Database Management System Project

Final Report

Group - 04 (Wall Breaker)

Name	ID	Section
Md. Akram Hossain	1830707	04
Md. Adib Al Supto	1921719	04
Ishraque Rahman	1930426	04
Md. Asif Intesar	2022430	04
Md. Abdullah Al Hasan	2030467	04
Nahin Fatima	2030657	04

Contents

CHAPTER 1 -

CHAPTER 2 - REQUIREMENT

CHAPTER 3 - LOGICAL SYSTEM

CHAPTER 4 - PHYSICAL SYSTEM DESIGN:

CHAPTER 5

CHAPTER 1 - INTRODUCTION:

The Independent University, Bangladesh (IUB) has robust and versatile schools - notably consisting of the following:

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

The university has been an active participant in the growth of the education sector in Bangladesh and has produced capable and knowledgeable scholars contributing both here and abroad. [1] IUB has achieved this through working closely with relevant government education institutions and organizations such as the University Grants Commission (UGC),Ministry of Education, and other necessary institutes for each of the schools, regular updating its curriculums and putting in a system to monitor student performance based on a quantified approach between course curriculum and standards set by UGC and the Bangladesh government and constantly tracking student performance for every semester – mainly, using Outcome-Based Education (OBE) for monitoring performance and setting university curriculum. [1]

The focus of this report is to study the current student performance monitoring system that IUB uses, do the required analysis of its processes, and propose a new and better improved system that reduces error, makes analysis of data and report generation easier by all vested quarters and produce/show valuable information needed for IUB and its collaborators in making necessary improvements in academia to produce better scholars. The first part focuses on the details of the organization in question and the project that we have undertaken for it. The second part focuses on the existing system and its shortcomings and an introduction of the proposed system that we plan to replace the existing system with. The third and fourth will be heavily technical and focus on how we plan to bring the proposed system into being.During our research into the existing system for student performance monitoring we have found many areas where valuable changes could be made to make each process of monitoring student performance faster, make communication between necessary stakeholders easier, take away chances for errors and data duplication, and most importantly make it easier for all stakeholders to easily surf through large datasets to get meaningful information to their requirement. As we go through this report, we will dig deeper into how the current student performance monitoring system operates, the business processes involved, where there are concerns and issues related to data management, and how we can make a better system to address these issues for fixing and improvement.

A. BACKGROUND OF THE ORGANIZATION- IUB:

Independent University, Bangladesh (IUB), established in 1993, is one of the oldest private universities in Bangladesh, currently has more than an estimated 7,048 undergraduate and graduate students and over 10,455 alumni. This student population is mostly predicted to grow at 10% annually. [2]IUB, over-time, has shown remarkable outcomes in producing graduates with marketable skills only because of staying disciplined and up to date with the on-going curriculum and progress system. Dedicating attention towards IUB's Departments, and more specifically focusing the Department of Computer Science and Electrical science into a well-funded research hub running several research projects. IUB is also committed to curve potential graduates of international standard who are mainly equipped to provide new leadership to the national economy through skilled employment, entrepreneurship and/or applied research. This is successful due to the overwhelming support of the Bangladesh Government and the UGC for IUB to be able to create state-of-the-art lab facilities in their department. It is because of IUB's approach to academics as an "Application Oriented Learning" philosophy that "not only teaches students the fundamental principles of learning, situation -handling, and have better overall perception by providing them with hands-on training sessions." [3] Continuously growing it's lab facilities and flourishing on its curriculum according to current market economic demands, the SECS and the Department of Computer Science and Engineering at IUB has constantly worked with IEB, UGC and the Ministry of Education to track their students overall performance under specific periods by quantifying specific courses and its relating assessments into measurable trackers to gain valuable insights for improvement of students over

the years as a student in a certain department. These processes and criteria credentials courses are ultimately set by IEB along with relevant government potentials to set the bar for up-coming graduating engineers from top universities in Bangladesh. These sets of standards come in the form of Program Educational Objectives (PEO)and Program Learning Outcomes (PLO) [1] for specific departments in an Accreditation Manual which are mapped to specific courses by relevant Course Instructors and Co-Ordinator. This allows the Department of CSE at IUB, SECS, IEB and all other relevant stakeholders to have a calculating assessment of the current state-of-affairs and the performance of each student under each course for every semester. This will also allow users to track performance of faculties, courses, departments and schools and provides valuable insight for making necessary improvements.

B. BACKGROUND OF THE PROJECT SPMS 3.0:

Measuring the output of students, faculties, departments, and their respective courses in order to measure their productivity in regard to the outcome relevance of the course activities. Basically, to provide a range of tools and data intended to help universities and education authorities such as IEB, UGC, as well as other stakeholders to evaluate the performance of students and inform strategies for improvements. Developing a national framework for Outcome-Based Education while at the same time leaving considerable freedom to universities in implementing local approaches.

C. OBJECTIVE OF THE PROJECT SPMS 3.0:

The SPMS 3.0 system monitors and summarizes the performances of the stakeholders - students,faculties, schools, and departments through the database of the assessments. For evaluation purposes the system would be able to store individual assessment marks (midterm, quizzes, assignment, projects, presentations and so on). As well as the marks of those assessments with respect to their Course Outcomes (CO) and Program Learning Outcomes (PLO) accordingly in the database of the system to observe the outcome and performance of the student's faculties, schools, and departments.

The students being the primary stakeholder, would be able to statistically directly monitor the overall performance to their satisfaction of certain course objectives. Hence based on their performances and faculty evaluation the higher stakeholders (Head of department and Admin) can understand and manage the degree in comparison to which different course outcomes targets and their achievements are being understood by the student, department, school, and university body as a whole. SPSMS 2.0 also monitors the impact of policies against overall administrative goals and targets by the system. The system's main target is to monitor the whole university activities through the database and produce analytics for the Head of Department, Faculty, School, Students, and their Courses in a given period of time (yearly and semester wise).

D. SCOPE OF THE PROJECT:

We did a complete analysis of the existing system and found out places in the business processes which can cause severe lapses in time and communication, which we will discuss in the next chapter.

Our solution is to create a Web application, called SPMS 3.0 (Student Performance Monitoring System 3.0), using a Relational Database Management System (RDMS) to store, edit, add, and update necessary data for monitoring student performance and producing and storing related OBE data, reports, and documents. We produced potential users for the web based SPMS 3.0 system and speculated how they would be using the system and the necessary information and data they would need access to. Since the problems can arise from many points of all business processes, we will make custom user interfaces and login capabilities for all stakeholders who will also be the users of this system. Since we use a (RDBMS) for data storage, retrieving necessary files, tabular data, page layouts and reports becomes incredibly easy and allows us to interact with the necessary data to occur real-time. We also create interfaces for all users to easily access these data and use them to generate and download reports. We build an interface for faculties to be able to collaborate with each other on developing course outlines, course reports, marksheets, assessments, mapping assessments to CO's and PLOs for PLO achievements, and record assessments of students throughout the semester for all their courses.

Students, the IUB leadership team and government agencies can also access the systems for drawing conclusions. Data will also be protected, and each stakeholder will be shown only that data, which is relevant to them, respectively.

CHAPTER 2 - REQUIREMENT ANALYSIS:

The Requirement Analysis is the process of researching and visualizing the current system and processes that go into the business operation of a specific organization using industry tools, methods, and standards. "The process of determining what the database will be used for is known as requirements analysis." It entails conducting interviews with user groups and other stakeholders to determine what functionality the database needs to provide, what types of data they want to process, and the most frequently performed operations." [4]
This allows us to observe each stakeholder and how they interact with one another. We use simple notations and symbols to explain how a business process works and how to dissect it.
As we shall see, this approach of analysis enables us to identify both obvious and less obvious issues with a current manual system of student performance monitoring that relies on the involvement of stakeholders and third party actors producing faults in the system.

A. RICH PICTURE – EXISTING BUSINESS SYSTEM:

A Rich Picture is a method for investigating, acknowledging, and defining a business process and then expressing it using diagrams to produce a rough mental model. A detailed description facilitates conversation and leads to a comprehensive knowledge that is shared by all parties. [5] The comprehensive image that is produced can help other stakeholders understand the issues with a current system while also allowing them to take into account a wide range of relevant factors. Rich images focus on the processes and structure of a particular setting. [6]

The Rich Picture Analysis also takes into account the following:

- **Structures**
- **Processes**
- **Climate**
- **People**
- **Issues expressed by people.**
- **Conflict**

As we can see, these factors were specifically taken into consideration when creating this rich image.

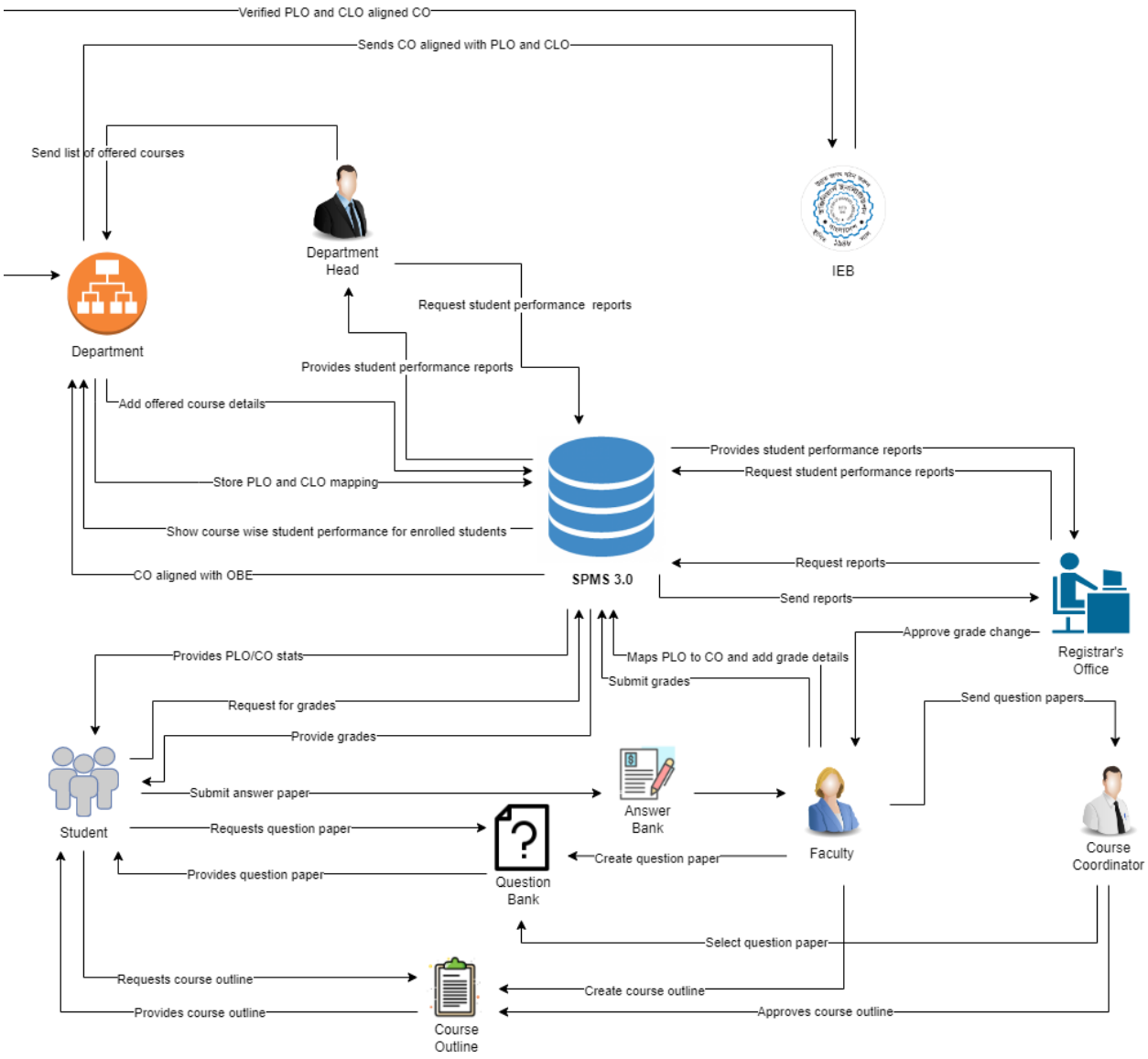


Figure 1.0: Rich Picture of Existing System to Monitor SPMS.

According to the Rich Picture Analysis, we have the following categories of stakeholders:

1. IEB
2. Head of Department/Dean of School
3. Department (working under Head of Department/Dean of School)
4. Faculty
5. Course Coordinators
6. Registrar's Office
7. Admin (working under Registrar's Office)
8. Students

We can also distinguish three different storage facilities or systems, namely:

1. The Department Storage
2. The Registrar's Office Storage
3. SPMS

We have identified seven processes from this "Rich Picture" that are essential to tracking student progress and enhancing the curriculum. These are the procedures:

1. Map Course Outcomes (COs) to Program Learning Outcomes (PLOs).
2. Record Student Course Performance Data.
3. View Course Reports over a given time-period for inspection and analysis of student performance trends.
4. Produce OBE Marksheet & Bloom's Taxonomy Report.
5. View Records OBE Marksheets, Course .
6. Request for Question Bank files.
7. Request for Course Outlines.

B. SIX ELEMENTS ANALYSIS - EXISTING BUSINESS SYSTEM:

The Six Elements Analysis gives a thorough explanation of each element's function in each process. The table below shows that Human entities predominate in all important system functions, particularly in the two processes that are most important—mapping course outcomes and viewing documents related to them. For instance, the current system is heavily reliant on manually processed and handled hardcopy databases. As a result, there is a considerable amount of waiting involved in the interdependent processes before the Human components may perform their obligations.

Process	System Roles					
	Human	**Non_computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Network & Communication**
Map Course Outcomes (COs) to Program Learning Outcomes (PLOs).	**IEB:** 1. Send verified PLO and CLO aligned CO to the Department. **Department:** 1. Received CO aligned with OBE. 2. Store PLO and CLO mapping. 3. Sends CO aligned with PLO and CLO to IEB. **Faculty:** 1. List the course material. 2. List the COs. 3. Connect course	**Pen and Paper:** 1. Is utilized for recording more advanced problem-solving ideas. **Board and Marker:** 1. Is utilized for recording more advanced problem-solving ideas.	**Computer:** 1. Course Coordinators create softcopies of the Course Outcomes (COs) of the particular courses in which they excel using computers. **Printer:** 1. To print out physical copies of Course Outcomes (COs).	**MS Word:** 1. Course coordinators create detailed course outlines in MS Word and course evaluation reports that map course outcomes (COs) to program learning outcomes (PLOs). **Excel Sheet:** 1. The course makes use of an Excel sheet. Coordinators will link particular midterm, final exam, and project works to particular course outcomes.		**Internal and Email:** 1. To connect with IEB or other stakeholders about crucial issues pertaining to the mapping of course outcomes to program learning outcomes, utilize the internet and email. **Others:** Use telephones or other physical contact to have essential conversations with stakeholders about the mapping course. Outcomes to Program Learning Outcomes.

	objectives to course content (COs). 4. Maps PLO to CO. 5. Map COs to specific project work, midterm, and final exam questions. 6. Using the course outline, course content, and COs, begin creating course assessment reports.					

Record Student Course Performance Data.	**Faculty:** 1. Assign assignments and project work. 2. Complete tests and quizzes all semester long. 3. Keep track of each student's evaluation data for each assessment (tests, assignments, projects, and exams) throughout the semester on both digital and paper copies. 4. Keep track of the scores for every particular question on the midterm and final exams. 5. Determin	**Pen and Paper:** 1. Record evaluation information and marks in tabular format using a pen and paper (hardcopies).	**Computer:** 1. Computers are used to create softcopies of all assessment data records for certain courses.	**Excel Sheet:** 1. Fill out Excel sheets with the appropriate assessment information and final grades. **SPMS:** 1. Upload students' final grades to SPMS so that they can view them or have them viewed by the registrar's office.	**Department Storage:** 1. The department office and registrar's office may save copies of student assessment data and final grades for future reference. **SPMS Database:** 1. Information on student grades is kept and maintained by SPMS using a database server.	**Internet:** 1. To communicate with IRAS and store student final grades, one uses the Internet.

	e the combined scores for all assignments, tests, midterms, and finals, then assign final grades to each student in a particular course. 6. Convert midterm and final marks. 7. Enter all of a student's grades for a course into a marksheet. 8. Grade the student. 9. Enter final grades for students on SPMS. 10. Send the Department the Marksheet. 11.					

	Deliver the Markshee t to the Office of the Registrar.					

View Course Reports over a given time-period for inspection and analysis of student performance trends.	**Faculty:** **a)**Take Assessments for specific courses. **b)**Send Assessment report to the Head of the Department. **c)**Announce a date for checking students' performance on the Assessments. **Students:** **a)** Check their performance in the given Assignments **b)** If There is any Query then inform faculty. **Head of the Department:** **a)** Check students'	**Pen And Paper:** **a)** Write Down the record of the course report. **Room:** **a)** if there is any query in the performance report then come to a specific room to check.	**Computer: a)** Computer used to record the performance report. b) Students may sign in to Google Classroom. **Printer:** **a)** may be printed a copy sent to the Department Head And CITS. **Google ClassRoom:** Faculty may use Google Classroom to Announce the date of check report. Mobile	**MS Word:** **a)**Record the Analysis report in Word. **Excel Sheet:** Mark All Individual report , Overall report And make a Spider web of Total Analysis report.	**Department Storage:** Department Storage is being used to record the analysis for comparison to previous semesters.	**Internet:** Faculty use the internet to upload notices for check reports in google classroom.

