

A MINI PROJECT REPORT

On

Quiz Mania

Submitted by

**Gaurav Rajput
Roll No: 131500117**

**Divyank Pahlazani
Roll No: 131500107**

Department of Computer Engineering & Applications
Institute of Engineering & Technology



**GLA University
Mathura- 281406, INDIA**

Contents

Certificate	i
Abstract	ii
Acknowledgments	iii
1. Introduction	1
1.1 Overview	1
1.2 Objective	2
2. Software Requirement Analysis	3
2.1 Define the problem	3
2.2 Modules and their functionalities.....	3
2.2.1 LogIn Module.....	3
2.2.2 Quiz Module.....	3
3. Software Design	4
3.1 Data Flow Diagram.....	5
3.1.1 Level 0 DFD.....	5
3.1.2 Level 1 DFD.....	6
3.2 UML Diagram.....	6
3.2.1 Class Diagram	6
3.2.2 Use Case Diagram	7
3.3 Database Design.....	8
3.3.1 ER Diagram	8
4. Testing	9
4.1 Introduction to Testing.....	9
4.2 Test Result.....	11
5. Implementation and User Interface	13
5.1 Implementation and User Interface.....	13
References/Bibliography	16
Appendices	17



Department of computer Engineering and Applications
GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan,
Mathura – 281406

Declaration

I hereby declare that the work which is being presented in the Mini-Project “**Quiz Mania**”, in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of my own work carried under the supervision of **Mr. K. G. Sharma, Assistant Professor, GLA University, Mathura.**

Name of Candidate: Gaurav Rajput

Roll. No.: 131500117

Course: B. Tech.

Year: 3rd

Semester: V

Signature of Candidate:

Name of Candidate: Divyank Pahlazani

Roll. No.: 131500107

Course: B. Tech.

Year: 3rd

Semester: V

Signature of Candidate:

Submitted To: Mr. K. G. Sharma

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech. project undertaken during the fifth semester of B. Tech. This project is itself an acknowledgement to the inspiration, drive and technical assistance contributed to it by 2 individuals. We take this opportunity to thank all those magnanimous persons who rendered their full services to our work.

It's with lot of happiness we are expressing gratitude to our guider Mr. K. G. Sharma, Assistant Professor in Computer Engineering and Application Dept., for timely and kind help, guidance and for providing us with most essential materials required for the completion of this project. We are very thankful for his indomitable guidance. This inspiration up to the last moment had made things possible in a nice manner.

We thank Mr. K. G. Sharma, Assistant Professor in Computer Engineering and Application Dept. and to Mr. Praveen Mittal, Assistant Professor in Computer Engineering and Application Dept. for the cooperation extended for the successful completion of the project.

Finally, we thank each and every one who helped to complete our project work with their cordial support.

CHAPTER 1

INTRODUCTION

1.1 Overview

There are numerous products (desktop or web Software) that allows users to take quiz. The development of this software is centred on the interactive way of taking quiz and that too quickly, effortlessly and with ease. It allows user to learn while playing. This project is better than the available Quiz Software products available in the market as it is user-friendly.

We knew that no one model would provide all the requirements and this did not matter as long as each system met the recognised standards for proper Quiz Taking System.

From the outset, we recognised that developing a suitable software which met the needs of both the faculty and the student. Assume you just need to conduct a Quiz Competition and check the knowledge of the candidate and you don't have enough time to assess candidates individually then Quiz Mania software can come handy.

This 'QUIZ MANIA' 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. 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 20 sec for answering the question. If the user is correct then he get 10 points on every question. If he is incorrect then 0 Points are awarded and at the end a score board is generated.

In this software there are two main modules:-

1. Taking Quiz.
2. Adding/Modifying Question in Database.

The software basically performs the various tasks in the following manner:-

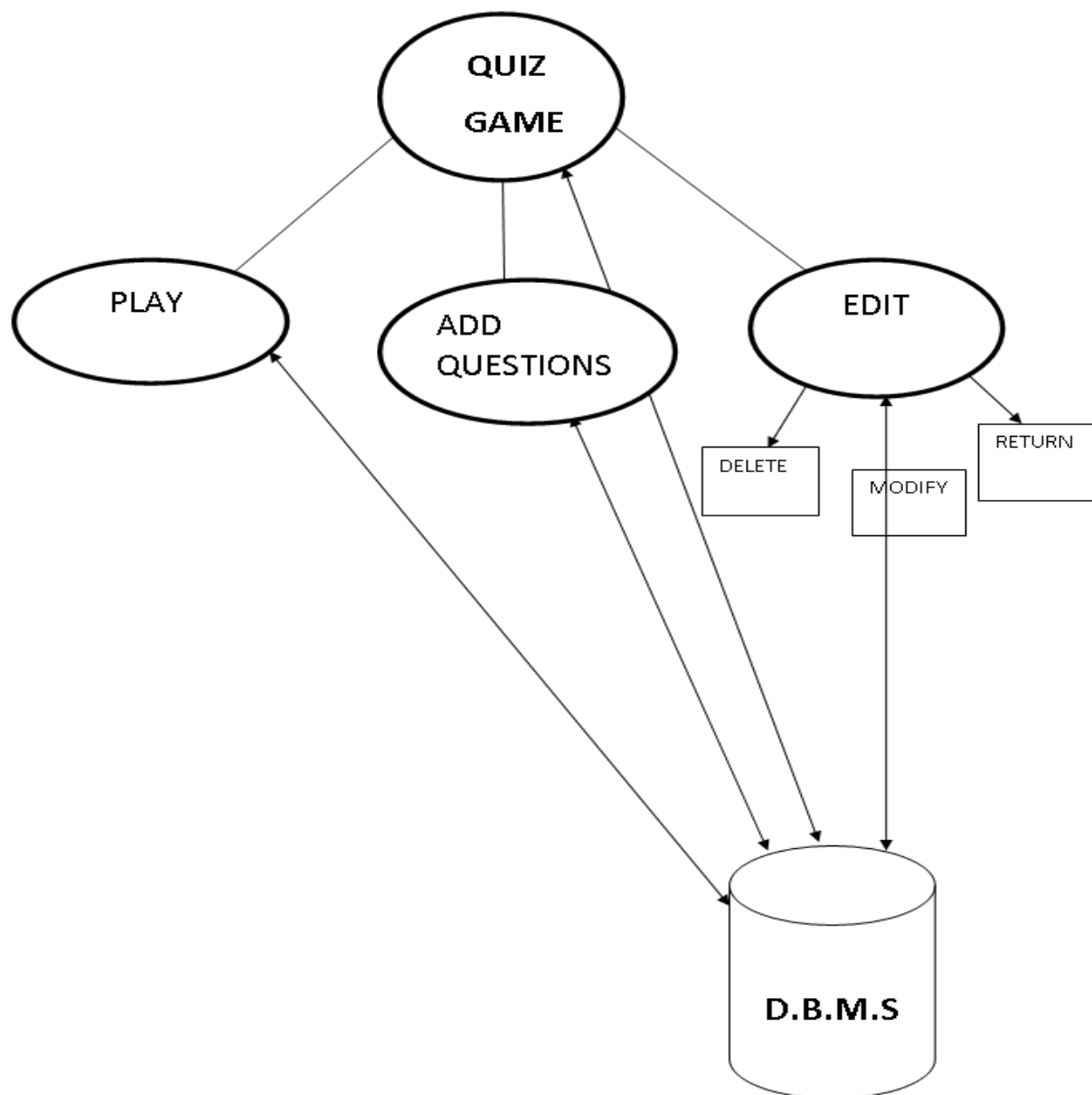


Figure 1.1: General Diagram

1.2 Objective

- To reduce paper work, and provide more accurate and speedy operation.
- Exam will save time.
- It will save manual work (i.e. Calculation of marks, Supervising etc.)
- Provide Security.

- Immediately Student can see their result.
- To prepare question bank so that in future it can be used to design exam.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATIONS AND ANALYSIS

2.1 Problem

In our university we don't have any offline software which can conduct a quiz competition in efficient and easy way for saving the precious time of our faculties and students.

From the outset, we recognized that developing a suitable software which met the needs of both the faculty and the student. Assume that you just need to conduct a Quiz Competition and you don't have enough time to take a pen and paper based test and evaluate it manually, this is where Quiz Mania software comes handy for you.

In order to overcome the issue of spending more time on evaluation of pen and paper based Test, this project has been developed.

2.2 Functionalities

The project is divided basically into two sections -:

1. LogIn Module
2. Quiz Module

2.2.1 LogIn Module

This module provides the user, the facility to login or register himself as a participant in a Quiz.

The LogIn module provides the following functionality -:

- Enter a username and password.
- A user can add a new section with a new name.
- A user can modify the password later if he/she wants to.

2.2.2 Quiz Module

Once the user passed the authentication or registered himself, then he/she can start attending the quizzes from the courses available. This will cover following choices:-

- Select a course.
- Editing the Questions and Courses (Admin privileges).
- Generation of Score Board.

CHAPTER 3

Software Design

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. It tries to specify how to fulfil the requirements mentioned in SRS.

Software Design Levels

Software design yields three levels of results:

- **Architectural Design** - The architectural design is the highest abstract version of the system. It identifies the software as a system with many components interacting with each other. At this level, the designers get the idea of proposed solution domain.
- **High-level Design**- The high-level design breaks the ‘single entity-multiple component’ concept of architectural design into less-abstracted view of sub-systems and modules and depicts their interaction with each other. High-level design focuses on how the system along with all of its components can be implemented in forms of modules. It recognizes modular structure of each sub-system and their relation and interaction among each other.
- **Detailed Design**- Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the previous two designs. It is more detailed towards modules and their implementations. It defines logical structure of each module and their interfaces to communicate with other modules.

