co.courseID AND co.poID=po.poID
204                     AND r.studentID='$_studentID'
205                     GROUP BY po.poNum";
206
207                 $result = mysqli_query($con, $sql);
208
209                 $po = array();
210                 $percent = array();
211
212                 while ($data = mysqli_fetch_array($result)) {
213
214                     array_push($po, "P0 " . $data['poNum']);
215                     array_push($percent, $data['percent']);
216                 }
217
218             ?>
219
220
221             var po = <?php echo json_encode($po); ?>;
222             var percent = <?php echo json_encode($percent); ?>;
223
224             for (var i = 0; i < percent.length; i++) {
225                 percent[i] = parseFloat(percent[i]);
226             }
227
228             document.getElementById("chart-container").innerHTML='';
229             document.getElementById("chart-container").innerHTML=<canvas style="background-color:white;height:500px;width:400px;" id="myChart"></canvas>;
230
231             const ctx = document.getElementById('myChart');
232
233             new Chart(ctx, {
234                 type: 'radar',
235                 data: {
236                     labels: po,
237                     datasets: [
238                         {
239                             label: 'P0 Achieved',
240                             data: percent,
241                             fill: true,
242                             backgroundColor: 'rgba(54, 162, 235, 0.2)',
243                             borderColor: 'rgb(54, 162, 235)',
244                             pointBackgroundColor: 'rgb(54, 162, 235)',
245                             pointBorderColor: '#ffff',
246                             pointHoverBackgroundColor: '#ffff',
247                             pointHoverBorderColor: 'rgb(54, 162, 235)'
248                         }
249                     ],
250                     options: {
251                         elements: {
252                             line: {
253                                 borderWidth: 3
254                             }
255                         }
256                     }
257                 }
258             });
259         }
260     
```

```

* spiderChart.php
260     function coView() {
261         <?php
262         $sql = "SELECT q.coNum,
263             AVG(ans.markObtained/q.markPerQuestion)*100) AS percent
264             FROM registration_t AS r, answer_t AS ans, question_t AS q,
265             co_t AS co, po_t AS po
266             WHERE r.registrationID=ans.registrationID
267             AND ans.examID=q.examID
268             AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
269             AND r.studentID='$studentID'
270             GROUP BY q.coNum";
271
272         $result = mysqli_query($con, $sql);
273
274         $co = array();
275         $percent = array();
276
277         while ($data = mysqli_fetch_array($result)) {
278
279             array_push($co, "CO " . $data['coNum']);
280             array_push($percent, $data['percent']);
281         }
282
283     ?>
284
285
286     var co = <?php echo json_encode($co); ?>;
287     var percent = <?php echo json_encode($percent); ?>;
288
289     for (var i = 0; i < percent.length; i++) {
290         percent[i] = parseFloat(percent[i]);
291     }
292     document.getElementById("chart-container").innerHTML="";
293     document.getElementById("chart-container").innerHTML=<canvas style="background-color:white;height:500px;width:400px;" id="myChart"></canvas>;
294     const ctx = document.getElementById('myChart');
295
296     new Chart(ctx, {
297         type: 'radar',
298         data: {
299             labels: co,
300             datasets: [{
301                 label: 'CO Achieved',
302                 data: percent,
303                 fill: true,
304                 backgroundColor: 'rgba(54, 162, 235, 0.2)',
305                 borderColor: 'rgb(54, 162, 235)',
306                 pointBackgroundColor: 'rgb(54, 162, 235)',
307                 pointBorderColor: '#fff',
308                 pointHoverBackgroundColor: '#fff',
309                 pointHoverBorderColor: 'rgb(54, 162, 235)'
310             }]
311         },
312         options: {
313             elements: {
314                 line: {
315                     borderWidth: 3
316                 }
317             }
318         }
319     });
320
321     </script>
322
323
324
325
326     </body>
327
328     </html>

```

Student Course Performance

studentID	sectionNum	semester	courseID	year	obtainGrade	co	co1	co2	co3	co4
1611001	2	summer	EEE131	2020	B+	80%	80%	80%	80%	80%
1711409	2	autumn	EEE131	2021	A-	85%	85%	85%	85%	85%
1910876	2	autumn	EEE131	2021	A-	85%	85%	85%	85%	85%
1720718	1	summer	EEE231	2020	A-	85%	85%	85%	85%	85%
1722021	1	autumn	ENG101	2021	A	90%	90%	90%	90%	90%
1810471	1	autumn	ENG101	2020	A-	85%	85%	85%	85%	85%
1811135	2	spring	EEE131	2021	B-	70%	70%	70%	70%	70%
1722021	2	spring	EEE131	2021	C	60%	60%	60%	60%	60%
1910876	2	spring	EEE131	2021	C-	55%	55%	55%	55%	55%
1821772	3	spring	ENG101	2020	A	90%	90%	90%	90%	90%
1822089	4	summer	ENG101	2020	B	75%	75%	75%	75%	75%
1910876	1	spring	CSC101	2021	A	90%	90%	90%	90%	90%
1931160	1	spring	CSC101	2021	A	90%	90%	90%	90%	90%
2020076	2	spring	MKT101	2020	A	90%	90%	90%	90%	90%
1711411	2	spring	MKT101	2020	C+	65%	65%	65%	65%	65%
1531176	1	summer	EEE131	2021	C+	65%	65%	65%	65%	65%
1910876	1	summer	EEE131	2021	B+	80%	80%	80%	80%	80%
1531176	1	summer	ENG101	2021	B-	70%	70%	70%	70%	70%

