Logo

Description automatically generated

**Database Management**

**SPMS 4.0**

**Group 26 – Team Halfdeads**

**Submitted By**

|  |  |
| --- | --- |
| **Name** | **ID No**. |
| Tasnim Nazifa Kamal | 2010664 |
| Md. Showkat Jamil | 2010020 |
| Farzana Islam | 1910058 |
| Shumiya Tabassum | 2022018 |
| Mohammad Taznim Azad | 1910432 |

Table of Contents

[**CHAPTER-1 INTRODUCTION** 2](#_Toc133468312)

[A. BACKGROUND OF THE ORGANIZATION - IUB: 2](#_Toc133468313)

[B. BACKGROUND OF THE PROJECT - SPMS 4.0: 2](#_Toc133468314)

[C. OBJECTIVE OF THE PROJECT - SPMS 4.0: 2](#_Toc133468315)

[D. SCOPE OF THE PROJECT: 3](#_Toc133468316)

[**CHAPTER-2 REQUIREMENT ANALYSIS** 3](#_Toc133468317)

[A. RICH PICTURE – EXISTING SYSTEM (SPMS 3.0): 3](#_Toc133468318)

[FIGURE 2.1: RICH PICTURE (AS IS) 4](#_Toc133468319)

[B. SIX ELEMENT ANALYSIS – EXISTING SYSTEM (SPMS 3.0): 5](#_Toc133468320)

[C. PROBLEM ANALYSIS – EXISTING SYSTEM (SPMS 3.0): 14](#_Toc133468321)

[D. RICH PICTURE – PROPOSED SYSTEM(SPMS4.0): 15](#_Toc133468322)

[FIGURE 2.2: RICH PICTURE (TO BE) 15](#_Toc133468323)

[E. SIX ELEMENT ANALYSIS – PROPOSED SYSTEM (SPMS 4.0): 16](#_Toc133468324)

[**CHAPTER-3 LOGICAL SYSTEM DESIGN** 23](#_Toc133468325)

[A. BUSINESS RULES – SPMS4.0 23](#_Toc133468326)

[B. ENTITY RELATIONSHIP DIAGRAM: 25](#_Toc133468327)

[FIGURE 3.1: ENTITY RELATIONSHIP DIAGRAM 25](#_Toc133468328)

[C. ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA: 26](#_Toc133468329)

[FIGURE 3.2: RELATIONAL SCHEMA 26](#_Toc133468330)

[D. NORMALIZATION: 27](#_Toc133468331)

[1NF: 29](#_Toc133468332)

[2NF: 30](#_Toc133468333)

[3NF: 30](#_Toc133468334)

[E.DATA DICTIONARY 30](#_Toc133468335)

[**CHAPTER-4 PHYSICAL SYSTEM DESIGN** 41](#_Toc133468336)

[A. INPUT FORMS: 41](#_Toc133468337)

[B. OUTPUT FORMS 49](#_Toc133468338)

[**CHAPTER-5 CONCLUSION** 56](#_Toc133468339)

[A. PROBLEM AND SOLUTION 56](#_Toc133468340)

[Analysis Phase: 56](#_Toc133468341)

[Designing Phase: 56](#_Toc133468342)

[Implementation Phase: 56](#_Toc133468343)

[Additional Features and Future Development: 56](#_Toc133468344)

[References 57](#_Toc133468345)

# **CHAPTER-1 INTRODUCTION**

## BACKGROUND OF THE ORGANIZATION - IUB:

Established in 1993, Independent University, Bangladesh (IUB) is the leading private university in Bangladesh with an explicit focus on research and global partnerships. With a current enrollment of around 10,000 at undergraduate and graduate levels, more than 450 highly qualified faculty members, 47% of whom have PhD degrees, and an alumni strength of close to 14,000, IUB is trying to ensure education all to promote the national agenda of sustainable inclusive development.[1]

The academic curriculum is based on the North American Liberal Arts Model and the medium of instruction is English. There are currently five academic schools: [1]

1) School of Business & Entrepreneurship

2) School of Engineering, Technology and Sciences

3) School of Environment & Life Sciences

4) School of Liberal Arts & Social Sciences

5) School of Pharmacy and Public Health

## BACKGROUND OF THE PROJECT - SPMS 4.0:

A paradigm for outcome-based education is the Student Performance Monitoring System (SPMS 4.0). (OBE). It analyzes students, course instructor, school, department, and program performance and aids the higher authorities of the educational establishment in developing improvement strategies.

## OBJECTIVE OF THE PROJECT - SPMS 4.0:

Through a database of assessments, including quizzes, midterm and final examinations, and other tests, SPMS 4.0 keeps track of and examines the performance of its stakeholders, including students, course instructors, departments, schools, and programs. The SPMS 4.0 database contains all the information needed to evaluate the performance of the stakeholders, including all exam question papers, answer scripts, course outlines, and marks for exams and assessments in relation to the Course Outcomes (CO), Program Learning Outcomes (PLO), and Program Outcomes (PO) attained by the students. Students can therefore statistically evaluate their own success. By giving Higher Authorities access to a variety of analytical reports based on student, course instructor, department, school, and program success, SPMS 4.0 also gives them the chance to make further improvements.

## SCOPE OF THE PROJECT:

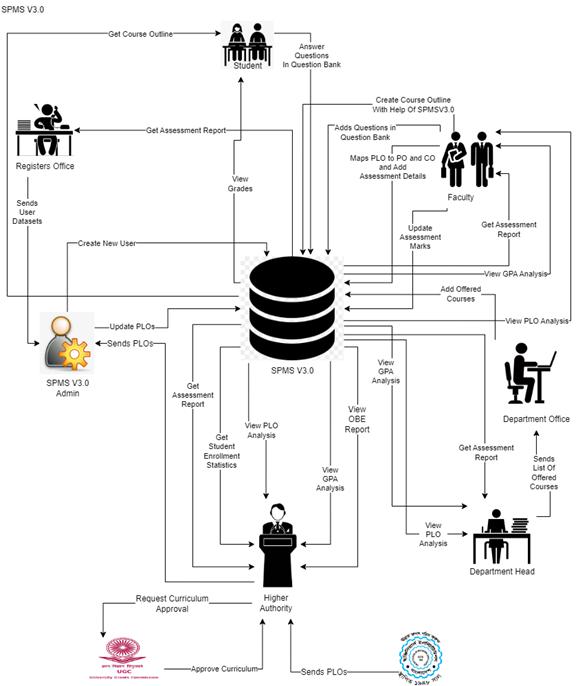
We thoroughly analyzed the current system (SPMS 3.0) and found some problems in the business processes that could make the process slow-moving, ineffective, and lead to communication breakdowns. The suggested remedy for those problems is to develop a better version of the system called SPMS 4.0 (Student Performance Monitoring System 4.0), which uses a Relational Database Management System (RDBMS) to store, update, and retrieve important documents like course outlines, exam question papers, and answer scripts, as well as other data required to monitor student performance and generate other OBE (outcome-Based Education) reports. All users of the system (SPMS 3.0) have been identified, along with information about how they will obtain the required data and information and interact with one another. We want to create user interfaces so that every user can access the data they need and use the system to create, examine, and download the reports and documents they want. Additionally, we want to create a platform that will enable faculty members to work together on creating course outlines, test questions, grade cards, and other materials. A new feature that determines the course outcome percentage based on the grades earned for each subject will also be implemented. Students and teachers will be able to view the calculated course outcome percentage through web application.

# **CHAPTER-2 REQUIREMENT ANALYSIS**

Requirement Analysis is the process of determining what the database is used for. It involves interviews with stakeholders to identify the functionality and system requirements they expect and require from the database, what operations need to be performed and what data they need to process. By doing so, we can get a proper understanding of the stakeholders and how they interact with each other.

## A. RICH PICTURE – EXISTING SYSTEM (SPMS 3.0):

A rich picture is a way to explore, acknowledge and define a situation and express it through diagrams to create a preliminary mental model and can help to open discussion and come to a broad, shared understanding of a situation.[3] A rich picture enables us to recognize connections and interactions that we might otherwise overlook. It aids in determining one or more themes that participants might wish to investigate and handle further. Rich pictures are thus always employed during the pre-analysis stage.



