

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi 110042

DEPARTMENT OF SOFTWARE ENGINEERING



Lab Code: SE-209n
Lab File

Submitted To:

Ankur Sir

**Department of Software
Engineering**

Submitted By:

Vivek kumar mandal
23/SE/176

Yash Jain
23/SE/177

Yashasvi Arya
23/SE/178

Yashika Gupta
23/SE/179

INDEX

S.NO	EXPERIMENT	DATE	SIGNATURE
1.	Problem Statement on Exam Management System		
2.	Use Case Diagram of Exam Management System		
3.	Use Case Description of Exam Management System		
4.	Data Flow Diagram for Exam Management System		
5.	Entity Relationship Diagram for Exam Management System		
6.	Software Requirement Specification for Exam Management System		
7.			
8.			
9.			
10.			
11.			
12.			
13.			

Experiment -1

AIM: Write the problem statement for Examination Management System

Introduction:

The Examination System aims to address the challenges faced in traditional examination methods by providing a digital platform for conducting exams. This document outlines the problems that the system intends to solve, the stakeholders involved, and the expected outcomes.

Problem Statement:

Context:

Examinations are an integral part of educational institutions, playing a critical role in evaluating student performance and progress. However, traditional methods of managing exams involve a lot of manual work, from scheduling to grading, which increases the chances of errors and inefficiencies. With the growing size and complexity of educational institutions, there is a need for an automated, efficient, and scalable Examination Management System (EMS).

Problem Overview:

Educational institutions, particularly universities, schools, and learning platforms, face numerous challenges in conducting and managing exams. These challenges impact various stakeholders including students, instructors, administrators, and invigilators. The traditional examination processes are often time-consuming, prone to human error, and lack transparency, resulting in dissatisfaction and inefficiencies. Additionally, with the increasing demand for remote and education, it becomes critical to implement a solution that allows examinations to be conducted seamlessly, whether in physical classrooms or environments.

Key Problems in Traditional Examination Management:

1. Manual Scheduling and Timetabling:

- Organizing exam schedules manually for a large number of students, across multiple courses, is prone to conflicts (e.g., overlapping exams, resource clashes).
- Adjustments due to sudden changes, such as rescheduled exams, lead to confusion and miscommunication.

2. Data Management Issues:

- Storing and managing vast amounts of data related to students, exam schedules, results, etc., becomes cumbersome when done manually.
- There is often a lack of integration with existing student information systems (SIS), making data retrieval and updates inefficient.

3. Lack of Transparency in Exam Results:

- Manual grading and result processing are susceptible to errors or biases, causing discrepancies in the results.
- Delayed result processing often frustrates students and hampers academic decision-making, such as course enrollment or graduation.

4. Security Concerns:

- The risk of question paper leaks or tampering with exam materials is high in manual systems.
- Ensuring the authenticity of examinees is particularly challenging in or remote examination setups.

5. Time-Consuming Examination Processes:

- Distributing question papers, collecting answer sheets, and manually grading large numbers of papers consumes significant time and resources.
- Instructors often face difficulties in grading assignments and exams consistently, especially when subjective evaluation is involved.

6. Limited Support for Examinations:

- With the rise of remote learning, traditional examination management methods struggle to provide secure and reliable exam delivery.
- Exams present additional challenges such as technical infrastructure, proctoring, and integrity checks.

7. Resource Management:

- Efficiently allocating rooms, invigilators, and other resources during exams is a logistical challenge in large institutions.
- Institutions struggle to make the best use of available resources without automated scheduling tools.

Consequences of these Problems:

- Administrative overhead increases, reducing overall efficiency.
- Students face confusion and frustration due to scheduling conflicts, delayed results, and a lack of transparency in grading.
- Instructors and examiners experience high levels of stress due to the workload involved in manual processes.
- There is a higher risk of errors, tampering, and security breaches, affecting the credibility of the examination system.

Need for an Examination Management System (EMS): An automated EMS can streamline and improve the efficiency of exam-related activities, addressing these challenges through:

1. **Automated Scheduling:** The system will automatically generate exam timetables, considering factors like student availability, course load, and resource constraints.
2. **Exam Support:** A robust platform for conducting exams, incorporating secure proctoring, authentication, and real-time monitoring of students.
3. **Grading Automation:** Automated tools for grading objective-type questions, with the ability to integrate subjective grading workflows for faster result processing.
4. **Data Security and Integrity:** Secure management of examination materials, student data, and results, ensuring no unauthorized access or tampering.
5. **Integrated Reporting:** A system that automatically generates reports for exam results, performance analysis, and feedback, helping instructors and students make informed decisions.
6. **Real-time Notifications and Alerts:** The system can send real-time notifications to students, instructors, and administrators about any changes in exam schedules, room allocations, and result declarations.

Conclusion:

A comprehensive, automated Examination Management System is essential for educational institutions to streamline the complex and time-consuming process of exam administration. It would ensure a secure, efficient, and transparent process that benefits all stakeholders while reducing administrative burden and improving the overall quality of examinations.

