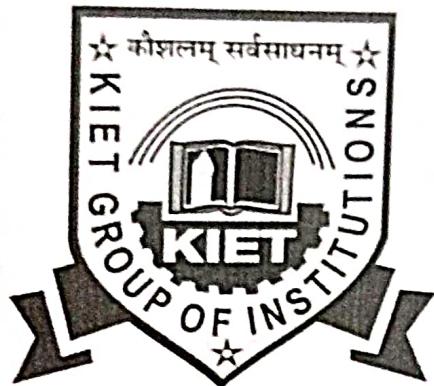


# KIET GROUP OF INSTITUTIONS

## DELHI-NCR, GHAZIABAD



Academic Year 2023-24  
Department : MCA

### Practical File

Full Name : Shubhangini Agrawal  
Roll No. : 2200290140151  
Semester : 3<sup>rd</sup>  
Section : C  
Subject : Software Engineering Lab  
Subject Code : KCA - 352

Submitted To : Dr. Amit Kumar  
(Assistant Professor)



**KIET Group of Institutions, Ghaziabad**  
**Department of Computer Applications (NBA Accredited)**

(An ISO – 9001: 2015 Certified & 'A+' Grade accredited Institution by NAAC)

**SOFTWARE ENGINEERING LAB (KCA-352)**

**MCA-III SEMESTER, 2023-24**

**PRACTICAL LIST**

S. No.	Program	Date of Completion	Faculty Signature	Remarks
1	Prepare an SRS document in line with the IEEE recommended standards for the specified Case Study. (Functional Requirements)			
2	Prepare an SRS document in line with the IEEE recommended standards for the specified Case Study. (Non-Functional Requirements).			
3	List out the entities and identify the relationship between them. Also, identify related attributes supposed to be recorded while considering the normalization rule.			
4	List out the required processes to be coded along with the required flow charts.			
5	Draw the activity diagram for the specified Case Study.			
6	Draw the DFD to be considered while coding the individual processes or functions.			
7	List out the categories of reports and the required data to be represented with them. Also, determine the layout of the reports that may be used by individual authorities of the intended organization.			
8	While considering the scope of the individual functionalities draw the use case diagram for each of it.			
9	On the basis of required validations map the different test cases to handle all possible critical cases which may arise during the life cycle of the software.			
10	Identify the classes. Classify them as weak and strong classes and draw the class diagram for the specified Case Study.			
11	Produce complete documentation along with PPT to be presented.			
12	Presentation.			

# Lab Work -1

Prepare an SRS document in line with the IEEE Recommended standards for the specified case study (Functional Requirements)

## 1. Introduction

### 1.1 Purpose

### 1.2 Scope

### 1.3 References

### 1.4 Overview

## 2. Overview

### 2.1 Product Perspective

### 2.2 Benefits

### 2.3 Features

### 2.4 Working Overview

## 3. Specific Requirements

### 3.1 Functional Requirements

## 4. Use Cases

## 5. Conclusion

## Introduction

### 1.1 Purpose

The purpose of the SRS document is to outline the detailed requirements, functionalities and specifications of the "Online Quiz Application System"

## 1.2 Scope

The application is a web-based and accessible on any device with a web browser, that will provide users with an engaging and educational experience.

## 1.3 References

There is no external reference to any other document in this SRS document.

## 1.4 Overview

Online Quiz Application typically has an user interface that allows users to browse through a list of quizzes, select a quiz to take and answer the questions.

## 2. Overview

### 2.1 Product Perspective

The perspective of "Online Quiz Application" is to provide users with an engaging and educational experience through a variety of quizzes on different topics. It has the purpose of enhanced learning, skill development, self assessment and employee training.

### 2.2 Product Benefits

The product has the following benefits.

- a. Convenience
- b. Flexibility
- c. Engagement
- d. Feedback
- e. Assessment

## 2.3 Product features

The application will have the following features

- Admins : Users will be able to create an account in order to add teachers & students
- Teacher : Teacher panel will be able to create (Quiz creation) their own quizzes
- Quiz Taking : The students will be able to take quizzes
- Quiz Categories : The application allows the teachers to add and students to take various quizzes

## 2.4 Product Working

The simple overview of how an quiz application might work

- The user logs into the application either as admin, teacher, or student
- If an admin, either adds a course or registers a teacher
- If a teacher, either adds a student or adds a category or adds questions
- If a student, takes quizzes and analyses performance

## 3. System Requirements

### 3.1 Functional Requirements

- User Accounts : Users should be able to create and manage their own accounts. Includes ability to set username & password

- Quiz Creation : Users should be able to create their own quizzes, who are teachers. Includes ability to add questions, choose the type of questions & add categories.
- Quiz taking : Users(students) should be able to take quizzes that have been created by teachers
- Scoring : The application automatically scores quizzes and provide teacher & student with the results
- Quiz categories : The application allow users to add various categories of quizzes.

#### 4. Use Cases.

The application can be used in any school or college for taking and creating quizzes. The intended users of the application can be

- Administrators
- Students
- Teachers

#### 5. Conclusion

The SRS document has specified the requirements (functional) for an online quiz application. It is a convenient, flexible, engaging, user friendly application.

# Lab Work- 2

Prepare an SRS document in line with the IEEE recommended standards for the specified case study  
(Non Functional Requirements)

## 1. Introduction

### 1.1 Purpose

### 1.2 Scope

### 1.3 References

### 1.4 Overview

## 2. Description

### 2.1 Product Perspective

### 2.2 Benefits

### 2.3 Features

### 2.4 Working Overview

## 3. Specific Requirements

### 3.1 Non Functional Requirements

## 4. Use Cases

## 5. Conclusion

## 1. Introduction

### 1.1 Purpose

The purpose of the SRS document is to outline the detailed requirements, functionalities and specifications of the "Online Quiz Application"

## 1.2 Scope

The application is web based and accessible on all devices through a web browser, that will provide users with an engaging and educational experience.

## 1.3 References

There is no external reference to any other document in this SRS document.

## 1.4 Overview

The Online Quiz Application typically has an user interface that allows users to browse through a list of quizzes, select a quiz to take and answer the questions.

## 2. Description

### 2.1 Product Perspective

The perspective of online quiz application is to provide users an engaging and educational experience through a variety of quizzes on different topics. It serves the purpose of enhanced learning, skill development, self assessment and employee training.

### 2.2 Product Benefits

The product has the following benefits

- a. Convenience
- b. Flexibility
- c. Engagement
- d. Feedback
- e. Assessment

## 2.3 Product features

The application will have the following features

- Admins: Users will be able to create an account in order to add teachers & students
- Quiz Creation: Teacher panel will be able to create their own quizzes.
- Quiz Taking: The students will be able to take quizzes.

## 2.4 Product Working

The simple overview of how an quiz application might work

- The user logs into the application, either as admin, teacher or student
- If an admin, either adds a course or register a teacher
- If an teacher, either adds a student or adds a category
- If a student, takes quizzes and analyses performance

## 3. System Requirements

### 3.1 Non Functional Requirements

The system shall meet the following non-functional requirements.

