University of Leicester

MULTIPLE CHOICE QUESTION AND ANSWER WEB APPLICATION

Mehmet Yasar Bayraktar

MAY 13, 2016

UNIVERSITY OF LEICESTER COMPUTER SCIENCE

University of Leicester

University of Leicester Computer Science Individual Project

DISSERTATION

PROJECT TITLE:

Multiple Choice Question and Answer Web App

Word Count: 11266

Student Name:

Mehmet Yasar Bayraktar

Username:

Myb1

Student Number:

149046276

Project Supervisor
Dr. Tom Ridge

Second Marker
Dr. Rob Van Stee

13/05/2016

Table of Contents

Table of Figures: iv
Abstract: 1
Declaration: 1
Acknowledgments: 2
1) Introduction: 3
1.1) Project Introduction: 3
1.2) Aims: 3
1.3) Objectives: 4
1.4) Challenges: 4
1.4.1) Reliability: 4
1.4.2) Look and Feel: 5
1.4.3) Functionalities and Database Structure: 5
1.4.4) Security: 5
1.4.5) Requirements: 6
1.5) Technical Information: 6
1.5.1) Entity Framework Introduction: 7
1.6) Working Plan: 8
2) Literature Review: 9
2.1) Background: 9
2.2) Existing Applications: 9
3) Project Requirements and Design: 11
3.1) The Purpose of the Project: 11
3.2) Users of MULCA: 11
3.3) Project Constraints: 12
3.4) Relevant Information: 13
3.5) UML Diagram of Application: 14
3.5.1) Use Case Diagram: 14
3.5.2) Elevator Statement for this Project: 15
3.5.3) User Story: 15
3.5.4) Applying User Stories to the Project 16
3.6) Activity Diagram of Application: 17
3.7) Entity Relationship Diagram: 20
3.7.1) Some Entities of the Project: 21

3.7.2) Relationships between Tables: 23
3.8) User Interface Design: 25
3.8.1) Login and Authorisation: 26
3.8.2) Choosing an Active Role: 27
3.8.3) Dashboard Page: 27
3.8.4) Lecturer Pages: 28
3.8.5) Students Pages: 30
3.8.6) Admin Pages: 31
3.8.7) User Pages: 33
4) Project Implementation: 35
4.1) Technical Tools for Implementation: 35
4.2) Source Code Excerpts: 37
4.2.1) Login with Cookie: 38
4.2.2) Adding a New Question: 39
4.2.3) Uploading a List of Users from an Excel Page: 40
4.2.4) Online Student Score Page: 41
5) Project Testing: 42
5.1) Authorisation Test: 42
5.2) Course Management Test: 43
5.3) Quiz Management Test: 43
5.4) Question Management Test: 44
5.5) Exam Full Page (Exam Code) Test: 44
5.6) Excel Upload Test: 45
6) Summary and Conclusion: 46
7) Source Code References: 48
7.1) High Chart: 48
7.2) Excel Upload: 48
7.3) Image Upload: 48
7.4) Image Ratio: 48
7.5) Atlant Template: 48
7.6) Sql JOB: 48
7.7) Use Case Diagram: 48
7.8) Captcha: 49
7.9) Encryption-Decryption Hash: 49
7.10) Email Sending: 49

7.11) File Process: 49
8) References 50

Table of Figures:

Figure 1: Working Plan 8

Figure 2: Use Case Diagram of the Exam Application 14

Figure 3: Structure of User Story 15

Figure 4: Authorisation Activity Diagram 18

Figure 5: Teacher's/Lecturer's Activity Diagram 19

Figure 6: Student Activity Diagram 20

Figure 7: One-to-Many Relation Example 24

Figure 8: ER Diagram of the Exam Application 25

Figure 9: User Authorisation 26

Figure 10: Choosing an Active Role 27

Figure 11: Dashboard for Users 28

Figure 12: Creating a New Quiz for Students 28

Figure 13: Adding New Question Page 29

Figure 14: Students Active Exams 30

Figure 15: Active Exam Tab Control 30

Figure 16: A Student's Score in a 3D Bar Chart 31

Figure 17: Menu Management 32

Figure 18: Menu Authorization 33

Figure 19: Massive User Upload 33

Figure 20: Notifications Page 34

Figure 21: User Suggestion Page 34

Figure 22: A Part of the Entity Framework Diagram 36

Figure 23: Architecture of the Project (3-Tiers) 37

Figure 24: N-tier Architecture Representation (N-TIER DESIGNS) 37

Figure 25: Authorization for the Application 38

Figure 26: Save User Identity and IP as a Cookie 38

Figure 27: Authorization for the Application 39

Figure 28: Reading Data from Excel File 40

Figure 29: Using the Web Method to Get Data from a Server 41

Figure 30: Total Number of Correct and Incorrect Answer 42

Abstract:

Online examination is a fundamental part of the online education, which enables the evaluation of education quality and students learning outcomes. Nowadays, many universities and education centres are trying to get involved with online education by allowing the students to take online courses/modules. Online testing and examination are also one of the major parts of the online education system. Therefore, an online exam application was developed which helps lecturers/teachers to create web-based exams/tests and measure their students' learning outcomes. A Microsoft SQL Server 2014 database was used to store the data of the application by using normalization rules. The C# language and Asp.net platform were chosen for development of the web application. In conclusion, a web-based exam application was released with many valuable functionalities such as uploading a list of users using with excel sheet, reporting students results and feedback in an elegant platform and showing the correct and incorrect answers of the students during exams'.

Declaration:

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specially acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Mehmet Yasar Bayraktar

Date: 13/05/2016

Acknowledgments:

I would like to acknowledge that I am greatly indebted to and must thank all of my lecturers and classmates at the University of Leicester who spent their extra time and effort educating me, especially to my supervisors **Dr. Tom Ridge** and **Dr. Rob Van Stee,** who were helpful by their ideas, which have driven me forward to complete my individual project.

Additionally thank you to my parents **Sevilay** and **Huseyin Bayraktar** and my siblings Mahir, Murat and Hilal for their social support and prayers. I believe that without their trust in me, I would not be able to continue my academic career. Moreover, I would like to wish all of my housemates (Dr. Mesut, Esat, and Idris) great success in life.

Finally; I am indebted to the Turkish Ministry of Education (MEB), who supported me with my living expenses and tuition fees. I will be happy to return and work with them. Without them, I would not have been able to study in the UK or meet the variety of cultures and friends I have met.

1) Introduction:

1.1) Project Introduction:

Web applications provide a cheaper and a more efficient means of accomplishing tasks. Online applications can enable one to measure the students' abilities and learning outcomes. One of the best ways to measure learning outcomes is through tests/quizzes/exams. These help lecturers/teachers/examiners to understand which topics have been understood and to what extent. Moreover, traditional testing requires a lot of intermediary processes, such as creating a test paper and making numerous copies of it. During this process, errors are likely to occur in the questions or designs, and examiners then have to correct their mistakes and repeat the same process again. Hence, it can cause waste of time and resources. Furthermore, asking assistants to copy a test or exam paper is not secure; there is the possibility of the leaking of exam questions. On the other hand, an online test makes this process easy and secure. It also has some additional advantages, such as allowing users to change the order of questions and their choices. So, during the exam, if any student tries to cheat by peeking at another student's answers, it will be harder for them to find the right question and therefore its answer. Additionally, it makes teachers' jobs relatively easy, as they can add many questions to the test and then request that the server generate a randomly chosen set of questions for each student/class, which will help them to create a lot of different tests from the list of questions which have already been uploaded. This feature can help teachers to save their time by using a question bank.

1.2) Aims:

The aims of this project are as follows:

- To offer online tests and exams to students.
- To reduce the use of paper and pencil protects the environment.
- To get feedback from students thanks to online exams and tests.
- To understand the percentage of students that have understood a topic.
- To give individual feedback for every student to improve their understanding.
- To give an automatic explanation of the questions which have been answered rightly or wrongly by the students.
- To notify users of upcoming tests and under what exam conditions via email.
- To be able to give a lecturers an independent platform.

1.3) Objectives:

The objectives for achieving the aims of this project are as follows:

- Lecturers and students should be able to register online by using the registration page.
- Lecturers should be able to create basic types of question with or without an image.
- Lecturers should be able to create a list of users, who will take the online test by uploading excel sheet files.
- Lecturers will be able to review exams online and give some feedback.
- Lecturers should be able to add new students to their courses.
- To provide a better way of making announcements using a forum for each course.
- Students will be notified by the online system when there is an upcoming test or exam.
- Students will be able to see the timer during the exam and when they have five minutes remaining, the panel which shows the remaining time will give some notification by changing its colour.
- Every user will have a profile name and picture, which can be updated by the user so that they can be recognised or remembered by profile name and picture.
- The administrator will be able to access the control panel which allows him or her to change menu authorisations and to create new menus, so, he/she can decide which menu will appear for which role.
- Password information should be encrypted or hashed in the database, so people who work with the database should not able to see other people's passwords.
- To protect the web application from malicious computer attack, captcha will be used to identify whether the client is a computer or a human being.

1.4) Challenges:

1.4.1) Reliability:

The multiple choice application should be able to execute its functions and method with minimal errors. When the application encounters an error, it should be able to handle it by redirecting the user to an error page which is user friendly. This page should clearly describe the error to the user/client, and an error code should be logged by the system so the developers are aware of the problem and can fix it.

1.4.2) Look and Feel:

When a user interacts with this application they should enjoy surfing the website. So, it is a rational choice to start with a new template which looked well designed (you can find it at 'http://aqvatarius.com/themes/index.php?template=atlant').The Atlant template is flexible and can be used on tablet or mobile and at any screen resolution. The template was designed using HTML5; additionally it is well documented, which makes it understandable and customisable.

1.4.3) Functionalities and Database Structure:

One the major difficulties of this project is that there are more than four types of questions (e.g. 'multiple choice single answer', 'fill in the flanks', 'true-false', 'multiple choice multiple answer', 'will be scored by teacher/lecturer') which makes it harder to create a web page that allows one to add/update each of these types of questions. Moreover, the database structure has to be normalised (there are three main types of normalisation rules which are helpful for avoiding redundancy and repetition) to add/delete questions to/from the database.

1.4.4) Security:

This application should have a security framework which protects the website from malicious attacks. An asp.net framework can be very helpful regarding these issues. Also, adding more security features will make the website more secure. In some cases, developers (or even hackers) who are working with the database will have access to the user information (i.e. password and username combinations). So, it is more secure to keep the password information 'hashed and salted'. Then, nobody will be able to reach the decrypted password information. If a user forgets his/her password, even he/she has to ask for a new password which demonstrates that there is not an easy way to find out the current password. Additionally, it is more secure to use captcha, so, the application does not allow machines or automated programs to infiltrate into the system.

1.4.5) Requirements:

1.4.5.1) Essential Requirements:

- Providing basic question types for the tests.
- Login and registration for students and lecturers.
- Developing security and authorisation controls.
- Providing a basic page for students to see exam questions and answer them.

1.4.5.2) Recommended Requirements:

- Joining the test by providing a security code.
- Automatically saving each answer when working on the next question.
- Calculating scores dynamically and giving default feedback for the test.
- Giving specific feedback for the open-ended questions.
- Creating a page for the lecturer to review the students' paper again.

1.4.5.3) Optional Requirements:

- Keeping log information of all activities and who has done them.
- Adding questions with an image.
- Showing students' results using bar or pie charts.
- Notifying the user who will take the test.
- Showing multiple questions on a single page and using a pager.
- Registering multiple users using excel sheets.

1.5) Technical Information:

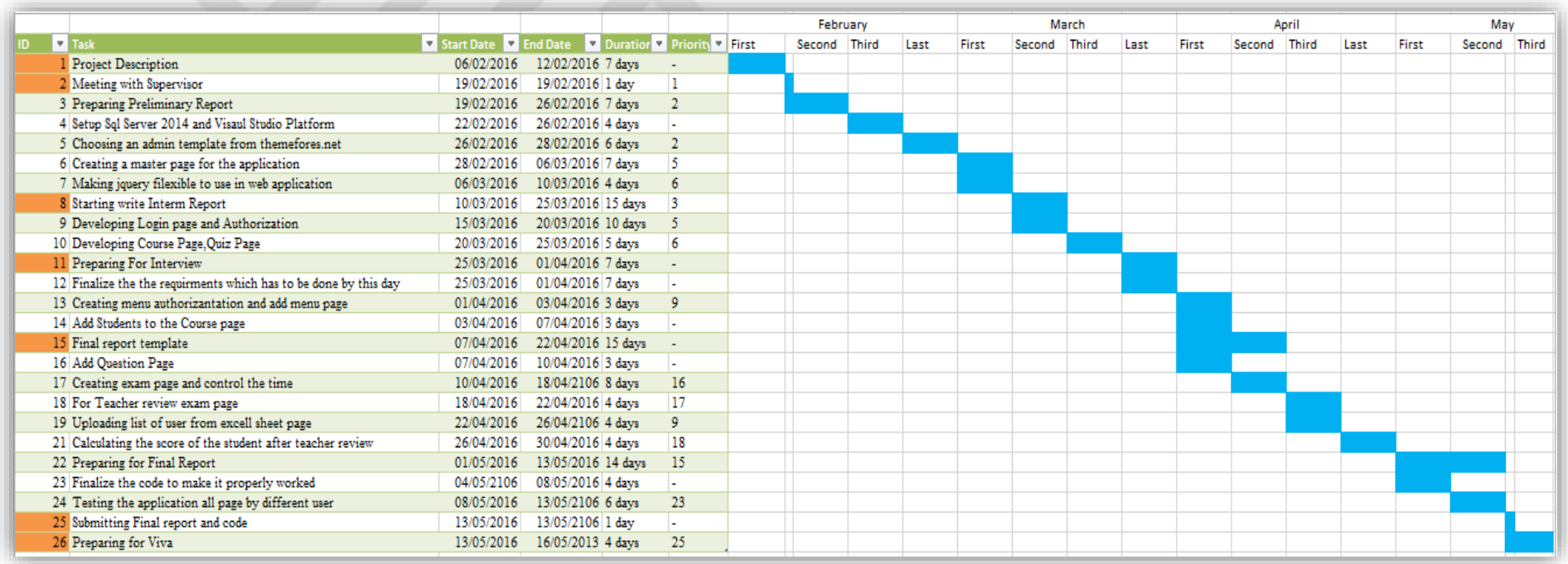
As a developer or software company, it is easy to understand that starting from scratch is time consuming (Siddharth, 2009, p. 1). Since, certain big companies have offered powerful frameworks, it is better to focus on implementing one of these frameworks and adding some features to it. This way, the development team will have enough time to applying new features and functionalities. Moreover, it is known that almost every framework comes up with Authentication, Session Management and Database Access Control, which allows the companies to create web applications easily and to be protected from many problems. (Choosing a Web Development Framework, 2009)

In conclusion; choosing a framework is a long-term choice; it is better to make the right choice for the future of the development and the company (Siddharth, p. 1). Therefore, we decided to choose asp.net and the Entity framework for the multiple choice test application. This is a very powerful framework from the point of security and authentication control. It can also be developed using two languages: C# and Visual Basic.

1.5.1) Entity Framework Introduction:

The Entity framework is an ORM (Object Relational Mapper) which generates entities and their relations from a database. It uses LINQ syntax, which is a uniform programming model that allows the use of 'CRUD'(Create, Read, Update, Delete) (Sheriff, 2001) operation queries to manage and manipulate the data by take the database type (MSSQL, ORACLE, MYSQL) in to account (Introducing Microsoft LINQ, 2007). LINQ is also an impressive tool which enables us to embed the SQL queries into the source code. Thereby, making it easier to manage and create the workflow process within of the source code. Moreover, the LINQ model generates a new object for each table of the database, allowing developers the flexibility to easily use OOP (Object Oriented Programming) to create more understandable and readable code. Additionally, they will be able to reach fields in every table, regardless of the syntax error, thanks to the Intellisense, which allows a developer to see the list of members to complete the word.

1.6) Working Plan:



ID	Task	Start Date	End Date	Duration	Priority
1	Project Description	06/02/2016	12/02/2016	7 days	-
2	Meeting with Supervisor	19/02/2016	19/02/2016	1 day	1
3	Preparing Preliminary Report	19/02/2016	26/02/2016	7 days	2
4	Setup Sql Server 2014 and Visaul Studio Platform	22/02/2016	26/02/2016	4 days	-
5	Choosing an admin template from themefores.net	26/02/2016	28/02/2016	6 days	2
6	Creating a master page for the application	28/02/2016	06/03/2016	7 days	5
7	Making jquery filexible to use in web application	06/03/2016	10/03/2016	4 days	6
8	Starting write Interm Report	10/03/2016	25/03/2016	15 days	3
9	Developing Login page and Authorization	15/03/2016	20/03/2016	10 days	5
10	Developing Course Page,Quiz Page	20/03/2016	25/03/2016	5 days	6
11	Preparing For Interview	25/03/2016	01/04/2016	7 days	-
12	Finalize the the requirments which has to be done by this day	25/03/2016	01/04/2016	7 days	-
13	Creating menu authorizantation and add menu page	01/04/2016	03/04/2016	3 days	9
14	Add Students to the Course page	03/04/2016	07/04/2016	3 days	-
15	Final report template	07/04/2016	22/04/2016	15 days	-
16	Add Question Page	07/04/2016	10/04/2016	3 days	-
17	Creating exam page and control the time	10/04/2016	18/04/2106	8 days	16
18	For Teacher review exam page	18/04/2016	22/04/2016	4 days	17
19	Uploading list of user from excell sheet page	22/04/2016	26/04/2106	4 days	9
21	Calculating the score of the student after teacher review	26/04/2016	30/04/2016	4 days	18
22	Preparing for Final Report	01/05/2016	13/05/2016	14 days	15
23	Finalize the code to make it properly worked	04/05/2106	08/05/2016	4 days	-
24	Testing the application all page by different user	08/05/2016	13/05/2106	6 days	23
25	Submitting Final report and code	13/05/2016	13/05/2106	1 day	-
26	Preparing for Viva	13/05/2016	16/05/2013	4 days	25

Figure 1: Working Plan

Figure 1, above, demonstrates the working plan of the project. To ensure we were able to finish ‘Multiple Choice Test Application’ on the time, we have created this working plan, which makes it easy to understand for the project manager to see whether the project is behind of the plan or not. The brown cells identify the times which has defined by the study guide.

2) Literature Review:

2.1) Background:

Thanks to recent technological developments, it is relatively easy to create web applications and/or find a company which specializes in developing them. Nowadays, there are also many professional developers and object-oriented languages (e.g. Java, C#) which can help us to create multifunctional web applications. Therefore, the uses of web applications have increased rapidly in the last decade (Appappeal, 2011), and it is easy to observe that nearly all companies are starting to provide services using web applications or via mobile "apps". Moreover, these companies encourage their customers to use their web applications by giving free gifts and/or discounts in order to reduce their costs (e.g. labour). In addition, many companies that use traditional desktop applications have recognised that they need additional applications which are independent from the platform because they increasingly require business functionalities during their (personnel's) travels, as the world of business is becoming more and more international.

Web-based applications can also be used in exam and survey settings. Therefore, we decided to create a web application that helps students to take online exams. This will enable us to reduce the time and paper spent when carrying out exams. In addition, taking online courses is becoming more and more popular (e.g. MOOCs). Many students want to obtain education/qualifications from abroad (Rynson W.H Lau, 2005, p. 99); however, many do not have the time nor the funds to go abroad. Thus, targeting online examinations could be a big business opportunity to attract students' attention.

2.2) Existing Applications:

There are some successful web applications on the internet that enable students to take online courses and exams. SCORM (Sharable Content Object Reference Model) Cloud is one of the best examples of online education. The SCORM app allows lecturers to create learning contents and exams. The application uses cloud SaaS (Software as a Service), which makes it available for developers and customers without them having to spend too much time or resources. However, it is important to know that SCORM sets a price based on registration instead of per user or PC. This means that when a lecturer has 10 (in total) exams/quizzes for his/her course, he/she has to pay 10 times more for each student. So it is important to take

course content into account when you are calculating the relative/owning cost of the course (SCORM Cloud Pricing).

Another of the best online applications is ProProfs Quiz Maker, which allows users to create more than seven types of question (such as multiple choice, checkbox, and video-audio). Also, it is possible to add an image or video to each question. Moreover, if you need to find a new question as a lecturer, it has its own library which enables you to access thousands of "ready-made" questions by providing a topic name. Additionally, this application has certain smart configurations which permit certificate customisation, setting quiz notifications and choosing security preferences. Furthermore, after creating a quiz, it is easy to share it via email or social account. And it is accessible across a wide range of devices such as laptops, tablets, and mobile (ProProf Quiz Maker).

Last but not least is CAQ (Create a Quiz/Test Maker). It is a mobile (android) application which facilitates the management of your quizzes using mobile devices. It includes export and import features, and it is possible to download its desktop application to work on your personal computer. Thanks to this mobile application, learning and studying are more entertaining and interesting (CAQ (Create a Quiz/Test Maker)).

3) Project Requirements and Design:

In this section, we will determine the project design and requirements of the Multiple Choice Question and Answer Application using the Volere requirement specifications. In this way we can create the functional and non-functional requirements in detail. The requirement and design phase is crucial for a development project, as it allows one to understand customer needs. Also, the requirement phase decreases the cost of the application, as it is less expensive to correct a bug found at this phase than later during the implementation or testing phases.

3.1) The Purpose of the Project:

Since, the exponential growth of web applications in the last quarter of the 20th Century, different sectors of society have moved towards using web applications to be up-to-date, MULCA (**Mul**tiple **C**hoice Questions and Answer **A**pplication) is an online web application that enables the creation of quizzes to test the students` learning outcomes. This application has been exclusively developed for the universities and colleges who have participants from all around the world beyond the physical borderlines.

3.2) Users of MULCA:

Users of this application have specific roles, which enables the application to identify their authorisations and limits. Therefore, they can see which pages they are able to access and which functionalities or features are available for them.

One of the fundamental roles is “Teacher”, which allows the user to create new quizzes and add different types of question to those quizzes. The second crucial role is “Student”, which enables users to access quizzes, to see their feedback for each question, and to see their results for each quiz. These are two types of role which are essential to Multiple Choice Application Question and Answer Application. The application also has other types of role to make it more qualified. One of them is the “Administrator” role, which enables admin to create basic authorisation and security rules for the application. For example, an administrator can add a new menu or a new page and identify which types of user can access this menu. Moreover, he or she can use bulk upload, which allows users to add thousands of users to the database in a short period of time. So companies who use this application will not have to spend time and money transferring their users to the system. Also, there is other fundamental work for administrators and programmers. They should able to see log information which identifies

malicious attacks or suspicious behaviour. Thus, the Administrator and the Programmer can check those suspicious behaviours to find out what the problem behind them are. Also, it would be better to create new roles which allow the user to test all applications, both as a student and as a teacher/lecturer. We can call these roles "StudentTester" and "TeacherTester". In this application, we will use the "multiple to multiple" relationship between the users table and the role table, which enables users to have multiple roles. So, a person can be student and teacher at the same time. Therefore, he or she does not have to create new account for a new role. Also, this relationship will allow us to add new roles and delete roles from users. Thus, the administrator of the system can change user authorisations by adding and deleting roles.

3.3) Project Constraints:

The project will be developed using the asp.net platform and C# language because of their flexibility and the researchers' experience with them. However, the project will have some constraints, certain elements which will be hard to apply to the project because of the asp.net platform. Additionally, the application server has to be installed on the windows server computer to be able to run the project. MSSQL works with windows products very efficiently, so it is reasonable to choose the MSSQL server as a database. However, if we want to migrate the database to Oracle or MySQL Database we can easily apply this process thanks to the LINQ database's flexibility.

Moreover, there are some types of authentication technique which allow the system to identify the user and whether he or she is the person he or she claims to be (Abrar, p. 1). There are three major ways to determine users, which are: Knowledge-Based Identification, Object-Based Identification and Biometrics- or Characteristics-Based Identification. When a system uses Object-Based Identification, the user has to be authenticated by citing or displaying a certain electronic chip, magnetic card or barcode. A system can also use Biometrics- or Characteristics-Based Identification to determine the user's characteristics. These can be "fingerprints, face recognition and e-signature" (Abrar, p. 1). Since we have limited time and budget, it might be hard and expensive to apply these types of authentication. Moreover, some users may wear scarves or make-up, making it difficult for the system to carry out face recognition. Therefore, it was decided to use Knowledge-Based Identification to identify the user. This type of identification works based on identical information. Username and password or email and password are marvellous examples for this purpose, and this is easy to implement in the application.

3.4) Relevant Information:

The application does not allow anonymous users to have access to it. If they want to test or analyse the system, they have to create an account as a “Teacher/Lecturer” or “Student”. Therefore, we can keep log information for their user identities. We could also create a default account as a student or teacher so they would be able to check the website and see what functionalities it has. Additionally; keeping log information will help developers to find software bugs and fix them. Moreover, when the developers recognise unauthorised behaviour they can trace that behaviour thanks to the log table.

Secondly; web applications can come under the malicious attacks and software security might not prevent these attacks; or the database of the application can entirely change because of an incorrect SQL query. In this case, the web application has to recover itself under the leadership of the database administrator by using backup files. So the database server has to create a backup file for every day or every hour. SQL Server Agent is perfectly suited for this purpose. SQL Server Agent is a tool of the MSSQL Server which enables the management of SQL Job and SQL Warning (Mike Gunderloy, 2006). SQL Job will be used to create backup files and save them to a folder. We will use the hard drive “D:\MyBackup” to make them securer. Also, the database administrator has to identify when SQL Job should be executed. It is recommended to choose a time after rush hours to avoid interrupting the users.

Thirdly; it might be better if the application supports multiple languages so it can be used world-wide. However, since English is a very common language all over the world, and lecturers can create their question contexts using their own languages, there is no urgent need for language preferences.

3.5) UML Diagram of Application:

A UML (Unified Modelling Language) diagram enables one to visualise how a system works and describes the interaction between the system's actors and components. This will make what the customer needs and which type of features/functions each user should have clearer for the developer (or software company). However, it is important to be aware that a UML diagram demonstrates what the application is supposed to do; it does not illustrate how to implement it (Schmuller, 1999, p. 5).

3.5.1) Use Case Diagram:

There are many desktop and online applications that can be used to draw helpful use case diagrams. In this project, a web application (Creately, URL: 'https://creately.com') was used to visualise relations between the components and the users of the exam application. (Creately, 2016).

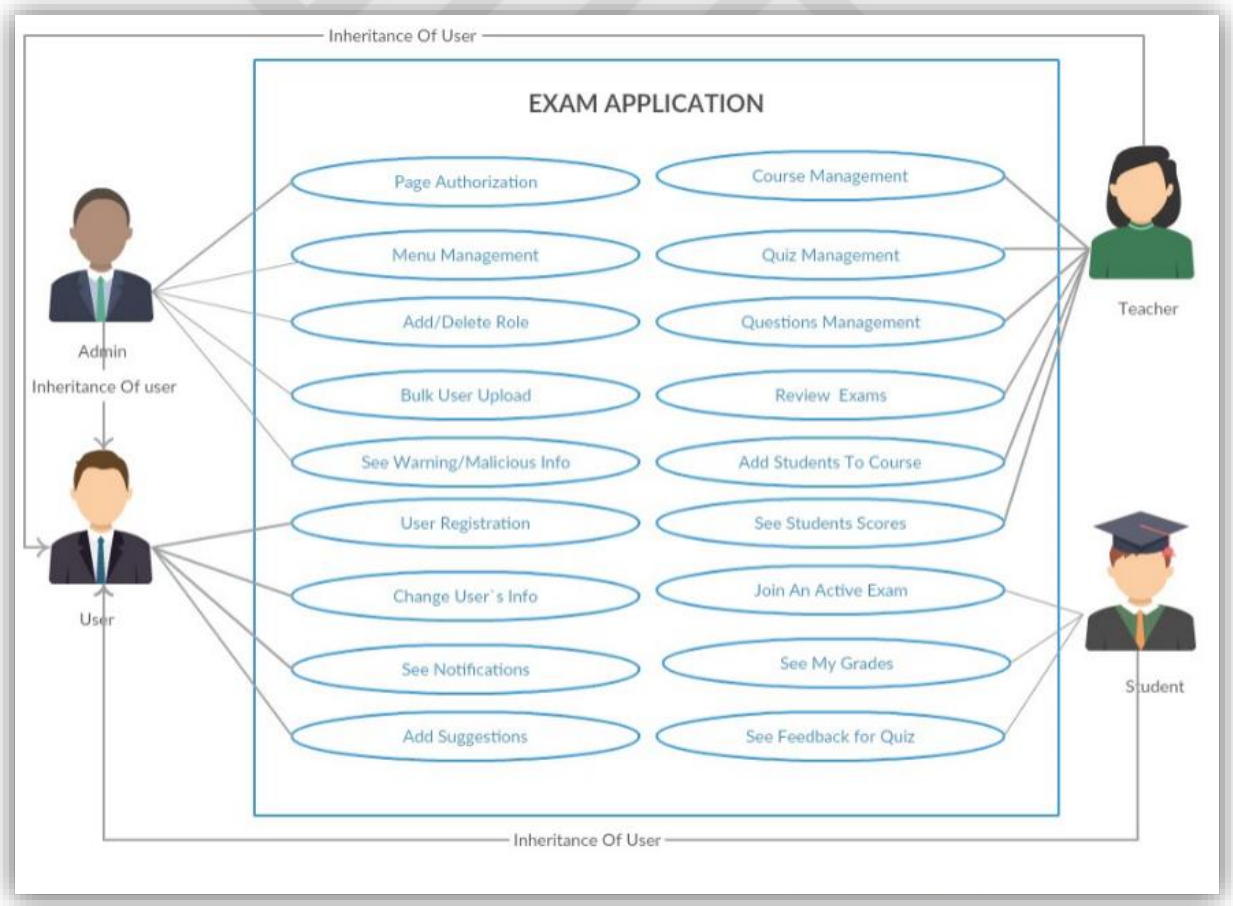


Figure 2: Use Case Diagram of the Exam Application.

3.5.2) Elevator Statement for this Project:

An elevator statement is a brief explanation of a product used to convince customers. This pitch should include the benefits of the product, tailored to the target customers, what the name of product is and what makes this product different from other competitors (Yubas, 2013). An example is given for the exam application:

'For **an education company** which **needs to measure students' learning outcomes and the success of education** with various criteria, **MULCA** is a web-based exam application that is specifically developed for **university and high schools** in the UK. Unlike our competitor '**SCORM'**, our product prevents cheating activity and ensures fairness by keeping track of the exam-tab status. Thus, when a student tries to check others' papers or documents during the exam, it will be evaluated as a cheating activity and the lecturer will be notified about this behaviour.'

3.5.3) User Story:

A user story card gives us information about which type of user needs which type of functionality and the goal of this function or feature. It is a small card with a short amount of basic information on it which can be shared by developers. Thus, developers do not have to spend time checking the entire set of requirements again. Figure 3, below, demonstrates the structure of the user story (Walkinshaw, p. 8).

Figure 3: Structure of User Story

3.5.4) Applying User Stories to the Project

Since 'User Stories' enable us to subdivide entire sets of requirements into small story cards, it is reasonable to apply them to this project to make it easier to develop code or create a page for every story. Examples are given below for every type of user.

3.5.4.1) Admin Users Stories:

As an Admin, I want to change and update menu or page authorisations, **so that** I will be able to disable/enable certain menus for certain type of roles.

As an Admin, I want to use bulk user upload, **so that** I will be able to save hundreds of users' information to the system without spending time, money or workforce.

As an Admin, I want to see log information for every user, **so that** I will be able to see suspicious behaviours on the system.

3.5.4.2) Lecturer/Teacher Users Stories:

As a Lecturer, I want to create my courses, **so that** I will be able to add my students to my course.

As a Lecturer, I want to create certain basic types of questions (multiple choices, open-ended, fill in the blanks), **so that**, I will be able to generate new exams for students.

As a Lecturer, I want to review exams online and give some feedback, **so that** I will be able to evaluate exams scores from all around the world regardless of physical location.

As a Lecturer, I want to create notification for my students, **so that** I will be able to remind them about new online exam

3.5.4.3) Student Users Stories:

As a Student, I want to be notified about exams, **so that** I will be aware of when there is an exam.

As a Student, I want to see a timer and a progress bar during the exam, **so that** I will be able to manage my time and strategy by checking the time and my progress.

As a Student, I want to see my active exam, **so tha**t I will be able to join my exam easily by providing the exam code.

3.5.4.4) Default Users Stories:

As a User, I want to change my profile information and picture, **so that** I will be able to update my information and profile picture.

As a User, my password should be kept hashed or encrypted, **so that**, my password information cannot be captured by another person.

As a User, I should be able to register using a registration form, **so that** I will be able to get some courses or tests.

3.6) Activity Diagram of Application:

Activity diagrams enable one to represent the dynamic behaviour of an application using a graphical interface (Daniel R. Windle, 2002, p. 82). The two main elements of activity diagrams are activities and transactions. It is practical to name activity diagrams using short, meaningful words which describe the behaviour of the activity diagram. This makes it easier for other people to understand the activities. These short and meaningful words are called activities; descriptions of how to move from one activity to another are called transactions. Activity diagrams are powerful diagrams that allow the description of parallel activities using "fork" and "join" tools.

The Multiple Choice Question and Answer Application has three main types of user, which are Admin, Teacher and Student. These three types of user interact with the same interface to be authorised to use the application. They all have to provide their username/email and password information correctly, after which they will be redirected to the "Choose Role" page. This page allows each user to choose his/her role (or one of his/her roles, if he/she has more than one). Almost all users have just one role; however, in some cases, a user can have multiple roles, such as Teacher and Student or Admin and Teacher. So the application allows access to the applicable pages by checking (and authorising) the selected user's role. If a user has no authority for a certain page, the application will force the user to choose another role by redirecting him/her to the "Choose Role" page.

Once a user has chosen an active role, he/she will be redirected to the default page of the active role. For example, if the user's role is "Teacher", he/she will be redirected to the "My Courses" page; he/she will be redirected to the "My Grade" page if he/she is a student. In some cases, the user may be logged out because of a session time-out or another security factor, but the

application will allow him/her to log in again and will then redirect him/her to the page he/she was working on previously (i.e. just before being "timed out").

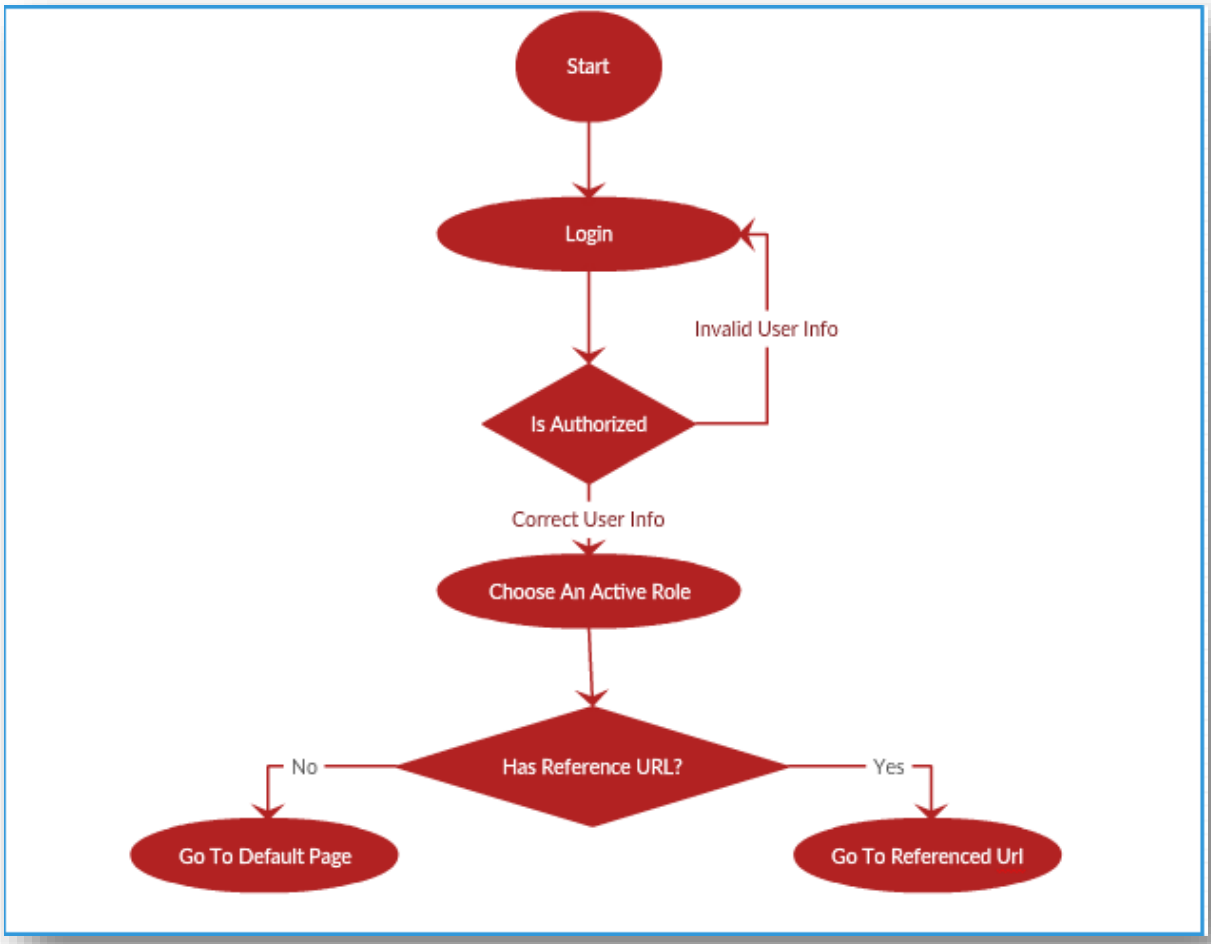


Figure 4: Authorisation Activity Diagram

As a teacher/lecturer, when one is authorised the application, one should be offered functions or interfaces which facilitate the creation of new courses, quizzes, and questions. Therefore, the teacher/lecturer can relatively easily use these functions without much training or time spent. Moreover, after adding some questions, lecturers should be able to view and update these questions. When the lecturers are creating their questions, they should also be able to provide default feedback for each wrong answer. Therefore, after the exam, students will receive some feedback about the questions straight away, and this will increase their learning outcomes. Additionally, if lecturers can check in real-time (i.e. during the exam) how many people have

answered the questions correctly, they can prevent the students from misunderstanding the questions and/or give more details about the questions to eliminate ambiguity.

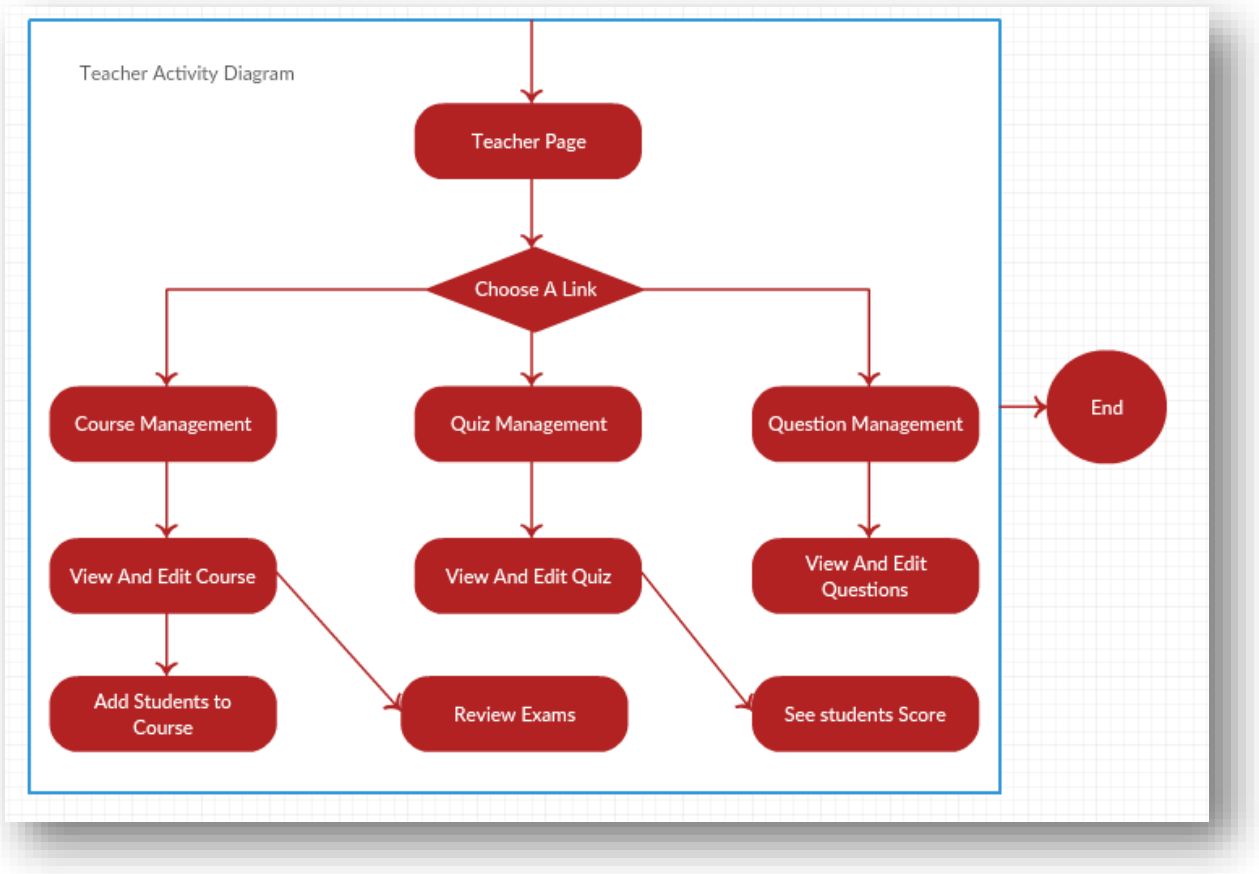


Figure 5: Teacher's/Lecturer's Activity Diagram

As a student, one should be able to see the active exams for today. So, when a teacher provides the students with an exam code, they can enter the exam by entering that code. Without the exam code, students will not be able to enter the exam or check the exam questions. It is a private key to the exam and the teacher must not share it before the exam starts. After verifying the exam code, the students' exam will start. One of most common security problems in exams – even if there are assistants who check on the students – is that some students try to find ways to cheat. It is also known that web browsers allow users to open new sections/tabs to search during the exam. For that reason, it is fundamentally essential to create a security control which checks the activity of the current tab. Whether for cheating reasons and other reasons, if one of the students changes their exam tab or inactivates it to check other documents, the application will count this as a cheating activity, and the assistants will be informed about it during the

exam, thus they can check those student more carefully. Figure 6 demonstrates the activity diagram for students.

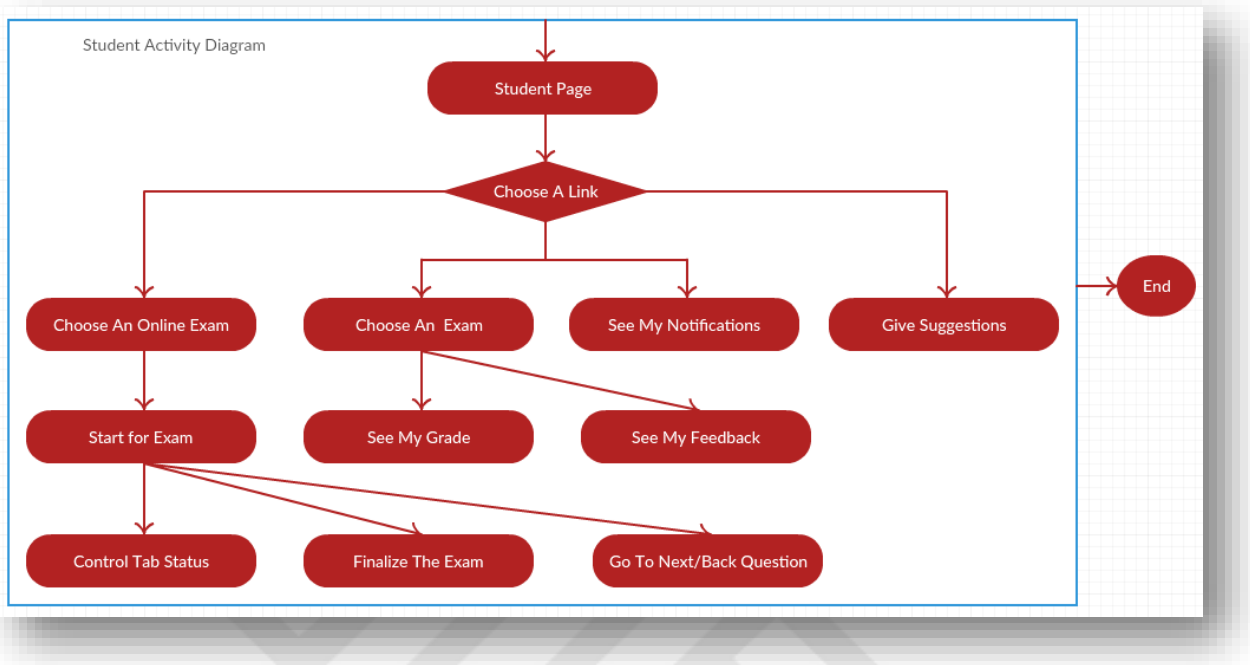


Figure 6: Student Activity Diagram

3.7) Entity Relationship Diagram:

50 years ago; almost 100 percent of any big company's information was stored in notebooks or on paper. (Patricia Ward, 2006, p. 2) They used clever techniques to reach this information, such as indexing and clustering. Thankfully, nowadays, it is really easy to save lots of information to a database, and the database enables one to search, delete, and update specific data in a short time. Since database management is a fundamental part of this project, it is reasonable to start with a relational, object-oriented database to represent the data structure of the 'Multiple Choice Questions and Answer Application'.

Identifying the requirements of the project made it easier to create the basic structure of the database. Entity Relation (ER) diagrams are one of best and most common models/tools for developing well-structured databases. ER diagrams are helpful to create database infrastructure. One of the best parts of a relational database is that it is uncomplicated and therefore easy for inexperienced people to figure out the basic structure of entities and relationships (Sikha Bagui, 2012, p. 26). These elements of database diagrams enable database designers to illustrate their project requirements from the perspectives of entities and

relationships (Sikha Bagui, 2012, p. 26). However, many educational reports demonstrate that numerous students who take database courses have difficulty when they implement real-world problems to the database structure (Sikha Bagui, 2012, p. 26).

3.7.1) Some Entities of the Project:

There are certain primary entities which have fundamental roles in the system. We will explain those entities and their relationships with each other. Before we give this information, it will be helpful to remark that almost every entity have certain shared common fields, which are IsActive, Deleted, InsertDate, UpdateDate, CreaterUserID and UpdaterUserID. These fields allow one to understand whether those information is active/deleted, who inserted/updated this record, and when those changes happened. Therefore, we can get detailed information about all records. Also, this will be helpful for finding out security gaps.

- ✓ **Tbl_Main_User:** This entity enables the system to store users' data. It includes fields such as UserID, UserName, FirstName, Email, PasswordHash. Moreover, this table allows the system to keep login information on all users. So, when a user wants to login to the application, he/she must provide an email/username and password, and the web application will check the 'Tbl_Main_User' table to find the authorisation status of that user.
- ✓ **Tbl_Main_Role:** This table gives information about user roles in the system. We can add additional roles and we can update some role by using this table.
- ✓ **Tbl_Main_UserInRole:** In this application 'Many to Many' relationship is used between user and role tables. This allows us to assign more than one roles to any user. Therefore, a user can be a teacher/lecturer or administrator on this system or this user can be student and teacher/lecturer at the same time. Thanks to the 'Tbl_Main_UserInRole' table, we can determine the roles of the user by providing UserID and RoleID fields.
- ✓ **Tbl_Main_User_Photo:** The profile pictures of users are recorded to the database as binary files.
- ✓ **Tbl_Main_User_Log:** This table records all activities of the application. It works as a kind of tracer. For example, when a new user logs in or registers, this information is recorded to this table. It has certain fields which gives detailed information about the records. These are the 'LogDescription', 'LogType', 'UserInfo' and 'UserIP' fields. Moreover,'LogType' information is helpful, warning the developers about security

problems or malicious attacks (such as 'Dangerous Log') so, they can find out those problems and fix them. Additionally, sometimes anonymous users or hackers try to guess passwords by testing; in this case, this table keeps records about the anonymous users, so security developers can find out which IP they are using and how many times they have tried to attack the system.

- ✓ **Tbl_Main_MailSettings:** It is easy to set up a dynamic email setting thanks to this table. So, when an administrator wants to change email sender, he/she just needs to update the new email information by using the interface for email settings; there will be no need to access the code-side and change email sender information.
- ✓ **Tbl_Main_LeftMenu:** This project enables users to use dynamic menus for different roles. The administrator can easily add new menus and determine the authorisation for the menu (the menu differs between the roles).The '**Tbl_Main_LeftMenuRoleAuthorization**' table is used to determine the authorisation for each roles.

The main tables, which have been mentioned above, can be used for any web application. However, we need additional tables to implement the 'Quiz Application' in our project. This has been addressed with other fundamental tables which are beneficial for keeping records about the quizzes and courses.

- ✓ **Tbl_Exm_Course:** This stores information about each course and its lecturer.
- ✓ **Tbl_Exm_Quiz:** This gives detailed information about each quiz, how many questions each quiz has, and how many minutes students will have for each exam.
- ✓ **Tbl_Exm_QuizQuestion**: This is a table that stores data about the questions and their types. Also, it gives information about the 'marks for the questions' and 'explanations of the questions` answers'. The project enables the certain of four types of questions, which are 'Fill in the blanks', 'Multiple Choice Single Answer', 'Multiple Choice Multiple Answer' and 'True-False' questions.
- ✓ **Tbl_Exm_Student_Answer:** During the exam and after the exam we store information on the students` answers and their scores for every question. Moreover, this table store teachers`/lecturers' feedback or default feedback.
- ✓ **Tbl_Exm_QuizQuetion_Options:** It is crucial to minimise data redundancy. Since we have a dynamic number of choices for a question, it was better to create new table to

store every choice, so we have created the '**Tbl_Exm_QuizQuetion_Options**' table to reduce data redundancy and null columns.

- ✓ **Tbl_Exm_UserQuizResults:** After finalising the review of the exam by teacher/lecturer, we need to store the students' results. Therefore, we can send notifications and emails to students about their exam results, and, they can access their final scores using this table.
- ✓ **Tbl_Main_UserNotification:** We can store notification information by using users' identities thanks to this table, so users will be able to access notifications and make them as 'seen' or 'read'.

There are other tables to store additional information. It is easy to understand them by skimming them, so we will not give explanations of them.

3.7.2) Relationships between Tables:

"The idea of the relational database model was first proposed by Dr. Edgar F. Codd, who is a research scientist in IBM and tried to find a solution to manage massive amount of data in 1960s. Dr. Edgar thinks that the problem of redundancy and uncertain integrity can be solved by using the power of mathematics" (Hernandez, 2013, pp. 35,36). Therefore, he came up with the idea of relational database, which is still one of the most common models.

Relationships allow database users to understand the structure of a database easily. Relationships provides data consistency within the database by deleting or updating related records which are in different tables. In some cases, there might be more than two tables which contain exactly same data; this demonstrates that the database has a data consistency problem (S. Sumathi, 2007). To promote data consistency it is better to add a new parent table which shares common fields and create a relationship with the child tables. Then there is no ambiguity

about which table has to be chosen when a record needs to be updated/read/deleted

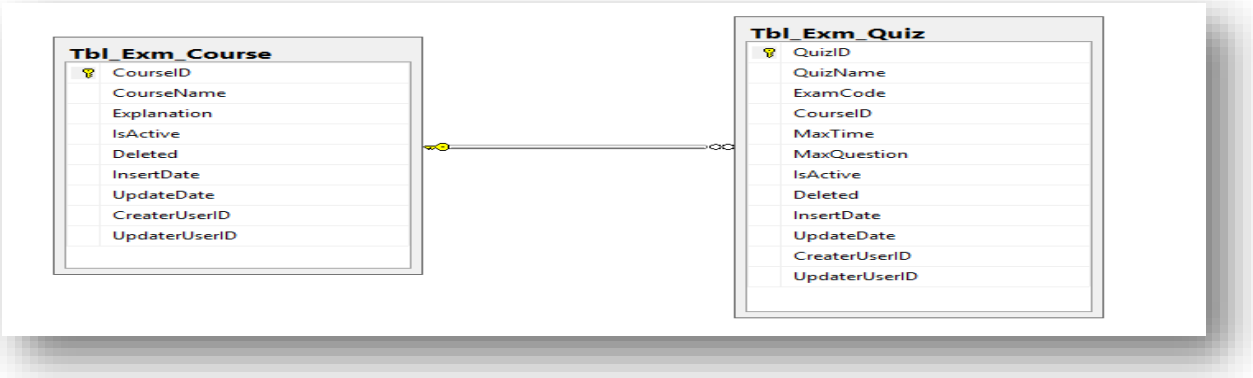


Figure 7: One-to-Many Relation Example

There are three main types of relations which are ‘One-To-One’, ‘One-To-Many’ and ‘Many-To-Many’. There should at least one key field which shows the relationships between tables. Figure 7, above, demonstrates a ‘One-To-Many’ relationships between a course and its quizzes.

In this application, we have used the three types of relationships to create a strong and powerful database. For example, there is a ‘One-To-One’ relationship between ‘Tbl_Main_User’ and ‘Tbl_Main_User_Photo’. Secondly, to be able to have multiple roles for application users, we have a ‘Many-To-Many’ relationship between ‘Tbl_Main_User’ and ‘Tbl_Main_Role’. The last relationship type, which is ‘One-To-Many’ has been showed as an example in ‘Figure 7’, above. Also, Figure 8 illustrates the Entity Relation (ER) Diagram of the exam application without showing the tables which are used for role, menu, and user management.

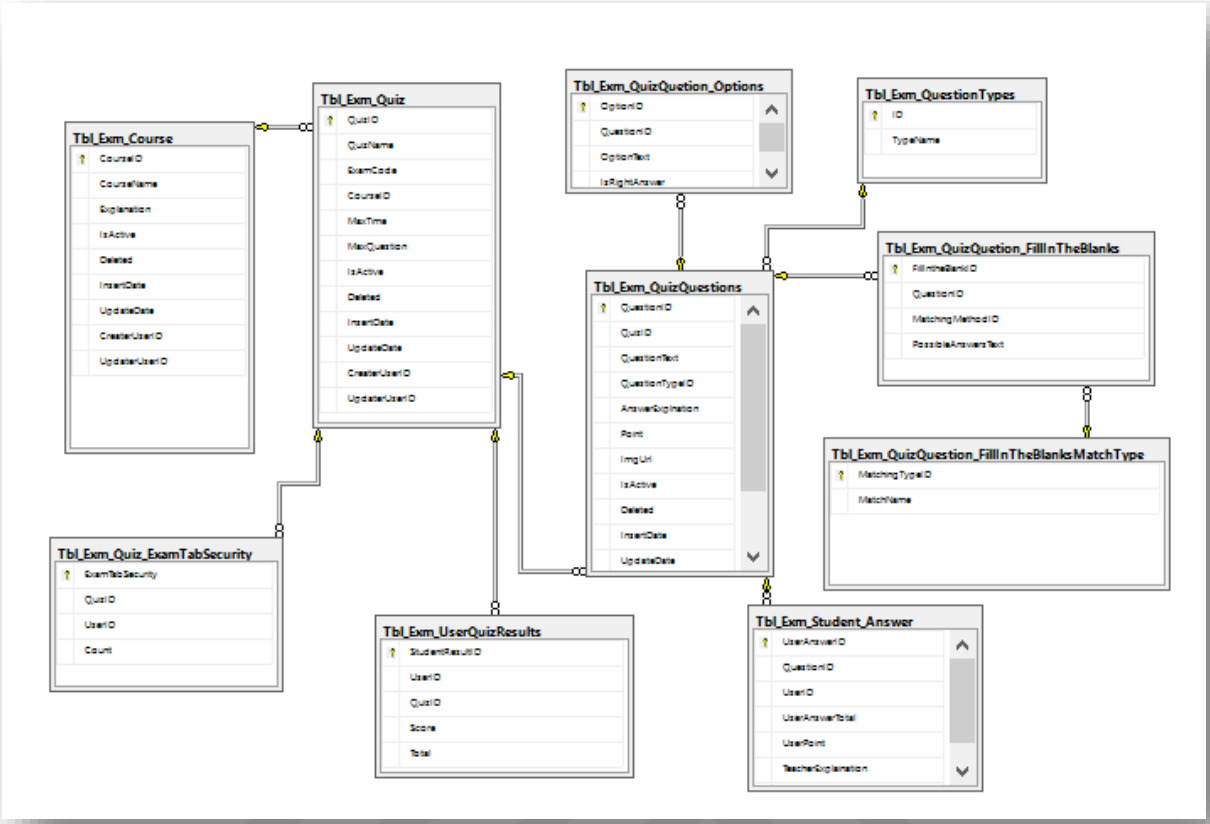


Figure 8: ER Diagram of the Exam Application

3.8) User Interface Design:

A great number of software developers do not want to work on user interface design, because they think that software coding is the key point of a successful application. However, it is known that when a company sells a software product, the code fragment will be the last part customers want to check (Galitz, 2007, p. 32). Even if they want to see the source code, they may not understand whether it is well-structured or not. On the other hand, it is certain that customers will judge the entire application based on the user interface and screen design. Since many developers do not like to design new interfaces, and there are many key indicators which will affect the success of an interface, it is reasonable to start with an admin template which is well-designed and well-structured. Therefore, we decided to use the Atlant template, which is available at http://themeforest.net/ and costs just $24.

3.8.1) Login and Authorisation:

To be authorised to use the application, users need to provide their credentials so the application can identify them. Figure 9 shows the authorisation page of the application. If the user provides incorrect information, he/she will get a warning and be advised to check his/her credentials.

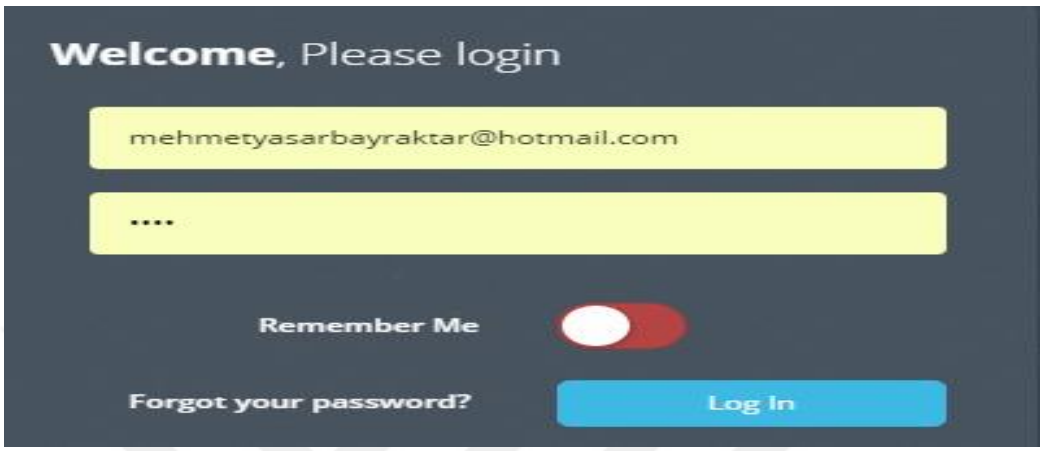


Figure 9: User Authorisation

Additionally, the application's users can check the "Remember Me" button, so after they login, they will be able to gain authorisation from the same computer without providing their credentials again, because their user information will be kept as a cookie. Those users' identities will be kept on the computer as a special encrypted file for security reasons.

3.8.2) Choosing an Active Role:

After authorisation, if a user has more than one role, he/she has to choose the role in which to continue (Figure 10). The application is able to redirect the user to default page by using this active role. However, if there is a referral URL on the link, the user will be sent to the referenced page.

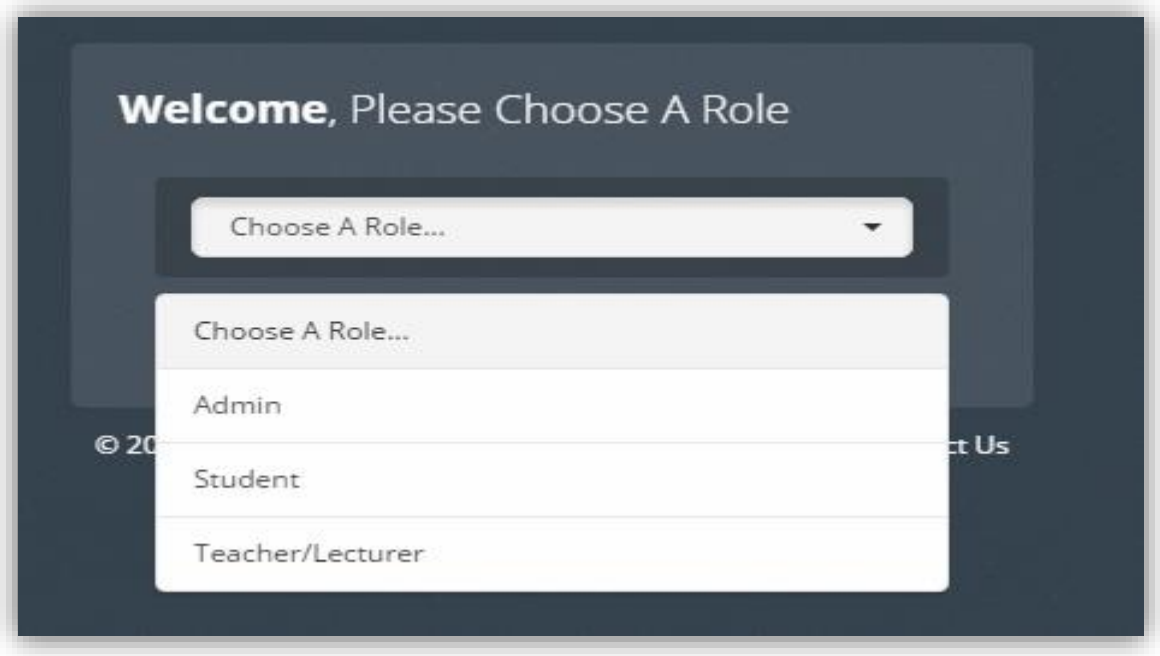


Figure 10: Choosing an Active Role

3.8.3) Dashboard Page:

A dashboard page offers valuable information about the software such as “How many users are active now?”, “How many users have registered in the last month?” and “How many educationalists are avaliable on the system?” (5 ADVANTAGES OF USING DASHBOARDS). A succesful dashboard should also have a graphical interface that helps to catch users’ or customers’ attention. Additionally, a dashboard page should allow developers to place advertisements or give general announcements about the website. Figure 11 demonstrates the dashboard of the Multiple Choice Question and Answer Web Application. This dashboard is flexible and can be accessed on any kind of device or screen thanks to the customisable and fully responsive Atlant template.

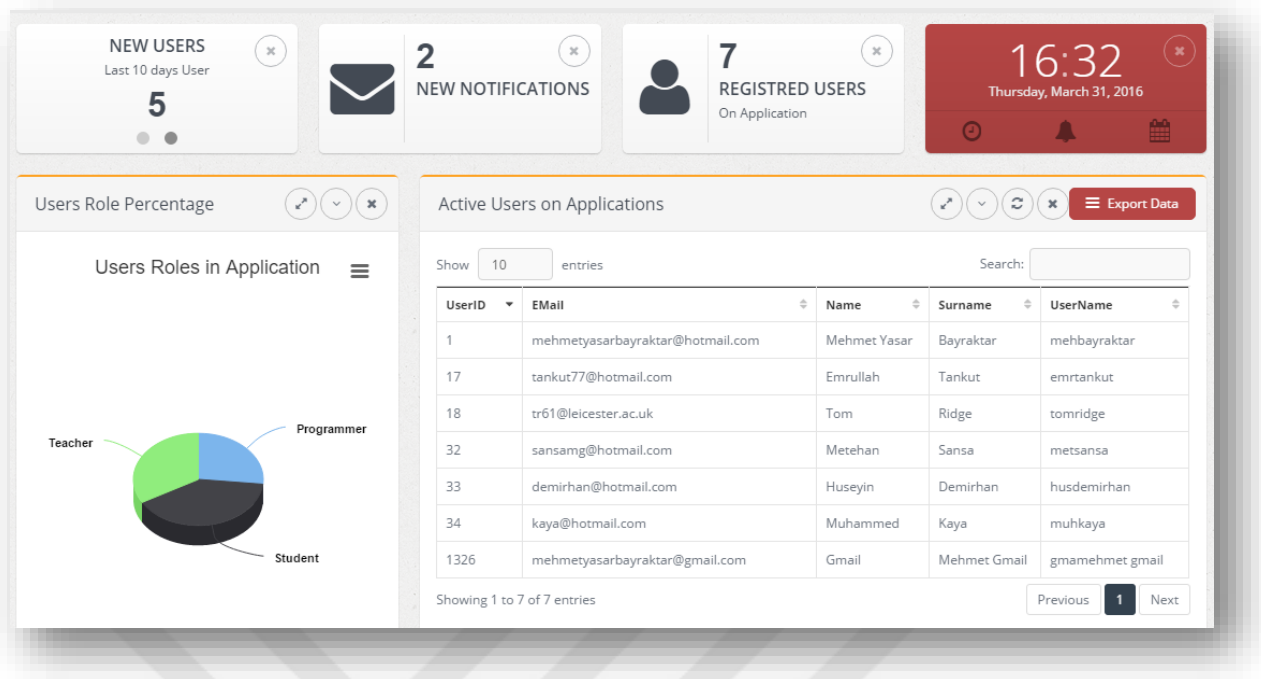


Figure 11: Dashboard for Users

3.8.4) Lecturer Pages:

There are three main pages for lecturers. A lecturer can manage his/her courses using the "Course.aspx" page. This page includes two parts: the top part allows them to enter a new course, and the other part shows the list of courses, allowing them to update or delete those courses. The lecturer is able to add quizzes or exams to the selected course. Moreover, he/she can manage/update the exam security code that allows students to join a quiz. They can also define a quiz time and a maximum question number (Figure 12).

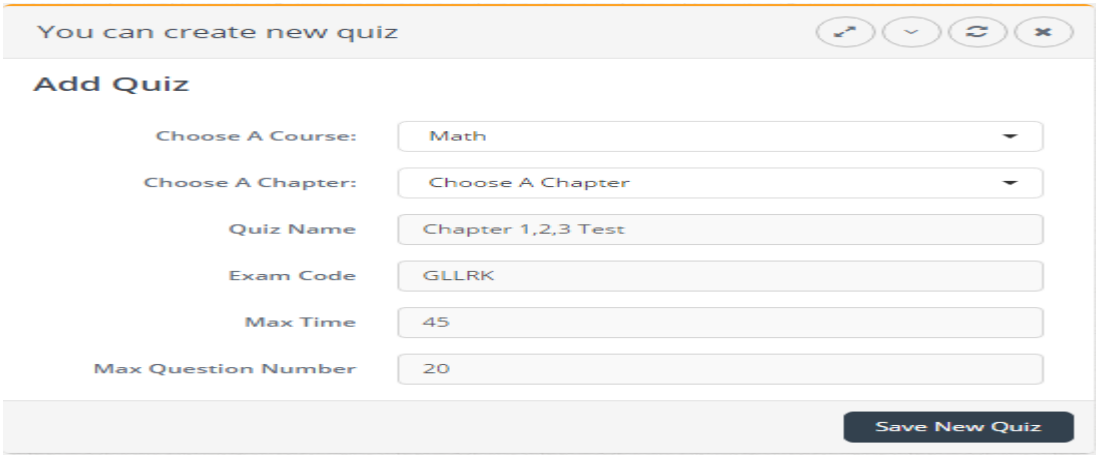


Figure 12: Creating a New Quiz for Students

After creating a quiz, lecturers are able to add new questions. There are four types of question, which are multiple choice single answer, multiple choice multiple answer, fill in the blanks (which has three matching types: "exactly", "contains" and "will be scored by teacher"), and true-false (Figure 13). The lecturer also is able to add an image for each question. After the exam, the lecturer can enter the "Review Exam" page and she/he can review the students' exam papers; if there is the need to change anyone's score, he/she will be able to do so. Additionally, he/she can give individual feedback for some students' questions, so those students can understand what mistakes they have made and why they got a different mark.

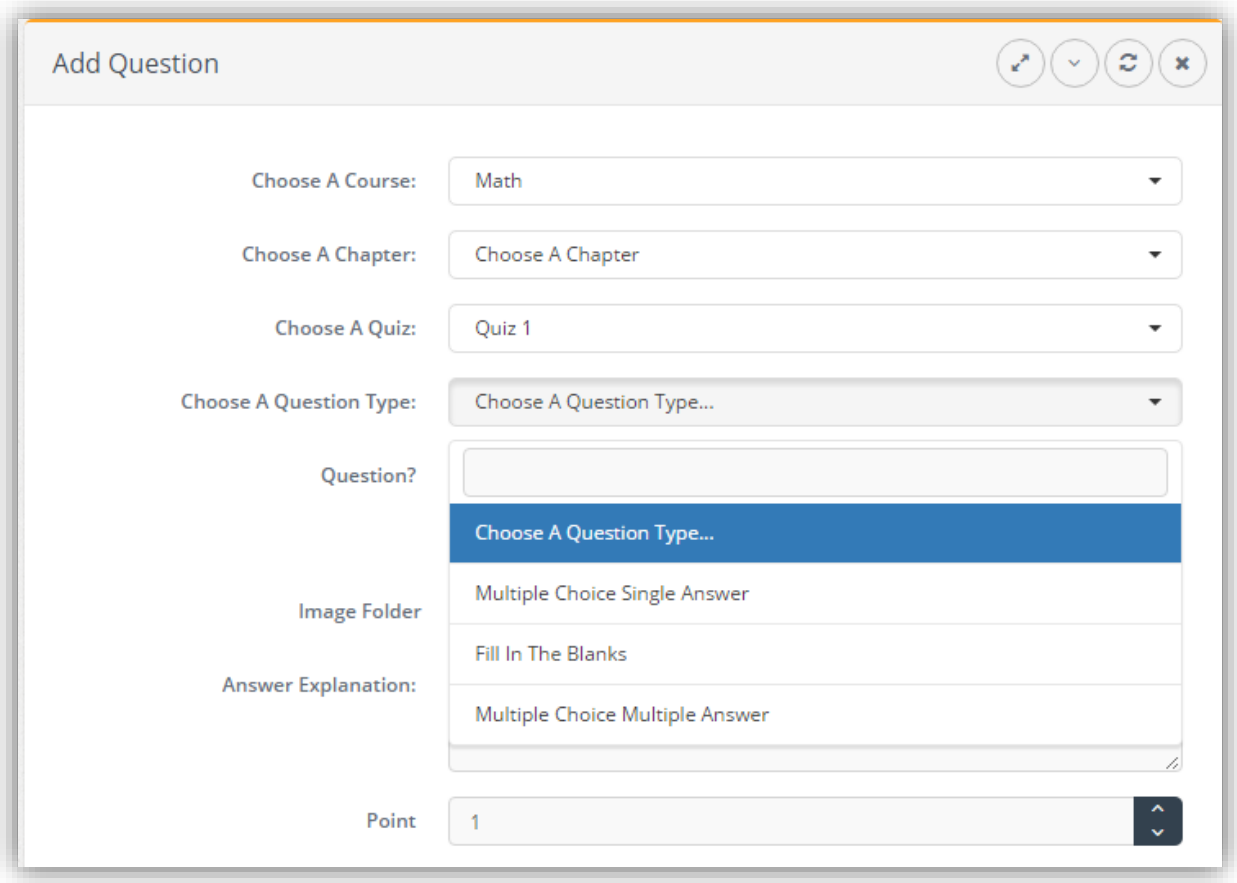


Figure 13: Adding New Question Page

Furthermore, there is a page which gives the lecturer a different insight during the exam. The "Online Students' Score" page enables the lecturer to keep track of the total number of correct and incorrect answers for each question. Therefore, the lecturer can correct students during the exam in order to avoid any misunderstanding. Also, they can use the "Students' Course" page to manage (add or delete) their students.

3.8.5) Students Pages:

Students are able to see active exams and join them by providing ‘Security Code’. Therefore, ‘OnlineActiveTests.aspx’ was created to show the available exams. Figure 14 shows the active exams which can be join by active students.

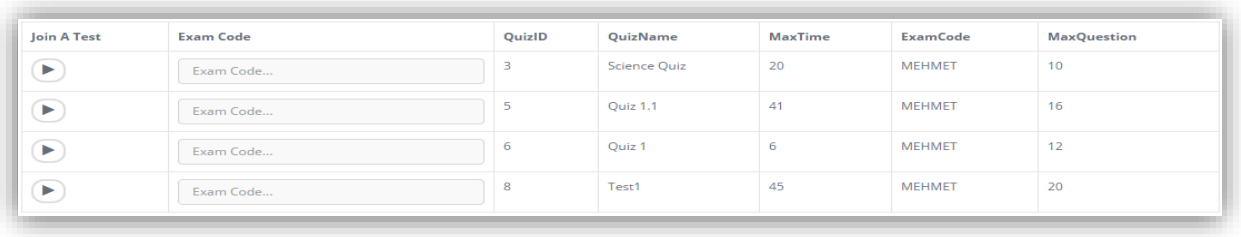


Figure 14: Students Active Exams

3.8.5.1) Active Exam Page:

After joining an exam by providing the ‘Security Code’, students will be redirected to ‘ExamFullPage.aspx’. First of all, this page includes a timer, which is useful for students to check the remaining time. There is also a progress bar, which enables the students to see ‘what percentage of the questions they have done’. On this page, students can see the exam questions and answer them. After they choose their answer, ‘Save Question’ button will be clicked automatically by JavaScript triggers. Therefore, students do not have click ‘Save Question’ button after each question. Moreover, if a student has already answered a question he/she will be able to see his/her past answer. . So, he/she does not need to remember what his/her answer was before. Additionally, we have come up with new security controls which do not allow cheating during the exam. It is known that some students try to open new tabs or documents to find the answers to questions, so the exam tab status is checked during the exam to identify students that try to cheat. Therefore, we are able to warn both the student and the teacher about this activity. Figure 15 shows the warning message which will be seen by students.

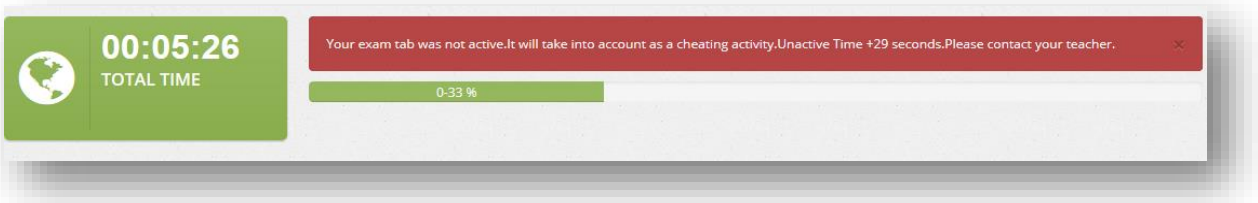


Figure 15: Active Exam Tab Control

To enable students to manage their time more efficiently, when there are five minutes left the timer's background colour will turn red. Additionally, the web application alerts students when they have just 10 seconds remaining by beeping once a second so they are able to finalise their answers before the exam finishes.

3.8.5.2) My Score Page:

'MyScores.aspx' page allows students to see their scores for each exam. They can see their score in a table and save this table to their computer by using the table export functionalities of the template.

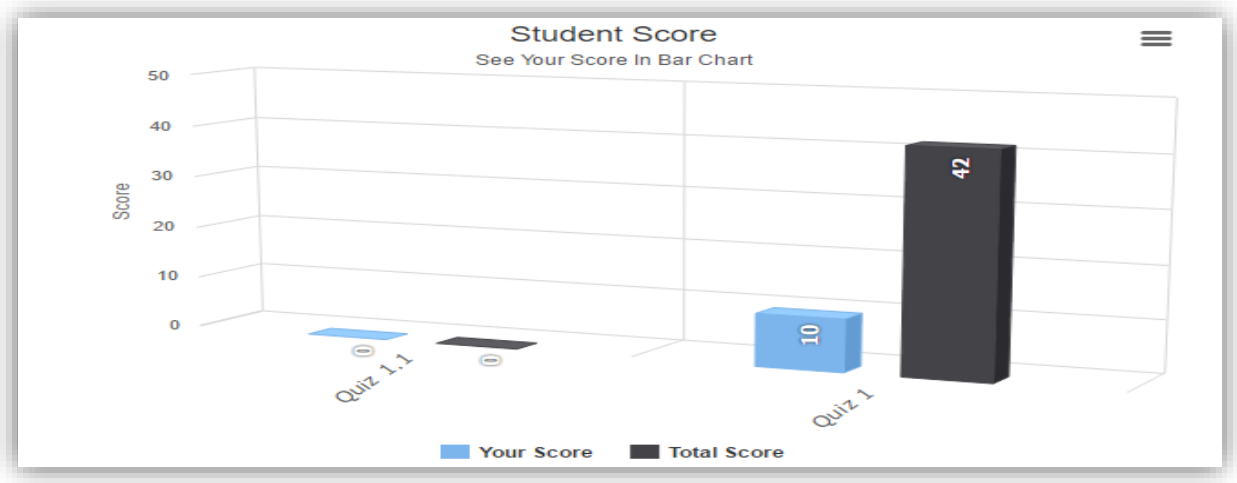


Figure 16: A Student's Score in a 3D Bar Chart

This page also uses a 3D bar chart (Figure 16) to show each student's scores and total score in a graphical interface. This bar chart is implemented using the http://www.highcharts.com/ website. It also allows students to download their 3D bar charts in different file formats (PNG, JPG, PDF or SVG).

3.8.6) Admin Pages:

As Administrators, users are able to work specially on 'Control Panel' menu. This menu includes four submenus, which are 'Menu Management', 'Menu Authorisation', 'Massive User Upload', and 'Role Management'.

3.8.6.1) Menu Management Page:

The menu page permits administrators to add new page links by providing a parent menu, a menu name and a menu link (Figure 17). Additionally, administrators are able to define an icon link that allows the menu to have an icon on its left. The order number field enables the administrator to order menu priorities. After successfully adding a menu, he/she can see all the menus within the parent menus in a list, and he/she will be able to update the menu when necessary.

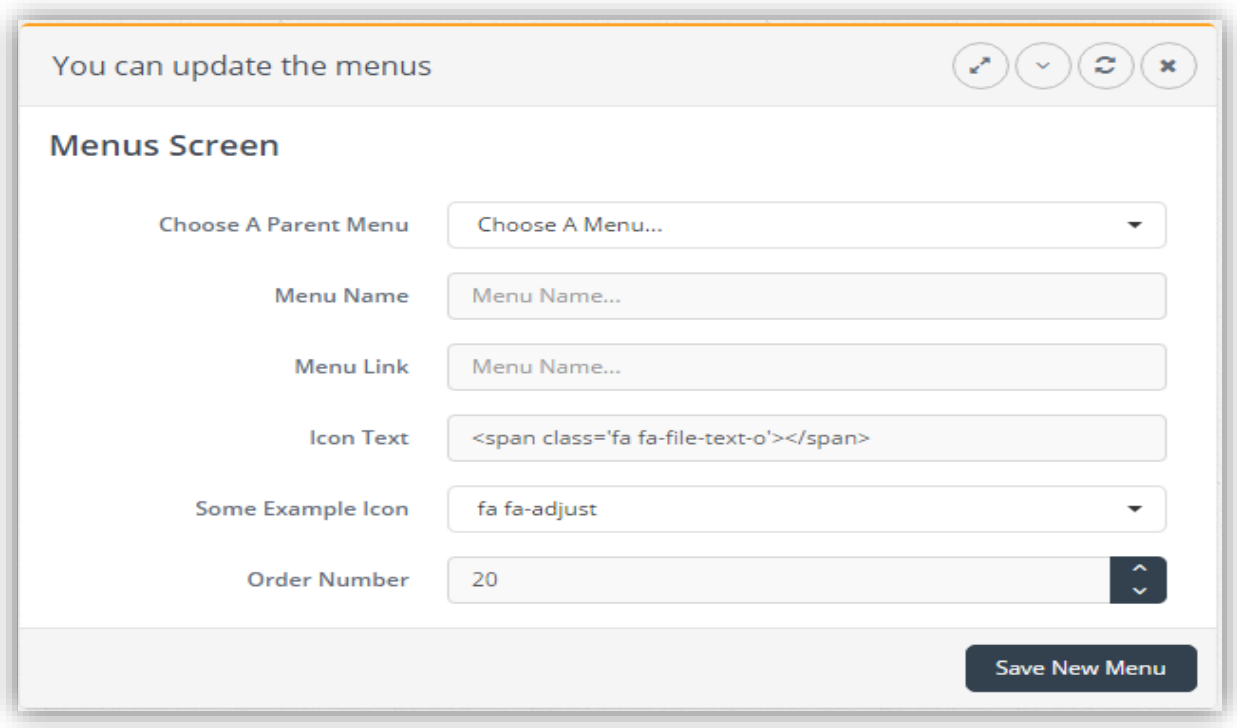


Figure 17: Menu Management

3.8.6.2) Menu Authorisation Page:

After adding the menu, the Administrator has to create rules that determine the types of user who can access the created menu. Hence, the menu authorisation page was created; this allows one to choose a menu and identify who is able to see this menu. Figures 18 demonstrates the interface of the menu authorisation page. So, the left menu of the website will be created dynamically according to the active user's role. However, if the user is an Administrator, all menus will be displayed for them.

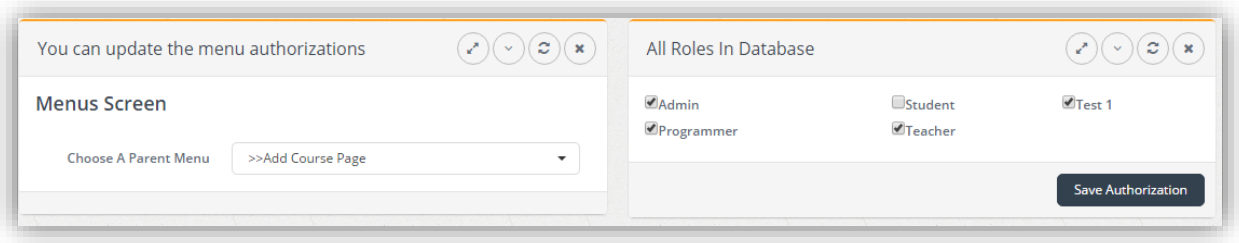


Figure 18: Menu Authorization

3.8.6.3) Massive User Upload Page:

When we are about to sell a new product, even if the product looks very functional, the companies who want to buy this product will have doubts about how to transfer all users which are already in a different database or excel sheet. So we decided that it would be perfect to create a page which allows Administrators to upload thousands of users in minutes, meaning those companies will have saved time and money. Figure 19 shows an example of a massive user upload. The system only accepts xlsx and xls files. However, we can develop the source code to support other types of file, such as csv, docx and xml files.

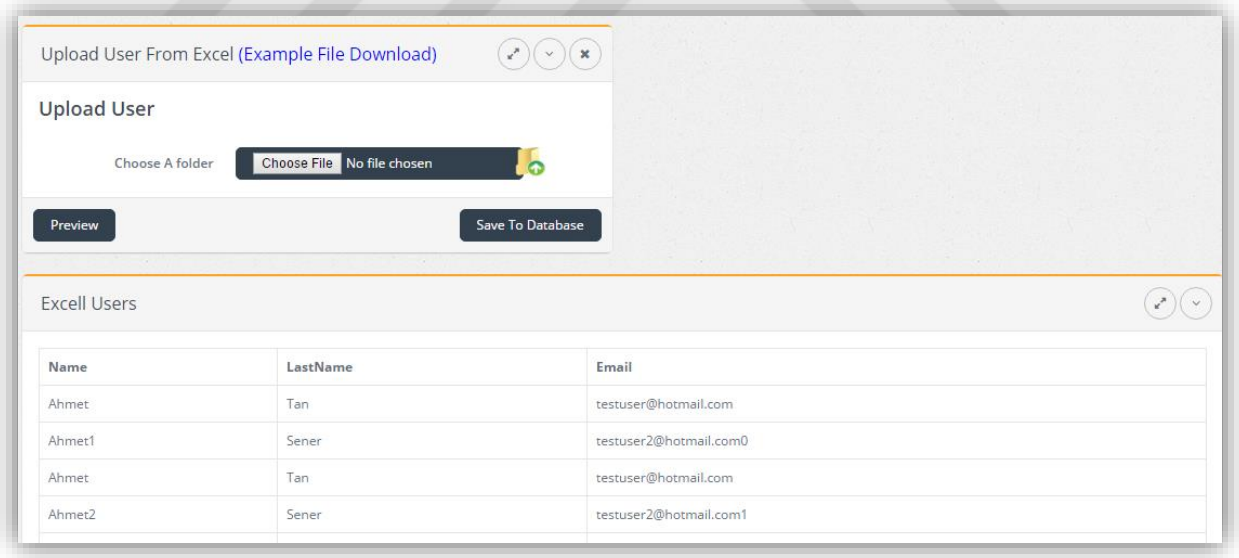


Figure 19: Massive User Upload

3.8.7) User Pages:

User pages are pages that have some functionalities which can be accessed by every type of user.

3.8.7.1) Notifications Page:

The notifications page allows the users to see their notifications, and they can marked those notifications as seen/read. Figure 20 shows the structure of the notification page. Users are able to export their notifications or marked them as read.

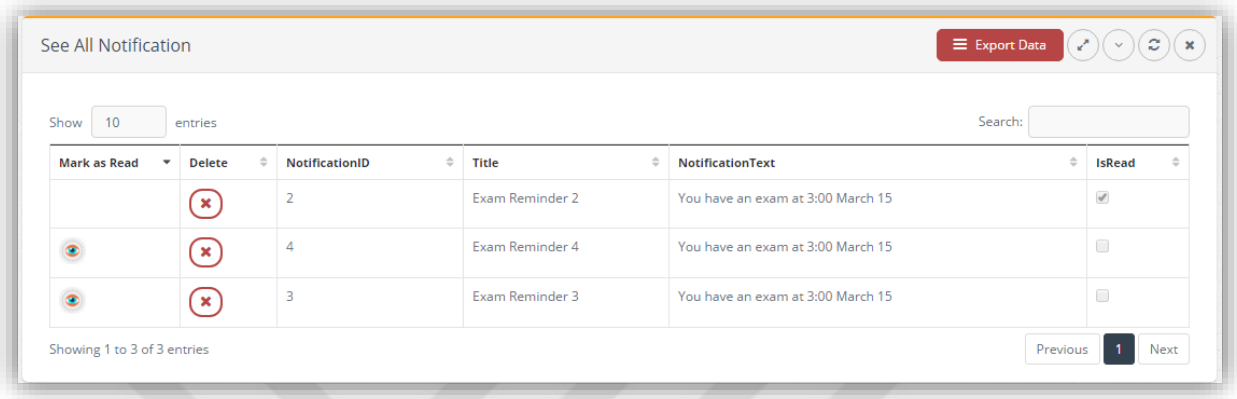


Figure 20: Notifications Page

3.8.7.2) Add Suggestion Page:

Suggestion page allows website users to post their ideas. As an education website, we are showing that we really care about their ideas and we appreciate/approve their opinions. And, we are informing that our customers` ideas are really important and indispensable to us.

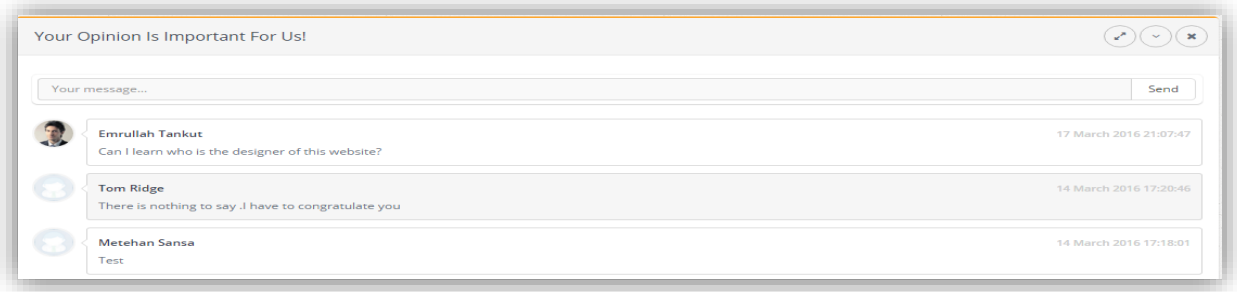


Figure 21: User Suggestion Page

Moreover, the website's users are able to change their information and profile pictures. Also, as an anonymous users, they can register as a student by using registration page.

4) Project Implementation:

4.1) Technical Tools for Implementation:

Microsoft Visual Studio 2015 was used for development process. It supports many for developing new projects, such as Visual Basic, C#, F #, and C++. It is also possible to start with a sample project which is helpful for the junior developers.

The asp.net platform was chosen for the web application, 'C#' language, the structure of which is very closed the structure of 'Java' language, was used for the source code development. However, if some developers need to use 'Visual Basic' language in a different page, asp.net platform is able to execute two languages in the same project by defining the active language at the top of each page.

Since we have chosen 'Microsoft Tools' for the application development, it was reasonable and more efficient to select MSSQL (Microsoft SQL Server) as a database server. MSSQL is a very strong and powerful database server for managing database processes. Moreover, we used additional technology to manage database queries: the 'Entity Framework' (Figure 22). The Entity Framework (EF) is an Object Relational Mapper (ORM) framework which is similar to Java's Hibernate. Thanks to EF, developers do not need to write some basic 'SQL query' code (such as Insert, Update, Delete), which enables them to develop an application very quickly. Figure 22, below, demonstrates a class view of the entity framework. EF also enables us to jump/pass from one table to another tables by using relationships ('One-To-One', 'One-To-Many', and 'Many-To-Many') between them.

Moreover, the IIS (Internet Information Services) Express Server is used to run this application on the local server. ISS enables to offers improved security of the framework and a powerful ability to work with the asp.net platform (Brown, 2003).

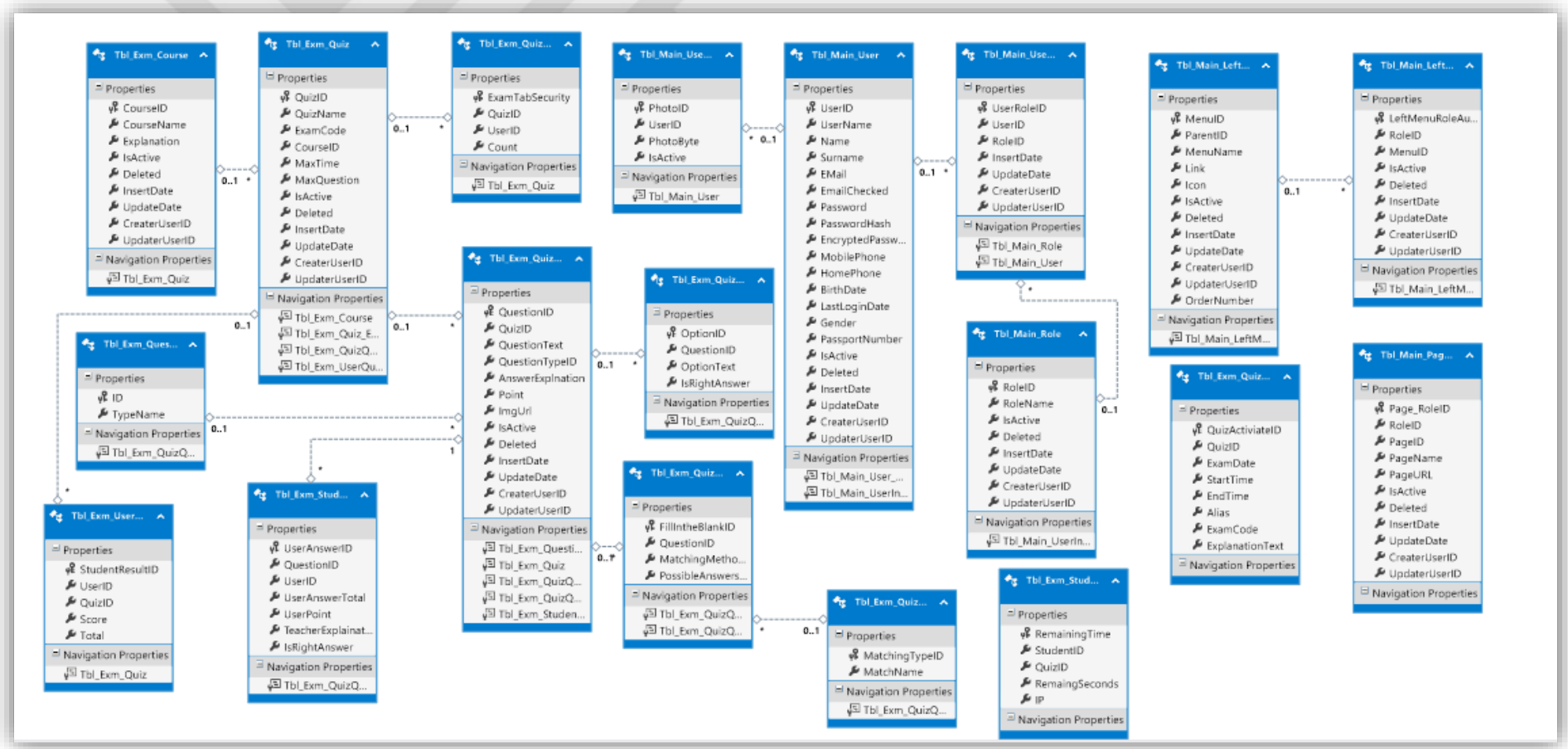


Figure 22: A Part of the Entity Framework Diagram

4.2) Source Code Excerpts:

The infrastructure of the web application should be understandable for the people who wants to improve this application. Therefore, some code excerpts of the source code will be examined on this section. However, before we start to explain the source code, it is necessary to give some information about the code structure. A '3-tier' structure was used on this project. The 3-tier are 'Data Access Layer', 'Business-Logic Layer', and 'Data-Presentation Layer', which enables to create reusable and flexible application and they also facilitate to generate strong security framework by splitting the source code into 3 layers and providing independent security rules for each layer.

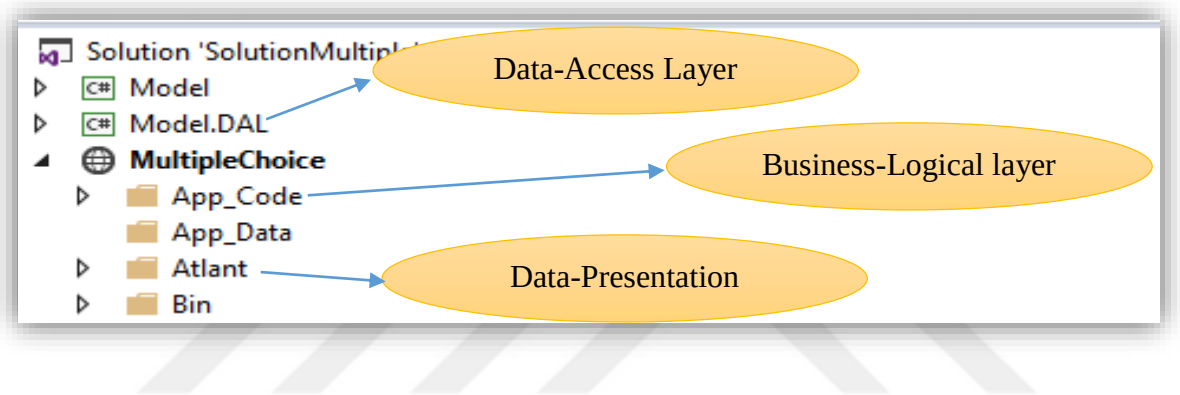
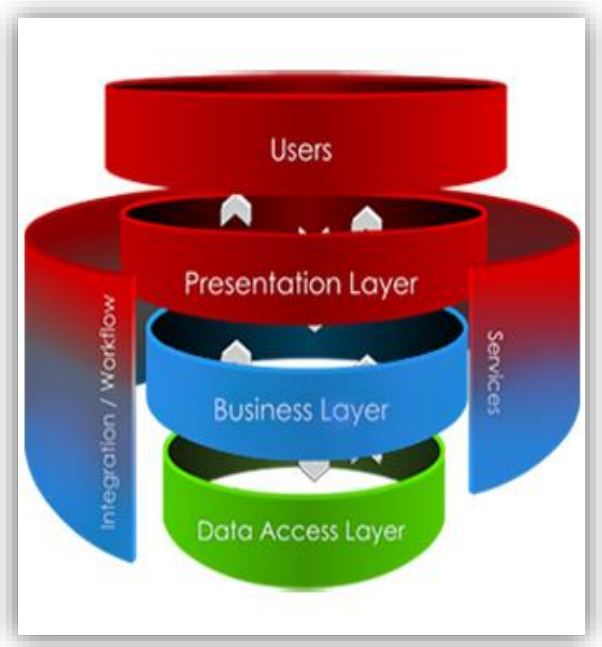


Figure 23: Architecture of the Project (3-Tiers)



The first n-tier architecture was '2-tier' architecture, which is called 'Client and Server' architecture. There are two types of '2-tier' architecture which are 'Fat Client-Thin Server' and 'Thin Client-Fat Server'. In this architecture, the client side is usually responsible for representing the user interface and managing business logic. And, the server side takes care of database management and data-access procedure (B'FAR, 2004, p. 779).However, it has been recognised that separating architecture into 3–tier or

Figure 24: N-tier Architecture Representation (N-TIER DESIGNS)

more tiers is a more efficient and flexible way to manage a project. Therefore, it was decided to use 3-tier architecture for this application.

4.2.1) Login with Cookie:

To be authorised for the application, users have to provide user credentials. Users can be authorized by providing either a ‘username or password’ or an ‘email and password’. Figure 25 shows the authorisation process for the application. For the traceability, and security, all behaviours of the client are saved to the ‘Tbl_Main_UserLog’ table.

```
//TODO:Captcha control
bool IsEmail = inpUserName.Value.Contains('@');
Tbl_Main_User user = GetActiveUser( IsEmail);
if (user == null)
{
    //It should be better create virtaul users
    Tbl_Main_Log_Save.Save_Log_ForAnonymous("Wrong information for log in", inpUserName.Value);
    ltrInfo.Text = Alert.GiveDanger("User Name and Password does not match...");
}
else if (user.Deleted == true)
{
    Tbl_Main_Log_Save.Save_Log(" You are deleted from database!!!You cant login", user);
    ltrInfo.Text = Alert.GiveDanger("You are deleted from database!!!You cant login");

}
else
{// if user has right credentials
    AuthManager.CurrentUser = user;
    user.LastLoginDate = DateTime.Now;
    tbl_Main_UserDb.SaveChanges();
    AuthManager.CurrentRoleID = 1;
    // User log on
    Tbl_Main_Log_Save.Save_Log(" You logged in ..", user);
     SaveCookie();
```

Figure 25: Authorization for the Application

After entering the right credentials, the user will be logged in to the web site. Additionally, some users may want to use the ‘Remember Me’ feature, therefore, they will be able to enter the website by using the same computer the next time without providing any information thanks to the ‘Encrypted User Identity’, which will be stored as a cookie for one year. Also, we combined user identity with ‘IP’ information to make it more secure.

```
private void SaveCookie()
{//To Remember user Information
    if (chkRememberMe.Checked == true)//Remember Me
    {
        Encryption en = new Encryption();
        HttpCookie cokUser = new HttpCookie("cokUser");
        cokUser.Expires = DateTime.Now.AddYears(1);
        string IP = SvrVar.GetIPOfClient();
        string encryotedIP = en.TextEncrypt(IP);
       //kept user identity encyripted
        cokUser["IPUser"] =encryotedIP.Substring(0,5)+ en.TextEncrypt(AuthManager.CurrentUser.UserID.ToString());
        cokUser["IPEnc"] = encryotedIP;
        Response.Cookies.Add(cokUser);//Add Cookie
    }
}
```

Figure 26: Save User Identity and IP as a Cookie

4.2.2) Adding a New Question:

Thanks to the 'Entity framework', it is easy to save a new record by creating an instance of the entity. Figure 27 demonstrates an example of how to add a new record to the table. From the figure, it is easy to see that after creating an instance of 'Tbl_Exm_QuizQuestions', we just need to add this entity to the table ('db.Tbl_Exm_QuizQuestions.Add(question);') and save the changes ('db.SaveChanges();'), which means that send the insert/update query to the database.

After adding 'questions text' and an 'answer explanation', we have to add additional records to other tables. For example, if the question type is 'Multiple Choice Single Answer', we must add all the options, which are given by the lecturer, to the 'Tbl_Exm_QuizQuetion_Options' table. Or, if the question type is 'Fill in the Blanks', we have to add new record to 'Tbl_Exm_QuizQuetion_FillInTheBlanks' table. By splitting these table, we are able to reduce redundancy and duplication.

```
private void SaveNewQuestion(int quizId, int questionTypeId)
{
    Tbl_Exm_QuizQuestions question = new Tbl_Exm_QuizQuestions();

    question.QuizID = quizId;
    question.QuestionTypeID = questionTypeId;
    question.QuestionText = txtQuestion.Text;

    question.ImgUrl = null;
    question.AnswerExplnation = txtAnswerEplanation.Text;
    question.Point = Convert.ToInt32(txtPoint.Text);
    question.ImgUrl = null;
    if (fileUpload.HasFile)
    {
        if (General.IsImage(fileUpload.FileName))
        {
            question.ImgUrl = UploadFolder(fileUpload);
        }
        {
            ltrInfo.Text += Alert.GiveDanger("The file which you are trying to add is not an image file");
        }

    }
    //Additionl fields for detail informaton
    question.UpdateDate = DateTime.Now.Date;
    question.InsertDate = question.UpdateDate;
    question.IsActive = true;
    question.Deleted = false;
    question.CreaterUserID = AuthManager.CurrentUser.UserID;
    question.UpdaterUserID = AuthManager.CurrentUser.UserID;
    db.Tbl_Exm_QuizQuestions.Add(question);
    db.SaveChanges();
```

Figure 27: Authorization for the Application

4.2.3) Uploading a List of Users from an Excel Page:

The aim of this page is to register thousands of users by using Excel sheets, so we have decided to use Excel cells as a column, and an Excel page as a table. To reduce the confusion, an example Excel file, which includes a list of user, is given at the top of 'UploadUser.aspx' page. It is easy to download that file and paste all user information into in the right format and order. After uploading this file to the server, the web-page will check the extension of the file to decide whether it is an Excel file or not using '`IsExcelFile(string fileWay)`' function. Figure 28 illustrates how to connect an Excel file by providing the path of the file and Excel sheet`s name. So, we will be able to read data from the Excel file by using the preview button and we can add those users to the application by clicking 'Save to Database' button.

```
1 reference
private void GetDataFromExcell()
{
    string myExcellConnectionStr = "";
    string fileWay = Server.MapPath("~/") + MainFolder + "/Uploads/" + ViewState["FileName"].ToString()

    if (Path.GetExtension(ViewState["FileName"].ToString()) == ".xls")
    {

        myExcellConnectionStr = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + fileWay + "; Extended
    }

    else if (Path.GetExtension(ViewState["FileName"].ToString()) == ".xlsx")
    {

        myExcellConnectionStr = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + fileWay + "; Extende
    }

    OleDbConnection connectExcel = new OleDbConnection(myExcellConnectionStr);
    if (connectExcel.State == ConnectionState.Closed)
    {
        connectExcel.Open(); // connect to excell
    }

    OleDbDataAdapter excelAdapter = new OleDbDataAdapter("Select Name,LastName, Email FROM [Sheet1$]",
    DataTable dt = new DataTable();

    excelAdapter.Fill(dt);

    connectExcel.Close();

    ViewState["data"] = dt;
```

Figure 28: Reading Data from Excel File

After the clicking the 'Save to Database' button, we will get summary information which shows how many records have been added and how many of them updated.

4.2.4) Online Student Score Page:

This page allows one to see the students' results in a live bar chart during the exam. Thus, lectures can check the total number of correct and incorrect answers for each question, and they will be able to warn the students about the questions which may be being misunderstood by many students. We have used the 'HighCharts App' to draw attractive charts (Figure 30) (Live Random Data) . The 'HighCharts' application draws interesting charts using static data. To create dynamic data, we have used the 'WebMethod' technique which allows one to get information without refreshing/posting the page. This is also available to use as a web service by providing the URL of the page and the function name. Figure 29 demonstrates an example of the 'WebMethod'.

```
[WebMethod]
0 references
public static String[,] GetLiveDataOfStudents(int QuizID, int index)
{
```

Figure 29: Using the Web Method to Get Data from a Server

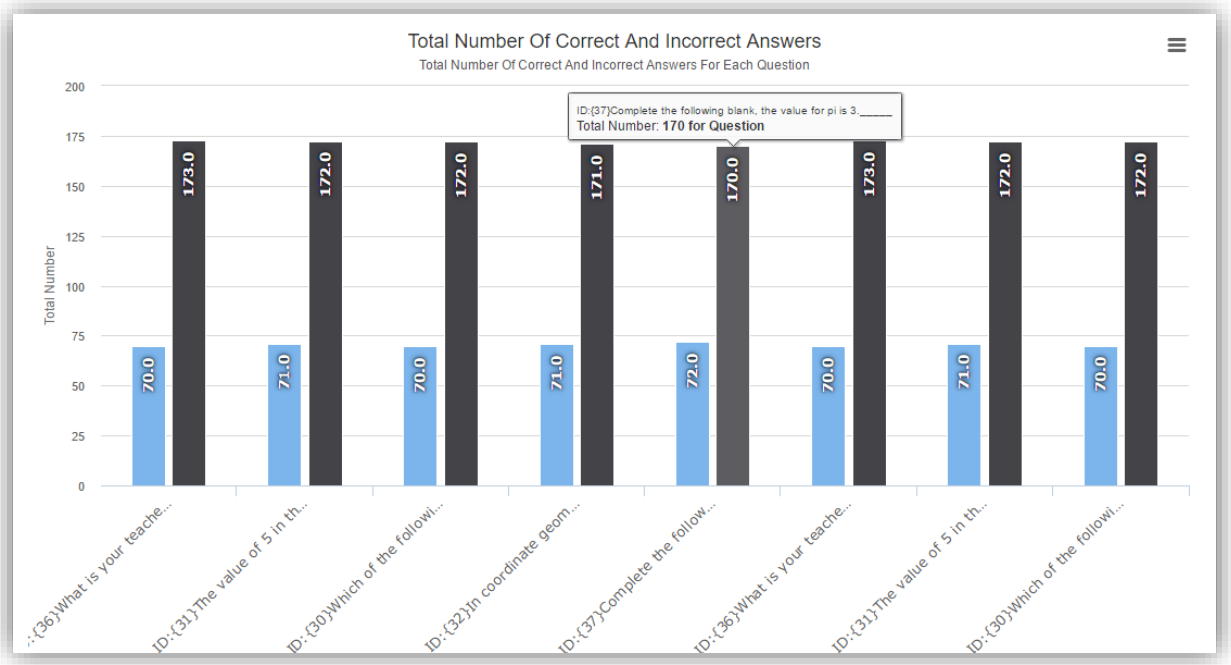


Figure 30: Total Number of Correct and Incorrect Answer

The idea of 'Calculating Correct and Incorrect Answers to Questions during the Exam' was suggested by my supervisor, whose name is 'Dr. Tom Ridge'.

5) Project Testing:

Software testing enables us to analyse the structure/behaviour of the project and measure the software quality. To understand the value of the software testing, it is helpful to refer to USA Government surveys, which demonstrate that $59.5 billion has been lost in business since 2000 as a result of the unqualified (low quality) software projects. (Everett & Raymond McLeod, 2007, p. xi). Also, nowadays, the health sector primarily uses software applications to make decisions; in this context one unqualified or incorrect decision can affect a person's life negatively or even cause a death (Burnstein, 2003, p. 1). Therefore, the quality of software projects has to be measured by a testing phase, which enables the creation of a strong and well-qualified application.

There are two main types of testing which are called 'White-Box (Structural) Testing' and 'Black-Box (Functional) Testing'. 'White-Box Testing' allows us to analyse and test the internal structure of a program while 'Black-Box Testing' enables us to validate and verify outputs against the inputs. Both testing types have some advantages and disadvantages when compared with each other. However, since 'Black-Box Testing' does not require programmer skills and enables one to verify that the requirements of the software have been captured, it was reasonable for us to carry out 'Black-Box Testing'. (Williams, 2006, pp. 43,44)

In this section, we will give some example of 'Black-Box Testing' which allows us to create some test cases using with use case diagrams and requirement phases.

5.1) Authorisation Test:

Test ID:1	**Name: Login Process**
Justification: This process allows to verification of user information which has been provided by the user. It enables the certain of user identity as a cookie in order to remember the user next time.	

Pre-Conditions: The user, who is anonymous to the application, wants to be authorised for the application. He/she might also have been logged out by the application because of a 'Session Time-Out' or 'Unauthorized Process'.	
Actions Steps: Users attempt to gain authorisation by providing their credentials.	
Expected Results: If the users' information is right, they will be redirected to default user page. Otherwise, there will be an appropriate message on a red background colour giving information about the incorrect credentials.	
Actual Results: In both cases, which are incorrect and correct credentials, we found the expected result.	
	Pass/Fail: Pass

5.2) Course Management Test:

Test ID:2	**Name: Course Management**
Justification: In this process, we should be able to add some courses and manage them.	
Pre-Conditions: User's role has to be 'Teacher/Lecturer' to access the 'Course.aspx' page.	
Actions Steps: User tries add a new course, and also to delete or update his/her courses.	
Expected Results: After the authorisation as a 'Teacher/Lecturer', the user should be able to manage his/her courses, if he/she has no authority, the software application should redirect him/her to 'Login.aspx' page.	
Actual Results: We discovered that there was no role authentication control for this page. Therefore, we have added the 'CheckAuthencitaionByID(int ActiveRoleID)' function to check the users` current role.	
	Pass/Fail: Pass

5.3) Quiz Management Test:

Test ID:3	**Name: Quiz/Exam Management**
Justification: As a lecturer, I can create a new quizzes and manage all my quizzes or exams.	
Pre-Conditions: Lecturers/Teachers are able to access the 'Quiz.aspx' page.	
Actions Steps: Teachers/Lecturers will try to add new quizzes and provide certain characteristics of these quizzes.	

Expected Results: After accessing the 'Quiz.aspx', a lecturers/teachers should be able to add new quizzes by providing certain properties of the quiz, if he/she has no authority to do so, the software application should redirect him/her to the'Login.aspx' page.	
Actual Results: We identified the fact that 'ExamCode' field did not updated, so we found the 'SaveNewQuiz()' function and modified it.	
	Pass/Fail: Pass

5.4) Question Management Test:

Test ID:4	**Name: Question Management**
Justification: After adding a new quiz, we should be able to add new questions for this quiz.	
Pre-Conditions: Lecturers/Teachers can add new questions to their exams or quizzes.	
Actions Steps: After choosing a quiz, a lecturer/teacher will add different types of questions to the selected quiz, with or without images. A traffic sign question with an images has been found on the internet that can be used to try this test.	
Expected Results: After filling in the required fields, the question should be added without problems.	
Actual Results: After finding a traffic sign question, we added this question including its options and an image, without failure.	
	Pass/Fail: Pass

5.5) Exam Full Page (Exam Code) Test:

Test ID:5	**Name: Exam Page(Exam Code)**
Justification: Students should be able to join exams using 'Exam Code' provided by the lecturers/teachers.	
Pre-Conditions: To be able to enter a quiz or exam, a student has to know 'Exam Security Code' which is handed out by the lecturer/teacher when they are about to take the quiz.	
Actions Steps: After selecting an active quizzes or exam, students will provide the 'Exam Security Code' which they have been given by their lecturer/teacher.	
Expected Results: If the 'Exam Security Code' is entered incorrectly, students will not be able to enter the exam, otherwise they will be redirected to the'ExamFullPage.aspx' page.	

Actual Results: After entering the correct 'Exam Security Code', the student entered 'ExamFullPage.aspx' page without failure; when he provided an incorrect code, he was not able to join the exam.	
	Pass/Fail: Pass

5.6) Excel Upload Test:

Test ID:6	**Name: Excel Upload**
Justification: Many companies want to transfer their users without spending time on it. In this context, the 'Excel Upload' page will help companies to transfer their users to this application in a short time.	
Pre-Conditions: The Admin role allows users to save lists of users using the 'UploadUser.aspx' page.The Lecturer/Techer and Student roles do not allow users to enter this page. However, in case sof necessity, we can allow the 'Lecturer/Teacher' role to enter this page by updating authorisation rules.	
Actions Steps: After generating an Excel containing user information, Admin will upload the Excel file.	
Expected Results: **:** If the file the user is trying to upload is not Excel file an appropriate error message will be shown on the page; otherwise the Admin will get a success message which explains that the file has been saved successfully.	
Actual Results: 5000 users` data was transferred to the application, however, rendering all these records took a long time and the page gave a 'Time-Out Exception'. Therefore, we disabled the function which enables to rendering of the details of each record. Also, we have increased the 'Time-Out Duration' to an hour to avoid 'Time-Out Exceptions'. After these changes, we were able to add 5000 users in five munities.	
	Pass/Fail: Pass

6) Summary and Conclusion:

In conclusion; an exam application which automates the examination on an online platform has been developed using the C# language. This application enables its users to create new courses and adding some quizzes and exams for those courses. Then, they can assign those exams to their students to learn about the students' outcomes and courses' success.

The application helps its users to generate four types of question for exams (this is explained in detail in section '3.7.4 ').One of the most useful question types is 'Fill in the Blanks' and it has three sub options: 'Exactly', 'Contains', and 'Will be scored by Teacher'. Theses question types allows users to create various types of exam which are no different from paper based exam. Moreover, lecturers are be able to review those exams and give feedback automatically, regardless of location. Also, the 'Exam Tab Status' function will check the students cheating activity during the exam and it sends reports to the lecturers/teachers.

In this project, for students, it is easier to register for online courses and exams like an IELTS (International English Language Testing System), and they are able to take exams from all around the world. Additionally, they are able to see their scores in attractive tables and 3D charts. Also, they can download this information in many different kinds of file format. Moreover, the exam page will help students during exams by warning them about timing and their progress.

Admin pages enable the creation of many authorisation roles without asking the developer thanks to menu management and menu authorisation pages. These pages allows the Admin to manage all application settings without asking the developer company for help, which is helpful to reduce maintenance costs.

Since MULCA is an exam management process, as a future plan, it is reasonable to create a new interface which helps lecturers to create online courses with some materials. Therefore, lecturers/teachers will be able to add/upload their course materials (videos, pdf files and presentation documents) to their courses and allow their students to access those materials regardless of location and time. This will also enable lecturers/teachers to measure how many hours have been spent on each module (or by each student) on average. Moreover, it will be a great business opportunity to introduce this application (by giving a free licence for three months or a year) to education companies so they will be able to see the advantages and disadvantages of online applications, and they might want to become partners/participants in

the application. Additionally, we can support this application with a mobile application, which makes education more enjoyable, interesting and easy to access. Furthermore, it will be logical to create some games (such as matching, spelling, and definition games) for this mobile application, which will help students to study for their modules by playing.

Since an exam application has been developed, it is now reasonable to enhance this project by adding a survey application. It is known that examination and survey applications have almost the same structure and question types. In both cases, the questions can be ‘Multiple Choice or ‘Fill in the Blanks Questions’, and in both cases, the students' answers have to be recorded to the system. Since MULCA already has those functionalities, by simply adding some new features and reports, it will be easy to use this application as a survey application. Therefore, we will have a software product which offers examination and questionnaire functionalities in one compound application.

In conclusion, we have learned that how to develop an online application using requirement phase and how to implement the requirements in the application. Also, to measure and improve the quality of the application, we have applied a testing phase based on ‘Use Case Diagrams’. All of these processes have allowed us to gain a great deal of experience in project management and software development.

7) Source Code References:

7.1) High Chart:

http://www.highcharts.com/

http://www.highcharts.com/demo/column-rotated-labels

http://www.highcharts.com/demo/dynamic-update

http://www.highcharts.com/demo/pie-basic

7.2) Excel Upload:

http://www.aspsnippets.com/Articles/Read-and-Import-Excel-File-into-DataSet-or-DataTable-using-C-and-VBNet-in-ASPNet.aspx

7.3) Image Upload:

http://www.mikesdotnetting.com/article/95/upload-and-crop-images-with-jquery-jcrop-and-asp-net

http://stackoverflow.com/questions/1690337/what-is-the-best-practice-for-storing-a-file-upload-to-a-memorystream-c

7.4) Image Ratio:

http://stackoverflow.com/questions/1940581/c-sharp-image-resizing-to-different-size-while-preserving-aspect-ratio

7.5) Atlant Template:

http://aqvatarius.com/themes/index.php?template=atlant

7.6) Sql JOB:

https://www.mssqltips.com/sqlservertip/1070/simple-script-to-backup-all-sql-server-databases/

7.7) Use Case Diagram:

https://creately.com/app/?tempID=gc7qvpsj1&login_type=demo

https://creately.com/diagram/im3dzrcw/YRDnvPh0TzXKsCqAXTboMr057s%3D

7.8) Captcha:

http://www.massimilianobianchi.info/max/articles/20/A-simple-CAPTCHA-with-ASP-NET.aspx

http://www.codeproject.com/Articles/99148/Simple-CAPTCHA-Create-your-own-in-C

7.9) Encryption-Decryption Hash:

http://www.codeproject.com/Articles/5719/Simple-encrypting-and-decrypting-data-in-C

http://www.codeproject.com/Tips/704372/How-to-Use-Rijndael-ManagedEncryption-with-Csharp

7.10) Email Sending:

http://stackoverflow.com/questions/9201239/send-e-mail-via-smtp-using-c-sharp

https://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient(v=vs.110).aspx

7.11) File Process:

http://stackoverflow.com/questions/12524398/directory-getfiles-how-to-get-only-filename-not-full-path

https://msdn.microsoft.com/en-us/library/07wt70x2(v=vs.110).aspx

Also, there are some other classes which have been added by asp.net platform to create some default setting, such as 'Global.asax', 'Web.config', 'App.config', 'Startup.Auth.cs', 'Startup.cs' etc.

8) References

1. (2007). In M. R. Paolo Pialorsi, *Introducing Microsoft LINQ* (p. 10). Microsoft Press.
2. *5 ADVANTAGES OF USING DASHBOARDS*. (n.d.). Retrieved from http://ontimebi.com/blog/5-advantages-of-using-dashboards/.
3. Abrar, U. (n.d.). AUTHENTICATION. *Profile Based Student Authentication in Online Examination*, 1.
4. *Appappeal*. (2011, April 19). Retrieved from Reasons to Use Web Apps Instead of Desktop Apps: http://www.appappeal.com/articles/10-reasons-to-use-web-apps-instead-of-desktop-apps
5. B'FAR, R. (2004). *MOBILE COMPUTING PRINCIPLES DESIGNING AND DEVELOPING MOBILE APPLICATIONS WITH UML AND XML*. Cambridge University Press.
6. Brown, M. C. (2003). *Microsoft® IIS 6 Delta Guide.* Sams Publishing.
7. Burnstein, I. (2003). *Practical Software Testing.* Springer-Verlag New York, Inc.
8. *CAQ (Create a Quiz/Test Maker)*. (n.d.). Retrieved from https://play.google.com/store/apps/details?id=com.dotpubs.customQuiz&hl=en
9. *Choosing a Web Development Framework*. (2009, December 7). Retrieved from http://code.tutsplus.com: http://code.tutsplus.com/tutorials/15-most-important-considerations-when-choosing-a-web-development-framework--net-8035
10. *Creately*. (2016, 03 22). Retrieved from Creately: https://creately.com
11. Daniel R. Windle, L. R. (2002). *Software Requirements Using the Unified.* Prentice Hall PTR.
12. Everett, G. D., & Raymond McLeod, J. (2007). *Software Testing Across the Entire Software Development Life Cycle.* John Wiley & Sons, Inc.
13. Galitz, W. O. (2007). *The Essential Guide to User Interface Design An Introduction to GUI Design Principles and Techniques.* Wiley Publishing, Inc.
14. Hernandez, M. J. (2013). *Database Design for Mere Mortals A Hands-on Guide to Relational Database Design.* U.S. Corporate and Government Sales.
15. *Live Random Data*. (n.d.). Retrieved from High Charts: http://www.highcharts.com/demo/dynamic-update
16. Mike Gunderloy, J. L. (2006). *Mastering Microsoft SQL Server 2005*. Wiley Publishing,.

17. *N-TIER DESIGNS*. (n.d.). Retrieved from http://www.teamibr.com: http://www.teamibr.com/n-tier-designs.php

18. Patricia Ward, G. D. (2006). *Database Management Systems.* Thomson Learning.

19. *ProProf Quiz Maker*. (n.d.). Retrieved from http://www.proprofs.com/: http://www.proprofs.com/quiz-school/

20. Rynson W.H Lau, Q. L. (2005). Advanced In Web-Based Learning -ICWL-2005.

21. S. Sumathi, S. E. (2007). *Fundamentals of Relational Database Management Systems.* Springer-Verlag Berlin Heidelberg.

22. Schmuller, J. (1999). *Sams Teach Yourself UML in 24 Hours.* Macmillan Computer Publishing.

23. *SCORM Cloud Pricing*. (n.d.). Retrieved from SCORM (Sharable Content Object Reference Model): http://scorm.com/scorm-solved/scorm-cloud-pricing/

24. *SCORM Introduction*. (n.d.). Retrieved from Cloud SCORM: http://cloud.scorm.com/doc/web-services/api.html#Introduction

25. Sheriff, P. D. (2001, November). *Intorudction ASP.NET*. Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/ms973868.aspx

26. Siddharth. (2009, December). *10 criteria for choosing the correct framework.* Retrieved from symfony.com: http://symfony.com/ten-criteria

27. Sikha Bagui, R. E. (2012). *Database Design Using Entity-Relationship Diagrams.* Taylor & Francis Group.

28. Walkinshaw, N. (n.d.). *Requirements Management.*

29. Williams, L. (2006). *Testing Overview and Black-Box Testing Techniques.*

30. Yubas, M. (2013). *How to Get Your Product into Retail.* Broadword Publishing.