#### FIGURE 2.1: RICH PICTURE (AS IS)

In this rich picture the stakeholders are:

1. UGC
2. IEB
3. Higher Authority (VC, Dean etc.)
4. Department Head
5. Department Office
6. SPMSV3.0 Admin (SPMS Manager)
7. Registers Office
8. Faculty
9. Student

The Main Storage is:

1. SPMS V3.0

## B. SIX ELEMENT ANALYSIS – EXISTING SYSTEM (SPMS 3.0):

From the rich picture we can see that there are 10 key processes:

1. **Creating storing and giving Course Outline**
2. **Add Questions to the question bank and grading the answer script**
3. **Course based student performance trend according to GPA**
4. **Faculty based student performance according to GPA**
5. **Course wise PLO achievement of a student**
6. **Student performance trend under VC/Dean/Head of Department**
7. **Course, Program, department, school CLO-PLO statistics**
8. **Course, student, department school wise expected vs achieved PLO**
9. **Department average of total PLO achieved and attempted students**
10. **Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.**

We can use six element analysis to analyze the impact of six elements in a process here the six elements are:

1. Human
2. Non computing Hardware
3. Computing Hardware
4. Software.
5. Database.
6. Network and Communication.

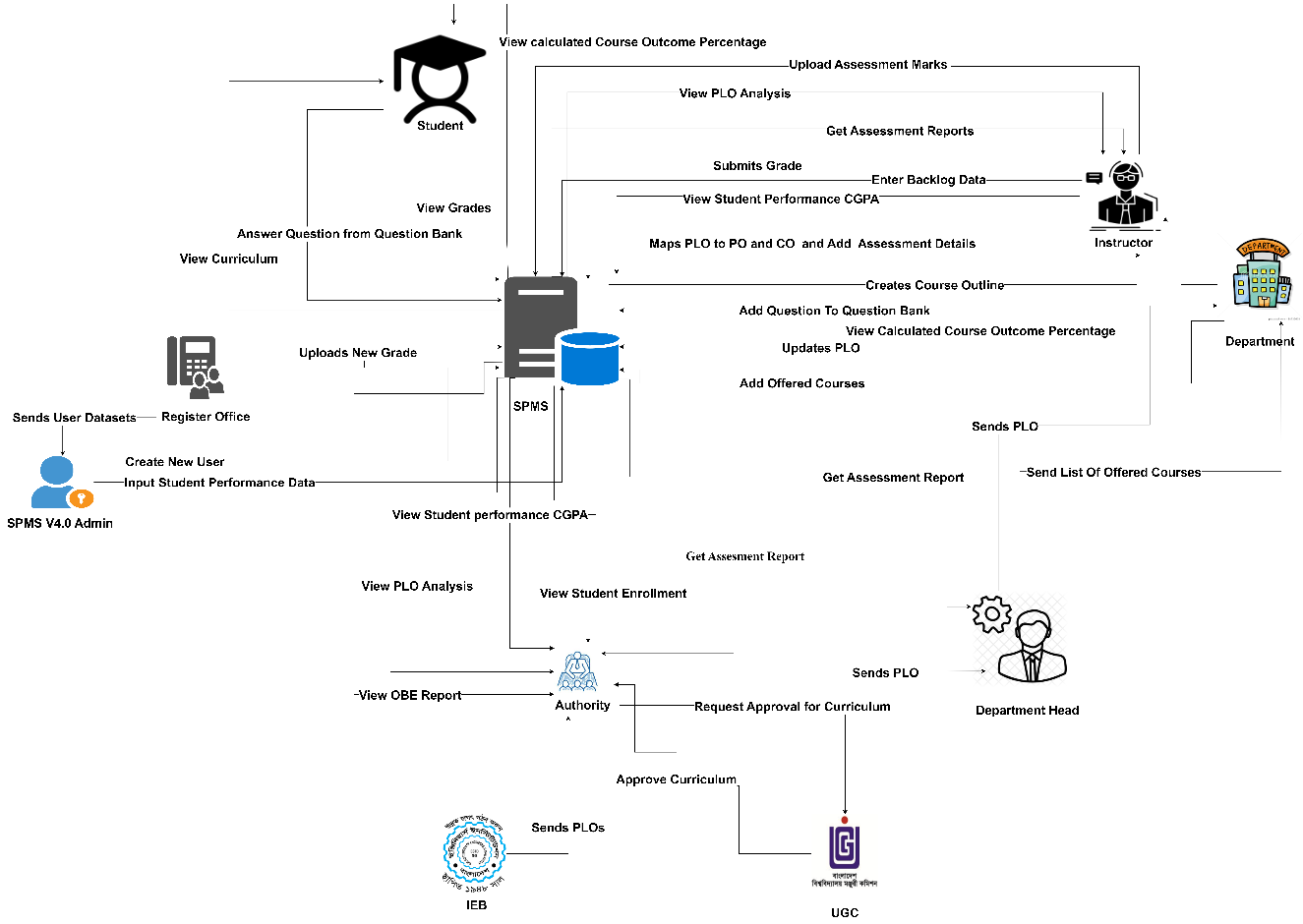
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Process** | **Human** | **Non**  **Computing Hardware** | **Computing Hardware** | **Software** | **Database** | **Network and Communication** |
| **Preparing storing and giving Course Outline** | **Faculty:**  1) Signs into System using their ID and Password.  2) Select Create Course Outline Tab.  3) Select From the options that they wish to add in their course outline.  4) Press the Create button.  5) Store course outline into system.  **Students:**  1) Signs into System using their ID and Password.  2) Select Course 3)View/Download Course Outline from System. |  | **Computer/ Laptop**  1)Used to Sign into SPMS3.0    **Printer**  1) Used to print hard copy of course outlines if required. | **SPMS2. 0**  1)Used to store Data into the database | **SPMS 2.0**  **Database**  1) All valid data are stored here which can be updated by SPMS 2.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |
| **Add Questions to the question bank and grading the answer script** | **Faculty:**  1) Signs into System using their ID and Password.  2) Select course and choose section’s that has to solve the question.  3) Input the question in the question bank. 4) Press the Assign Button. 5) Grade the answers submitted by the students.  **Student:**  1) Signs into System using their ID and Password.  2) Answer the question assigned by the faculty in the answer bank.  3) Press the Submit button.  4) Check grade in SPMS3.0 after faculty is done checking |  | **Computer/ Laptop**  1)Used to Sign into SPMS3.0    **Printer**  1) Used to print the grades gotten by the whole | **SPMS3.0**  1)Used to store Data into the database or generate result graph using data from the database | **SPMS 3.0 Database**  1)All valid data are stored here which  can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS2.0 |
| **Course based student performance trend according to GPA** | **Department Head:**  1) Signs into System using their ID and Password.  2) Input the time period and course ID to be viewed.  3) View student progress through a graph made after analysis and the GPA earned by maximum/minimum/average students.  **Faculty:**  1) Signs into system using their ID and Password.  2) Search for the course that they are teaching using course ID and time period and view the progress of that students of that course.  **Student:**  1)Signs into System using their ID and Password. 2) Search for the course using course ID and View their progress of that course and the GPA they earned.  **Dean/VC :**  1) Signs into system using their ID and Password.  2) Search for the course using course ID and time period and view the progress of the students of that course. |  | **Computer/ Laptop**  1)Used to Sign into SPMS2.0.  **Printer**  1) Used to print hard copy of the progress of current semester’s students and compare with the progress of the previous semester’s students who did that course. | **SPMS3. 0**  1)Used to store student Data into the database or generate perform ance analysis graph using data from the database | **SPMS 3.0 Database**    1) All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |
| **Faculty based student Performa-nce according to GPA** | **Faculty:**  1)Signs into system using their ID and Password. 2) View the Progress of the students who are being taught by them.  **Department Head:**  1)Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty’s name. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students.  **Dean/VC:**  1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty’s name and Department ID. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students. |  | **Computer/ Laptop**  1)Used to Sign into SPMS3.0  **Printer**  1) Used to print hard copy of the progress of students taught by a faculty. | **SPMS3. 0**  1)Used to store student Data into the database or generate perform ance analysis graph using data from the database. | **SPMS 3.0 Database**  1) All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |
| **Course wise PLO achieve ment of a student** | **VC/ Dean:**  1)Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the student.  **Department Head:**  1)Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the students.  **Faculty:**  1)Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the students in a course.  **Student:**  1) Signs into system using their ID and Password. 2) View PLOs they have achieved so far and how many they need to achieve to complete the course. |  | **Computer/ Laptop**  1)Used to Sign into SPMS3.0  **Printer**  1) Used to print hard copy of a report of students who completed most the PLO achievements If needed. | **SPMS3. 0**  1)Used to store Data and generate PLO automat ically based on the CO provided. | **SPMS 3.0 Database**  1) All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |
| **Student performa nce trend under VC/Dean/ Head of Departm ent** | **Dean** :  1) Signs into system using their ID and Password.  2) Search for Department Head to be checked using their Name and Department ID. 3) View student progress under them or them.  **VC:**  1)Signs into system using their ID and Password. 2) Search for a Dean or Department Head to be checked using their Name and either School ID or Department ID. 3) View student progress under them.  **Department Head:**  1) Signs into system using their ID and Password. 2) View student progress under them. |  | **Computer/ Laptop**  Used to Sign into SPMS3.0  **Printer**  1) Used to print the hard copy of the progress report if needed | **SPMS3. 0**  1)Used to store  Data into the database or generate perform ance analysis graph using data from the database | **SPMS 3.0 Database**  1) All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |
| **Course, Program, departm ent, school CLO-PLO statistics** | **Dean/VC :** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Department Head:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students.  **Faculty:**  1)Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students.  **Student:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by them and other students. |  | **Computer/ Laptop**  1) Used to Sign into SPMS3.0 **Printer**  1) Used to print the hard copy of the progress report if needed | **SPMS3. 0**  1)Used to store Data into the database and generate CLO-PLO statistical data or graphs. | **SPMS 3.0 Database**  1)All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet** 1)Used to Sign into SPMS3.0 |
| **Course, student, departm ent school wise expected vs achieved PLO** | **Dean/VC :**  1)Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered which is inputted and comparison between expected and achieved.    **Department Head:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved.  **Faculty:** 1) Sign into the system using ID and Password.  2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved.  **Student:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved. |  | **Computer/ Laptop**  Used to Sign into SPMS3.0    **Printer**  1) Used to print the hard copy of both the previous and current semester’s achieved PLO to compare. | **SPMS3. 0**  1)Used to store Data into the database or generate perform ance analysis graph using data from the database. | **SPMS 3.0 Database**    **1)** All valid data are stored here which can be updated by SPMS 2.0 admin s | **Internet**  1)Used to Sign into SPMS2.0 |
| **Departm ent average of total PLO achieved and attempted students** | **Dean/VC :**  1)Sign into the system using ID and Password. 2) Enter the time period of the semester wished to be viewed. 3) View the departmental average of total PLO achieved along with the number of students who attempted.  **Department Head:**  1) Sign into the system using ID and Password. 2) Enter the time period of the semester wished to be viewed. 3) View the departmental average of total PLO achieved along with the number of students who attempted.  **Faculty:**  1) Sign into the system using ID and Password. 2) View the total departmental average of the PLO achieved by the students.  **Student:**  1) Sign into the system using ID and Password. 2) View the total departmental average of the PLO achieved by the students |  | **Computer/ Laptop**  1) Used to Sign into SPMS3.0  Printer 1) Used to print the hard copy of PLO reports | **SPMS3. 0**  1)Used to store Data into the database or generate perform ance analysis graph using data from the database | **SPMS 3.0 Database**  1) All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |
| **Student Enrollme nt Statistics VC-wise, Dean-wise, Departm ent Head-wise.** | **VC:**  1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab  3) View Student Enrollment Statistics of That Year and Semester.  **Dean:**  1)Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester.  **Department Head:**  1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab  3) View Student Enrollment Statistics of That Year and Semester. |  | **Computer/ Laptop**  1) Used to Sign into SPMS3.0  **Printer**  1) Used to print the hard copy of Student Enrollment Statistics If Needed. | **SPMS3. 0**  1)Used to store Data into the database and generate Student Enrollm ent Statistics graphs. | **SPMS 3.0 Database**  1) All valid data are stored here which can be updated by SPMS 3.0 admins. | **Internet**  1)Used to Sign into SPMS3.0 |

## C. PROBLEM ANALYSIS – EXISTING SYSTEM (SPMS 3.0):

The problems in SPMS3.0 were analyzed, and the proposed solution are given in the following.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Process Name** | **Stake Holders** | **Concerns(Problems)** | **Analysis(Reason of the problem)** | **Proposed Solution** |
| **Course, Program, department, school CLO-PLO statistics** | **Dean/VC,**  **Department Head, Faculty,**  **Student** | Calculating Course Outcome based on student grades. | 1)Unreliable storage  2)Change of pattern in student Grading | We can add new data fields to the existing web application to allow the calculation of course.  outcomes based on student Grade. The user will be able to input the data using a manual form and also can import a csv file from which the data points can be extracted and inputted into the database. Specifically, students and faculty can view the calculated course outcome percentage  through the web application. |
| **Enter Backlog Data** | **Faculty, Student** | Entering Course grade of previous students | 1)Storing Data from Past  2)Data Security | Faculties can add grade of previous semester which will be stored in the DBMS. Students can see their respective grades and CO percentage from those grades. |

## RICH PICTURE – PROPOSED SYSTEM(SPMS4.0):



#### FIGURE 2.2: RICH PICTURE (TO BE)

**In this rich picture the stakeholders are:**

1) UGC

2) IEB

3) Higher Authority (VC, Dean etc.)

4) Department Head

5) Department Office

6) SPMSV4.0 Admin (SPMS Manager)

7) Registers Office

8) Faculty

9) Student

**The Main Storage is**

1. SPMS V4.

## SIX ELEMENT ANALYSIS – PROPOSED SYSTEM (SPMS 4.0):

From the rich picture we can see that there are 10 key processes:

1. **Course based student performance trend according to GPA.**
2. **Instructor based student performance according to GPA**
3. **Course wise PLO achievement of a student**
4. **Student performance trend under VC/Dean/Head of Department**
5. **Course, Program, department, school CLO-PLO statistics**
6. **Course, student, department school wise expected vs achieved PLO.**
7. **Department average of total PLO achieved and attempted students.**
8. **Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise**
9. **Add Questions to the question bank and grading the answer script.**
10. **Preparing storing and giving Course Outline**
11. **Calculated Course Outcome Percentage from Backlog Data Student-wise, Faculty-wise.**

We can use six element analysis to analyze the impact of six elements in a process here the six elements are:

1. Human

2. Non computing Hardware

3. Computing Hardware

4. Software.

5. Database.

6. Network and Communication.

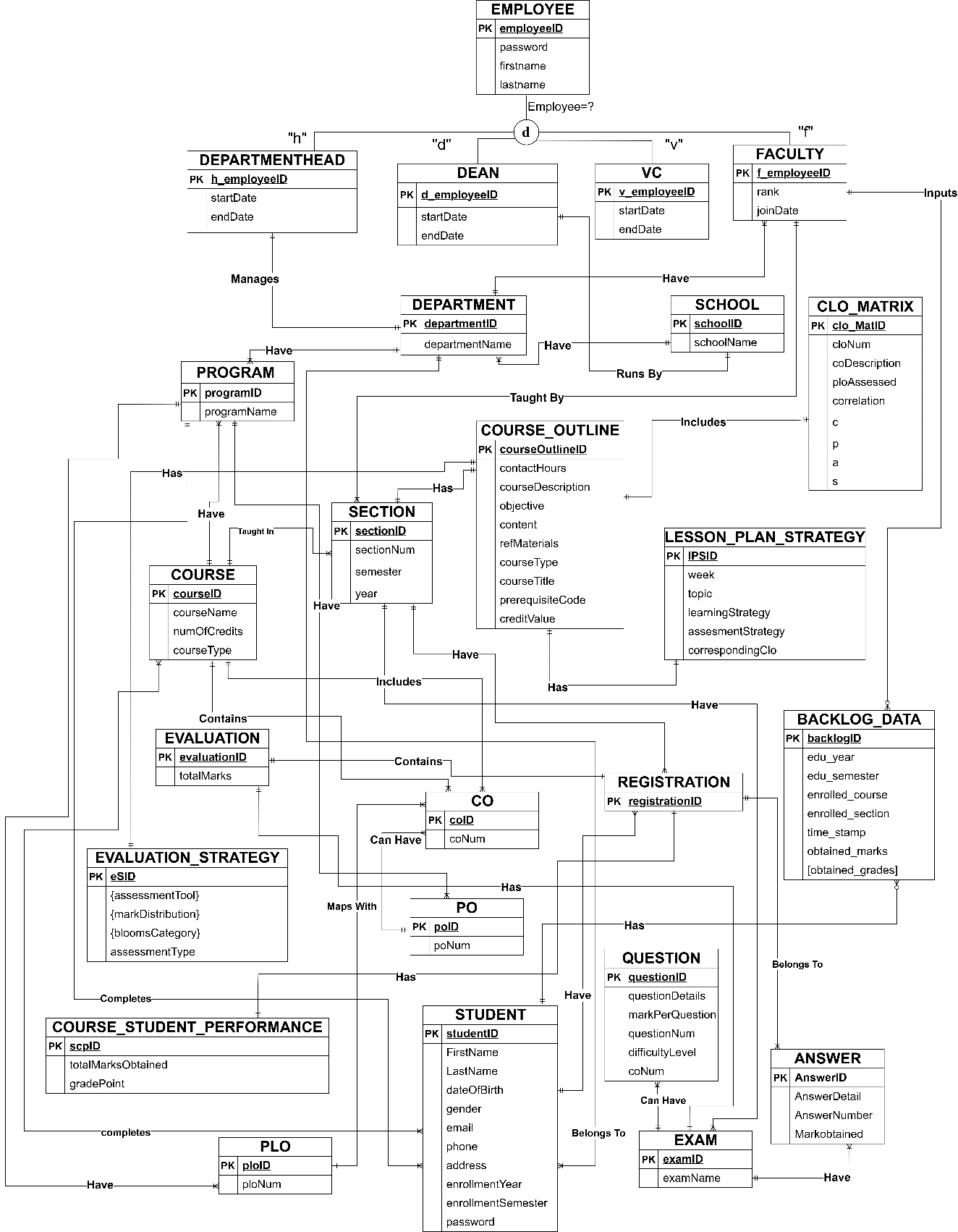
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Process** | **Human** | **Non-computing**  **Hardware** | **Computing**  **Hardware** | **Software** | **Database** | **Network and Communication** |
| **Course based student performance trend according to GPA** | **Department Head:**  1)Signs into System using their ID and Password. 2) Input the time period and course ID to be viewed. 3) View student progress through a graph made after analysis and the GPA earned by maximum/minimum/average students.  **Faculty:**  1)Signs into system using their ID and Password. 2) Search for the course that they are teaching using course ID and time period and view the progress of that students of that course.  **Student:**  1) Signs into System using their ID and Password. 2) Search for the course using course ID and View their progress of that course and the GPA they earned.  **Dean/VC:**  1) Signs into system using their ID and Password. |  | **Computer/Laptop:**  1) Used to Sign into SPMS 4.0.  **Printer / Scanning Machine:**  1) Used to print hard copies of the progress report of current and previous semesters of students. | **SPMS4.0:**  1)Used to store student  Data into the database or generate performance analysis graph using data from the database | **SPMS4.0 Database:**  1) All valid data are stored here which can be updated by SPMS 4.0 admins. | **Internet:**  1)Used to Sign into SPMS4.0 |
| **Faculty based student performance according to GPA** | **Faculty:** 1) Signs into system using their ID and Password. 2) View the Progress of the students who are being taught by them.  **Department Head:** 1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty’s name. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students. **Dean/VC:** 1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty’s name and Department ID. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students. |  | **Computer/Laptop:**  1) Used to Sign into SPMS 4.0.  **Printer / Scanning Machine:**  1) Used to print hard copies of the progress report of current and previous semesters of students. | **SPMS4.0:**  1)Used to store student  Data into the database or generate performance analysis graph using data from the database | **SPMS4.0 Database:**  1) All valid data are stored here which can be updated by SPMS 4.0 admins. | **Internet:**  1)Used to Sign into SPMS4.0 |
| **Course wise PLO achievement of a student** | **VC:**  1) Signs into system using their ID and Password.  2) Select PLO achievement Tab and search using Course ID  3) View PLOs achieved by the student.  **Department Head:**  1) Signs into system using their ID and Password.  2) Select PLO achievement Tab and search using Course ID  3) View PLOs achieved by the students.  **Faculty:**  1) Signs into system  using their ID and Password.  2) Select PLO achievement Tab and search using Course ID  3) View PLOs achieved by the students in a course.  **Student:**  1) Signs into system using their ID and Password.  2) View PLOs they have achieved so far and how many they need to achieve to complete the course. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0  **Printer**  1) Used to print hard copy of a report of students who completed most the PLO achievements  If needed. | **SPMS4.0**  1)Used to store  Data and generate PLO automatically based on the CO provided | **SPMS4.0 Database**  1) All valid data are stored here which can be updated by SPMS4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Student performance trend under VC/Dean/Head of Department** | **VC:**  1) Signs into system using their ID and Password.  2) Search for Dean or Department Head to be checked using their Name and either School ID or Department ID.  3) View student progress under them or them.  **Department Head:**  1) Signs into system using their ID and Password.  2) View student progress under them.  **Dean :**  1) Signs into system using their ID and Password.  2) Search for Department Head to be checked using their Name and Department ID. 3) View student progress under them or them. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0  **Printer**  1) Used to print the hard copy of the progress report if needed. | **SPMS4.0**  1)Used to store  Data into the database or generate performance analysis graph using data from the database. | **SPMS4.0 Database**  1) All valid data are stored here which can be updated by SPMS4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Course, Program, department, school CLO-PLO statistics** | **Dean/VC:**  1) Signs into system using their ID and Password.  2) View CLO-PLO mapped statistics achieved by students.  **Department Head:**  1) Signs into system  using their ID and Password.  2) View CLO-PLO mapped statistics achieved by students.  **Faculty:**  1) Signs into system using their ID and Password.  2) View CLO-PLO mapped statistics achieved by students.  **Student:**  1) Signs into system using their ID and Password.  2) View CLO-PLO mapped statistics achieved by them and other students. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0  **Printer**  1) Used to print the hard copy of the progress report if needed. | **SPMS4.0**  1)Used to store  Data into the database and generate CLO-PLO statistical data or graphs. | **SPMS4.0 Database**  1) All valid data are stored here which can be updated by SPMS4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Course, student, department school wise expected vs achieved PLO** | **VC/Dean:**  1) Sign into the system using ID and Password.  2) View the achieved PLO of the students during time entered  that has been inputted and comparison between expected and achieved.  **Department Head:**  1) Sign into the system using ID and Password.  2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved.  **Instructor:**  1) Sign into the system using ID and Password.  2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected  and achieved.  **Student:**  1) Sign into the system using ID and Password.  2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0  **Printer**  1) Used to print the hard copy of  both the previous and current semester’s achieved PLO to compare. | **SPMS4.0**  1)Used to store  Data into the database or generate performance analysis  graph using data from the database | **SPMS4.0 Database**  1) All valid data are stored here which can be updated by  SPMS4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Department average of total PLO achieved and attempted students** | **Dean/VC:**  1) Sign into the system using ID and Password.  2) Enter the time period of the semester wished to be viewed.  3) View the departmental average of total PLO achieved along with the number of students who attempted.  **Department Head:**  1) Sign into the system using ID  and Password.  2) Enter the time period of the semester wished to be viewed.  3) View the departmental average of total PLO achieved along with the number of students who attempted.  **Faculty:**  1) Sign into the system using ID and Password.  2) View the total departmental average of the PLO achieved by the students.  **Student:**  1) Sign into the system using ID and Password.  2) View the total departmental average of the PLO achieved by the students |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0  **Printer**  1) Used to print the hard copy of  PLO reports | **SPMS4.0**  1)Used to store  Data into the database or generate performance analysis graph using data from the database | **SPMS4.0 Database**  1) All valid data are stored here which can be updated by SPMS4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Add Questions to the question bank and grading the answer script** | **Faculty:**  1)Signs into System using their ID and Password. 2) Select course and choose section’s that has to solve the question. 3) Input the question in the question bank. 4) Press the Assign Button. 4) Grade the answers submitted by the students.  **Student:**  1) Signs into System using their ID and Password. 2) Answer the question assigned by the faculty in the answer bank 3) Press the Submit button 4) Check grade in SPMS4.0 after faculty is done | **Pen and Paper**  1)Can be Used to create  Rough assessment questions.  2) Used to answer assessment questions. | **Computer/ Laptop**  1)Used to Sign into SPMS4.0  **Printer**  1) Used to print the grades gotten by the whole section | **SPMS4. 0**  1)Used to store Data into the database or generate result graph using data from the database | **Physical Storage**  1)Used for  Storing all the answer scripts returned by students | **Internet**  1) Used to Sign into Google  Classroom if communication is required.  2)Used to Sign into SPMS4.0 |
| **Student Enrollment Statistics Authority wise, Department**  **Head-wise**. | **VC**  1)Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester.  **Dean**  1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics Of That Year and Semester. **Department Head** 1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0  **Printer**  1) Used to print the hard copy of  Student Enrollment Statistics If Needed. | **SPMS4.0**  1)Used to store  Data into the database and generate Student Enrollment Statistics graphs. | **SPMS4.0 Database**  1) All valid data are stored here which can be updated by SPMS4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Preparing and storing Course Outline** | **Instructor:**  1)Signs into System using their ID and Password. 2) Select Create Course Outline Tab. 3) Select From the options that they wish to add in their course outline. 4) Press the Create button. 5) Store course outline into system.  **Students:** 1) Signs into System using their ID and Password. 2) Select Course 3)View/Download Course Outline from System. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0 **Printer**  1) Used to print hard copy of course outlines if required | **SPMS4. 0**  1)Used to store Data into the database | **SPMS4.0Database**    1) All valid data are stored here which can be updated by SPMS 4.0 admins. | **Internet**  1)Used to Sign into SPMS4.0 |
| **Calculated Course Outcome Percentage From Backlog Data Student-wise, Faculty-wise.** | **Faculty:**  1)Signs into System using their ID and Password. 2) Select Calculated course outcome percentage.  3) Select Educational year, Educational semester, Enrolled course from the options.  **Student:**  1)Signs into System using their ID and Password. 2) Select Calculated course outcome percentage.  3) Select Educational year, Educational semester, Enrolled course, Enrolled section from the options.  **SPMS4.0 Admin:**  1)Signs into System using their ID and Password.  2) Input previous student data.  3) Select calculate course outcome percentage.  3) Input required data. Press “Convert CO from Grade”. Press “SAVE” to save the new data. |  | **Computer/Laptop**  1) Used to Sign into SPMS4.0 **Printer**  1) Used to print hard copy of calculated CO | **SPMS4. 0**  1)Used to store Data into the database | **SPMS4.0Database**    1) All valid data are stored here which can be updated by SPMS 4.0 admins | **Internet**  1)Used to Sign into SPMS4.0 |