	Performance reports for the course. **b)** Send a Performance report to the **CITS**. **CITS:** **a)**Preserve the performance report in the database.					
Produce OBE Marksheet & Bloom's Taxonomy Report.	**Faculty:** 1. Calculate total marks received for each CO by calculating the marks received for questions and/or other Assessments mapped to CO's.	**Pen and Paper** 1. OBE marksheet Stored in hardcopy. Additional markings may be made to further separate between students.	**Computer/ Phone:** 1. Uses computers to make softcopies of the OBE Marksheet and Course Assessment Reports. **Printer:** 1. Print hardcopies of final	**Coded Excel sheet:** 1.Faculty /Course Coordinator uses automated excel sheets to calculate the student's success/ failure in Achieving PLOs. **MS Word:** 1. Used to make Bloom'	**Department Storage:** 1. Records of students' assessment data and final grades will be saved in the department for future reference.	**Internet/Mail: 1.** An Online platform (such as Google Sheets) may be used for processing the OBE assessment data spreadsheet and Bloom's taxonomy datasheet.

	2. Calculate total percentages received for each COs on the OBE Marksheet. 3. Declare if a student has achieved a specific CO (if CO percentage is greater than or equal to 40). 4. Declare if a student has received a PLO for a related CO. 5. Make a table giving		versions of the OBE Marksheets and Course Assessment Reports.	s taxonomy Report softcopies.	**Registrar's Office Storage:** 1. OBE Marksheets, Course Assessment Reports and other documents submitted by the department is stored for future reference.	

	the verdict and analysis of how many students were able to receive a certain CO and PLO and other documents containing necessary information and data. 6. Design Course Assessment Report using Course Outline, Course Content and					

	Course Outcomes. 7. Send the final version of the OBE Marksheet to the Dept. Office. **Department Office:** 1. Send the OBE marksheet, Course Assessment Report and others to the Registrar's Office. 2. Store the OBE Marksheet and Course Assessment					

	Repor t in the depart ment. **Registra r's Office:** 1. Stores the OBE Marksh eet and Course Assessm ent Reports and other docume nts and reports in the Registr ar's Office.					
View Records OBE Marksheet s, Course.	**Faculty: a)**Make an individual OBE mark sheet for Students. **b)**Reques t to CITS to Upload in SPMS. **Student:**	**Pen and paper:** Faculty use to note specific students and also mark for the out performers within the Students.	**Compute r: a)**Student s send request through Computer Using his ID. **Mobile:** Students send requests through	**MS Word:** Faculty Record report in MSW Word. **Excel Sheet:** For individual performanc e and mapping ousting performers.	**Departm ent Storage:** Departme nt Storage is being used to record the analysis for compariso n to previous semesters.	**Internet:** To use a Web Browser for Sending request Internet is must needed.

	a) Login to IRAS**.** **b)**Request IRASH to get the report of OBE markshee t. c) get markshee t **SPMS:** a)SPMS got a request to upload an OBE markshee t. b) Upload Markshee t c) Got a request from a student to see their markshee t report. c)Upload Specific markshee t to the student.		Mobile using his ID. **Printer:** Faculty may print out the whole performa nce Coy and discuss in the Classroo m	**Web Browser:** Students or faculty both have to go through the web browser to send requests to SPMS. **PDF Viewer:** when Students send requests to see their marks sheet, The CITS send a PDF copy of mark Sheet.	**SPMSDa tabase:** SPMSpre serve all the Data to the Database so that anytime any student can ask to see his mark Sheet.	
Request for Question Bank files.	**Faculty:** 1. prepare question papers. 2. Give away the question paper to the course	**Pen & Paper:** Students Submit their Previous year question.	**Compute r:** If Exam held on Online then the faculty ,Student both use Computer	**Web Browser:** Students or faculty both have to go through the web browser to attain an	**Google Drive:** Exam Question will be stored in Google Drive of the Classroo	**Internet:** To have Access in Google Classroom Student and teacher Both need internet.

	coordinator for selection. 3. Take exams from returned Question papers. 4. Got Script from Students 5. Check exam scripts of students. **Course Coordinators:** 1. Receive question papers from faculty. 2. Moderate question paper 3.Send selected question paper for exam. **Students:** 1. Ask for exam's question papers. 2. Perform exam		**Mobile:** **a)** Students may use Mobile to Attaining in the Exam. b) if the exam is physical then after exam Faculty take a Snap the Questions and upload in the Google classroom for making a Question bank. **Printer:** If Exam held on Physically then faculty print out hard Copy of the Question.	Exam. **PDF Viewer:** If it is in online Then the faculty upload a pdf file in the classroom and students have to access it in the PDF file.	m.	

	3.Submit exam's script to faculty.					
Request for Course Outlines.	**Faculty:** 1. Generate course outline. 2. Send the course outline to the course coordinator. **Course Coordinators:** 1. Authorize course outline. **Students:** 1. Seek for course outline. 2. Receive course outline.	**Pen & Paper:** **a)** Faculty Send a hard copy of outlines to the Course coordinator. **b)**Students receive a hard copy of the Approved course outline from the faculty.	**Computer:** Students may use computers to get their course outlines in the Google Classroom. **Printer:** Faculty may give Student outline hardcopies.	**Web Browser:** Students or faculty both have to go through the web browser to Upload the course outline as PDF in online Google Classroom. **PDF Viewer:** If it is in online Then the faculty upload a pdf file in the classroom and students have to access it in the PDF file.	**Google Drive:** Outlines will be stored in Google Drive of the Classroom.	**Internet:** To have Access in Google Classroom Student and teacher Both need internet.

C. PROCESS MODEL – EXISTING BUSINESS SYSTEM:

A business process model can specify business processes using the Business Process Model and Notation (BPMN) in a graphical format. [7] To break down each of the business processes outlined in the preceding part, we use diagrams from business process models.
The participants in the processes, their interactions, and the decisions that each of them must make are broken down into different diagrams.

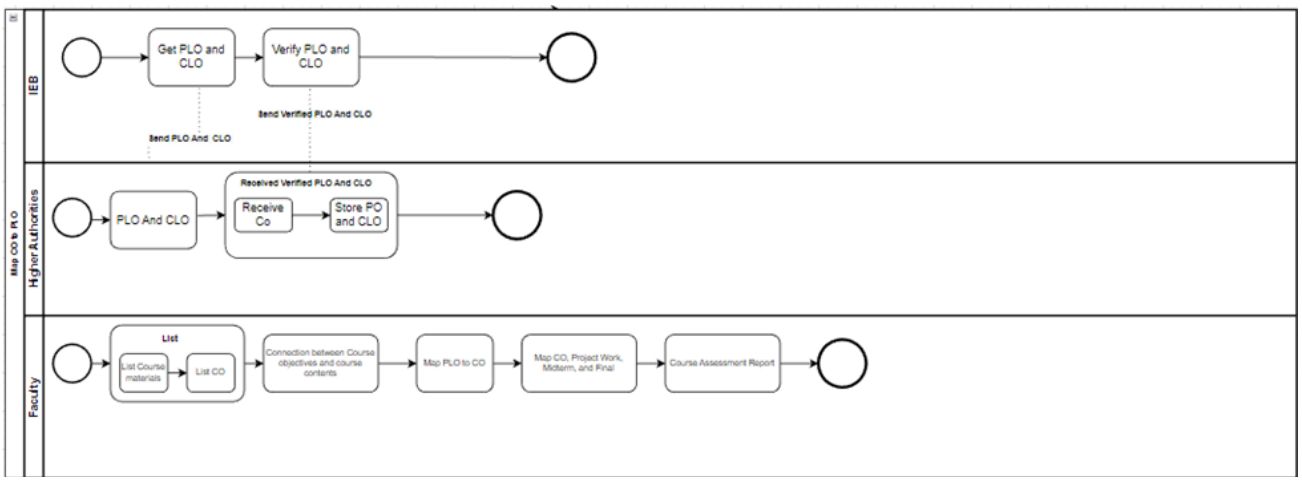


Fig 1.1: Map CO To PLO

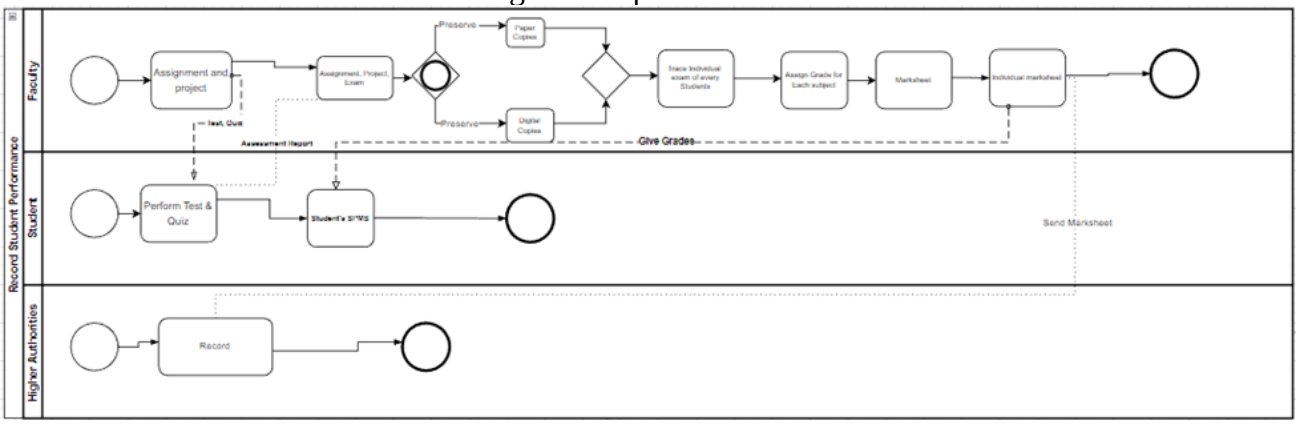


Fig 1.2: Record Student Performance

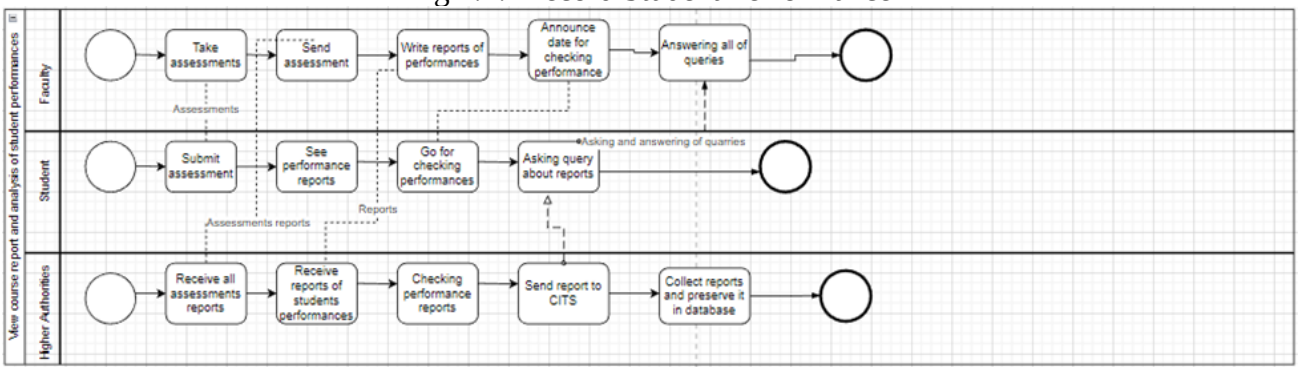


Fig 1.3: View course report and analysis of student performances

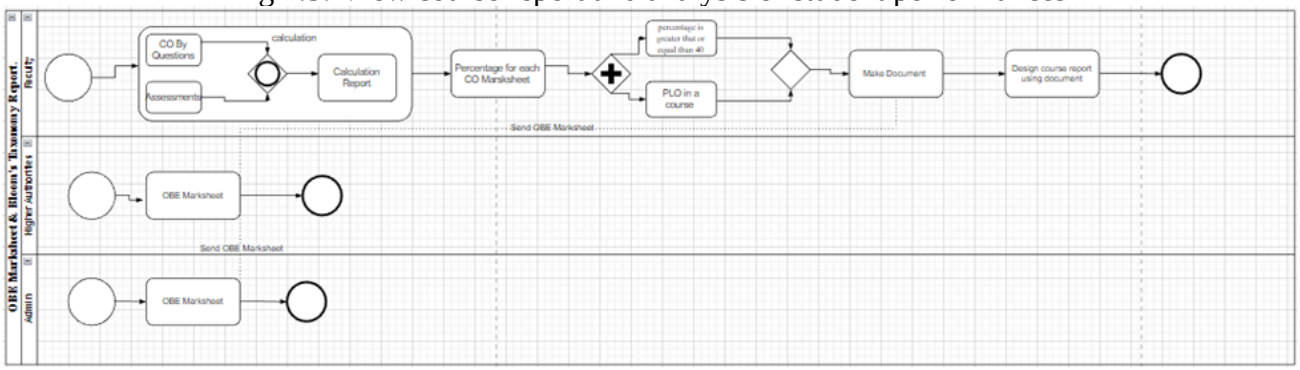


fig 1.4: Produce OBE Marksheet & Bloom's Taxonomy Report.

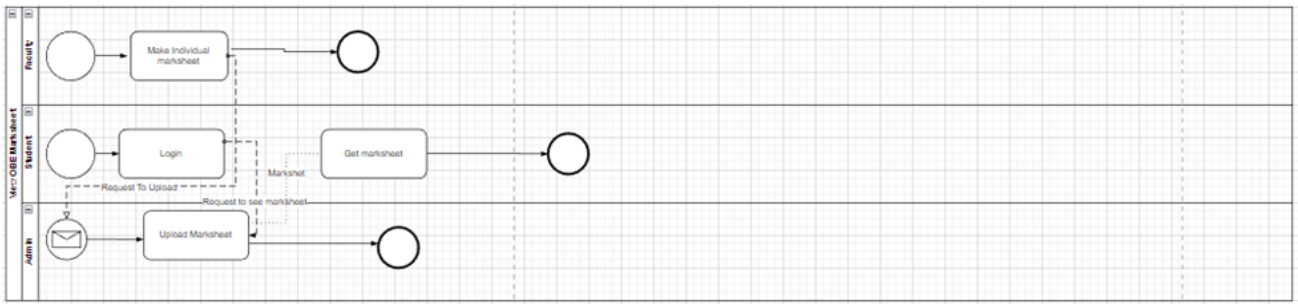


Fig 1.5: View Records OBE Marksheets Course.

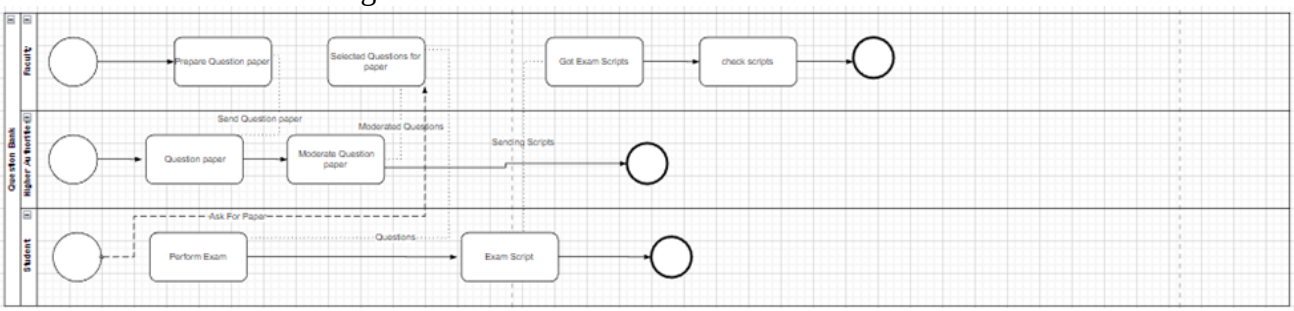


Fig 1.6: Request for Question Bank.

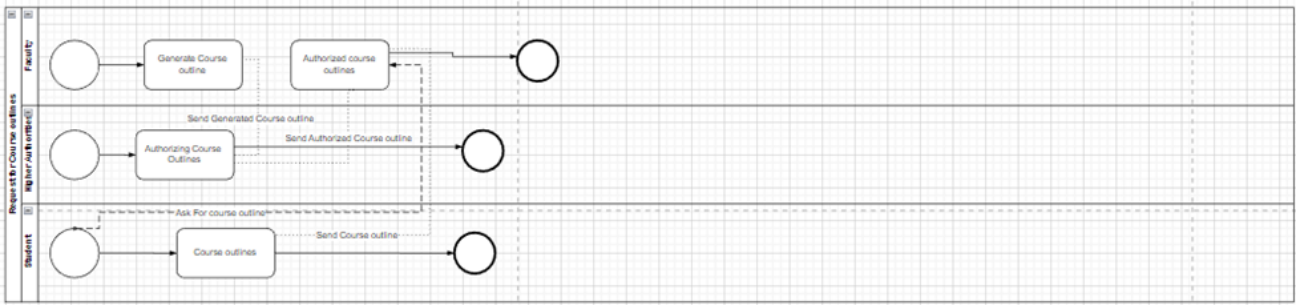


Fig 1.7: Request for Course Outline.

