“QUIZ MANIA”

Report-II

20 November 2015

Gaurav Rajput
&
Divyank Pahlazani

Table of Contents

1. Introduction

This 'QUIZ GAME' Project is designed for a question in which you can generate and manage a simple database for questions. The question number is automatically generated by the software and is stored in a binary file by the name 'QUESTION'. This database is basically used as a MASTER file to be used as a look-up table for information like Quiz Option, Sub Menu. In this software you can ask for any Menu on the basis of the Menu Question will be asked to the user & the user get 10 sec for answering the question. If the user is correct then he get 10 points on every question. If he is incorrect then he lost the game & game is closed.

1.1 Purpose

The sole intention behind the consideration of this Project is to generate and manage a simple database for question.
This project is developed considering “QUIZ” information keeping context of the customer in mind.

1.2 Scope

The system comprises of 2 basic users:

1. Administrator
2. User

Scope of Admin

- The Admin of the system will be able to add user i.e he will be able to add Login Id and Password of user in the Database.
- The admin will also be able to add questions and their options in the database.
- Only the Admin will have the access to the database.

Scope of User

- He will be able to login using login Id and Password.
- The user (usually the student) will be able to take the Quiz.

1.3 Definitions, Acronyms, and Abbreviations

- **GUI**: Graphical User Interface

  It is a type of user interface that allows users to interact with electronic devices with images rather than text commands.

- **Admin**: Administrator

  He is the main controller of the Quiz Management system. He has the authority to add/delete registered user.

- **DBMS**: Database Management System

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

- **C++**: C Plus Plus

It is the programming language in which the whole project will be made.

- **Usr:** User

The user will be the person who will take the quiz. Usually the user will be a student.

1.4 References

Learning UML, O'Reilly Publications: Sinan Si Alhir.
Software Engineering - A Practitioner's Approach: Roger S. Pressman.
NPTEL (nptel.iitm.ac.in) - Software Engineering: Rajib Mall.
Object Oriented Programming – E Balagurswamy.
Wikipedia – en.wikipedia.org/wiki.
Google – www.google.com 1.5 Overview
This subsection should:
(1) Describe what the rest of the SRS contains
(2) Explain how the SRS is organized.

2. General Description

The general factors that affect the product and its requirements are:

- Absence of C++ compiler.
- User are unable to understand the functionality.

2.1 Product Perspective

This subsection of the SRS puts the product into perspective with other related products or Projects.

2.2 Product Functions

This Software provides the following functionalities:

- All the users including admin can login into the system.
- The Admin can update/view/delete the courses and questions in the quizzes.
- The User can view the available courses and can attempt the Quizzes.
- The User and Admin both can view the Score Board after the Quiz is conducted.

2.3 User Characteristics

- User should know how to operate a computer system and its platform.
- User should know some basics of C++ programming language and how to use its Turbo C compiler.

- User should be literate enough to understand the functionality of the software.

2.4 General Constraints

Main constraints for this system is the programming language i.e. restriction to use only C++ and Turbo C compiler.
This Software has the restriction to use file system only for storing and managing data, use of database system and graphical user interface (GUI).

2.5 Assumptions and Dependencies

- It is assumed that the computer system has DOS or windows or its above version to run this Software and all the hardware and software requirements are provided to this system.
- This system runs on the assumption that the admin had entered all the information.
- Software entirely depends upon the File Handling, if it is not provided then this system will not work properly.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 Hardware Interfaces

- 1.7 GHz Pentium IV
- 64 MB RAM
- A Hard-drive space for storing questions.

3.1.2 Software Interfaces

- Operating System: Windows XP Service Pack 1
- TurboC3 compiler.

1. **Front End**

   - **C++**

     C++ is an object oriented programming language. It is a superset of c language. C++ adds a number of object oriented features such as object inheritance, function overloading, operator overloading to C. C++ is a versatile language for handling very large programs.

   - **Uses of C++**

     C++ can be used to build a variety of system such as editor, compiler, database, communication system.
     C++ is easily maintainable and expandable.

2. **Back end:**

   - Any file for storing database.
     Many real life problems handle large volumes of data so we need to use some

devices such as floppy disk or hard disk to store a data. The data is stored in these devices using a concept of files. A file is a collection of data stored in a particular area on a disk. Programs can be designed to perform read and write operations on these files.

3.1.3 Communications Interfaces

File Handling

3.2 Functional Requirements

This Software provides the following functionalities:

- All the users including admin can login into the system.
- The Admin can update/view/delete the courses and questions in the quizzes.
- The User can view the available courses and can attempt the Quizzes.
- The User and Admin both can view the Score Board after the Quiz is conducted.

3.2.1 Introduction

Basic functional requirements is a working system with DOS/Windows OS.

3.2.2 Inputs

Inputs will be taken from User to select the course and attend the Quiz.

3.2.3 Processing

For processing TC++ with TC++ compiler is required.

3.2.4 Outputs

As an outputs Score Card will get displayed.

3.2.5 Error Handling

None

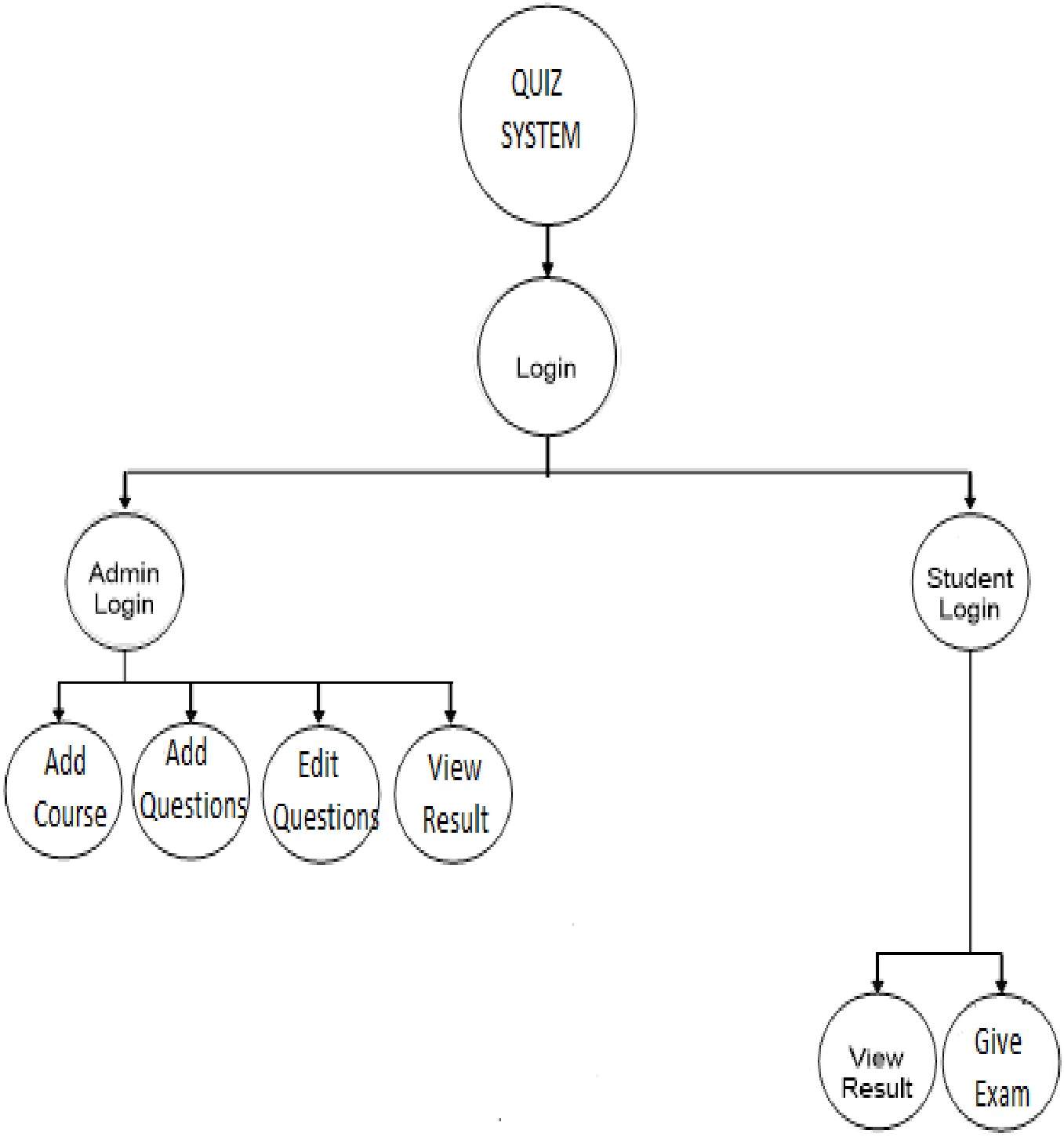
3.3 Other Requirements

Catchall section for any additional requirements.