#### 3.1.1 Performance

The application should be able to handle a large number of quizzes without any performance issues.

### 3.1.2 Scalability

The application should be able to scale to accommodate a growing number of users.

### 3.1.3 Security

The application should be secure and protect data from unauthorized access.

### 3.1.4 Reliability

The application should be reliable and available to users when they need it.

## 4. Use Cases

The application can be used in any school or college for taking and creating quizzes. The intended users of the application can be

- Administrators
- Students
- Teachers.

## 5. Conclusion

The SRS document specifies has the non-functional requirements for an online quiz application. It is a convenient, flexible, engaging and user-friendly application.

## Lab-Work-3

List out the entities and identify the relationship between them. Also, identify related attributes supposed to be recorded while considering the normalization rule.

To identify the entities and relationships in a database schema, as well as the related attributes, we take context of "Online Quiz Application".

### 1. Entities

#### 1.1 User

Attributes: UserId (PK), UserName, Email, Password, FirstName, LastName, UserType, ProfilePicture, Registration Date, InrollId (FK)

#### 1.2 Quiz

Attributes: QuizId (PK), Title, Description, Category, CreatorId, CreatedDate

#### 1.3 Question

Attributes: QuestionId, Text, Type

#### 1.4 Option

Attributes: OptionId, QuestionId (FK), Text, IsCorrect

#### 1.5 QuizAttempt

Attributes: AttemptId (PK), UserId, QuizId, StartTime, EndTime, Score, Status.

## 1.6 Category

Attributes :- CategoryId, UserId, CourseId, QuizId

## 1.7 Course

Attributes : CourseId, CourseName, Description,

## 1.8 Enrollment

Attributes : enrollmentId, studentId, courseId, EnrollmentDate

## 2. Relationships

### 2.1 User - Quiz Relationship

- a. User can create multiple Quizzes.
- b. User can attempt multiple Quizzes.

### 2.2 Quiz - Question Relationship

- a. A quiz can have multiple questions.
- b. a question can belong to one quiz only