D. PROBLEM ANALYSIS – EXISTING BUSINESS SYSTEM:

The shortcomings in each process were determined using a Six Elements Analysis of the existing systems. The far-right column of this table displays a recurring pattern. The creation of a private online platform seems to have numerous positive effects on the system.

Process Name	Stakeholder	Concerns(Problem)	Analysis(Reason of the Problem)	Proposed Solution

Update the PLO's initial mapping	1. Department	The department has to manually map each course under the present system using paper and pen. Therefore, the department must manually prepare the mapping again if updates are necessary. This is a serious problem.	The department must create the mapping manually under the current system, which takes time, adds more labor, and does not make the most use of available resources.	Therefore, the following are the methods to address this issue: 1. A weight or level will be given to each PLO and course. 2. The PLO and Courses are mapped based on this weight/level. The resulting matrix would be used to perform the initial mapping. 3. Depending on how many courses and courses that have PLO, the admin may quickly adjust the mapping.
Reviewing scripts and creating grade sheets.	1. Student 2. Faculty 3. Department	1. Answer scripts are manually reviewed and marked by specific faculty members. 2. The grade sheets' marks must be manually calculated, tabulated, and graded by the members.	Given the number of students, manually reviewing answer scripts takes a lot of time, and then there's the manual creation of grade sheets. The likelihood of error while checking the scripts rises as a result.	These problems can be resolved through automation: 1. The system will automatically check scripts and provide the mark sheet for exams with multiple-choice questions. If the exam is in quiz format, for example, our system shows the MCQ marks and also provides the answer scripts.

				2. For tests of this nature, faculty members must hand check, mark, and grade the answer sheets. However, the system creates the grade sheet after the marks are submitted. The system will provide the grades and marks to the students.
Automated mapping to prepare for a particular course preparation for course assessment.	1. Faculty	It takes time and won't always be consistent for faculty to base their course preparation on the prior PLO or first mapping of the PLO. They have to keep track of how many PLOs they are mapping when mapping PLO and CO, which can be difficult and cause other issues.	The faculty must manually construct the mappings under the current system, which increases the likelihood of mistakes and issues when they map PLO and CO.	Our system will have predefined PLO labels and course labels to address this issue. If the faculty is happy with the suggested number of CO and assessments, the system will generate a table showing how the CO and assessments are mapped out for them. (They may update the mappings if they are not happy.)
When creating the test, make suggestions for questions from the question bank	1. Faculty members.	Because the faculty must constantly manually map the COs to the questions and construct the question paper, designing	Because it is not possible to effectively recycle all known prior question papers, the question papers are thrown away after an exam.	All verified question papers will be saved in our system as soft copies that the faculty can access when creating new tests. For instance, it will

		questions takes more time and effort.	The professors don't have a binder with former exam papers or an exam history.	be advised to look through the exam history of the midterm papers for that course if a faculty is about to create a midterm question paper. In this manner, the time and effort needed to design a paper are both decreased.
Generate Continuous Quality Improvement report	1. Faculty Members 2. SPMS 2.0 Storage	1. The SPMS is used to gather the progress report. 2. Faculty members must find the lowest percentage for each PLO a student has for all PLOs after personally checking the proportion. 3. Reports must be created from the data.	These activities take time. First, it takes time for the report to be transferred from SPMS to the faculty. Additionally, faculty members are more likely to make mistakes while determining the lowest proportion. The instructor may unintentionally enter some incorrect data when assembling the information. Furthermore, since each semester will be subject to change depending on the students' performance, it is needlessly difficult to pinpoint issues	A Continuous Quality Improvement report with the following information will be produced in our system: 1. A graph showing the number of students enrolled in each department over a given time frame/number of semesters. 2. A course-wise student performance trend for a given time period/semester based on GPA. 3. Student performance trends for a certain time period/semester based on instructors, using

			and find solutions.	GPA. 4. Trend in student performance for a selected subject, according to the instructor, over a given time frame/semester. 5. The lowest percentage of each PLO for each student and the PLO percentage relating to the particular course. Describe potential fixes or ideas for enhancing the kids' performance. 6. A comparison of the proportion of PLO attempts with the percentage of PLOs that were successful
Check question difficulties level	1. Faculty 2. Department 3. Student	With the current system, the department has no scope to check the difficulty level of the exam questions. As a faculty is preparing the course planning, they must know the ques level when they are making it.	In the current system, the department has to come up with a scope to check the difficulty level of the questions. But there is no implemented scope to check this in the SPMS system.	As such, these are the ways to combat this problem: 1. Bloom's Taxonomy feature has to be implemented here. 2. The faculty member must check the question difficulty level while they were

				making the questions. 3. Also graph displaying the level of the questions by following the keywords of the questions.

E. RICH PICTURE - PROPOSED SYSTEM:

Based on the issues and issues we discovered throughout the problem analysis, we would use several user interfaces created for particular user needs. The report of a student may be viewed by the department head, dean of the school, course instructor, coordinator, faculty, administrative assistant, student, IEB, UGC, ministry of education, vice chancellor, board of trustees, and department staff, among other state parties.

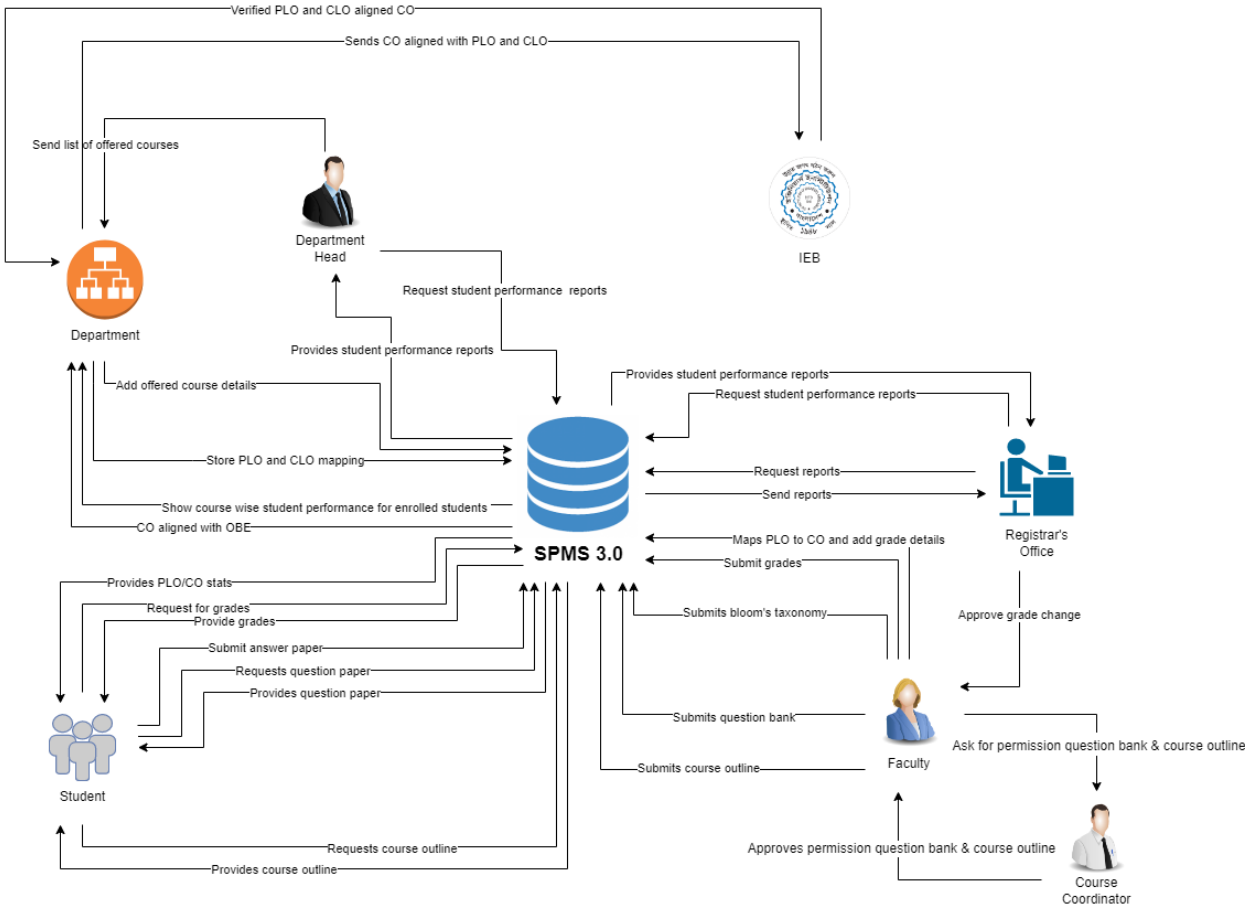


Figure 1.2: Rich Picture of Proposed System to Monitor Student Performance.

F. SIX ELEMENTS ANALYSIS – PROPOSED SYSTEM :

The new system, an online platform called SPMS, where it will have its own database that host the data of all the courses, faculties, as well as updated tables every semester to keep track of which courses have been assigned to which faculties in a given semester, will make the Course Outcomes (COs) and Program Learning Outcomes (PLOs) visible. We are developing the new system in order to track student performance as well as faculty members who are instructing a particular course or the performance of students over time in a course.In a nutshell, we can observe that the SPMS relational database (a non-human) quite literally plays a vital role in the student performance monitoring system. Additionally, compared to other other processes, this one has the most connections.
The suggested system's six-element analysis is the next step in a sequence of analysis where each analysis builds on the one before it. The table below provides additional insight into the function of each component of the new system based on the detailed picture.

Process	**System Roles**					
	Human	**Non_computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Network & Communication**

Review and update the Initial Mapping of PLO's (Course-wise)	**Department:** 1. Department must enter the system's website to map PLO and CLO aligned with CO. 2. Click on mapping to map and then a new table is generated and shown. 3. The tables contain the lists of all courses and PLOs. It also shows how a PLO can be mapped to various courses as well as how many PLOs can be mapped with a certain course. 4. The PLOs and courses will have labels. The PLOS will be shown based on the level that will help the department to identify PLO mapping. 5. Then the department can store it in our system database through a click.	**Pen & Paper:** PLOs and the courses are mapped by using pen and paper.	**Computer:** Computer is used for entering our website and updating the PLO and course mapping. Moreover, the course outline and course details can be viewed.	**SPMS 3.0:** SPMS 3.0 is required to update PLO and CO mapping. **Operating system:** Any operating system can be used by users, Windows, Mac, Linux etc.	**SPMS 3.0 Database (Unsure):** The mappings of PLOs and COs are stored here.	**Internet:** It is an online website. It is required to upload the PLO and course planning.

Course Progress	**Faculty:** 1. The faculty logs in successfully using ID and password. 2. Goes to the question paper creation section. 3. Designates an exam type as well as dictates the total marks. 4. Sets question numbers.	**Pen & Paper:** Some questions may have to be answered in a paper and scanned for upload. Rough work may be done.	**Computer/ Laptop/ Smartphone:** Both the students and the faculty need devices to conduct the examination successfully.	**Internet Browser:** Suitable internet browsers for website navigation such as Google Chrome, Mozilla Firefox, Safari etc.	**SPMS 3.0 Database:** For storing the information of the faculty and the student users and also question papers, data.	**Internet:** Used for accessing the SPMS 3.0 software and database by both students and faculty members.

Make grade sheets after checking Answer Script	**Faculty:** 1. The faculty has to input the answers into the system and check script as well as mark accordingly. For example, the website will show the marks and answers for the MCQs in a quiz. 2. The system composes mark sheets in excel files which the faculty collects. **Student:** 1. Students can look over their marks and answer scripts directly on the website. 2. They can detect their grade on the website.	**Paper:** When the faculty has to print the mark sheets and grade sheets, he/she uses paper.	**Computer/ Laptop:** In the purpose of logging into the website for checking the marks and grades by both the students and faculty members. **Printer:** For printing the necessary documents such as mark sheets, grade sheets etc.	**SPMS 3.0:** Requisite for examining answer scripts, marks and grade sheets.	**SPMS 3.0 Database:** Requisite for storing answer scripts, marks and grade sheets.	**Internet:** To access the SPMS 3.0 software and database, it is used by both the students and faculty members.
Mapping a specific course automatically and prepare the course assessment Planning	**Faculty:** 1. The initial mapping of PLO and course is already done by the department. Faculty members have to enter the website first and then log in with their IDs.	**Stationary:** Paper is used for printing the necessary instructions for the course outline and assessments planning as CO and PLO based details.	**Computer:** For logging into our website and adopting the PLO and CO mapping assessment planning and mapping. By using a computer or	**SPMS 3.0:** SPMS 3.0 is needed for updating the PLO and CO mapping, assessment and course outcome mapping etc.	**SPMS 3.0 Database:** The mappings of PLOs and COs are reserved here.	**Internet:** The internet is required to update the PLO and CO mapping and also the assessment planning.

	2. They can view their assigned course and the PLOs for that course that are suggested by the department. 3. If a report is available, the faculty can view the CQI report. If it is necessary then the faculty members can also update the PLO mapping. 4 System will provide the faculty with a few suggestions. For example, the number of course outcomes and PLO with the label wise courses, the specific PLOs, the number of assessments etc. If the faculty members want to change something then they have to select the number of CO and map that CO with PLO. 5. Designs a specific question in an alike pattern by browsing	Necessary tools such as calculator, ruler, pencil, eraser etc. are needed for solving all questions and writing them.	laptop, the course CQI report and course can also be viewed. If there is something to be changed based on the COI report then the changes will be made by using a computer.	**Operating System:** The user may use any OS such as Windows, Mac, Linux etc.		

	through the previous papers of the almost identical types of exam. 6. Set out the marking for that question. 7. If it is needed then make another question by using the same procedures 4-6. 8. Clicks on the "Save" option and successfully saves the paper as usual. 9. In addition, include further information regarding the exam. For instance, the duration topics etc. **Student:** 1. The student logs into the website by successfully using ID and password. 2. Clicks on the exam section for the exam history and the					

	upcoming exam announcements for all courses the student is enrolled in during the registration of that ongoing semester. 3. They can learn about details of other exams and the syllabus by clicking upcoming exams options.					
Update student enrollment information in SPMS	**Registrar Office:** 1. SPMS 3.0 gives the notice of updating the student information to the registrar office. 2. The updated enrollment report for the student is submitted. **Higher Authority (Imperium):** 1. Requests to access the student enrollment	**Paper:** Used for printing necessary documents. **Pen:** Used for writing something on the report.	**Computer:** Used for logging to the website and conducting respective tasks by higher authority and registrar office members. **Database Server:** Receiving data from the registrar office as well as sending data to them	**SPMS 3.0:** Used for updating the student enrollment information. **Operating System:** The user may use any OS such as Windows, Mar, Linux etc.	**SPMS 3.0 Database:** The updated student enrollment information is stored here.	**Internet:** To access the SPMS 3.0 software and the database, the registrar office personnel and higher authority use the internet.

	report. 2. View the student enrollment report in the form of a graph.		in order to store or update information into the database.			
Generate CQI Report	**Faculty:** 1. Launch the website first. 2. Find and select the desired course. 3. The system will present all activities upon clicking the student performance option. 4. The CQI report button will display PLO percentage upon click. 5. Check if a student falls below a certain PLO (soft copy) on that exam section. 6. Give feedback via rating on that specific exam.	**Paper:** When a faculty wants to print any types of documents then the paper is being used.	**Computer:** Computer is used by both the students and faculty members to log into the website and generate the report. **Database Server:** The faculty has access to the database where they can store or update the information into the database.	**SPMS 3.0:** The report has originated through the system. **Operating System:** The user may use any OS such as Windows, Mac, Linux etc.	**SPMS 3.0 Database:** The database is used for the purpose of storing the updated report.	**Internet:** Used by the faculty members to access the SPMS 3.0 software and database.

G. PROCESS MODEL - PROPOSED SYSTEM:

The Business Process Model and Notation provide an unambiguous description of the precise order of steps that will be taken to complete each process after understanding the role of each element in each process. Each module in this diagram will act as the high-level foundation from which the implementation specifics in the following chapter will be derived.