# **CHAPTER-3 LOGICAL SYSTEM DESIGN**

## BUSINESS RULES – SPMS4.0

1. A student must have one department. A STUDENT has studentID, FirstName, LastName, dateOfBirth, gender, email, phone, address, departmentID, programID, enrollmentYear, enrollmentSemester, password. A department must have one or many Students.
2. Student may perform many registrations. REGISTRATION includes registrationID, sectionID, studentID. A registration must be performed by at least one student.
3. A section mandatorily have many registrations. A registration has at least one section. A section includes sectionID, sectionNum, courseID, facultyID, semester, year.
4. A registration may belong to many EVALUATIONS. An evaluation must belong to one registration. An EVALUATION contains evaluationID, examID, registrationID, totalMarks.
5. A CO must map with one PLO. A PLO’s must map with one or many CO’s. PLO includes ploID, ploNum, 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 course.
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 four sub-type( Dean, Department Head, Faculty, VC). An EMPLOYEE includes employeeID,password, firstName, lastName.
10. A school must be run by exactly one. A dean must run exactly one school. A DEAN has schoolID, startDate, endDate.
11. A Department must be run by exactly one Department head. A department head must manage exactly one department. A DEPARTMENTHEAD includes departmentID, startDate, endDate.
12. A Faculty must have exactly one Department. A department must have one or many Faculties. A FACULTY includes departmentID, rank, joinDate. A faculty may teach many sections. A section must be taught by exactly one faculty
13. A courseOutline 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,creditValue,prerequisiteCode.
14. 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.
15. 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 lPSID ,week ,topic ,learningStrategy , assessmentStrategy, correspondingClo, courseOutlineID.
16. An exam has exactly one evaluation. An Evaluation for an exam is done exactly once. An exam belongs to exactly one section. An EXAM includes examID, examName, sectionID. A section must have one or many exams.
17. An exam must have one or many questions. Every question must belong to exactly one exam. A QUESTION includes questionID , questionDetails , marksPerQuestion , questionNum , difficutltyLevel , examID , coNum. A Question is answered exactly once. An answer has exactly one question.
18. A PO belongs to exactly one program A program must have one or many PO.PO includes poID , poNum , programID. A PO must belong to one or many CO. A CO must have exactly one PO.
19. A student course performance evaluation is done for registration exactly once. A registration has student course performance done exactly once. A registration has only one evaluation. An Evaluation has exactly one registration. registrationID, scpID, gradePoint ,totalMarksObtained included in COURSE\_STUDENT\_PERFORMACE.
20. A CO belongs to exactly one course. A course must have one or many CO. CO includes coID, coNum, courseID, ploID, poID, Student ID.
21. A Backlog\_data belongs to exactly one student. A student can have multiple Backlog Data. BACKLOG\_DATA includes backlogID,edu\_year,edu\_semester, enrolled\_course,enrolled\_section,time\_stamp,obtained\_marks, obtained\_grades.

## ENTITY RELATIONSHIP DIAGRAM:



#### FIGURE 3.1: ENTITY RELATIONSHIP DIAGRAM

## ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA:

Chart

Description automatically generated

#### FIGURE 3.2: RELATIONAL SCHEMA

## NORMALIZATION:

|  |  |  |
| --- | --- | --- |
| **EMPLOYEE(e)** | employeeID | e1 |
|  | password | e2 |
|  | firstName | e3 |
|  | lastName | e4 |
| **VC(v)** | v\_employeeID | v1 |
|  | startDate | v2 |
|  | endDate | v3 |
| **DEAN(w)** | d\_employeeID | w1 |
|  | schoolID | h1 |
|  | startDate | w2 |
|  | endDate | w3 |
| **FACULTY(f)** | f\_employeeID | f1 |
|  | departmentID | d1 |
|  | rank | f2 |
|  | joinDate | f3 |
| **COURSE\_OUTLINE (c)** | courseOutlineID | c1 |
|  | sectionID | y1 |
|  | contactHours | c2 |
|  | courseDescription | c3 |
|  | objective | c4 |
|  | content | c5 |
|  | refMaterials | c6 |
|  | courseType | c7 |
|  | courseTitle | c8 |
|  | prerequsiteCode | c9 |
|  | creditValue | c10 |
| **DEPARTMENTHEAD(k)** | h\_employeeID | k1 |
|  | departmentID | d1 |
|  | startDate | k2 |
|  | endDate | k3 |
| **DEPARTMENT(d)** | departmentID | d1 |
|  | departmentName | d2 |
|  | schoolID | h1 |
| **COURSE(u)** | courseID | u1 |
|  | courseName | u2 |
|  | numOfCredits | u3 |
|  | courseType | u4 |
|  | programID | r1 |
| **PROGRAM (r)** | programID | r1 |
|  | programName | r2 |
|  | departmentID | d1 |
| **PO (x)** | poID | x1 |
|  | poNum | x2 |
|  | programID | r1 |
| **QUESTION(q)** | questionID | q1 |
|  | questionDetails | q2 |
|  | markPerQuestion | q3 |
|  | questionNum | q4 |
|  | difficultyLevel | q5 |
|  | examID | e1 |
|  | courseID | u1 |
|  | coNum | q6 |
| **REGISTRATION (g)** | registrationID | g1 |
|  | sectionID | y1 |
|  | studentID | s1 |
| **SECTION(y)** | sectionID | y1 |
|  | sectionNum | y2 |
|  | courseID | u1 |
|  | facultyID | f1 |
|  | semester | y3 |
|  | year | y4 |
| **STUDENT(s)** | studentID | s1 |
|  | firstName | s2 |
|  | lastName | s3 |
|  | dateOfBirth | s4 |
|  | gender | s5 |
|  | email | s6 |
|  | phone | s7 |
|  | address | s8 |
|  | departmentID | d1 |
|  | programID | r1 |
|  | enrollmentSemester | s9 |
|  | enrollmentYear | s10 |
|  | password | s11 |
| **SCHOOL (h)** | schoolID | h1 |
|  | schoolName | h2 |
| **CLO\_MATRIX (m)** | clo\_MatID | m1 |
|  | cloNum | m2 |
|  | coDescription | m3 |
|  | ploAssessed | m4 |
|  | correlation | m5 |
|  | courseOutlineID | c1 |
|  | c | m6 |
|  | p | m7 |
|  | a | m8 |
|  | s | m9 |
| **PLO (p)** | ploID | p1 |
|  | ploNum | p2 |
|  | programID | r1 |
| **CO (o)** | coID | o1 |
|  | coNum | o2 |
|  | courseID | u1 |
|  | ploID | p1 |
|  | poID | x1 |
| **ANSWER (a)** | answerID | a1 |
|  | answerDetails | a2 |
|  | answerNum | a3 |
|  | markObtained | a4 |
|  | registrationID | g1 |
|  | examID | e1 |
| **EVALUATION\_STRATEGY (t)** | eSID | t1 |
|  | assessmentType | t2 |
|  | courseOutlineID | c1 |
| **ASSESSMENT\_TOOL(at)** | asID | at1 |
|  | eSID | t1 |
| **MARK\_DISTRIBUTION(j)** | mdID | j1 |
|  | eSID | t1 |
| **BLOOMS\_CATAGORY(b)** | bcID | b1 |
|  | eSID | t1 |
| **EVALUTION (n)** | evaluationID | n1 |
|  | examID | e1 |
|  | registrationID | g1 |
|  | totalMarks | n2 |
| **EXAM (i)** | examID | i1 |
|  | examName | i2 |
|  | sectionID | y1 |
| **LESSON\_PLAN\_STRATEGY (l)** | IPSID | l1 |
|  | week | l2 |
|  | topic | l3 |
|  | learningStrategy | l4 |
|  | assessmentStrategy | l5 |
|  | correspondingClo | l6 |
|  | courseOutlineID | c1 |
| **STUDENT\_COURSE\_PERFORMANCE(z)** | scpID | z1 |
|  | registrationID | g1 |
|  | totalMarksObtained | z2 |
|  | gradePoint | z3 |
| **BACKLOG(ba)** | backlogID | ba1 |
|  | studentID | s1 |
|  | enrolled\_course | ba2 |
|  | enrolled\_section | ba3 |
|  | edu\_semester | ba4 |
|  | edu\_year | ba5 |
|  | obtained\_marks | ba6 |
|  | time\_stamp | ba7 |
|  | f\_employeeID | f1 |