EXPERIMENT 2-

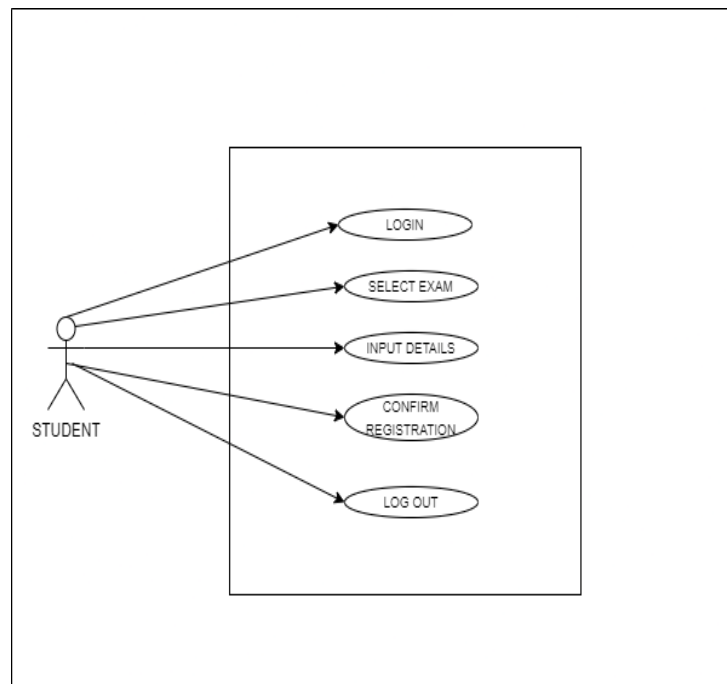
AIM- To draw the use case diagram for examination management system.

THEORY-

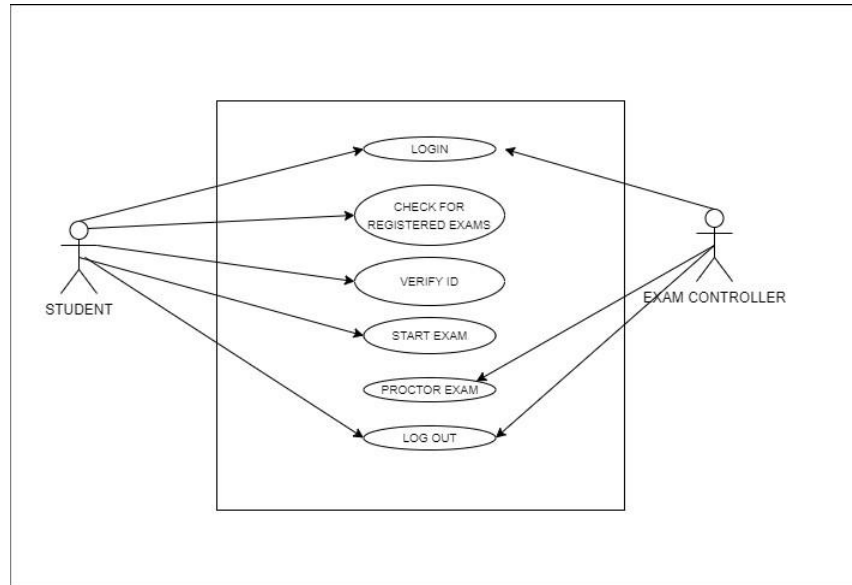
A use case diagram is a visual representation that describes how users (or "actors") interact with a system to achieve specific goals or use cases.

Key Components:

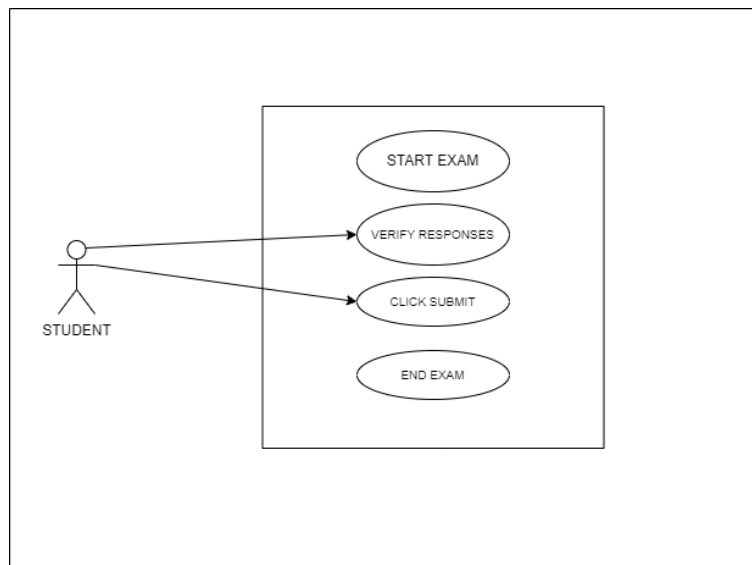
1. **Actors:** Represent users or other systems that interact with the system. Actors are typically depicted as stick figures.
2. **Use Cases:** Represent the actions or services the system provides. These are shown as ovals with descriptive names.
3. **System Boundary:** A rectangle that encloses all the use cases, defining the scope of the system.
4. **Relationships:** Lines or arrows that connect actors to use cases, showing interactions. Relationships include:



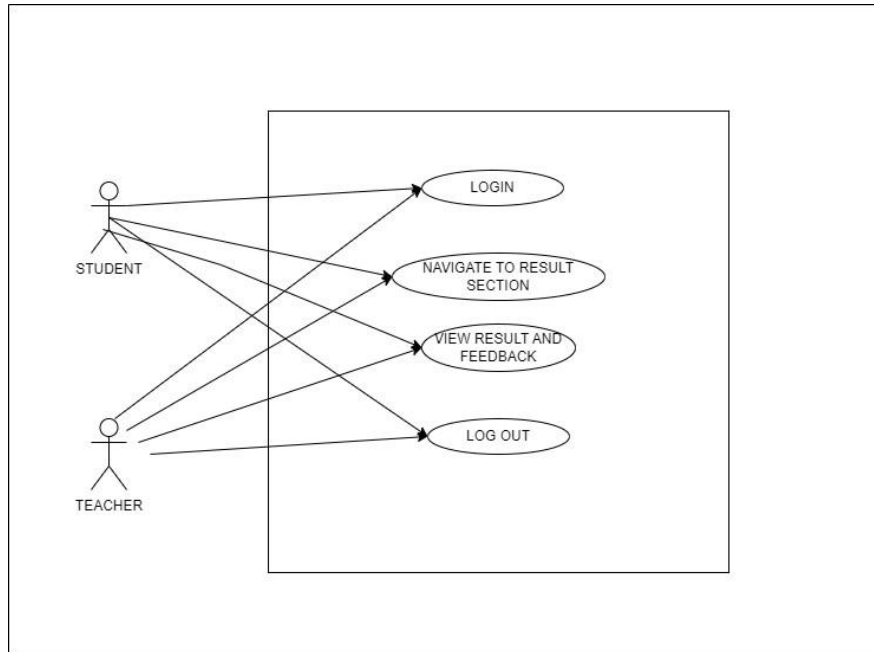
USE CASE1- REGISTER FOR EXAM



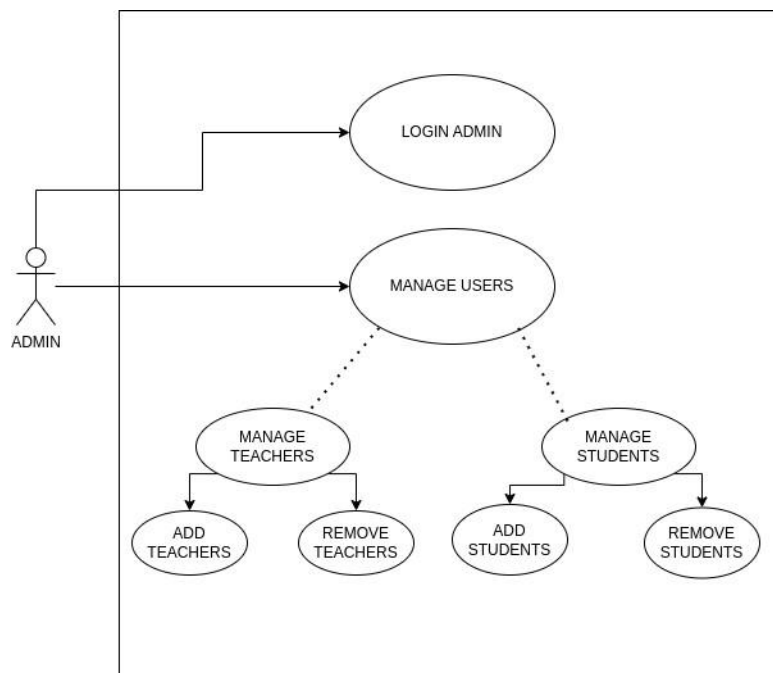
USE CASE 2- TAKE EXAM



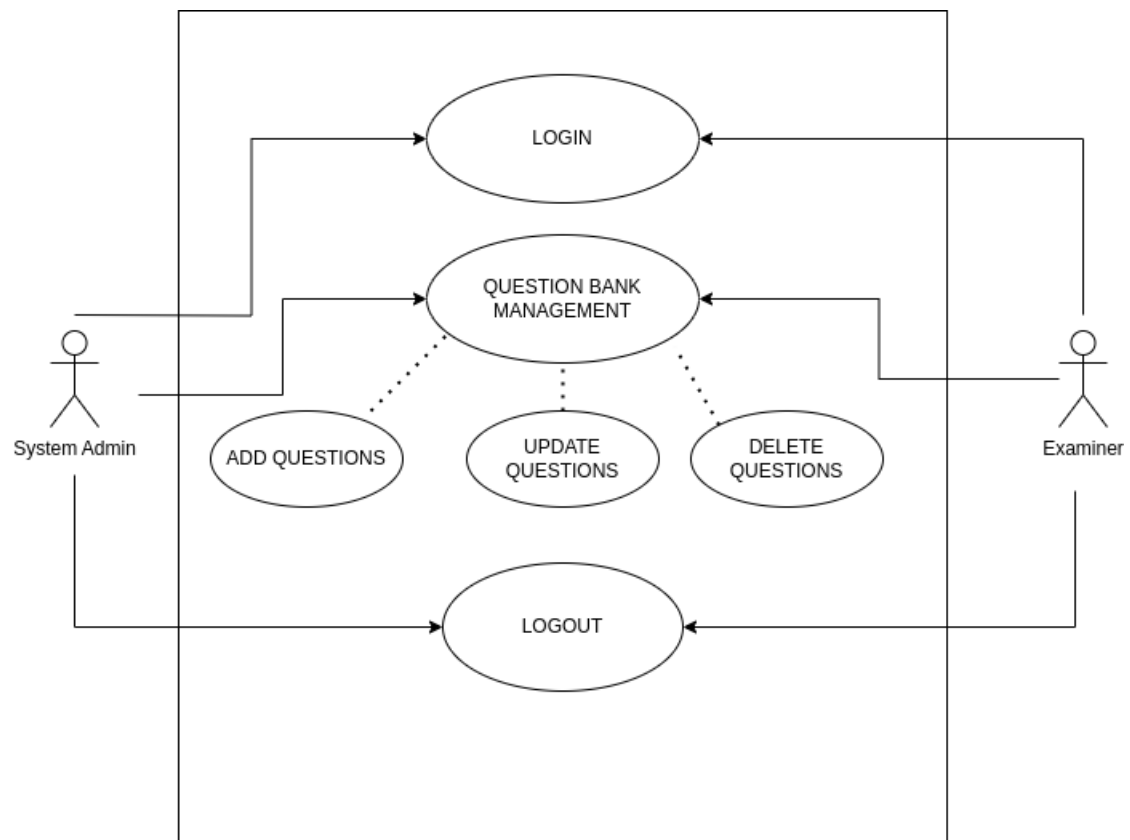
USE CASE 3- SUBMIT EXAM



USE CASE 4- VIEW RESULTS



USE CASE 5- MANAGE USERS



USE CASE 6- ADD AND MANAGE QUESTION BANKS

EXPERIMENT 3-

AIM- To state the use case description for an examination management system.