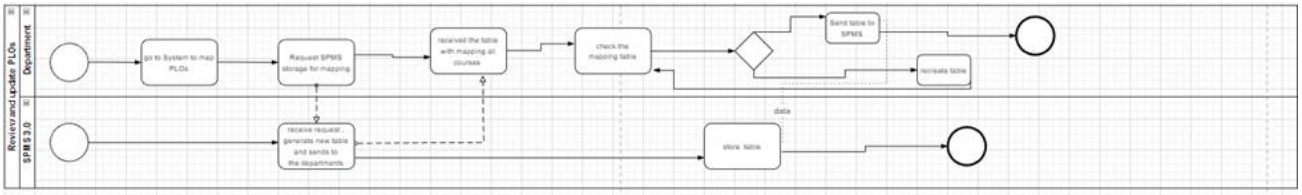


Fig 1.8: Review and update PLOs

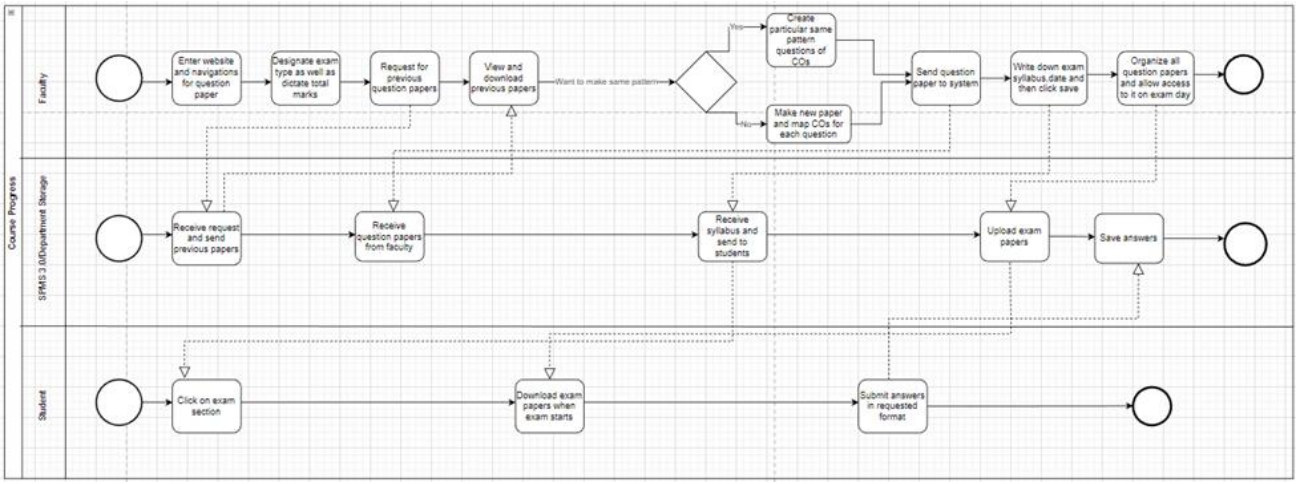


Fig 1.9: Course Progress

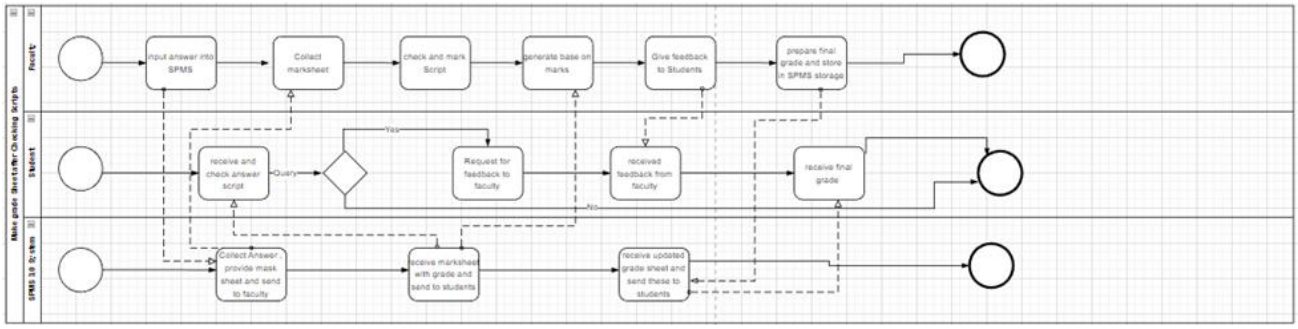


Fig 1.10: Make grade Sheet after Checking Scripts

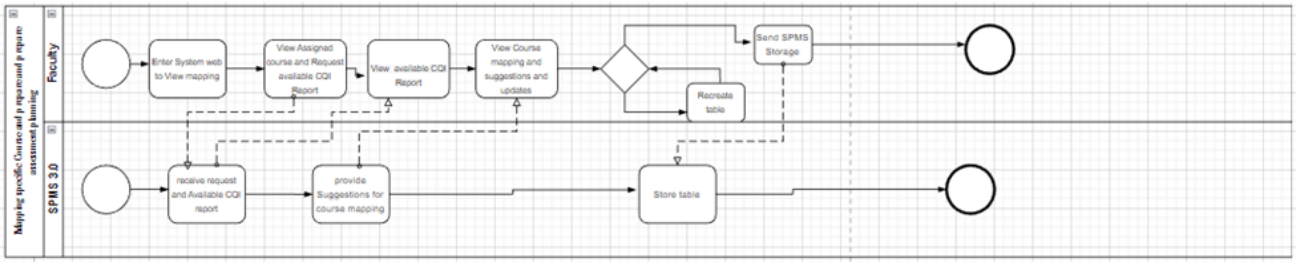


Fig 1.11: Mapping specific Course and prepare and prepare assessment planning

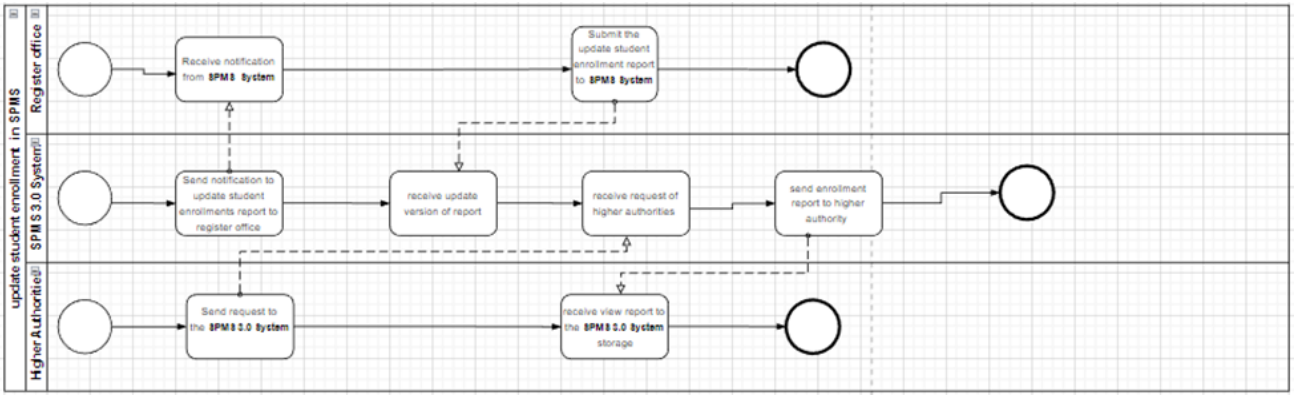


Fig 1.12: update student enrollment in SPMS

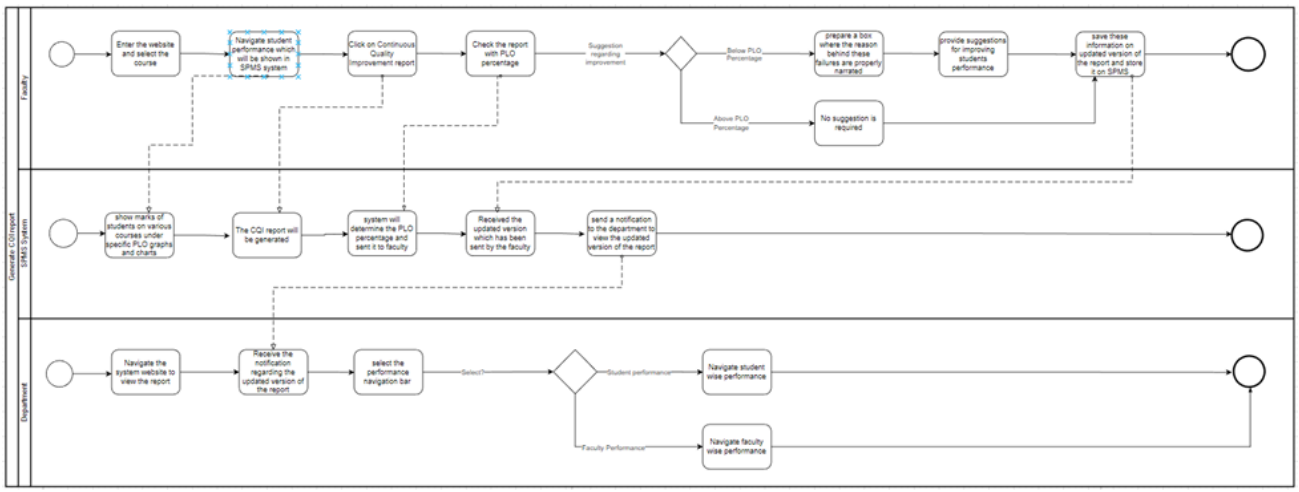


Fig 1.13: Generate CQI report

CHAPTER 3 - LOGICAL SYSTEM DESIGN:

In this chapter, we'll go through the steps of building a data model for our hypothetical system so that the data may be stored in a database. The links between various data objects, the rules, and the conceptual representation of the data objects are all included in this data model. Data modeling supports the visual representation of data and applies corporate policies, legal requirements, and governmental directives to the data. The consistency of naming conventions, default values, semantics,

and security are all ensured by data models, which also guarantee the accuracy of the data. For a better representation of all the data, we will be constructing our suggested system.

A. BUSINESS RULE [SPMS 3.0]:

Business rules outline the procedures, concepts, and limitations that control the data model. They are written in standard English sentences as opposed to the ERD so that a stakeholder who is not technically inclined can understand information about the data model without being aware of notational conventions. Our data model is governed by the following business rules:

1. A student must have one department. A STUDENT has StudentID, FirstName, LastName, DateofBirth, Gender, Email, Phone, Address, EnrollmentDate. A department must have many students.

2. A section mandatorily has many students. A student may enroll in many sections. A section includes SectionID, SectionNum, CourseID, FEmployeeID, Semester, Year.

3. Students may complete many evaluations. An EVALUATION includes EvaluationID, ObtainedMarks, StudentID, CourseID, QuestionID. An evaluation must be performed by at least one student.

4. An evaluation must have one Question. A Question must have many evaluations. Question assigns QuestionID, ExamType, TotalMarks, QuestionContent, Bloom'sTaxonomyLevel, FEmployeeID, COID. A question must create one faculty. A faculty creates many questions.

5. A CO's must map with one PLO's. A PLO's must map with one or many CO's. PLO includes PLOID, PLONum, Details, ProgramID.

6. A PLO must contain one program. A program contains one or many PLO's. A program has ProgramID, ProgramName, DepartmentID. A program must contain one or many courses. A Course must contain one program.

7. A program must belong to one department. A department must belong to one or many programs. A department contains DepartmentID, DepartmentName, SchoolID.

8. A department must contain one school. A School must contain one or many departments. A school includes SchoolID, SchoolName.

9. An employee has three sub-type(Dean, Department Head, Faculty). An employee

includes EmployeeID, FirstName, LastName, Email, Address, EmployeeType.

10. A school must be run by one or many Dean. A dean must run one school. A Dean has SchoolID, StartDate, EndDate.

11. A Department must manage one or many Department head. A department head must manage one department. A department head includes DepartmentID, StartDate, EndDate.

12. A Faculty must have one Department. A department must have one or many Faculties. A Faculty includes FEmployeeID, DepartmentID,COutlineID, Rank, JoinDate, ConsultantHour. A faculty may teach many sections. A section must be taught by one faculty.

13. A question must map with one CO's. A CO's maps with one or many questions. A CO's includes CONum, COID, CourseID, PLOID. And questions include QuestionNum,ExamType,TotalMarks,Bloom'sTaxonomyCategory, Bloom'sTaxonomyLevel. A CO must contain one Course. A Course contains one or many CO's.

14. A course may have many prerequisites. A course includes CourseID, CourseName, CourseType, CreditValues, ProgramID, COutlineID. A course must contain one course outline. A course outline may be one course.

15. A course outline must affiliate one mark distribution. A mark distribution may affiliate many courses. A course outline includes MarkDistribution. Mark distribution represents multi valued course assessment and percentage exam type wise(quiz, midterm, final, project) and CoursePolicy. Faculty must prepare one course outline. A course outline may prepare one faculty.