### 1NF:

1. There are no repeating groups
2. There is at least one primary key

Table

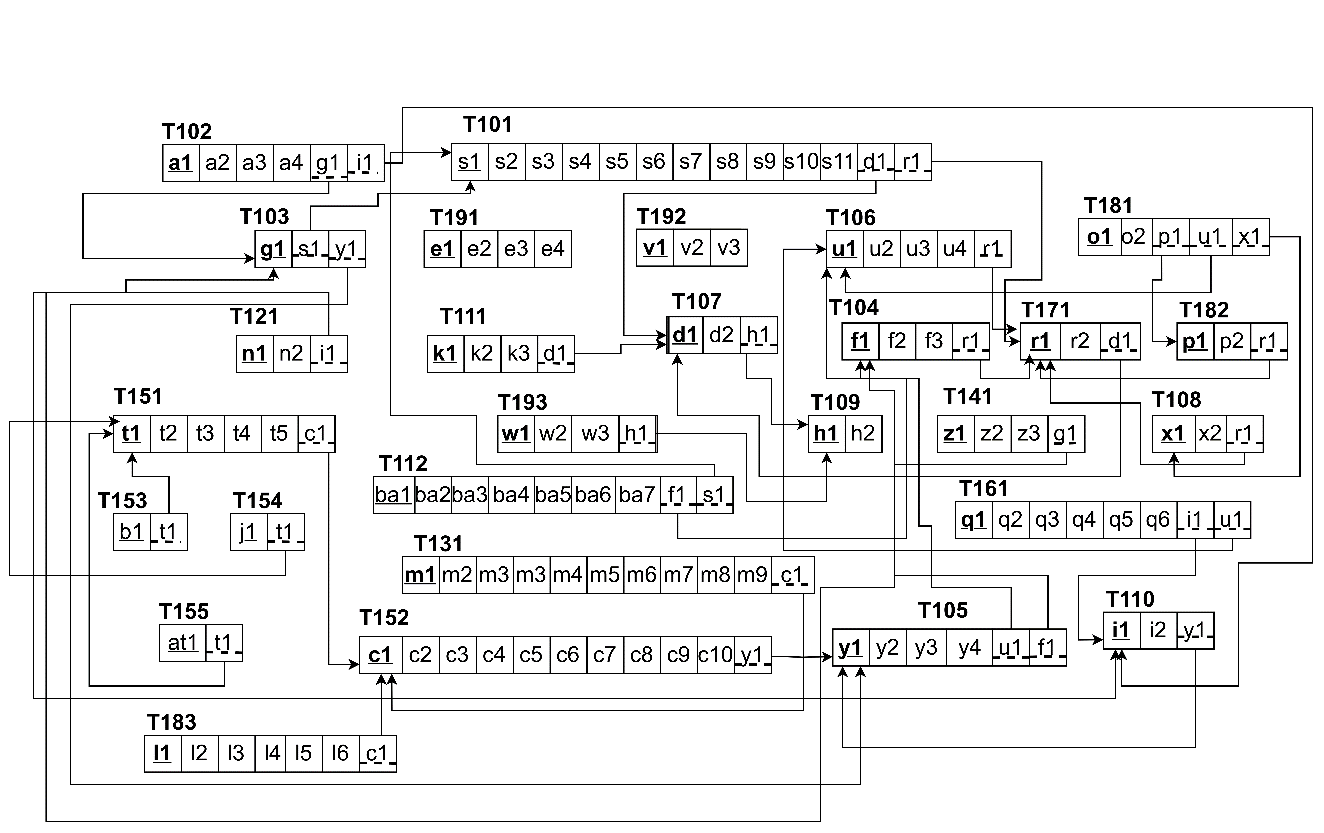
Description automatically generated

### 2NF:

1)Partial dependency has been removed

### 3NF:

1)Has no transitive dependencies



## E.DATA DICTIONARY

VC\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| v\_employeeID | INTEGER | 11 | This is the foreign key from the Employee table. E.g: “4250” |
| startDate | DATE |  | This is starting date for the VC. E.g: “01-03-2020” |
| endDate | DATE |  | This is the date VC retire from his post. E.g:“01-03-2024” |

STUDENT\_COURSE\_PERFORMANCE\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| scpID | INTEGER | 11 | This is the primary key for this table |
| registrationID | INTEGER | 11 | This is the foreign Key from registration table |
| totalMarksObtained | INTEGER | 11 | This is the total marks obtained by the student |
| gradePoint | FLOAT |  | This is the grade point achieved by the student |
| obtainedGrade | VARCHAR | 24 | This is the obtain grade achieved by the student. |

SCHOOL\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| schoolID | VARCHAR | 5 | This is the primary key of School. E.g. “SETS” |
| schoolName | VARCHAR | 50 | This is the name of the School. E.g: “School of Engineering, Technology & School |

STUDENT\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| studentID | INTEGER | 11 | This is the primary key for the Student table. E.g: “2010664”. |
| firstName | VARCHAR | 30 | This is the first name of the student. E.g: “Tasnim”. |
| lastName | VARCHAR | 30 | This is the last name of the student. E.g: “Nazifa”. |
| dateOfBirth | DATE |  | This is the birth date of the student. E.g: “24- 10-2001”. |
| gender | VARCHAR | 6 | This is the gender of the student. E.g: “Female”. |
| email | VARCHAR | 30 | This is the email of the student.  E.g:“2010[664@iub.edu.bd](mailto:664@iub.edu.bd)” |
| phone | VARCHAR | 11 | This is the phone number of the student.  E.g: “01XXXXXXXXX”. |
| address | VARCHAR | 50 | This is the address of the student. E.g: “House 6,Road2 ,Block A, Bashundhara RA |
| departmentID | VARCHAR | 3 | This is the foreign key from the  Department table. E.g: “CSE” |
| programID | INTEGER | 11 | This is the foreign key from the Program table.  E.g: “1” |
| enrollmentSemester | VARCHAR | 10 | This is the enrollment  semester of the student. |
| enrollmentYear | VARCHAR | 4 | This is enrollment year of the student. |

SECTION\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| sectionID | INTEGER | 11 | This is the Primary Key for Section.  E.g: “1” |
| sectionNum | INTEGER | 11 | This is the section number.  E.g: “1” |
| semester | VARCHAR | 6 | This is the semester of the section.E.g: “spring” |
| year | YEAR | 4 | This is the year of the semester when this section was taken.E.g: “2020” |
| courseID | VARCHAR | 6 | This is the foreign key from the Course table.  E.g: “CSE101” |
| facultyID | INTEGER | 11 | This is the foreign key from Faculty  table. E.g: “1801” |