### 2.3 Question Option Relationship

- a. A question can have multiple option

### 2.4 Quiz - Category Relationship

- a. A quiz can belong to one category

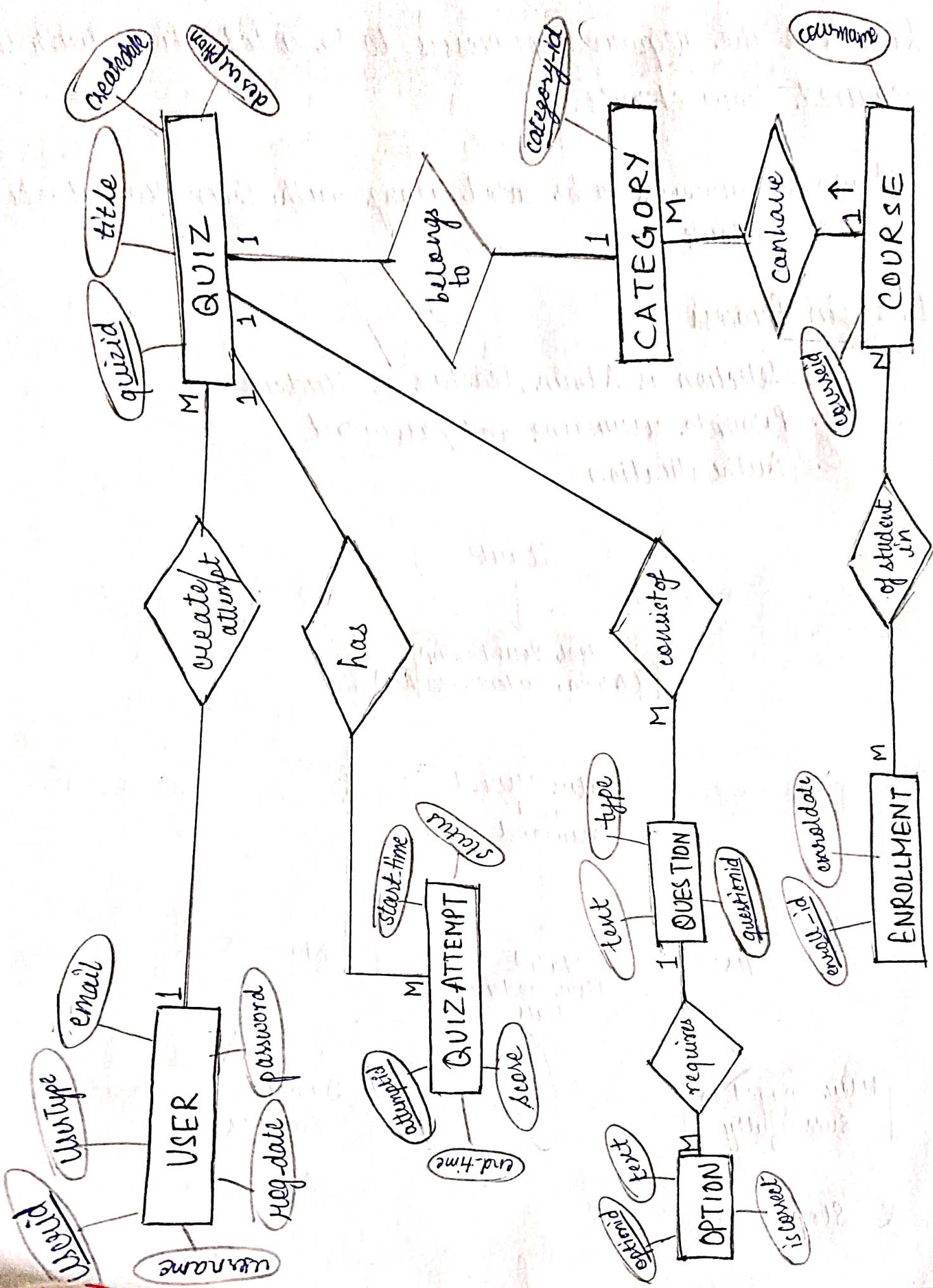
### 2.5 Category - Course Relationship

- a. A course can have multiple category quizzes

### 2.6 Enrollment - Course Relationship

- a. A student can enroll in ~~one~~ multiple courses