THEORY-

A use case description outlines the interactions between users (actors) and the system to achieve a specific goal. Here's a detailed use case description for an Examination Management System (EMS):

Use Case: User Registration

Use Case ID: UC01

Use Case Name: User Registration

Actor: Student, Admin

Description: This use case allows a new user (student or admin) to register on the examination management system.

Preconditions

- The user must have access to the registration page.
- The user must provide valid details (e.g., username, password, email).

Postconditions

- The user is registered successfully and can log in to the system.
- A confirmation message is displayed.

Basic Flow

1. The user navigates to the registration page.
2. The user enters the required information:
 - Username
 - Password
 - Email
 - Role (Student/Admin)
3. The user submits the registration form.
4. The system validates the input data.
5. If validation is successful:
 - The system creates a new user account.
 - A confirmation email is sent to the user.
 - A success message is displayed.
6. The user is redirected to the login page.

Alternative Flows

- **Invalid Data Entry:**
 - If the user enters invalid or incomplete data (e.g., email format is incorrect):
 1. The system displays an error message indicating the issue.
 2. The user corrects the input and resubmits the form.
- **Username/Email Already Exists:**
 - If the username or email is already registered:
 1. The system displays an error message stating that the username/email is already taken.
 2. The user can choose a different username/email.

Use Case: Create Exam

Use Case ID: UC02

Use Case Name: Create Exam

Actor: Admin

Description: This use case allows an admin to create a new exam in the system.

Preconditions

- The admin must be logged in.
- The exam details must be defined (e.g., name, subject, date).

Postconditions

- The exam is created and stored in the system.
- A confirmation message is displayed.

Basic Flow

1. The admin navigates to the "Create Exam" section.
2. The admin enters the exam details:
 - Exam Name
 - Subject
 - Date
 - Duration
 - Total Marks
3. The admin submits the exam details.
4. The system validates the input data.
5. If validation is successful:
 - The system creates the new exam.
 - A success message is displayed.
 - The exam is listed in the "Exams" section.

Alternative Flows

- **Invalid Exam Details:**
 - If any exam detail is invalid (e.g., date is in the past):
 1. The system displays an error message indicating the issue.
 2. The admin corrects the input and resubmits the form.

Use Case: Register for Exam

Use Case ID: UC03

Use Case Name: Register for Exam

Actor: Student

Description: This use case allows a student to register for a scheduled exam.

Preconditions

- The student must be logged in.
- The exam must be available for registration.

Postconditions

- The student is registered for the exam.
- A confirmation message is displayed.

Basic Flow

1. The student navigates to the "Available Exams" section.
2. The student selects an exam to register for.
3. The student clicks the "Register" button.
4. The system processes the registration.
5. A confirmation message is displayed.

Alternative Flows

- **Exam Already Full:**
 - If the exam has reached its capacity:
 1. The system displays a message indicating the exam is full.
 2. The student can choose another exam.

Use Case: Conduct Exam

Use Case ID: UC04

Use Case Name: Conduct Exam

Actor: Student

Description: This use case allows a registered student to start and complete an exam.

Preconditions

- The student must be registered for the exam.
- The exam must be available to attempt.

Postconditions

- The exam is submitted successfully.
- The student's answers are stored.

Basic Flow

1. The student logs in and navigates to "My Exams."
2. The student selects the exam to attempt.
3. The student clicks the "Start Exam" button.
4. The system loads the exam questions.
5. The student answers the questions.
6. The student submits the exam.
7. The system processes the answers and stores them.
8. A confirmation message is displayed.

Alternative Flows

- **Timeout:**
 - If the exam time expires:
 1. The system automatically submits the exam.
 2. A message is displayed indicating that the exam has been submitted due to timeout.

Use Case: View Results

Use Case ID: UC05

Use Case Name: View Results

Actor: Student

Description: This use case allows a student to view their exam results.

Preconditions

- The student must be logged in.
- The exam results must be available.

Postconditions

- The student's results are displayed.

Basic Flow

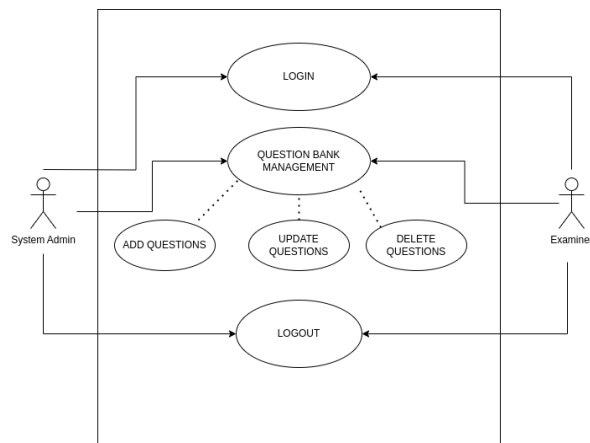
1. The student logs in and navigates to "My Results."
2. The student selects an exam to view results.
3. The system retrieves and displays the results.

Alternative Flows

- **No Results Available:**
 - If results are not yet available:
 1. The system displays a message indicating that results will be available soon.

Conclusion

These use case descriptions provide a clear understanding of the interactions between users and the Examination Management System. They outline the basic functionalities, including registration, exam management, registration for exams, conducting exams, and viewing results. Each use case also includes potential alternative flows to handle exceptions and ensure a comprehensive approach to system design.



Experiment 4

AIM – Create data flow diagram of Examination Management System

Introduction

Data Flow Diagrams (DFDs) are used to represent the flow of data within a system, illustrating how inputs are transformed into outputs through various processes. In the context of an Examination Management System (EMS), a DFD can help visualize the interactions between different components, users, and data stores.

Key Components of DFD

1. **Processes:** Activities or functions that transform inputs into outputs.
2. **Data Stores:** Repositories where data is stored within the system.
3. **External Entities:** Sources or destinations of data outside the system, such as users or external systems.
4. **Data Flows:** Arrows that represent the movement of data between processes, data stores, and external entities.

Levels of DFD

DFDs can be broken down into different levels:

- **Level 0 DFD (Context Diagram):** Provides a high-level overview of the system, showing the system as a single process with its external entities.
- **Level 1 DFD:** Breaks down the main process into subprocesses, showing how data flows between them.
- **Level 2 DFD:** Further decomposes subprocesses into more detailed processes and data flows.

Level 0 DFD (Context Diagram)

External Entities

- **Students:** Interact with the system to register for exams and view results.
- **Admins:** Manage exams, questions, and user registrations.

Main Process

- **Examination Management System:** The central system that handles all functionalities.

Data Flows

- Students send registration requests and view results.
- Admins create and manage exams.

Level 1 DFD

This level will break down the main process into several key subprocesses:

1. **User Registration**
 - Process for registering users (students/admins).
 - Inputs: User details.
 - Outputs: Confirmation of registration.
2. **Exam Management**
 - Process for creating and managing exams.
 - Inputs: Exam details.
 - Outputs: Confirmation of exam creation.
3. **Student Registration for Exams**
 - Process for students to register for specific exams.
 - Inputs: Exam selection.
 - Outputs: Confirmation of exam registration.
4. **Conducting Exams**
 - Process for students to attempt exams.

- Inputs: Exam data.
 - Outputs: Submitted answers.
5. **Results Processing**
- Process for calculating and displaying results.
 - Inputs: Exam answers.
 - Outputs: Results for students.

Level 2 DFD

This level could further break down subprocesses like **Conducting Exams** into individual steps such as:

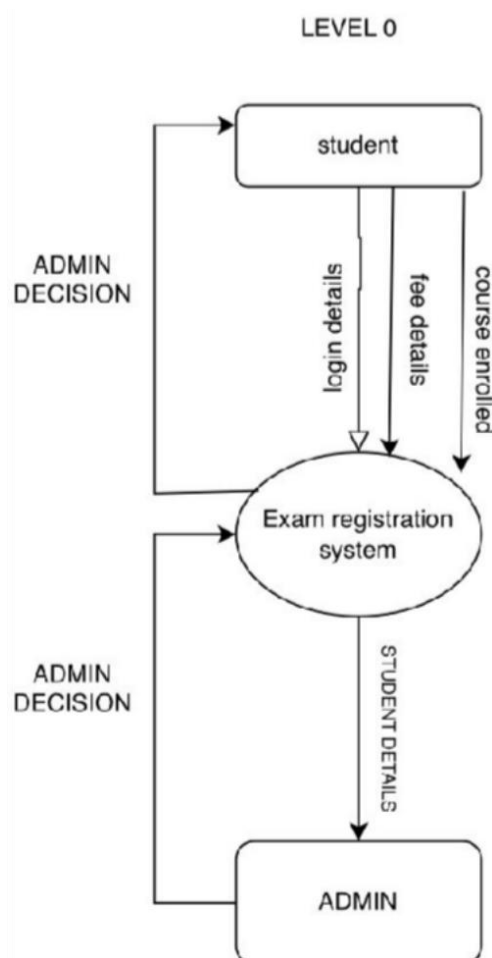
1. **Load Exam Questions**
2. **Display Questions to Students**
3. **Capture Student Answers**
4. **Submit Exam**

Each of these subprocesses would also interact with data stores, like the database of questions, student answers, and results.