B. ENTITY RELATIONSHIP DIAGRAM:

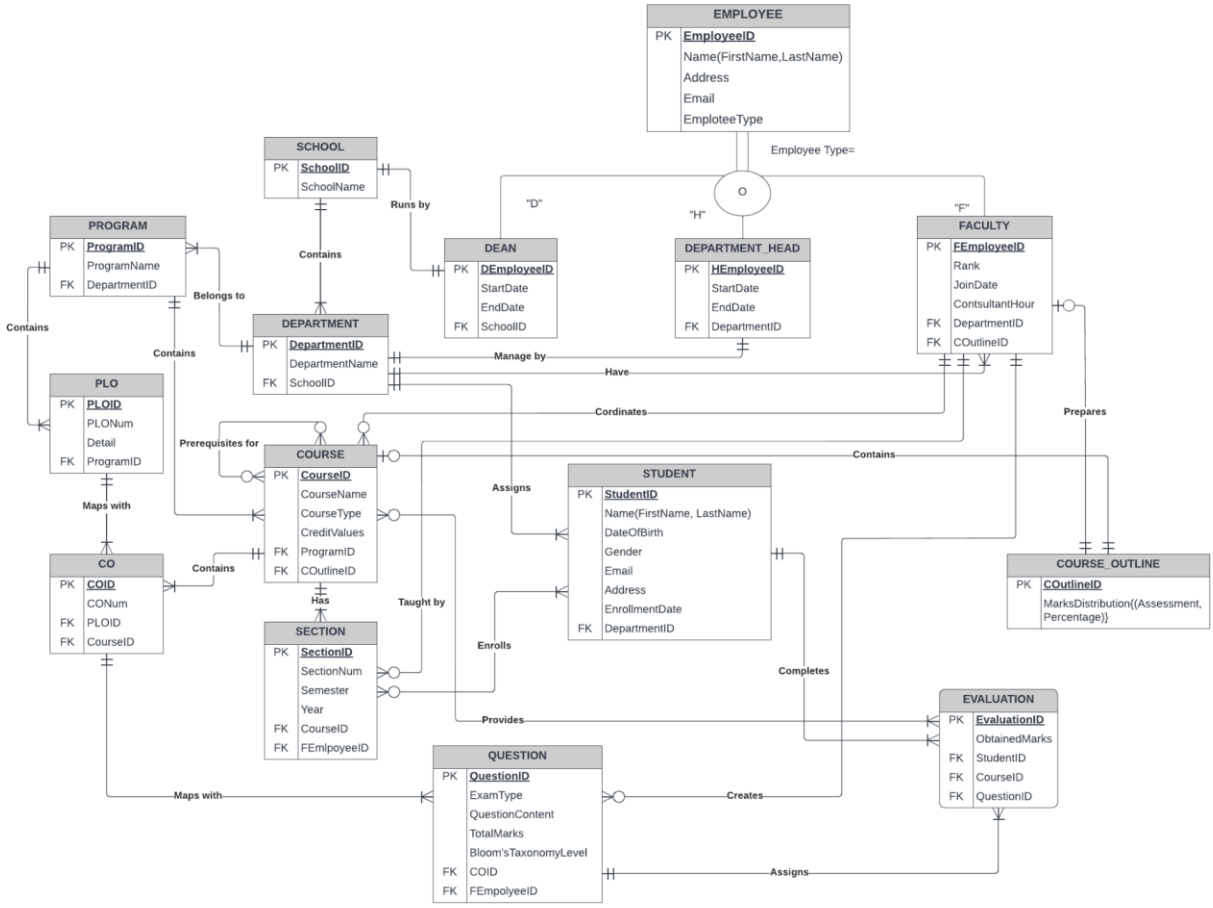


Figure 3.1: Entity relationship diagram

C. ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA:

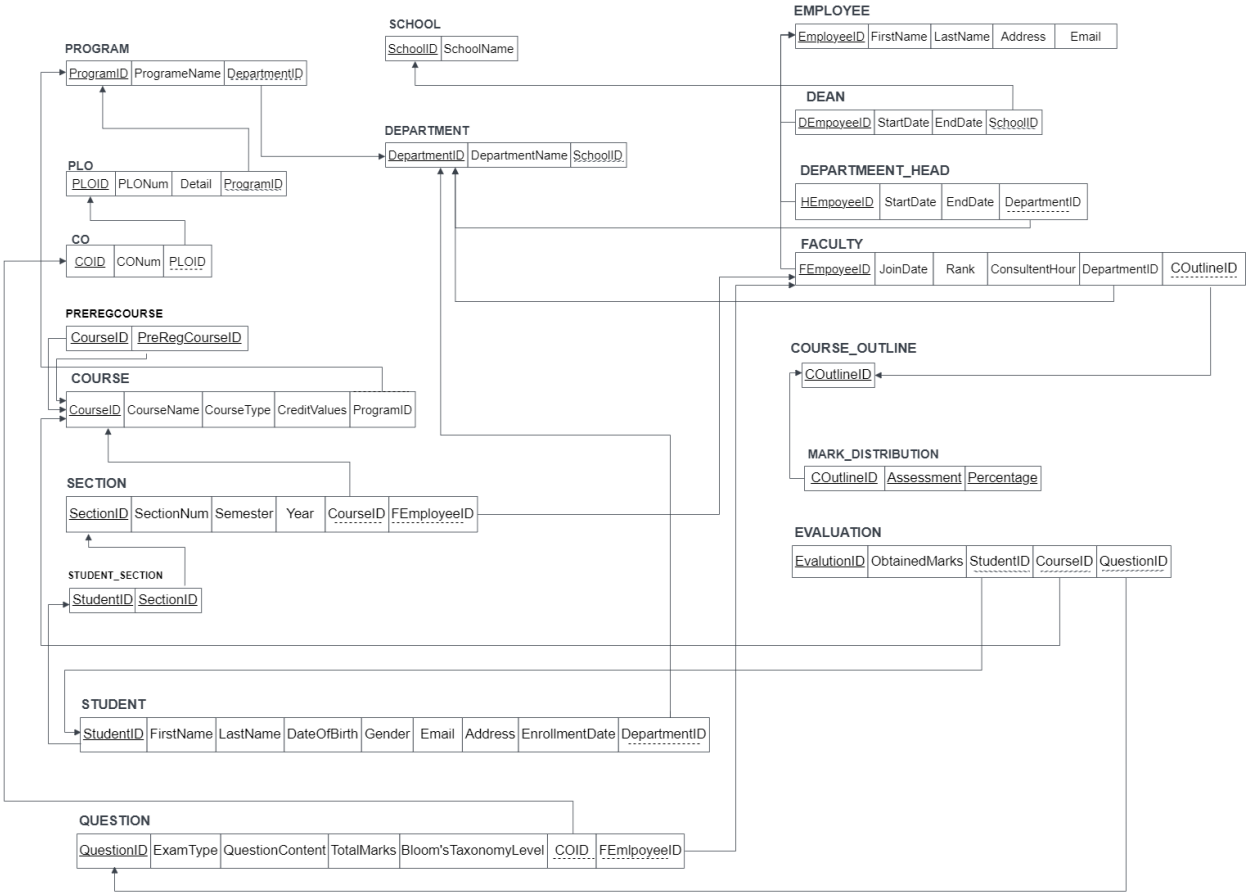


Figure 3.2: Entity relationship diagram

D. NORMALIZATION:

SchoolID L1	SchoolName L2
DepartmentID D1	DepartmentName, SchoolID D2, L1
ProgramID P1	ProgramName, DepartmentID P2, D1
PLOID A1	PLONum, Detail, ProgramID A2, A3, P1
COID B1	CONum, PLOID B2, A1
CourseID R1	PreRegCourseID R2
CourseID C1	CourseName, CourseType, CreditValues, ProgramID C2, C3, C4, P1
SectionID G1	SectionNum, Semester, Year, CourseID, FEmployeeID G2, G3, G4, C1, F1
StudentID N1	SectionID G1
StudentID S1	Firstname, LastName, DateOfBirth, Gender, Email, address, EnrollmentDate, DepartmentID S2, S3, S4, S5, S6, S7, S8, D1

QuestionID Q1	ExamType, QuestionContent, TotalMarks, Bloom'sTaxonomyCategory, Bloom'sTaxonomyLevel, COID, FEmployeeID Q2, Q3, Q4, Q5, Q6, B1, F1
EmployeeID E1	FirstName, LastName, Address, Email E2, E3, E4, E5
DEmployeeID I1	StartDate, EndDate, SchoolID I2, I3, L1
HEmployeeID H1	StartDate, EndDate, DepartmentID H2, H3, D1
FEmployeeID F1	Join date, Rank, ConsultentHour, DepartmentID, COutlineID F2, F3, F4, D1, J1
COutlineID J1	CoursePolicy J2
COutlineID K1	Assessment, Percentage K2, K3
EvalutionID M1	ObtainedMarks, StudentID, CourseID, QuestionID M2, S1, C1, Q1

L1 -> L2
D1 -> D2, L1
P1 -> P2, D1
A1 -> A2, A3, P1
B1 -> B2, B3, A1
R1 -> R2
C1 -> C2, C3, C4, P1
G1 -> G2, G3, G4, C1, F1
N1 -> G1
S1 -> S2, S3, S4, S5, S6, S7, D1
Q1 -> Q2, Q3, Q4, Q5, Q6, B1, F1
E1 -> E2, E3, E4, E5
I1 -> I2, I3, L1
H1 -> H2, H3, D1
F1 -> F2, F3, F4, D1
J1 -> J2
K1 -> K2, K3
M1 -> M2, S1, C1, Q1

1NF: A relation that has a primary key and in which there are no repeating groups.

Figure 3.3: 1NF

2NF: A relation in the first normal form in which every non-key attribute is fully functionally dependent on the primary key.

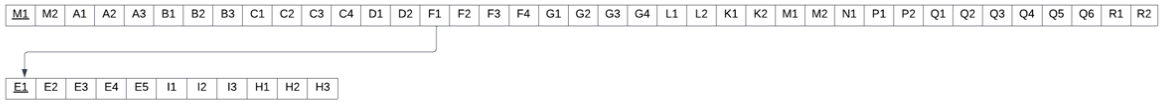


Figure 3.4: 2NF

3NF: A relation that is in second normal form and has no transitive dependencies.

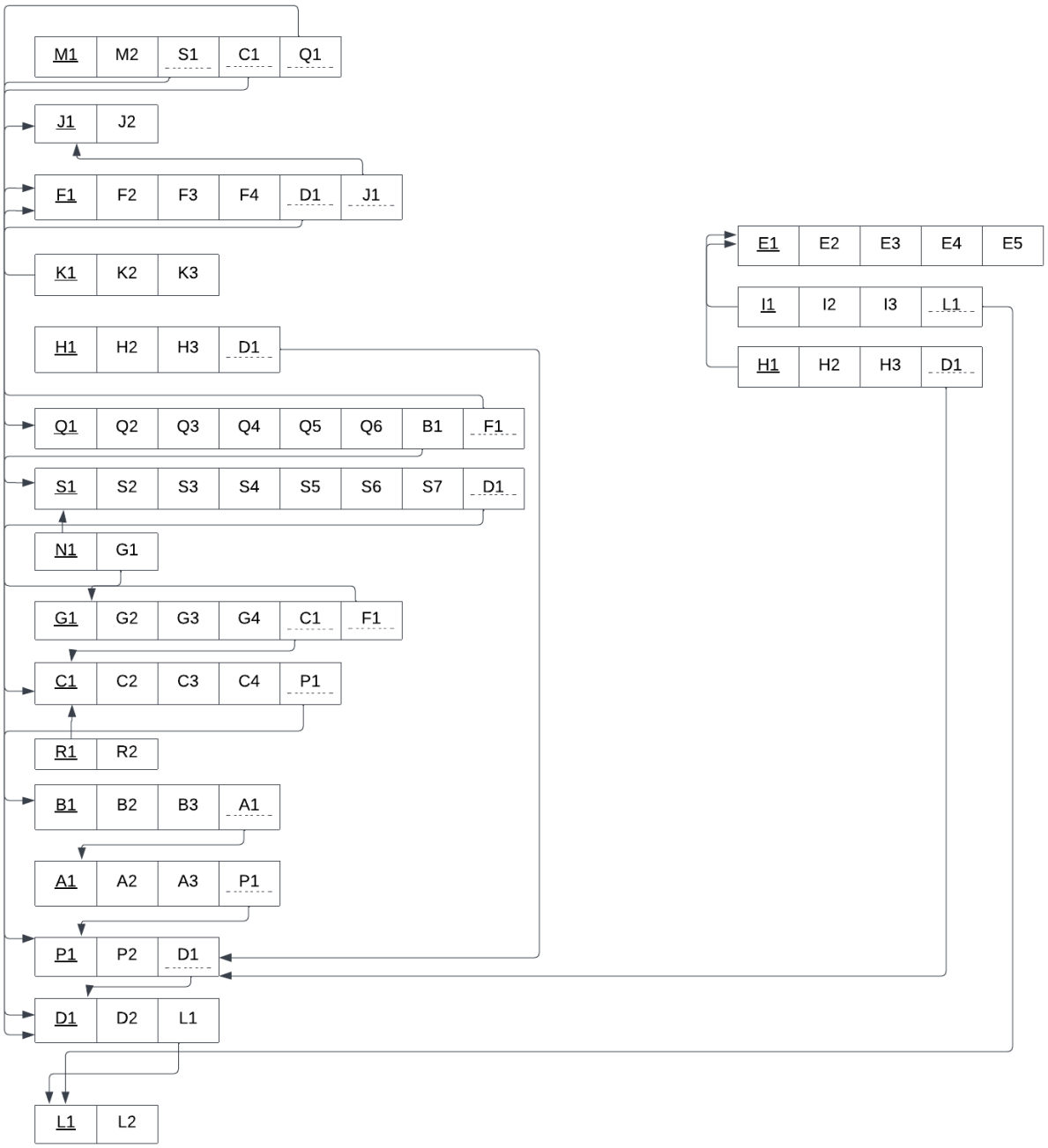


Figure 3.5: 3NF

BCNF: All determinants are candidate keys. There is no determinant that is not a

unique identifier. Here, all the relations already are in BCNF.

E. DATA DICTIONARY:

School_T

Name	Data Type	Size	Remarks
SchoolID	VARCHAR	5	This is the primary key of School. E.g: “SETS’’ or “SLASS”
SchoolName	VARCHAR	45	This is the name of the School. E.g: “School of Engineering, Technology & Science”.

Program_T

Name	Data Type	Size	Remarks
ProgramID	INTEGER		This is the primary key for a program. E.g: “1”
ProgramName	VARCHAR	30	This is the name of the program. E.g: “Bachelor of Science”
DepartmentID	VARCHAR	4	This is the foreign key from the

			Department table. E.g: “CSE” or “BBA’’

Department_T

Name	Data Type	Size	Remarks
DepartmentID	VARCHAR	5	This is the primary key for the Department table. E.g: “CSE”
DepartmentName	VARCHAR	45	This is the name of the department. E.g: “Computer Science and Engineering”.
SchoolID	VARCHAR	5	This is a foreign key from the School table. E.g: “SETS”or “SLASS’’.

Student_T

Name	Data Type	Size	Remarks
StudentID	INTEGER		This is the primary key for the Student table. E.g: “1830707”.
FirstName	VARCHAR	20	This is the first name of the student. E.g: “Md Akram”.

LastName	VARCHAR	20	This is the last name of the student. E.g: “Hossain”.
DateOfBirth	DATE	DD-MM-YYYY	This is the birth date of the student. E.g: “31-12-1998”.
Gender	VARCHAR	6	This is the gender of the student. E.g: “Male”.
Email	VARCHAR	30	This is the email of the student. E.g: “1830707@iub.edu.bd”
Phone	NUMERIC	11	This is the phone of the student. E.g: “01XXXXXXXXX”.
Address	VARCHAR	50	This is the address of the student. E.g: “House 238,Road 8,Tejgaon,Dhaka
DepartmentID	VARCHAR	5	This is the foreign key from the Department table. E.g: “CSE”
ProgramID	INTEGER		This is the foreign key from the Program table. E.g: “1”
EnrollmentDate	DATE	dd-mm-yyyy	This is the enrollment date of the student.

			E.g.: “1-1-2018”

CO_T

Name	Data Type	Size	Remarks
COID	VARCHAR	5	This is the primary key for the CO table. E.g: “CO1”.
CONum	INTEGER		This is the CO number. E.g: 1,2 etc.
CourseID	VARCHAR	8	This is the foreign key from the Course table. E.g: “CSE303”
PLOID	VARCHAR	5	This is the foreign key from the PLO table. E.g: “PLO1”

PLO_T

Name	Data Type	Size	Remarks
PLOID	VARCHAR	5	This is the primary key for Program Learning Outcome. E.g: “PLO1”
PLONum	INTEGER		This is the PLO number. E.g: “1”
Details	VARCHAR	50	This is the details for Program Learning

			Outcome. E.g: “An ability to select and apply the knowledge, technique, skills and modern tools of the computer science and engineering discipline ”
ProgramID	INTEGER		This is a foreign key from the Program table. E.g: “1”

Employee_T

Name	Date Type	Size	Remarks
EmployeeID	INTEGER		This is the primary key for Employee table. E.g: “1001”
FirstName	VARCHAR	20	This is the first name of the faculty. E.g: “Sadita”
LastName	VARCHAR	20	This is the last name of the faculty. E.g: “Ahmed”
Email	VARCHAR	30	This is the email address of the Student. E.g: “1675231@iub.edu.bd ”

Address	VARCHAR	30	This is the address of the Faculty. E.g: “House 14, Road 21, Sector 11,Baridara,Dhaka, Bangladesh”
EmployeeType	CHAR	1	This is the type of the employee. E.g: “F”

Course_T

Name	Data Type	Size	Remarks
CourseID	VARCHAR	8	This is the Primary Key for the Course. E.g: “CSE203”
CourseName	VARCHAR		This is the name of the Course. E.g: “Database Management System”
CreditValues	INTEGER		This is the number of credits for the Course. E.g: “3”
CourseType	VARCHAR		This is the type of the Course. E.g: “Core”
ProgramID	INTEGER		This is the foreign key from the program table. E.g: “1”
COutlinID	INTEGER		This is the Foreign Key from Course table.

Section_T

Name	Data Type	Size	Remarks
SectionID	INTEGER		This is the Primary Key for Section. E.g: “1”
SectionNum	INTEGER		This is the section number. E.g: “1”
CourseID	VARCHAR	8	This is the foreign key from the Course table. E.g: “CSE101”
FEmployeeID	NUMERIC	4	This is the foreign key from the Faculty table. E.g: “1001”
Semester	VARCHAR	6	This is the semester of the section. E.g: “Summer”

Question_T

Name	Data Type	Size	Remarks
QuestionID	INTEGER		This is the Primary Key for Question.
ExamType	VARCHAR	10	This is the name of the question. E.g: “Midterm”

TotalMarks	NUMBER		This is the total marks of the question. E.g: “30”
BloomsTaxonomyCategory	VARCHAR	10	This is the category of the question. E.g: “Creating”
BloomsTaxonomyLevel	VARCHAR	10	This is the difficulty of the question. E.g: “Midium”
COID	INTEGER		This is the Foreign Key from the Course Outcome table.
QuestionContent	INTEGER		This is the question number for question. E.g: “1,2,3....”
SectionID	INTEGER		This is the Foreign Key from Section table.
FEmployeeID			This is the Foreign Key from Faculty table.

Evalution_T

Name	Data Type	Size	Remarks
EvaluationID	INTEGER		This is the Primary Key for Enrollment.
ObtainedMarks	DECIMAL	5,2	This is the obtained marks of the student. E.g: “24.5”
QuestionID	INTEGER		This is the foreign key

			from the Question table.
CourseID	VARCHAR	8	This is the foreign key from the Course table. E.g: “CSE101”
StudentID	INTEGER		This is the foreign key from the Student table.

Dean_T

Name	Data Type	Size	Remarks
DEmployeeID	INTEGER		This is the foreign key from the Employee table. E.g: “4250”
SchoolID	VARCHAR	5	This is the SchoolID of the school DEAN manages. E.g: “SETS”
StartDate	DATE	dd-mm-yyyy	This is the starting date. E.g: “01-03-2020”
EndDate	DATE	dd-mm-yyyy	This is the date DEAN retire from his post. E.g: “01-03-2024”

DepartmentHead_T

Name	Data Type	Size	Remarks
HEmployeeID	INTEGER		This is the foreign key

			from the Employee table. E.g: “4250”
DepartmentID	VARCHAR	5	This is the DepartmentID of the department HEAD manages. E.g: “CSE”
StartDate	DATE	dd-mm-yyyy	This is the starting date. E.g: “01-03-2020”
EndDate	DATE	dd-mm-yyyy	This is the date HEAD retire from his post. E.g: “01-03-2024”

Faculty_T

Name	Data Type	Size	Remarks
FEmoployeeID	INTEGER		This is the foreign key from the Employee table. E.g: “4250”
DepartmentID	VARCHAR	5	This is the DepartmentID of the department faculty belongs to. E.g: “CSE”
JoinDate	DATE	dd-mm-yyyy	This is the starting date. E.g: “01-03-2020”
Rank	VARCHAR	20	This is the rank of the faculty. E.g: “Assistant

			Professor”
COutlineID	INTEGER		This is the Foreign Key from Course Outline table.