```

// Get status message
if(!empty($_GET['status'])){
    switch($_GET['status']){
        case 'success':
            $statusType = 'alert-success';
            $statusMsg = 'Data has been imported successfully.';
            break;
        case 'error':
            $statusType = 'alert-danger';
            $statusMsg = 'Some problem occurred, please try again.';
            break;
        case 'invalid_file':
            $statusType = 'alert-danger';
            $statusMsg = 'Please upload a valid CSV file.';
            break;
        default:
            $statusType = '';
            $statusMsg = '';
    }
}
1 reference
function gradeToPercentage($grade){
    switch($grade){
        case 'A': return mt_rand(90, 99);
        case 'A-': return mt_rand(85, 89);
        case 'B+': return mt_rand(80, 84);
        case 'B': return mt_rand(75, 79);
        case 'B-': return mt_rand(70, 74);
        case 'C+': return mt_rand(65, 69);
        case 'C': return mt_rand(60, 64);
        case 'C-': return mt_rand(55, 59);
        case 'D+': return mt_rand(50, 54);
        case 'D': return mt_rand(45, 49);
        default: return mt_rand(1, 44);
    }
}

```

Dashboard PLO Analysis (Overall, CO Wise, Course Wise) Logout

Student Course Performance

studentID:

Enter studentID

sectionNum:

Enter sectionNum

semester:

Enter semester

courseID:

Enter courseID

year:

Enter year

obtainGrade:

Enter obtainGrade

Submit

Student ID	Section Number	Semester	Course ID	Year	Obtain Grade	Co	Co1	Co2	Co3	Co4
------------	----------------	----------	-----------	------	--------------	----	-----	-----	-----	-----

```

1 <?php
2
3 // Include database connection and header files
4 require_once(__DIR__ . "/testfunction.php");
5 require_once(__DIR__ . "/connect.php");
6 require_once(__DIR__ . "/user_header.php");
7 $result = false;
8 $grade_mapping = [
9     "A" => 90,
0     "A-" => 85,
1     "B+" => 80,
2     "B" => 75,
3     "B-" => 70,
4     "C+" => 65,
5     "C" => 60,
6     "C-" => 55,
7     "D+" => 50,
8     "D" => 45,
9     "F" => 43,
0 ];
1 $grade_point_mapping = [
2     "A" => 4.0,
3     "A-" => 3.7,
4     "B+" => 3.3,
5     "B" => 3.0,
6     "B-" => 2.7,
7     "C+" => 2.3,
8     "C" => 2.0,
9     "C-" => 1.7,
0     "D+" => 1.3,
1     "D" => 1.0,
2     "F" => 0.0,
3 ];
4
5
6
7 if ($_SERVER["REQUEST_METHOD"] == "POST") {
8     if(isset($_POST["studentID"]) && $_POST["sectionNum"] && isset($_POST["semester"]) && isset($_POST["courseID"]) && isset($_POST["year"]) && isset

```

5

CONCLUSION

5.1 Problems and Solutions

- Analysis Phase:

One of the major problems faced was the confusion around the Rich Picture and Six Element Analysis of the organizational operations since there was no data available regarding those operations. However, Faculty members and other stakeholders were interviewed to overcome such confusion, and information received during the interview was collected to get a better understanding of the system that was being developed.

- Design Phase:

We face many serious problems during the design phase and ERD. Many systems and calculations were missing in the program. The front end didn't show what was expected. the dashboard was the same student and faculty. Somehow, we manage to overcome all those problems.

- Implementation Phase:

We try to implement all the new requirements and update all previous bugs. For that implementation, we used –

Front-End Developing tools: HTML, CSS, JavaScript, Google Charts, Chart JS

Back-End Developing tools: PHP, JSON

Database-integration: MySQL.

5.2 Additional features and future development

By integrating extracurricular activity monitoring into the SpmsV4 system, you could enable students to easily log and track their participation in various school activities, clubs, and events. This could include sports teams, academic clubs, music programs, volunteer work, and more. The system could then generate reports and analytics that highlight the student's level of participation, leadership roles, achievements, and areas for improvement.

From an educator's perspective, this data could be invaluable for gaining a holistic view of each student's engagement in school life. Teachers and administrators could use this information to identify students who may need additional support or encouragement or to recognize those who have demonstrated exceptional leadership and teamwork skills. This could ultimately help to foster a more inclusive and supportive school environment.

To implement this feature, you would need to consider how to securely collect and store extracurricular activity data, how to ensure privacy and consent from students, and how to present the data in an intuitive and meaningful way. You may also want to consider integrating the feature with existing systems, such as student profiles and attendance tracking, to provide a more complete picture of each student's performance and progress.

Reference:

1. <http://www.iub.edu.bd/AboutIUB/ataglance>
2. <http://www.cse.iub.edu.bd/degrees/1>
3. <https://www.youtube.com/watch?v=neUOwhzLa0w>

THE END