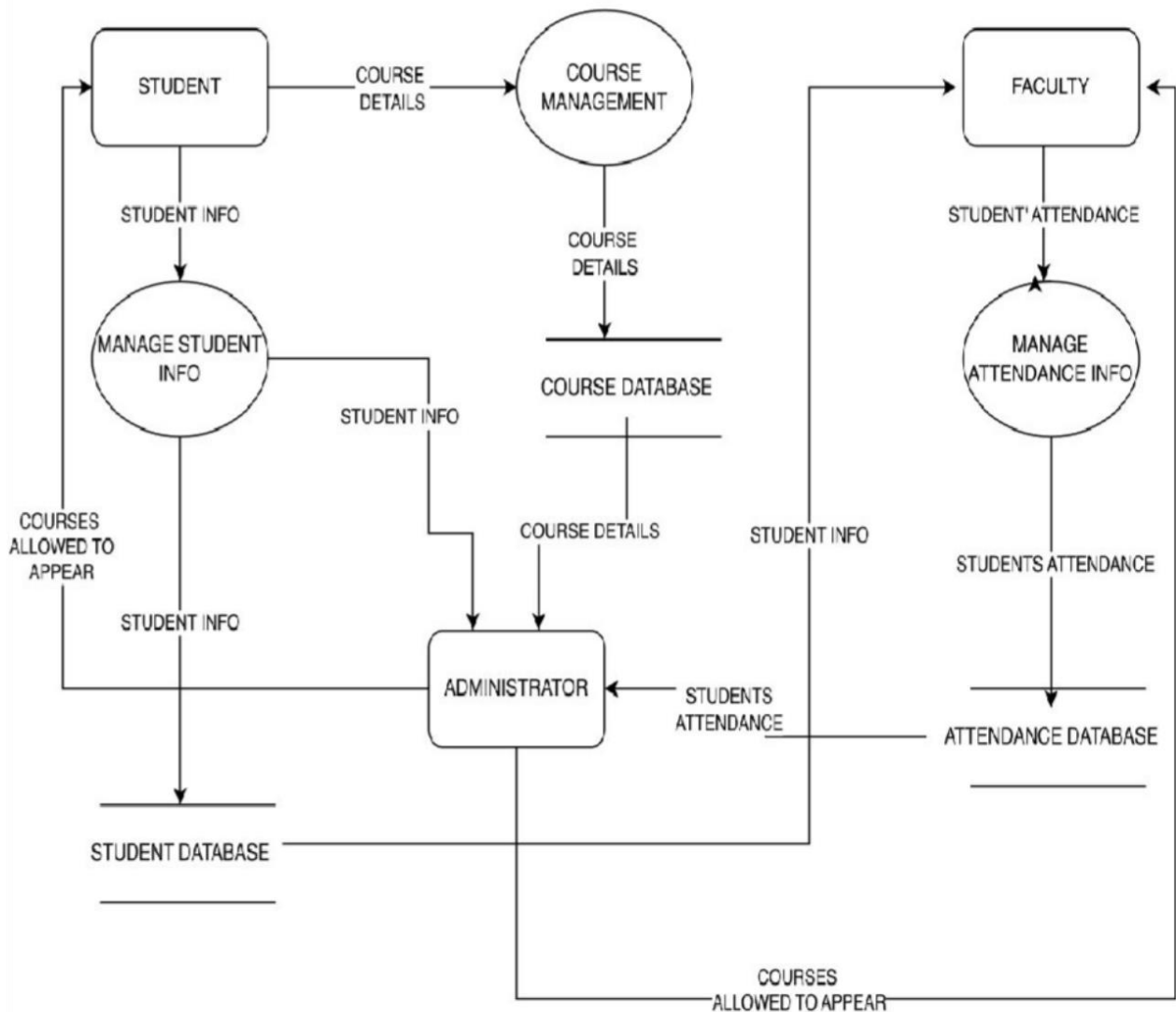
Conclusion

Data Flow Diagrams are a powerful tool for visualizing how data moves through an Examination Management System. By breaking down processes into detailed components, DFDs help clarify system functionalities, improve communication among stakeholders, and guide the development of the system architecture. They serve as a valuable part of the system design process, ensuring that all data interactions are well understood and properly implemented.

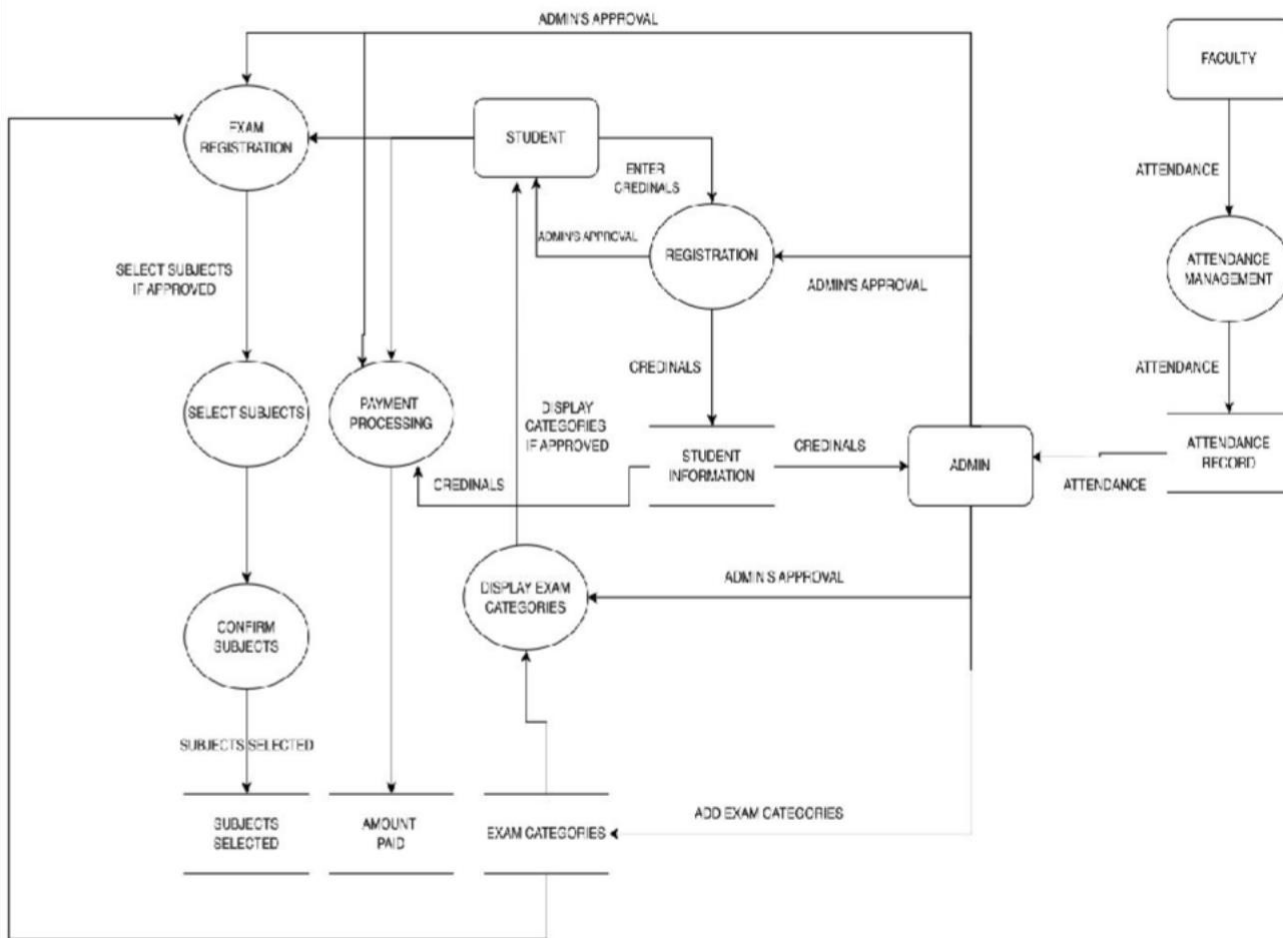
○



LEVEL 1



LEVEL 2



Experiment – 5

AIM:- Design an entity relationship diagram for the examination management system

