##### eExamination

##### Online Examination System

##### A PROJECT REPORT

###### *Submitted by*

###### <Student Name>

***in partial fulfillment for the award of the degree***

***of***

##### MCA (Master of Computer Application)

IN

MUMBAI

**XXX Engineering College**

Mumbai University : Mumbai - 400001

EEXAMINATION – A PROJECT ONLINE EXAMINATION SYSTEM

##### A PROJECT REPORT

###### *Submitted by*

##### <Student Name>

***in partial fulfillment for the award of the degree***

***of***

**Master of Computer Application**

XXX ENGINEERING COLLEGE, MUMBAI

MUMBAI UNIVERSITY:: MUMBAI 400001

MUMBAI UNIVERSITY: MUMBAI 400 001

**BONAFIDE CERTIFICATE**

Certified that this project report **“…………….eExamination…………………..”**

is the bonafide work of “**……………..…..<Atanu Maity>…………”** who carried out the project work under my supervision.

<<Signature of the Head of the Department>> <<Signature of the Supervisor>>

**SIGNATURE SIGNATURE**

<<Name>> <<Name>>

**HEAD OF THE DEPARTMENT SUPERVISOR**

<<Academic Designation>>

<<Department>> <<Department>>

<<Full address of the Dept & College >> <<Full address of the Dept & College >>

**TABLE OF CONTENTS**

[1. Problem Definition and scope of project 7](#_Toc82110068)

[1.1 Purpose 7](#_Toc82110069)

[1.2 Objective 9](#_Toc82110070)

[1.3 Project Scope 10](#_Toc82110071)

[1.3.1 Member and Student Module 10](#_Toc82110072)

[1.3.2 Administration Module 11](#_Toc82110073)

[1.4 Technologies 12](#_Toc82110074)

[1.4.1 Operating Environment 12](#_Toc82110075)

[1.4.2 Deployment Environment 13](#_Toc82110076)

[1.4.3 Development Tools and Technologies 14](#_Toc82110077)

[1.4.4 Development Environment 17](#_Toc82110078)

[2. Overall Description 18](#_Toc82110079)

[2.1 User Characteristics 18](#_Toc82110080)

[2.1.1 Member 18](#_Toc82110081)

[2.1.2 Administrator 18](#_Toc82110082)

[2. 2.Assumptions 18](#_Toc82110083)

[3. System Features 19](#_Toc82110084)

[3. 1.Systern features 19](#_Toc82110085)

[3.1.1 .Description: 19](#_Toc82110086)

[3.1 .2.Constraints 19](#_Toc82110087)

[3.1.4 Action/result 19](#_Toc82110088)

[4. Requirement Analysis 20](#_Toc82110089)

[4.1 Business goals 20](#_Toc82110090)

[4.2 Design goals 21](#_Toc82110091)

[4.3 Feasibility Study 22](#_Toc82110092)

[4.3.1 Economic Feasibility 22](#_Toc82110093)

[4.3.2 Technical Feasibility 22](#_Toc82110094)

[4.3.3 Behavioural Feasibility 22](#_Toc82110095)

[4.4 Business Requirements 23](#_Toc82110096)

[*4.5 User Requirements* 24](#_Toc82110097)

[4.5.1 Student 25](#_Toc82110098)

[4.5.2 Administration 25](#_Toc82110099)

[4.6 Operational requirements 25](#_Toc82110100)

[4.7 System Requirements 26](#_Toc82110101)

[4.8 System Constraints 26](#_Toc82110102)

[4.9 Behavioral Description 27](#_Toc82110103)

[5. Nonfunctional Requirements 28](#_Toc82110104)

[5.1 Performance Requirements 28](#_Toc82110105)

[5.2 Availability Requirements 29](#_Toc82110106)

[5.3 Reliability Requirement 31](#_Toc82110107)

[5.4 Scalability Requirement 32](#_Toc82110108)

[5.5 Security Requirements 36](#_Toc82110109)

[Security Features of .NET Technologies 38](#_Toc82110110)

[5.6 Interoperability 40](#_Toc82110111)

[6. Project Management 41](#_Toc82110112)

[6.1 Development Methodologies 41](#_Toc82110113)

[6.1.1 Waterfall model. 41](#_Toc82110114)

[6.1.2 Spiral model. 41](#_Toc82110115)

[6.1.3 MSF Process Model: 42](#_Toc82110116)

[6.2 Project Development Life Cycle (PDLC) 42](#_Toc82110117)

[6.2.1 Envisioning 42](#_Toc82110118)

[6.2.2 Planning 43](#_Toc82110119)

[6.2.3 Developing and Designing the System 43](#_Toc82110120)

[6.2.4 Stabilizing 43](#_Toc82110121)

[7. Estimation and Planning 44](#_Toc82110122)

[7.1 Planning 44](#_Toc82110123)

[7.1 Application of the MSF model in our project: 44](#_Toc82110124)

[6.2 Estimation 48](#_Toc82110125)

[6.2. 1 Basic COCOMO 48](#_Toc82110126)

[6.2.2 Function Point Estimation 49](#_Toc82110127)

[7. Preliminary Design 51](#_Toc82110128)

[7.1 Use Case 51](#_Toc82110129)

[7.2 Specification of actors 52](#_Toc82110130)

[7.2.1 Student 52](#_Toc82110131)

[7.2.2 Administrator 52](#_Toc82110132)

[7.3 Specification of Use Cases 53](#_Toc82110133)

[7.3.1 Use Case Student Registration 53](#_Toc82110134)

[7.3.2 Use Case. Mange Question Banks 55](#_Toc82110135)

[7.3.3 Use Case. Examination 57](#_Toc82110136)

[7.3.4 Use Case. System Login 59](#_Toc82110137)

[7.4 Process Flow Chart 61](#_Toc82110138)

[7.5 ER Diagram 62](#_Toc82110139)

[7.6 Schema Diagram 63](#_Toc82110140)

[7.7 Data Dictionary 64](#_Toc82110141)

[7.7.1 Table: answer\_master 64](#_Toc82110142)

[7.7.2 Table: exam\_master 65](#_Toc82110143)

[7.7.3. Table: member\_master 66](#_Toc82110144)

[7.7.4 Table: question\_master 67](#_Toc82110145)

[7.7.5. Table: test\_master 68](#_Toc82110146)

[7.7.6 Table: user\_master 69](#_Toc82110147)

[7.8 Data Flow Diagram DFD 70](#_Toc82110148)

[7.8.1 Context Level 70](#_Toc82110149)

[7.8.2 Level 1 (High Level Diagram) 71](#_Toc82110150)

[7.9 Class Diagram 72](#_Toc82110151)

[7.10 Project Planning And Scheduling 73](#_Toc82110152)

[Hardware Interfaces 73](#_Toc82110153)

[7.11 Scheduling 75](#_Toc82110154)

[7.11.1 Gantt chart: 76](#_Toc82110155)

[7.11.2 PERT chart: 77](#_Toc82110156)

[8 Coding and Web Pages 77](#_Toc82110157)

[8.1 eExamination Home Page 78](#_Toc82110158)

[8.2 eExamination – Registration Page 80](#_Toc82110159)

[8.3 eExamination – My Account 84](#_Toc82110160)

[8.4 eExamination – New Examination 87](#_Toc82110161)

[8.5 eExamination – Give Examination 90](#_Toc82110162)

[8.6 eExamination – Results 102](#_Toc82110163)

[8.7 eExamination –View Answers 106](#_Toc82110164)

[8.8 eExamination – Admin Question Bank 112](#_Toc82110165)

[8.9 eExamination – Admin View Results 124](#_Toc82110166)

[8.10 eExamination – Admin Members 126](#_Toc82110167)

[9. Test Plan 128](#_Toc82110168)

[Introduction 128](#_Toc82110169)

[9.1 Test Scope 128](#_Toc82110170)

[9.2 Test Strategy 128](#_Toc82110171)

[9.3 Preconditions 129](#_Toc82110172)

[9.4 Test Priorities 129](#_Toc82110173)

[9.5 Test Techniques 130](#_Toc82110174)

[9.6 Test Organization 130](#_Toc82110175)

[Roles and Responsibilities 130](#_Toc82110176)

[9.7 Deliverables 130](#_Toc82110177)

[9.8 Test Environment 131](#_Toc82110178)

[Hardware and Software 131](#_Toc82110179)

[9.9 Testing Automation Software 131](#_Toc82110180)

[9.10 Application Configuration 131](#_Toc82110181)

[9.11 Test Management 132](#_Toc82110182)

[9.12 Testing Schedules 132](#_Toc82110183)

[9.13 Threats to Testing 132](#_Toc82110184)

[10. Future Enhancement 133](#_Toc82110185)

[11. Bibliography 134](#_Toc82110186)

[11.1 Websites 134](#_Toc82110187)

[11.2 Books 134](#_Toc82110188)

# 1. Problem Definition and scope of project

## 1.1 Purpose

Today internet become reality and usage of internet become very much popular and there is tremendous increase of internet in all over the world for education purpose. The Online Examination System is easy to use, full-featured and flexible Testing, Examination and Assessment web portal. It allows Administrators to set different examination and question banks for registered students and members. The eExamination System provides complete functionality of evaluating and assessing student's performance skills. The eExamination System comprises of Question Bank effectively blended with a whole set of Features. Using different features of the Examination System an administrator can set an exam name, select an examination, and assign questions for examination.

The question bank will form the backbone of the automated process and will play an important role in random generation of unique sets of question papers. The more the number of questions the better randomization is achieved.

Utilize advanced assessment settings, including flexible database driven question pooling, skipping, and randomization and timed response with feature to either conduct the conduct the exam on online. It facilitates in creation of question paper based on a set of parameters. eExamination System provides automated online evaluation of Objective type questions. For online exanimation, the answers are check along with the conduction of the examination and the examinee can be made available the results instantly. Storage of answers, scores, and results in database. Additional to the already available questions in the Question Bank, the Administrator has the ability to add more questions as per the subject.

System should have built in security features to handle different security threat like SQL injection, cross scripting, spamming.

This project report describes the software functional and nonfunctional requirements for release 1.0 of the eExamination system. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here is high priority and committed for release 1.0.

## 1.2 Objective

Online Examination System – eExamination is complete end to end solution to cover all aspects of online examination system.

The basic objective of developing this project is:

* Provides complete web site solution, including member registration, giving tests, storing of results. Complete web based administration.
* The online examination system can automatically add the marks allocated in each question to determine the total mark for the test.
* Using this feature the questions can be selected from the question Bank, and also the appearance of questions in the Random Order.
* Add multiple product to shopping card, automatic calculate the shipping price, calculate tax and calculate total amount.
* Additional to the already available questions in the Question Bank, the Administrator has the ability to add more questions as per the subject.
* The Administrator also has the ability to modify the User-defined Questions and not the already Available Questions in the Question Bank.
* Complete web based system no installation required to run the application in client system.
* Student result must save and can be retrieving later on.
* Question must provide answer type like multiple right answer selection and one right answer selection. Test screen must provide options accordingly.
* System must able to evaluate the examination and display the result instantly, student must view the correct answer after complete the test.
* System has powerful logical access management in place, each user must be identified by login id and strict password policy is applied to secure the system
* Generate various reports for administration like, Total students/members registered for examination, no of available examination, test given by the student.

## 1.3 Project Scope

The eExamination System will permit to take and give online examination and maintaining master information and generating various reports of test. The main users of the project are Student or Member and system Administrator.

From an end-user perspective, the eExamination System Project consists of two functional elements: enhanced Member modules for registration, giving examination, manage profile. And Administration module for Mange members, examination, question bank and view the reports.

### 1.3.1 Member and Student Module

An enhanced interface for student to registration, edit profile, login, select examination, and give examination, view results, view answers Following modules pages are available for members.

**CM-1: Home –** It is the default page for the site. All links are available in this page.

**CM-2: Login –** Student need to login to view his account information and buy the product. If student forgets his password he can get back old password from Forget password link. New student can register for this site by click on register link.

**CM-3: Register** – New Student need to register to give examination. Type all the details of the student like email id, name, address, contact details and submit. System validate for email id, it should be unique. In login process student need to type email id and password. Student should type valid email id because if he forget password old password will be sent to this email address.

**CM-4: My Account** – It shows the details of currently logged student details, previously saved test results can be viewed here and other links like Edit Profile, Logout, and Change Password.

**CM-5: Edit Profile** – Student can edit his profile like personal details, address, contact no, display name however student can not edit email address once register.

**CM-6: Change Password** – Student can change his password from this link. Student must type his old password to change the password with new password.

**CM-7: Logout** – By clicking this link user logged out from this site all user session reset to default value.

**CM-8: Examination** – Student can view the available examination by selecting any examination from list, system display details of test like, examination name, description, total available question, number questions for test, total time, passing marks.

**CM-9: Test** – It display the test page, system randomly select questions from question bank and display one by one to the student. Student navigate the questions and select the right answer from available options, after complete the examination system display the results instantly and save the test details in system for future reference.

**CM-10: View Answers** – After completing the examination student can view the answer for each question.

### 1.3.2 Administration Module

Administration module used to enter and edit examination details, question banks, maintain the members and view the reports; The Web-based administration module will include the following features:

**AM-1: Login** – Login page for the administrator. All admin users are identified by the user name, password. Admin user can create new examination maintain question banks.

**AM-2: Members** – It show list of members registered in the site.

**AM-3: Examination** – Add edit the examination details.

**AM-4: Question Bank** – Maintain the question bank for selected examination, Administrator can add, edit the question and answers.

**AM-5: Results** – Display list of test results.

**AM-6: Logout** – By clicking this link admin user logged out from this site all user session reset to default value.

Both of these areas of functionality will be delivered as the first version of the eExamination Application for online examination system is released. Functionality is described in more detail later in this document.

## 1.4 Technologies

### 1.4.1 Operating Environment

OE-1: The eExamination web application will operate with the following Web Browsers: Microsoft Internet Explorer and Google Chrome

OE-2: The eExamination web application shall operate on a server running the latest versions of IIS (Internet Information Server).

OE-3: The eExamination web application shall permit user access from Internet connection

OE-4: Operating System: Windows 7, 10

OE-5: Software requirements: SQL Server 2014, .net framework 4.5.

OE-6: Languages used are asp.net using c# and scripting is done using JavaScript.

OE-7: Hardware Requirements: 4(minimum)/8(recommended) GB RAM

OE-8: Hard disc- nGB depending upon the requirement to store data minimum of 25GB.

### 

### 1.4.2 Deployment Environment

DE-1: Database Server

OS – Win 2019 Enterprise Server

SQL Server 2014

HDD – Min 10 GB, Recommended 25 GB

RAM – Min 4 GB, Recommended 8 GB

Processor - Pentium Dual Xenon Processor

DE-2: Application Server

OS – Win 2019 Enterprise Server

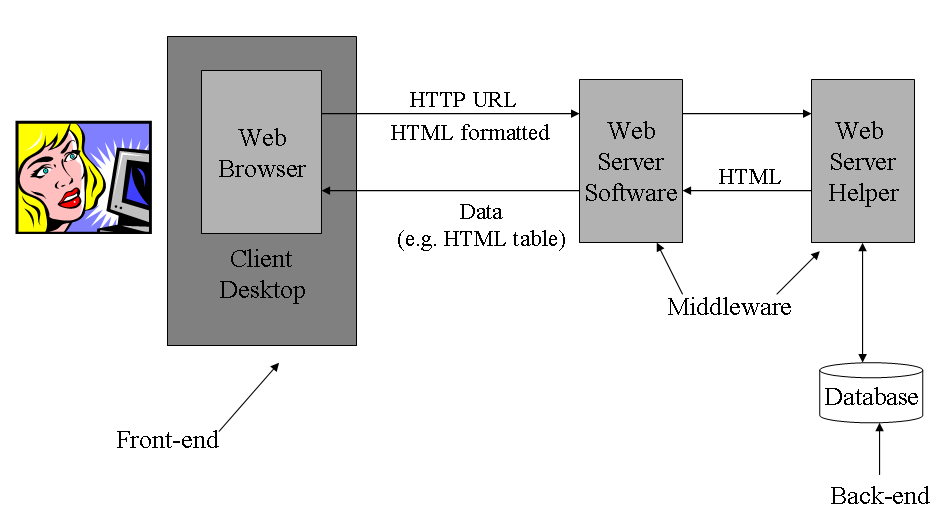
IIS – Internet Information Server

HDD – Min 5 GB, Recommended 10 GB

RAM – Min 4GB, Recommended 8 GB

Processor - Pentium Dual Xenon Processor

DE-3: The eExamination web application will operate with the following Web Browsers: Microsoft Internet Explorer and Google Chrome.



[eExamination System Architecture]

### 1.4.3 Development Tools and Technologies

**DT-1: ASP.Net**

As we need to develop Web Application for eExamination Application for Online Examination System. We will use ASP.Net as it is the new Microsoft technology to develop any application which support and integrate other server product for the internet. By ASP.Net we can develop in web application by .NET framework and manage environment with any .NET support language like VB.Net and C#.

**DT-2: C#**

.NET is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Web-based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability, and manageability for the global, Web-enabled enterprise. The Windows Server System integrated software products are built for interoperability using open Web standards such as XML and SOAP.

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead of targeting a particular Hardware/OS combination. The components that make up the .NET platform are collectively called the .NET Framework.

The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like, allocation of memory for the storage of data and instructions, granting and denying permissions to the application, managing execution of the application and reallocation of memory for resources that are not needed.

The .NET Framework is designed for cross-language compatibility. Cross-language compatibility means, an application written in Visual Basic .NET may reference a DLL file written in C# (C-Sharp). A Visual Basic .NET class might be derived from a C# class or vice versa.

The .NET Framework consists of two main components:

Common Language Runtime (CLR)

Class Libraries

The advantage of C# includes

Powerful Windows-based Applications

Building Web-based Applications

Simplified Deployment

* Powerful, Flexible, Simplified Data Access
* Improved Coding
* Direct Access to the Platform
* Full Object-Oriented Constructs
* XML Web Services
* COM Interoperability

**DT-3: SQL Server 2014**

EExamination Application uses SQL Server 2014 as storing the data. Microsoft SQL Server 2014 as our database and it has so many features which is ideal for our dot net based application. Features Includes

* + Support for Multiple Platforms
  + Integration with Windows Back office servers
  + Integration with Microsoft .NET Enterprise Servers
  + Scalability
  + Replication
  + Centralized Management
  + Reliability
  + Automating Tasks

### 1.4.4 Development Environment

**DE-1: 1.Visual Studio 2019 IDE**

Most advanced integrated development environment for developing .NET application. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It allows plug-ins to be added that enhance the functionality at almost every level - including adding support for source control systems (like Subversion and Visual SourceSafe) to adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports languages by means of language services, which allow any programming language to be supported (to varying degrees) by the code editor and debugger, provided a language-specific service has been authored. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), and C# (via Visual C#). Support for other languages such as Chrome, F#, Python, and Ruby among others has been made available via language services which are to be installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Language-specific versions of Visual Studio also exist which provide more limited language services to the user. These individual packages are called Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

# 2. Overall Description

## 2.1 User Characteristics

### 2.1.1 Member

Able to register, login, edit personal information, results, select the examination, give test, view results and view answers.

### 2.1.2 Administrator

Administrator will have all the access rights. Administrator can create new examination, edit examination details, maintain the question bank, view the member and test results.

## 2. 2.Assumptions

1) System User and Administrator communicate with each other via emails.

# 3. System Features

## 3. 1.Systern features

### 3.1.1 .Description:

A web based eExamination application which will use to give and take examination over the internet. Proposed eExamination application system is end to end solution for online examination process includes member registration, selection of examination, give examination, view results, configuring examinations, maintain question bank, and generation of reports for the management.

### 3.1 .2.Constraints

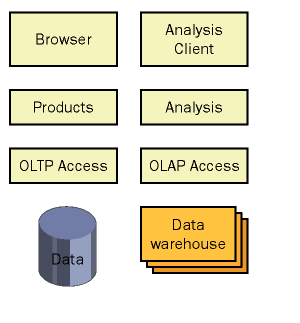
Linking and integration with academic and institution system. Upload questions into system in different formats. Connecting to third-party OLAP applications for generating reports. Etc.

### 3.1.4 Action/result

|  |  |
| --- | --- |
| **Action** | **Result** |
| Member registers himself | After validation, naive users becomes registered Users. |
| User/Admin Logins via Login page | According to the access rights redirected to the accordance page. |
| Manage Question Bank (Admin) | Redirected to question page for add, edit question and answer. |
| Start new examination (Member) | Redirected to Start Examination page. |
| View Answers (Member) | Redirected to view answer page from result page |
| View Test Results (Admin) | Redirect to result page |

# 4. Requirement Analysis

The requirement analysis outlines the approach the development team will take to meet the goals of the project and provides the basis for proceeding to the planning phase. After identifying the business problem and defining the vision and scope, the team creates the solution concept that explains in general terms how the team intends to meet the requirements of the project.



[Solution concept of eExamination]

For a project to be successful, it is essential that we correctly identify the goals of the project. Project goals can be categorized as follows:

* Business goals
* Design goals

## 4.1 Business goals

Business goals represent what the organization wants to achieve with the solution. Business goals form the basis for determining the success criteria of the solution. The purpose of defining business goals is to clearly articulate the objectives for the project and to ensure that your solution supports those business requirements. The team needs to determine the best method for identifying the goals and agreeing on them.

Throughout the life of the project, the team makes tradeoffs among resources, schedule, and features. It is important that business goals are prioritized in a way that will allow the team to have a clear understanding about which ones the customer believes are most important, in case some of the goals cannot be achieved.

For this eExamination project, business goals might include the following:

* Expand the institute’s s geographic reach beyond the current physical location.
* Shorten the time to conduct examination and result process by using more efficient online sites.
* Integrate all students worldwide by using a online site process, and shorten the overall examination process.
* More transparent process of taking the examination.
* No additional and specific system requirement to operate the online examination site.
* It must be secure.
* Easy to use and no addition knowledge require to operate the system.
* Reliable and accurate and their should not be any manual intervention in examination process.
* No requirement of examiner and examination center.

## 4.2 Design goals

Design goals are similar to business goals in many ways. The difference is that design goals focus more on the attributes of the solution and less on what the solution will accomplish for the business. Design goals address not only what the team wants to accomplish but also what the team is not trying to accomplish with the solution. As with business goals, you need to prioritize design goals so that the team knows which goals must be accomplished, in case the project cannot achieve all of them.

Consider the case of this e-commerce Web site. Some of the design goals for the online shopping cart might include:

* Improve the user experience by reducing page-download wait times to 5 seconds or less.
* Limit dependency on connectivity with the server.
* Reduce the time and level of effort required for a user to complete the online registration.
* The service and all supporting applications must be localized for users worldwide.
* The service must have an availability of 99.99 percent.
* The service cannot lose data.
* The service must permit access only by authorized users.

## 4.3 Feasibility Study

### 4.3.1 Economic Feasibility

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly knows as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management, because very often the top management does

not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases. In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving lot of time.

### 4.3.2 Technical Feasibility

Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

### 4.3.3 Behavioural Feasibility

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

## 4.4 Business Requirements

The following preliminary lists are based on initial interviews and study of existing manual system,

The business goal for the application is to support an increase the security, speed, accuracy and complete automation of existing manual or semi automatic Examination process. Business requirements are discussed in the Scope section, with the following additional detail:

* Improve the examination search facility and student should get all the information in a second.
* System should able to save the results and test details.
* Admin user must able to monitor all the activities.
* The Administrator should be able to enter or update examination details and question bank from web interface only
* System Administrator must able to control the access rights by each user as per requirement.
* The application should support the capability to use multi user environment.
* The Admin user should able to generate all type of reports as and when required by the management.
* Can be used anywhere any time as it is a web based application (user location doesn’t matter).
* No restriction that examiner has to be present when the candidate takes the test.
* A Student Sign-in page must allow registered student to sign in with their e-mail addresses and passwords. New student must be able to register from this page.
* A Sign-in Information page must allow student to input or change their e-mail addresses, passwords, and other personal information.
* The following question types are essential:
  + **Multiple Choice**  
    **This question type allows the user to select ONE correct answer**  
    - Options can be randomised  
    - More than one option can be correct but the user can only select ONE option  
    - Score can be set per option  
    - No limitation on number of options  
    - If there are a number of Options, these can be presented in a column(s).
  + **Multiple Response**   
    **This question type allows the user to select more than ONE correct answer**  
    - Options can be randomised.  
    - Score can be set per option.  
    - No limitation on number of options.  
    - If there are a number of Options, these can be presented in a column(s).  
    - The number of responses students enter can be limited.
* Students can access tests they have completed.
* Question paper must be randomized order of question must be randomized no two student must not get same question in same order.

The system users want to improve their current ability to analyze student. In particular, they want to focus on identifying their best student. To enable them to accomplish this goal, they want to extract meaningful data that easily answers the following questions:

* What are the early warning signs of problems?
* Who is my best student?
* What are my student’s issues as groups?
* What type of examination are my student wanted?

### *4.5 User Requirements*

User requirements are categorized by user type.

### 4.5.1 Student

* Able to register in the site
* Able to edit personal information, contact details.
* Able to select the examination and give examination online.
* Able to view result online after completing the examination.
* Able to view right and wrong answers after completing the examination.
* Able to view previous examination and test details in My Account page.

### 4.5.2 Administration

* Administrator must able to add new examination details.
* Must able to maintain the question bank.
* Generate Reports as per requirement.

## 4.6 Operational requirements

The following requirements provide a high-level view of how the system will run:

* Processor usage should not exceed 80 percent during concurrent uses.
* Backups will occur incrementally throughout the day.
* A full weekly backup is required to WORM drives.
* Ensure that information is easy to access either, and meaningful for the system users and the company.
* Minimize the technical knowledge that system users and student need to access the data, generate ad hoc queries, search and view examination and test information.
* Any change to information must be reflected immediately, and the changes must be propagated to the search engine so that system users that perform searches see this new information.
* The application should work with the existing communications and networking infrastructure.
* The application should deploy with a minimum of additional operational processes, manual or otherwise.

## 4.7 System Requirements

These are additional constraints from a system perspective:

* The administrator must be able to monitor everything from the IT department.
* The information must be accessible by everyone in the organisation in intranet and in internet for the students and administrator as per the rights specify.
* All data especially question bank must be up to date.

## 4.8 System Constraints

Constraints indicate the parameters to which the final business solution must adhere. They are aspects of the business environment that cannot or will not be changed. Often, these constraints become design goals for the application. If constraints are not identified properly, the project team might design a product that cannot be deployed within the business.

Examples of possible constraints that you should document include:

* Budget limitations
* Characteristics of earlier supporting systems
* Network system architecture
* Security requirements
* Operating systems
* Planned upgrades to technologies
* Network bandwidth limitations
* Maintenance and support agreements and structures
* Knowledge level of development or support staff
* Learning limitations of users

## 4.9 Behavioral Description

The user of eExamination system registers himself by simply accessing the system’s home page and fills the simple form. When the student register, he or she can able give examination, view results and find right and wrong answer. Student can view the *my account* page to see the previous test details.

Administrator is the user of the system with administrative rights. Administrator manages Examination. Maintain the question bank and monitor the member and tests he also view and generate various reports.

# 5. Nonfunctional Requirements

## 5.1 Performance Requirements

An application’s performance is defined by metrics such as transaction throughput and resource utilization. A user might define an application’s performance in terms of its response time. No more than 5-percent degradation in average query response is allowed while all concurrent users are using the system. Processor utilization should not exceed 80 percent during all concurrent users are using the system.

We must define performance requirements before the team proceeds to the developing phase. To define a good performance requirement, we must identify project constraints, determine services that the application will perform, and specify the load on the application.

**PR-1: Identifying constraints -** Constraints in the project include budget, schedule, infrastructure, and the choice of development tools or technologies. For example, we might need to deploy this eExamination application by a specific date. We might also need to use a specific development tool because the team has expertise in that tool only. We might not be able to design and develop applications that are processor intensive because the client computers do not have adequate hardware. We need to design an application so that it meets its performance goals within the limitations of the constraints. Instead of changing some aspects of a project to improve performance, you can modify aspects of the project that are not constrained to determine how we can improve performance. For example, can the team be trained so that they can create components by using a different tool? Can data access be improved by changing the data access technology?

**PR-2: Determining features** - The features of this application correspond to use cases and usage scenarios. We can identify the usage scenarios that affect the performance of the application and, for each such scenario, specify what the user does and what the application does in response, including how databases and other system services are accessed. In addition, you need to determine how often each feature will be used. This information can help you create tests for measuring performance that resemble actual usage of the application as closely as possible.

**PR-3: Specifying the load** - We can specify the load of this eExamination application as the number of students that will use the application. In addition, we can examine how the load might vary over time. For example, the number of requests for this eExamination site will be higher during certain times of year. We can use the load to define the performance metrics of this application.

## 5.2 Availability Requirements

Availability is a measure of how often the application is available to handle service requests as compared to the planned run time. Availability also takes into account repair time because an application that is being repaired is not available for use.

Designing for availability includes anticipating, detecting, and resolving hardware or software failures before they result in service errors, faults, or data corruption, thereby minimizing downtime. To ensure availability, provide multiple routes to application services and data. Use only tested proven processes (both automated and people-based) that support the application throughout its life cycle.

In addition to unplanned downtime, planned downtime must be reduced. Planned downtime can include maintenance changes, operating system upgrades, backups, or any other activity that temporarily removes the application from service.

Availability of an application also depends on its reliability. For a highly available and reliable application, we need a reliable foundation: good application design, rigorous testing, and certification. Some of the techniques used for designing for availability include:

**AR-1: Reduce planned downtime -** To avoid planned downtime, use rolling upgrades. For example, to update a component on a server, move the server’s resource groups to another server, take the server offline, update the component, and then bring the server online. Meanwhile, the other servers handle the workload, and this application experiences no downtime. You can use this strategy in an application that scales out.

**AR-2: Reduce unplanned downtime with clustering** - Clustering is a technology for creating high-availability applications. A cluster consists of multiple computers that are physically networked and logically connected using cluster software. By using clustering, a multiple server Web site can withstand failures with no interruption in service. When the active server fails, the workload is automatically moved to a passive server, current client processes are switched over, and the failed application service is restarted automatically. If a resource fails, customers connected to that server cluster might experience a slight delay, but the service will be completed. Cluster software can provide failover support for applications, file and print services, databases, and messaging systems that have been designed as cluster-aware and assigned to a cluster.

**AR-3: Use network load balancing** - Network load balancing (NLB) is used to distribute traffic evenly across available servers. NLB also helps increase the availability of an application: if a server fails, you can use NLB to redefine the cluster and direct traffic to the other servers. NLB is especially beneficial for eExamination applications that link external clients with transactions to data servers. As client traffic increases, you can scale out the Web server farm by adding up to 32 servers in a single cluster. NLB automatically detects server failures and redirects client traffic to the remaining servers, all the time maintaining continuous, unbroken client service.

**AR-4: Use redundant array of independent disks (RAID) for data stores. -** RAID uses multiple hard disks to store data in multiple places. If a disk fails, the application is transferred to a mirrored data image and the application continues running. The failed disk can be replaced without stopping the application.

**AR-5: Isolate mission-critical applications** - An application is constantly performing tasks and requesting resources such as network communications, data access, or process threads. Each of these resource requests can affect the performance and availability of applications sharing the same resources. If an application shares these services on the same servers, the workload and throughput characteristics for these servers might change unfavorably. It is recommended that mission-critical applications use dedicated infrastructures and private networks.

**AR-6: Use queuing** - Queuing enables your application to communicate with other applications by sending and receiving asynchronous messages. Queuing guarantees message delivery; it does not matter whether the necessary connectivity currently exists (with mobile applications, for example). Queuing removes a failure point from the application. Queuing is also a solution for managing peak workloads that can require a lot of hardware. In addition, by increasing the number of routes for successful message delivery, an application can increase the chances for successful and immediate message completion.

Calculation of availability

| Measurement Types for Calculating Availability | | |
| --- | --- | --- |
| Name | Calculation | Definition |
| Mean Time Between Failure (MTBF) | Hours / Failure Count | Average length of time the application runs before failing |
| Mean Time To Recovery (MTTR) | Repair Hours / Failure Count | Average length of time needed to repair and restore service after a failure |

The formula for calculating availability is:

Availability = (MTBF / (MTBF + MTTR)) × 100

For example, the typical availability requirement for this eExamination application is that the site is available 24 hours a day, 7 days a week. If you assume 1000 continuous hours as a checkpoint, two 1-hour failures during this time period results in availability of:

((1000 / 2) / ((1000 / 2) + 1)) × 100 = (500 / 501) × 100 = .998 × 100 = 99.8%.

A popular way to describe availability is by the nines, for example, three nines for 99.9 percent availability. However, the implication of measuring by nines is often misunderstood. We need to do the arithmetic to discover that three nines (99.9 percent availability) represent about 8.5 hours of service outage in a single year. The next level, four nines (99.99 percent), represents about 1 hour of service outage in a year. Five nines (99.999 percent) represent about 5 minutes of outage per year.

## 5.3 Reliability Requirement

The reliability of an application refers to the ability of the application to provide accurate results. Reliability and availability are closely related. While availability measures the capacity to handle all requests and to recover from a failure with the least loss of access to the application, reliability measures how long the application can execute and produce expected results without failing. Users bypass unreliable Web sites, resulting in lost revenue and reduced future sales. In addition, the expense of repairing corrupted data increases the cost of application failure. Unreliable systems are also difficult to maintain or improve because the failure points are typically hidden throughout the system. Because of the need no single point failure, automatic failover will be required. In addition, existing disaster recovery and backup plans and procedures must be revised to incorporate the eExamination Application.

To design for reliability, you need to examine the application’s expected usage pattern, create a reliability profile, and create a solution that meets the profile. You must examine how a particular service is provided, evaluate failure scenarios, and design preferred alternatives. In addition, you need to consider the application’s interactions with other applications.

It is difficult to identify reliability problems and solutions for a system that has not been developed. However, we can begin by analyzing the currently running applications in the organization. Such analysis reveals the failure frequency and distribution, root causes, and possible improvements for existing systems. We can use this information to design a reliable solution.

A reliable solution ensures error-free data input, data transformations, state management, and non-corrupting recovery from any failure conditions. Creating a high-reliability application depends on the entire software development lifecycle, from the planning phase, through development and testing, to deployment and stabilizing. The following tasks can help you create a reliable application:

* Putting reliability requirements in the specification
* Using a good architectural infrastructure
* Including management information in the application
* Using redundancy
* Using quality development tools
* Using reliability checks that are provided by the application
* Implementing error handling
* Reducing the application’s functionality instead of completely failing the application

## 5.4 Scalability Requirement

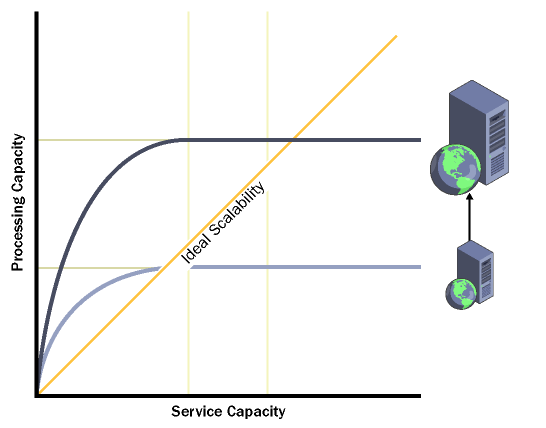
Scalability is defined as the capability to increase resources to produce an increase in the service capacity. In a scalable application, you can add resources to manage additional demands without modifying the application itself.

A scalable application requires a balance between the software and hardware used to implement the application. You might add resources to either software or hardware to increase the scalability of the application. Adding these resources might produce a benefit; however, it could also have a negative or null effect, with the application showing no significant increase in service capacity. For example, you might implement load balancing in an application. This will help only minimally if the application has been written to make synchronous method calls or to retrieve lengthy datasets in response to a user’s request.

eExamination Application an average load of 1500 concurrent users after the system is fully operational, and expects that to grow by 25 percent each year for the next five years.

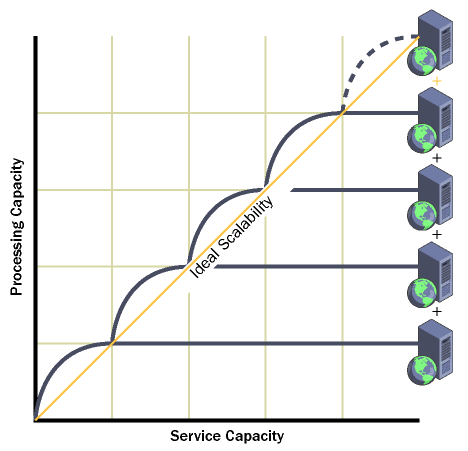
The two most common approaches to scalability are:

**SR-1: Scaling up** - Refers to achieving scalability by improving the existing server’s processing hardware. Scaling up includes adding more memory, more or faster processors, or migrating the application to a more powerful computer. The primary goal of scaling up an application is to increase the hardware resources available to the application. Typically, you can scale up an application without changing the source code. In addition, the administrative effort does not change drastically. However, the benefit of scaling up tapers off eventually until the actual maximum processing capability of the machine is reached.



[Scaling Up]

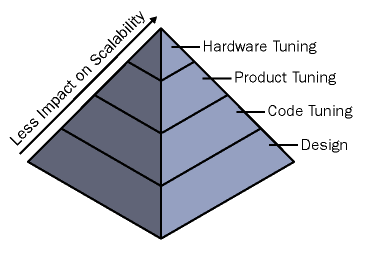
**SR-2: Scaling out** - Refers to distributing the processing load across more than one server. Although scaling out is achieved by using multiple computers, the collection of computers continues to act as the original device configuration from the end-user perspective. Again, the balance between software and hardware is important. The application should be able to execute without needing information about the server on which it is executing. This concept is called location transparency. Scaling out also increases the fault tolerance of the application.



[Scaling Out]

Good design is the foundation of a highly scalable application. The planning phase has the greatest impact on the scalability of an application.

Bellow figure illustrates the role of design, code tuning, product tuning, and hardware tuning in the scalability of an application. Design has more impact on the scalability of an application than the other three factors. As you move up the pyramid, the impact of various factors decreases. The pyramid illustrates that effective design adds more scalability to an application than increased hardware resources.



[Scalability pyramid]

To design for scalability, we need to following guidelines:

**SR-3: Design processes such that they do not wait -** A process should never wait longer than necessary. A process can be categorized as synchronous or asynchronous. A synchronous process waits for another process to complete before it continues. Such processes must wait for another process to succeed or fail completely before performing another operation. Applications that implement synchronous processes encounter bottlenecks for resources. These bottlenecks affect both the performance and the scalability of the application. One way to achieve scalability is to implement asynchronous processes. In applications that have asynchronous processes, long-running operations can be queued for completion later by a separate process.

**SR-4: Design processes so that processes do not compete for resources -** One of the biggest causes of scalability problems is competition for resources such as memory, processor cycles, bandwidth, or database connections. Plan your resource usage to minimize these problems:

* Sequence resource usage to use the most plentiful resources first and the least plentiful resources last.
* Acquire resources as late as possible. The shorter the amount of time a process uses a resource, the sooner the resource becomes available to another process.

**SR-5: Design processes for commutability** - Two or more operations are called commutative if they can execute in any order and still obtain the same result.

**SR-6: Design components for interchangeability** - An interchangeable component is designed to release its resources, move into a pool managed by a resource manager, and be re-initialized for use by a new client. Design the component so that no client-specific state persists from client to client. In addition, the component should support aggregation and not be bound to a specific thread. Resource pooling schemes such as COM+ component pooling and Open Database Connectivity (ODBC) connection pooling use interchangeable resources. For example, you can use the Component Services Administration tool to enable object pooling, set minimum and maximum pool size, and create timeout settings.

**SR-7: Partition resources and activities** - Minimize relationships between resources and activities by partitioning them. This helps you avoid the risk of bottlenecks. Partitioning activities can also ease the load on critical resources such as the processor and bandwidth. For example, using Secure Sockets Layer (SSL) to provide a secure connection results in significant overhead. Therefore, you might decide to use SSL only for pages that require a high level of security and use dedicated Web servers to handle SSL sessions. You can also partition resources and activities by creating many small components rather than a few large components, and by limiting cross-device communication. However, partitioning can make a system more complex. Dividing resources that have dependencies can add significant overheads to an operation.

## 5.5 Security Requirements

Malicious attackers use various methods to exploit system vulnerabilities to achieve their goals. Vulnerabilities are weak points or loopholes in security that an attacker exploits to gain access to an organization’s network or to resources on the network. Some vulnerabilities, such as weak passwords, are not the result of application or software development design decisions. However, it is important for an organization to be aware of such security weaknesses to better protect its systems. Common vulnerabilities of applications include:

**SR-1: Weak passwords** - A weak password might give an attacker access not only to a computer, but to the entire network to which the computer is connected.

**SR-2: Misconfigured software** - Often the manner in which software is configured makes the system vulnerable. If services are configured to use the local system account or are given more permissions than required, attackers can exploit the services to gain access to the system and perform malicious actions on the system.

**SR-3: Social engineering** - A common form of discovering passwords that generally occurs when users are not aware of security issues and can be deceived into revealing their passwords. For example, an attacker posing as a help desk administrator might persuade a user to reveal his or her password under the pretext of performing an administrative task.

**SR-4: Internet connections** - The default installation of Internet Information Services (IIS) version 5.0 often enables more services and ports than are necessary for the operation of a specific application. These additional services and ports provide more opportunities for potential attacks. For example, modem connections bypass firewalls that protect networks from outside intruders. If an intruder can identify the modem telephone number and password, the intruder can connect to any computer on the network.

**SR-5: Unencrypted data transfer** - If the data sent between a server and the users is in clear text, there is a possibility that the data can be intercepted, read, and altered during transmission by an attacker.

**SR-6: Buffer overrun** - Malicious users probe applications looking for ways to trigger a buffer overrun because they can use a buffer overrun to cause an application or an operating system to crash. They can then find more security weaknesses by reading error messages.

**SR-7: SQL injection** - SQL injection occurs when developers dynamically build SQL statements by using user input. The attacker can modify the SQL statement and make it perform operations that were not intended.

**SR-8: Secrets in code** - Many security problems are created when a malicious user is able to find secrets that are embedded in code, such as passwords and encryption keys.

To design a secure eExamination application, we should be familiar with the following principles of security and employ them when creating security strategies:

**SR-9: Rely on tested and proven security systems** - Whenever possible, we should rely on tested and proven security systems rather than creating your own custom solution. Use industry-proven algorithms, techniques, platform-supplied infrastructure, and vendor-tested and supported technologies. If we decide to develop a custom security infrastructure, validate our approach and techniques with expert auditing and security review organizations before and after implementing them.

**SR-10: Never trust external input -** We should validate all data that is entered by users or submitted by other services.

**SR-11: Assume that external systems are not secure -** If our application receives unencrypted sensitive data from an external system, assume that the information is compromised.

**SR-12: Apply the principle of least privilege -** Do not enable more attributes on service accounts than those minimally needed by the application. Access resources with accounts that have the minimal permissions required.

**SR-13: Reduce available components and data -** Risk will increase with the number of components and amount of data you have made available through the application, so you should make available only the functionality that you expect others to use.

**SR-14: Default to a secure mode -** Do not enable services, account rights, and technologies that you do not explicitly need. When we deploy the application on client or server computers, its default configuration should be secure.

**SR-15: Do not rely on security by obscurity -** Encrypting data implies having keys and a proven encryption algorithm. Secure data storage will prevent access under all circumstances. Mixing up strings, storing information in unexpected file paths, and so on, is not security.

**SR-9: Follow STRIDE principles** - Each letter in the STRIDE acronym specifies a different category of security threat: spoofing identity, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. These are classes of security vulnerabilities a system needs to protect itself against.

### Security Features of .NET Technologies

.NET Web applications implement one or more of the logical services by using technologies such as Microsoft ASP.NET, Enterprise Services, XML Web services, remoting, Microsoft ADO.NET, and Microsoft SQL Server. To create effective security strategies, we need to understand how to fine-tune the various security features within each product and technology area, and how to make them work together.

**Authentication**

Authentication is the process of discovering and verifying the identity of a user by examining the user’s credentials and then validating those credentials against some authority. A variety of authentication mechanisms are used, some of which can be used with .NET Framework role-based security.

Examples of commonly used authentication mechanisms include the operating system, Passport, and application-defined mechanisms, such as NTLM and Kerberos version 5 authentication.

**Authorization**

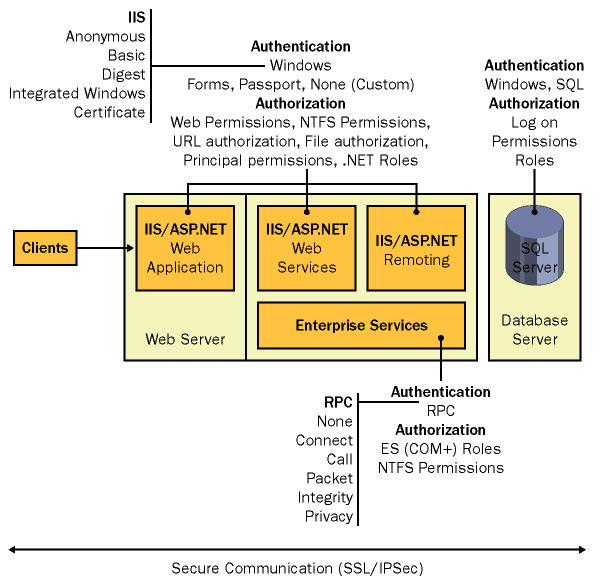
Authorization is the process of determining whether a user is allowed to perform a requested action. Authorization occurs after authentication and uses information about a user’s identity and roles to determine what resources that user can access. You can use .NET Framework role-based security to implement authorization.

**ASP.NET security**

ASP.NET provides a useful tool for application developers to use to create Web pages. When a Web site records a user’s credit card information, the file or database that stores such information must be secured from public access. ASP.NET, in conjunction with IIS, can authenticate user credentials such as names and passwords

**ADO.NET and SQL Server**

ADO.NET provides data access services. It is designed for distributed Web applications, and supports disconnected scenarios. When we build Web-based applications, it is essential that we must use a secure approach to accessing and storing data. ADO.NET and SQL Server provide several security features that can be used to ensure secure data access.



[Security architecture]

## 5.6 Interoperability

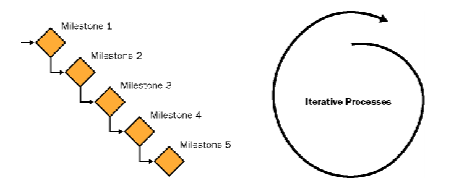
Typically, medium and large organizations have heterogeneous computing environments. For example, many organizations deploy distributed n-tier client/server applications that require access to data or transactions on existing systems. In addition, our application might need to interact with applications that have been developed using proprietary or third-party software. In Version 1.0 of the eExamination Application for Online Examination System, there are no requirements for interoperability with other systems.

# 6. Project Management

## 6.1 Development Methodologies

What Are Process Models and development methodology?

A process model guides the order of project activities and represents the life cycle of a project. Historically, some process models are static and others do not allow checkpoints. Two such process models are the waterfall model and the spiral model.



[The waterfall model and the spiral model]

These models provide two different approaches to the project life cycle. The preceding illustration shows the waterfall model’s cascading checkpoints and the spiral model’s circular approach to process.

### 6.1.1 Waterfall model.

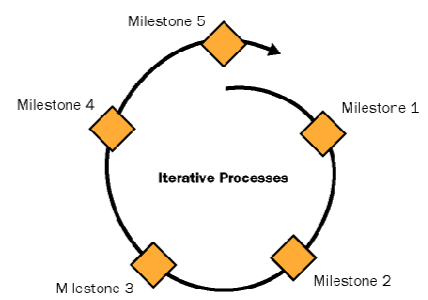
This model uses milestones as transition and assessment points. When using the waterfall model, you need to complete each set of tasks in one phase before moving on to the next phase. The waterfall model works best for projects in which the project requirements can be clearly defined and are not liable to modifications in the future. Because this model has fixed transition points between phases, you can easily monitor schedules and assign clear responsibilities and accountability.

### 6.1.2 Spiral model.

This model is based on the continual need to refine the requirements and estimates for a project. The spiral model is effective when used for rapid application development of very small projects. This approach can generate great synergy between the development team and the customer because the customer is involved in all stages by providing feedback and approval. However, the spiral model does not incorporate clear checkpoints. Consequently, the development process might become chaotic.

### 6.1.3 MSF Process Model:

The MSF – Microsoft Solution Framework Process Model combines the best principles of the waterfall and spiral models. It combines the waterfall model’s milestone-based planning and resulting predictability with the spiral model’s benefits of feedback and creativity.



[MSF Process Model]

In eExamination application devolvement process we have used MSF Process Model. This is a combination of spiral and waterfall process.

## 6.2 Project Development Life Cycle (PDLC)

The project will completed by this four distinct phases as per MSF Process Model.

### 6.2.1 Envisioning

Envisioning is gathering the requirement of the project from different sources. Some techniques for gathering information are interviewing, shadowing, user instructions, and prototyping. Creating and identify the project scope. The scope of the project specifies what will and will not be included in the project. In this phase we have created the scope document. It includes information about the team and project structure, the problem statement, the vision statement, the scope of the project, the solution concept, user profiles, and project goals.

### 6.2.2 Planning

The planning phase results in the architecture and design of the solution, the plans to accomplish the development and deployment of the solution, and the schedules associated with tasks and resources. There are three design processes in the planning phase: conceptual, logical, and physical design.

### 6.2.3 Developing and Designing the System

Design of any application is not complete without a way for users to interact with the system. User interaction takes place through the application’s presentation layer. The presentation layer is the part of the application that provides a communication mechanism between the user and the business service layer of the system.The most simple presentation layers contain user interface components, such as Windows Forms or ASP.NET Web Forms. For more complex user interactions, we can design user process components to orchestrate the user interface elements and control the user interaction. User interface components display data to users, acquire and validate data from user input, and interpret user gestures that indicate the user wants to perform an operation on the data. Additionally, the user interface should filter the available actions to let users perform only the operations that are appropriate at a certain point in time.

### 6.2.4 Stabilizing

The purpose of the stabilizing phase is to reduce the risks of releasing the solution to production.A successful stabilizing phase requires that the team make the transition from a mindset focused on building features to one focused on getting the solution to a known state of quality. Deliverables of the deploying phase are operations and support information systems, repository of all versions of documentation and code, and project closeout reports.

# 7. Estimation and Planning

## 7.1 Planning

The MSF – Microsoft Solution Framework Process Model describes a generalized sequence of activities for building and deploying enterprise solutions. This process is flexible and can accommodate the design and development of a broad range of enterprise projects. The MSF Process Model is a phase-based, milestone-driven, and iterative model that can be applied to developing and deploying traditional applications, enterprise solutions for e-commerce, and Web-distributed applications.

MSF guidance includes disciplines for managing the people, processes, and technology elements that most projects encounter. The three key MSF disciplines are risk management, readiness management, and project management.

### 7.1 Application of the MSF model in our project:

**7.1.1 Envisioning process:**

Each phase in the MSF Process Model has interim milestones and a major milestone. Interim milestones are associated with the various activities that are performed in a phase, such as creating a team and creating a vision/scope document. The major milestone indicates that the team can progress to the next phase in the MSF Process Model. For example, the major milestone of the envisioning phase is the vision/scope approved milestone. When the team reaches this milestone, the team can progress to the planning phase of the MSF Process Model. The team creates deliverables for each task in the envisioning phase. Together, these deliverables provide context and direction for the team for the remainder of the project, and communicate the project vision and scope to the customer. The deliverables that the team creates during the envisioning phase include:

* Vision/scope - Problem statements and business objectives, A review of the existing processes, A broad definition of user requirements.
* Project structure - A project structure and process standards for the team to follow
* Risk assessment - A preliminary risk assessment, Plans for mitigating or eliminating the identified risks

**7.1.2 Planning process:**

During the planning phase, the team determines what to develop and plans how to create the solution. The team prepares the functional specification, creates a design of the solution, and prepares work plans, cost estimates, and schedules for the various deliverables.

The planning phase involves the analysis of requirements. These requirements can be categorized as business requirements, user requirements, operational requirements, and system requirements. These requirements are used to design the solution and its features and to validate the correctness of the design.

After gathering and analyzing the requirements, the team creates the design of the solution. The team creates user profiles that specify the various users of the solution and their roles and responsibilities. The team then creates a series of usage scenarios. A usage scenario specifies the activity performed by a particular type of user. Therefore, the team needs to create usage scenarios for all user profiles. After creating usage scenarios, the team creates use cases for the usage scenarios. A use case specifies the sequence of steps that a user will perform in a usage scenario.

The planning phase deliverables provide the basis for making future tradeoff decisions. The following deliverables are produced during the planning phase:

* Functional specification
* Risk management plan
* Master project plan and master project schedule

**7.1.3 Design Process:**

The three design stages are:

* Conceptual design, in which we view the problem from the perspective of the users and business requirements and define the problem and solution in terms of usage scenarios.
* Logical design, in which we view the solution from the perspective of the project team and define the solution as a set of services.
* Physical design, in which you view the solution from the perspective of the developers and define the technologies, component interfaces, and services of the solution.

We document the solution design in the functional specification. The functional specification describes the behavior and appearance of each feature of the solution. It also describes the architecture and the design for all features.

**7. 1.4 Development process:**

During the developing phase, the project team creates the solution. This process includes creating the code that implements the solution and documenting the code. In addition to developing code, the team also develops the infrastructure for the solution.

The deliverables of the developing phase include:

* Source code and executable files
* Installation scripts and configuration settings for deployment
* Finalized functional specification
* Performance support elements
* Test specifications and test cases

**7. 1.5 Stabilization process:**

During the stabilizing phase, the team performs integration, load, and beta testing on the solution. In addition, the team tests the deployment scenarios for the solution. The team focuses on identifying, prioritizing, and resolving issues so that the solution can be prepared for release. During this phase, the solution progresses from the state of all features being complete as defined in the functional specification for this version to the state of meeting the defined quality levels. In addition, the solution is ready for deployment to the business

The deliverables of the stabilizing phase are as follows:

* Final release
* Release notes
* Performance support elements
* Test results and testing tools
* Source code and executable files
* Project documents
* Milestone review

**7. 1.6 Deployment process:**

During this phase, the team deploys the solution technology and site components, stabilizes the deployment, transfers the project to operations and support, and obtains final customer approval of the project. After deployment, the team conducts a project review and a customer satisfaction survey. The deploying phase culminates in the deployment complete milestone.

The deliverables of the deploying phase are as follows:

* Operation and support information systems
  + Procedures and processes
  + Knowledge base, reports, and logbooks
* Documentation repository for all versions of documents and code developed during the project
* A training plan
* Project completion report
  + Final versions of all project documents
  + Customer satisfaction data
  + Definition of next steps

## 6.2 Estimation

### 6.2. 1 Basic COCOMO

The COnstructive COst MOdel (COCOMO) is an algorithmic Software Cost Estimation Model developed by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics. Constructive Cost Model: It is a hierarchy of estimation models that address: Application composition model: Used during the early stage of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount..

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes

The basic COCOMO equations take the form

Effort Applied = ab([KLOC](http://en.wikipedia.org/wiki/KLOC))bb [ [man-months](http://en.wikipedia.org/wiki/Man-month) ]

Development Time = cb(Effort Applied)db [months]

People required = Effort Applied / Development Time [count]

The coefficients ab, bb, cb and db are given in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software Project | a | b | c | d |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi – Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.2 | 2.5 | 0.32 |

E =3.0(KLOC)1.12

= 3.0(5000)^1.12

= 18 person months

D = 2.5E0.35

= 2.5(22)0. 35

= 6.8 months

Therefore No. of persons required =18/6.8=3persons.

### 6.2.2 Function Point Estimation

A function point is a unit of measurement to express the amount of business functionality an information system provides to a user. Function points are the units of measure used by the IFPUG Functional Size Measurement Method. The IFPUG FSM Method is an ISO recognised software metric to size an information system based on the functionality that is perceived by the user of the information system, independent of the technology used to implement the information system.

The method of measuring the size of an information system and expressing it in a number of function points is called function point analysis (FPA). The method is kept up to date by worldwide cooperating FPA user groups like NESMA and IFPUG. A function point analysis expresses the functional size of an information system in a number of function points (for example: the size of a system is 314 FPs). There are many uses and benefits of function points and the functional size may be used as input into many types of project and organization decisions including determining the:

* Budget for application development or enhancement costs.
* Budget for the annual maintenance costs of the application portfolio.
* Project productivity after completion of the project.
* Software Size for cost estimating.

Function-Oriented Metrics

**FP = count\_total \* [0.65 + 0.01 \* sum of F*i*]**

1. Does the system require reliable backup and recovery=5

2. Are data communications required=4

3. Are there distributed processing functions=2

4. Is performance critical=5

5. Will the system run an existing, heavily utilized operational environment=5

6. Does system requires online data entry=5

7. Does online data entry req. input transaction to be build on multiple screens=3

8. Are master files updated online =4

9. Are I/ps , 0/ps, files and inquires complex=3

10.Is essential processing complex=5

11. Is code reusable=4

12.Are conversion and installation included in design=2

13.Is system supports multiple installations =2

14. Is application designed to facilitate change and ease of use by user=5

sum of **Fi = 54**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Measurement  Parameter | Count |  | Simple | Average | Complex |  |  |
| Number of User Inputs | 12 | \* | 3 | 4 | 6 | = | 36 |
| Number of User outputs | 4 | \* | 4 | 5 | 7 | = | 16 |
| Number of User inquires | 1 | \* | 3 | 4 | 6 | = | 6 |
| Number of files | 5 | \* | 7 | 10 | 15 | = | 35 |
| Number of External interface | 3 | \* | 5 | 7 | 10 | = | 15 |
| Count Total |  |  |  |  |  |  | 108 |

FP = Count\_total \* [0.65 + 0.01 \* sum of F*i*]

FP=108\*[0.65 + 0.01 \* 54]

FP=129

# 7. Preliminary Design

## 7.1 Use Case

eExamination Application 1.0 will address the following use cases. The complete usage scenarios will be completed during the information-gathering process. Use cases will be created and prioritized. Selected use cases will be expanded into usage scenarios and features that are derived from both use cases and the usage scenarios, as represented in the following diagram:

Students

Administrator

[eExamination Usage Scenario – This usage scenario, or scenario for short, describes a real-world example of how one or more people or organizations interact with eExamination system.  It describe the steps, events, and/or actions which occur during the interaction.  This Usage scenarios indicating exactly how someone works with the user interface, or reasonably high level describing the critical business actions but not the indicating how they’re performed.  ]

## 7.2 Specification of actors

The following actors are defined so far in the analysis phase of the eExamination Application for Online Examination System development process.

### 7.2.1 Student

| Student | |
| --- | --- |
| Element | Details |
| Description | A student is a member of the eExamination system. |
| Examples | A Student select examination take the examination, view results online. |

### 7.2.2 Administrator

| Administrator | |
| --- | --- |
| Element | Details |
| Description | The Administrator is the person who can update all the information like examination and question banks. |
| Examples | Administrator updates the examination details, add new question and answers in the eExamination application. |

## 7.3 Specification of Use Cases

### 7.3.1 Use Case Student Registration

| Student Registration | |
| --- | --- |
| Element | Details |
| Actor | Student |
| Trigger | Student is not registered in the System and the student wish to register for a test. |
| Pre Conditions | Student’s Email ID is not listed in the system. Student opens the registration page, and the registration page is displayed. |
| Post Conditions | Student is registered in the system, the student is logged into the system, and the system menu is displayed. |
| Normal course | 1. Registration form appears on the screen 2. System increment the last registered Student ID by 1 to get the ID for the new student. 3. Student fill in the Student’s information including Email ID, Name, Adress(es), Phone number(s), 4. System update |
| Alternative courses | 3a. Not all mandatory data fields are filled  3a1.Student fills in the missing data fields |

**Use Case Diagram : Student Registration**

Student

[Use Case: Student Registration – This use case scenario, or scenario for short, describes how student will registered into eExamination System.  It describes the steps, events, and/or actions which occur during the interaction.  This Usage scenarios indicating exactly how someone works with the student registration interface. ]

### 7.3.2 Use Case. Mange Question Banks

| Mange Question Banks | |
| --- | --- |
| Element | Details |
| Actor | Administrator |
| Trigger | Add new question and answer for an examination |
| Pre Conditions | The question is not in the question bank, the administrator is logged into the system, and the system menu is displayed. |
| Post Conditions | The question and associate answers is in the question bank, the administrator is logged into the system, and the system menu is displayed. |
| Normal Event Flow | 1. The administrator open the question bank page form by choosing the menu item for this action 2. Administrator enters all necessary information about the question and answers in the system. 3. Administrator update the System by confirming the data entered into the registration form. 4. Administrator repeats the steps until complete entry for all questions. |
| Variations | 3a. Mandatory fields in the registration form are missing  3a1. The system reject the entry and system update with an error message about missing mandatory fields. |

**Use Case Diagram: Manage Question Bank**

Administrator

[Use Case: Manage Question Bank – This use case scenario, or scenario for short, describes how administrator will add new question and answer details into eExamination System.  It describes the steps, events, and/or actions which occur during the interaction. ]

### 7.3.3 Use Case. Examination

| EXamination | |
| --- | --- |
| Element | Details |
| Actor | Student |
| Trigger | Student wants to give an examination |
| Pre Conditions | The student must register into the system, the student is logged into the system, and the system menu is displayed. |
| Post Conditions | The examination is completed and result is saved and displayed. User is logged out from the system, and the system menu is displayed. |
| Normal event flow | 1. Student logged in. 2. Student selects the examination and start the examination.. 3. Navigate the question and answers, mark the answers 4. The system calculates the result and displayed the results. 5. Student logged out from the system. |
| Variations | 5a.The system check for the available question from question bank.  2a1.The system informs if found mismatch |

**Use Case Diagram: Examination**

Student

### 

[Use Case: Sales Registration – This use case scenario, or scenario for short, describes how student will give examination and view the results in eExamination system  It describes the steps, events, and/or actions which occur during the interaction. ]

### 7.3.4 Use Case. System Login

| System Login | |
| --- | --- |
| Element | Details |
| Actor | Student, Administrator |
| Trigger | The user wish to start using the system. |
| Pre Conditions | The user is not logged into the system. |
| Post Conditions | The user is logged into the system, and the system menu is displayed. |
| Normal course | 1. The user click the link for the eExamination application and a login page appear on the screen. 2. The user types his username and password into the form and press the login button. 3. The system confirms that the user is logged on. |
| Alternative courses | 2a. The user is not a valid user or the user name or the password is  mistyped.  2a1. The system reject login with an error message that express wrong  login name or password. |

**Use Case Diagram: System Login**

Student

System User

Administrator

[Use Case: System Login – This use case scenario, or scenario for short, describes how actors will perform login eExamination System.  It describes the steps, events, and/or actions which occur during the interaction. ]

## 7.4 Process Flow Chart

Start

Registered

Student Registration

Login

YES

NO

Examination?

View Account Status?

NO

View/Search Examination

View Account Status

Select Examination

Give Examination

Complete Examination?

View Results

View Answers

Logout

Stop

YES

YES

NO

NO

YES

# 7.5 ER Diagram

question\_master

answer\_master

member\_master

test\_master

1: N

Belongs to

1: N

Belongs to

1: N

Belongs to

exam\_master

Exam\_master

1: N

Belongs to

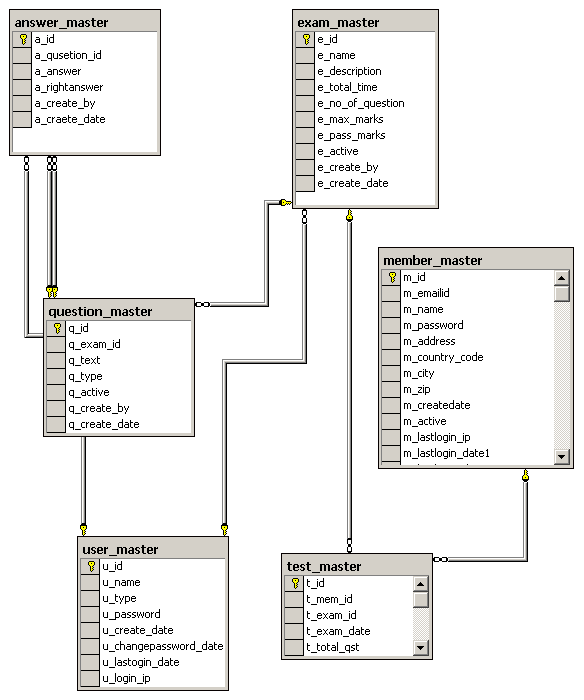
Belongs to

1: N

1: N

Belongs to

# 7.6 Schema Diagram



# 7.7 Data Dictionary

### 7.7.1 Table: answer\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Type** | **Primary Key** | **Foreign Key** |
| 1 | a\_id | int | Y |  |
| 2 | a\_qusetion\_id | int |  | question\_master(q\_id) |
| 3 | a\_answer | varchar(3500) |  |  |
| 4 | a\_rightanswer | int |  |  |
| 5 | a\_create\_by | int |  | user\_master(u\_id) |
| 6 | a\_craete\_date | datetime |  |  |

[*answer\_master* table is used to store multiple answers for every questions, *a\_question\_id* is reference to question\_master table and link to each question]

### 7.7.2 Table: exam\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Type** | **Primary Key** | **Foreign Key** |
| 1 | e\_id | int | Y |  |
| 2 | e\_name | varchar(50) |  |  |
| 3 | e\_description | varchar(250) |  |  |
| 4 | e\_total\_time | int |  |  |
| 5 | e\_no\_of\_question | int |  |  |
| 6 | e\_max\_marks | int |  |  |
| 7 | e\_pass\_marks | int |  |  |
| 8 | e\_active | int |  |  |
| 9 | e\_create\_by | int |  | user\_master(u\_id) |
| 10 | e\_create\_date | datetime |  |  |

[*exam\_master* table is used to store each examination details like name of examination, total marks, passing marks, *e\_id* is the primary key and used in *question\_master, exam\_master* tables as foreign key]

### 7.7.3. Table: member\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Type** | **Primary Key** | **Foreign Key** |
| 1 | m\_id | int | Y |  |
| 2 | m\_emailid | varchar(50) |  |  |
| 3 | m\_name | varchar(50) |  |  |
| 4 | m\_password | varchar(50) |  |  |
| 5 | m\_address | varchar(150) |  |  |
| 6 | m\_country\_code | varchar(50) |  |  |
| 7 | m\_city | varchar(50) |  |  |
| 8 | m\_zip | varchar(6) |  |  |
| 9 | m\_createdate | datetime |  |  |
| 10 | m\_active | int |  |  |
| 11 | m\_lastlogin\_ip | varchar(50) |  |  |
| 12 | m\_lastlogin\_date1 | datetime |  |  |
| 13 | m\_lastlogin\_date2 | datetime |  |  |
| 14 | m\_password\_change\_date | datetime |  |  |

[*member\_master* table is used to storestudent/member details of the student it stores details of mem\_id, name, email id, contact details, login details]

### 7.7.4 Table: question\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Type** | **Primary Key** | **Foreign Key** |
| 1 | q\_id | Int | Y |  |
| 2 | q\_exam\_id | Int |  | exam\_master(e\_id) |
| 3 | q\_text | Varchar(3500) |  |  |
| 4 | q\_type | Varchar(10) |  |  |
| 5 | q\_active | int |  |  |
| 6 | q\_create\_by | int |  |  |
| 7 | q\_create\_date | Date |  |  |

[Table *question\_master* is used to store questiondetails, it stores record of question id, question text, question type wheter it is single selection or multiple choice. *q\_id* is the primary key and represent for the question in eExamination database]

### 7.7.5. Table: test\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Type** | **Primary Key** | **Foreign Key** |
| 1 | t\_id | Int | Y |  |
| 2 | t\_mem\_id | Int |  | member\_master(mem\_id) |
| 3 | t\_exam\_id | Int |  | exam\_master(e\_id) |
| 4 | t\_exam\_date | Int |  |  |
| 5 | t\_total\_qst | Int |  |  |
| 6 | t\_total\_attempt | Int |  |  |
| 7 | t\_total\_right | Int |  |  |
| 8 | t\_total\_wrong | Int |  |  |
| 9 | t\_total\_marks | Int |  |  |
| 10 | t\_marks\_get | Int |  |  |
| 11 | t\_percent | Int |  |  |

[Table *test\_master* is used to store examination result details, it stores record of for each successfully saved test results for all students.]

### 7.7.6 Table: user\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Type** | **Primary Key** | **Foreign Key** |
| 1 | u\_id | Int | Y |  |
| 2 | u\_name | Varchar(50) |  |  |
| 3 | u\_type | Varchar(50) |  |  |
| 4 | u\_password | Varchar(50) |  |  |
| 5 | u\_create\_date | Date |  |  |
| 6 | u\_changepassword\_date | Date |  |  |
| 7 | u\_lastogin\_date | Date |  |  |
| 8 | u\_login\_ip | Varchar(50) |  |  |

### 

[Table *userter* is used to store login details for system user includes admin user]

## 7.8 Data Flow Diagram DFD

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

### 

### 7.8.1 Context Level

Examination

Question Bank

Student

Results

This context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modelled purely in terms of data flows across the system boundary. This context diagram shows the entire eExamination as a single process,

### 7.8.2 Level 1 (High Level Diagram)

Student

Examination  
Master

Question Bank

This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the eExamination system and their interrelation. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1, e.g. the "Select Examination" data flow could be spilt into "View Details" and "View Results" and still be valid.

## 7.9 Class Diagram

|  |
| --- |
| **StudentLogin** |
| * Login() * CheckTrn() * UpdateLastLogin() * checkStuInfo() |

|  |
| --- |
| **StudentInfo** |
| * GetDetails(id) * GenID() |

**1 \***

**1**

**1**

**\***

**1**

|  |
| --- |
| **ExamDetails** |
| * ExamDetails() * ExamResult() |

|  |
| --- |
| **LoginSecutity** |
| * GetLastLogin() * ValidPassword() |

## 7.10 Project Planning And Scheduling

Planning is very important in every aspect of development work. Software project plan indicated scope of the project, milestones and deliverables, project estimates, resource allocation, risk management, scheduling techniques and quality control and standard. Software project plan can be viewed as the following:

**User Interfaces**

The use of Textbox and List box for accepting data. The input data are then stored in the database. These data can also be retrieved from the database and displayed in the form of tables.

## Hardware Interfaces

Raw data inserted into this software are permanently stored in tables; hence processor having free space will be useful for faster storage and retrieval of data. Here we have used 3.6 GHz P4 processor, which will have an additional support for our software.

**Software Interfaces**

This software is operated in a windows environment. This will help for firewall security provisions if the software is used along with internet connection.

Java is itself very secure, as it doesn’t have any memory pointers, and programs run inside virtual machine sandbox.

**Cost of Implementation**

This project can be implemented in the organization in one to two weeks. The cost of this project is derived from effort, hardware cost, travel expenses, training cost, telecommunication costs etc.

**a. Effort:** It includes the total number of manpower per months. As this project is completely computerized hence less number of manpower will be used to successfully run this project. At least 2-3 persons will be enough to maintain this project.

**b. Hardware cost:**  It includes 2 INTEL P4 Standalone Computers. Cost around 60,000.

**c. Training Cost:** One Software personnel will be allotted for providing training to the manpower allotted

**d. Project duration:** It will take complete 2 months for completion. After that it will take another 2-3 weeks for implementation and testing. Another 1-2 weeks is kept in hand for any inconvenience.

**With respect to the customer:** Weekly or timely meetings will be scheduled with the customers for getting time to time feedback. These meetings will be accompanied with presentation reports. After getting feedback further modifications and developments will be done.

## 7.11 Scheduling

Scheduling of a project can be correlated to prioritizing various jobs with respect to their cost, time and duration. Scheduling can be done with resource constraint or time constraint in mind.

**Testing**

**Coding**

**Design**

**Requirement Analysis**

**Maintenance**

**Problem Definition**

**Requirement specification**

**S/w and h/w requirement**

**Planning and scheduling**

g

**Data Design**

**Modular design**

**Unit testing**

**Integration testing**

**System testing**

**Online Examination System**

### 7.11.1 Gantt chart:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PHASE** | **TIME REQUIRED(IN weeks)** | | | | | | | | |
|  | **WK1** | **WK2** | **WK3** | **WK4** | | **WK5** | **WK6** | |
| **reQUIREMENT Gathering** |  |  |  |  | |  |  | |
| **ReQUIREMENT analysis** |  |  |  |  | |  |  | |
| **design** |  |  |  |  |  |  |  | |
| **coding** |  |  |  |  | |  |  | |
| **testing** |  |  |  |  | |  |  |  |
| **implementa-tion** |  |  |  |  | |  |  | |

### 7.11.2 PERT chart:

# 8 Coding and Web Pages

**Req.Gathering**

01 Feb – 08 Feb Wk 1

**Design**

15 Feb-25 Feb

Wk3+4days

**Testing**

03 Mar –13Mar

Wk5+4days

**Implementation**

13 Mar–20Mar

Wk6

**Coding**

25 Feb – 03 Mar

Wk 4

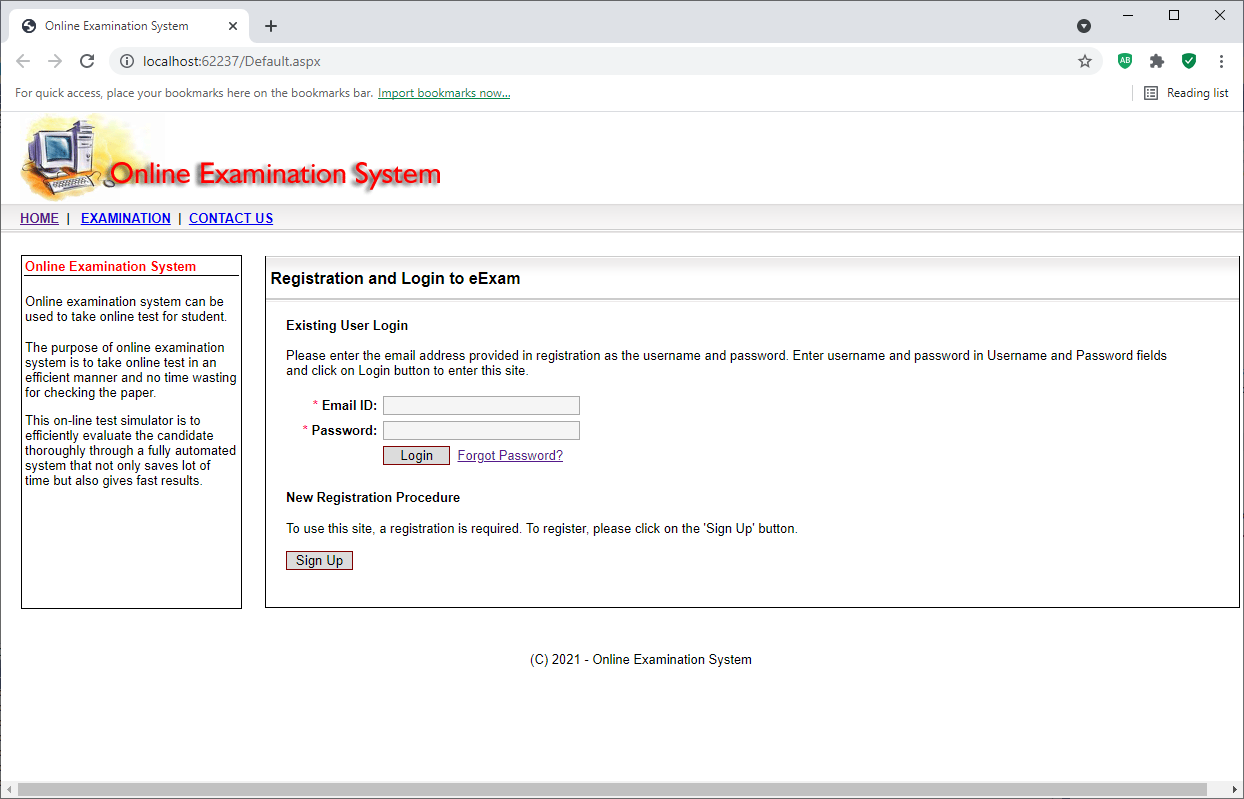
**Analysis**

08 Feb-15 Feb

Wk2

## 8.1 eExamination Home Page

**default.aspx**



**default.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class \_Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

Session["uid"] = "";

Session["uname"] = "";

Session["uemailid"] = "";

}

protected void CmdSignUp\_Click(object sender, EventArgs e)

{

Response.Redirect("newuser.aspx");

}

protected void CmdLogin\_Click(object sender, EventArgs e)

{

//check for login

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter("select \* from member\_master where m\_active=1 and m\_emailid='" + TxtEmailID.Text + "' and m\_password='" + TxtPassword.Text + "' ", cn);

DataSet ds = new DataSet();

da.Fill(ds, "member\_master");

if (ds.Tables["member\_master"].Rows.Count > 0)

{

DataRow R;

R = ds.Tables["member\_master"].Rows[0];

Session["uid"] = R["m\_id"].ToString();

Session["uname"] = R["m\_name"].ToString();

Session["uemailid"] = R["m\_emailid"].ToString();

//update last login date

string strIP;

strIP = Server.MachineName.ToString();

SqlCommand com = new SqlCommand("update member\_master set m\_lastlogin\_ip='" + strIP + "',m\_lastlogin\_date2=m\_lastlogin\_date1, m\_lastlogin\_date1=getdate() where m\_id =" + R["m\_id"].ToString(), cn);

int i;

i = com.ExecuteNonQuery();

Response.Redirect("myaccount.aspx");

}

else

{

ClsMain.CreateMessageAlert(this, "Login fail, Invalid user name/password.", "123");

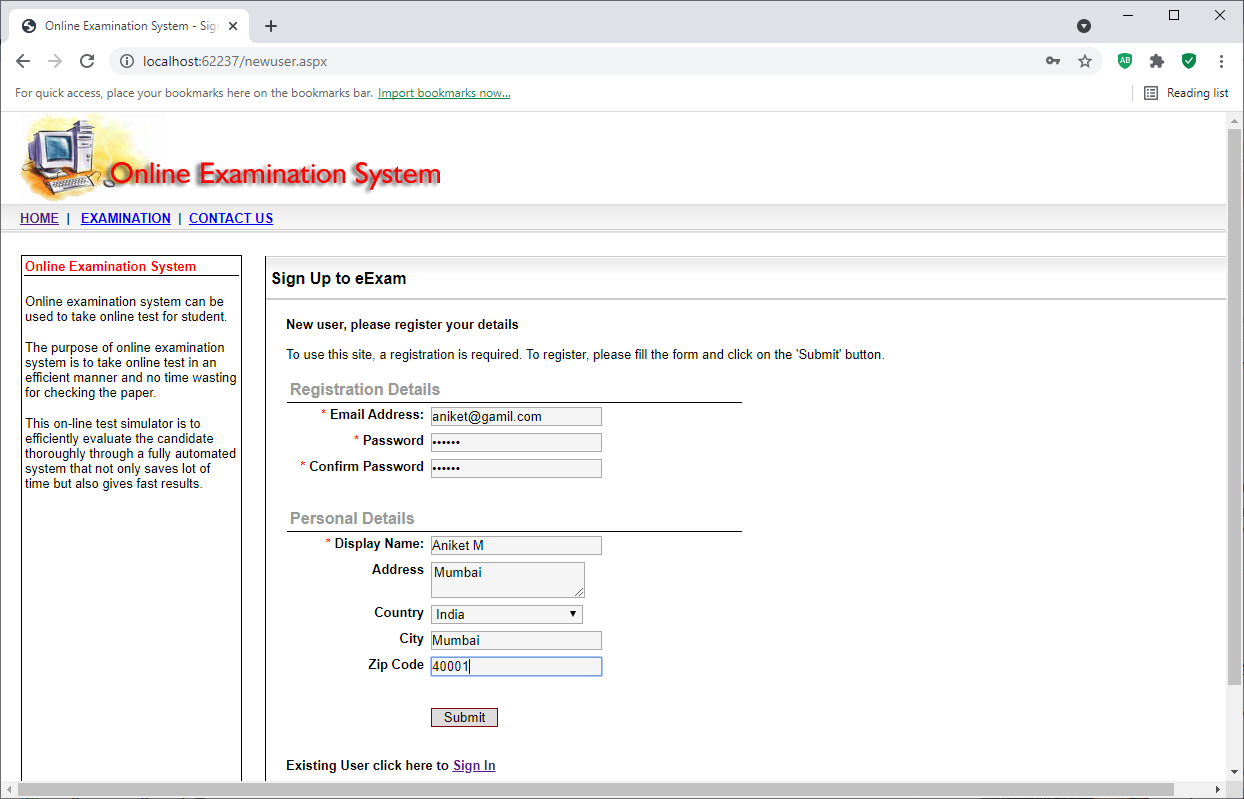
}

}

}

## 8.2 eExamination – Registration Page

**newuser.aspx**



**newuser.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class newuser : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (IsPostBack == true)

{

return;

}

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

DdlCountry.Items.Clear();

SqlDataAdapter da = new SqlDataAdapter("select c\_name from country\_master order by c\_name ", cn);

DataTable dt=new DataTable() ;

da.Fill(dt);

DdlCountry.Items.Add("[--Select Country--]");

if (dt.Rows.Count > 0)

{

int i;

for (i = 0; i <= dt.Rows.Count - 1; i++)

{

DdlCountry.Items.Add(dt.Rows[i]["c\_name"].ToString() );

}

}

}

protected void CmdSubmit\_Click(object sender, EventArgs e)

{

// validation

if (TxtEmailID.Text == "")

{

ClsMain.CreateMessageAlert(this, "Email ID cannot be blank", "123");

return;

}

//check for first name

if (TxtFirstName.Text == "")

{

ClsMain.CreateMessageAlert(this, "First name cannot be blank", "123");

return;

}

//check for country

if (DdlCountry.Text == "[--Select Country--]")

{

ClsMain.CreateMessageAlert(this, "Select country from list.", "123");

return;

}

//open connection

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlCommand com = new SqlCommand("select count(\*) from member\_master where m\_emailid ='" + TxtEmailID.Text + "'", cn);

int i;

i = int.Parse(com.ExecuteScalar().ToString());

if (i > 0)

{

ClsMain.CreateMessageAlert(this, "Email ID already used, enter diffrent email id.", "123");

return;

}

//check for password

if (TxtPassword1.Text.Length < 3)

{

ClsMain.CreateMessageAlert(this, "Password should be atleast 4 characters long", "123");

return;

}

if (TxtPassword1.Text != TxtPassword2.Text)

{

ClsMain.CreateMessageAlert(this, "Password and confirm password dose not match.", "123");

return;

}

//save the record and redirect to login page

SqlTransaction tr;

tr = cn.BeginTransaction();

SqlDataAdapter da = new SqlDataAdapter("select \* from member\_master where 1=2", cn);

da.SelectCommand.Transaction = tr;

SqlCommandBuilder cb = new SqlCommandBuilder(da);

DataSet ds = new DataSet();

da.Fill(ds, "member\_master");

//get the new user\_id

int lastid;

com = new SqlCommand("select max(m\_id) from member\_master ", cn, tr);

lastid = int.Parse(com.ExecuteScalar().ToString());

//new row

DataRow r;

r = ds.Tables["member\_master"].NewRow();

r["m\_id"] = lastid + 1;

r["m\_emailid"] = TxtEmailID.Text;

r["m\_password"] = TxtPassword1.Text;

r["m\_name"] = TxtFirstName.Text;

com = new SqlCommand("select c\_code from country\_master where c\_name ='" + DdlCountry.Text + "'", cn, tr);

r["m\_country\_code"] = com.ExecuteScalar().ToString();

r["m\_address"] = TxtAddress.Text;

r["m\_city"] = TxtCity.Text;

r["m\_zip"] = TxtZipCode.Text;

r["m\_active"] = 1;

r["m\_createdate"] = System.DateTime.Now.ToString();

ds.Tables["member\_master"].Rows.Add(r);

da.Update(ds, "member\_master");

Session["uid"] = lastid + 1;

Session["uname"] = TxtFirstName.Text ;

Session["uemailid"] = TxtEmailID.Text ;

//update last login date

string strIP;

strIP = Server.MachineName.ToString();

com = new SqlCommand("update member\_master set m\_lastlogin\_ip='" + strIP + "',m\_lastlogin\_date2=m\_lastlogin\_date1, m\_lastlogin\_date1=getdate() where m\_id =" + lastid + 1, cn,tr);

i = com.ExecuteNonQuery();

tr.Commit();

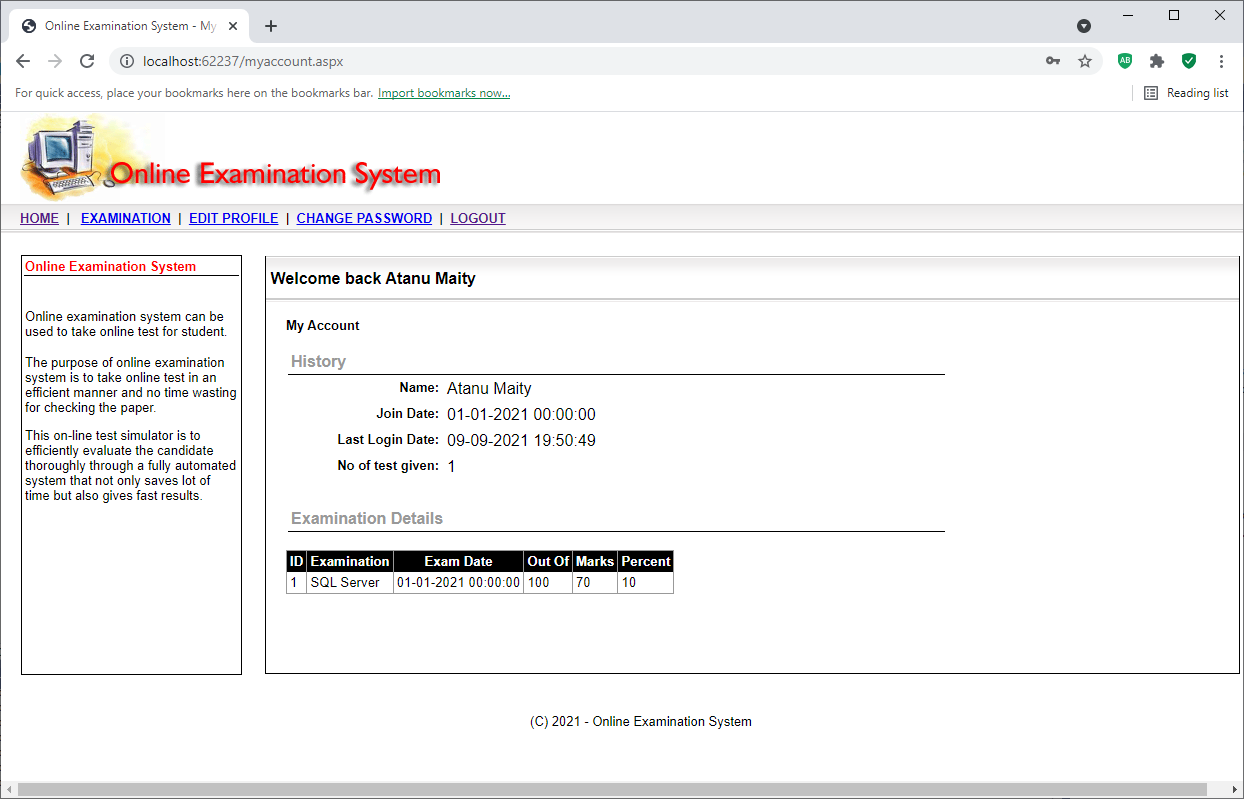
this.ClientScript.RegisterStartupScript(this.GetType(), "Alert", "<script language=\"javaScript\">" + "alert('New user added successfully!');" + "window.location.href='myaccount.aspx';" + "<" + "/script>");

}

}

## 8.3 eExamination – My Account

**myaccount.aspx**



**myaccount.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class myaccount : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//session time out

if (Session["uid"] == null || Session["uid"] == "")

{

Response.Redirect("default.aspx");

return;

}

sp\_welcome.InnerText = Session["uname"].ToString();

//query to get user deatils

String StrQr = "SELECT \* from member\_master where m\_id = " + Session["uid"].ToString();

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataSet ds = new DataSet();

da.Fill(ds, "member\_master");

if (ds.Tables["member\_master"].Rows.Count > 0)

{

DataRow r;

r = ds.Tables["member\_master"].Rows[0];

sp\_name.InnerText = r["m\_name"].ToString();

sp\_joindate.InnerText = r["m\_createdate"].ToString();

sp\_last\_logindate.InnerText = r["m\_lastlogin\_date2"].ToString();

}

// show paper deatils

StrQr = " SELECT test\_master.t\_id as [ID] , exam\_master.e\_name as [Examination], test\_master.t\_exam\_date as [Exam Date], test\_master.t\_total\_marks as [Out Of], test\_master.t\_marks\_get as [Marks], test\_master.t\_percent As [Percent] " +

" FROM test\_master INNER JOIN exam\_master ON test\_master.t\_exam\_id = exam\_master.e\_id where test\_master.t\_mem\_id = " + Session["uid"].ToString() + " order by test\_master.t\_id ";

//query to get user deatils

da = new SqlDataAdapter(StrQr, cn);

ds.Clear();

ds = new DataSet();

da.Fill(ds, "test\_master");

sp\_test\_given.InnerText = ds.Tables["test\_master"].Rows.Count.ToString();

ViewState["ds"] = ds;

GridView1.DataSource = ViewState["ds"];

GridView1.DataMember = "test\_master";

GridView1.DataBind();

}

protected void GridView1\_RowDataBound(object sender, GridViewRowEventArgs e)

{

}

protected void GridView1\_PageIndexChanged(object sender, EventArgs e)

{

}

protected void GridView1\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

GridView1.DataSource = ViewState["ds"];

GridView1.DataMember = "test\_master";

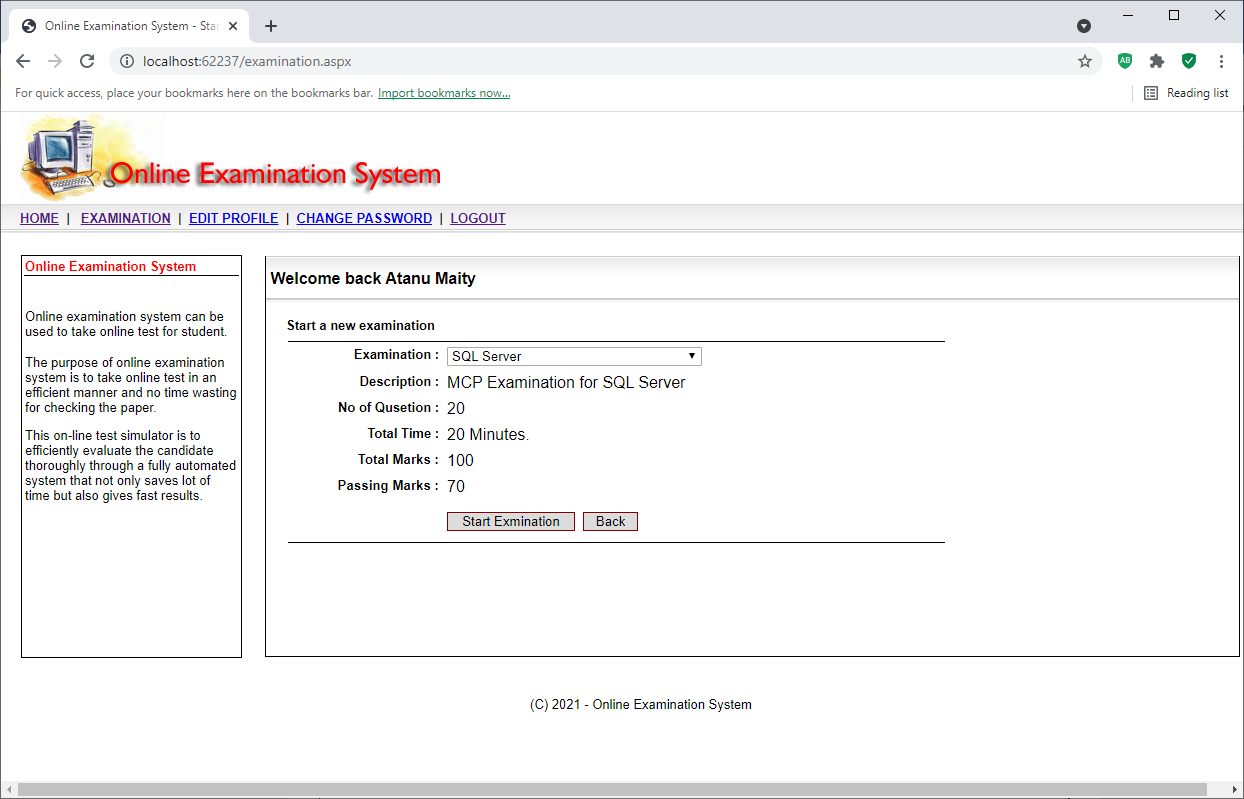
GridView1.DataBind();

}

}

## 8.4 eExamination – New Examination

**examination.aspx**



**examination.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class examination : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//session time out

if (Session["uid"] == null || Session["uid"] == "")

{

Response.Redirect("default.aspx");

return;

}

sp\_welcome.InnerText = Session["uname"].ToString();

if (Page.IsPostBack == true)

{

return;

}

//query to get exam deatils

String StrQr = "SELECT distinct e\_name from exam\_master where e\_active =1 and e\_name is not null order by e\_name";

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataTable dt = new DataTable();

da.Fill(dt);

DdlExamination.Items.Add("---Select Examination---");

int i;

for (i = 0; i <= dt.Rows.Count - 1; i++)

{

DdlExamination.Items.Add(dt.Rows[i]["e\_name"].ToString());

}

}

protected void BtnBack\_Click(object sender, EventArgs e)

{

Response.Redirect("myaccount.aspx");

}

protected void BtnStart\_Click(object sender, EventArgs e)

{

if (DdlExamination.Text == "---Select Examination---")

{

ClsMain.CreateMessageAlert(this, "Select examination from list.", "123");

return;

}

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

int qcount,tcount;

//get the quetion count

String StrQr = "SELECT e\_no\_of\_question from exam\_master where e\_name='" + DdlExamination.Text + "' ";

SqlCommand com=new SqlCommand (StrQr,cn );

tcount=int.Parse (com.ExecuteScalar().ToString());

//get the question count and store in dataset

StrQr = "SELECT count(\*) from question\_master where q\_exam\_id = (select e\_id from exam\_master where e\_name='" + DdlExamination.Text + "')";

com=new SqlCommand (StrQr,cn );

qcount=int.Parse(com.ExecuteScalar().ToString());

if (qcount <tcount )

{

ClsMain.CreateMessageAlert(this, "Unable to start examination, qustion is not available.", "123");

return;

}

Response.Redirect("exammain.aspx?exam=" + DdlExamination.Text);

}

protected void DdlExamination\_SelectedIndexChanged(object sender, EventArgs e)

{

//clear text

sp\_description.InnerHtml = "";

sp\_no\_qusetion.InnerHtml = "";

sp\_passing\_marks.InnerHtml = "";

sp\_total\_marks.InnerHtml = "";

sp\_total\_time.InnerHtml = "";

String StrQr = "SELECT \* from exam\_master where e\_name ='" + DdlExamination.Text + "'";

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataTable dt = new DataTable();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

DataRow r;

r = dt.Rows[0];

sp\_description.InnerHtml = r["e\_description"].ToString();

Session["exam\_id"] = r["e\_id"].ToString();

Session["exam\_name"] = DdlExamination.Text;

Session["exam\_description"] = r["e\_description"].ToString();

sp\_no\_qusetion.InnerHtml = r["e\_no\_of\_question"].ToString();

sp\_passing\_marks.InnerHtml = r["e\_pass\_marks"].ToString();

Session["exam\_pass"] = r["e\_pass\_marks"].ToString();

sp\_total\_marks.InnerHtml = r["e\_max\_marks"].ToString();

Session["exam\_marks"] = r["e\_max\_marks"].ToString();

sp\_total\_time.InnerHtml = r["e\_total\_time"].ToString() + " Minutes.";

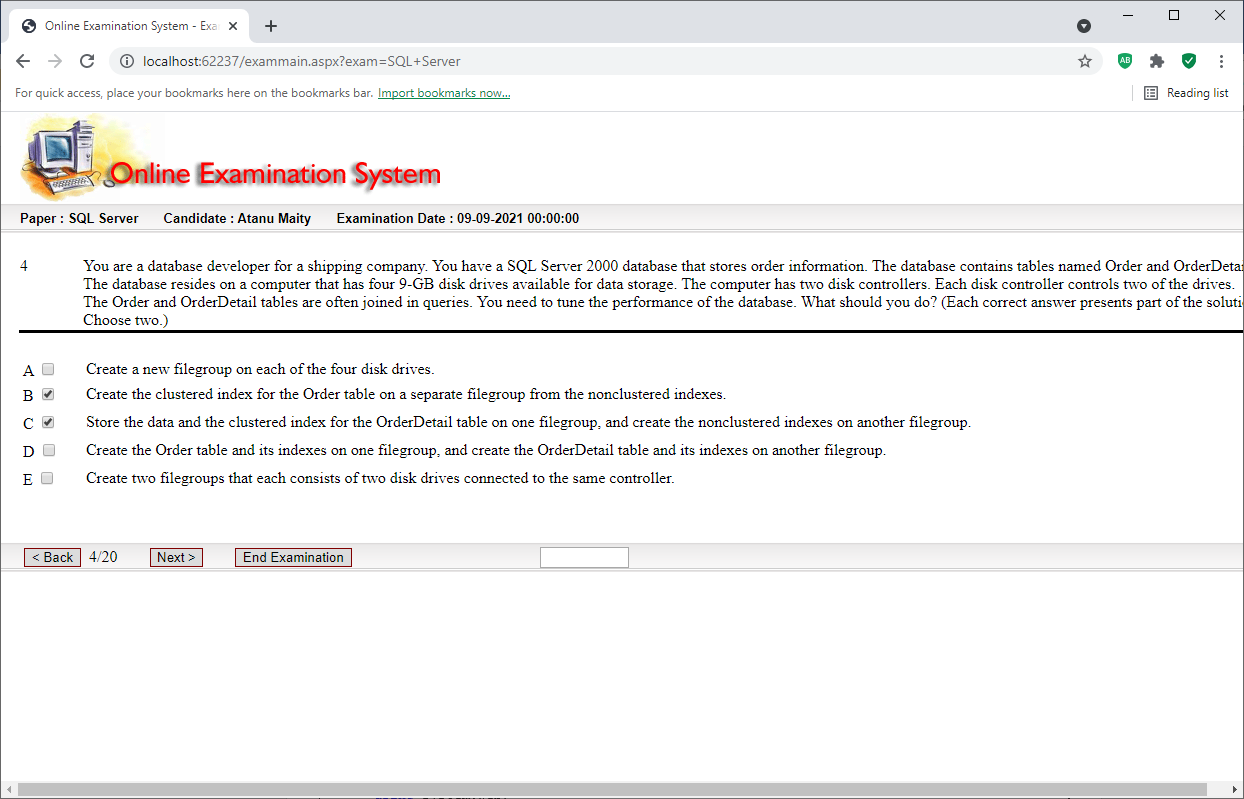
}

}

}

## 8.5 eExamination – Give Examination

**exammain.aspx**



**exammain.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class exammain : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//session time out

if (Session["uid"] == null || Session["uid"] == "")

{

Response.Redirect("default.aspx");

return;

}

sp\_candidate.InnerText = Session["uname"].ToString();

sp\_paper.InnerText = Request.QueryString["exam"].ToString();

sp\_date.InnerText = System.DateTime.Now.Date.ToString();

if (Page.IsPostBack == true)

{

return;

}

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

//create temp data set

DataSet ds = new DataSet();

DataTable dt\_q = new DataTable("q");

dt\_q.Columns.Add("q\_no");

dt\_q.Columns.Add("q\_id");

dt\_q.Columns.Add("q\_text");

dt\_q.Columns.Add("q\_type");

dt\_q.Columns.Add("q\_attempt");

ds.Tables.Add (dt\_q);

DataTable dt\_a = new DataTable("a");

dt\_a.Columns.Add("a\_no");

dt\_a.Columns.Add("a\_q\_no");

dt\_a.Columns.Add("a\_text");

dt\_a.Columns.Add("a\_right\_answer");

dt\_a.Columns.Add("a\_answer\_given");

ds.Tables.Add(dt\_a);

int tq;

//get the quetion count

String StrQr = "SELECT e\_no\_of\_question from exam\_master where e\_name='" + Request.QueryString["exam"].ToString() + "' ";

SqlCommand com = new SqlCommand(StrQr,cn);

tq = int.Parse(com.ExecuteScalar().ToString());

Session["tq"] = tq;

//get the question and store in dataset

StrQr = "SELECT \* from question\_master where q\_exam\_id = (select e\_id from exam\_master where e\_name='" + Request.QueryString["exam"].ToString() + "')";

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataTable dt = new DataTable();

da.Fill(dt);

//take random id from array

int[] a1;

a1 = new int[dt.Rows.Count];

RandomNo(dt.Rows.Count, a1,tq);

DataRow r,r1;

DataRow qr, ar;

int i,j;

for (i = 0; i <= dt.Rows.Count - 1; i++)

{

r = dt.Rows[a1[i]];

qr = ds.Tables["q"].NewRow();

qr["q\_no"] = i + 1;

qr["q\_id"] = r["q\_id"].ToString();

qr["q\_text"] = r["q\_text"].ToString();

qr["q\_type"] = r["q\_type"].ToString();

qr["q\_attempt"] = "";

ds.Tables["q"].Rows.Add(qr);

//get the answer for this qusetion

StrQr = "SELECT \* from answer\_master where a\_qusetion\_id = " + r["q\_id"].ToString();

da = new SqlDataAdapter(StrQr, cn);

DataTable dt1 = new DataTable();

da.Fill(dt1);

for (j=0;j<=dt1.Rows.Count -1;j++)

{

r1=dt1.Rows[j];

ar = ds.Tables["a"].NewRow();

ar["a\_no"] = r1["a\_id"];

ar["a\_q\_no"] = r1["a\_qusetion\_id"];

ar["a\_text"] = r1["a\_answer"];

ar["a\_right\_answer"] = r1["a\_rightanswer"];

ar["a\_answer\_given"] = "";

ds.Tables["a"].Rows.Add(ar);

}

}

Session["qp"] = "0";

ViewState["ds"] = ds;

DisplayQst();

}

protected void BtnBack\_Click(object sender, EventArgs e)

{

int p;

p = int.Parse(Session["qp"].ToString());

//save the ans

DataSet ds1 = new DataSet();

ds1 = (DataSet)ViewState["ds"];

DataRow qr, ar;

qr = ds1.Tables["q"].Rows[p];

string qtype;

qtype = qr["q\_type"].ToString();

DataRow[] dr1;

dr1 = ds1.Tables["a"].Select("a\_q\_no = '" + qr["q\_id"].ToString() + "'");

if (qtype == "SINGLE")

{

if (r1.Visible == true)

{

if (r1.Checked == true)

{

dr1[0]["a\_answer\_given"] = "1";

}

else

{

dr1[0]["a\_answer\_given"] = "0";

}

}

if (r2.Visible == true)

{

if (r2.Checked == true)

{

dr1[1]["a\_answer\_given"] = "1";

}

else

{

dr1[1]["a\_answer\_given"] = "0";

}

}

if (r3.Visible == true)

{

if (r3.Checked == true)

{

dr1[2]["a\_answer\_given"] = "1";

}

else

{

dr1[2]["a\_answer\_given"] = "0";

}

}

if (r4.Visible == true)

{

if (r4.Checked == true)

{

dr1[3]["a\_answer\_given"] = "1";

}

else

{

dr1[3]["a\_answer\_given"] = "0";

}

}

if (r5.Visible == true)

{

if (r5.Checked == true)

{

dr1[4]["a\_answer\_given"] = "1";

}

else

{

dr1[4]["a\_answer\_given"] = "0";

}

}

}

else

{

if (c1.Visible == true)

{

if (c1.Checked == true)

{

dr1[0]["a\_answer\_given"] = "1";

}

else

{

dr1[0]["a\_answer\_given"] = "0";

}

}

if (c2.Visible == true)

{

if (c2.Checked == true)

{

dr1[1]["a\_answer\_given"] = "1";

}

else

{

dr1[1]["a\_answer\_given"] = "0";

}

}

if (c3.Visible == true)

{

if (c3.Checked == true)

{

dr1[2]["a\_answer\_given"] = "1";

}

else

{

dr1[2]["a\_answer\_given"] = "0";

}

}

if (c4.Visible == true)

{

if (c4.Checked == true)

{

dr1[3]["a\_answer\_given"] = "1";

}

else

{

dr1[3]["a\_answer\_given"] = "0";

}

}

if (c5.Visible == true)

{

if (c5.Checked == true)

{

dr1[4]["a\_answer\_given"] = "1";

}

else

{

dr1[4]["a\_answer\_given"] = "0";

}

}

}

ViewState["ds"] = ds1;

///

if (p == 0)

{

ClsMain.CreateMessageAlert(this, "First Question.", "123");

return;

}

Session["qp"] = p - 1;

DisplayQst();

}

protected void BtnNext\_Click(object sender, EventArgs e)

{

int p;

p=int.Parse(Session["qp"].ToString());

//save the ans

//save the ans

DataSet ds1 = new DataSet();

ds1 = (DataSet)ViewState["ds"];

DataRow qr,ar;

qr = ds1.Tables["q"].Rows[p];

string qtype;

qtype=qr["q\_type"].ToString();

DataRow[] dr1;

dr1 = ds1.Tables["a"].Select("a\_q\_no = '" + qr["q\_id"].ToString() + "'");

if (qtype == "SINGLE")

{

if (r1.Visible == true)

{

if (r1.Checked == true)

{

dr1[0]["a\_answer\_given"] = "1";

}

else

{

dr1[0]["a\_answer\_given"] = "0";

}

}

if (r2.Visible == true)

{

if (r2.Checked == true)

{

dr1[1]["a\_answer\_given"] = "1";

}

else

{

dr1[1]["a\_answer\_given"] = "0";

}

}

if (r3.Visible == true)

{

if (r3.Checked == true)

{

dr1[2]["a\_answer\_given"] = "1";

}

else

{

dr1[2]["a\_answer\_given"] = "0";

}

}

if (r4.Visible == true)

{

if (r4.Checked == true)

{

dr1[3]["a\_answer\_given"] = "1";

}

else

{

dr1[3]["a\_answer\_given"] = "0";

}

}

if (r5.Visible == true)

{

if (r5.Checked == true)

{

dr1[4]["a\_answer\_given"] = "1";

}

else

{

dr1[4]["a\_answer\_given"] = "0";

}

}

}

else

{

if (c1.Visible == true)

{

if (c1.Checked == true)

{

dr1[0]["a\_answer\_given"] = "1";

}

else

{

dr1[0]["a\_answer\_given"] = "0";

}

}

if (c2.Visible == true)

{

if (c2.Checked == true)

{

dr1[1]["a\_answer\_given"] = "1";

}

else

{

dr1[1]["a\_answer\_given"] = "0";

}

}

if (c3.Visible == true)

{

if (c3.Checked == true)

{

dr1[2]["a\_answer\_given"] = "1";

}

else

{

dr1[2]["a\_answer\_given"] = "0";

}

}

if (c4.Visible == true)

{

if (c4.Checked == true)

{

dr1[3]["a\_answer\_given"] = "1";

}

else

{

dr1[3]["a\_answer\_given"] = "0";

}

}

if (c5.Visible == true)

{

if (c5.Checked == true)

{

dr1[4]["a\_answer\_given"] = "1";

}

else

{

dr1[4]["a\_answer\_given"] = "0";

}

}

}

ViewState["ds"] = ds1;

///

if (p >= int.Parse(Session["tq"].ToString())-1 )

{

ClsMain.CreateMessageAlert(this, "Last Question.", "123");

return;

}

Session["qp"] = p + 1;

DisplayQst();

}

protected void BtnEnd\_Click(object sender, EventArgs e)

{

Session["mainds"] = (DataSet)ViewState["ds"];

Response.Redirect("examresults.aspx");

}

void DisplayQst()

{

//reset qst and ans

sp\_ans\_op1.InnerText = "";

sp\_ans\_op2.InnerText = "";

sp\_ans\_op3.InnerText = "";

sp\_ans\_op4.InnerText = "";

sp\_ans\_op5.InnerText = "";

sp\_ans1.InnerText = "";

sp\_ans2.InnerText = "";

sp\_ans3.InnerText = "";

sp\_ans4.InnerText = "";

sp\_ans5.InnerText = "";

sp\_qst.InnerText = "";

sp\_qst\_no.InnerText = "";

r1.Visible =false;

r2.Visible =false ;

r3.Visible =false ;

r4.Visible = false;

r5.Visible = false;

r1.Checked = false;

r2.Checked = false;

r3.Checked = false;

r4.Checked = false;

r5.Checked = false;

c1.Visible = false;

c2.Visible = false;

c3.Visible = false;

c4.Visible = false;

c5.Visible = false;

c1.Checked = false;

c2.Checked = false;

c3.Checked = false;

c4.Checked = false;

c5.Checked = false;

int i,p;

p = int.Parse( Session["qp"].ToString() );

DataSet ds1 = new DataSet() ;

ds1 = (DataSet)ViewState["ds"];

DataRow qr, ar;

qr = ds1.Tables["q"].Rows[p];

sp\_qst\_no.InnerText = qr["q\_no"].ToString();

sp\_qst.InnerText = qr["q\_text"].ToString();

//get answers

DataRow[] dr1;

dr1=ds1.Tables["a"].Select("a\_q\_no = '" + qr["q\_id"].ToString() + "'");

for (i = 0; i <= dr1.Length - 1; i++)

{

ar = dr1[i];

if (qr["q\_type"].ToString() == "SINGLE")

{

if (i == 0)

{

r1.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r1.Checked = false;

}

else

{

r1.Checked = true;

}

}

if (i == 1)

{

r2.Visible = true;

if (string.Equals(ar["a\_answer\_given"],"1")==false)

{

r2.Checked = false;

}

else

{

r2.Checked = true;

}

}

if (i == 2)

{

r3.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r3.Checked = false;

}

else

{

r3.Checked = true;

}

}

if (i == 3)

{

r4.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r4.Checked = false;

}

else

{

r4.Checked = true;

}

}

if (i == 4)

{

r5.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r5.Checked = false;

}

else

{

r5.Checked = true;

}

}

}

if (qr["q\_type"].ToString() == "MULTIPLE")

{

if (i == 0)

{

c1.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c1.Checked = true;

}

}

if (i == 1)

{

c2.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c2.Checked = true;

}

}

if (i == 2)

{

c3.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c3.Checked = true;

}

}

if (i == 3)

{

c4.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c4.Checked = true;

}

}

if (i == 4)

{

c5.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c5.Checked = true;

}

}

}

if (i == 0)

{

sp\_ans\_op1.InnerText = "A";

sp\_ans1.InnerText = ar["a\_text"].ToString();

}

if (i == 1)

{

sp\_ans\_op2.InnerText = "B";

sp\_ans2.InnerText = ar["a\_text"].ToString();

}

if (i == 2)

{

sp\_ans\_op3.InnerText = "C";

sp\_ans3.InnerText = ar["a\_text"].ToString();

}

if (i == 3)

{

sp\_ans\_op4.InnerText = "D";

sp\_ans4.InnerText = ar["a\_text"].ToString();

}

if (i == 4)

{

sp\_ans\_op5.InnerText = "E";

sp\_ans5.InnerText = ar["a\_text"].ToString();

}

}

LblPos.Text = qr["q\_no"].ToString() +"/" + Session["tq"].ToString();

}

void RandomNo(int max, int[] a ,int tq)

{

Random RandomClass = new Random();

int RandomNumber;

//a = new int[tq];

int i, j,p;

p = 0;

bool ch;

L1:

ch = false;

for (i = 0; i <= tq - 1; i++)

{

RandomNumber = RandomClass.Next(0, max);

ch = false;

for (j = 0; j <= tq-1 ; j++)

{

//if (a[j] == 0)

//{

// ch = true;

// goto L1;

//}

if (a[j] == RandomNumber )

{

ch = true;

goto L1;

}

}

if (ch == false)

{

a[p] = RandomNumber;

p++;

if (p >= tq)

{

return;

}

}

}

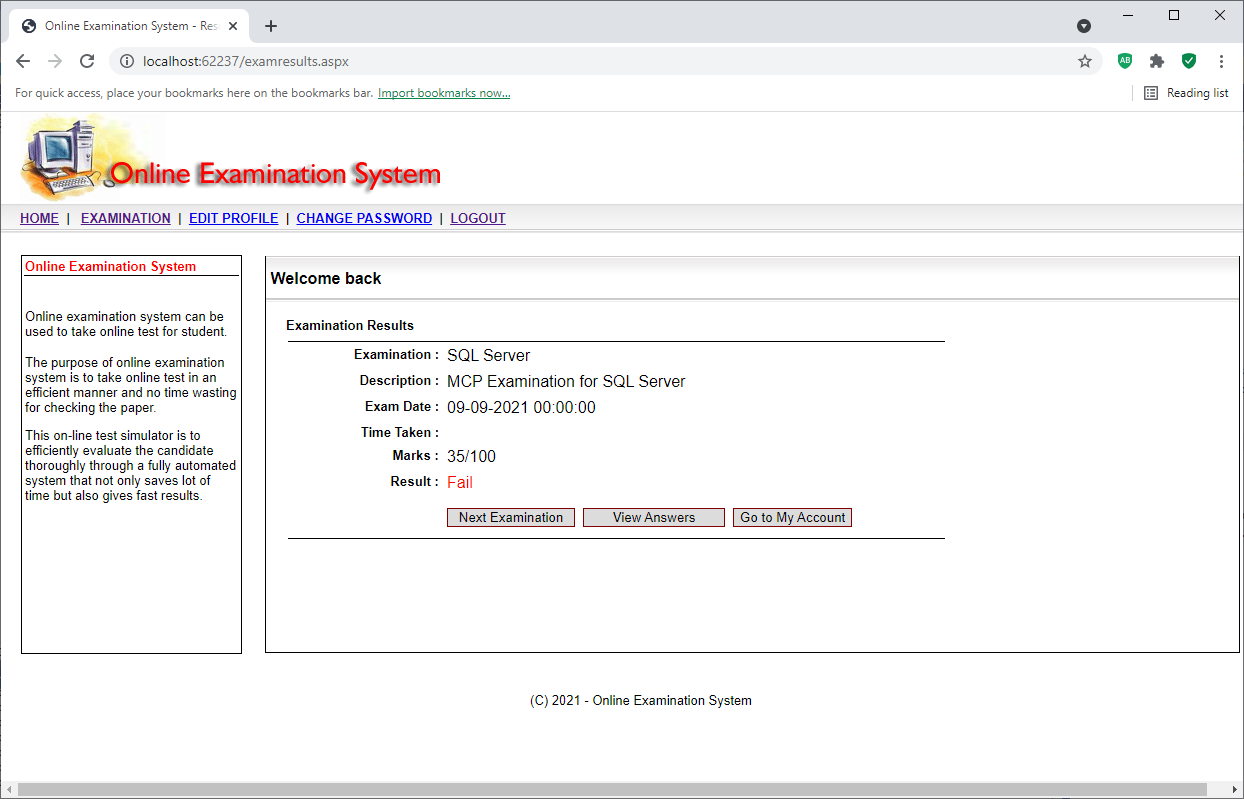
//L2:

}

}

## 8.6 eExamination – Results

**results.aspx**



**results.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class examresults : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Page.IsPostBack == true)

{

return;

}

//dispaly exam details

sp\_examination.InnerText = Session["exam\_name"].ToString();

sp\_description.InnerText = Session["exam\_description"].ToString();

sp\_exam\_date.InnerText = System.DateTime.Now.Date.ToString();

//calculate results

DataSet ds1 = new DataSet();

ds1 = (DataSet)Session["mainds"];

DataRow qr, ar;

int i,j, c,ta;

ta=0;

c = 0;

bool ch\_noattempt, ch\_wronganswer;

for (i = 0; i <= ds1.Tables["q"].Rows.Count - 1; i++)

{

ch\_noattempt = false;

ch\_wronganswer = false;

int a1, a2, a3 ;

a1 = 0;

a2 = 0;

a3 = 0;

qr = ds1.Tables["q"].Rows[i];

DataRow[] dr1;

dr1 = ds1.Tables["a"].Select("a\_q\_no = '" + qr["q\_id"].ToString() + "'");

for (j = 0; j <= dr1.Length - 1; j++)

{

ar = dr1[j];

if (string.Equals(ar["a\_answer\_given"].ToString(),"1")==true && string.Equals(ar["a\_right\_answer"].ToString(),"1")==true )

{

a2 = a2 + 1;

}

a1=a1 + int.Parse(ar["a\_right\_answer"].ToString());

if (string.Equals(ar["a\_answer\_given"].ToString(), "1") == true)

{

ch\_noattempt = true;

}

}

if (ch\_noattempt ==false )

{

ta+=1;

}

if (a1==a2)

{

c += 1;

}

}

int x,t;

x = int.Parse(Session["exam\_marks"].ToString()) / int.Parse(Session["tq"].ToString());

t = x \* c;

sp\_marks.InnerText = t.ToString() + "/" + Session["exam\_marks"].ToString();

int p;

p = c / int.Parse(Session["tq"].ToString()) \* 100;

string r;

if (p >= int.Parse(Session["exam\_pass"].ToString()))

{

r = "<font color='blue'>Pass</blue>";

}

else

{

r = "<font color='red'>Fail</font>";

}

sp\_result.InnerHtml = r;

//save results

String StrQr = "SELECT \* from test\_master where 1=2 ";

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

SqlCommandBuilder cb = new SqlCommandBuilder(da);

DataTable dt = new DataTable();

da.Fill(dt);

DataRow r1;

r1 = dt.NewRow();

r1["t\_mem\_id"] = int.Parse( Session["uid"].ToString());

r1["t\_exam\_id"] = int.Parse(Session["exam\_id"].ToString());

r1["t\_exam\_date"] = System.DateTime.Now.ToString();

r1["t\_total\_qst"] = int.Parse(Session["tq"].ToString());

r1["t\_total\_attempt"] = ta;

r1["t\_not\_atempt"] = int.Parse(Session["tq"].ToString()) - ta;

r1["t\_total\_right"] = c;

r1["t\_total\_wrong"] = int.Parse(Session["tq"].ToString()) - c;

r1["t\_total\_marks"] = int.Parse(Session["exam\_marks"].ToString());

r1["t\_marks\_get"] = t;

r1["t\_percent"] = p;

dt.Rows.Add(r1);

da.Update(dt);

}

protected void BtnStart\_Click(object sender, EventArgs e)

{

Response.Redirect("examination.aspx");

}

protected void BtnBack\_Click(object sender, EventArgs e)

{

Response.Redirect("viewanswers.aspx");

}

protected void BtnMyAccount\_Click(object sender, EventArgs e)

{

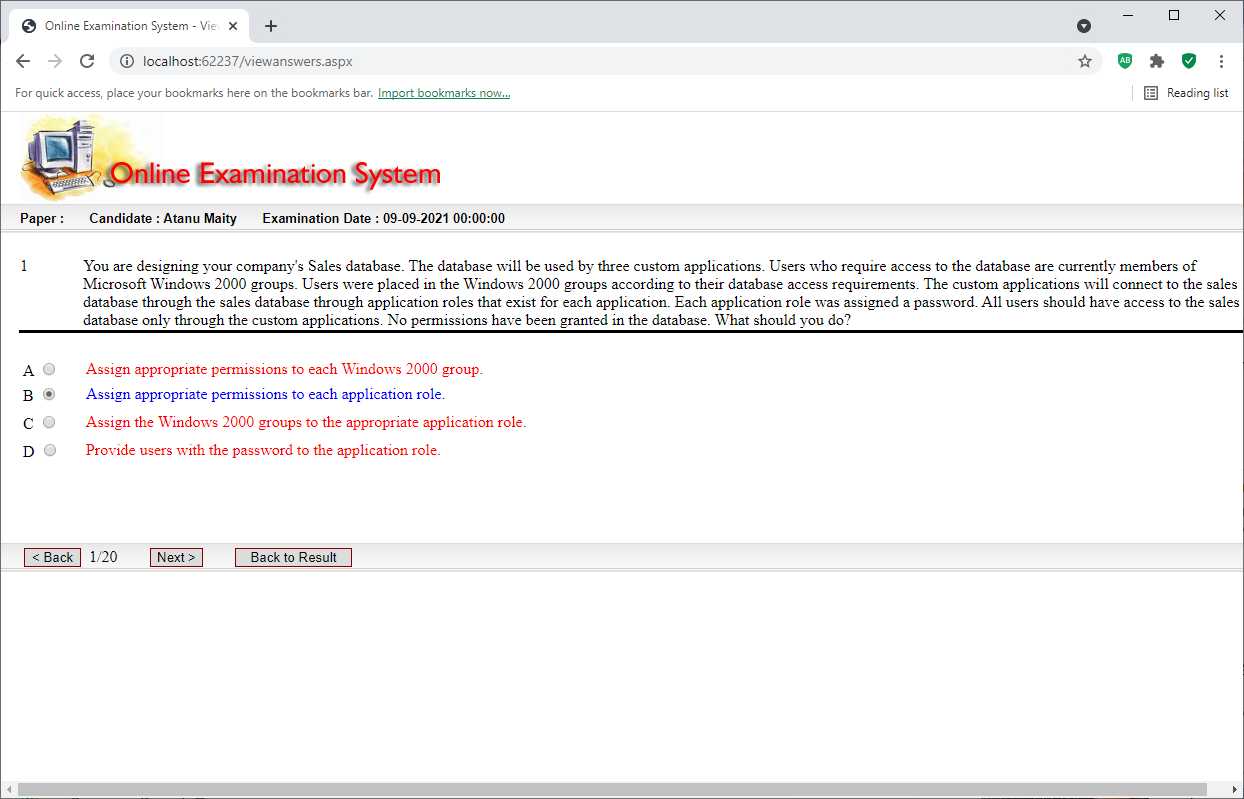
Response.Redirect("myaccount.aspx");

}

}

## 8.7 eExamination –View Answers

**viewanswers.aspx**



**viewanswers.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class viewanswers : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Page.IsPostBack == true)

{

return;

}

sp\_candidate.InnerText = Session["uname"].ToString();

//sp\_paper.InnerText = Request.QueryString["exam"].ToString();

sp\_date.InnerText = System.DateTime.Now.Date.ToString();

DataSet ds1 = new DataSet();

ds1 = (DataSet)Session["mainds"];

ViewState["ds"] = ds1;

Session["qp"] = 0;

DisplayQst();

}

protected void BtnEnd\_Click(object sender, EventArgs e)

{

Response.Redirect("examresults.aspx");

}

protected void BtnBack\_Click(object sender, EventArgs e)

{

int p;

p = int.Parse(Session["qp"].ToString());

///

if (p == 0)

{

ClsMain.CreateMessageAlert(this, "First Question.", "123");

return;

}

Session["qp"] = p - 1;

DisplayQst();

}

protected void BtnNext\_Click(object sender, EventArgs e)

{

int p;

p = int.Parse(Session["qp"].ToString());

if (p >= int.Parse(Session["tq"].ToString()) - 1)

{

ClsMain.CreateMessageAlert(this, "Last Question.", "123");

return;

}

Session["qp"] = p + 1;

DisplayQst();

}

void DisplayQst()

{

//reset qst and ans

sp\_ans\_op1.InnerText = "";

sp\_ans\_op2.InnerText = "";

sp\_ans\_op3.InnerText = "";

sp\_ans\_op4.InnerText = "";

sp\_ans\_op5.InnerText = "";

sp\_ans1.InnerText = "";

sp\_ans2.InnerText = "";

sp\_ans3.InnerText = "";

sp\_ans4.InnerText = "";

sp\_ans5.InnerText = "";

sp\_qst.InnerText = "";

sp\_qst\_no.InnerText = "";

r1.Visible = false;

r2.Visible = false;

r3.Visible = false;

r4.Visible = false;

r5.Visible = false;

r1.Checked = false;

r2.Checked = false;

r3.Checked = false;

r4.Checked = false;

r5.Checked = false;

c1.Visible = false;

c2.Visible = false;

c3.Visible = false;

c4.Visible = false;

c5.Visible = false;

c1.Checked = false;

c2.Checked = false;

c3.Checked = false;

c4.Checked = false;

c5.Checked = false;

int i, p;

p = int.Parse(Session["qp"].ToString());

DataSet ds1 = new DataSet();

ds1 = (DataSet)ViewState["ds"];

DataRow qr, ar;

qr = ds1.Tables["q"].Rows[p];

sp\_qst\_no.InnerText = qr["q\_no"].ToString();

sp\_qst.InnerText = qr["q\_text"].ToString();

//get answers

DataRow[] dr1;

dr1 = ds1.Tables["a"].Select("a\_q\_no = '" + qr["q\_id"].ToString() + "'");

for (i = 0; i <= dr1.Length - 1; i++)

{

ar = dr1[i];

if (qr["q\_type"].ToString() == "SINGLE")

{

if (i == 0)

{

r1.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r1.Checked = false;

}

else

{

r1.Checked = true;

}

}

if (i == 1)

{

r2.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r2.Checked = false;

}

else

{

r2.Checked = true;

}

}

if (i == 2)

{

r3.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r3.Checked = false;

}

else

{

r3.Checked = true;

}

}

if (i == 3)

{

r4.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r4.Checked = false;

}

else

{

r4.Checked = true;

}

}

if (i == 4)

{

r5.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == false)

{

r5.Checked = false;

}

else

{

r5.Checked = true;

}

}

}

if (qr["q\_type"].ToString() == "MULTIPLE")

{

if (i == 0)

{

c1.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c1.Checked = true;

}

}

if (i == 1)

{

c2.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c2.Checked = true;

}

}

if (i == 2)

{

c3.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c3.Checked = true;

}

}

if (i == 3)

{

c4.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c4.Checked = true;

}

}

if (i == 4)

{

c5.Visible = true;

if (string.Equals(ar["a\_answer\_given"], "1") == true)

{

c5.Checked = true;

}

}

}

string fc;

fc = "<font color='red'>";

if (i == 0)

{

sp\_ans\_op1.InnerText = "A";

if (string.Equals(ar["a\_right\_answer"], "1") == true)

{

fc = "<font color='blue'>";

}

sp\_ans1.InnerHtml =fc + ar["a\_text"].ToString() + "</font>";

}

if (i == 1)

{

sp\_ans\_op2.InnerText = "B";

if (string.Equals(ar["a\_right\_answer"], "1") == true)

{

fc = "<font color='blue'>";

}

sp\_ans2.InnerHtml = fc + ar["a\_text"].ToString() + "</font>";

}

if (i == 2)

{

sp\_ans\_op3.InnerText = "C";

if (string.Equals(ar["a\_right\_answer"], "1") == true)

{

fc = "<font color='blue'>";

}

sp\_ans3.InnerHtml = fc + ar["a\_text"].ToString() + "</font>";

}

if (i == 3)

{

sp\_ans\_op4.InnerText = "D";

if (string.Equals(ar["a\_right\_answer"], "1") == true)

{

fc = "<font color='blue'>";

}

sp\_ans4.InnerHtml = fc + ar["a\_text"].ToString() + "</font>";

}

if (i == 4)

{

sp\_ans\_op5.InnerText = "E";

if (string.Equals(ar["a\_right\_answer"], "1") == true)

{

fc = "<font color='blue'>";

}

sp\_ans5.InnerHtml = fc + ar["a\_text"].ToString() + "</font>";

}

}

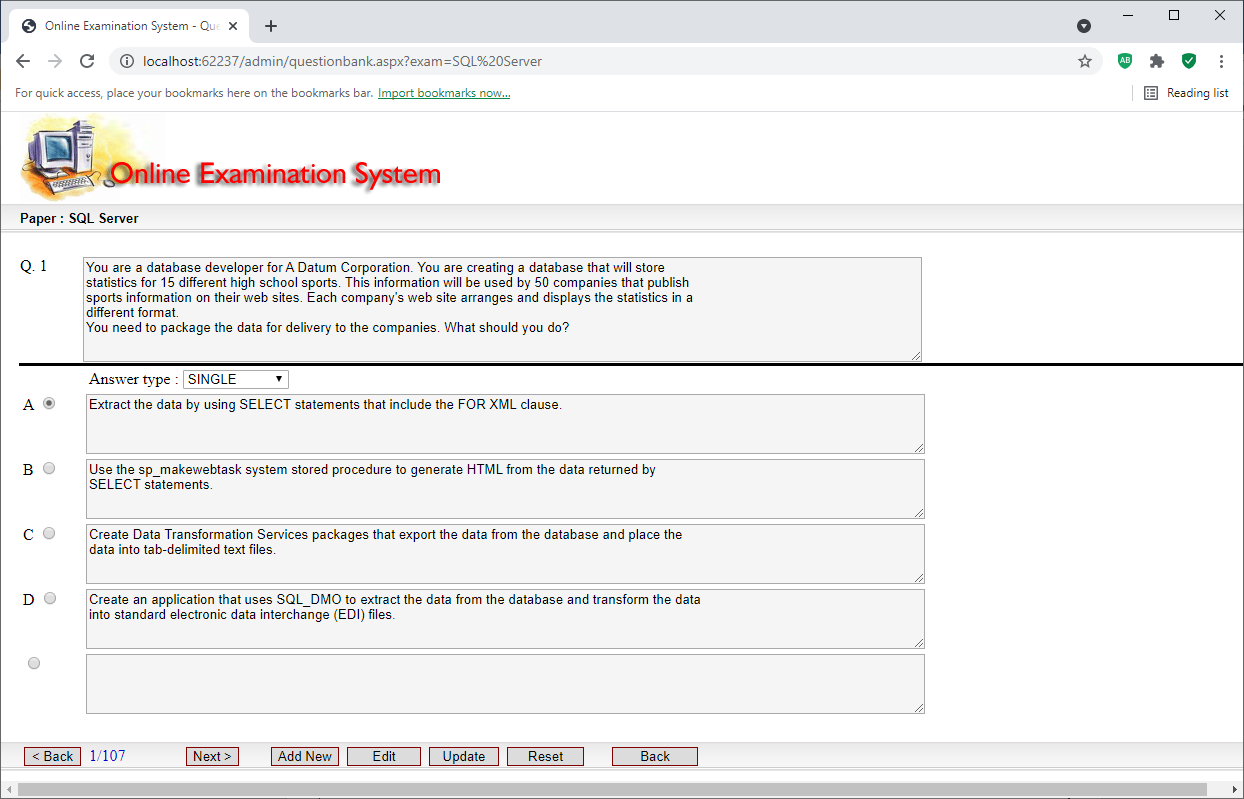
LblPos.Text = qr["q\_no"].ToString() + "/" + Session["tq"].ToString();

}

}

## 8.8 eExamination – Admin Question Bank

**qusetionbank.aspx**



**qusetionbank.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class admin\_questionbank : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

sp\_paper.InnerText = Session["exam"].ToString();

if (Page.IsPostBack == true)

{

return;

}

ed(true, false);

cleartext();

string en;

en = Request.QueryString["exam"].ToString();

//query to get questions

String StrQr = "SELECT \* from question\_master where q\_exam\_id in (select e\_id from exam\_master where e\_name ='" + en + "' ) order by q\_id ";

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataSet ds = new DataSet();

da.Fill(ds,"q");

ViewState["ds"] = ds;

int p;

p = 0;

Session["p"] = p;

if (ds.Tables["q"].Rows.Count > 0)

{

display(ds, p,cn);

}

}

void display(DataSet ds, int p,SqlConnection cn)

{

cleartext();

int p1;

p1 = p + 1;

DataRow r;

r = ds.Tables["q"].Rows[p];

sp\_qst\_no.InnerText = "Q. " + p1.ToString() ;

TxtQst.Text = r["q\_text"].ToString();

ddltype.Text = r["q\_type"].ToString();

LblPos.Text = p1.ToString() + "/" + ds.Tables["q"].Rows.Count.ToString();

//display question

displayop(ddltype.Text);

SqlDataAdapter da = new SqlDataAdapter("select \* from answer\_master where a\_qusetion\_id =" + r["q\_id"].ToString() + " order by a\_id", cn);

DataSet ds1 = new DataSet();

da.Fill(ds1, "a");

if (ds1.Tables["a"].Rows.Count > 0)

{

int i;

for (i = 0; i <= ds1.Tables["a"].Rows.Count - 1; i++)

{

DataRow r11;

r11 = ds1.Tables["a"].Rows[i];

if (i == 0)

{

sp\_ans\_op1.InnerText = "A";

txtans1.Text = r11["a\_answer"].ToString();

if (string.Equals(r11["a\_rightanswer"].ToString(), "1") == true)

{

if (ddltype.Text == "SINGLE")

{

r1.Checked = true;

}

else

{

c1.Checked = true;

}

}

}

if (i == 1)

{

sp\_ans\_op2.InnerText = "B";

txtans2.Text = r11["a\_answer"].ToString();

if (string.Equals(r11["a\_rightanswer"].ToString(), "1") == true)

{

if (ddltype.Text == "SINGLE")

{

r2.Checked = true;

}

else

{

c2.Checked = true;

}

}

}

if (i == 2)

{

sp\_ans\_op3.InnerText = "C";

txtans3.Text = r11["a\_answer"].ToString();

if (string.Equals(r11["a\_rightanswer"].ToString(), "1") == true)

{

if (ddltype.Text == "SINGLE")

{

r3.Checked = true;

}

else

{

c3.Checked = true;

}

}

}

if (i == 3)

{

sp\_ans\_op4.InnerText = "D";

txtans4.Text = r11["a\_answer"].ToString();

if (string.Equals(r11["a\_rightanswer"].ToString(), "1") == true)

{

if (ddltype.Text == "SINGLE")

{

r4.Checked = true;

}

else

{

c4.Checked = true;

}

}

}

if (i == 4)

{

sp\_ans\_op5.InnerText = "E";

txtans5.Text = r11["a\_answer"].ToString();

if (string.Equals(r11["a\_rightanswer"].ToString(), "1") == true)

{

if (ddltype.Text == "SINGLE")

{

r5.Checked = true;

}

else

{

c5.Checked = true;

}

}

}

}

}

}

protected void BtnBack\_Click(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

int p;

p = int.Parse(Session["p"].ToString());

if (p <= 0)

{

ClsMain.CreateMessageAlert(this, "First Record.", "123");

return;

}

p = p - 1;

DataSet ds = new DataSet();

ds = (DataSet)ViewState["ds"];

Session["p"] = p;

if (ds.Tables["q"].Rows.Count > 0)

{

display(ds, p,cn);

}

}

protected void BtnNext\_Click(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

DataSet ds=new DataSet();

ds=(DataSet )ViewState["ds"] ;

int p;

p = int.Parse(Session["p"].ToString());

if (p>= ds.Tables["q"].Rows.Count -1)

{

ClsMain.CreateMessageAlert(this, "Last Record.", "123");

return;

}

p=p+1;

Session["p"] = p;

if (ds.Tables["q"].Rows.Count > 0)

{

display(ds, p,cn);

}

}

protected void BtnEnd\_Click(object sender, EventArgs e)

{

Response.Redirect("questions.aspx");

}

protected void BtnNew\_Click(object sender, EventArgs e)

{

ed(false, true);

cleartext();

Session["ae"] = "ADD";

}

protected void BtnEdit\_Click(object sender, EventArgs e)

{

ed(false, true);

Session["ae"] = "EDIT";

}

protected void BtnUpdate\_Click(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da1;

SqlCommandBuilder cb;

DataSet ds1;

int qid;

DataRow rq,ra;

if (Session["ae"].ToString() == "EDIT" )

{

DataSet ds = new DataSet();

ds = (DataSet)ViewState["ds"];

int p;

p = int.Parse(Session["p"].ToString());

if (ds.Tables["q"].Rows.Count <= 0)

{

ClsMain.CreateMessageAlert(this, "No record to edit.","123");

return;

}

qid = int.Parse(ds.Tables["q"].Rows[p]["q\_id"].ToString());

da1 = new SqlDataAdapter("select \* from question\_master where q\_id = " + qid.ToString(), cn);

cb = new SqlCommandBuilder(da1);

ds1=new DataSet();

da1.Fill(ds1, "q");

rq = ds1.Tables["q"].Rows[0];

rq["q\_text"] = TxtQst.Text;

rq["q\_type"] = ddltype.Text;

da1.Update(ds1, "q");

}

else

{

int examid;

SqlCommand com = new SqlCommand("select max(q\_id) from question\_master", cn);

qid = int.Parse(com.ExecuteScalar().ToString())+1;

com = new SqlCommand("select e\_id from exam\_master where e\_name ='" + Session["exam"].ToString() + "' ", cn);

examid = int.Parse(com.ExecuteScalar().ToString());

da1 = new SqlDataAdapter("select \* from question\_master where 1=2", cn);

cb = new SqlCommandBuilder(da1);

ds1 = new DataSet();

da1.Fill(ds1, "q");

da1.Fill(ds1, "q");

rq = ds1.Tables["q"].NewRow();

rq["q\_id"] = qid;

rq["q\_exam\_id"] = examid;

rq["q\_text"] = TxtQst.Text;

rq["q\_type"] = ddltype.Text;

rq["q\_active"] = 1;

rq["q\_create\_by"] = 1;

rq["q\_create\_date"] = DateTime.Now.ToString();

ds1.Tables["q"].Rows.Add(rq);

da1.Update(ds1, "q");

}

//delete and insert

SqlCommand com1 = new SqlCommand("delete from answer\_master where a\_qusetion\_id =" + qid.ToString(),cn);

com1.ExecuteNonQuery();

da1 = new SqlDataAdapter("select \* from answer\_master where 1=2", cn);

cb = new SqlCommandBuilder(da1);

ds1 = new DataSet();

da1.Fill(ds1, "a");

if (txtans1.Text != "")

{

ra=ds1.Tables["a"].NewRow();

ra["a\_qusetion\_id"]=qid;

ra["a\_answer"]=txtans1.Text ;

int x;

if (ddltype.Text =="SINGLE")

{

if (r1.Checked ==true )

{

x=1;

}

else

{

x=0;

}

}

else

{

if(c1.Checked ==true )

{

x=1;

}

else

{

x=0;

}

}

ra["a\_rightanswer"] = x;

ra["a\_create\_by"] = 1;

ra["a\_craete\_date"] = DateTime.Now.ToString();

ds1.Tables["a"].Rows.Add (ra);

da1.Update(ds1, "a");

}

if (txtans2.Text != "")

{

ra = ds1.Tables["a"].NewRow();

ra["a\_qusetion\_id"] = qid;

ra["a\_answer"] = txtans2.Text;

int x;

if (ddltype.Text == "SINGLE")

{

if (r2.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

else

{

if (c2.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

ra["a\_rightanswer"] = x;

ra["a\_create\_by"] = 1;

ra["a\_craete\_date"] = DateTime.Now.ToString();

ds1.Tables["a"].Rows.Add(ra);

da1.Update(ds1, "a");

}

if (txtans3.Text != "")

{

ra = ds1.Tables["a"].NewRow();

ra["a\_qusetion\_id"] = qid;

ra["a\_answer"] = txtans3.Text;

int x;

if (ddltype.Text == "SINGLE")

{

if (r3.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

else

{

if (c3.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

ra["a\_rightanswer"] = x;

ra["a\_create\_by"] = 1;

ra["a\_craete\_date"] = DateTime.Now.ToString();

ds1.Tables["a"].Rows.Add(ra);

da1.Update(ds1, "a");

}

if (txtans4.Text != "")

{

ra = ds1.Tables["a"].NewRow();

ra["a\_qusetion\_id"] = qid;

ra["a\_answer"] = txtans4.Text;

int x;

if (ddltype.Text == "SINGLE")

{

if (r4.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

else

{

if (c4.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

ra["a\_rightanswer"] = x;

ra["a\_create\_by"] = 1;

ra["a\_craete\_date"] = DateTime.Now.ToString();

ds1.Tables["a"].Rows.Add(ra);

da1.Update(ds1, "a");

}

if (txtans5.Text != "")

{

ra = ds1.Tables["a"].NewRow();

ra["a\_qusetion\_id"] = qid;

ra["a\_answer"] = txtans5.Text;

int x;

if (ddltype.Text == "SINGLE")

{

if (r5.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

else

{

if (c5.Checked == true)

{

x = 1;

}

else

{

x = 0;

}

}

ra["a\_rightanswer"] = x;

ra["a\_create\_by"] = 1;

ra["a\_craete\_date"] = DateTime.Now.ToString();

ds1.Tables["a"].Rows.Add(ra);

da1.Update(ds1, "a");

}

ClsMain.CreateMessageAlert(this, "New Changes saved.", "123");

ed(true, false);

if (Session["ae"].ToString() == "ADD")

{

string en;

en = Request.QueryString["exam"].ToString();

//query to get questions

String StrQr = "SELECT \* from question\_master where q\_exam\_id in (select e\_id from exam\_master where e\_name ='" + en + "' ) order by q\_id ";

//SqlConnection cn = new SqlConnection();

//cn.ConnectionString = ClsMain.ConnStr;

//cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataSet ds = new DataSet();

da.Fill(ds, "q");

ViewState["ds"] = ds;

int p;

p = ds.Tables["q"].Rows.Count-1;

Session["p"] = p;

if (ds.Tables["q"].Rows.Count > 0)

{

display(ds, p, cn);

}

}

}

protected void BtnReset\_Click(object sender, EventArgs e)

{

ed(true, false);

}

void ed(bool t1, bool t2)

{

BtnBack.Enabled = t1;

BtnNext.Enabled = t1;

BtnNew.Enabled = t1;

BtnEdit.Enabled = t1;

//TxtQst.Enabled = t2;

//txtans1.Enabled = t2;

//txtans2.Enabled = t2;

//txtans3.Enabled = t2;

//txtans4.Enabled = t2;

//txtans5.Enabled = t2;

//r1.Enabled = t2;

//r2.Enabled = t2;

//r3.Enabled = t2;

//r4.Enabled = t2;

//r5.Enabled = t2;

//c1.Enabled = t2;

//c2.Enabled = t2;

//c3.Enabled = t2;

//c4.Enabled = t2;

//c5.Enabled = t2;

BtnUpdate.Enabled = t2;

BtnReset.Enabled = t2;

}

void cleartext()

{

TxtQst.Text ="";

txtans1.Text = "";

txtans2.Text = "";

txtans3.Text = "";

txtans4.Text = "";

txtans5.Text = "";

r1.Checked = false;

r2.Checked = false;

r3.Checked = false;

r4.Checked = false;

r5.Checked = false;

c1.Checked = false;

c2.Checked = false;

c3.Checked = false;

c4.Checked = false;

c5.Checked = false;

}

void displayop(string ty)

{

if (ty == "SINGLE")

{

r1.Visible = true;

r2.Visible = true;

r3.Visible = true;

r4.Visible = true;

r5.Visible = true;

c1.Visible = false;

c2.Visible = false;

c3.Visible = false;

c4.Visible = false;

c5.Visible = false;

}

else

{

r1.Visible = false;

r2.Visible = false;

r3.Visible = false;

r4.Visible = false;

r5.Visible = false;

c1.Visible = true;

c2.Visible = true;

c3.Visible = true;

c4.Visible = true;

c5.Visible = true;

}

}

protected void ddltype\_SelectedIndexChanged(object sender, EventArgs e)

{

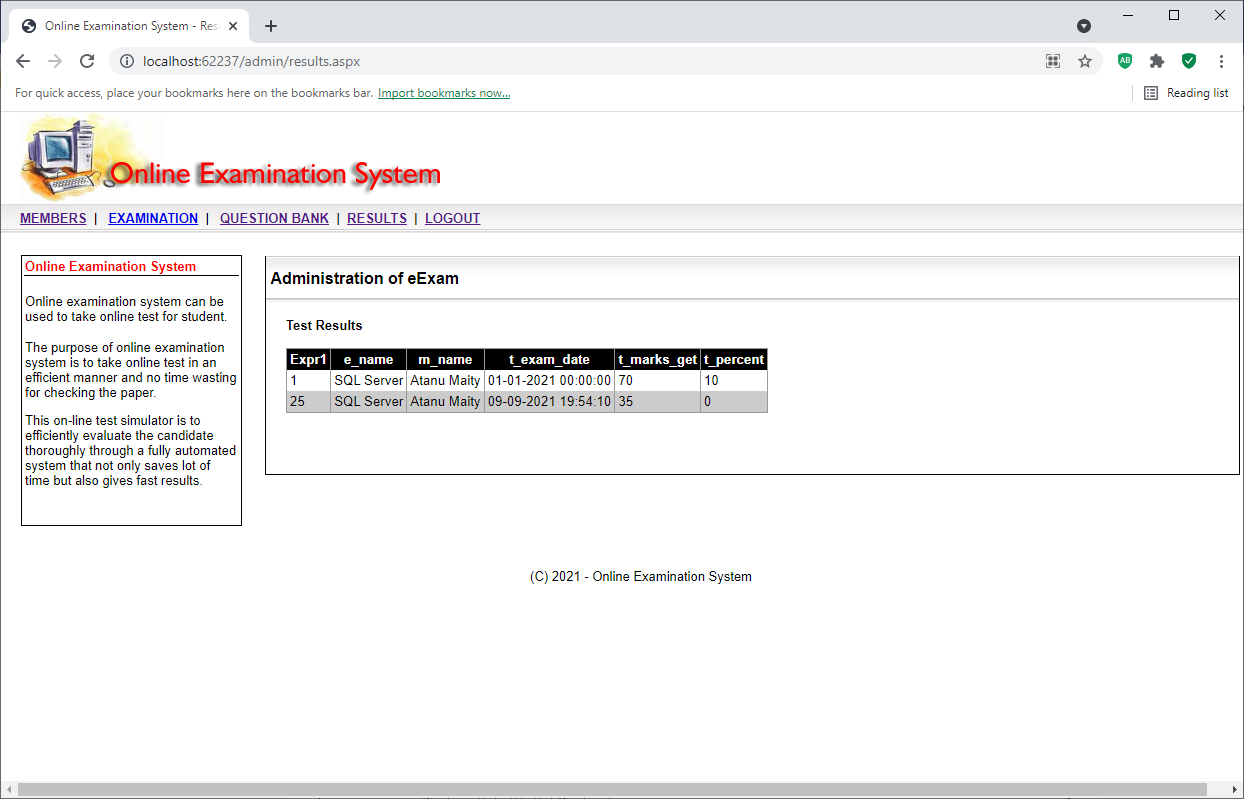
displayop(ddltype.Text);

}

}

## 8.9 eExamination – Admin View Results

**results.aspx**



**results.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class admin\_results : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

String StrQr = "SELECT test\_master.t\_id AS Expr1, exam\_master.e\_name, member\_master.m\_name, test\_master.t\_exam\_date, test\_master.t\_marks\_get, test\_master.t\_percent " +

" FROM test\_master INNER JOIN exam\_master ON test\_master.t\_exam\_id = exam\_master.e\_id INNER JOIN " +

" member\_master ON test\_master.t\_mem\_id = member\_master.m\_id ";

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataSet ds = new DataSet();

da.Fill(ds, "exam\_master");

ViewState["ds"] = ds;

GridView1.DataSource = ViewState["ds"];

GridView1.DataMember = "exam\_master";

GridView1.DataBind();

}

protected void GridView1\_PageIndexChanging1(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

GridView1.DataSource = ViewState["ds"];

GridView1.DataMember = "exam\_master";

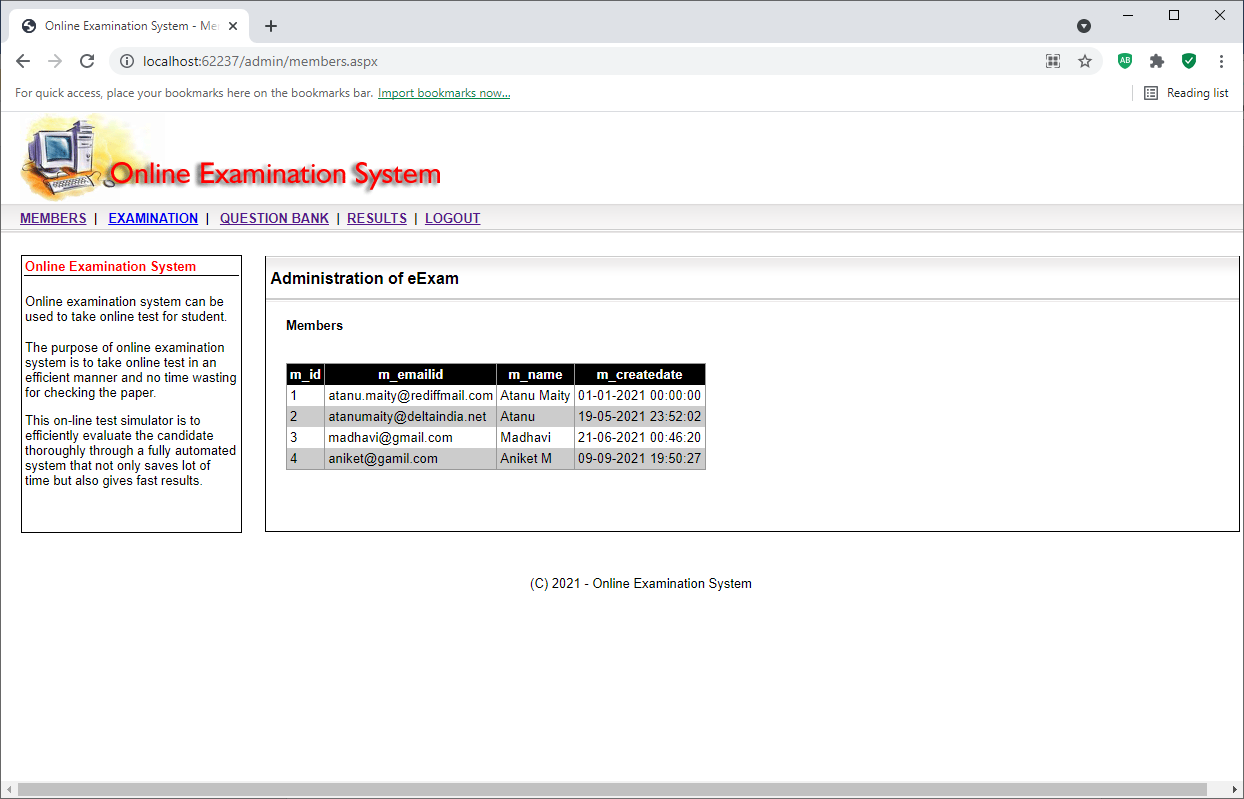
GridView1.DataBind();

}

}

## 8.10 eExamination – Admin Members

**members.aspx**



**members.aspx.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class admin\_members : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

String StrQr = "SELECT m\_id,m\_emailid,m\_name,m\_createdate from member\_master order by m\_id";

SqlConnection cn = new SqlConnection();

cn.ConnectionString = ClsMain.ConnStr;

cn.Open();

SqlDataAdapter da = new SqlDataAdapter(StrQr, cn);

DataSet ds = new DataSet();

da.Fill(ds, "member\_master");

ViewState["ds"] = ds;

GridView1.DataSource = ViewState["ds"];

GridView1.DataMember = "member\_master";

GridView1.DataBind();

}

protected void GridView1\_PageIndexChanging1(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

GridView1.DataSource = ViewState["ds"];

GridView1.DataMember = "member\_master";

GridView1.DataBind();

}

}

# 9. Test Plan

## Introduction

This document describes the user acceptance test plan for the eExamination Application for Online Examination System. The complete test strategy for the eExamination Application for Online Examination System is to perform the following kinds of tests, in sequence:

1. **Component testing** of each component that makes up the eExamination Application for Online Examination System
2. **Integration testing** of the eExamination Application for Online Examination System, to ensure the correct interworking of its components
3. **Validation testing** of the eExamination Application for Online Examination System, to ensure that it works correctly in a pseudo-live environment
4. **User acceptance testing** of the eExamination Application for Online Examination System, to ensure that its function is acceptable to its users

Acceptance testing is the last set of tests to be performed before the application goes officially live.

## 9.1 Test Scope

The scope of the user acceptance testing covers:

* Version 1 of the eExamination Application for Online Examination System
* User-facing functionality defined by a set of use cases
* Administrator-facing functionality defined by a set of use cases

The aim of the testing is to determine how well the application meets its functional requirements from the perspective of the user, and to identify any issues so they can be resolved. Also, the testing serves to compile a set of test data and results that can be used during subsequent test cycles, to test for non-regression of the software in later releases or after the application is in maintenance.

Working practices might vary from user to user and are considered outside the scope of the testing.

## 9.2 Test Strategy

The basis of user acceptance testing is that other tests were completed successfully, so the application and its required infrastructure are considered to be stable and reliable. Acceptance testing concentrates on the application from the user’s perspective, that is, how the application is used and whether it meets the necessary quality criteria.

Change requests will be sent to the development team as the actionable documentation. Change criteria will be determined by the Test team and the Development team prior to the beginning of testing. For instance, criteria may include *impact to desired functionality, amount of code impacted by proposed change,* and *design required by proposed change*. The tester will evaluate the criteria. The test lead will determine Change Required or not. Once a bug has been determined as Change Required, the bug report will be translated into a Change Request and passed on to development.

The member of the acceptance testing is the System Users, Supervisor, Manager and MIS Executive for eExamination Application for Online Examination System. The progress of the acceptance testing will be reported to the customer, together with any issues that are discovered and their planned resolutions. Sign-off of the tests, and therefore the acceptance of the application, will be performed by the customer or a selected representative.

## 9.3 Preconditions

The following items are required before testing can take place:

* A complete and coherent functional specification of the eExamination Application for Online Examination System expressed as use cases and usage scenarios
* A complete and validation-tested release of eExamination Application for Online Examination System, delivered according to the delivery plan
* An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process
* A set of test specifications describing how each functional area of the eExamination Application for Online Examination System is to be acceptance tested
* An implemented test environment for the testing
* Sufficient, suitable resources to carry out the testing
* Available standards for the acceptance testing

## 9.4 Test Priorities

During testing of the eExamination Application for Online Examination System, the following qualities will be tested in order of priority:

* Functionality—whether the required functions are available and working as expected
* Usability—how user-friendly and intuitive the eExamination Application for Online Examination System is
* Security—how well-protected and guaranteed corporate and user data is
* Performance—whether the response times are within acceptable limits
* Customization—how straightforward it is to use the application in new, unpredicted ways

## 9.5 Test Techniques

The following techniques will be applied:

* Scripted tests—sequences of user interactions (based on the use case and usage scenarios) using predefined data sets against predicted results
* Unscripted tests—based on scripted tests, the tester tries to modify the scenarios to explore what-if possibilities
* Penetration tests—scripted tests to attempt unauthorized entry into the system
* Usability checklists—tests to determine the complexity of interactions
* Performance statistics—generation of performance information to check against desired performance criteria

## 9.6 Test Organization

### Roles and Responsibilities

The following roles are defined:

* QA lead/test manager—responsible for planning and ensuring the smooth running of the test process
* Tester—carries out the tests according to the test plan, and then reports the results
* Product manager—ensures that the tests are carried out successfully from a user perspective
* Project sponsor/client—acts as main stakeholder, and ensures that the needs of the customer community as a whole are considered
* Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

## 9.7 Deliverables

The following deliverables will be expected from the user acceptance testing process:

* Test plan—this document, together with any updates that have occurred during the testing process
* Change requests—any bugs, defects, or other changes required to the eExamination Application for Online Examination System as a result of the testing process
* Weekly reports—progress reports to enable the status of the testing process to be determined
* Completion report—a report to be signed off by the customer, to signify the successful completion of the user acceptance testing

## 9.8 Test Environment

### Hardware and Software

The test environment will consist of:

**Server**

A single Intel-based computer running:

* Microsoft Windows
* eExamination Application for Online Examination System components

Client Workstations

Two Intel-based client laptop computers, each running:

* Microsoft Windows XP Professional
* Microsoft Office
* Internet Explorer 6 or greater

The following additional hardware will be required:

* One laser printer to print reports
* One color printer (laser or inkjet) to print screen dumps
* One CD-ROM drive to enable clean installation of the eExamination Application for Online Examination System
* Networking connectivity to permit interconnection of the server, clients.

## 9.9 Testing Automation Software

No testing automation software packages are selected at present.

## 9.10 Application Configuration

The following user accounts will be configured on the server:

* System Administrator
* System Users 1
* System Users 2
* Student 1
* Student 2

## 9.11 Test Management

Tests shall be managed according to the corporate test management standards, which cover:

* Conduct of tests
* Reporting of test results
* Defect tracking and resolution
* Configuration management of the test environment
* Configuration control of test deliverables.

## 9.12 Testing Schedules

The user acceptance testing schedules are shown in the project structure document and resulting Gantt charts.

## 9.13 Threats to Testing

Potential threats to the testing process are as follows:

* **Insufficient resources available for testing.** Testing resources have been seconded from the development departments, whose time is at a premium. Mitigation: ensure department heads apply a high priority to the testing of the eExamination Application for Online Examination System.
* **Availability of sales personnel for testing.** The test team should be overseen by at least one administrator.

# 10. Future Enhancement

This project was developed to fulfill user requirement; however there are lots of scope to improve the performance of the eExamination Application for Online Examination System in the area of user interface, database performance, and query processing time. Etc.

So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows.

* Linking and integration of other online educational web sites.
* Integration with university database through Web Services
* Connection to third-party OLAP applications
* In the area of data security and system security.
* Provide more online tips and help.
* To optimize the query which is embedded in the system.

# 11. Bibliography

## 11.1 Websites

Following websites are referring to create this project reports.

* <http://www.google.com>
* <http://www.microsoft.com>
* <http://www.programmer2programmer.net>
* <http://www.codeproject.com>
* <http://www.asp.net>
* <http://www.asp123.com>
* <http://www.wikipedia.org>

## 11.2 Books

Following books and ebook are used to complete this project reports.

* Mastering C# (Paperback)
* SQL Server Bible (Paperback)
* .NET Black Book (Paperback)
* Professional C#, 2nd Edition (Paperback)
* Professional ASP.NET (Paperback)
* MCAD/MCSD Self-Paced Training Kit: Developing Web Applications with Microsoft® Visual Basic® .NET and Microsoft Visual C#® .NET, Second Edition
* MCAD/MCSE/MCDBA Self-Paced Training Kit: Microsoft SQL Server
* Database Design and Implementation, Exam 70-229, Second Edition