LESSON\_PLAN\_STRATEGY\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| lpsID | INTEGER | 11 | This is the primary key of the table |
| week | INTEGER | 11 | This is the week number |
| topic | MEDIUMTEXT |  | This is the topic name |
| learningStrategy | MEDIUMTEXT |  | This is the lesson plan strategy of  that topic |
| assessmentStrategy | VARCHAR | 10 | This is the assessment  strategy of that topic |
| courseOutlineID | INTEGER | 11 | This is the foreign key from course  outline table |

REGISTRATION\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| registrationID | INTEGER | 11 | This is the Primary Key for Registration. E.g: “0101010101” |
| sectionID | INTEGER | 11 | This is the foreign key from section table |
| studentID | INTEGER | 11 | This is the foreign  key from student table |

QUESTION\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| questionID | INTEGER | 11 | This is the primary key of this table |
| questionDetails | MEDIUMTEXT |  | This is the question. |
| markPerQuestion | INTEGER | 11 | This is the mark  each question contains |
| questionNum | INTEGER | 11 | This is the number of the question |
| difficultyLevel | INTEGER | 11 | This is the difficulty level of the question |
| examID | VARCHAR | 20 | This is the foreign key from exam table |
| courseID | VARCHAR | 6 | This is the foreign key from course table |
| coNum | INTEGER | 11 | This is the CO number of the question |

PROGRAM\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| programID | INTEGER | 11 | This is the primary Key for a program. E.g: “1” |
| programName | VARCHAR | 50 | This is the name of the program. E.g: “Bachelor of Science” |
| departmentID | VARCHAR | 3 | This is the foreign key from the Department table. E.g: “CSE” |

PO\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| poID | VARCHAR | 5 | This is the primary key for Program Outcome. E.g: “PO1” |
| poNum | INTEGER | 11 | This is the PO number. E.g: “1” |
| programID | INTEGER | 11 | This is a foreign key from Program table.E.g: “1” |

PLO\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| ploID | INTEGER | 11 | This is the primary key for Program Learning Outcome. E.g: “PLO1” |
| ploNum | INTEGER | 11 | This is the PLO number. E.g: “1” |
| programID | INTEGER | 11 | This is a foreign key  from Program table. E.g: “1” |

EXAM\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| examID | INTEGER | 11 | This is the primary key for this table |
| examName | VARCHAR | 30 | This is the name of the exam |
| sectionID | INTEGER | 11 | This is the foreign key from exam  table |

FACULTY\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| f\_employeeID | INTEGER | 11 | This is the foreign key from the  Employee table. E.g: “4250” |
| departmentID | VARCHAR | 3 | This is the DepartmentID of the department faculty belongs to. E.g:  “CSE” |
| rank | VARCHAR | 30 | This is the rank of the faculty. E.g:  “Assistant Professor” |
| joinDate | DATE |  | This is starting date. E.g: “01-03-2020” |

EMPLOYEE\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| employeeID | INTEGER | 11 | This is the primary key for Employee table. E.g: “1801” |
| password | VARCHAR | 10 | This is the password of the employee |
| firstName | VARCHAR | 50 | This is the last name of the faculty.  E.g: “Ahmed” |
| lastName | VARCHAR | 50 | This is the last  name of the faculty. E.g: “Ahmed” |

DEPARTMENT\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| departmentID | VARCHAR | 3 | This is the primary key for the Department table. E.g: “CSE” |
| departmentName | VARCHAR | 50 | 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”. |

EVALUATION\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| evaluationID | INTEGER | 11 | This is the primary key for this table |
| examID | VARCHAR | 20 | This is the foreign  key from exam table |
| registrationID | INTEGER | 11 | This is the foreign key from registration table |
| totalMarks | INTEGER | 11 | This is the total marks achieved by the student in a  specific exam |

EVALUATION\_STRATEGY\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| eSID | INTEGER | 11 | This is the primary key for this table |
| courseOutlineID | INTEGER | 11 | This is the foreign key from course  outline table |
| assesmentType | VARCHAR | 30 | This is the type of assessment done for evaluation. |

ASSESSMENT\_TOOL\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| asID | INTEGER | 11 | This is the primary key for this table |
| eSID | INTEGER | 11 | This is the foreign  key from evaluation strategy table |

MARK\_DISTRIBUTION\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| mdID | INTEGER | 11 | This is the primary key for this table |
| eSID | INTEGER | 11 | This is the foreign key from evaluation strategy table |

BLOOMS\_CATEGORY\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| bcID | INTEGER | 11 | This is the primary key for this table |
| eSID | INTEGER | 11 | This is the foreign key from  evaluation strategy table |

DEPARTMENTHEAD\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| h\_employeeID | INTEGER | 11 | This is the foreign key from the  Employee table. E.g: “4250” |
| departmentID | VARCHAR | 3 | This is the DepartmentID of the department HEAD manages.  E.g: “CSE” |
| startDate | DATE |  | This is starting date. E.g: “01-03 2020” |
| endDate | DATE |  | This is the date HEAD retire from his post. E.g: “01-03-2024” |

BACKLOG\_DATA\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| backlogID | INTEGER | 11 | This is the primary key for this table |
| f\_employeeID | INTEGER | 11 | This is the foreign key from faculty table |
| studentID | INTEGER | 11 | This is the foreign  key from student table |
| enrolled\_course | CHAR | 6 | This is the course code of the Course. E.g: “CSE201” |
| enrolled\_section | INTEGER | 11 | This is the section number.  E.g: “1” |
| edu\_semester | CHAR | 6 | This is the semester of the section. E.g: “Summer” |
| edu\_year | YEAR | 4 | This is the year of the specific section of the specific course which was taken by a specific faculty |
| obtained\_marks | DECIMAL | (4,1) | This is the obtained number for a student. |
| timestamp | TIMESTAMP |  | This is the timestamp for backlog data.Eg:”21-12-2022  11:45:28” |

DEAN\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| d\_employeeID | INTEGER | 11 | 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 |  | This is starting date.  E.g: “01-03-2020” |
| endDate | DATE |  | This is the date DEAN retire from his post. E.g: “01-  03-2024” |

COURSE\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| courseID | VARCHAR | 6 | This is the Primary Key for the Course. E.g: “CSE203” |
| courseName | VARCHAR | 40 | This is the name of the Course. E.g: “Discreet Mathematics” |
| numOfCredits | INTEGER | 11 | This is the number of credits for the Course. E.g: “3” |
| courseType | VARCHAR | 10 | This is the type of the Course. E.g: “Core” |
| programID | INTEGER | 11 | This is the foreign key from the program table.  E.g: “1” |

COURSE\_OUTLINE\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| courseOutlineID | INTEGER | 11 | This is the primary key for this table |
| sectionID | INTEGER | 11 | This is the foreign  key from the section table |
| courseDescription | MEDIUMTEXT |  | This is the description of the  course |
| objective | MEDIUMTEXT |  | This is the objective of the course |
| content | MEDIUMTEXT |  | This is the content  of the course |
| refMaterials | MEDIUMTEXT |  | This is the reference material |
| courseTitle | VARCHAR | 1000 | This is the title of  the course |
| prerequsiteCode | VARCHAR | 6 | This is the  prerequisite course code |
| creditValue | INTEGER | 11 | This is the credit value of the course |

CO\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| coID | INTEGER | 11 | This is the primary key for the CO table. E.g: “CO1”. |
| coNum | INTEGER | 11 | This is the CO number.  E.g: 1,2 etc. |
| courseID | VARCHAR | 6 | 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” |
| poID | VARCHAR | 6 | This is the foreign key from the PLO table. E.g: “PO1” |

CLO\_MATRIX\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| clo\_MatID | INTEGER | 11 | This is the primary key for this table |
| cloNum | INTEGER | 11 | This is the clo number |
| coDescription | MEDIUMTEXT |  | This is the co description |
| ploAssessed | VARCHAR | 10 | This is the name of the plo assessed |
| correlation | INTEGER | 11 | This is the correlation value or  number |
| courseOutlineID | INTEGER | 11 | This is the foreign  key from the course outline table |
| c | INTEGER | 11 | This is the bloom’s category level |
| p | INTEGER | 11 | This is the bloom’s category level |
| a | INTEGER | 11 | This is the bloom’s category level |
| s | INTEGER | 11 | This is the bloom’s category level |

ANSWER\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| answerID | INTEGER | 11 | This is the primary key for this table |
| answerDetails | MEDIUMTEXT |  | This is the answer details |
| answerNum | INTEGER | 11 | This is the number of the answer |
| markObtained | INTEGER | 11 | This is the mark obtained by the  student for each answer |
| registrationID | INTEGER | 11 | This is the foreign key from  registration table |
| examID | INTEGER | 11 | This is the foreign  key from the exam table |