3.1 Data Flow Diagram

3.1.1 Level 0 DFD

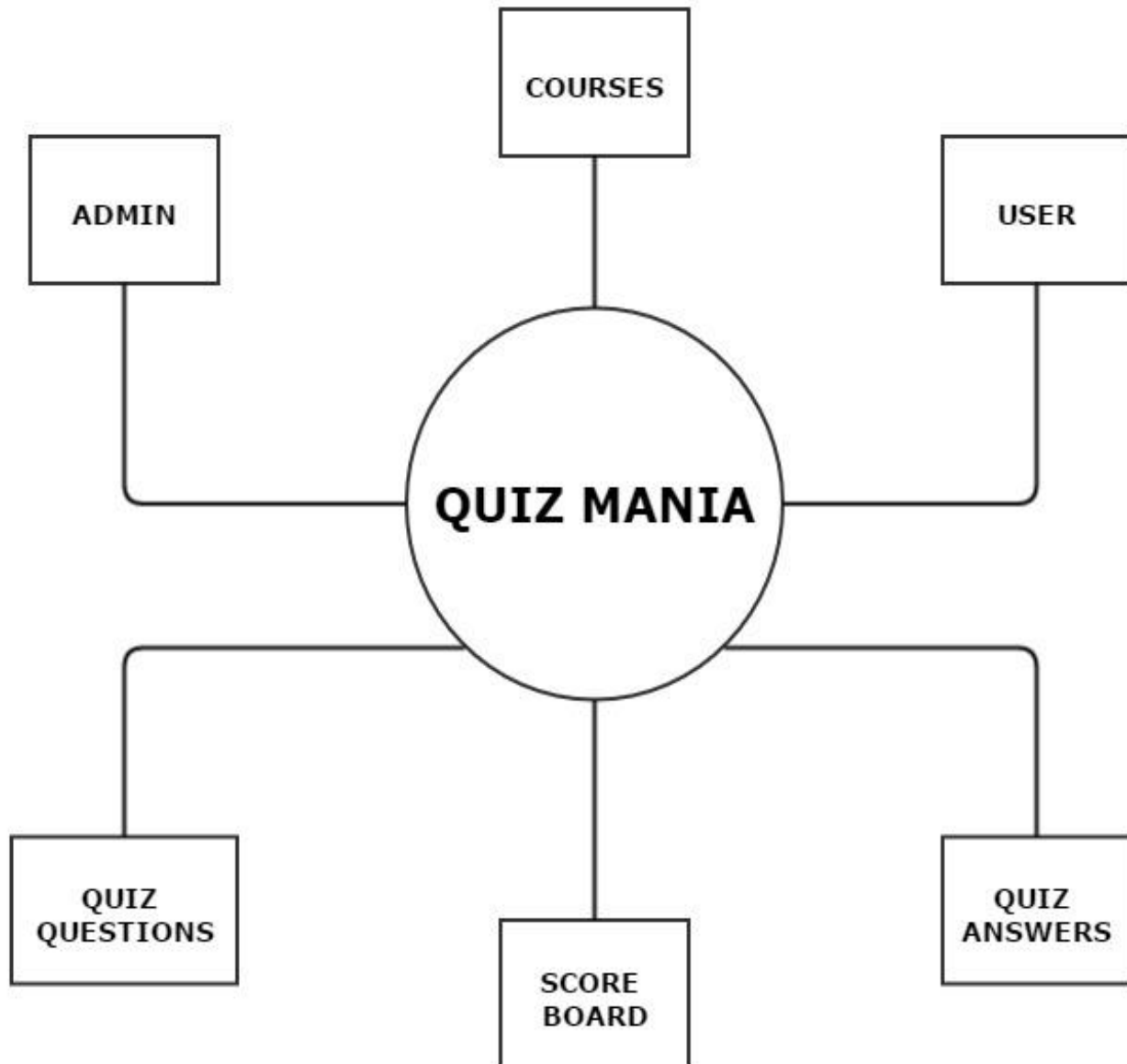


Figure 3.1: Context Level Diagram

3.1.2 Level 1 DFD

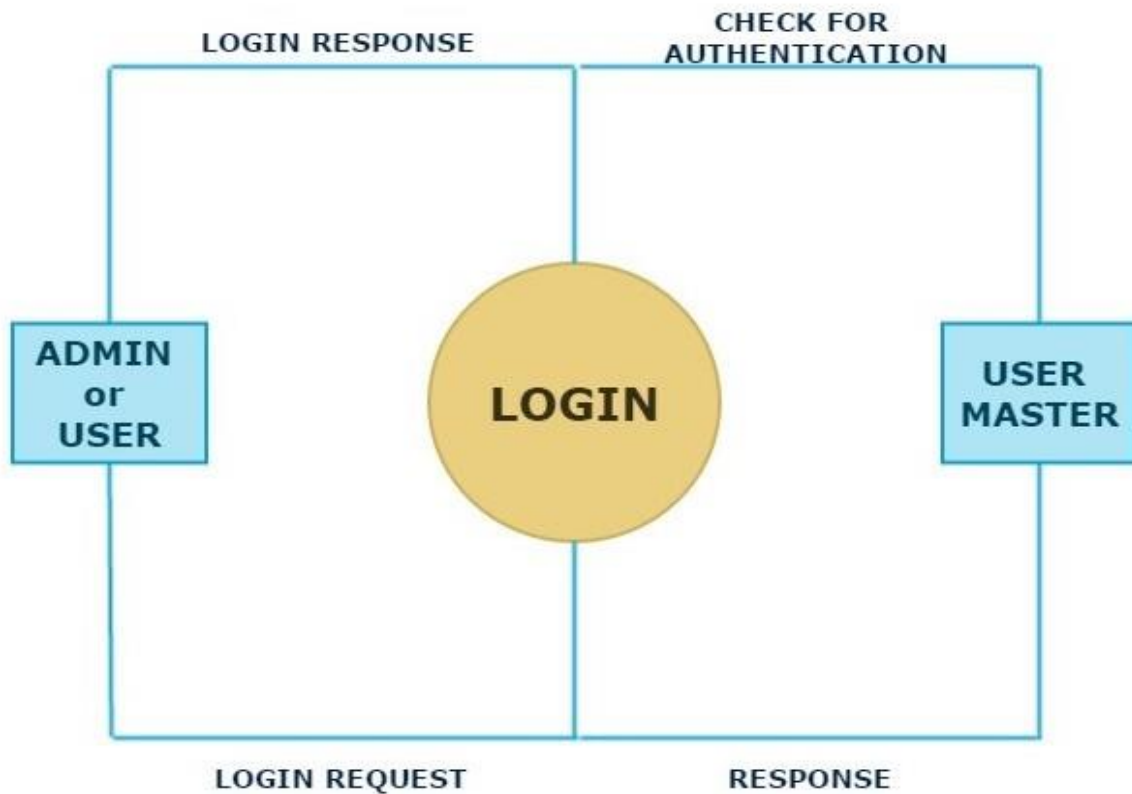
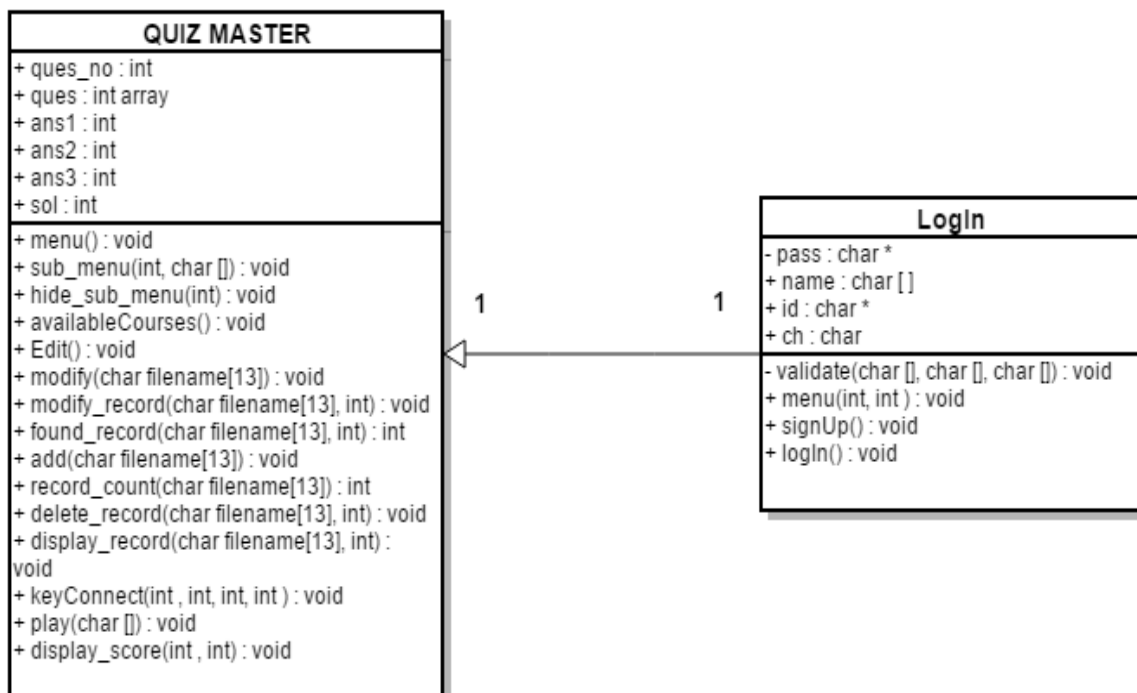