PreReqCourse_T

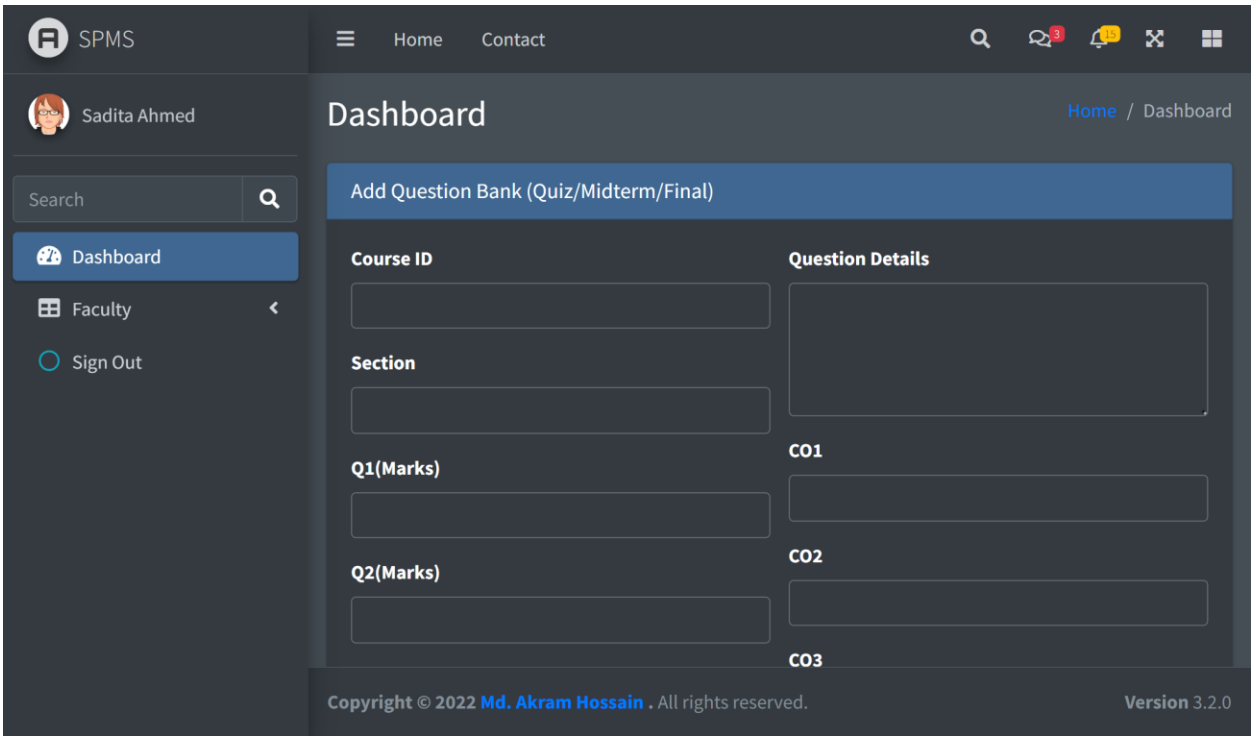
Name	Data Type	Size	Remarks
CourseID	VARCHAR	8	This is the foreign key from the Course table. E.g: “CSE303”
PreReqCourseID	VARCHAR	8	This is the foreign key from the Course table . E.g: CSE203

CourseOutline_T

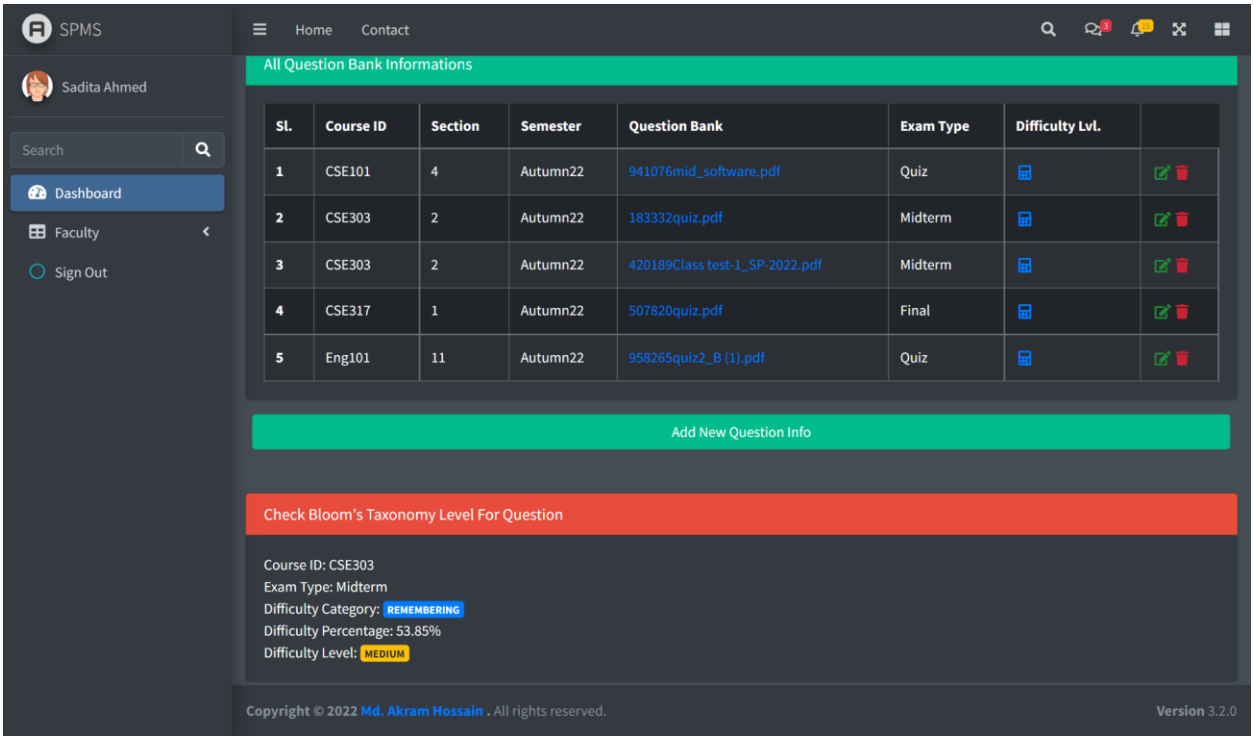
Name	Data Type	Size	Remarks
COutlineID	INTEGER		This is the primary key from the Course Outline table. E.g: “1233”
MarkDistribution	VARCHAR	15	This is the percentage range for assessment. E.g: “Project- 50%, Assessment-50%”.

CHAPTER 4 - PHYSICAL SYSTEM DESIGN:

A. INPUT FORM:



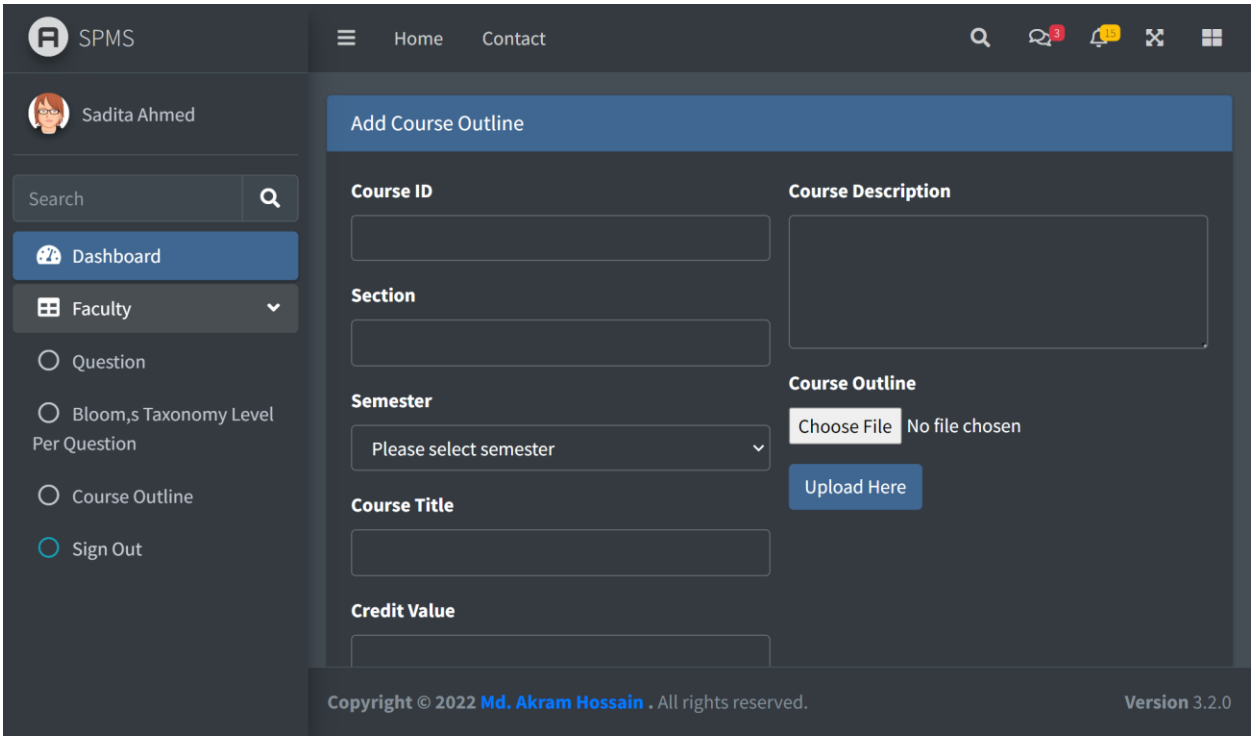

```php
$sql = " SELECT * FROM questionmark ";
$all_exam = mysqli_query($db, $sql);
$i = 0;

while ($row = mysqli_fetch_assoc($all_exam)) {
  // code...
  $id             = $row['id'];
  $course_id      = $row['course_id'];
  $section        = $row['section'];
  $semester       = $row['semester'];
  $question       = $row['question_details'];
  $q1             = $row['q1'];
  $q2             = $row['q2'];
  $q3             = $row['q3'];
  $q4             = $row['q4'];
  $co1            = $row['co1'];
  $co2            = $row['co2'];
  $co3            = $row['co3'];
  $co4            = $row['co4'];
  $exam_type      = $row['exam_type'];
  $i++;
  ?>

  <tr>
    <th scope="row"><?php echo $i; ?></th>

    <td><?php echo $course_id; ?></td>
    <td><?php echo $section; ?></td>
    <td><?php echo $semester; ?></td>
    <td><?php echo $question; ?></td>
    <td><?php echo $q1; ?></td>
    <td><?php echo $q2; ?></td>
    <td><?php echo $q3; ?></td>
    <td><?php echo $q4; ?></td>
    <td><?php echo $co1; ?></td>
    <td><?php echo $co2; ?></td>
    <td><?php echo $co3; ?></td>
    <td><?php echo $co4; ?></td>
    <td><?php echo $exam_type; ?></td>
```




```php
$category1 = array("choose","define","find","how","label","list","match","name","omit","recall","relate","select","show","spell","tell","what","when","where","which","who","why");
$category2 = array("compare","contrast","demonstrate","explain","extend","illustrate","infer","interpret","outline","relate","rephrase","show","summarize","translate");
$category3 = array("apply","build","choose","construct","develop","experiment with","identify","interview","make use of","model","organise","plan","select","solve","utilize");
$category4 = array("analyze","assume","categorize","classify","conclusion","contrast","discover","dissect","distinguish","divide","examine","function","interence","inspect","list","motive","relationships","simplify","survey","take part in","test for","theme");
$category5 = array("agree","appraise","assess","award","choose","conclude","criteria","criticize","decide","deduct","defend","determine","disprove","estimate","evaluate","explain","importance","influence","interpret","judge","justify","mark","measure","opinion","perceive","prioritize","prove","rate","recommend","rule on","select","support","value");
$category6 = array("adapt","bulid","change","choose","combine","compile","compose","construct","create","delete","design","develop","discuss","elaborate","estimate","formulate","happen","imagine","improve","invent","make up","maximize","minimize","modify","original","originate","plan","predict","propose","solution","solve","suppose","test","theory");
$count1 = 0; $count2 = 0; $count3 = 0; $count4 = 0; $count5 = 0; $count6 = 0;

$q1 = array("how", "list", "infer", "choose", "classify", "conclude", "combine", "compose","delete");

if (isset($_GET['Calculate'])) {
  # code...
  $cal_id = $_GET['Calculate'];
  $sql = " SELECT * FROM questionbank WHERE id = $cal_id ";
  $all_word = mysqli_query($db, $sql);

  while ($rows = mysqli_fetch_array($all_word)) {
    # code...
    $word_id        = $rows['id'];
    $course_id      = $rows['course_id'];
    $word           = $rows['word'];
    $exam_type      = $rows['exam_type'];
  }

  echo "Course ID: " . $course_id . "<br>";
  echo "Exam Type: " . $exam_type . "<br>";
```

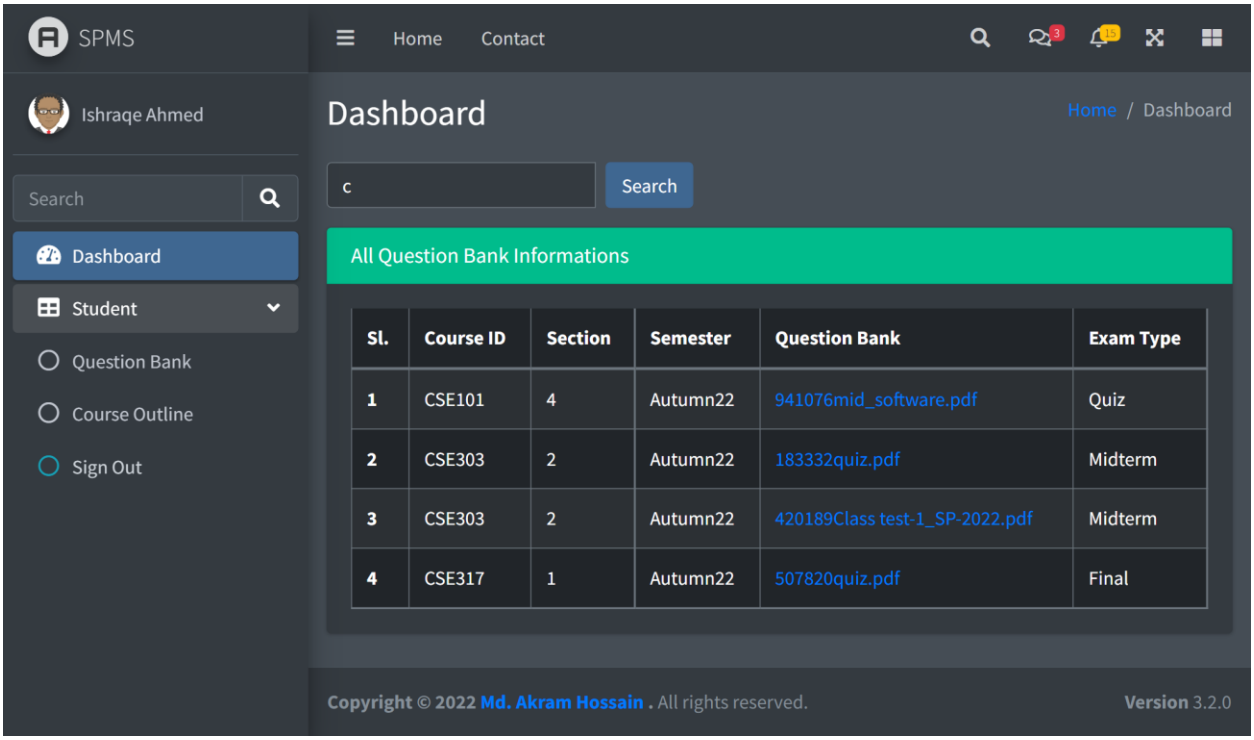
```php
}

// echo "Count: " . $count1. "\n". $count2. "\n" . $count3. "\n" . $count4. "\n" . $count5. "\n" .
$count6. "<br>";

$sum = array("$count1", "$count2", "$count3", "$count4", "$count5", "$count6");
$max = (max(array("$count1", "$count2", "$count3", "$count4", "$count5", "$count6")));

$percentage = ($max/array_sum($sum))*100 ;

if ($count1 == $max) {
  # code...
  echo 'Difficulty Category: <span class="badge badge-primary">REMEMBERING</span>' . "<br>";
}
elseif ($count2 == $max) {
  # code...
  echo 'Difficulty Category: <span class="badge badge-primary">UNDERSTANDING</span>' . "<br>";
}
elseif ($count3 == $max) {
  # code...
  echo 'Difficulty Category: <span class="badge badge-primary">APPLYING</span>' . "<br>";
}
elseif ($count4 == $max) {
  # code...
  echo 'Difficulty Category: <span class="badge badge-primary">ANALYZING</span>' . "<br>";
}
elseif ($count5 == $max) {
  # code...
  echo 'Difficulty Category: <span class="badge badge-primary">EVALUATING</span>' . "<br>";
}
elseif ($count6 == $max) {
  # code...
  echo 'Difficulty Category: <span class="badge badge-primary">CREATING</span>' . "<br>";
}
else echo "Not Found." . "<br>";

echo "Difficulty Percentage: " . round($percentage, 2) . "%" . "<br>";

if (($percentage >= 0) && ($percentage <= 33) ) {
  # code
```

```php
253                 }
254                 elseif ($count3 == $max) {
255                   # code...
256                   echo 'Difficulty Category: <span class="badge badge-primary">APPLYING</span>' . "<br>";
257                 }
258                 elseif ($count4 == $max) {
259                   # code...
260                   echo 'Difficulty Category: <span class="badge badge-primary">ANALYZING</span>' . "<br>";
261                 }
262                 elseif ($count5 == $max) {
263                   # code...
264                   echo 'Difficulty Category: <span class="badge badge-primary">EVALUATING</span>' . "<br>";
265                 }
266                 elseif ($count6 == $max) {
267                   # code...
268                   echo 'Difficulty Category: <span class="badge badge-primary">CREATING</span>' . "<br>";
269                 }
270                 else echo "Not Found." . "<br>";
271
272                 echo "Difficulty Percentage: " . round($percentage, 2) . "%" . "<br>";
273
274                 if (($percentage >= 0) && ($percentage <= 33) ) {
275                   # code...
276                   echo 'Difficulty Level: <span class="badge badge-success">LOW</span>' . "<br>";
277                 }
278                 elseif (($percentage >= 34) && ($percentage <= 63) ) {
279                   # code...
280                   echo 'Difficulty Level: <span class="badge badge-warning">MEDIUM</span>' . "<br>";
281                 }
282                 elseif ($percentage >= 64 ) {
283                   # code...
284                   echo 'Difficulty Level: <span class="badge badge-danger">HIGH</span>' . "<br>";
285                 }
286                 else echo "Not Found." . "<br>";
287
288               }
289
290
291               // for ($i=0; $i < sizeof($category1); $i++) {
```