### 3. E-R Diagrams .



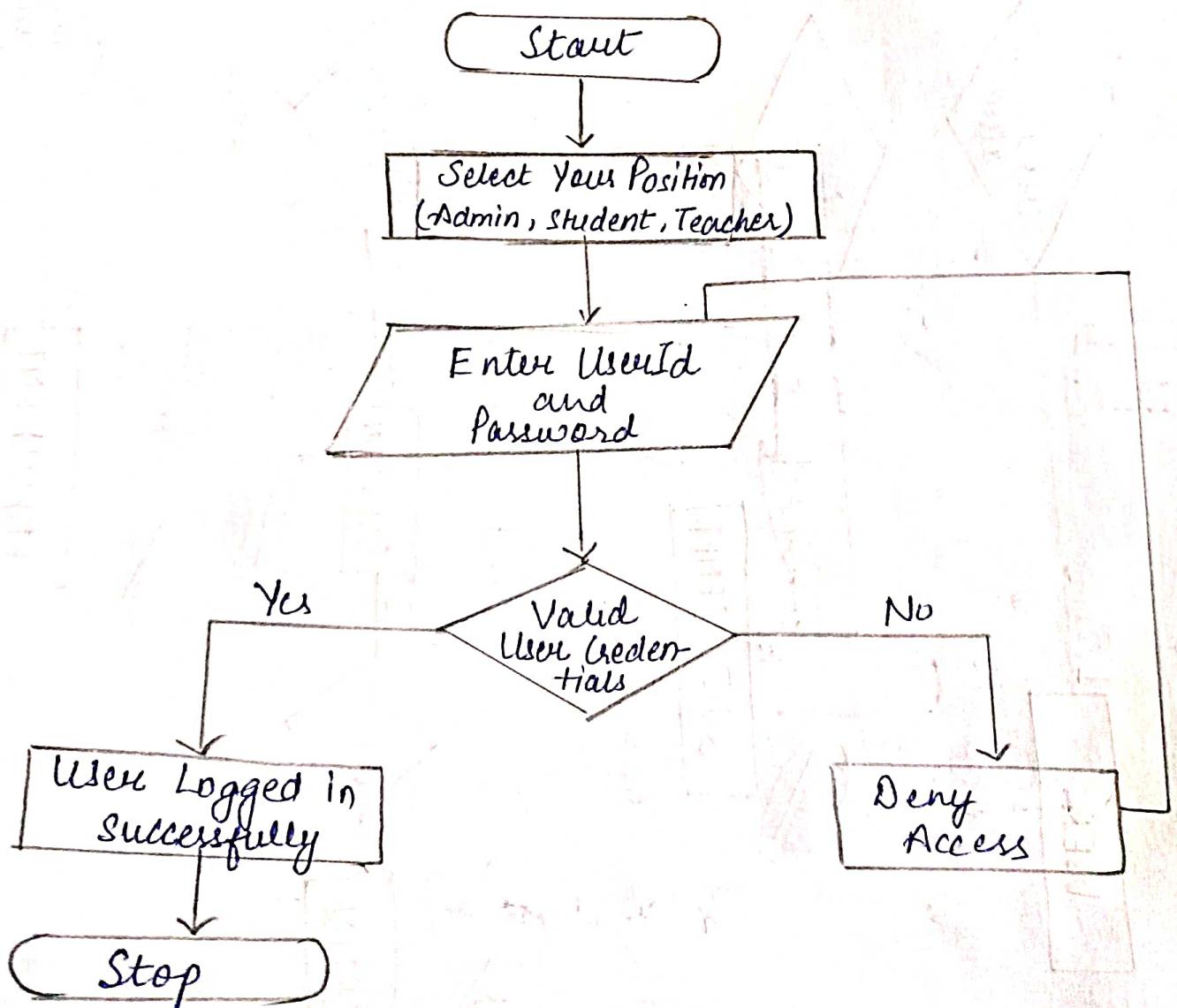
## Lab-Work -4

List out the required processes to be coded along with the required flowcharts.

Various processes to be coded along with their flowcharts is as follows.

### 1. Login Process

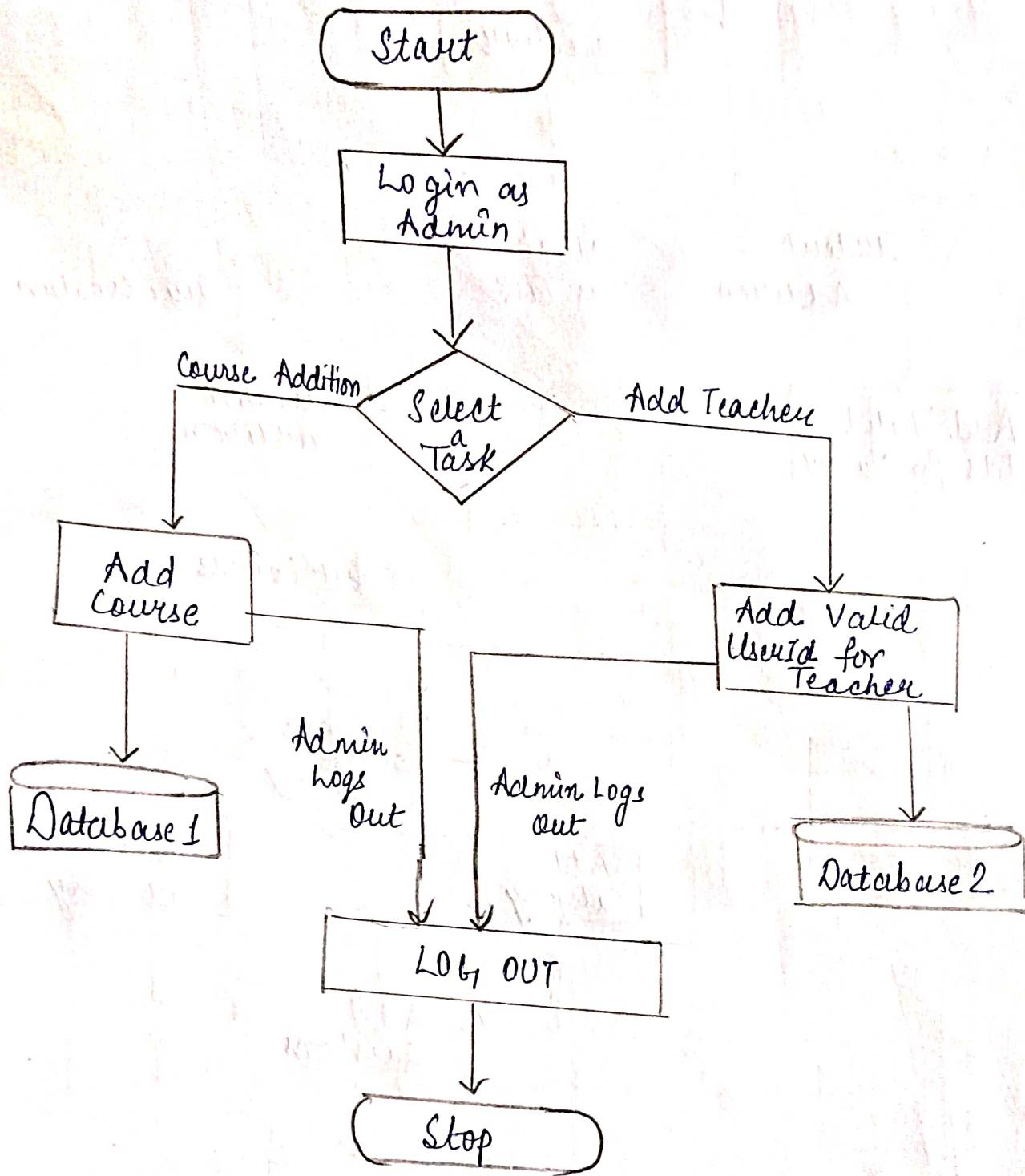
- Selection as Admin, Teacher or Student
- Provide username and password
- Authentication