Figure 3.2: Level 1 DFD

3.2 UML Diagram

3.2.1 Class Diagram



3.2.2 Use Case Diagram

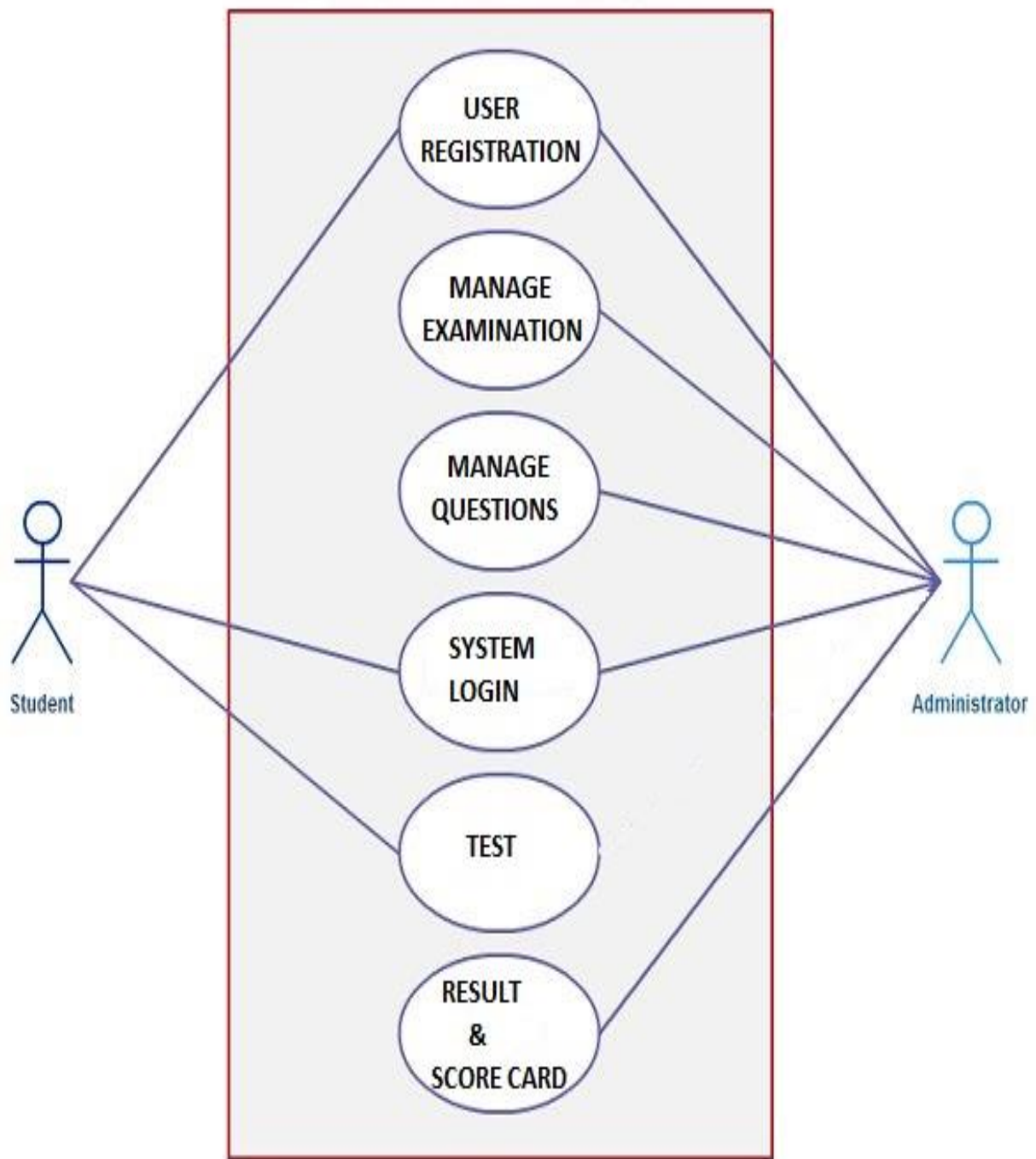
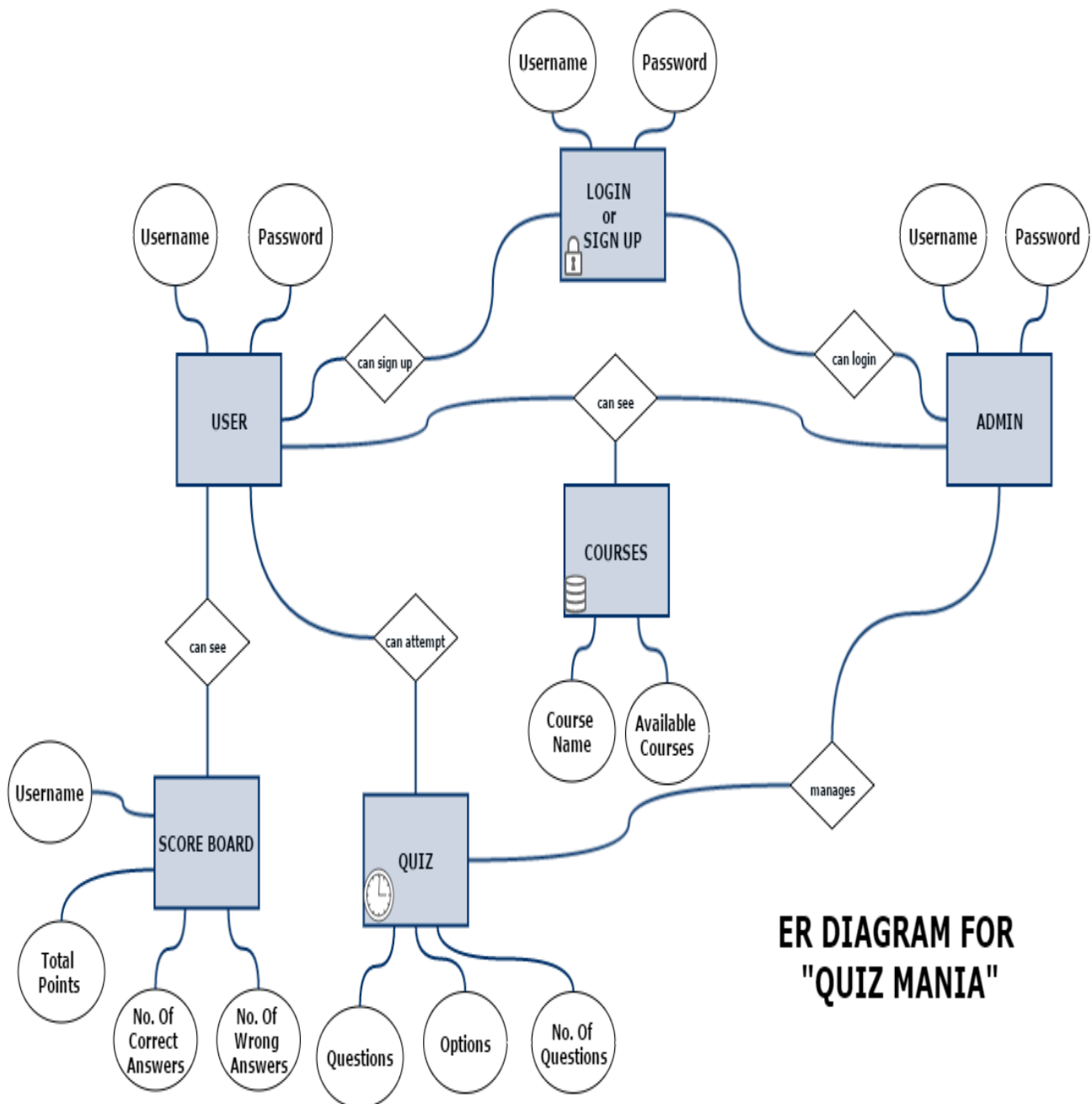


Figure 3.1: Use Case Diagram

3.3 Database Design

3.3.1 ER Diagram



**ER DIAGRAM FOR
"QUIZ MANIA"**

Figure 3.5: E-R Diagram

CHAPTER 4

TESTING

4.1 Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting of a system using various test data. Preparation of the test data plays a vital role in the system testing. After preparation the test data, the system under study is tested those test data. Errors were found and corrected by using the following testing steps and corrections are recorded for future references. Thus, series of testing is performed on the system before it is already for implementation.

The development of software systems involves a series of production activities where opportunities for injection of human errors are enormous. Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified as well as in later design and development stages. Because of human in ability to perform and communicate with perfection, software development is followed by assurance activities.

Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. And of course it includes assurances that the system meets the specification and the requirements for its intended use and performance. The various levels of quality assurance are described in the following sub sections.

System Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specifications, design and coding. The testing phase involves the testing of system using various test data; Preparation of test data plays a vital role in the system testing. After preparation the test data, the system under study is tested.

Those test data, errors were found and corrected by following testing steps and corrections are recorded for future references. Thus a series testing is performed on the system before it is ready for implementation.

The various types of testing on the system are:

- Unit testing
- Integrated testing
- Validation testing
- Output testing
- User acceptance testing

Unit testing

Unit testing focuses on verification effort on the smallest unit of software design module. Using the unit test plans. Prepared in the design phase of the system as a guide, important control paths are tested to uncover errors within the boundary of the modules. The interfaces of each of the modules under consideration are also tested. Boundary conditions were checked. All independent paths were exercised to ensure that all statements in the module are executed at least once and all error-handling paths were tested. Each unit was thoroughly tested to check if it might fall in any possible situation. This testing was carried out during the programming itself. At the end of this testing phase, each unit was found to be working satisfactorily, as regarded to the expected out from the module.

Integration Testing

Data can be across an interface one module can have an adverse effect on another's sub function, when combined may not produce the desired major function; global data structures can present problems. Integration testing is a symmetric technique for constructing tests to uncover errors associated with the interface. All modules are combined in this testing step. Then the entire program was tested as a whole.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the consumer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions

exists.

- The function or performance characteristics confirm to specification that are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since a system is useful if it does not produce the required output in the specific format required by them tests the output generator displayed on the system under consideration. Here the output is considered in two ways: - one is onscreen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. As far as hardcopies are considered it goes in terms with the user requirement. Hence output testing does not result any correction in the system.

User Acceptance Testing

User acceptance of the system is a key factor for success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required.

4.2 TEST RESULT: UNIT TESTING

Front Module

Sr. No.	Test Case	Excepted Result	Test Result
1	Login Form: login as a Admin or Participant	Software should display Login window	Successful

2	Invalid Details	Error Message and then the Software should again display the choice.	successful
---	-----------------	--	------------

Sign Up Module

Sr. No.	Test Case	Excepted Result	Test Result
1	Entering the name	Should move to the next field.	Successful
2.	Entering the Password	Should move to the next field.	Successful
3.	Entering the Type of User (Admin/student)	Should move to the next field.	Successful

CHAPTER 5

Implementation and User Interface

The project implementation starts with login or registration of the user, and then based on the type of user it can either attend the quiz of a particular course or even edit the questions or courses available (Admin privileges).

A user can also save the changes made by him in the database.

The project start with an administrator permission dialog box. After conforming from the user, if it is yes only then software will start. Then the user will only be able to attend the quizzes.

5.1 Entering the Details of user

The project implements by opening the login screen.

The User will provide some mandatory details such as username, password and type of user (Guest or Admin).

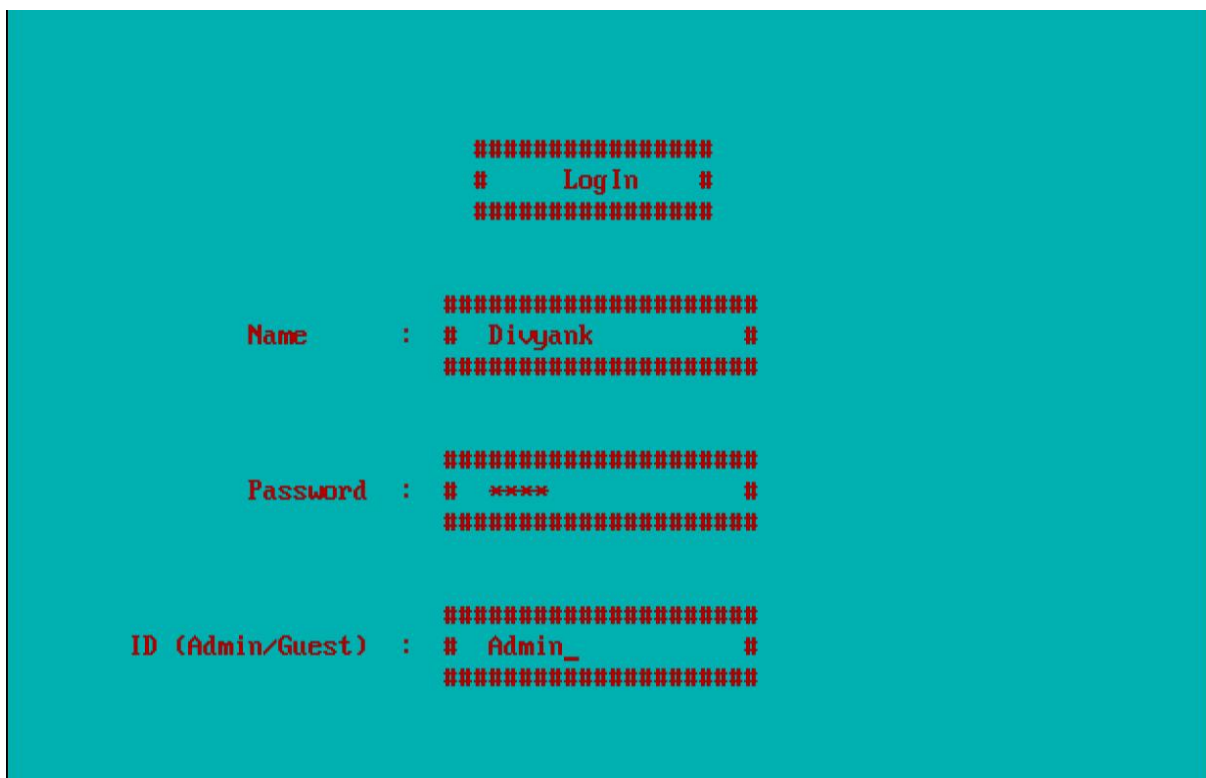


Figure 5.1: LogIn Selection

If the details provided by user were correct i.e. user “successful login”. Now User can select a course of his choice and start attempting the quiz or add the questions to the selected course.

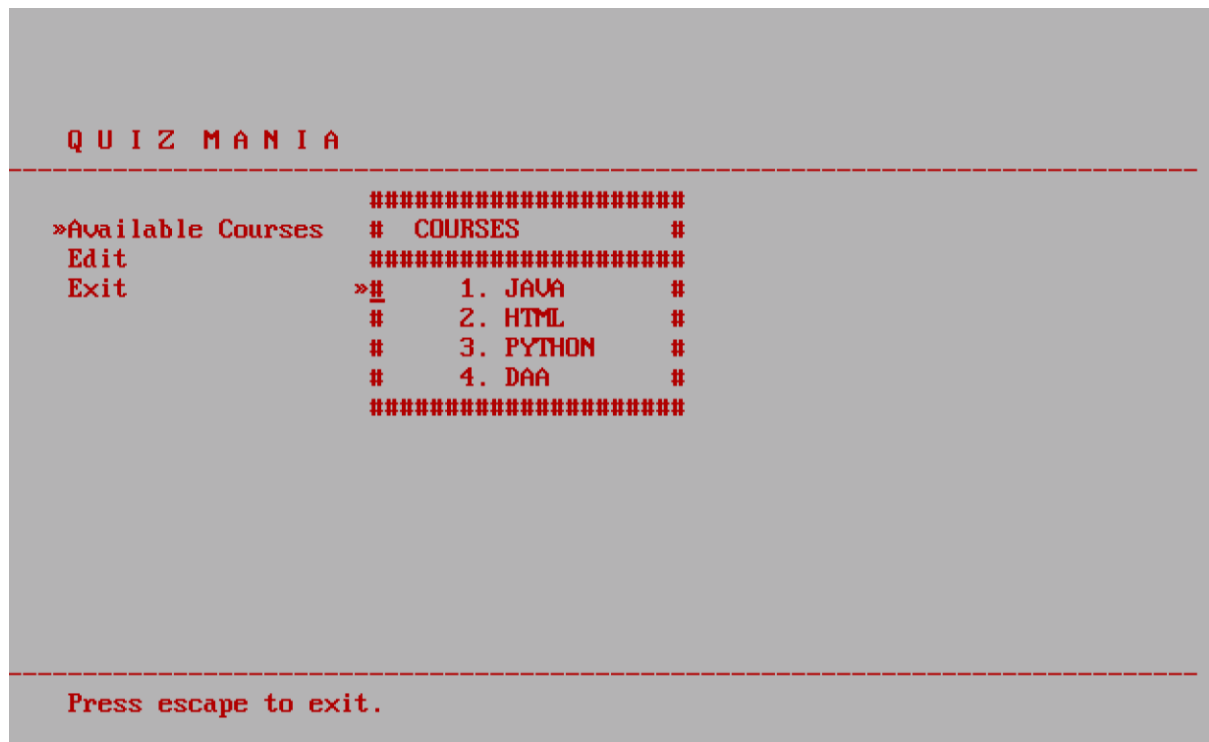


Figure 5.2: Main Menu

After selecting a particular course he can start attempting the quiz. User will get 20sec to answer a question otherwise error message “Sorry! Times up” will get displayed.

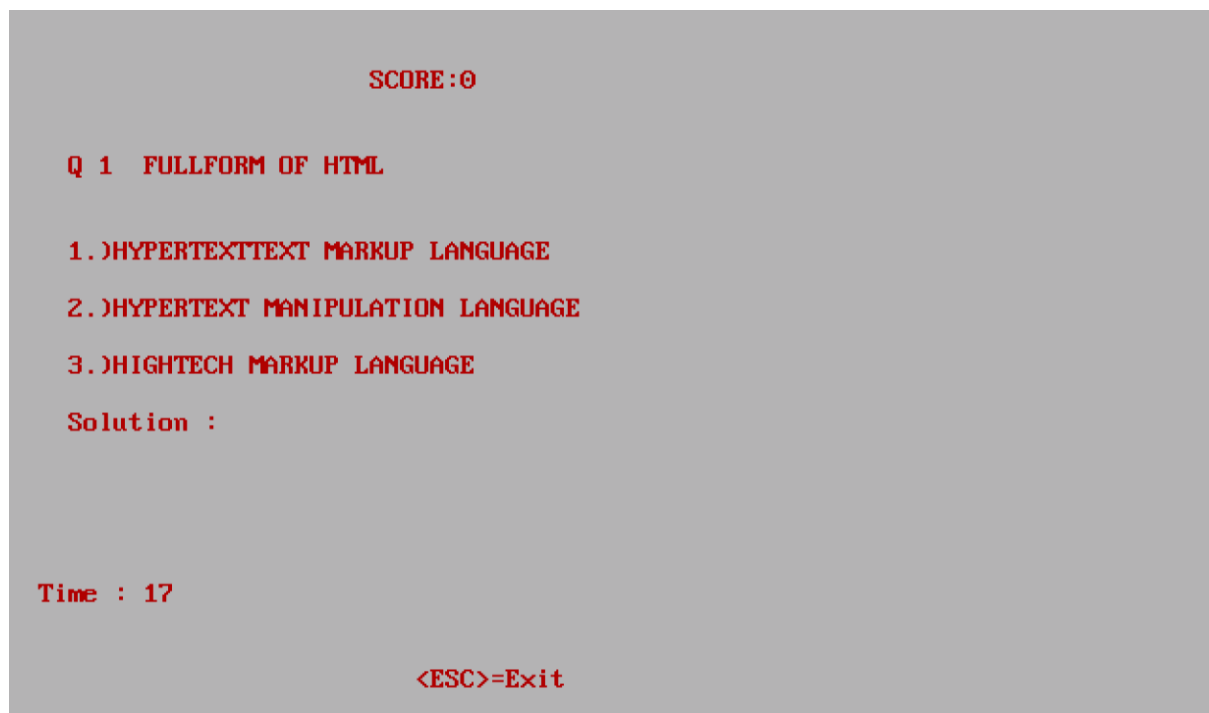


Figure 5.3: Attempting Quiz

Finally after attempting all the questions Score Board will get displayed.



Figure 5.4: Score Board

If user login as Admin he can also add questions to a course.

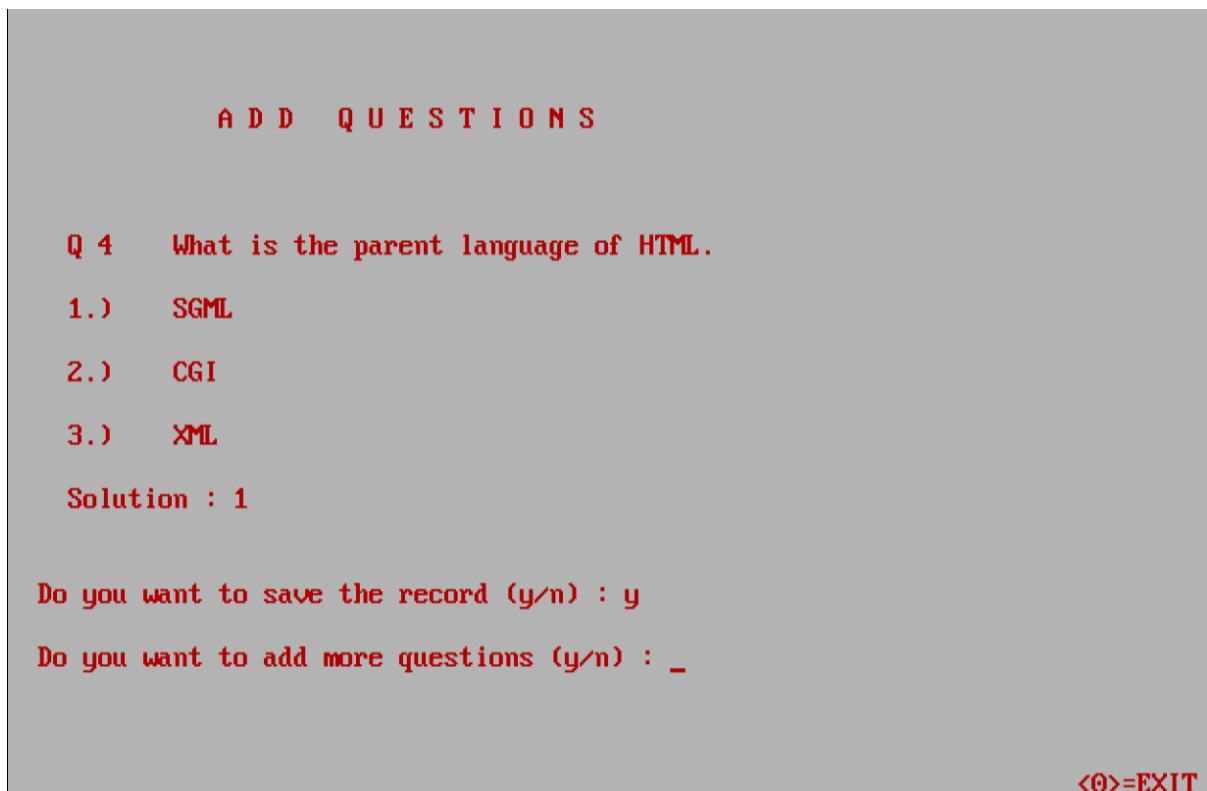


Figure 5.5: Adding a question

References/Bibliography

Text Books:

- Object Oriented Programming with C++ (5th edition) by E Balagurusamy, Tata McGraw- Hill.
- Software Engineering (3rd edition) by K. K Aggarwal and Yogesh Singh © New Age International Publishers, 2007.

Online References:

- Wikipedia - www.wikipedia.com
- Tutorials Point - www.tutorialspoint.com
- Cplusplus.com - www.cplusplus.com

Appendices

Coding Templates

Validation Module:

```
void LogIn :: validate(char n[], char p[], char id[]){
    char userName[15], pass[15], identity[10];
    ifstream read("key.dat");
    read >> userName; read >> pass; read >> identity;
    cout<<identity;
    if(strcmp(n, userName) == 0 && strcmp(p, pass) == 0 && strcmp(identity, id) == 0){
        Menu menu;
        participant = n;
        menu.welcome(n);
        Quiz quiz;
        quiz.menu();
    }
    else{
        gotoxy(30,25); cout<<"Invalid User";
        menu(7, 6);
    }
}
```

Score Board Module:

```
void Quiz :: display_score(int played, int correct){
    clrscr() ;
    textbackground(WHITE) ; textcolor(BLACK+BLINK) ;
    gotoxy(5,4) ; cprintf("          ") ;
    gotoxy(9,4) ; cprintf("S C O R E   B O A R D") ;
    textbackground(BLACK) ; textcolor(LIGHTGRAY) ;
    gotoxy(5,7) ; cout <<"Player's Name : " <<participant ;
    gotoxy(5,9) ; cout <<"questions Attempted : " <<played ;
    gotoxy(5,11) ; cout <<"Correct answer      : " <<correct ;
    gotoxy(5,13) ; cout <<"Wrong answer       : " <<played-correct ;
    int score = correct * 10 ;
    gotoxy(5,16) ; cout <<"Score : " <<score ;
    int tplayed = played * 10 ;
    float percentage = ((score*100)/tplayed) ;
    gotoxy(5,18) ; cout <<"Percentage : " <<percentage ;
    getch() ;
}
```

Attempt Quiz Module:

```
void Quiz :: play(char filename[13]){
    clrscr() ;
    char t_sol, ch ; int correct=0, played=0, valid ; fstream file ;
    file.open(filename,ios::in) ; file.seekg(0,ios::beg) ;
    while (file.read((char *) this, sizeof(Quiz))) {
        clrscr() ;
```

```

gotoxy(25,3) ; cout <<"SCORE:" <<correct*10 ;
valid = 1 ;
gotoxy(28,24) ; cout <<"<ESC>=Exit" ;
gotoxy(5,6);  cout<<"Q " <<ques_no<<" " <<ques;
gotoxy(5,9);  cout<<"1.)" <<ans1;
gotoxy(5,11); cout<<"2.)" <<ans2;
gotoxy(5,13); cout<<"3.)" <<ans3;
gotoxy(5,15); cout<<"Solution : ";
int i=0, j=0, time=20 ;
do{
    valid = 1 ;
    gotoxy(3,21) ; cout <<"Time : " <<time ;
    gotoxy(28,12) ;
    do{
        delay(1) ;
        i++ ; j++ ;
        if (j == 600){
            j = 0 ; time-- ;
            gotoxy(11,21) ; cout <<" " ;
            gotoxy(10,21) ; cout <<time ;
            gotoxy(28,14) ;
        }
    } while (!kbhit() && i<12000) ;
    if (i >= 12000){
        played++ ;
        sound(500) ; delay(100) ; nosound() ;
        gotoxy(13,16) ; cout <<"Sorry Time up!" ;
        gotoxy(1,1) ; getch() ;
        if (played)
            display_score(played,correct) ;
        return ;
    }
    gotoxy(3,21) ; cout <<" " ;
    gotoxy(28,12) ; t_sol = getche() ;
    if (t_sol == 27){
        if (played)
            display_score(played,correct) ;
        return ;
    }
    if (t_sol != '1' && t_sol != '2' && t_sol != '3'){
        valid = 0 ;
        sound(500) ; delay(100) ; nosound() ;
        gotoxy(28,12) ; cout <<" " ;
    }
} while (!valid) ;
played++ ;
if (t_sol != sol){
    sound(500) ; delay(100) ; nosound() ;
    gotoxy(14,16) ; cout <<"wrong answer" ;
    gotoxy(13,17) ; cout <<"Correct answer : " <<sol ;
    gotoxy(1,1) ; ch = getch() ;
    gotoxy(14,15) ; cout <<" " ;
}

```

```

        gotoxy(13,16) ; cout <<"          " ;
        gotoxy(3,24) ; cout <<"          " ;
    }
    else{
        correct++ ;
        sound(500) ; delay(100) ; nosound() ;
        gotoxy(16,16) ; cout <<"GOOD ! CORRECT ANSWER" ;
        gotoxy(25,3) ; cout <<"SCORE:" <<correct*10 ;
        gotoxy(1,1) ; ch = getch() ;
    }
    if (ch == 27){
        display_score(played,correct) ;
        return ;
    }
}
gotoxy(3,22) ; cout <<"No more questions available" ;
getch() ;
if (!played)
    return ;
display_score(played,correct) ;
}

```

Add Question Module:

```

void Quiz :: add(char filename[13]){
    int valid ; int t_ques_no ;
    t_ques_no = record_count(filename) ; t_ques_no++ ;
    if (t_ques_no == 1){
        ques_no = t_ques_no ; strcpy(ques,"abc") ;
        strcpy(ans1,"-") ; strcpy(ans2,"-") ; strcpy(ans3,"-") ;
        sol = '0' ;
        fstream file ;
        file.open(filename, ios::out | ios::app ) ;
        file.write((char *) this, sizeof(Quiz)) ;
        file.close() ;
        delete_record(filename,t_ques_no) ;
    }
    char ch ;
    do{
        clrscr() ;
        gotoxy(15,4); cout<<"A D D   Q U E S T I O N S";
        gotoxy(5,8);  cout<<"Q " <<t_ques_no;
        gotoxy(5,10); cout<<"1.";
        gotoxy(5,12); cout<<"2.";
        gotoxy(5,14); cout<<"3.";
        gotoxy(5,16); cout<<"Solution : ";
        do{
            valid = 1 ;
            gotoxy(1,25) ; clreol() ;
            gotoxy(72,25) ; cout <<"<0>=EXIT" ;
            gotoxy(3,25) ; cout <<"ENTER QUESTION" ;
            gotoxy(12,8) ; gets(ques) ; strupr(ques) ;

```



```

        if (ques[0] == '0')
            return ;
    } while ( !valid ) ;
    for(i=1;i<=3;i++){
        do{
            valid = 1 ;
            gotoxy(1,25) ; clrhol() ;
            gotoxy(72,25) ; cout <<"<0>=EXIT" ;
            gotoxy(3,25) ; cout <<"ENTER ANSWER FOR THE QUESTION" ;
            gotoxy(12,10) ; gets(ans[i]) ; strupr(ans[i]) ;
            if (ans1[0] == '0')
                return ;
        } while ( !valid ) ;
    }
    do{
        valid = 1 ;
        gotoxy(1,25) ; clrhol() ;
        gotoxy(72,25) ; cout <<"<0>=EXIT" ;
        gotoxy(3,25) ; cout <<"ENTER CORRECT SOLUTION FOR THE QUESTION
(1/2/3)" ;
        gotoxy(16,16) ; sol = getche() ;
        if (sol == '0')
            return ;
        if (sol != '1' && sol != '2' && sol != '3'){
            valid = 0 ;
            sound(500) ; delay(100) ; nosound() ;
            gotoxy(14,16) ; clrhol() ;
        }
    } while ( !valid ) ;
    gotoxy(1,25) ; clrhol() ;
    gotoxy(1,24) ; clrhol() ;
    gotoxy(72,25) ; cout <<"<0>=EXIT" ;
    gotoxy(3,19) ; cout <<"Do you want to save the record (y/n) : " ;
    do{
        valid = 1 ;
        gotoxy(42,19) ; ch = getche() ; ch = toupper(ch) ;
        if (ch == '0')
            return ;
        if (ch != 'Y' && ch != 'N'){
            valid = 0 ;
            sound(500) ; delay(100) ; nosound() ;
            gotoxy(42,19) ; clrhol() ;
        }
    } while ( !valid ) ;
    if (ch == 'Y'){
        ques_no = t_ques_no ;
        fstream file ;
        file.open(filename, ios::app ) ;
        file.write((char *) this, sizeof(Quiz)) ;
        file.close() ;
        t_ques_no++ ;
    }
}

```

```
gotoxy(72,25) ; cout <<"<0>=EXIT" ;
gotoxy(3,21) ; cout <<"Do you want to add more questions (y/n) : " ;
do{
    valid = 1 ;
    gotoxy(45,21) ; ch = getche() ; ch = toupper(ch) ;
    if (ch == '0')
        return ;
    if (ch != 'Y' && ch != 'N'){
        valid = 0 ;
        sound(500) ; delay(100) ; nosound() ;
        gotoxy(45,21) ; clrhol() ;
    }
} while ( !valid ) ;
} while (ch == 'y');
}
```