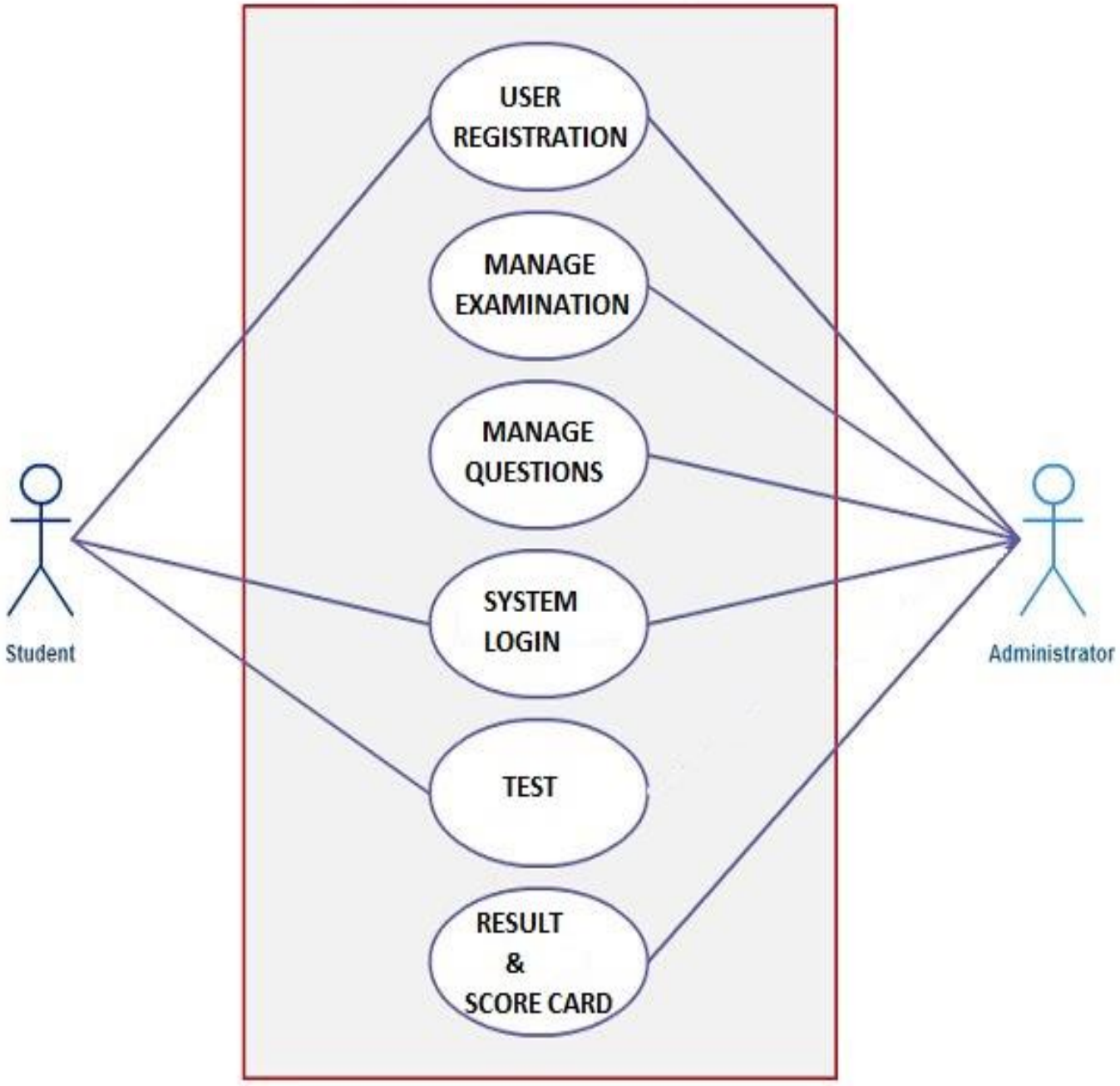
4. Analysis Models

This section contains all the related diagrams (i.e. Use Case Diagram, Flow Chart, ER Diagram) with their description.

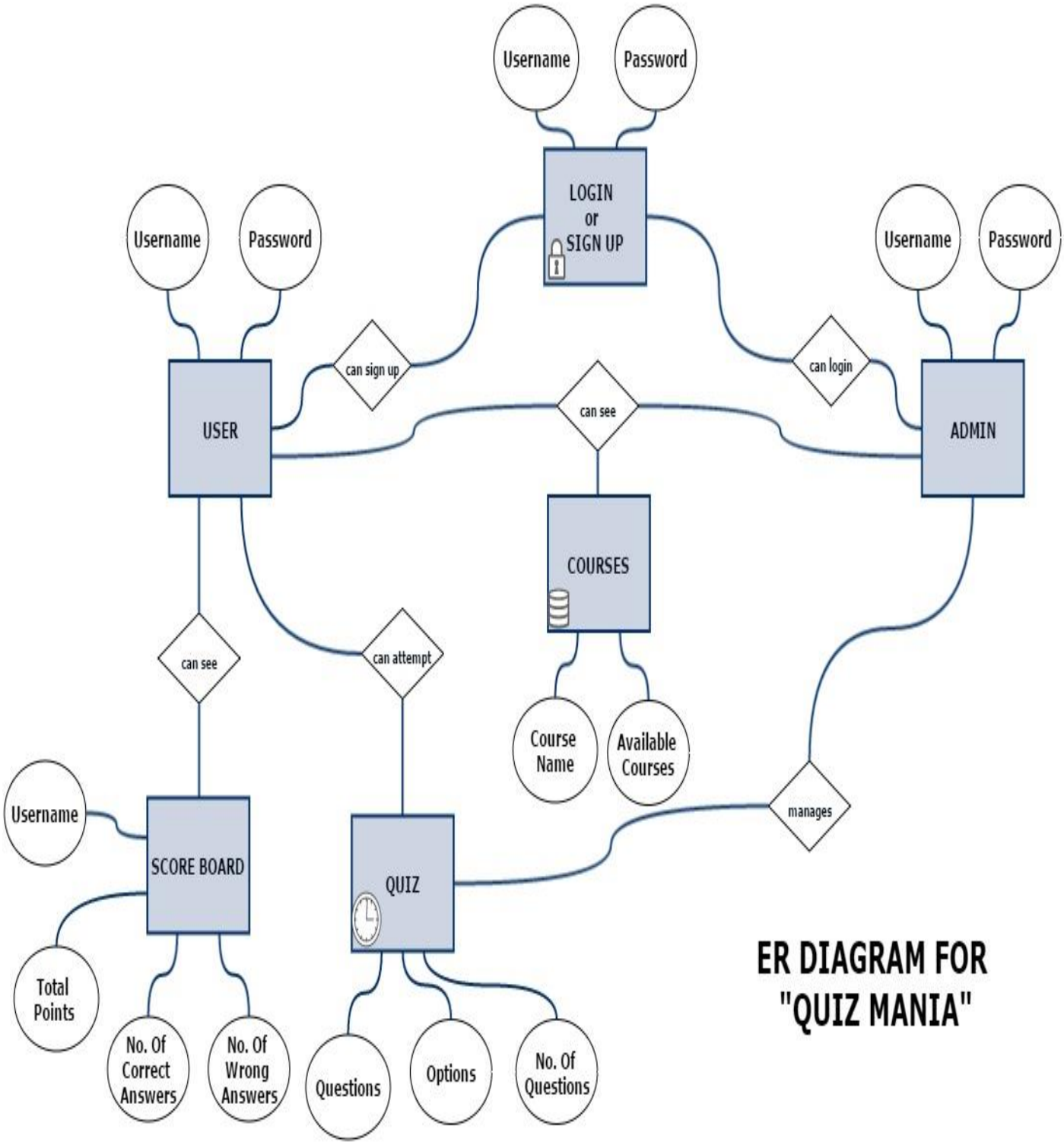
4.1 Flow Chart



4.2. Use Case Diagram

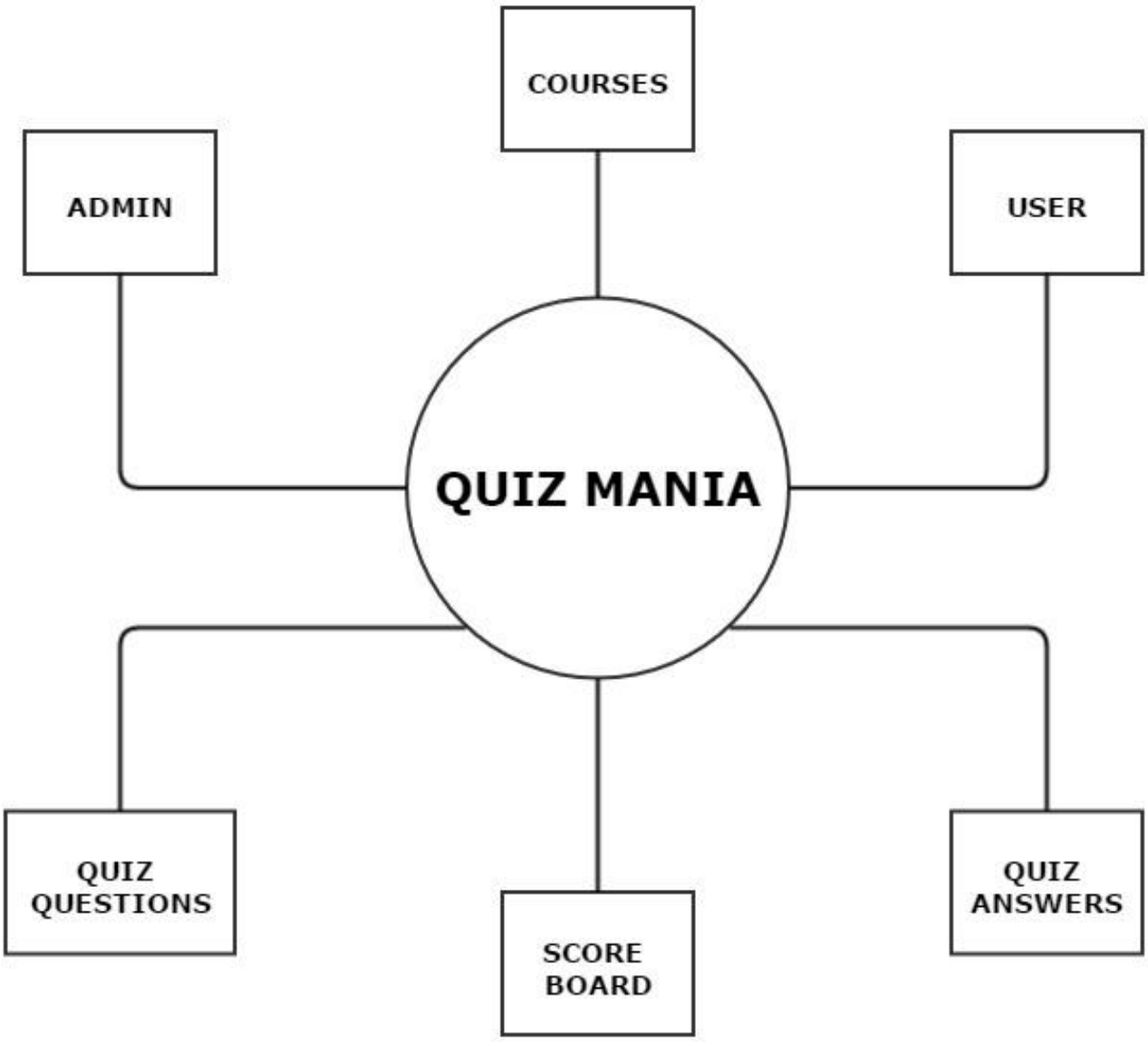


4.3. ER Diagram

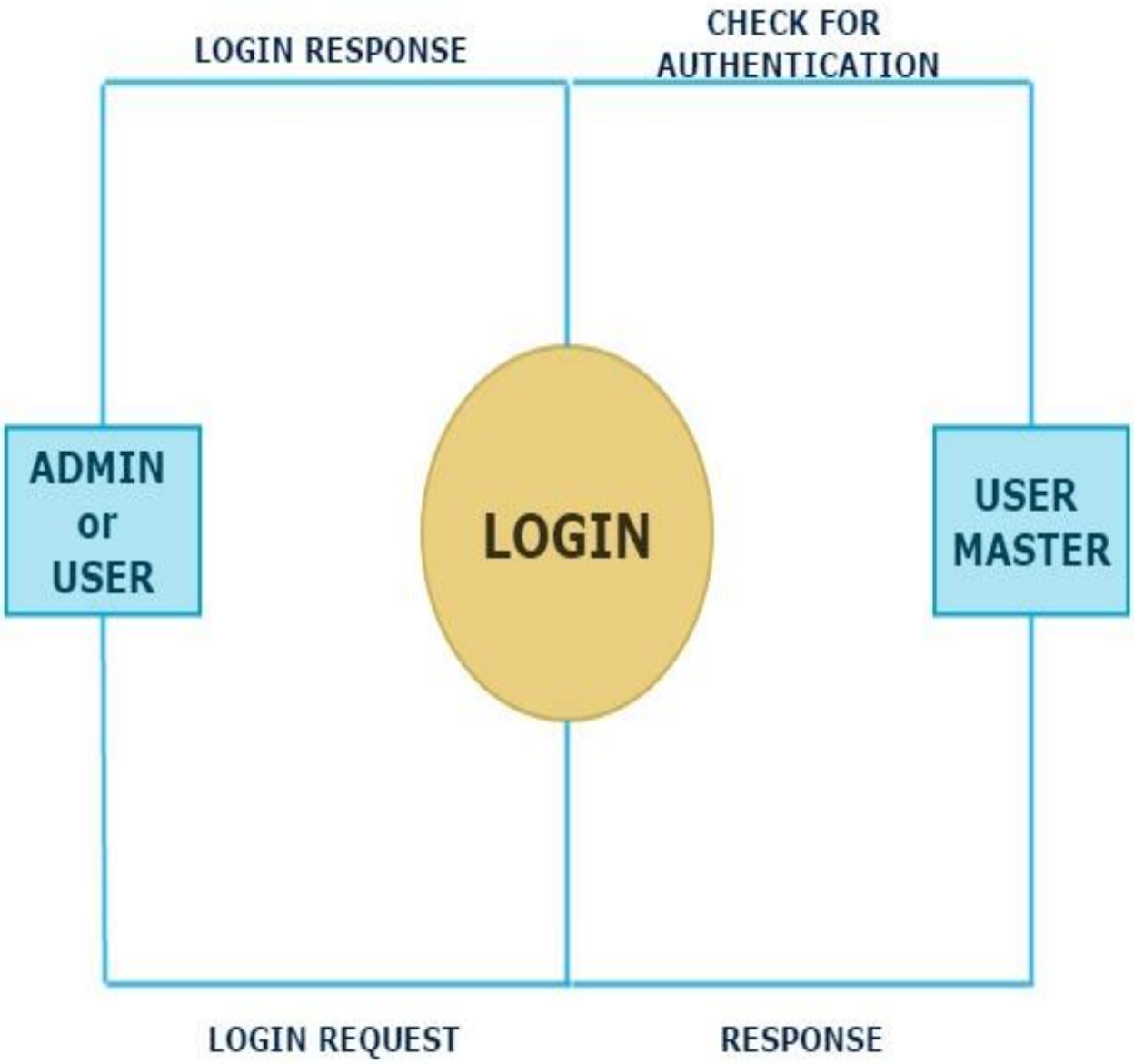


4.4 DATA FLOW DIAGRAM

4.4.1 Level 0 DFD



4.4.2 Level 1 DFD



4.5 Module Description

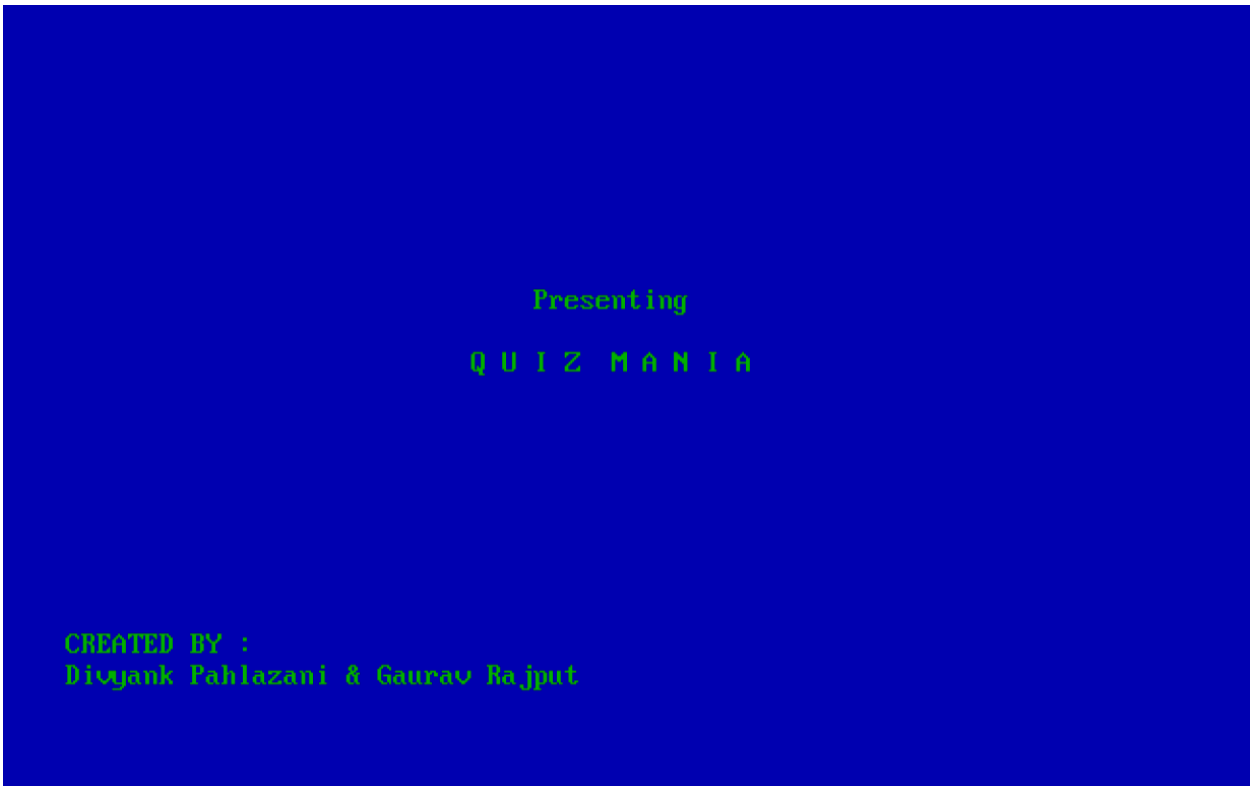
4.5.1 Login panel

In this module the login system is defined which is used to check the authenticity of member.
This module will search for the respective id and password matches in the database. If the member is valid then according to the member type (i.e. User/Admin) the panel will be open for the further functionality, if the member is not valid then it shows an error of authentication.
This module is accessible to Users and Admin.

4.5.2 Admin panel

This panel is open after the successful login of the admin. This module will provide the functionality of update/delete/modify the Courses and Quizzes.
This module is accessible to Admin only.

5. Screenshots




Available Courses
1. JAVA
2. HTML
3. Python
4. DAA


QUIZ MANIA
»Play
Add Questions
Edit

A. Appendices

A.1 Appendix 1 (Use Case Diagram)

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

- **Use Case**

  A use case is a list of action or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal.

- **Actor**

  An actor in the Unified Modeling Language (UML) specifies a role played by a user or any other system that interacts with the subject.

  An Actor models a type of role played by an entity that interacts with the subject (e.g. by exchanging signals and data), but which is external to the subject.

  Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e. “role”) of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.