## 2. Admin Panel

a. Login

b. Select to either add course or add teacher

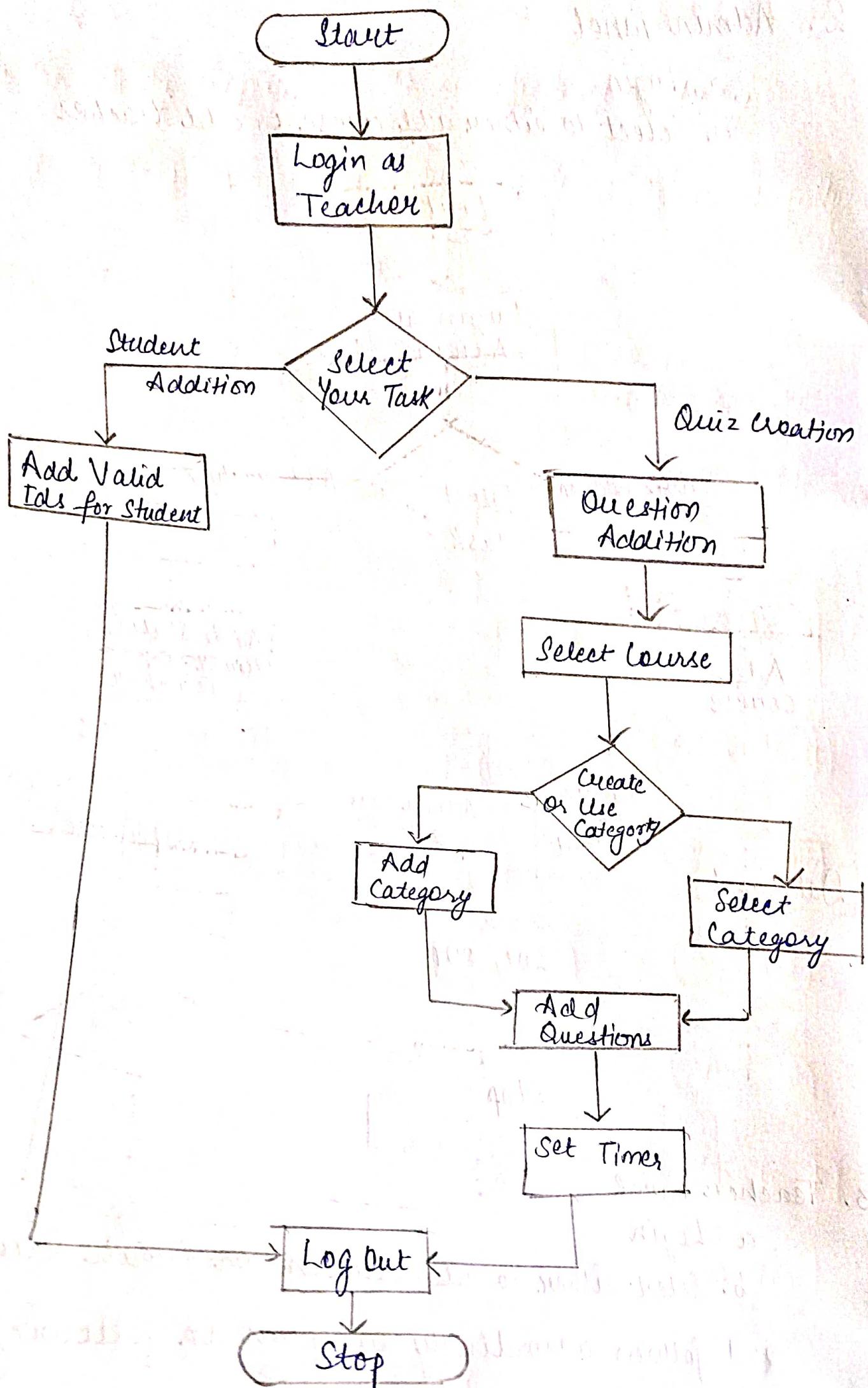


## 3. Teacher Panel

a. Login

b. Select either to add Student or Create Quiz

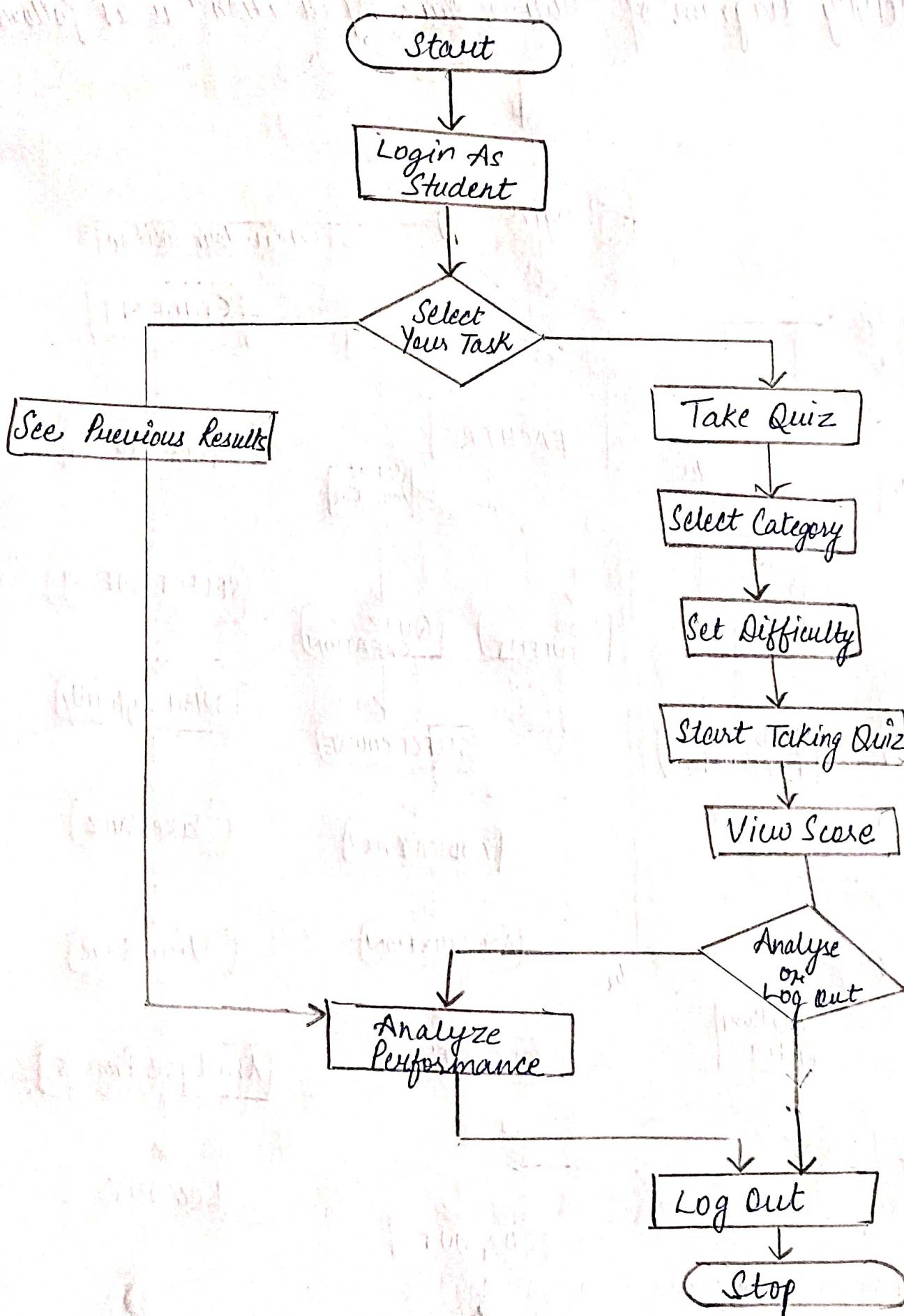
and follow accordingly as given in following flowchart