THEORY:-

An **Entity-Relationship Diagram (ERD)** for an Examination Management System (EMS) typically models the relationships between different entities involved in the process, such as students, exams, courses, results, and administrators. Below is a description of the key entities and their relationships, followed by the structure of the ERD:

Key Entities and Relationships:

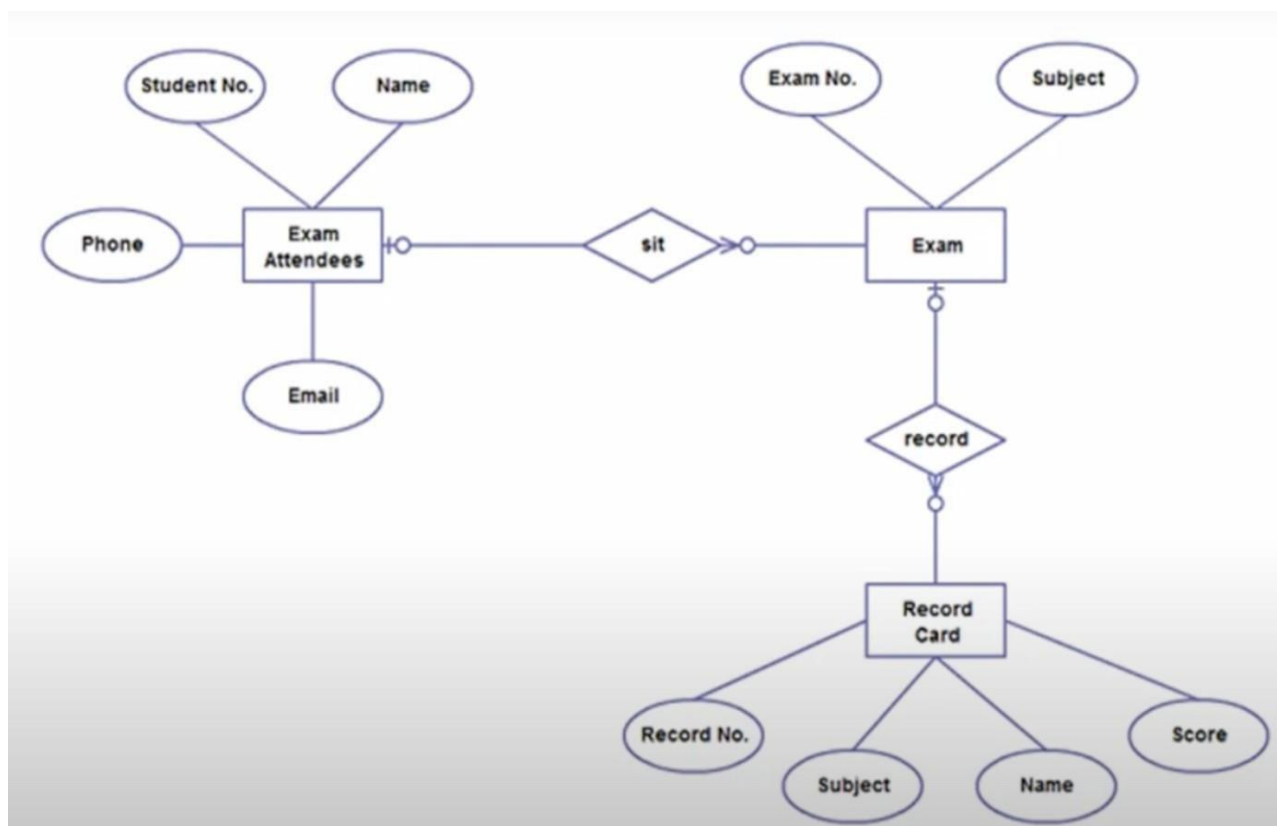
1. **Student:**
 - **Attributes:** Student_ID (Primary Key), Name, Email, Contact_Number, Department, Course_Enrolled, Year, Status
 - A **Student** can register for multiple exams.
2. **Exam:**
 - **Attributes:** Exam_ID (Primary Key), Course_ID (Foreign Key), Date, Time, Duration, Total_Marks
 - An **Exam** is scheduled for a **Course** and taken by multiple **Students**.
3. **Course:**
 - **Attributes:** Course_ID (Primary Key), Course_Name, Credits, Department, Faculty_ID (Foreign Key)
 - A **Course** has multiple **Exams** associated with it.
4. **Faculty:**
 - **Attributes:** Faculty_ID (Primary Key), Name, Department, Contact_Number, Email
 - **Faculty** members manage **Courses** and upload question papers for **Exams**.
5. **Question Paper:**
 - **Attributes:** Question_Paper_ID (Primary Key), Exam_ID (Foreign Key), Paper_File (encrypted), Date_Uploaded
 - A **Question Paper** is linked to an **Exam** and is uploaded by a **Faculty** member.
6. **Result:**
 - **Attributes:** Result_ID (Primary Key), Student_ID (Foreign Key), Exam_ID (Foreign Key), Marks_Obtained, Grade, Status (Pass/Fail)
 - A **Result** is the performance outcome of a **Student** in an **Exam**.
7. **Administrator:**
 - **Attributes:** Admin_ID (Primary Key), Name, Role, Contact_Number, Email
 - **Administrator** manages the exam schedules, results, and has access to all student and exam information.
8. **Exam Registration:**
 - **Attributes:** Registration_ID (Primary Key), Student_ID (Foreign Key), Exam_ID (Foreign Key), Registration_Date, Status
 - **Students** register for **Exams**, and the relationship is captured in this entity.

Relationships:

- **One-to-Many** between **Student** and **Exam Registration** (A student can register for multiple exams).
- **Many-to-Many** between **Student** and **Exam** via **Exam Registration** (A student can take multiple exams, and an exam can be taken by multiple students).
- **One-to-Many** between **Course** and **Exam** (A course can have multiple exams).
- **One-to-Many** between **Faculty** and **Course** (A faculty member can teach multiple courses).
- **One-to-Many** between **Exam** and **Question Paper** (Each exam has one question paper).
- **One-to-Many** between **Student** and **Result** (A student can have multiple results).

- **One-to-Many** between **Exam** and **Result** (An exam can generate multiple results for different students).
- **One-to-One** between **Faculty** and **Exam** (Each exam is managed by a single faculty member for question paper upload).

ENTITY RELATIONSHIP DIAGRAM FOR EXAMINATION MANAGEMENT SYSTEM



EXPERIMENT -6

AIM:- Design a SRS document for examination managementsystem

THEORY:

1. Introduction

Purpose

The purpose of this document is to describe the functional and non-functional requirements for the Examination Management System (EMS). This system will be designed to manage examinations for educational institutions, handling processes from scheduling exams, managing question papers, grading, and result publication.

Scope

The Examination Management System will:

- Allow the scheduling of exams
- Generate and manage question papers
- Manage student information and exam registrations
- Record exam results and grades
- Publish results to students and stakeholders

Definitions, Acronyms, and Abbreviations

- **EMS:** Examination Management System
- **GUI:** Graphical User Interface
- **DBMS:** Database Management System
- **SRS:** Software Requirements Specification

References

- IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
- Institution XYZ Examination Handbook (Version 2024)

Overview

This document provides a detailed description of the system functionality, system constraints, assumptions, and dependencies. It includes both functional and non-functional requirements.

2. General Description

Product Perspective

The EMS is a standalone web-based application with integration points to other existing systems like the institution's Student Information System (SIS) and Learning Management System (LMS). The system will follow a client-server architecture with a back-end database to store and retrieve data.

Product Functions

The EMS will provide the following key functions:

- **Exam Scheduling:** Schedule exams for different courses and notify students.
- **Question Paper Management:** Generate, upload, and manage question papers securely.
- **Student Registration:** Allow students to register for exams.
- **Grading and Result Management:** Record grades, manage re-evaluations, and publish results.
- **Reports:** Generate exam reports for faculty and administrators.

User Characteristics

- **Administrator:** Responsible for managing users, scheduling exams, and generating reports.
- **Faculty:** Responsible for uploading question papers, grading exams, and reviewing results.
- **Students:** View exam schedules, register for exams, and access their results.
- **Examiners/Moderators:** Oversee exam activities, review questions, and ensure fairness.

Constraints

- The system must comply with institutional examination regulations.
- Security protocols must be in place to ensure confidentiality of question papers and student results.

Assumptions and Dependencies

- Internet connectivity is assumed for all users.
- The EMS will depend on integration with the Student Information System (SIS) for student details.
- Browser compatibility with the latest versions of Chrome, Firefox, and Edge is assumed.

3. Specific Requirements

.1 Functional Requirements

Exam Scheduling

- **FR1.1:** The system shall allow administrators to schedule exams for courses.
- **FR1.2:** The system shall notify students via email and SMS when exam schedules are published.
- **FR1.3:** The system shall allow rescheduling of exams with proper conflict resolution.

Question Paper Management

- **FR2.1:** The system shall allow faculty to upload question papers securely.
- **FR2.2:** The system shall support the generation of random question papers.
- **FR2.3:** The system shall store uploaded question papers in an encrypted format.

Student Registration

- **FR3.1:** The system shall allow students to register for exams based on their enrolled courses.
- **FR3.2:** The system shall provide students with a confirmation of registration.

Grading and Results Management

- **FR4.1:** The system shall allow faculty to enter grades for exams.
- **FR4.2:** The system shall calculate total marks and grades based on institution grading policies.
- **FR4.3:** The system shall publish results after approval by the examiners.

Reports and Analytics

- **FR5.1:** The system shall generate exam reports (e.g., pass/fail statistics, average scores) for faculty and administrators.
- **FR5.2:** The system shall provide performance reports for individual students.

Non-Functional Requirements

Performance Requirements

- The system should handle up to 10,000 concurrent users.
- Exam scheduling should take no more than 2 seconds to load.

Security Requirements

- The system must encrypt sensitive data such as question papers and exam results.
- Role-based access control must be implemented to restrict access based on user roles.

Usability Requirements

- The system interface should be intuitive, allowing non-technical users to manage exams without extensive training.
- The system should be accessible to people with disabilities in compliance with WCAG 2.1 standards.

Reliability and Availability

- The system should have 99.9% uptime.
- Daily backups of all data must be performed to prevent data loss.

Maintainability

- The system should allow easy updates to question banks, schedules, and exam policies without downtime.
- Code documentation should follow industry best practices to ensure maintainability.

External Interface Requirements

User Interfaces

- **UI1:** The system shall provide a web-based user interface accessible through standard web browsers.
- **UI2:** The system shall have different interfaces for students, faculty, and administrators.

Hardware Interfaces

- **HI1:** The system shall run on standard desktop and mobile hardware.

Software Interfaces

- **SI1:** The system shall integrate with the Student Information System (SIS) for retrieving student data.
- **SI2:** The system shall use a REST API for third-party integration.

System Attributes

- **Reliability:** The system shall recover from any failures within 5 minutes.
- **Scalability:** The system shall scale to accommodate the growing number of students and faculty.
- **Portability:** The system shall be deployable on various cloud platforms.

4. Verification

- **Test Case 1:** Verify that the exam scheduling system works as intended.
- **Test Case 2:** Verify secure upload of question papers.
- **Test Case 3:** Ensure student registration process is functional.
- **Test Case 4:** Test the calculation and publication of exam results.

5. Appendices

- **Appendix A:** Glossary of Terms
- **Appendix B:** User Guide for EMS
- **Appendix C:** Data Flow Diagrams and System Architecture

This SRS document follows the IEEE standards and provides a complete framework for developing and testing the Examination Management System.