# **CHAPTER-4 PHYSICAL SYSTEM DESIGN**

## INPUT FORMS:

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

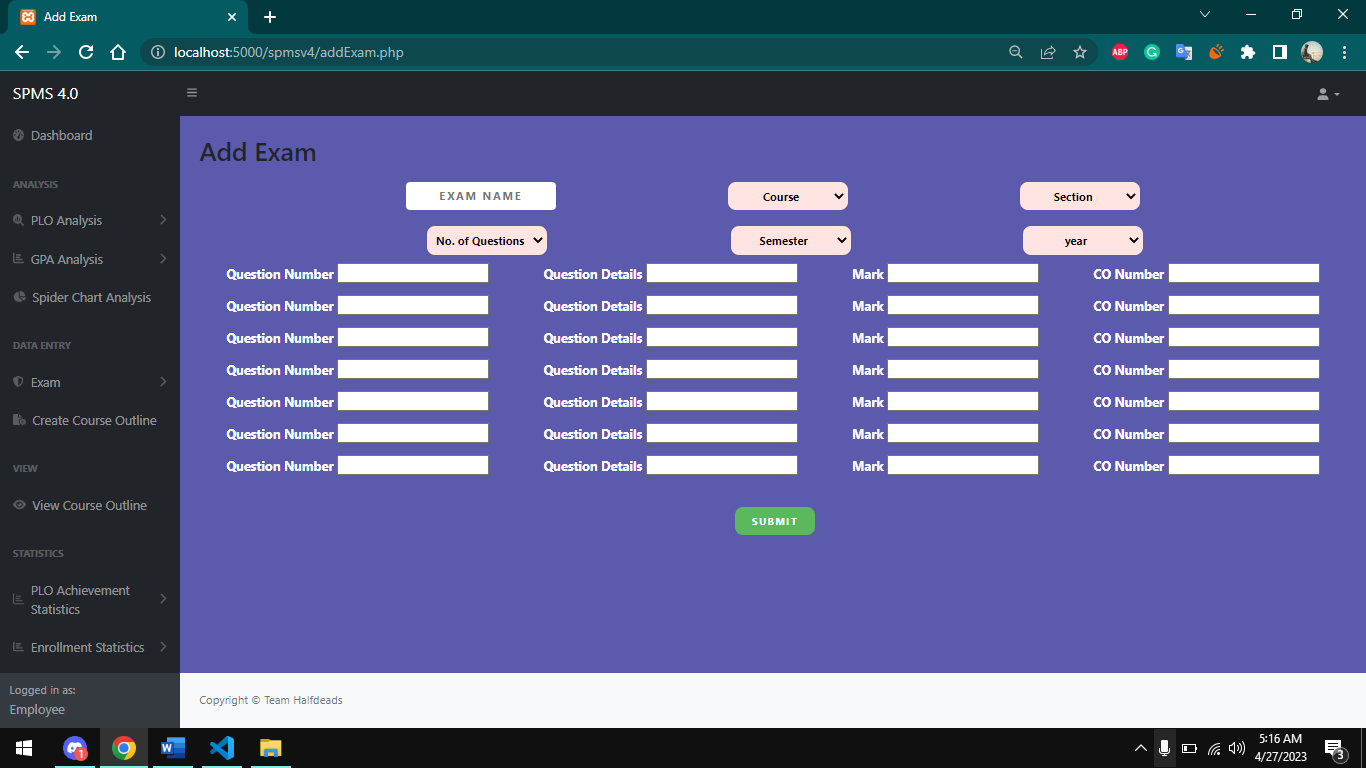
Description automatically generated

Text

Description automatically generated

Chart, radar chart

Description automatically generated



A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated

## OUTPUT FORMS

A picture containing text, screenshot, monitor, electronics

Description automatically generated

Text

Description automatically generated

A person standing in front of a screen

Description automatically generated with low confidence

Text

Description automatically generated

A screenshot of a computer

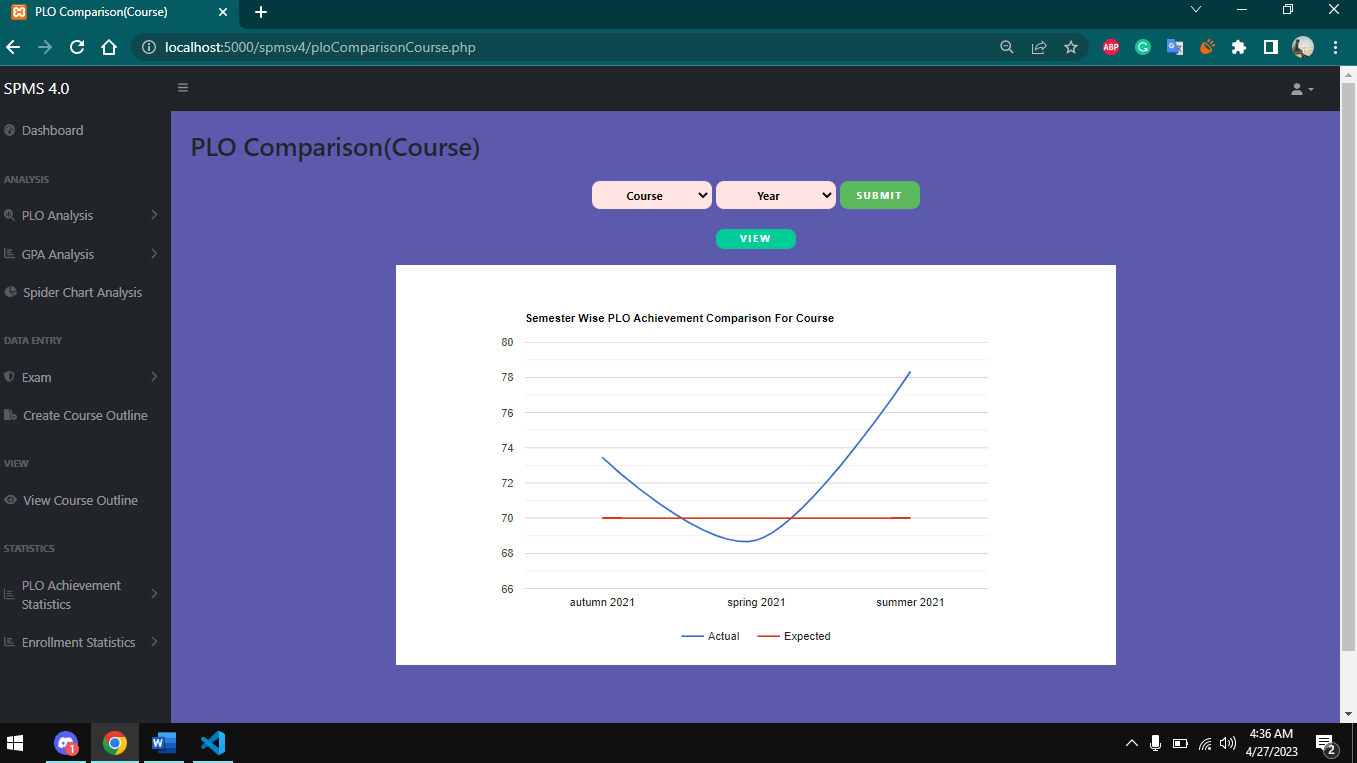
Description automatically generated

Text

Description automatically generated

Text

Description automatically generated



Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, application, website

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# **CHAPTER-5 CONCLUSION**

## PROBLEM AND SOLUTION

### Analysis Phase:

During the 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 in order to overcome such confusions, and information received during the interview was collected in order to get a better understanding of the system that was being developed.

### Designing Phase:

Some problems were faced while creating the EERD and Relational Schema during the

Design Phase, However, constant feedbacks from the faculty were enough to overcome

those issues.

### Implementation Phase:

All the System Requirements were completed successfully.

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

Database-integration: MySQL

### Additional Features and Future Development:

One new feature could be added to this system in the near future which can monitor previous course curriculum and then provide reports and analytics based on the student’s performance in that particular course to improve student performance rate.

## References

1. http://www.iub.edu.bd/AboutIUB/ataglance. [Online].
2. http://www.cse.iub.edu.bd/degrees/1. [Online].
3. https://www.betterevaluation.org/methods-approaches/methods/rich-pictures
4. https://www.ibm.com/cloud/blog/bpmn