#### 4. Student Panel

a. Login

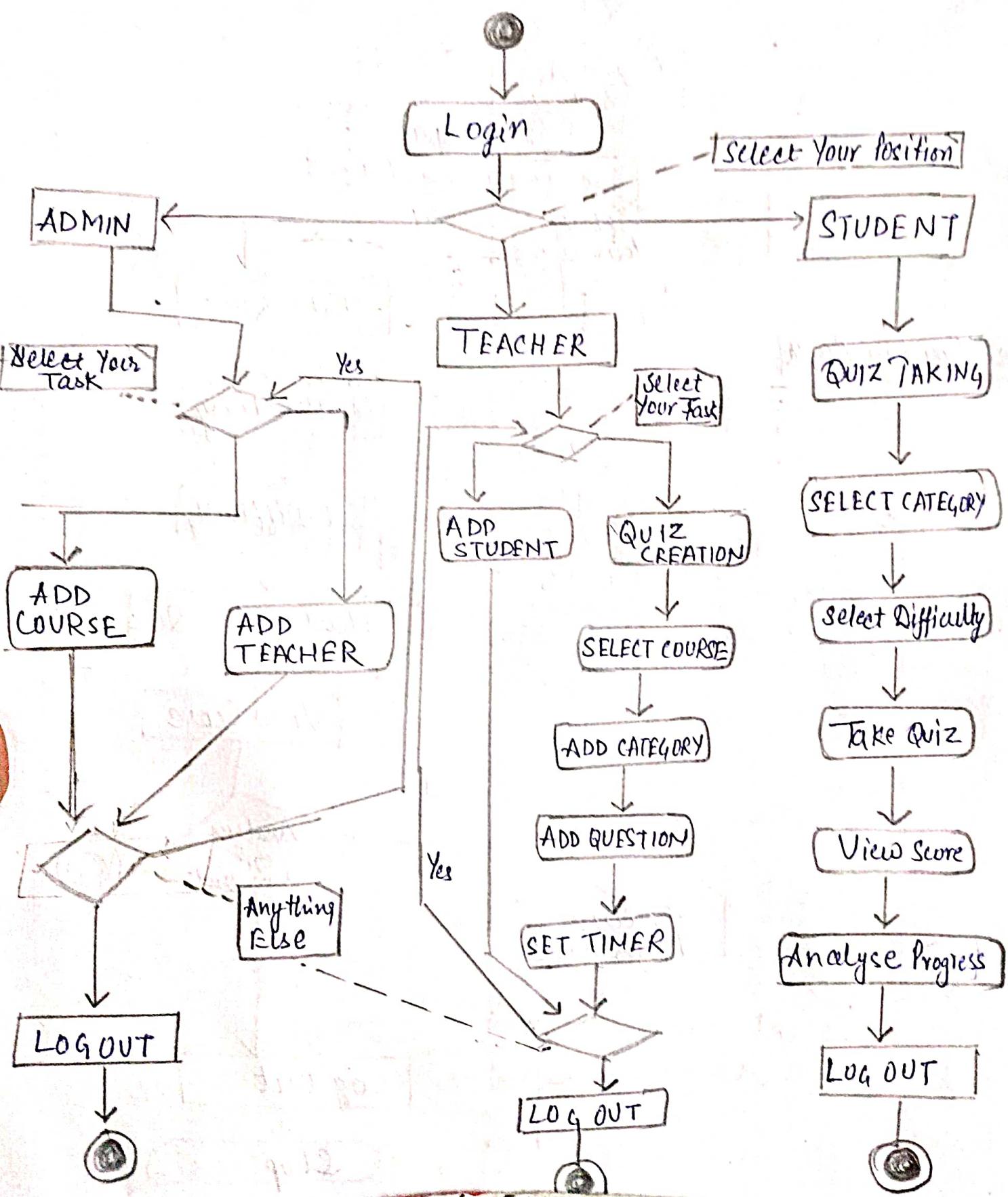
b. Select to Take Quiz or View Progress



## Lab-Work 5

Draw the activity diagram for the specified case study

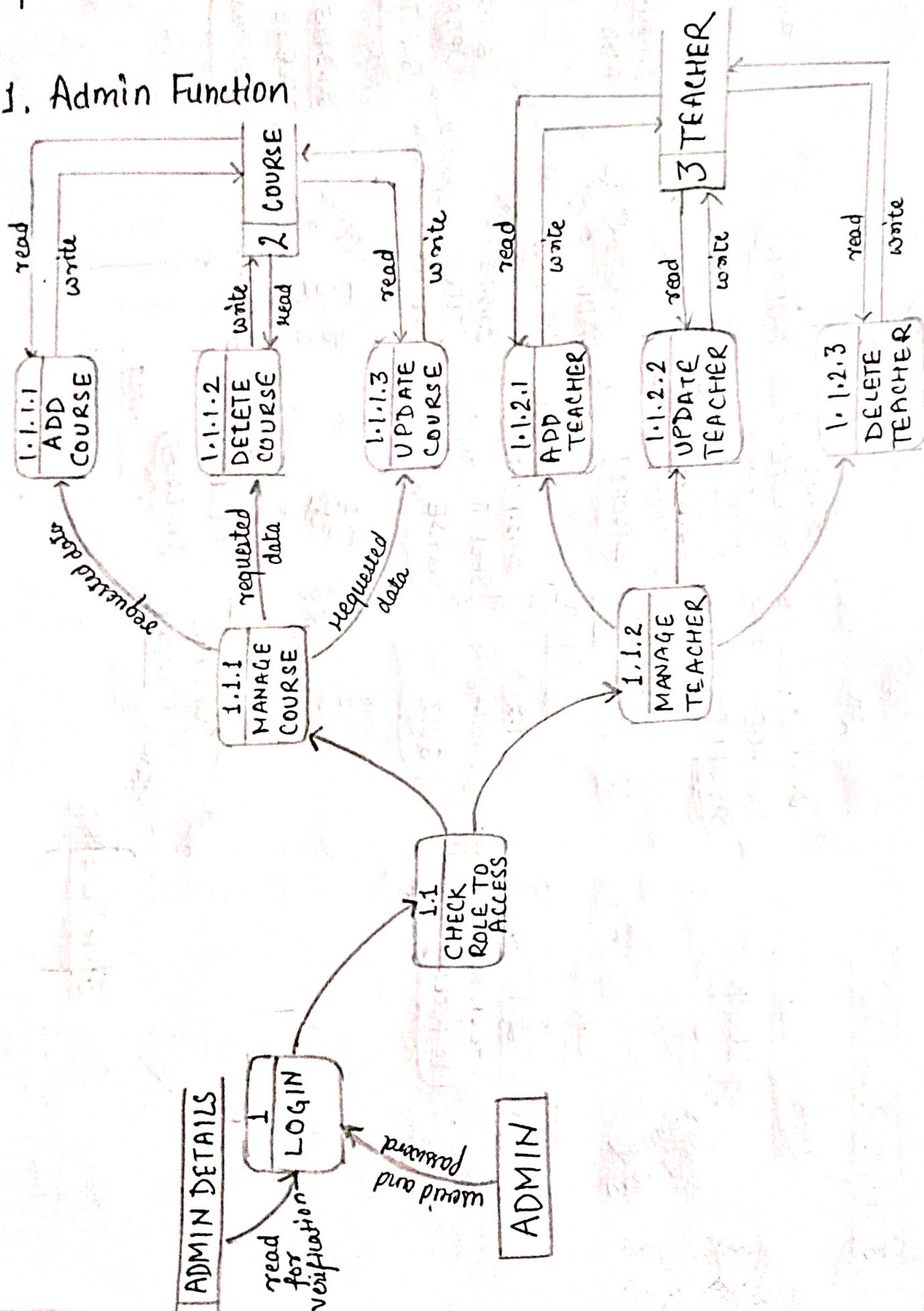
The activity diagram of "Online Quiz Application" is as follows



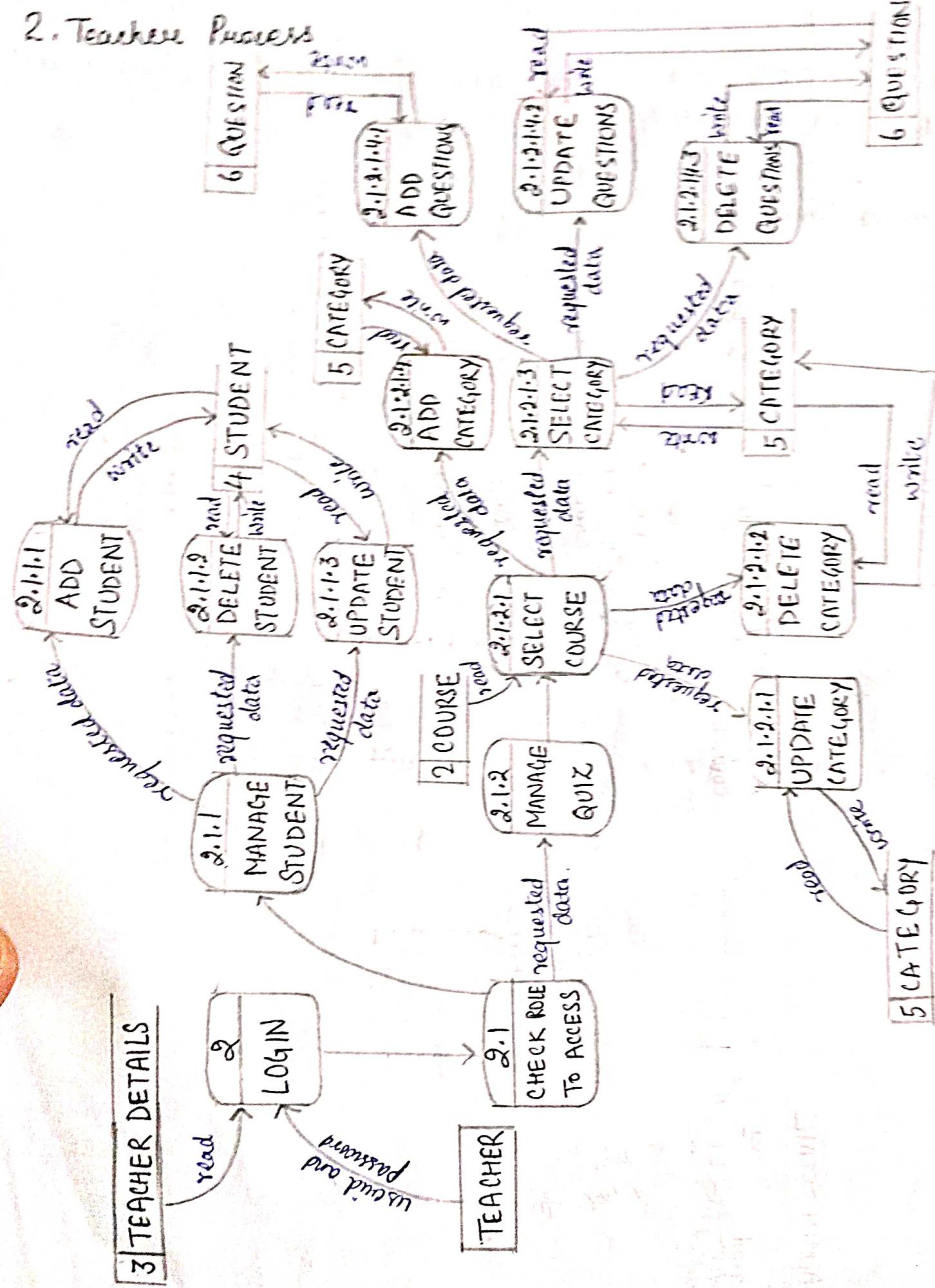
# Lab-Work - 6

Draw the DFD to be considered while coding the individual processes or functions.

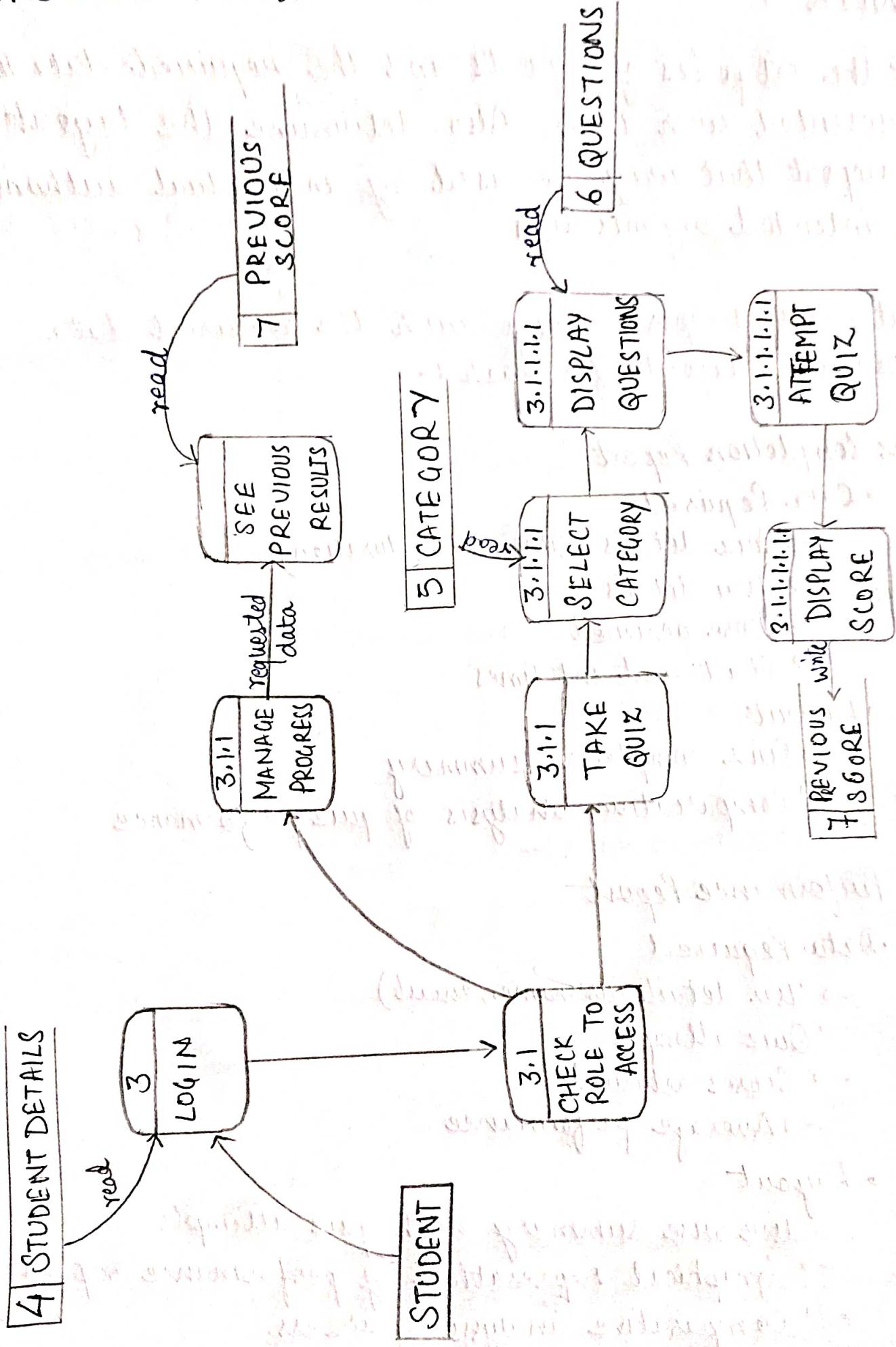
## 1. Admin Function



## 2. Teacher Process



### 3. Student Process.



## Lab - Work 7

List out the categories of reports and the required data to be represented with them. Also, determine the layout of the report that may be used by individual authorities of the intended organization.

The categories of reports along with the required data and potential layouts for each :-

### 1. Quiz Completion Report

- Data Required

- Quiz details (time, date, duration)
- User details
- Score achieved
- Start and end times

- Layout

- Quiz completion summary
- Comparative analysis of quiz performance

### 2. User Performance Report

- Data Required

- User details (username, email)
- Quiz attempts
- Scores achieved
- Average performance

- Layout

- User wise summary with quiz attempts
- Graphical representation of performance report
- Comparative analysis of users.

### 3. Question Analysis Report

- Required Data

- Question details (type, difficulty)
- User responses
- Correct / incorrect answer
- Average time spent per question

- Layout

- Analysis of each question type
- Heatmap of correct and incorrect responses
- Time distribution for each question

### 4. Security and Integrity Report

- Required Data

- User access logs
- Security incidents
- Quiz attempts and completion logs

- Layout

- Integrity check and validations
- Recommendations for improvement
- Summary of security related event

### 5. Feedback and Survey Report

- Required data

- User feedback on quizzes
- Rating and reviews

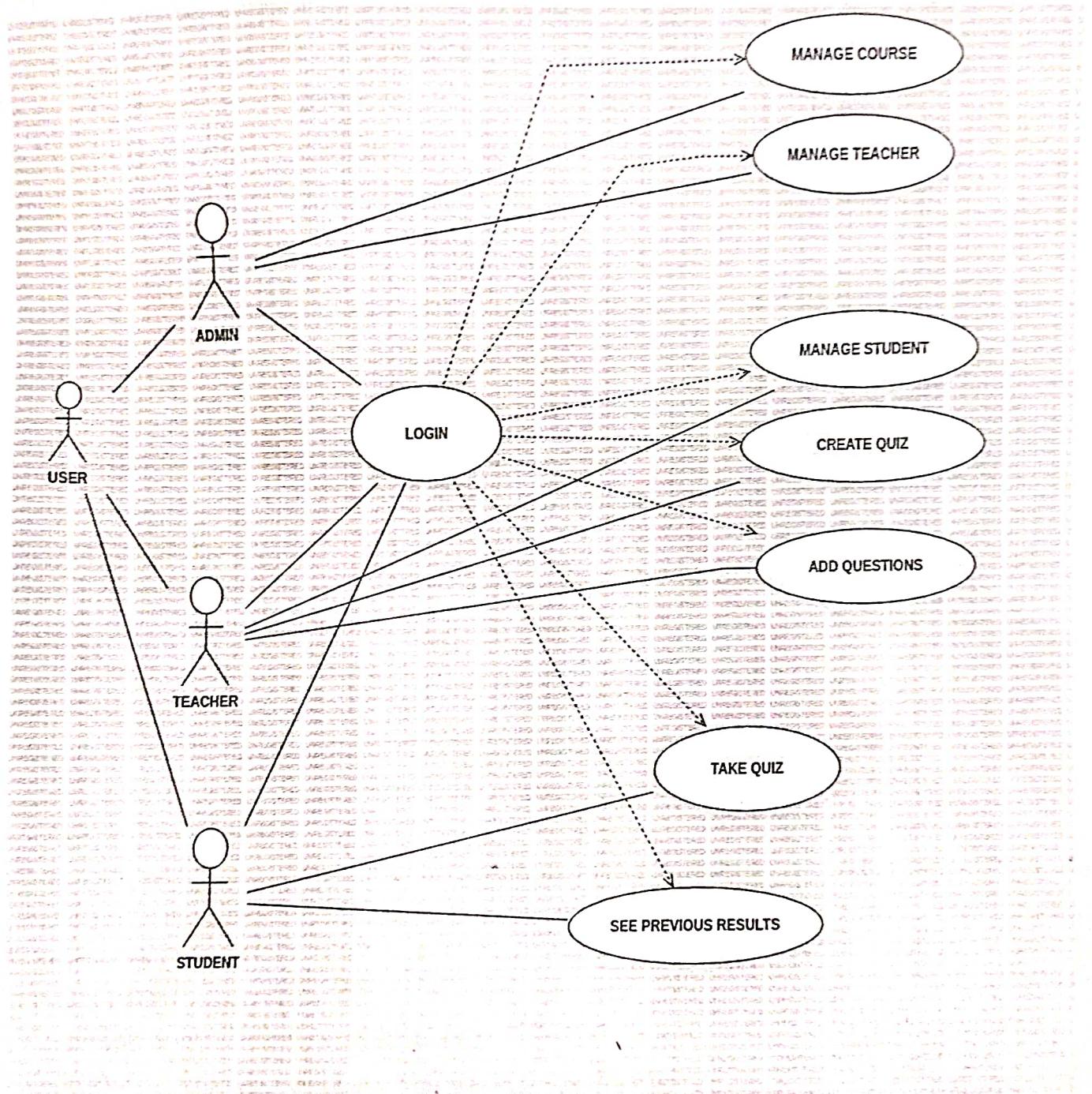
- Layout

- Summary of user feedback
- Visual representation of survey reports
- Common themes in suggestions

## LAB WORK – 8

While considering the scope of the individual functionalities draw the use case diagram for each of it.

The Use Case Diagram of the Online Quiz Application is



## **LAB WORK – 9**

On the basis of required validations map the different test cases to handle all possible critical cases which may arise during the life cycle of the software

### **1. User Registration and Authentication:**

#### a. Positive Test Cases:

- Verify successful user registration with valid details.
- Ensure users can log in with valid credentials.

### **2. Quiz Creation and Configuration:**

#### a. Positive Test Cases:

- Ensure quizzes can be configured with different question types, difficulty levels, and time limits.

#### b. Negative Test Cases:

- Verify the system handles errors gracefully when creating a quiz with incomplete information.

### **3. Quiz Taking:**

#### a. Positive Test Cases:

- Ensure users can start and complete quizzes successfully.
- Validate that the timer functions correctly during quiz attempts.

#### b. Negative Test Cases:

- Check if the system handles interruptions (e.g., network issues) during a quiz.
- Verify that users cannot submit quizzes after the time limit has expired.

### **4. Question Types and Validation:**

#### a. Positive Test Cases:

- Confirm that each question type (multiple choice, true/false, etc.) functions correctly.
- Verify that correct answers are accepted, and incorrect answers are rejected.

#### b. Negative Test Cases:

- Attempt to submit incomplete answers and ensure the system prompts for completion.
- Validate that the system handles unexpected inputs appropriately.

### **5. Scoring and Results:**

#### a. Positive Test Cases:

- Confirm accurate scoring for correct answers.
- Verify that users receive immediate feedback on quiz completion.

#### b. Negative Test Cases:

- Test scoring for partially correct answers.
- Ensure the system handles errors gracefully and does not crash during result processing.

## **6. Security and Privacy:**

### a. Positive Test Cases:

- Verify that user data is securely stored and transmitted.
- Confirm that only authorized users have access to sensitive information.

### b. Negative Test Cases:

- Attempt unauthorized access to user data.
- Test for potential security vulnerabilities, such as SQL injection or cross-site scripting.

## **7. Cross-browser and Cross-device Compatibility:**

### a. Positive Test Cases:

- Ensure the application works well on different browsers (Chrome, Firefox, Safari, etc.).
- Confirm usability on various devices (desktop, tablet, mobile).

### b. Negative Test Cases:

- Identify and address issues related to browser-specific incompatibilities.
- Check for responsiveness issues on different screen sizes.

## **8. Performance and Scalability:**

### a. Positive Test Cases:

- Test the application's response time under normal usage conditions.

### b. Negative Test Cases:

- Simulate heavy traffic to identify performance bottlenecks.
- Ensure the application gracefully degrades under extreme load conditions.

## **9. Accessibility:**

### a. Positive Test Cases:

- Confirm that the application adheres to accessibility standards (WCAG).
- Test with screen readers and ensure a positive user experience for individuals with disabilities.

### b. Negative Test Cases:

- Identify and address accessibility issues, such as missing alt text for images.
- Ensure that the application is navigable using keyboard controls.

These test cases cover a range of critical scenarios throughout the lifecycle of an online quiz application, from user registration to quiz completion and system maintenance. It's essential to continuously update and expand the test cases based on the evolving requirements and features of the application.

## LAB WORK - 10

Identify the classes. Classify them as weak and strong Study. classes and draw the class diagram for the specified Case

Let's identify some key classes for an online quiz application

Classes of Quiz Management System Class Diagram:

- Quiz Class : Manage all the operations of Quiz
- Participate Class : Manage all the operations of Participate
- Score Class : Manage all the operations of Score
- Academic Class : Manage all the operations of Academic
- Winner Class : Manage all the operations of Winner
- Prize Class : Manage all the operations of Prize

Classes and their attributes of Quiz Management System Class Diagram:

- Quiz Attributes : quiz\_id, quiz\_name, quiz\_type, quiz\_description
- Participate Attributes : participant\_id, participant\_name, participant\_mobile, participant\_email, participant\_username, participant\_password, participant\_address
- Score Attributes : score\_id, score\_name, score\_total, score\_type, score\_description
- Academic Attributes : academic\_id, academic\_name, academic\_type, academic\_description
- Winner Attributes : winner\_id, winner\_name, winner\_mobile, winner\_email, winner\_username, winner\_password, winner\_address
- Prize Attributes : prize\_id, prize\_winner\_id, prize\_name, prize\_type, prize\_description

Classes and their methods of Quiz Management System Class Diagram:

- Quiz Methods : addQuiz(), editQuiz(), deleteQuiz(), updateQuiz(), saveQuiz(), searchQuiz()
- Participate Methods : addParticipate(), editParticipate(), deleteParticipate(), updateParticipate(), saveParticipate(), searchParticipate()
- Score Methods : addScore(), editScore(), deleteScore(), updateScore(), saveScore(), searchScore()
- Academic Methods : addAcademic(), editAcademic(), deleteAcademic(), updateAcademic(), saveAcademic(), searchAcademic()
- Winner Methods : addWinner(), editWinner(), deleteWinner(), updateWinner(), saveWinner(), searchWinner()
- Prize Methods : addPrize(), editPrize(), deletePrize(), updatePrize(), savePrize(), searchPrize()

Class Diagram Image:

