**ONLINE QUIZ SYSTEM**

By

**KAVIN SAGAR S (1518106041)**

**KUMARAN R (1518106045)**

**VIJAY R (1518106106)**

of

**SONA COLLEGE OF TECHNOLOGY**

Salem – 636 005

**A PROJECT REPORT**

**Submitted to the**

U15IT604R SOFTWARE DESIGN AND TESTING LABORATORY

*In partial fulfillment of the requirements*

*for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**ANNA UNIVERSITY**

**CHENNAI**

**APRIL 2021**

**BONAFIDE CERTIFICATE**

Certified that this project report title **ONLINE QUIZ SYSTEM** is the bonafide work of **KAVIN SAGAR S, KUMARAN R, VIJAY R** who carried out the project work under my supervision.

**Mr.P.IYANNAR., M.E., Ph.D.,**  Dr.J.AKILANDESWARI**,**

Assistant Professor (Senior Grade), Professor and Head

Department of Information Technology, Department of Information Technology

Sona College of Technology, Sona College of Technology,

Salem-636005. Salem-636005.

*Submitted for End Semester Practical Examination held on ……………….*

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**Table of CONTENTS**

**Chapter no title page no**

**lIST OF FIGURES** v

**List of TABLES** vii

**1 Problem definition 1**

**2 Problem scope 2**

**3 Software requirements specification 3**

**4 System analysis 16**

4.1 Attribute matrix 16

4.2 Traceability matrix 18

4.3 Traceability tree 20

**5 System design 23**

5.1 Use case diagram 23

5.2 Activity diagram 24

5.3 Class diagram 31

5.4 Sequence diagram 35

5.5 Collaboration diagram 40

5.6 Component diagram 44

5.7 Deployment diagram 50

**6 System development 51**

6.1 LOGIN 51

6.2 CREATE QUIZ 51

6.3 CHOOSE SUBJECT 51

6.4 TAKE QUIZ 51

6.5 SCORE BOARD 51

6.6 LOGOUT 51

**7 System implementation 52**

7.1 System pre-requisites 52

7.2 Steps to install software 52

**8 System testing 53**

8.1 Test plan 53

8.2 Test cases 55

8.2.1 Unit Testing 55

8.2.2 Functional Testing 56

8.3 Test reports 57

8.3.1 Unit Testing 57

8.2.2 Functional Testing 58

**9 Conclusion and future enhancement 63**

**APPENDIX 1 Sample code 64**

**APPENDIX 2 Sample screen shots 77**

**LIST OF FIGURES**

**FIGURE NO TITLE PAGE NO**

3.1 Overall Description 4

4.1 Stack Holder Requirement Attribute Matrix 13

4.2 Use Case Requirement Attribute Matrix 14

4.3 Stack Holder Vs Stack Holder Requirement 15

4.4 Use Case Vs Use Case Requirement 16

4.5 Use Case Vs Stack Holder Requirement 16

4.6 Stack Holder Requirement Traceability Tree 17

4.7 Use Case Requirement Traceability Tree 18

5.1 Quiz Use Case Diagram 19

5.2 Login Activity Diagram 20

5.3 Create Quiz Activity Diagram 21

5.4 Choose Subject Activity Diagram 22

5.5 Take Quiz Activity Diagram 23

5.6 Score Board Activity Diagram 24

5.7 Logout Activity Diagram 25

5.8 Login Class Diagram 26

5.9 Create Quiz Class Diagram 27

5.10 Choose Subject Class Diagram 28

5.11 Take Quiz Class Diagram 29

5.12 Score Board Class Diagram 30

5.13 Logout Class Diagram 31

5.14 Login Sequence Diagram 32

5.15 Create Quiz Sequence Diagram 33

5.16 Choose Subject Sequence Diagram 34

5.17 Take Quiz Sequence Diagram 35

5.18 Score Board Sequence Diagram 36

5.19 Logout Sequence Diagram 37

5.20 Login Collaboration Diagram 38

5.21 Create Quiz Collaboration Diagram 39

5.22 Choose Subject Collaboration Diagram 40

5.23 Take Quiz Collaboration Diagram 41

5.24 Score Board Collaboration Diagram 42

5.25 Logout Collaboration Diagram 42

5.26 Quiz System Component Diagram 43

5.27 Login Component Diagram 44

5.28 Create Quiz Component Diagram 45

5.29 Choose Subject Component Diagram 46

5.30 Take Quiz Component Diagram 47

5.31 Score Board Component Diagram 48

5.32 Logout Component Diagram 49

5.33 Quiz System Deployment Diagram 50

8.1 Rational Administrator 58

8.2 Rational Robo-Recorder 59

8.3 Rational Robo-GUI Shell Script 60

8.4 Test Log Run 61

8.5 Test Log- Result 62

A 1.1 Student Login 77

A 1.2 Admin Login 78

A 1.3 Admin Home Page 79

A 1.4 Admin user page 80

A 1.5 Add Quiz page 81

A 1.6 Remove Quiz Page82

A 1.7 Student Home Page 82

A 1.8 Student History Page 83

A 1.9 Student Ranking Page 83

**CHAPTER-1**

**PROBLEM STATEMENT**

**QUIZ SYSTEM** is a software developed to conduct an Online Quiz. Quiz System is accessed by entering the username and e-mail id which is added to the database. Before start of the Quiz, the rules and regulations are displayed that includes description of number of questions to be answered and scoring methods. Quiz is started by displaying ten questions with four options each based on category choose by Student on the Subject posted by Admin(Staff). If the answer is correct, scores incremented by two and no negative marks for wrong answers. Final score will be displayed and updated in the database with username

**CHAPTER-2**

**PROJECT SCOPE**.

The Main Scope of **Online Quiz** is User-friendly systems are not only needed for the creator, but also for participants. A responsive design is an approach where the web designer wants to reach an optimal web experience for a wide range of devices. Multiple choice is the options you can use with our online Quiz system. Having more than one option is necessary to check several types of knowledge. Auto quizzes publish, log in with username and password. Automatically check answers and give the result to the students.

**CHAPTER-3**

**SOFTWARE REQUIREMENT SPECIFICATION**

# Introduction

## Purpose

This web application provides facility to conduct online examination in a college. It saves time as it allows large number of students to attend the exam at the same time and display the result just after the exam is finished. The teacher in this system has the privilege to create, modify and delete the test papers and its questions.

## Document Conventions

The following are the list of conventions and acronyms used in this document:

**Administrator** – A loin id representing a user with user administration privileges to the software.

**User** – A general login id assigned to the user.

## Intended Audience and Reading Suggestions

This document has Introduction, Overall Description, External Interface Requirement, System Feature, Functional and non-functional requirements and other requirements. Developers can have an overall outline view of Online Quiz System in this document. It helps the students to learn more about the online quiz system development.

## Product Scope

This can be used in a lot of educational institutions. It can be used by many students to attend their exams any time as it is a web-based application. It also improves the standard of education through online.

## References

* IEEE Std 830-1998(Revision of IEEE Std 830-1993)
* IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.
* IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
* R. Pressman 5E, reference book

# Overall Description

TOPICS

QUIZ/TEST

HOME

STAFF

LOGIN

LOGOUT

**3.1 Overall Description**

## Product Perspective

The proposed Language Skill Exam System is an on-line Exam System. The online test created for taking online test has following stages.

* + - Login
    - Test
    - Result

## Login:

There is a quality login window because this is more secure than other login forms as in a normal login window there are multiple logins available so that more than one person can access to test with their individual login. But in this project, there is only one login id i.e., administrator id and password by which a person enter the site. Hence it is more secure and reliable than previously used on-line test simulators.

## Test:

Test page is the most creative and important page in this project. It consists of 2 modules namely.

* + Subject selection
  + Utilities

## Subject Selection:

From the given choices the candidate can select his field (like C, C++ and JAVA etc.) for taking on with the test.

## Utilities:

It includes:

Skip and come back to the question afterwards if needed.

Gives the list of attempted and unattempt questions and can go to any question directly and can either attempt or change the answer of the already attempted question.

## Product Functions

The product includes many functionalities of the organization. These functionalities are:

* New student registration by admin.
* User database management.
* Score evaluation.
* To maintain records of student scores.
* To conduct online quizzes.

## User Classes and Characteristics

Many users such as the Administrator, Teacher and Students, can use system. Only administrator of the system should be able to enter user information or modify any kind of information in the system, but every user, whether a teacher or a student, with a valid login ID and password, should be able to view their details such as their personal information, marks in previous tests etc.

## Operating Environment

The product will be operating in windows environment. Also, it will be compatible with the IE 6.0. Most of the features will be compatible with the Google Chrome,

Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

## Design and Implementation Constraints

The implementation of this product is done with two modules.

1. Module I:

This first module tracks the path of Administrator. In this module it shows how the admin can log into the system with a valid password and can add new valid user to the database. His name and exam are saved in the database. Also, he can manipulate the database and add new question in the database from the webpage itself. It also shows how many questions are in database.

1. Module II:

This second module tracks the path for the examinee. The examinee can log into the system with a valid ID issued by the admin. After successful login into the system the examinee moves to the instruction web page where he will get instruction about the examination process. Then after clicking the start button the exam starts and timer also starts .In this manner examinee can take up the test and clicking the submit button , he will get the result of that section immediately. At the end, system displays the initial web page.

* + - Front end is fully developed using HTML and CSS in Visual studio.
    - Xampp server is used to carry out the quiz. It stores the questions and answers in its database. PHP is used for accessing the database.

## User Documentation

The product will include user manual. The user manual will include the information about the product overview, complete configuration of the used software, technical details and contact information which will include email address.

## Assumptions and Dependencies

We assume all users have basic computer knowledge and our quiz system provides good user interface and help the user at any moment during their visit to the website

# External Interface Requirements

## User Interfaces

It incorporates with effective GUI concepts and focuses on user-friendly systems. It has good, appealing, attractive, and aesthetic web pages with optimum hyperlink to select the required process.

## Hardware Interfaces

**Server Side:**

* + - Operating System: Windows 9x/xp ,Windows ME
    - Processor: Pentium 3.0 GHz or higher
    - RAM: 256 Mb or more
    - Hard Drive: 10 GB or more

## Client side:

* + - Operating System: Windows 9x or above, MAC or UNIX.
    - Processor: Pentium III or 2.0 GHz or higher.
    - RAM: 256 Mb or more

## Software Interfaces

Interfacing with several modules to perform various operations, it requires Database connectivity which is provided by MYSQL and Server interfacing which is provided by APACHE.

## Communications Interfaces

The Customer must connect to the Internet to access the Website:

* + - Dialup Modem of 52 kbps
    - Broadband Internet
    - Dialup or Broadband Connection with an Internet Provider

# 

# 4. System Features

## System Features 1:

**4.1.1 Description and Priority**

Proposed Database is intended to store, retrieve, update, and manipulate information related to university which include.

* + - Profile of both users
    - Student details
    - My account
    - Test results

**4.1.2 Stimulus/Response Sequences**

**Responses for Administrator:**

The administrator can Login and Logout. When the Administrator Logs into the Online Exam system. The system will check for validity of login .If the Login and password are valid, the response to this action is the administrator will be able to modify, view, add, deleting and all other functions that can be performed on the database.

**Examination:**

First, the user/examinee gets a valid identification number (same as the roll for a normal examination).The user can log on with this identification no. and can take up the examination. After logging in the user can see various options and can choose the option from the menu

**4.2 System Features 2**

This section gives the list of Functional and nonfunctional requirements which are applicable to the Online Quiz System. Functional requirements are nothing, but the services provided by the system to its end users. There are three sub modules in this phase.

* Candidate module.
* Examiner module.
* Administrator module.

# 

# 5.Other Nonfunctional Requirements

## Performance Requirements

Requirements include our total no of tasks such as login, new user registration will take certain time periods for both normal and peak workload conditions. Here we measure each task would be processed in less than 1 sec. It means 100% task performed in less than 1 sec.

## Safety Requirements

The system has been designed in a way to protect the users from malwares. It also allows only the valid uses to access the system.

## Security Requirements

* The login page asks for use rid and password.
* The use rid and password has certain conditions for security needs.

## Software Quality Attributes

* **Reliability:**

This system is highly reliable so that it gets updated every time when it is updated.

## Availability:

This system is available around the time so that the users can take up the test any time and can also attend a scheduled test.

## Maintainability:

The maintenance instructions will be given in the user manual so that the user can use the website properly.

## Business Rules

This product has certain business rules like a user can login with one account in only one window.

**CHAPTER-4**

**SYSTEM ANALYSIS**

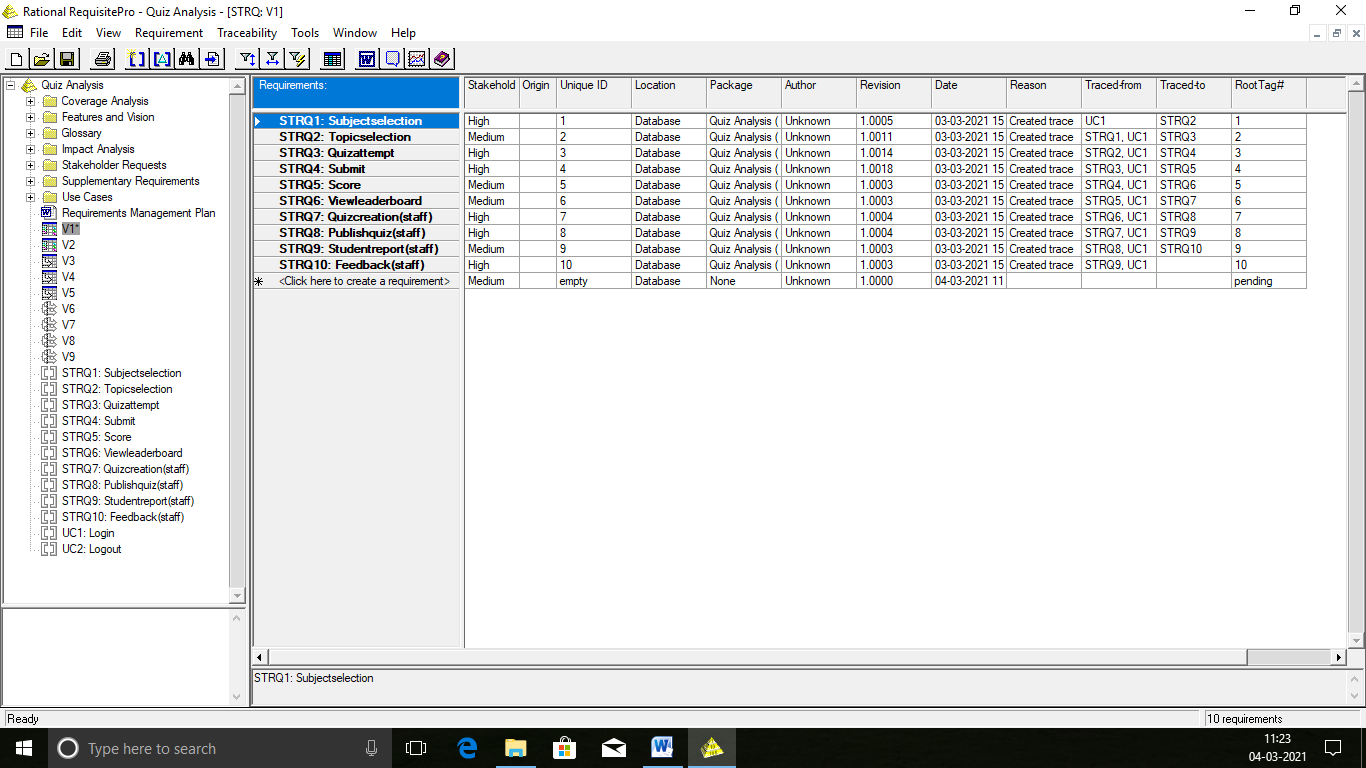
The next workflow in the RUP is the Analysis of the requirements which have been specified in the SRS. The Analysis is done with the help of Rational Requisite Pro. The three views or reports which form the basis for analysis are

Attribute Matrix, Traceability Matrix, Traceability Tree

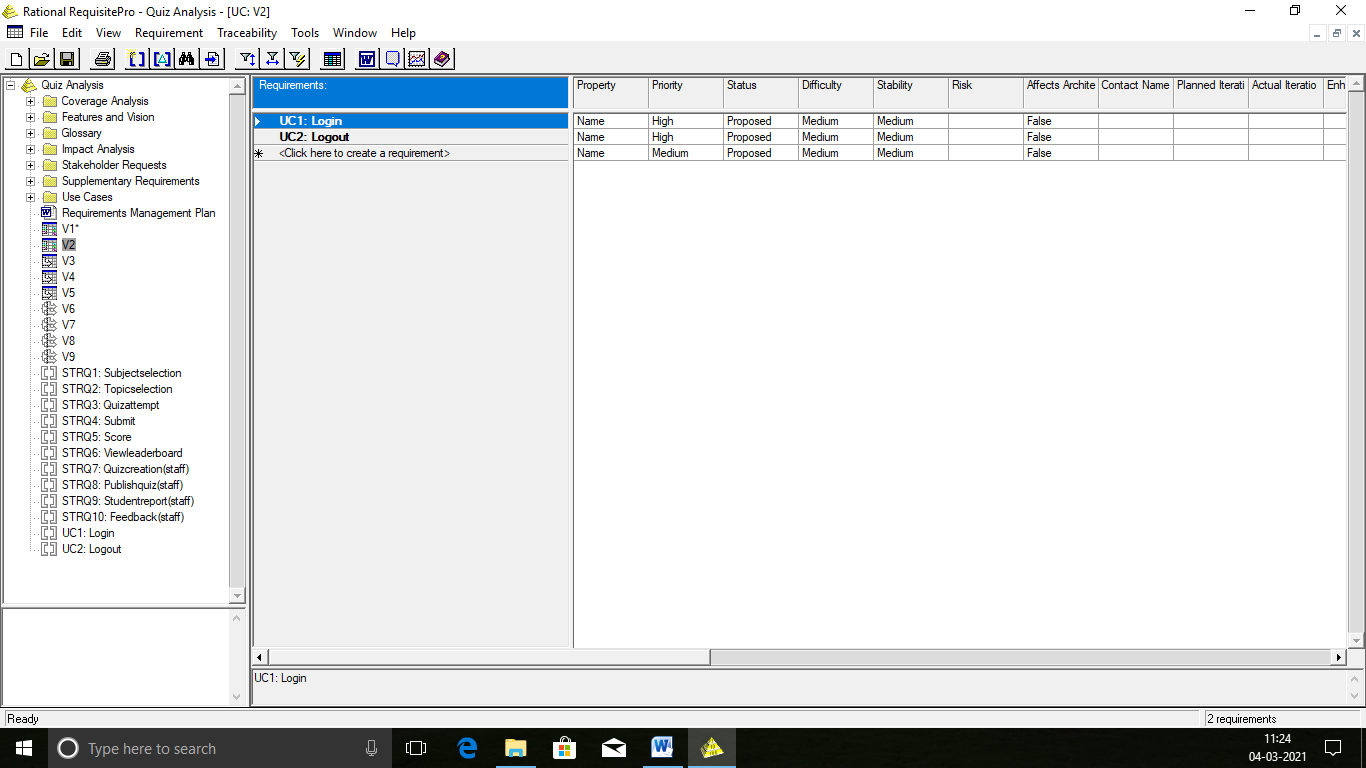
**4.1 ATTRIBUTE MATRIX**

The Attribute Matrix view is a spreadsheet like display that lists the requirements of a specific requirements type and their attributes. Requirements are arranged in rows, listed by tag number and followed by requirement name. Attributes are arranged in columns.

**4.1.1 STACKHOLDER REQUIREMENT SPECIFICATION**



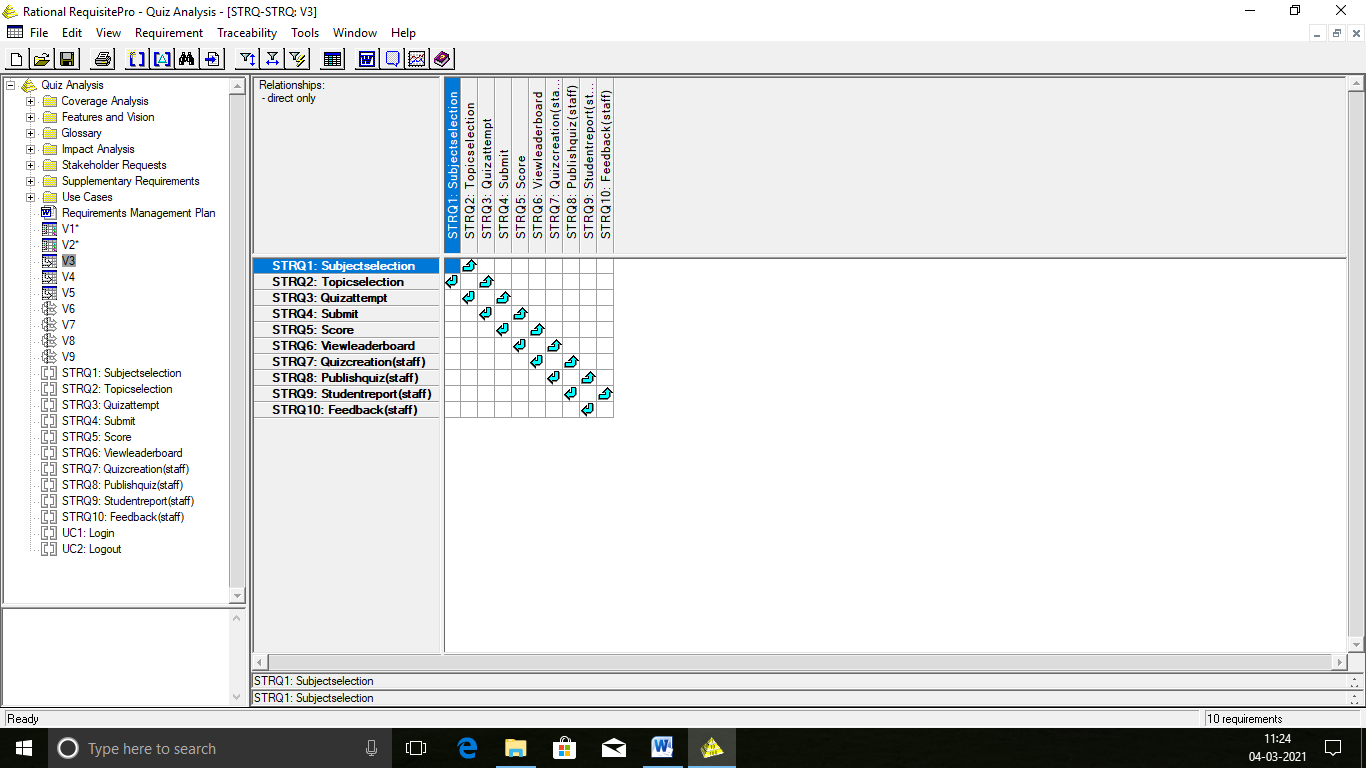
**4.1.2 USECASE REQUIREMENT SPECIFICATION**



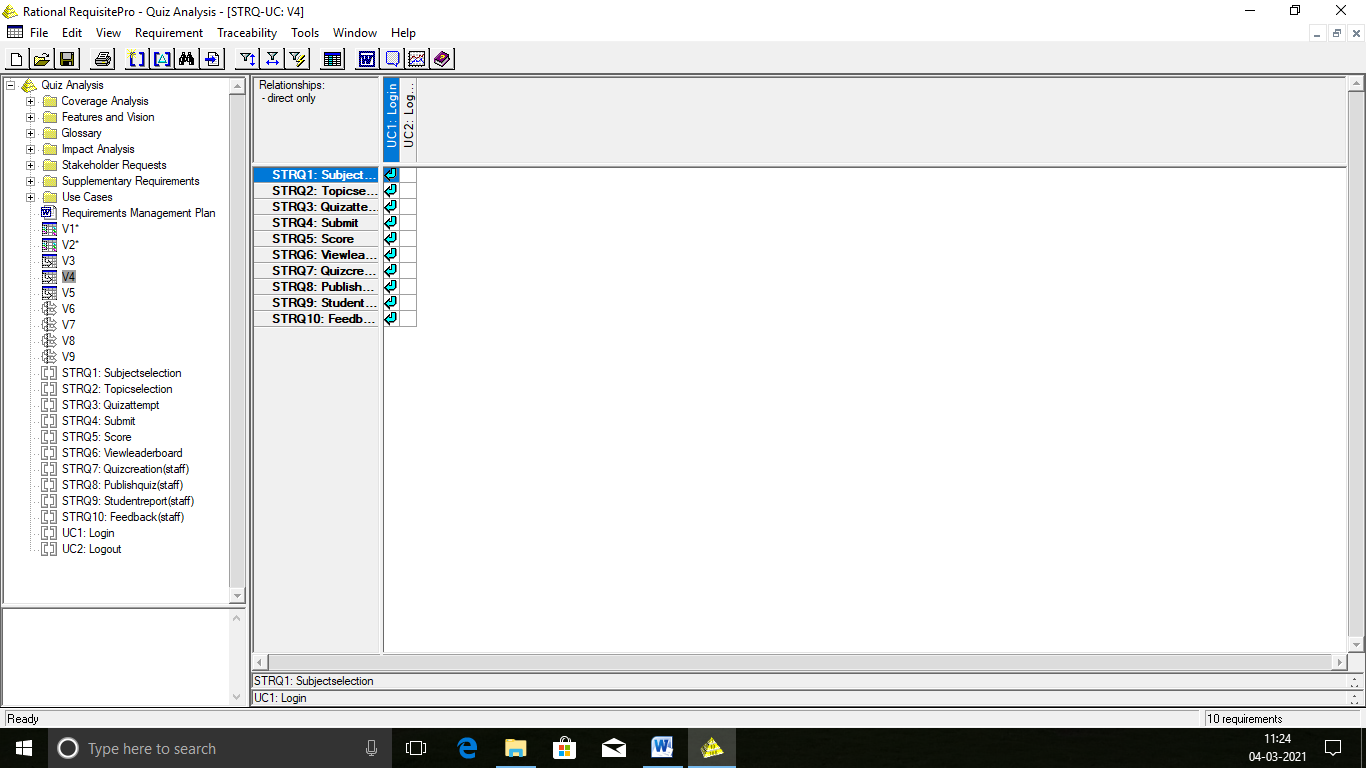
**4.2 TRACEABILITY MATRIX**

Traceability Matrix is a view that illustrated the relationships between requirements of the same or different types. We can use this matrix to create, modify and delete traceability relationships and view indirect relationships and view direct relationships and traceability relationships with a suspect state. We can also use the traceability matrix to filter and sort the requirements and columns requirements separately.

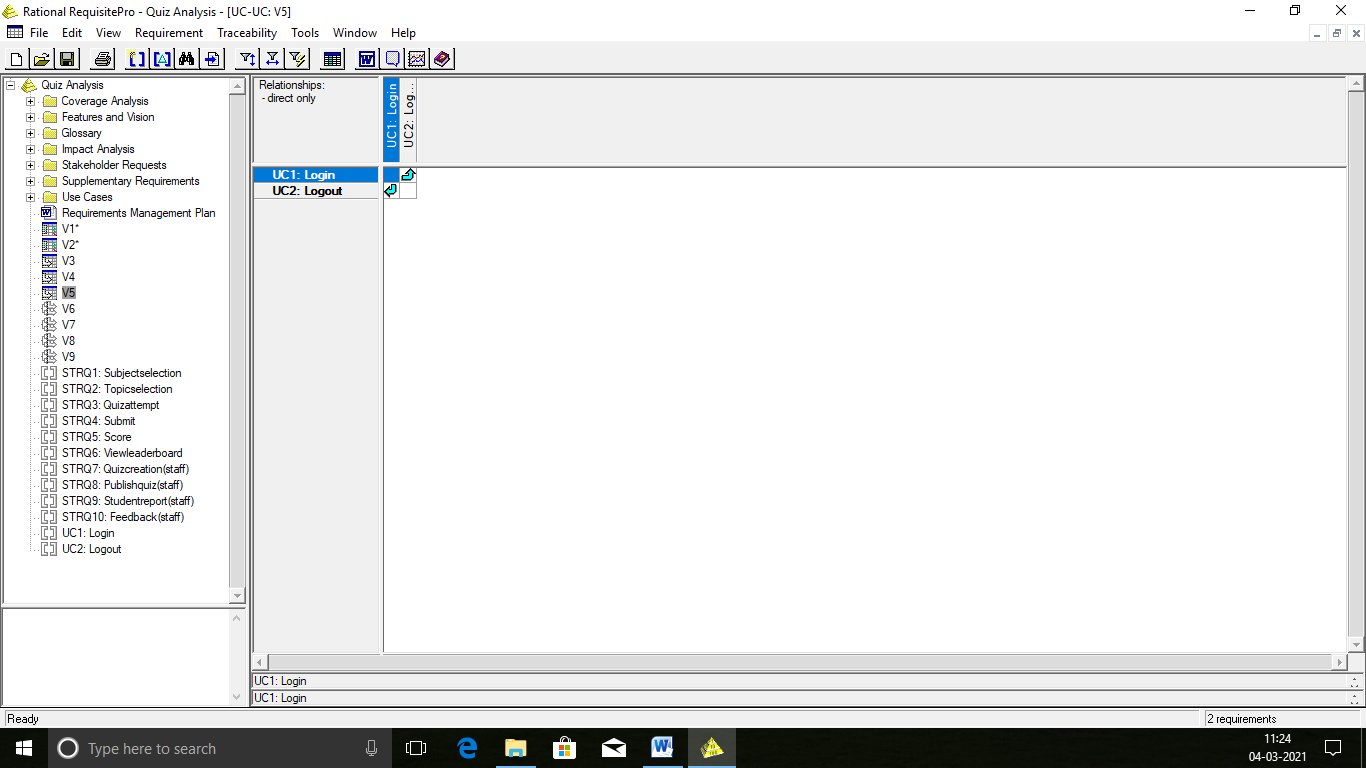
**4.2.1 STAKEHOLDER VS STAKEHOLDER REQUIREMENT TYPE**



**4.2.2 STAKEHOLDER VS USECASE REQUIREMENT TYPE**



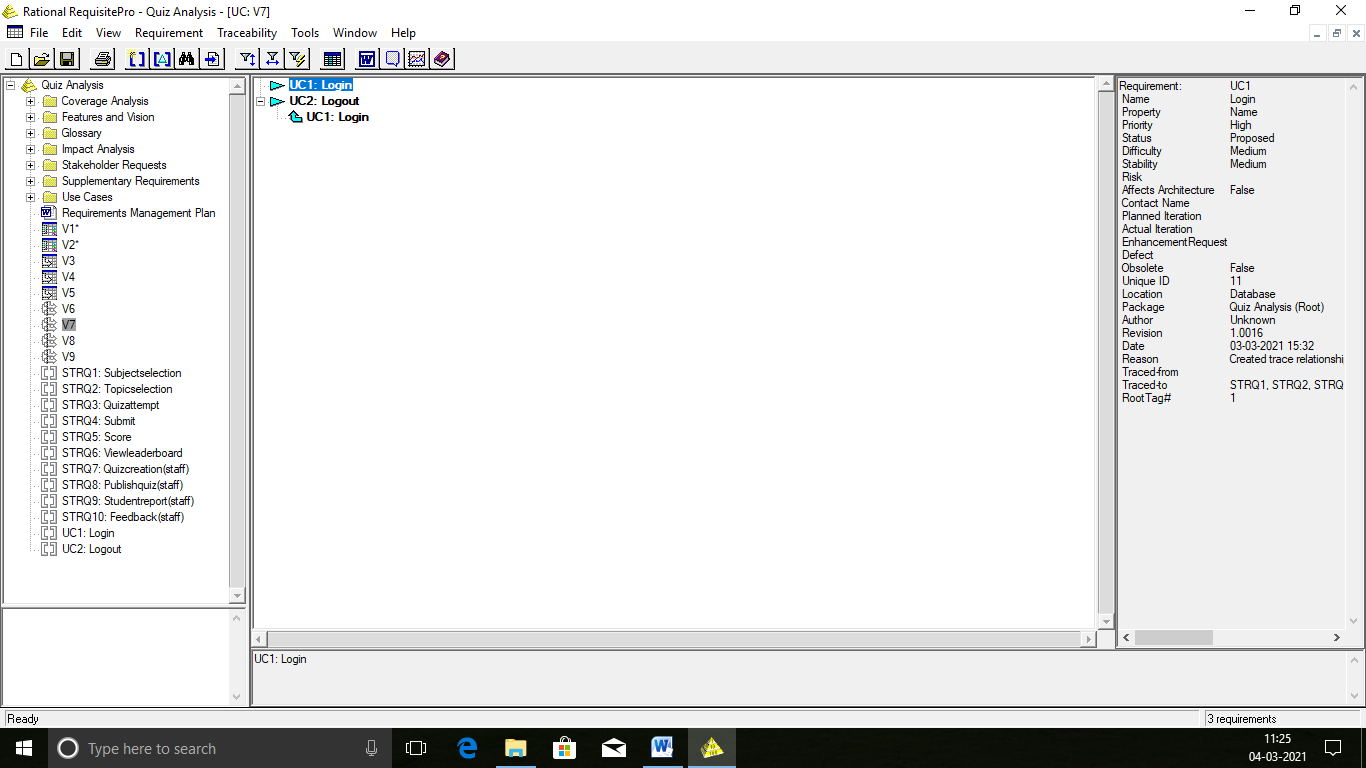
**4.2.3 USECASE VS USECASE REQUIREMENT TYPE**



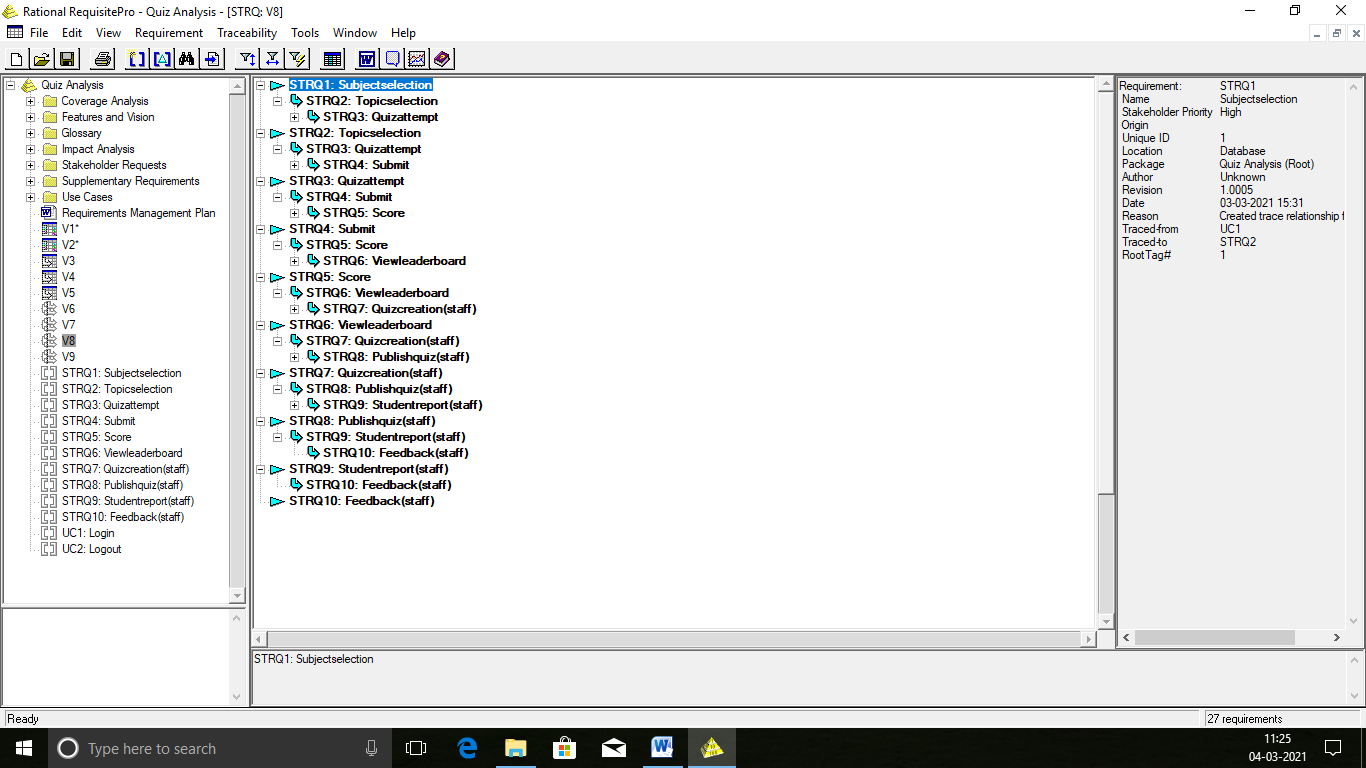
**4.3 TRACEABILITY TREE**

A view that displays all internal and external requirements traced to or from a requirement. The traceability tree only displays the first level project traceability.

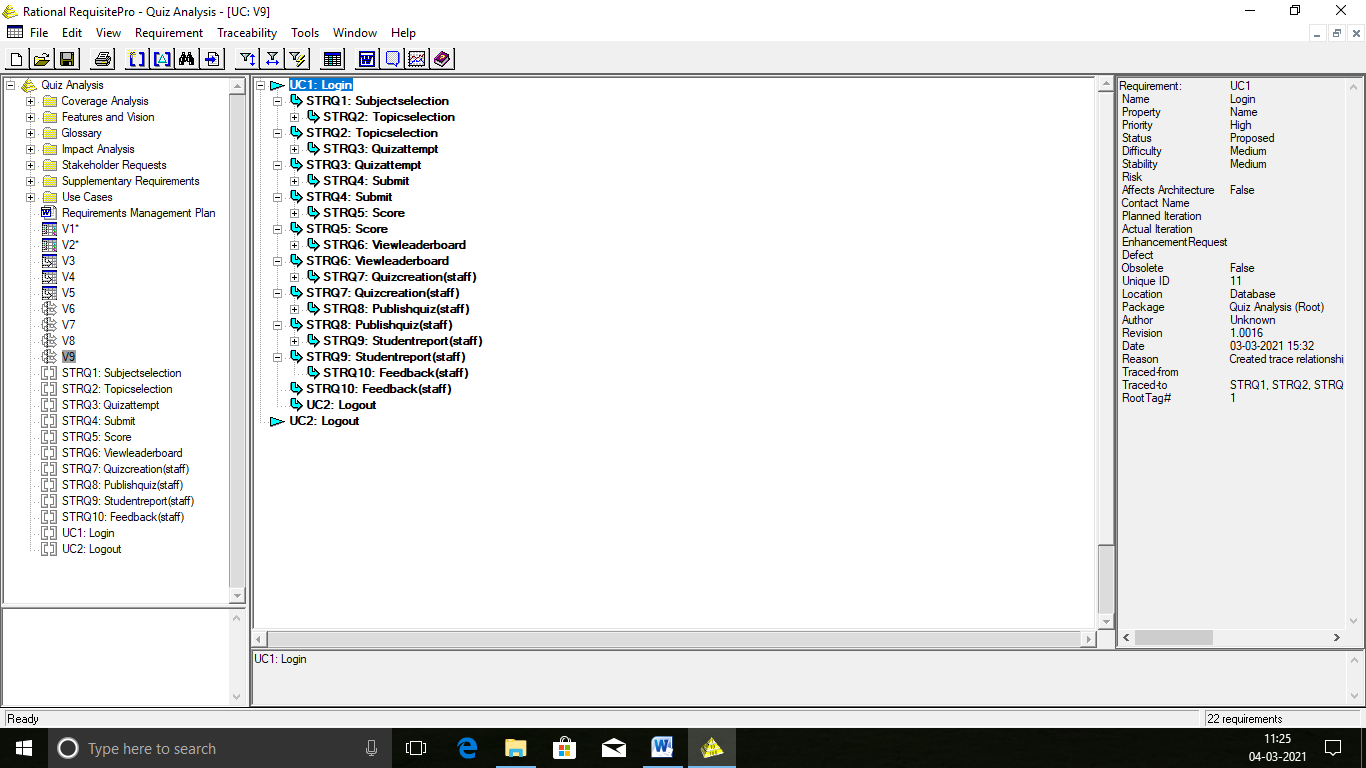
**4.3.1 TRACEABILITYTREE(TRACEDINTO)-USECASE REQUIREMENT TYPE**



**4.3.2 TRACEABILITYTREE(TRACEDINTO)-STAKEHOLDER REQUIREMENT TYPE**



**4.3.3 TRACEABILITYTREE(TRACEDOUTOFF)-STAKEHOLDER REQUIREMENT TYPE**



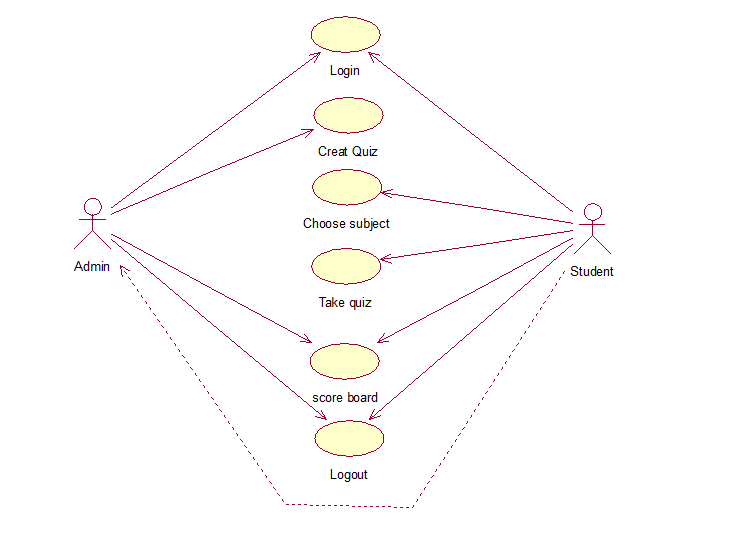
**CHAPTER-5**

**SYSTEM DESIGN**

**5.1 USECASE DIAGRAM**

A Use Case Diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication association between the actors and the use cases and generalization among the use cases. A use case corresponds to a sequence of transactions, in which each transaction is invoked from outside the system and engages internal objects to interact with each other. An actor is anything that interacts with the use case.

**5.1.1 USECASE DIAGRAM FOR QUIZ SYSTEM**

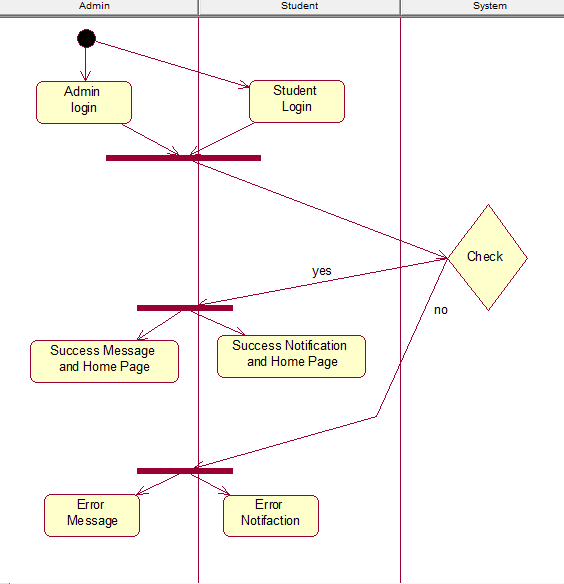


**Figure 5.1 Use Case Diagram**

**5.2 ACTIVITY DIAGRAM**

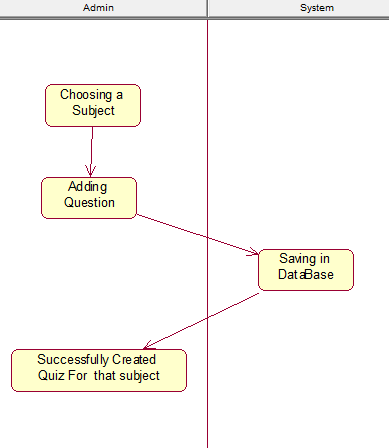
Activity Diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

**5.2.1 LOGIN**



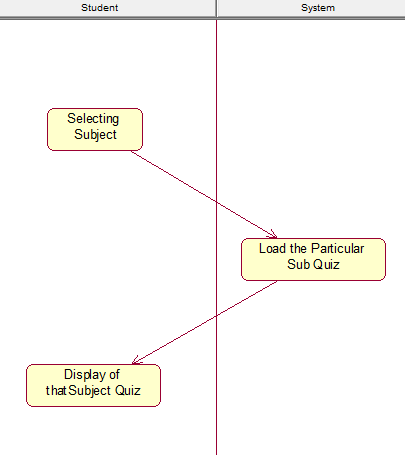
**Figure 5.2 Login Activity Diagram**

**5.2.2 CREATE QUIZ**



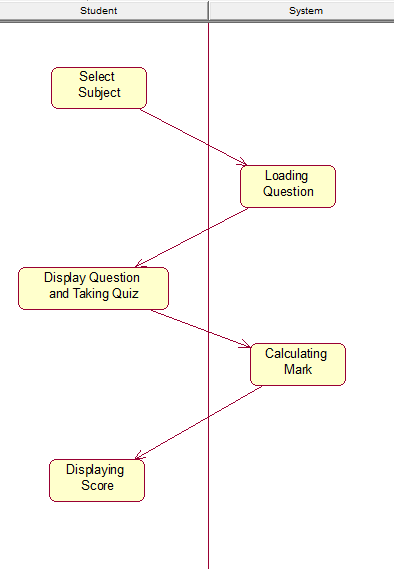
**Figure 5.3 Create Quiz Activity Diagram**

**5.2.3 CHOOSE SUBJECT**



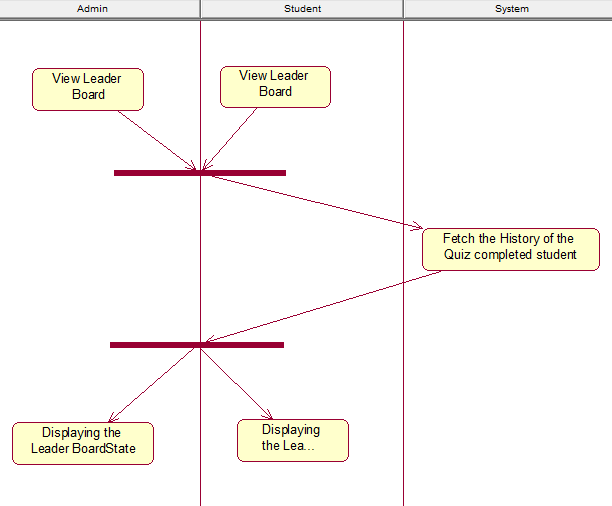
**Figure 5.4 Choose Subject Activity Diagram**

**5.2.4 TAKE QUIZ**



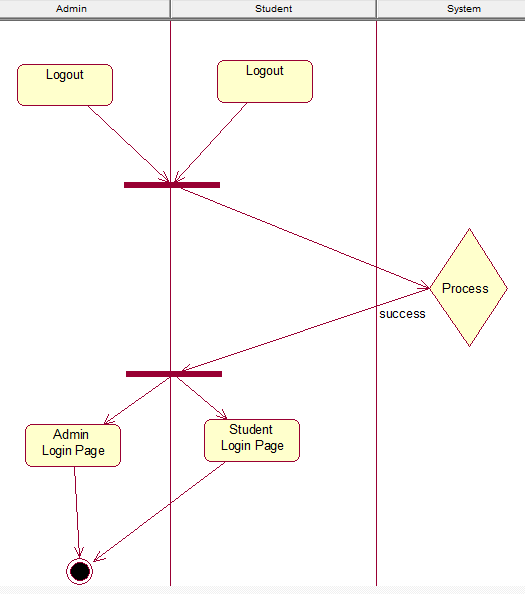
**Figure 5.5 Take Quiz Activity Diagram**

**5.2.5 SCORE BOARD**



**Figure 5.6 Score Board Activity Diagram**

**5.2.6 LOGOUT**

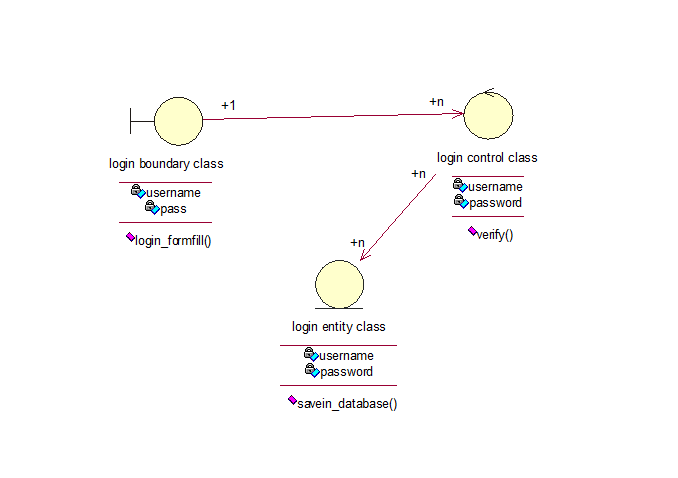


**Figure 5.7 Logout Activity Diagram**

**5.3 CLASS DIAGRAM**

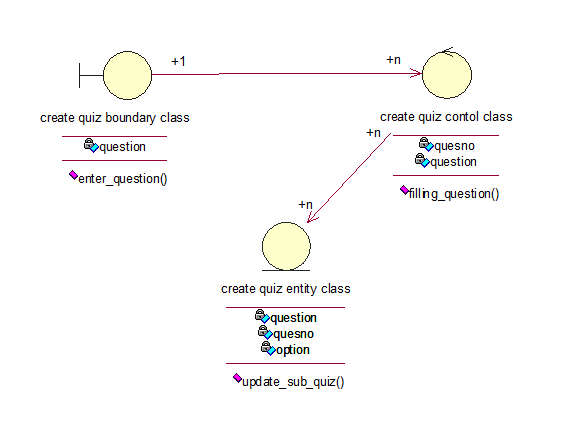
A Class Diagram is a collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents. These diagrams show the static structures of the model.

**5.3.1 LOGIN**



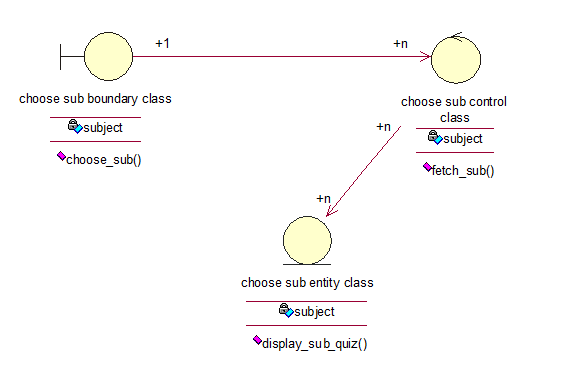
**Figure 5.8 Login Class Diagram**

**5.3.2 cREATE QUIZ**



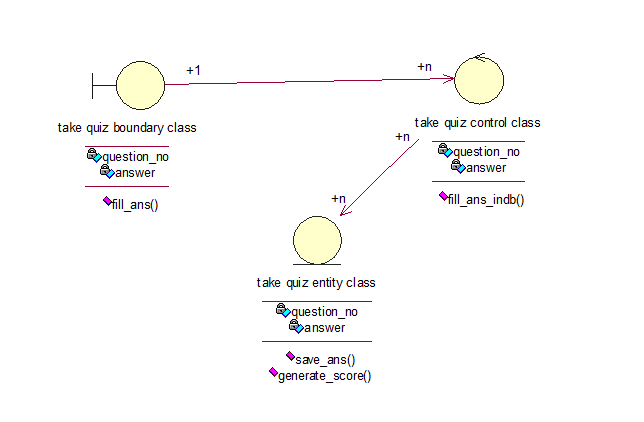
**Figure 5.9 Create Quiz Class Diagram**

**5.3.3 CHOOSE SUBJECT**



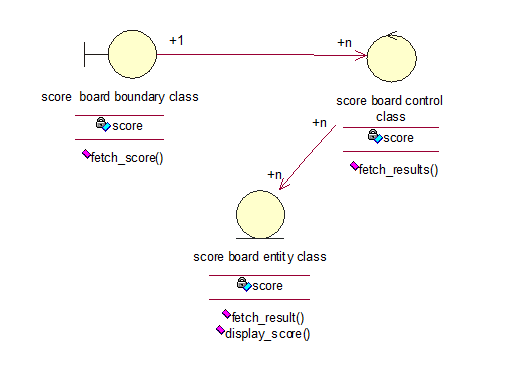
**Figure 5.10 Choose Quiz Class Diagram**

**5.3.4 TAKE QUIZ**



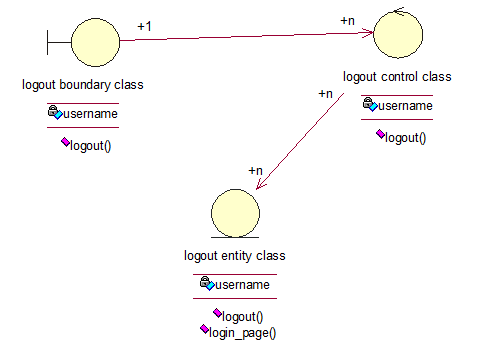
**Figure 5.11 Take Quiz Class Diagram**

**5.3.5 SCORE BOARD**



**Figure 5.12 Score Board Class Diagram**

**5.3.6 LOGOUT**



**Figure 5.13 Logout Class Diagram**

**5.4 SEQUENCE DIAGRAM**

Sequence Diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows the interaction arranged in a time sequence. It shows the object participating by their lifelines and the messages they exchange, arranged in a time sequence.

**5.4.1 LOGIN**



**Figure 5.14 Login Sequence Diagram**

**5.4.2 CREATE QUIZ**



**Figure 5.15 Create Quiz Sequence Diagram**

**5.4.3 CHOOSE SUBJECT**



**Figure 5.16 Choose Subject Sequence Diagram**

**5.4.4 TAKE QUIZ**



**Figure 5.17 Take Quiz Sequence Diagram**

**5.4.5 SCORE BOARD**



**Figure5.18 Score Board Sequence Diagram**

**5.4.6 LOGOUT**



**Figure5.19 Logout Sequence Diagram**

**5.5 COLLABORATION DIAGRAM**

A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desire outcome.

**5.5.1 LOGIN**



**Figure 5.20 Login Collaboration Diagram**

**5.5.2 CREATE QUIZ**



**Figure 5.21 Create Quiz Collaboration Diagram**

**5.5.3 CHOOSE SUBJECT**



**Figure 5.22 Choose Subject Collaboration Diagram**

**5.5.4 TAKE QUIZ**



**Figure 5.23 Take Quiz Collaboration Diagram**

**5.5.5 SCORE BOARD**



**Figure 5.24 Score Board Collaboration Diagram**

**5.5.6 LOGOUT**

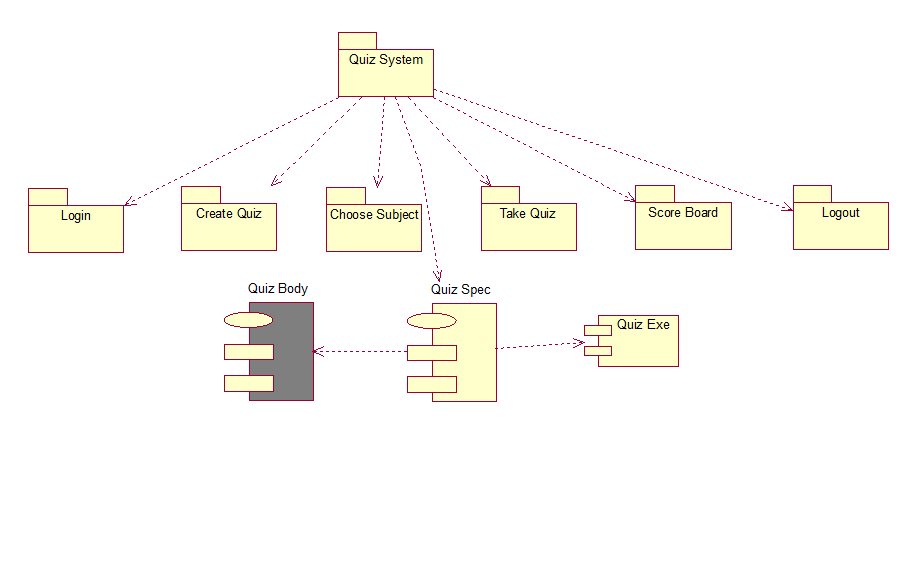


**Figure 5.25 Logout Collaboration Diagram**

**5.6 COMPONENT DIAGRAM**

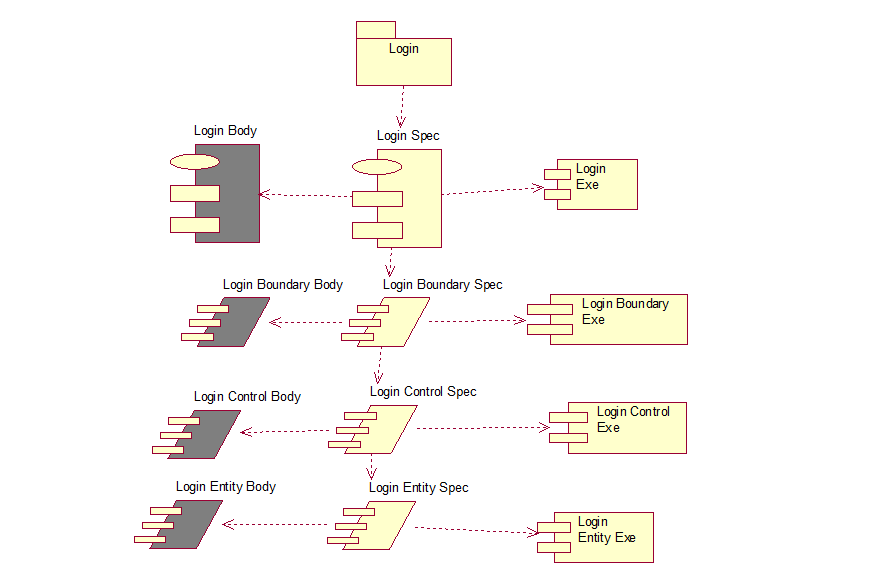
Component diagram models the physical components such as source codes, executable programs, user interface in a design. These high-level physical components may or may not be equivalent to many smaller components we use in creation of application.

**5.6 MAIN COMPONENT DIAGRAM**



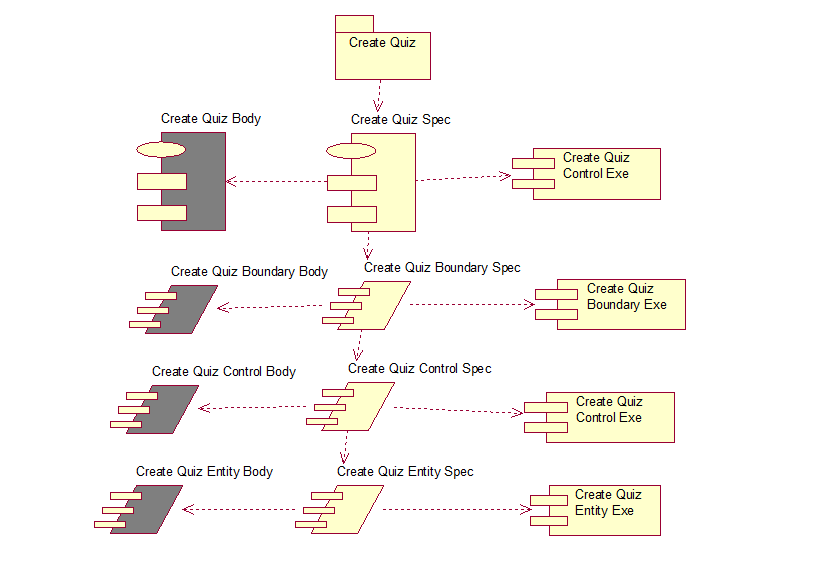
**Figure 5.26 Main Component Diagram**

**5.6.1 LOGIN**



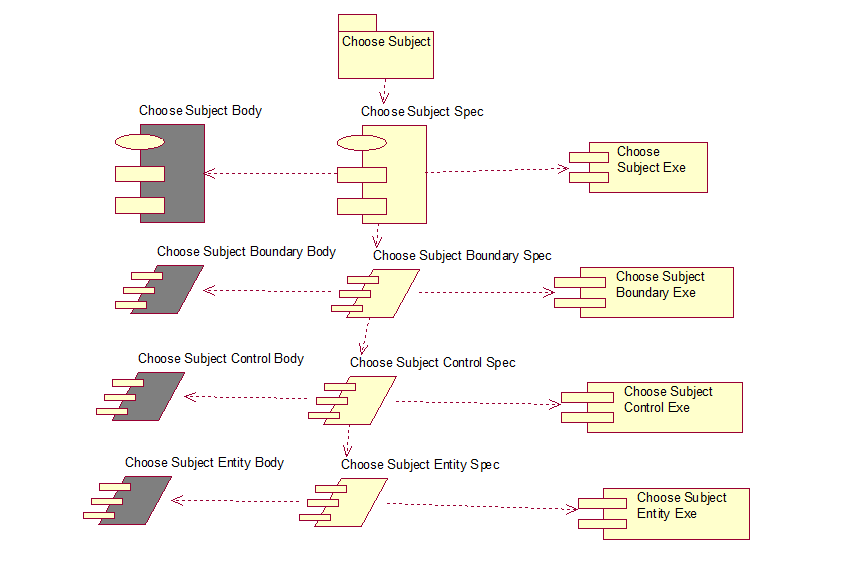
**Figure 5.27 Login Component Diagram**

**5.6.2 CREATE QUIZ**



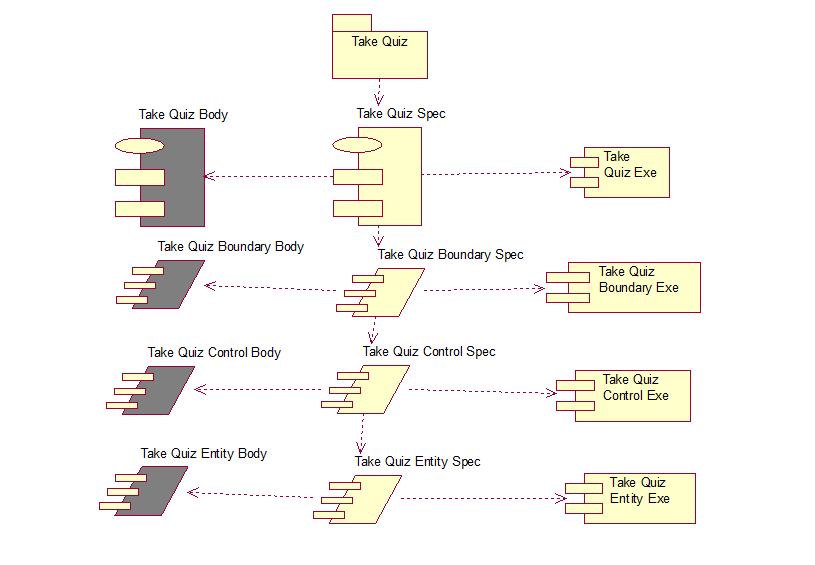
**Figure 5.28 Create Quiz Component Diagram**

**5.6.3 CHOOSE SUBJECT**



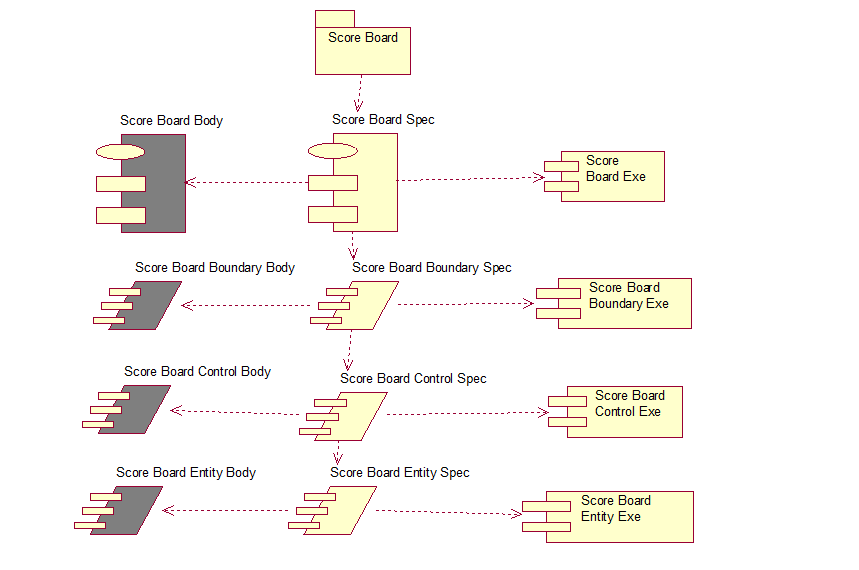
**Figure 5.29 Choose Subject Component Diagram**

**5.6.4 TAKE QUIZ**



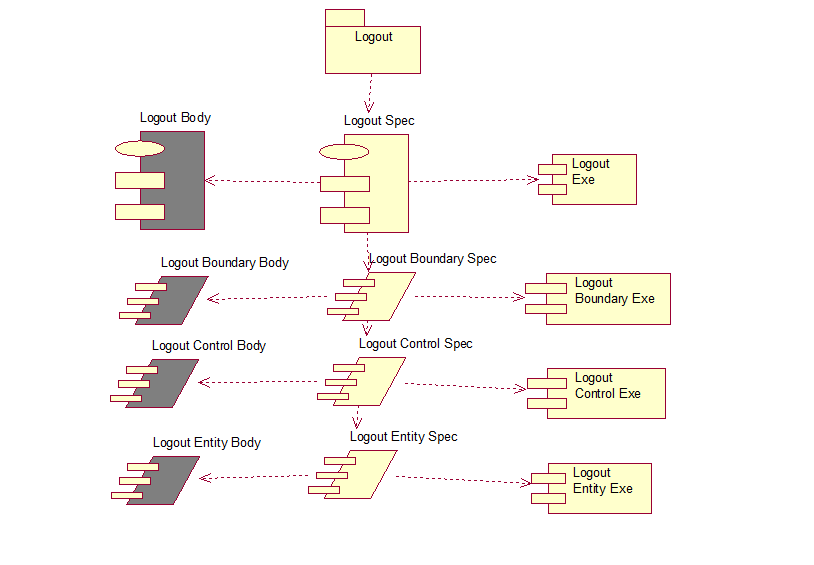
**Figure 5.30 Take Quiz Diagram**

**5.6.5 SCORE BOARD**



**Figure 5.31 Score Board Component Diagram**

**5.6.6 LOGOUT**

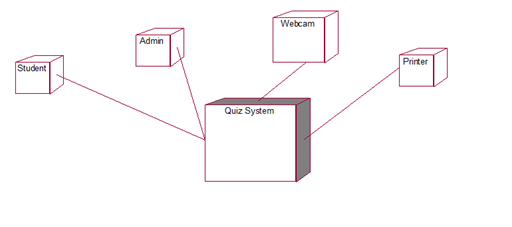


**Figure 5.32 Logout Component Diagram**

**5.7 DEPLOYMENT DIAGRAM**

Deployment diagram shows the configuration of run time processing elements and the software components, processes and objects that live in them.

**5.7.1 QUIZ SYSTEM**



**Figure 5.33 Quiz System Deployment Diagram**

**CHAPTER-6**

**SYSTEM DEVELOPMENT**

**6.1 LOGIN**

This feature of the product allows only the authentication user to work with the application. This ensures security and restricts any unauthorized persons from accessing the application.

**6.2 CREATE QUIZ**

This feature enables the authenticated Admin to create the quiz on any particular Subject , any number of question and he can also set the mark for correct question and wrong question.

**6.3 CHOOSE SUBJECT**

This feature enables the authentication Student to login their page and allow them to take quiz on subject which was created by Admin.

**6.4 TAKE QUIZ**

This feature enables the Student to Take quiz on the subject which they want.

**6.5 SCORE BOARD**

This feature will enable the both the student and admin to see their result in a particular page without any difficulty.

**6.6 LOGOUT**

This feature will enable the user or Admin to logout of their Page.

**CHATER-7**

**SYSTEM IMPLEMENTATION**

**7.1 SYSTEM PREREQUISITE**

* HARDWARE REQUIREMENT

1. Processor – Pentium II 433 MHz or above
2. RAM – 256 MB
3. Hard disk – 10 GB or above

* SOFTWARE REQUIREMENT

1. Operating system – Windows 10
2. Sublime
3. HTML & PHP
4. MySQL

**7.2 STEP TO INSTALL SOFTWARE**

1. Check whether the system has Windows 10 OS, if not install the OS in the corresponding system.

2. Check whether the system has Sublime, if not install the Sublime in the corresponding system.

3. Check whether the system has XAMPP , if not install the XAMPP in the corresponding system.

4. Check whether the system has MySQL, if not install MySQL in the corresponding system.

5. Then double click on the application to use it.

**CHAPTER-8**

**SYSTEM TESTING**

**8.1. TEST PLAN**

**8.1.1 pROJECT DESCRIPTION**

The four basic steps in project are

1. Extract the problem domain statement from the user.
2. Classify into nouns and verbs based on the Noun Phrase Approach.
3. Generate the classes.
4. Place the operations in appropriate classes.

**8.1.2 RELATED DOCUMENTS**

The related documents in our project are project proposal and plan, Software Requirements Specification, Software Design Specification and test plan.

**8.1.3 TESTING STRATEGY**

Unit test and Functional test is used in each phase of testing.

**8.1.4 TESTING LEVEL PLAN**

White Box Testing is given higher priority than Black Box Testing.

**8.1.5 UNIT TEST**

Unit test is generally accomplished by the person who wrote the code. Generally white box testing is used at this level, since errors in logic are easier to find when testing each path through the code.

**8.1.6 MODULE TEST**

Module testing is usually done by a single programmer or a small group of programmers writing units that work together in a single module. Test cases are often used for the module testing, especially if the module is self-contained.

**8.1.7 INTEGRATION TEST**

Usually, the development team handles integration test. Integration test is best accomplished incrementally, by adding one module at a time to isolate errors.

**8.1.8 ALPHA TEST**

In this test, “Internal” testers run live data through the system to shake out bugs not found in integration test, the customer may wish to observe the alpha test or to provide some of the “Real” data.

**8.1.9 BETA TEST**

Beta test is the next step, where the programmers released to the customer with the understanding that the program is still being tested. The customer agrees to stress the application and to report any discovered bugs or problems to the development team. The team agrees to be a “Friendly” user but to really put the system through its paces, trying to break it.

**8.1.10 REGRESSION TEST**

After the product is released, errors may be found, or enhancements suggested by the customers in the field. As these are corrected or implemented, the rest of the system must also be tested again to make sure that the new fixes did not break any of the old code. Regression test is usually an automated script that runs a set of test cases known to exercise the entire system.

**8.1.11 USABILITY TEST**

Usability testing is a special form of testing that looks for bugs not in the functionality of the program, but in the layout and utility of the user interface. This step is often a prototype before the actual system code is written, so it is easy to change if needed.

**CHAPTER-9**

**CONCLUSION AND FUTURE ENHANCEMENTS**

The Quiz System was implemented to reach a greater number of students and to take easier, Flexible and secure way to Take Test . In further, the Site could be improved by setting webcam to make more reliable. his project has helped us in getting a clearer understanding of real-world application development. o It has provided us a deeper insight into connecting databases with servers the entire learning outcome of this project has proved to be immensely beneficial for our future application development.

 In our future we are decided to provide more security to our website which may not be hacked.  And we give the choice to student to add their name under the faculty who they wish and get advice for their betterment. It will be more empowering. Next, we are aiming to provide some online classes into our website.

**APPENDIX 1**

**SAMPLE CODE**

<!---Index.php---!>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>| Online Quiz System |</title>

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="shortcut icon" type="image/png" href="image/logo.png" />

<style type="text/css">

body {

width: 100%;

background: url(image/book.png) ;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<body>

<center>

<div class="intro">

<h1> online quiz system </h1>

<a href="login.php" class="btn"> login </a> &emsp;

<a href="register.php" class="btn"> register </a>

<h2> Good &nbsp;Luck. </h2>

</div>

</center>

</body>

</html>

<!---login.php---!>

<?php

require('database.php');

session\_start();

if(isset($\_SESSION["email"]))

{

session\_destroy();

}

$ref=@$\_GET['q'];

if(isset($\_POST['submit']))

{

$email = $\_POST['email'];

$pass = $\_POST['password'];

$email = stripslashes($email);

$email = addslashes($email);

$pass = stripslashes($pass);

$pass = addslashes($pass);

$email = mysqli\_real\_escape\_string($con,$email);

$pass = mysqli\_real\_escape\_string($con,$pass);

$str = "SELECT \* FROM user WHERE email='$email' and password='$pass'";

$result = mysqli\_query($con,$str);

if((mysqli\_num\_rows($result))!=1)

{

echo "<center><h3><script>alert('Sorry.. Wrong Username (or) Password');</script></h3></center>";

header("refresh:0;url=login.php");

}

else

{

$\_SESSION['logged']=$email;

$row=mysqli\_fetch\_array($result);

$\_SESSION['name']=$row[1];

$\_SESSION['id']=$row[0];

$\_SESSION['email']=$row[2];

$\_SESSION['password']=$row[3];

header('location: welcome.php?q=1');

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Login | Online Quiz System</title>

<link rel="stylesheet" href="scripts/bootstrap/bootstrap.min.css">

<link rel="stylesheet" href="scripts/ionicons/css/ionicons.min.css">

<link rel="stylesheet" href="css/form.css">

<style type="text/css">

body{

width: 100%;

background: url(image/book.png) ;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<body>

<section class="login first grey">

<div class="container">

<div class="box-wrapper">

<div class="box box-border">

<div class="box-body">

<center> <h5 style="font-family: Noto Sans;">Login to </h5><h4 style="font-family: Noto Sans;">Online Quiz System</h4></center><br>

<form method="post" action="login.php" enctype="multipart/form-data">

<div class="form-group">

<label>Enter Your Email Id:</label>

<input type="email" name="email" class="form-control">

</div>

<div class="form-group">

<label class="fw">Enter Your Password:

<a href="javascript:void(0)" class="pull-right">Forgot Password?</a>

</label>

<input type="password" name="password" class="form-control">

</div>

<div class="form-group text-right">

<button class="btn btn-primary btn-block" name="submit">Login</button>

</div>

<div class="form-group text-center">

<span class="text-muted">Don't have an account?</span> <a href="register.php">Register</a> Here..

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<script src="js/jquery.js"></script>

<script src="scripts/bootstrap/bootstrap.min.js"></script>

</body>

</html>

<!---Logout.php---!>

<?php

session\_start();

if(isset($\_SESSION['email'])){

session\_destroy();}

$ref= @$\_GET['q'];

header("location:$ref");

?>

<!---Update.php---!>

<?php

include\_once 'database.php';

session\_start();

$email=$\_SESSION['email'];

if(@$\_GET['demail'])

{

$demail=@$\_GET['demail'];

$r1 = mysqli\_query($con,"DELETE FROM rank WHERE email='$demail' ") or die('Error');

$r2 = mysqli\_query($con,"DELETE FROM history WHERE email='$demail' ") or die('Error');

$result = mysqli\_query($con,"DELETE FROM user WHERE email='$demail' ") or die('Error');

header("Location: dashboard.php?q=1");

}

if(@$\_GET['q']== 'rmquiz')

{

$eid=@$\_GET['eid'];

$result = mysqli\_query($con,"SELECT \* FROM questions WHERE eid='$eid' ") or die('Error');

while($row = mysqli\_fetch\_array($result))

{

$qid = $row['qid'];

$r1 = mysqli\_query($con,"DELETE FROM options WHERE qid='$qid'") or die('Error');

$r2 = mysqli\_query($con,"DELETE FROM answer WHERE qid='$qid' ") or die('Error');

}

$r3 = mysqli\_query($con,"DELETE FROM questions WHERE eid='$eid' ") or die('Error');

$r4 = mysqli\_query($con,"DELETE FROM quiz WHERE eid='$eid' ") or die('Error');

$r4 = mysqli\_query($con,"DELETE FROM history WHERE eid='$eid' ") or die('Error');

header("location:dashboard.php?q=5");

}

if(@$\_GET['q']== 'addquiz')

{

$name = $\_POST['name'];

$name= ucwords(strtolower($name));

$total = $\_POST['total'];

$sahi = $\_POST['right'];

$wrong = $\_POST['wrong'];

$id=uniqid();

$q3=mysqli\_query($con,"INSERT INTO quiz VALUES ('$id','$name' , '$sahi' , '$wrong','$total', NOW())");

header("location:dashboard.php?q=4&step=2&eid=$id&n=$total");

}

if(@$\_GET['q']== 'addqns')

{

$n=@$\_GET['n'];

$eid=@$\_GET['eid'];

$ch=@$\_GET['ch'];

for($i=1;$i<=$n;$i++)

{

$qid=uniqid();

$qns=$\_POST['qns'.$i];

$q3=mysqli\_query($con,"INSERT INTO questions VALUES ('$eid','$qid','$qns' , '$ch' , '$i')");

$oaid=uniqid();

$obid=uniqid();

$ocid=uniqid();

$odid=uniqid();

$a=$\_POST[$i.'1'];

$b=$\_POST[$i.'2'];

$c=$\_POST[$i.'3'];

$d=$\_POST[$i.'4'];

$qa=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$a','$oaid')") or die('Error61');

$qb=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$b','$obid')") or die('Error62');

$qc=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$c','$ocid')") or die('Error63');

$qd=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$d','$odid')") or die('Error64');

$e=$\_POST['ans'.$i];

switch($e)

{

case 'a': $ansid=$oaid; break;

case 'b': $ansid=$obid; break;

case 'c': $ansid=$ocid; break;

case 'd': $ansid=$odid; break;

default: $ansid=$oaid;

}

$qans=mysqli\_query($con,"INSERT INTO answer VALUES ('$qid','$ansid')");

}

header("location:dashboard.php?q=0");

}

if(@$\_GET['q']== 'quiz' && @$\_GET['step']== 2)

{

$eid=@$\_GET['eid'];

$sn=@$\_GET['n'];

$total=@$\_GET['t'];

$ans=$\_POST['ans'];

$qid=@$\_GET['qid'];

$q=mysqli\_query($con,"SELECT \* FROM answer WHERE qid='$qid' " );

while($row=mysqli\_fetch\_array($q) )

{ $ansid=$row['ansid']; }

if($ans == $ansid)

{

$q=mysqli\_query($con,"SELECT \* FROM quiz WHERE eid='$eid' " );

while($row=mysqli\_fetch\_array($q) )

{

$sahi=$row['sahi'];

}

if($sn == 1)

{

$q=mysqli\_query($con,"INSERT INTO history VALUES('$email','$eid' ,'0','0','0','0',NOW())")or die('Error');

}

$q=mysqli\_query($con,"SELECT \* FROM history WHERE eid='$eid' AND email='$email' ")or die('Error115');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

$r=$row['sahi'];

}

$r++;

$s=$s+$sahi;

$q=mysqli\_query($con,"UPDATE `history` SET `score`=$s,`level`=$sn,`sahi`=$r, date= NOW() WHERE email = '$email' AND eid = '$eid'")or die('Error124');

}

else

{

$q=mysqli\_query($con,"SELECT \* FROM quiz WHERE eid='$eid' " )or die('Error129');

while($row=mysqli\_fetch\_array($q) )

{

$wrong=$row['wrong'];

}

if($sn == 1)

{

$q=mysqli\_query($con,"INSERT INTO history VALUES('$email','$eid' ,'0','0','0','0',NOW() )")or die('Error137');

}

$q=mysqli\_query($con,"SELECT \* FROM history WHERE eid='$eid' AND email='$email' " )or die('Error139');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

$w=$row['wrong'];

}

$w++;

$s=$s-$wrong;

$q=mysqli\_query($con,"UPDATE `history` SET `score`=$s,`level`=$sn,`wrong`=$w, date=NOW() WHERE email = '$email' AND eid = '$eid'")or die('Error147');

}

if($sn != $total)

{

$sn++;

header("location:welcome.php?q=quiz&step=2&eid=$eid&n=$sn&t=$total")or die('Error152');

}

else if( $\_SESSION['key']!='suryapinky')

{

$q=mysqli\_query($con,"SELECT score FROM history WHERE eid='$eid' AND email='$email'" )or die('Error156');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

}

$q=mysqli\_query($con,"SELECT \* FROM rank WHERE email='$email'" )or die('Error161');

$rowcount=mysqli\_num\_rows($q);

if($rowcount == 0)

{

$q2=mysqli\_query($con,"INSERT INTO rank VALUES('$email','$s',NOW())")or die('Error165');

}

else

{

while($row=mysqli\_fetch\_array($q) )

{

$sun=$row['score'];

}

$sun=$s+$sun;

$q=mysqli\_query($con,"UPDATE `rank` SET `score`=$sun ,time=NOW() WHERE email= '$email'")or die('Error174');

}

header("location:welcome.php?q=result&eid=$eid");

}

else

{

header("location:welcome.php?q=result&eid=$eid");

}

}

if(@$\_GET['q']== 'quizre' && @$\_GET['step']== 25 )

{

$eid=@$\_GET['eid'];

$n=@$\_GET['n'];

$t=@$\_GET['t'];

$q=mysqli\_query($con,"SELECT score FROM history WHERE eid='$eid' AND email='$email'" )or die('Error156');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

}

$q=mysqli\_query($con,"DELETE FROM `history` WHERE eid='$eid' AND email='$email' " )or die('Error184');

$q=mysqli\_query($con,"SELECT \* FROM rank WHERE email='$email'" )or die('Error161');

while($row=mysqli\_fetch\_array($q) )

{

$sun=$row['score'];

}

$sun=$sun-$s;

$q=mysqli\_query($con,"UPDATE `rank` SET `score`=$sun ,time=NOW() WHERE email= '$email'")or die('Error174');

header("location:welcome.php?q=quiz&step=2&eid=$eid&n=1&t=$t");

}

?>

<!---Dashboard.php---!>

<?php

include\_once 'database.php';

session\_start();

if(!(isset($\_SESSION['email'])))

{

header("location:login.php");

}

else

{

$name = $\_SESSION['name'];

$email = $\_SESSION['email'];

include\_once 'database.php';

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Dashboard | Online Quiz System</title>

<link rel="stylesheet" href="css/bootstrap.min.css"/>

<link rel="stylesheet" href="css/bootstrap-theme.min.css"/>

<link rel="stylesheet" href="css/welcome.css">

<link rel="stylesheet" href="css/font.css">

<script src="js/jquery.js" type="text/javascript"></script>

<script src="js/bootstrap.min.js" type="text/javascript"></script>

</head>

<body>

<nav class="navbar navbar-default title1">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="Javascript:void(0)"><b>Online Quiz System</b></a>

</div>

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-left">

<li <?php if(@$\_GET['q']==0) echo'class="active"'; ?>><a href="dashboard.php?q=0">Home<span class="sr-only">(current)</span></a></li>

<li <?php if(@$\_GET['q']==1) echo'class="active"'; ?>><a href="dashboard.php?q=1">User</a></li>

<li <?php if(@$\_GET['q']==2) echo'class="active"'; ?>><a href="dashboard.php?q=2">Ranking</a></li>

<li class="dropdown <?php if(@$\_GET['q']==4 || @$\_GET['q']==5) echo'active"'; ?>">

<li><a href="dashboard.php?q=4">Add Quiz</a></li>

<li><a href="dashboard.php?q=5">Remove Quiz</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li <?php echo''; ?> > <a href="logout1.php?q=dashboard.php"><span class="glyphicon glyphicon-log-out" aria-hidden="true"></span>&nbsp;Log out</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="row">

<div class="col-md-12">

<?php if(@$\_GET['q']==0)

{

echo "<h1> WELCOME TO Admin Page!!

</h1>";

}

if(@$\_GET['q']== 2)

{

$q=mysqli\_query($con,"SELECT \* FROM rank ORDER BY score DESC " )or die('Error223');

echo '<div class="panel title"><div class="table-responsive">

<table class="table table-striped title1" >

<tr style="color:red"><td><center><b>Rank</b></center></td><td><center><b>Name</b></center></td><td><center><b>Score</b></center></td></tr>';

$c=0;

while($row=mysqli\_fetch\_array($q) )

{

$e=$row['email'];

$s=$row['score'];

$q12=mysqli\_query($con,"SELECT \* FROM user WHERE email='$e' " )or die('Error231');

while($row=mysqli\_fetch\_array($q12) )

{

$name=$row['name'];

$college=$row['college'];

}

$c++;

echo '<tr><td style="color:#99cc32"><center><b>'.$c.'</b></center></td><td><center>'.$e.'</center></td><td><center>'.$s.'</center></td>';

}

echo '</table></div></div>';

}

?>

<?php

if(@$\_GET['q']==1)

{

$result = mysqli\_query($con,"SELECT \* FROM user") or die('Error');

echo '<div class="panel"><div class="table-responsive"><table class="table table-striped title1">

<tr><td><center><b>S.N.</b></center></td><td><center><b>Name</b></center></td><td><center><b>College</b></center></td><td><center><b>Email</b></center></td><td><center><b>Action</b></center></td></tr>';

$c=1;

while($row = mysqli\_fetch\_array($result))

{

$name = $row['name'];

$email = $row['email'];

$college = $row['college'];

echo '<tr><td><center>'.$c++.'</center></td><td><center>'.$name.'</center></td><td><center>'.$college.'</center></td><td><center>'.$email.'</center></td><td><center><a title="Delete User" href="update.php?demail='.$email.'"><b><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></b></a></center></td></tr>';

}

$c=0;

echo '</table></div></div>';

}

?>

<?php

if(@$\_GET['q']==4 && !(@$\_GET['step']) )

{

echo '<div class="row"><span class="title1" style="margin-left:40%;font-size:30px;color:#fff;"><b>Enter Quiz Details</b></span><br /><br />

<div class="col-md-3"></div><div class="col-md-6">

<form class="form-horizontal title1" name="form" action="update.php?q=addquiz" method="POST">

<fieldset>

<div class="form-group">

<label class="col-md-12 control-label" for="name"></label>

<div class="col-md-12">

<input id="name" name="name" placeholder="Enter Quiz title" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="total"></label>

<div class="col-md-12">

<input id="total" name="total" placeholder="Enter total number of questions" class="form-control input-md" type="number">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="right"></label>

<div class="col-md-12">

<input id="right" name="right" placeholder="Enter marks on right answer" class="form-control input-md" min="0" type="number">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="wrong"></label>

<div class="col-md-12">

<input id="wrong" name="wrong" placeholder="Enter minus marks on wrong answer without sign" class="form-control input-md" min="0" type="number">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for=""></label>

<div class="col-md-12">

<input type="submit" style="margin-left:45%" class="btn btn-primary" value="Submit" class="btn btn-primary"/>

</div>

</div>

</fieldset>

</form></div>';

}

?>

<?php

if(@$\_GET['q']==4 && (@$\_GET['step'])==2 )

{

echo '

<div class="row">

<span class="title1" style="margin-left:40%;font-size:30px;"><b>Enter Question Details</b></span><br /><br />

<div class="col-md-3"></div><div class="col-md-6"><form class="form-horizontal title1" name="form" action="update.php?q=addqns&n='.@$\_GET['n'].'&eid='.@$\_GET['eid'].'&ch=4 " method="POST">

<fieldset>

';

for($i=1;$i<=@$\_GET['n'];$i++)

{

echo '<b>Question number&nbsp;'.$i.'&nbsp;:</><br /><!-- Text input-->

<div class="form-group">

<label class="col-md-12 control-label" for="qns'.$i.' "></label>

<div class="col-md-12">

<textarea rows="3" cols="5" name="qns'.$i.'" class="form-control" placeholder="Write question number '.$i.' here..."></textarea>

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'1"></label>

<div class="col-md-12">

<input id="'.$i.'1" name="'.$i.'1" placeholder="Enter option a" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'2"></label>

<div class="col-md-12">

<input id="'.$i.'2" name="'.$i.'2" placeholder="Enter option b" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'3"></label>

<div class="col-md-12">

<input id="'.$i.'3" name="'.$i.'3" placeholder="Enter option c" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'4"></label>

<div class="col-md-12">

<input id="'.$i.'4" name="'.$i.'4" placeholder="Enter option d" class="form-control input-md" type="text">

</div>

</div>

<br />

<b>Correct answer</b>:<br />

<select id="ans'.$i.'" name="ans'.$i.'" placeholder="Choose correct answer " class="form-control input-md" >

<option value="a">Select answer for question '.$i.'</option>

<option value="a"> option a</option>

<option value="b"> option b</option>

<option value="c"> option c</option>

<option value="d"> option d</option> </select><br /><br />';

}

echo '<div class="form-group">

<label class="col-md-12 control-label" for=""></label>

<div class="col-md-12">

<input type="submit" style="margin-left:45%" class="btn btn-primary" value="Submit" class="btn btn-primary"/>

</div>

</div>

</fieldset>

</form></div>';

}

?>

<?php

if(@$\_GET['q']==5)

{

$result = mysqli\_query($con,"SELECT \* FROM quiz ORDER BY date DESC") or die('Error');

echo '<div class="panel"><div class="table-responsive"><table class="table table-striped title1">

<tr><td><center><b>S.N.</b></center></td><td><center><b>Topic</b></center></td><td><center><b>Total question</b></center></td><td><center><b>Marks</b></center></td><td><center><b>Action</b></center></td></tr>';

$c=1;

while($row = mysqli\_fetch\_array($result)) {

$title = $row['title'];

$total = $row['total'];

$sahi = $row['sahi'];

$eid = $row['eid'];

echo '<tr><td><center>'.$c++.'</center></td><td><center>'.$title.'</center></td><td><center>'.$total.'</center></td><td><center>'.$sahi\*$total.'</center></td>

<td><center><b><a href="update.php?q=rmquiz&eid='.$eid.'" class="pull-right btn sub1" style="margin:0px;background:red;color:black"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span>&nbsp;<span class="title1"><b>Remove</b></span></a></b></center></td></tr>';

}

$c=0;

echo '</table></div></div>';

}

?>

</div>

</div>

</div>

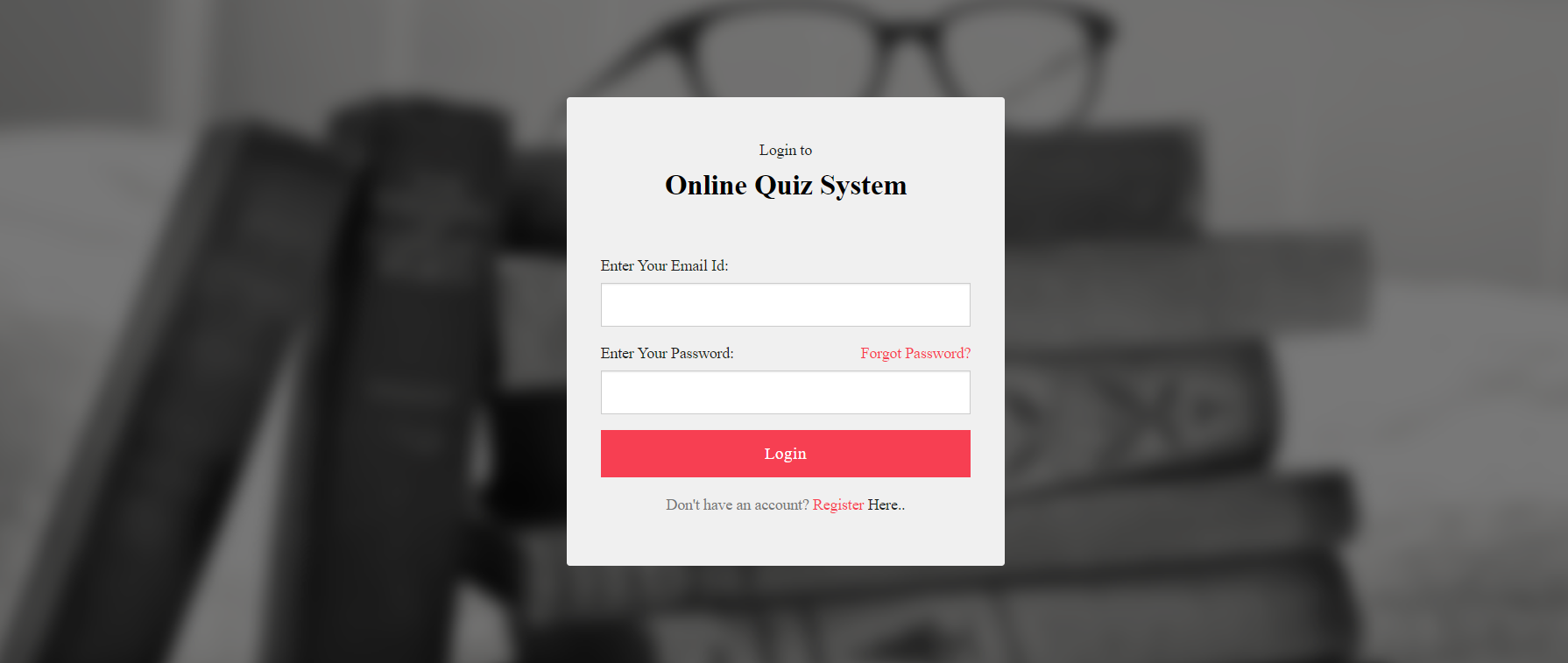
</body>

</html>

**APPENDIX 2**

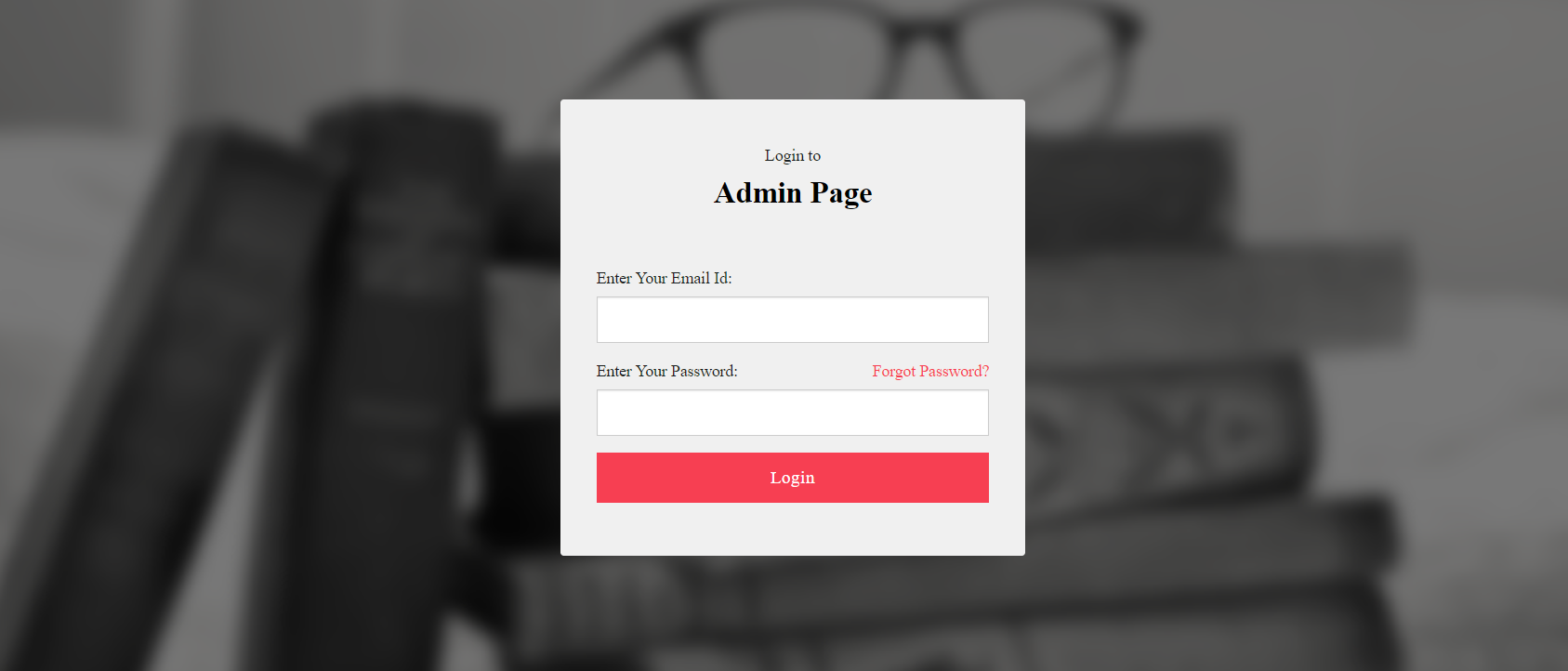
**SAMPLE SCREEN SHOTS**

**Student Login Screen**

****

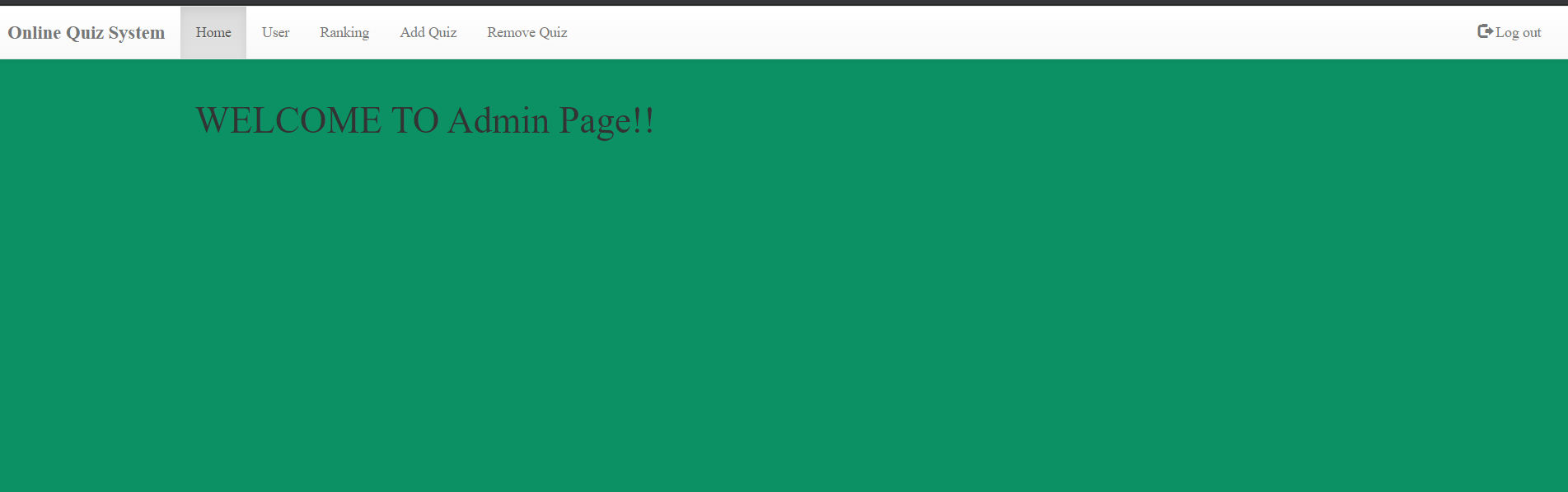
**Figure A 1.1 Student Login Screen**

**Admin Login window**



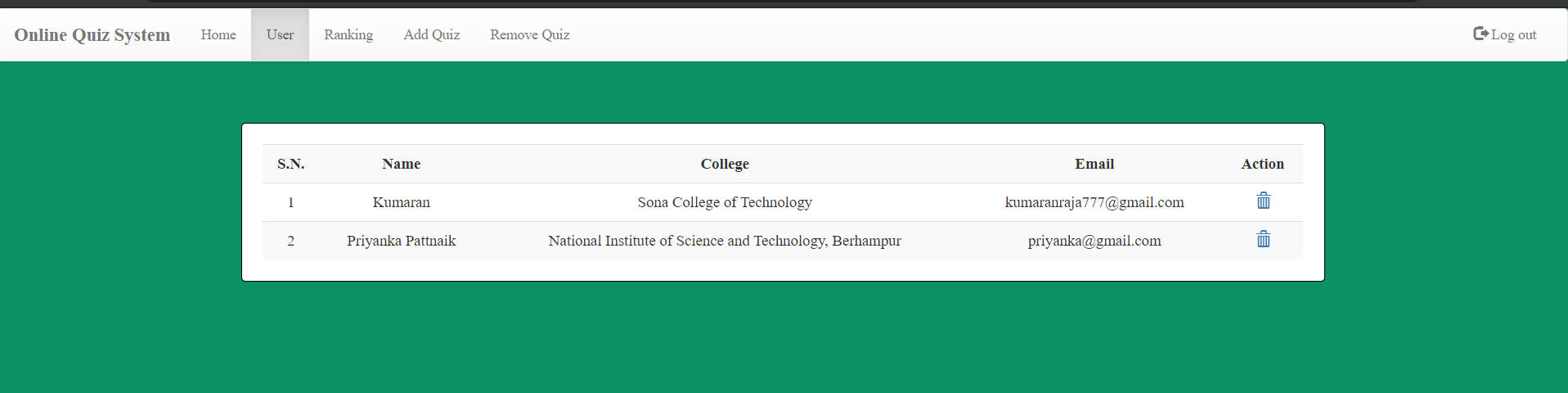
**Figure A 1.2 Admin Login window**

**Admin Home window**



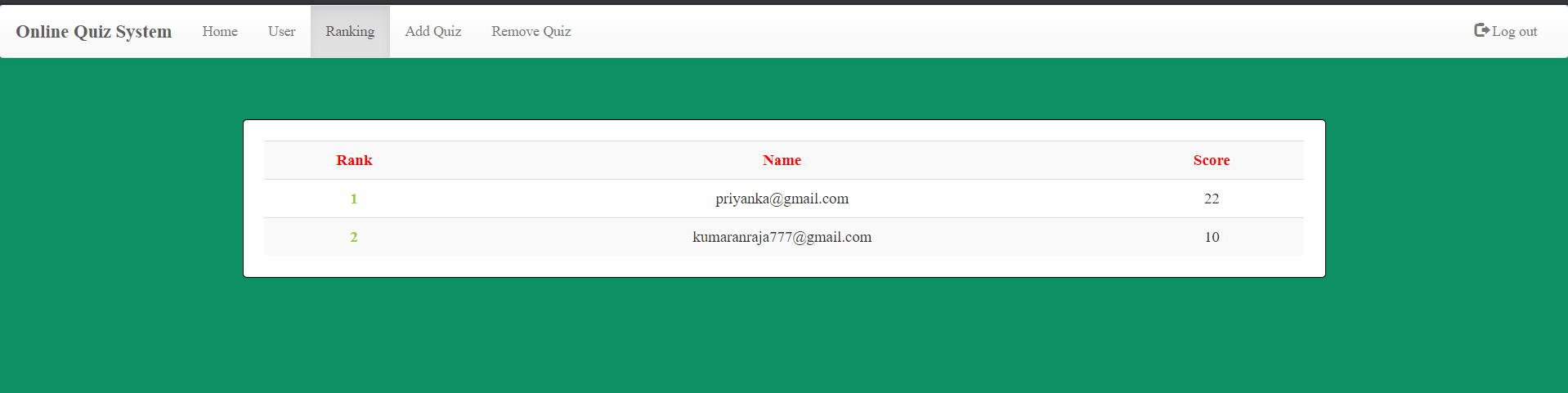
**Figure A 1.3 Admin Home window**

**Admin user window**



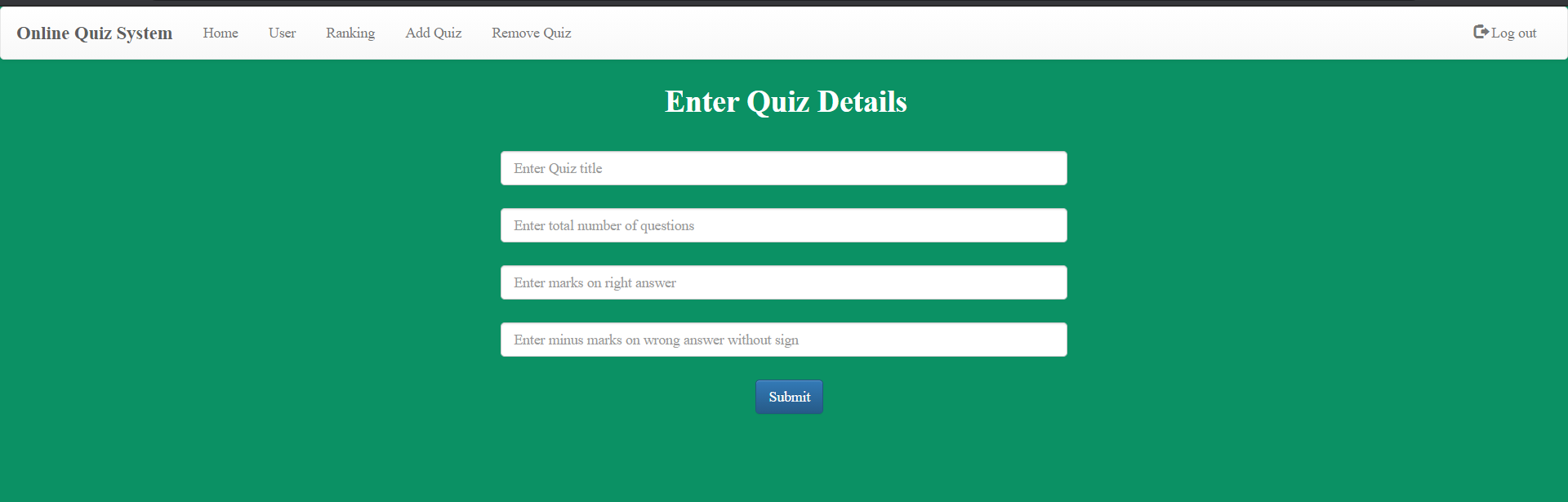
**Figure A 1.4 Admin user withdraw**

**Admin Ranking window**



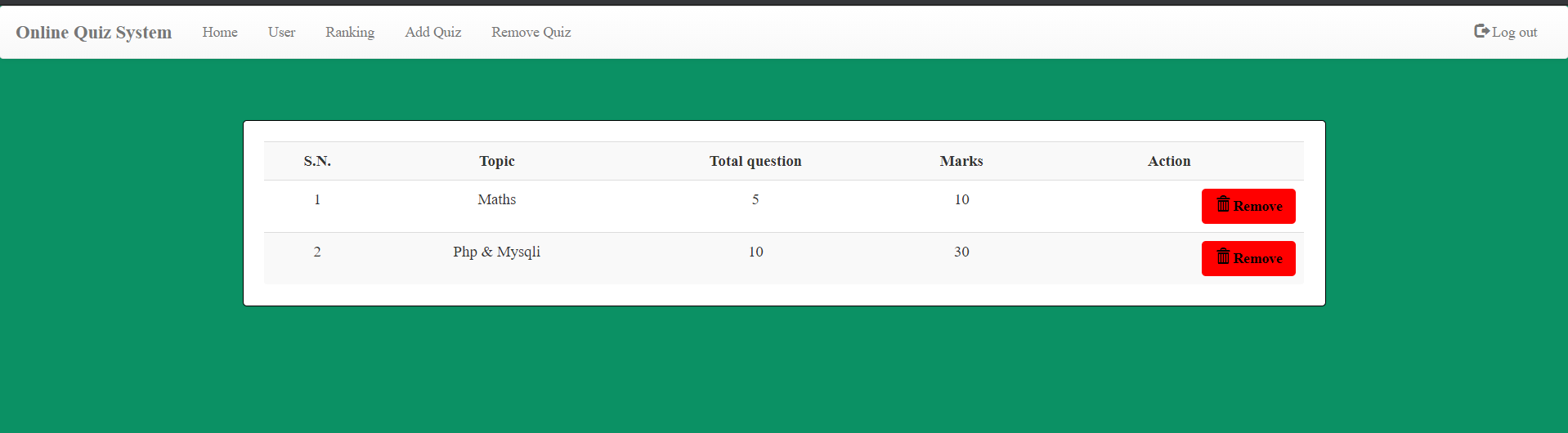
**Figure A 1.5 Admin Ranking Window**

**Add Quiz window**

****

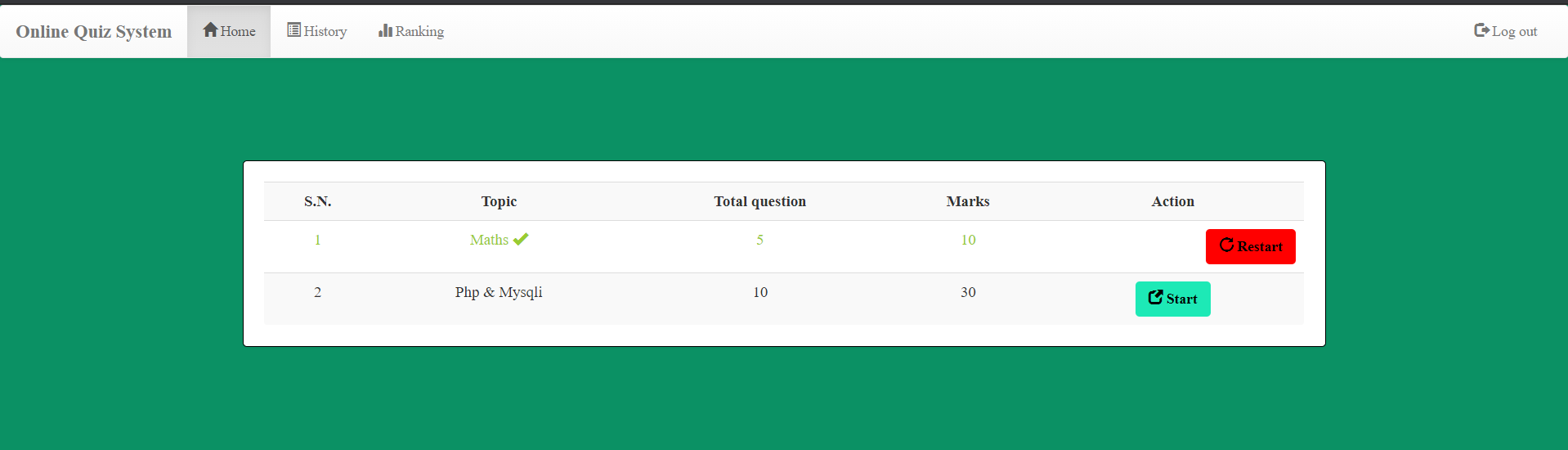
**Figure A 1.5 Add Quiz window**

**Remove Quiz window**

****

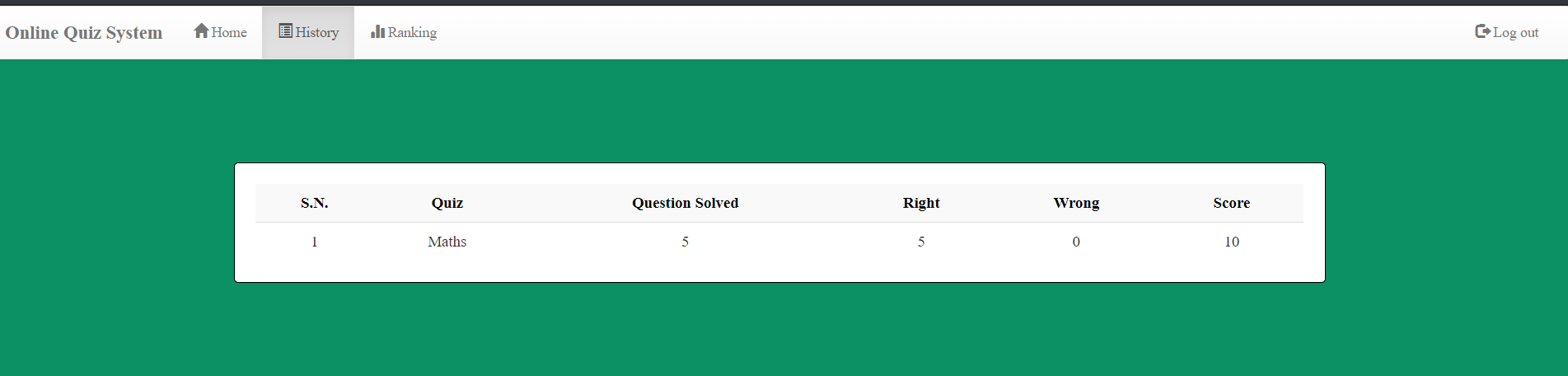
**Figure A 1.6 Remove Quiz window**

**Student Home window**

****

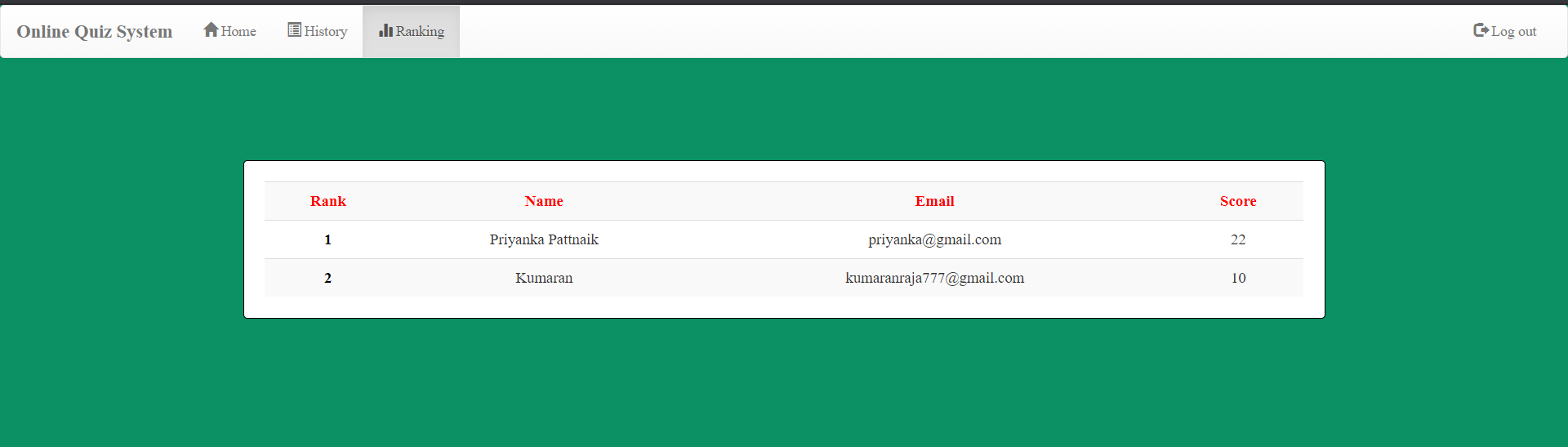
**Figure A 1.7 Student Home window**

**Student History window**

****

**Figure A 1.8 Student History window**

**Student Ranking window**

****

**Figure A 1.9 Student Ranking window**