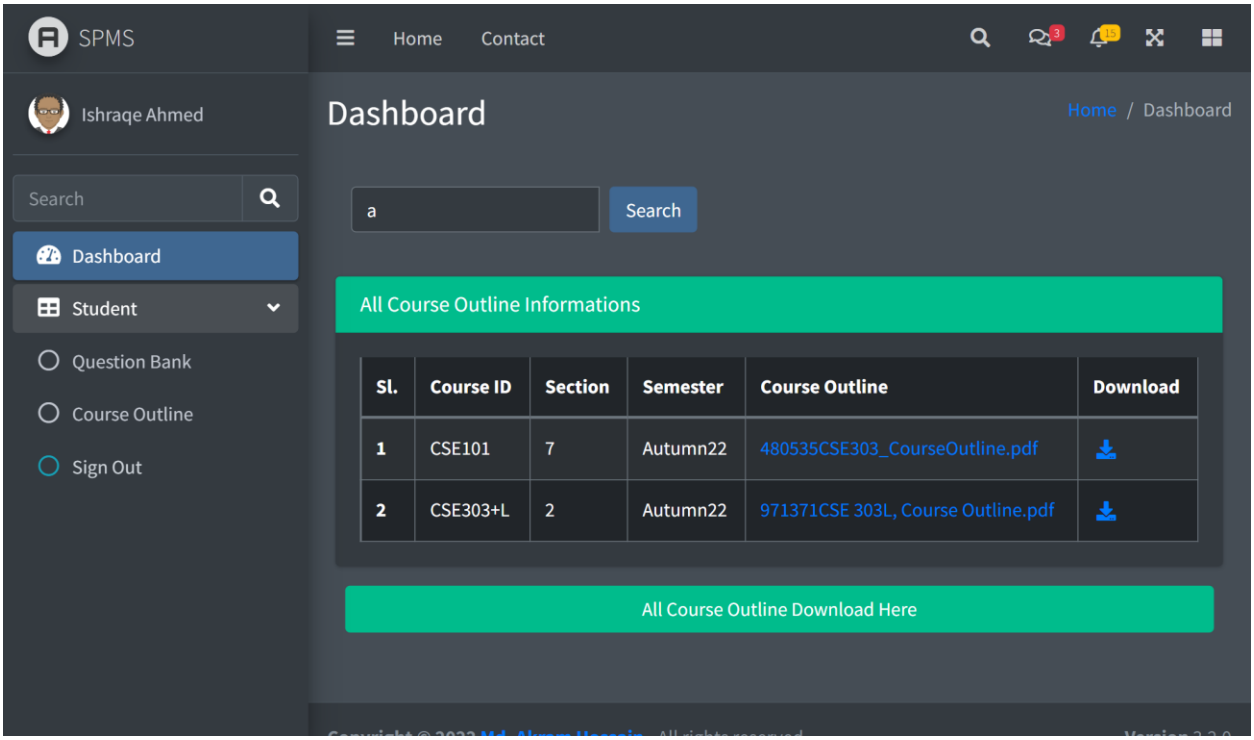



```php
      </section>
    <?php
  }
  elseif ($do == "Insert") {
    // code...
    if (isset($_POST['upload'])) {
      // code...
      $course_id      = $_POST['courseid'];
      $section        = $_POST['section'];
      $course_title   = $_POST['course_title'];
      $credit_value   = $_POST['credit_value'];
      $duration       = $_POST['duration'];
      $course_desc    = $_POST['course_desc'];



      $pdf       = $_FILES['pdf']['name'];
      $tmp_pdf   = $_FILES['pdf']['tmp_name'];

      $random_number = rand(0, 1000000);
      $pdfFile = $random_number.$pdf;

      move_uploaded_file($tmp_pdf, "dist/img/users/" . $pdfFile);

      $sql = " INSERT INTO courseoutline( course_id, section, semester, course_outline, course_title, credit_value, duration,
        course_desc ) values( '$course_id', '$section', '$semester', '$pdfFile', '$course_title', '$credit_value', '$duration',
        '$course_desc' ) ";
      $uploadInfo = mysqli_query($db, $sql);

      if ($uploadInfo) {
        // code...
        header("Location: courseoutline.php");
      }
      else {
        echo "Query Filed" . mysqli_error($db);
      }

    }
  }
  elseif ($do == "Edit") {
```

B. OUTPUT FORMS:



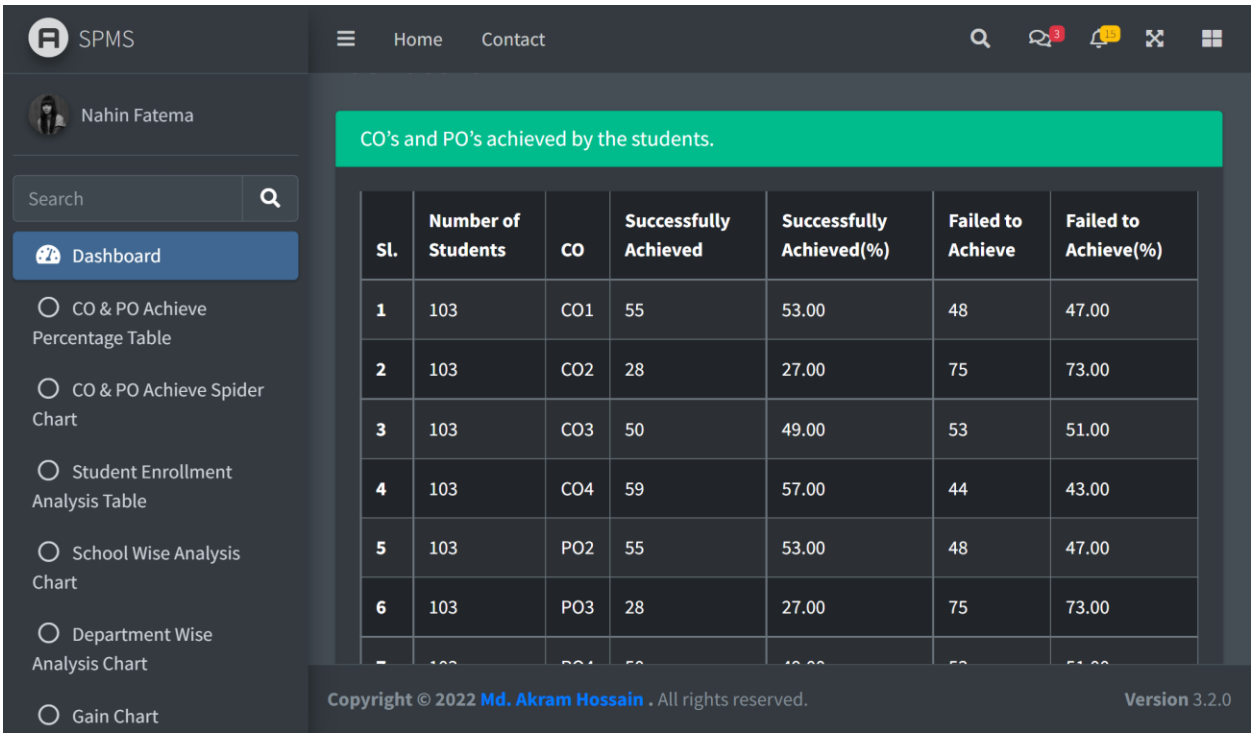

```php
<?php

  if (isset($_GET['search'])) {
    // code...
    $filtervalue = $_GET['search'];
    $sql = " SELECT * FROM questionbank WHERE CONCAT(course_id,semester,exam_type) LIKE '%$filtervalue%' ";
    $all_question = mysqli_query($db, $sql);
    $i = 0;

    while ($row = mysqli_fetch_assoc($all_question)) {
      // code...
      $id             = $row['id'];
      $course_id      = $row['course_id'];
      $section        = $row['section'];
      $semester       = $row['semester'];
      $question       = $row['question'];
      $exam_type      = $row['exam_type'];
      $i++;
      ?>

      <tr>
        <th scope="row"><?php echo $i; ?></th>

        <td><?php echo $course_id; ?></td>
        <td><?php echo $section; ?></td>
        <td><?php echo $semester; ?></td>
        <td><a href="dist/img/users/<?php echo $question; ?>" target="_blank" id="anchor"><?php echo $question
          ; ?></a></td>
        <td><?php echo $exam_type; ?></td>
        <!-- <td></td> -->

      </tr>
      <?php
    }
  }
  else { echo "No Record Found "; }
```




```php
<?php

  if (isset($_GET['search'])) {
    // code...
    $filtervalues = $_GET['search'];
    $sql = " SELECT * FROM courseoutline WHERE CONCAT(course_id,semester) LIKE '%$filtervalues%' ";
    $all_course_outline = mysqli_query($db, $sql);
    $i = 0;

    while ($row = mysqli_fetch_assoc($all_course_outline)) {
      // code...
      $id                 = $row['id'];
      $course_id          = $row['course_id'];
      $section            = $row['section'];
      $semester           = $row['semester'];
      $course_outline     = $row['course_outline'];
      $i++;
      ?>

      <tr>
        <th scope="row"><?php echo $i; ?></th>

        <td><?php echo $course_id; ?></td>
        <td><?php echo $section; ?></td>
        <td><?php echo $semester; ?></td>
        <td><a href="dist/img/users/<?php echo $course_outline; ?>" target="_blank" id="anchors"><?php echo $
          course_outline; ?></a></td>
        <!-- <td></td> -->
        <td>
          <div class="btn-group">
            <a href="singlepdf.php?PDF=<?php echo $id; ?>">
              <i class="fas fa-download"></i>
            </a>
          </div>
        </td>

      </tr>
      <?php
    }
```

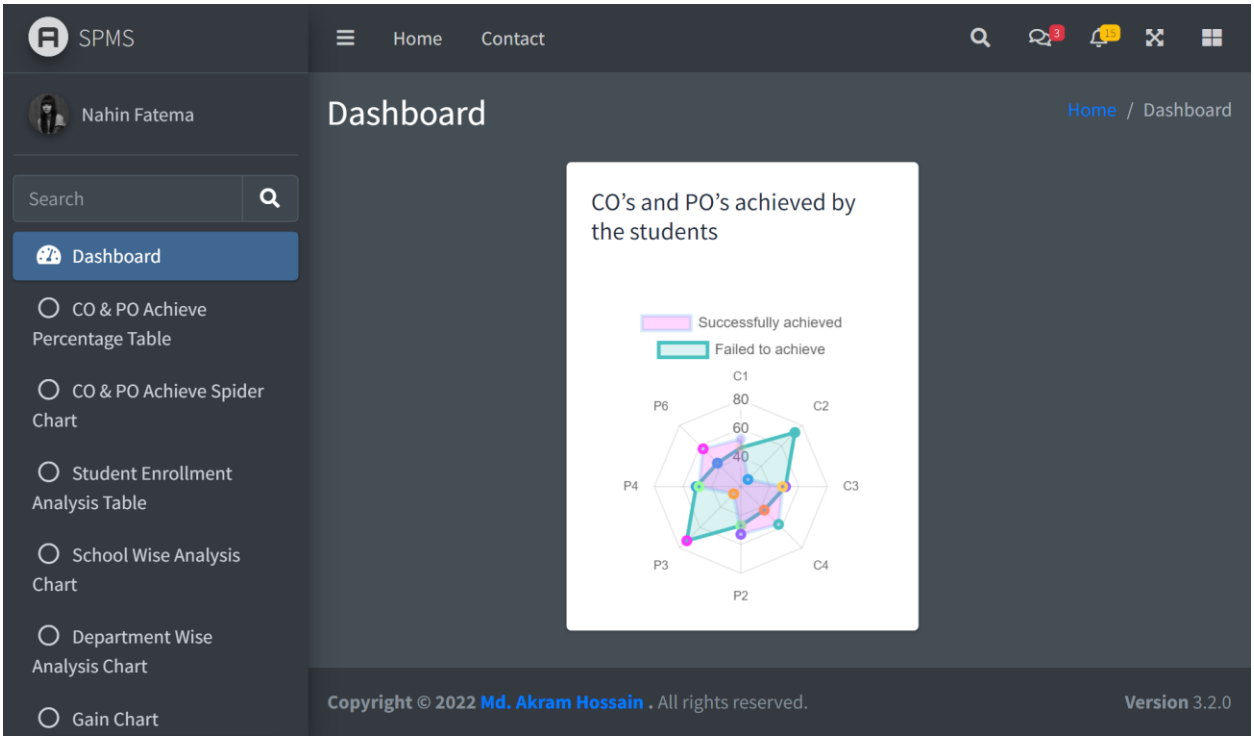
```php
<?php

include"include/db.php";
require('fpdf/fpdf.php');

if (isset($_GET['PDF'])) {
    # code...
    $co_id = $_GET['PDF'];
    $sql = " SELECT * FROM courseoutline WHERE id = '$co_id' ";
    $selected_co = mysqli_query($db, $sql);

    while ($row = mysqli_fetch_assoc($selected_co)) {
        # code...
        $pdf = new FPDF('P','mm','A3');

        $pdf->AddPage();
        $pdf->SetFont('Times','',12);

        $pdf->Cell(260,10,'Course ID: ' . $row['course_id'],0,1,'C',false);
        $pdf->Cell(260,10,'Course Title: ' . $row['course_title'],0,1,'C',false);
        $pdf->Cell(260,10,'Semester: ' . $row['semester'],0,1,'C',false);
        $pdf->Cell(260,10,'Credit Value: ' . $row['credit_value'],0,1,'C',false);
        $pdf->Image('dist/img/users/outline.png',0,80);
        $pdf->Image('dist/img/users/md.png',0,200);
        $pdf->Image('dist/img/users/cps.png',8,308);
        $pdf->Cell(260,10,'Contact Hour/Week: ' . $row['duration'] . ' minitues',0,1,'C',false);

        // $pdf->Cell(60,10,'Course Description: ' . $row['course_desc'] ,0,1,'L',false);
        $pdf->ln(30);

    }
    $pdf->Output();
}

// $pdf = new FPDF('P','mm','A4');
// $pdf->AddPage();
// $pdf->SetFont('Arial','B',16);
// $pdf->Cell(40,10,'Hello World!');


```

```php
<?php

include"include/db.php";
require('fpdf/fpdf.php');

$sql = " SELECT * FROM courseoutline ";
$selected_co = mysqli_query($db, $sql);
$pdf = new FPDF('P','mm','A3');

while ($row = mysqli_fetch_assoc($selected_co)) {
    # code...

    $pdf->AddPage();
    $pdf->SetFont('Times','',12);

    $pdf->Cell(260,10,'Course ID: ' . $row['course_id'],0,1,'C',false);
    $pdf->Cell(260,10,'Course Title: ' . $row['course_title'],0,1,'C',false);
    $pdf->Cell(260,10,'Semester: ' . $row['semester'],0,1,'C',false);
    $pdf->Cell(260,10,'Credit Value: ' . $row['credit_value'],0,1,'C',false);
    $pdf->Image('dist/img/users/outline.png',0,80);
    $pdf->Image('dist/img/users/md.png',0,200);
    $pdf->Image('dist/img/users/cps.png',8,308);
    $pdf->Cell(260,10,'Contact Hour/Week: ' . $row['duration'] . ' minitues',0,1,'C',false);

    // $pdf->Cell(60,10,'Course Description: ' . $row['course_desc'] ,0,1,'L',false);
    $pdf->ln(30);
}
$pdf->Output();


?>
```



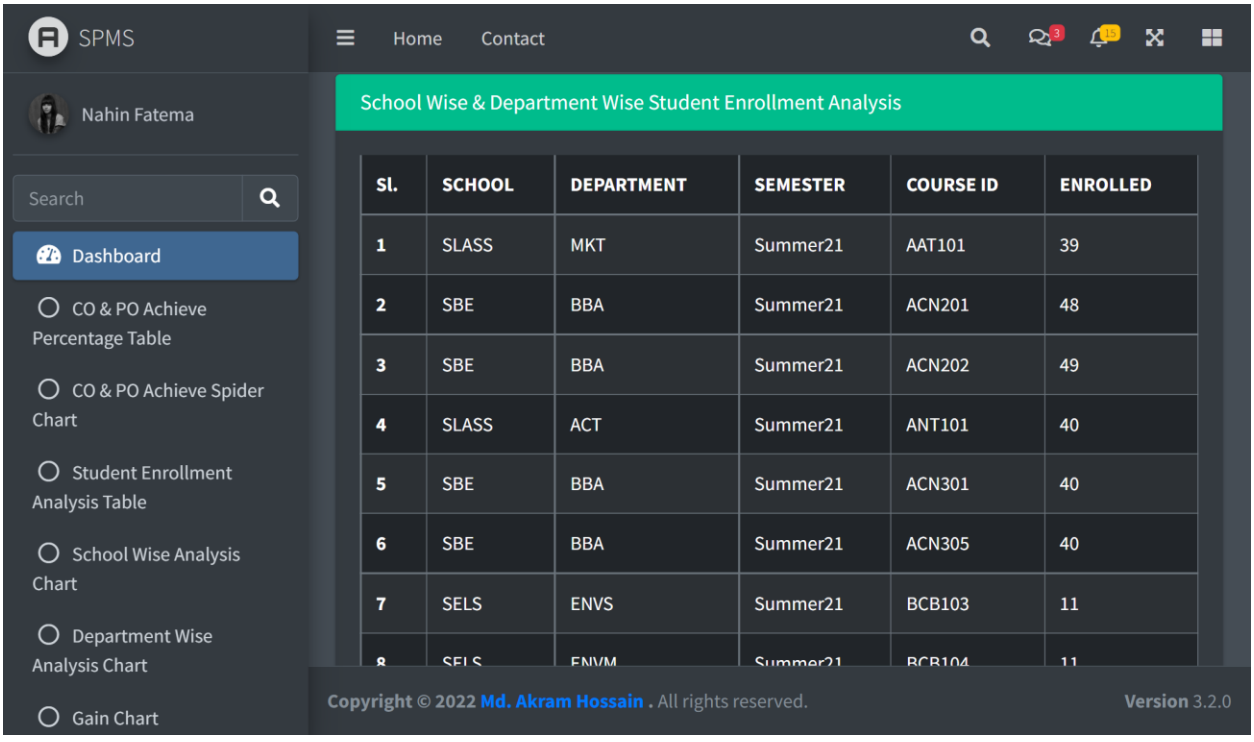
Sl.	Number of Students	CO	Successfully Achieved	Successfully Achieved(%)	Failed to Achieve	Failed to Achieve(%)
1	103	CO1	55	53.00	48	47.00
2	103	CO2	28	27.00	75	73.00
3	103	CO3	50	49.00	53	51.00
4	103	CO4	59	57.00	44	43.00
5	103	PO2	55	53.00	48	47.00
6	103	PO3	28	27.00	75	73.00

```
    <tr>
    <th scope="col">Sl.</th>
    <th scope="col">Number of Students</th>
    <th scope="col">CO</th>
    <th scope="col">Successfully Achieved</th>
    <th scope="col">Successfully Achieved(%)</th>
    <th scope="col">Failed to Achieve</th>
    <th scope="col">Failed to Achieve(%)</th>
    </tr>
</thead>
<tbody>
<?php

    $sql = "SELECT * FROM spiderchart";
    $all_cat = mysqli_query($db, $sql);
    $i = 0;

    while ($row = mysqli_fetch_assoc($all_cat)) {
        // code...

        $cat_id                       = $row['id'];
        $cat_total_student            = $row['number_of_students'];
        $cat_co                       = $row['CO'];
        $cat_successfully_achieved    = $row['successfully_achieved'];
        $cat_successfully_achieved_per = $row['successfully_achieved_per'];
        $cat_failed_to_achieve        = $row['failed_to_achieve'];
        $cat_failed_to_achieve_per    = $row['failed_to_achieve_per'];
        $i++;

        ?>

        <tr>
        <th scope="row"><?php echo $i; ?></th>
        <td><?php echo $cat_total_student; ?></td>
        <td><?php echo $cat_co; ?></td>
        <td><?php echo $cat_successfully_achieved; ?></td>
        <td><?php echo $cat_successfully_achieved_per; ?></td>
        <td><?php echo $cat_failed_to_achieve; ?></td>
        <td><?php echo $cat_failed_to_achieve_per; ?></td>
```

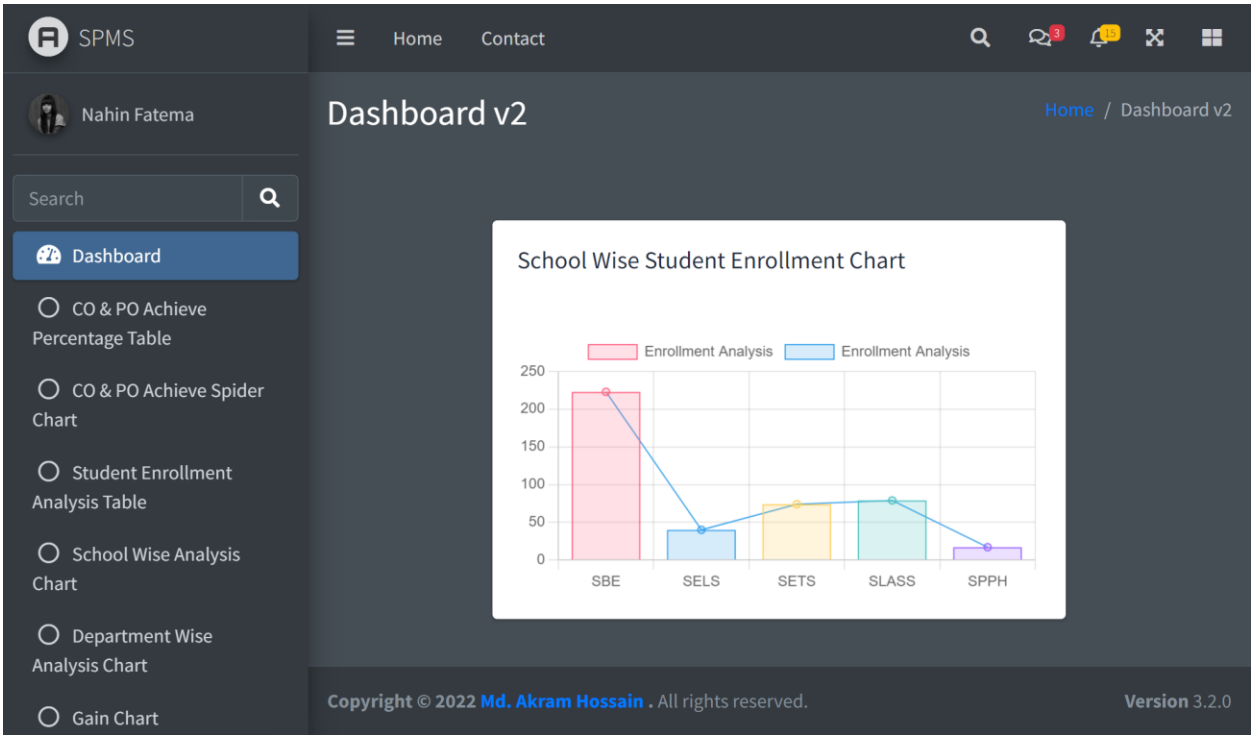



```php
                $sql = "SELECT * FROM spiderchart";
                $all_cat = mysqli_query($db, $sql);
                $i = 0;

                while ($row = mysqli_fetch_assoc($all_cat)) {
                    // code...

                    $cat_successfully_achieved_per[] = $row['successfully_achieved_per'];
                    $cat_failed_to_achieve_per[]     = $row['failed_to_achieve_per'];

                }

            ?>

                <div class="col-lg-5 offset-3 my-0">
                        <div class="card bg-white">
                            <div class="card-body">
                                <h5>CO's and PO's achieved by the
students</h5>
                            </div>
                            <div class="card-body">
                                <canvas id="myChart"></canvas>
                            </div>
                        </div>
                </div>
                <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
                <script>
                  const cat_successfully_achieved_per = <?php echo json_encode($cat_successfully_achieved_per); ?>;
                  const cat_failed_to_achieve_per = <?php echo json_encode($cat_failed_to_achieve_per); ?>;
                  var ctx = document.getElementById('myChart').getContext('2d');
                  var myChart = new Chart(ctx, {
                      type: 'radar',
                      data: {
                          labels: ['C1', 'C2', 'C3', 'C4', 'P2', 'P3', 'P4', 'P6'],
```

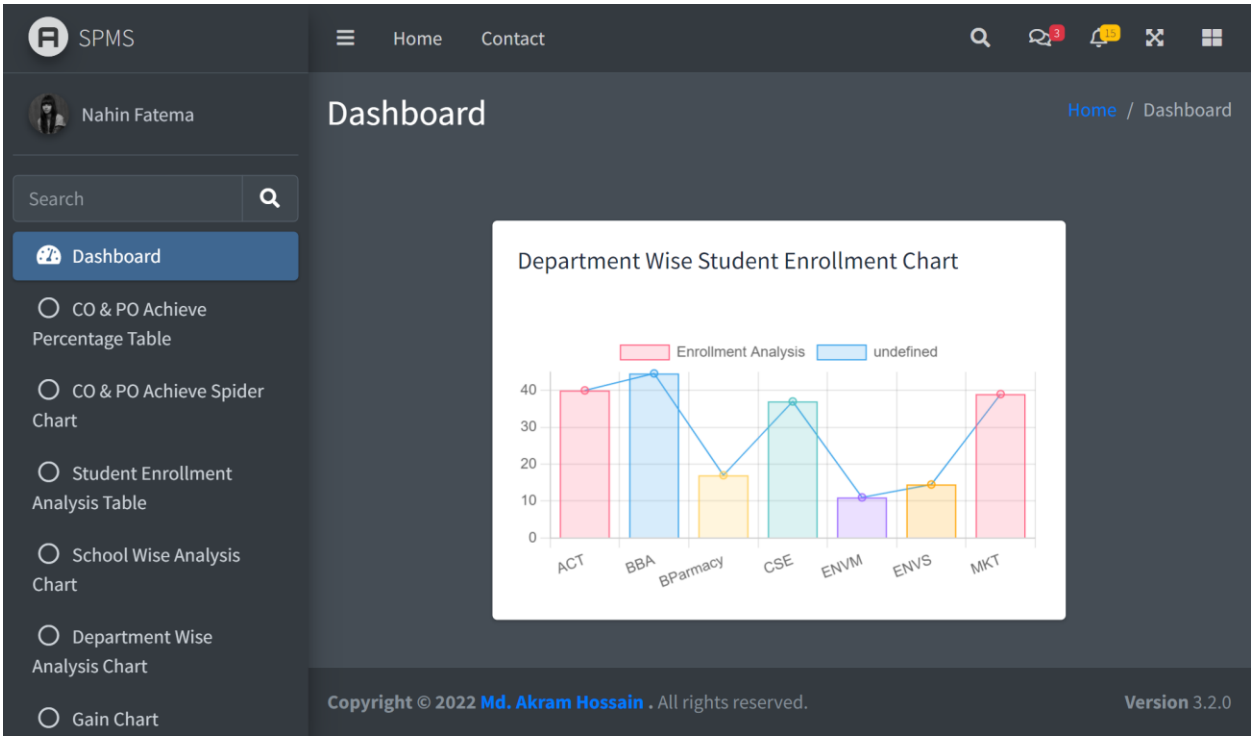
```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
  const cat_successfully_achieved_per = <?php echo json_encode($cat_successfully_achieved_per); ?>;
  const cat_failed_to_achieve_per = <?php echo json_encode($cat_failed_to_achieve_per); ?>;
  var ctx = document.getElementById('myChart').getContext('2d');
  var myChart = new Chart(ctx, {
      type: 'radar',
      data: {
          labels: ['C1', 'C2', 'C3', 'C4', 'P2', 'P3', 'P4', 'P6'],
          datasets: [{
              label: 'Successfully achieved',
              data: cat_successfully_achieved_per,
              backgroundColor: [
                  'rgba(255, 51, 253, 0.2)',
                  'rgba(54, 162, 235, 0.2)',
                  'rgba(255, 206, 86, 0.2)',
                  'rgba(75, 192, 192, 0.2)',
                  'rgba(153, 102, 255, 0.2)',
                  'rgba(255, 159, 64, 0.2)',
                  'rgba(153, 255, 153, 0.2)',
                  'rgba(75, 192, 192, 0.2)'
              ],
              borderColor: [
                  'rgba(54, 162, 235, 0.2)',
                  'rgba(54, 162, 235, 1)',
                  'rgba(255, 206, 86, 1)',
                  'rgba(75, 192, 192, 1)',
                  'rgba(153, 102, 255, 1)',
                  'rgba(255, 159, 64, 1)',
                  'rgba(153, 255, 153, 1)',
                  'rgba(255, 51, 253, 1)'
              ],
              borderWidth: 3
          },{
              label: 'Failed to achieve',
              data: cat_failed_to_achieve_per,
              backgroundColor: [
                  'rgba(255, 206, 86, 0.2)',
                  'rgba(75, 192, 192, 0.2)',
```



```
                <tr>
                <th scope="col">Sl.</th>
                <th scope="col">SCHOOL</th>
                <th scope="col">DEPARTMENT</th>
                <th scope="col">SEMESTER</th>
                <th scope="col">COURSE ID</th>
                <th scope="col">ENROLLED</th>
                <!-- <th scope="col"></th> -->
                </tr>
            </thead>
            <tbody>
            <?php

                $sql = "SELECT * FROM enrollment";
                $all_enroll = mysqli_query($db, $sql);
                $i = 0;

                while ($row = mysqli_fetch_assoc($all_enroll)) {
                    // code...

                    $enroll_id    = $row['id'];
                    $school       = $row['school'];
                    $department   = $row['department'];
                    $semester     = $row['semester'];
                    $course_id    = $row['course_id'];
                    $enrolled     = $row['enrolled'];
                    $i++;

                    ?>

                    <tr>
                    <th scope="row"><?php echo $i; ?></th>
                    <td><?php echo $school; ?></td>
                    <td><?php echo $department; ?></td>
                    <td><?php echo $semester; ?></td>
                    <td><?php echo $course_id; ?></td>
                    <td><?php echo $enrolled; ?></td>

                    </tr>
```



```php
<?php

    $sql = "SELECT school, SUM(enrolled) AS total
            FROM enrollment
            GROUP BY school";
    $all_enroll = mysqli_query($db, $sql);
    $i = 0;

    // $total = array();

    while ($row = mysqli_fetch_assoc($all_enroll)) {
        // code...

        $school[]  = $row['school'];
        $total[]   = $row['total'];
        $i++;


    }

?>

    <div class="col-lg-8 offset-2 my-5">
            <div class="card bg-white">
                <div class="card-body">
                    <h5>School Wise Student Enrollment Chart</h5>
                </div>
                <div class="card-body">
                    <canvas id="myChart"></canvas>
                </div>
            </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script>
        // Setup Block
        const school = <?php echo json_encode($school); ?>;
        const total = <?php echo json_encode($total); ?>;
        const data = {
        labels: school,
```



```php
<?php

    $sql = "SELECT department, AVG(enrolled) AS total
            FROM enrollment
            GROUP BY department";
    $all_enrolls = mysqli_query($db, $sql);
    $i = 0;

    // $total = array();

    while ($row = mysqli_fetch_assoc($all_enrolls)) {
        // code...

        $department[]  = $row['department'];
        $total[]   = $row['total'];
        $i++;


    }

?>

    <div class="col-lg-8 offset-2 my-5">
            <div class="card bg-white">
                <div class="card-body">
                    <h5>Department Wise Student Enrollment Chart</h5>
                </div>
                <div class="card-body">
                    <canvas id="myChart"></canvas>
                </div>
            </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script>
        // Setup Block
        const department = <?php echo json_encode($department); ?>;
        const total = <?php echo json_encode($total); ?>;
        const data = {
        labels: department,
            datasets: [{
```

CHAPTER 5 - CONCLUSION:

A. PROBLEM AND SOLUTION:

Analysis Phase

Because there was no discrete data available, the majority of the work assumptions and questions were established when working on the rich picture and six element analysis of the organization's operations. To comprehend the situation better and to avoid it,

There were misunderstandings, respected faculty members, and stakeholder interviews conducted.

Designing Phase

The Relational Schema design also included the retention of created entities at their Significant levels based on descriptive study. The instructor's feedback was also highly important and valid in this situation.

Implementation Phase

All the Software System Requirements (SSR's) reached successfully!
Front-End Development tools: HTML, CSS, Bootstrap JavaScript, Chart Js
Back End Development tools: PHP, XAMPP
Database-integration: SQLlite3

B. ADDITIONAL FEATURE AND FUTURE DEVELOPMENT:

Future Developing Purposes:

- Plans for the project is, to add another feature which can predict A candidate's grade based on his/her past grades and performances.
- Difficulty of the current semester can be compared with the previous semester question difficulty percentage.

REFERENCES-

[1] Independent University - Bangladesh, "Curriculum for B Sc. in Computer Science and Engineering (Version 2.2)," Independent University - Bangladesh, March 19, 2017.
[2] Independent University - Bangladesh, "www.iub.edu.bd," 2020. [Online]. Available: www.iub.edu.bd.
[3] I. Department of Computer Science and Engineering, "Department of Computer Science and Engineering, IUB," 2020. [Online]. Available: http://www.cse.iub.edu.bd/.
[4] D. o. C. S. &. S. Engineering, "The University of Western Australia," 2006. [Online]. Available: https://teaching.csse.uwa.edu.au/units/CITS3240/Lectures/db-er1-nup4.pdf.
[5] K. Stevens, "BetterEvaluation," 2020. [Online]. Available: https://www.betterevaluation.org/en/evaluation-options/richpictures.
[6] Wikipedia, "Wikipedia," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Rich_picture.
[7] Wikipedia, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation.