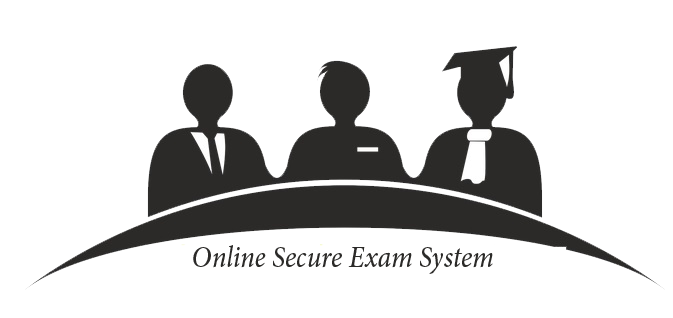


**Junior Project**

****

Yarmouk Private University Faculty of Informatics and Communication Engineering Department of Software Engineering Major

**By:**

Seif Uddean Alkowatli

Saleem Jammoul

Anas Qannas

**Supervisors**

PhD. Wassim Juneidi

Eng.Mounir Sabbagh

Semester **2016/2**

# 1.Contents

1. Content …………………………………………………………………..… 2
2. Abstract …………………………………………………………...…….…. 4
3. Chapter 1: Introduction …………...………….….………….….………….. 5
   1. What is Online Exam? ……………………………………………... 5
   2. Online Exam VS. Traditional College Programs …………………… 6
   3. Advantages & Disadvantages of OES ……………………………… 7
   4. Online Exam System Features ………………………….…………... 8
4. Chapter 2: History …………….………………………………………….. 10
   1. Background …………………….………………………………….. 10
   2. System1 …………………….……………………………………… 11
   3. System 2 …………………….……………………………………... 13
   4. System 3 …………………….……………………………………... 15
5. Chapter 3: System Architecture ….………….….………….…….………. 18
   1. Database Server …….…………….…………….………….……… 20
      1. Entity Framework ………..…………………………………….. 20
      2. Database ERD ………..………………………………………… 22
   2. Web API Server …………………………………………………… 23
      1. Why to use ASP.NET Web API ……………………………….. 23
      2. Server & Client Sides in Web API ….…………………………. 25
      3. WCF VS. ASP.NET Web API ……….………………...……… 26
   3. ASP MVC Clients…….………………………………….……...… 28
      1. What is ASP MVC ….……………………………………….… 28
      2. ASP MVC Request ……..……………………………………... 29
      3. Technologies used in ASP MVC endpoints ….……………….. 30
6. Chapter 4: System Algorithms …………………………………………... 32
   1. Question Fetching Algorithm ………………………….…………... 32
   2. Auto Correction and auto Grading.……………………………….... 34
7. Chapter 5: Security ………………………………………………………. 36
   1. Denial of Service …………………………………………………... 36
   2. SQL Injection ……………………………………………………… 37
   3. Cross Site Scripting …………………………………….………….. 39
8. Chapter 6: System Diagrams.……………………………….….…….…… 40
   1. Use-Case Diagram …………………………………………..…….. 40
   2. Specifications ……….……………………………………………... 41
   3. Sequence diagrams………………………………………………… 65
   4. Collaboration Diagrams …………………………………………… 69
9. Chapter 7: Testing ……………………………………………….……….. 72
   1. Test Suits ………………………………………………………..…. 72
10. Conclusion …………………….………………………………………….. 74
11. References …………………….………………………………………….. 75
12. Index …….………………………………….…………………………….. 76

# 2.Abstract

In this project we developed an Online Examination System That works on the intranet and a website which is portal for students and teachers and contains an admin panel. Our system is distributed system which is consist of three main servers: Database server, API server and ASP MVC which is our endpoint. we try apply basic security on our project against SQL injection, DoS attacks and XSS attacks. To make sure that all student is have fair exam and all students receive their result immediately we developed two algorithms. The first algorithm is question fetching algorithm which is work on Grundy’s Game principle in Game Theory and the second is auto correction and auto grading algorithm. At the end we build an online exam system which is a distributed system, secure and work with efficient algorithms.

# 3.Chapter 1: Introduction

# 3.1. What is Online Exam?

An exam or examination is an official test that shows your knowledge or ability in a particular subject. Exam is the word most commonly used. Examination is more formal and is used mainly in written English.

**There is two type of exams:**

* Traditional exams programs which including oral exam and open book exams
* The new born idea of online exam systems

**What is online examination?**

Online examination is a new technique to conduct an examination through internet to measure the knowledge of the participants on a given topic.

With online examination students can do the exam online, in their own time and with their own device, regardless where they life. You online need a browser and internet connection.

**How online examination system (OES) works?**

The teacher or course builder creates an account with an exam builder. In such an exam system you can create questions and add them to the exam. You can choose between multiple choice questions or free text questions. The students are provided with a link to the online exam, they sign up and can take the exam. They see the results immediately afterwards

**Who uses an online examination system?**

Everybody who needs an exam to be taken by a group of students. Our customers range from schools and teachers to companies.

**What are typical features of an online examination system?**

Obviously it must be possible to make an exam. And create questions. Preferably the exam provides for the option to create a question bank. And you have to be able to set the rules when a student passes or fails the exam. Those features are just the basics.

# 3.2. OES VS. Traditional College Programs

One of the most attractive features of online programs is flexibility. Busy professionals are flocking to online programs because these allow students to review lectures and complete assignments in their spare time. In a traditional classroom setting, you will be expected to attend class when the class meets, and attendance usually comprises part of your grade. While this kind of flexibility in online education allows students to attend classes who otherwise would not due to career and family obligations, this can be a double-edged sword. It means that you will have to be the one to set your schedule and stick to it.

Another major difference between traditional and online college programs is feedback. Instructors for online courses typically offer a higher level of feedback on assignments and papers. Unlike traditional courses where the professor might provide a comment or two, online feedback can be an extensive written critique or even a video clip. The overall level of feedback for traditional and online courses is similar, however, because in traditional classes feedback is more frequent.

The number and types of resources available to you can differ widely for both traditional and online schools. For an online program, access to course materials is generally better as there are no library resources to share with other students in the class. If you enroll in an online program that is part of a traditional university, then you can typically have access to the school’s facilities such as the university library, recreation facilities, and healthcare services. Colleges that are fully online are greatly expanding the resources offered to their students, but it is worthwhile to check what is available before choosing a particular program.

One of the biggest concerns that students have about online education is interaction with other students. Students actually have a huge range of choices in this area. Universities are opening satellite facilities across the nation and around the world where online students can meet and participate in events. Also, there are programs where students meet infrequently, usually once a month, for guest lectures and opportunities to socialize as well as work on group projects. In a fully online program, how much interaction you have with others is up to you. Just like in a traditional classroom, you’ll need to reach out to others and find common ground to build friendships.

Whether you are searching only for online programs or deciding between an online or traditional one, make a list of the qualities that you need from a program. This will help you find the program that best suits your needs and has the resources, scheduling and interaction with instructors and students that will give you the total college experience.

# 3.3. Advantages & Disadvantages of OES

# –––

**3.3.1. Advantages**

* **Save papers:** you never have to print an exam for your students and hand them out. Saves paper. Saves trees. Everybody happy.
* **Save time:** you can setup an exam in such a way that it will auto-grade itself. If you only use multiple choice questions you never have to check an exam again. The online exam system will take care of that hassle. Completely automated. And after the exam students get their result instantly.
* **Save you money:** no need to buy any paper.
* **OES is more reliable:** OES can have a huge question bank so and an algorithm which randomly chose exam questions. So it so rare that two students will share the same the question at the same time.

**3.3.2. Disadvantages**

* **Attendance:** students have to come to the classes for the examination, which means not solving the problem of student attendance.
* **Time management issues:** Because of a limited time for each question, it does not give the student a chance to answer. He may leave a specific question for a later time and solve a problem that is less complicated or more complicated and can be solved more quickly.
* **Labs Cost:** the high cost of hardware needed for our system.
* **High Cost of Fully Recovering System:** in case of failure there is a high maintenance cost.

# 3.4. Online Exam Features

With the development of the Internet technology, online examination has become more and more popular since it helps people save much energy and time. As an efficient and effective way of teaching and learning, online examination can prevent from cheating and ensure the fairness of the examination results. A good examination system should be developed based on the needs of a real examination.

Here are some features an examination system should provide.

1. **Examination Instructions**

The examination instructions will show the basis contents of the examination and something that should be paid attention to. Usually the following elements such as total questions, full score, passing rate, passing score, time limit, etc. are included in the instructions page.

1. **Accessibility of Examination Content**

Preset accounts and passwords can be set to protect the contents. And participants can register themselves online. Only authorized users can take the assessment contents. What’s more, domain hosting limitations can be set. Even if the participants have downloaded the flash assessment, it cannot be played.

1. **Valid Time of Examination Content**

The participants can log in the examination system with their accounts and passwords to access the relevant page to complete the testing in the valid time. The examination will be unavailable when the valid time expires.

1. **Time Limit of the Examination**

Participants may log in the system to take the examination after the start time. When the testing time is used up, the examination results will be submitted by participants. Even if the examination is not completely finished at that moment, it will be submitted to the reporting system too. The start time and end time will be recorded precisely in seconds.

1. **Time Reminder for the Examination**

The time reminder will tell the participants how much time left during their examination. When it comes to the end of the testing, some alter audio will remind you of submitting your examination paper.

1. **Result Release**

Immediate release or timed release can be set to the examination. For some examinations consisting of objective questions, the scores can be calculated automatically when the testing is over. The participants can view the results instantly. If you don’t want the results to be viewed so early or there are some subjective questions needs manual grading, you can finish the grading job first and then the participants can visit the system website to check their testing results.

**Statistic shows that there is over 6 million student studying online and doing cyber exams and also there is over 100 university and virtual university ranked and recognized with good well secured and reliable online exam and programs.**

**What should we use to build our exam system??**

There is a lot of available. We decided to make it distributed system and to do so we used the following technologies.

ASP Web API, ASP MVC, Entity Framework.

# 4. Chapter 2: History

# –––

**4.1. Background:**

Early adopters include the University of Cambridge Local Examinations Syndicate, (which operates under the brand name Cambridge Assessment) which conducted its first major test of e-marking in November 2000. Cambridge Assessment has conducted extensive research into e-marking and e-assessment. The syndicate has published a series of papers, including research specific to e-marking such as: Examining the impact of moving to on-screen marking on concurrent validity.

In 2007, the International Baccalaureate implemented e-marking. In 2012, 66% of nearly 16 million exam scripts were "e-marked" in the United Kingdom. **Ofqual** reports that in 2015, all key stage 2 tests in the United Kingdom will be marked onscreen.

In 2014, the Scottish Qualifications Authority (SQA) announced that most of the National 5 question papers would be e-marked.

In June 2015, the Odisha state government in India announced that it planned to use e-marking for all Plus II papers from 2016.

Here are Three Examples of Online Examination Systems.

# 4.2. System 1

# –––

Information System and Technology Department, Sur University College, Jordan 2013  
by Mohammad a Sarrayrih, Mohammed Ilyas

**Abstract**  
In this paper, we propose a system that provides security to improve online examination by utilizing technologies such as biometric authentication, internet-firewall, cryptography, network protocol and object oriented paradigms. Furthermore, we propose a framework for conducting online exams through insecure internet backbone. However, the proposed system will provide a secure communication based cryptography and group communications. In our research paper, we discuss the performance of student’s online course exam with respect to security and main challenges faced by online course exams within the university. We conclude that by improving the security system using biometrics face recognition that can be incorporated into the proposed system to fulfill the  
challenge of online exam.

**Literature Review:**

Most modern online education uses Web-based commercial courses management software such as Web CT, blackboard or software developed in-house. This software is not used widely for online exams, due to security vulnerabilities, and the system must rely on students’ honesty or their having an honor code. In this research, we try to bring out the challenges and some best solutions that may solve the problems. This paper considers the Challenge of personal identity and unauthorized invention of other users in the network using other clients Solutions for the above challenge

* **Challenge of personal identity:**

The special cameras of 360o and finger print recognition device will be incorporated for identifying the identity. The camera and the finger print device will be placed at one location in each lab. The biometric scan devices (finger print scanner and camera 360o) will check the students from the data base which is collected and stored in the registration department. The 360o camera is used for dual purpose of identifying and controlling of examination hall activities. Thus, we are utilizing the same resource for identifying the students.

* **Unauthorized interference of other users in the network using other clients**

To solve this challenge of students entering from different IPs into the domain and attempting the exam for their fellow students, we propose a system, where we create a domain with the set of student’s user id’s allocated by the university domain and each instructor will add all the student’s user id’s of his course; then he will give them the specific permissions like read and write for the specific time of that particular course exam. The students who enter from the different IP’s cannot use the allocated domain and thus the system is secure

The systems are connected using the star topology. The camera and finger print scanner inside the lab are connected to the security server; once the security server authenticates the biometrics of user, then the users are allowed to write the exam at the specific terminal provided to them. When an unauthorized user attempts to access the system from different location he is not allowed.

**Conclusion:**We believe the online format is considerably superior to paper-and-pencil exams for our courses. We have come to the conclusion that the above mentioned challenges can be solved by introducing the following security systems. Using biometrics, we overcome the traditional way of checking the ID cards of the students after they start the exam. Biometrics will identify the student as he enters the exam hall. The IP address check allows as follows:

* Using online signature or displaying student photo
* Using fingerprint
* We can provide more security to identify the students by using online cameras which are more useful than the traditional method of checking the ID cards.

Since we check the identity before the start of the exam, there are some more security problems regarding the questions and answers which are for a  
further research. This type of online exam system reduces the examination work. The future scope of this research can be the security of online remote exam systems.

# 4.3. System 2

# –––

**Research and Development of Online Examination System  
Department of Electrical and Mechanical Engineering & Department of Computer  
North China Institute of Science and Technology  
Beijing, China 2012  
by Zhao Qiao-fang Li Yong-fei**

**Abstract**Examination System was necessary to separate teaching and testing. A web-based Examination System was developed with Java Web technologies. The system provided the functions, including question management, paper generation and test online. Also the combination of client-side programming and server-side programming techniques were used and analyzed.

**Model 1 in JSP**

Browser/Server model was an important network application development model. It was a special kind of Client/Server model, which used standard Web browser as Client-side, and Web Server as the Server-side. In Browser/Server model, the main business logic of program was implemented on the server side. Such application called Web application, had the advantages of good reusability and Easy maintenance. Our Examination System was developed with JSP, and runs on the campus network. It was called online examination system. JSP had become the main technology to create Web applications, because it was easy to master and could achieve rapid development. There were two types of building models JSP developing, respectively called Model 1 and Model 2. It was easily to combine with business logic, the server-side process and HTML in the JSP page. Therefore, it could be implemented to place display, business logic and process control in a JSP page, which result in rapid development of application. There was a large number of small-scale Web applications constituted by a group of JSP pages. The JSP-centric development model was called JSP.

**Architecture of Online Examination System**

was the system structure of online examination system, which was designed based on Model 1. In the system, Web browser was used as client, JSP Engine as the business logic tier to achieve its function, and database system as the data layer.

* Client: Client was Web Browser, which implemented the system's display logic. The function was to send request to Web Server (including JSP Engine) through the Web browser by users (teachers or students). While Web Server return the requested HTML pages or HTML pages dynamically generated by JSP page to the client, which were shown in the Web browser.
* Business Logic Tier: Business logic tier was achieved mainly by JSP and JavaBean running in the JSP Engine. It responded to client requests and achieved the business logic with the Web Server. Tomcat, an open source software, was used as the JSP Engine and Web Server.
* Data Tier: Data tier was realized with database system, used to store the business data such as questions and papers and control data such as user data. MS ACCESS was used to achieve the data tier. The JSP development model based on Model 1 is very suitable for quick and small scale application development.

**Function Module of Online Examination System**

was the function module diagram of online examination system? There were three modules in the system, including question management, paper generation and test online.

**Functions of Online Examination System**

* Question Management: The functions of this module were querying, adding, deleting, and modifying the questions.
* Paper Generating
* Test Online

# 4.4. System 3

# –––

**E- Exams System for Nigerian Universities  
with Emphasis on Security and Result Integrity  
Olawale Adebayo M.Sc.  
Shafi’i Muhammad Abdulhamid M.Sc.  
Department of Cyber and Security Science, Federal University of Technology Minna,  
Niger State, Nigeria 2008.**

**Introduction**Electronic examination has been highly interested and suitable in both educational and pedagogical aspects. Examination is one of the best methods of evaluating the knowledge and ability of an individual. To this end, various methods has been employed in examining the ability of an individual, starting from manual means of using paper and pencil to electronic, from oral to written, practical to theoretical and many others. The present information technology means of examining students is the use of electronic systems in place of manual or paper method which was characterized by massive examination leakages, impersonations, demand for gratification by teachers, bribe-taking by supervisors and invigilators of examinations.

**Architecture of the Existing System**

Ayo et al (2007) and Akinsanmi (2010) presented a 3-tier architecture comprising the presentation tier, the logic tier and the database tier. The presentation tier offers an interface to the user, the logic tier serves as the middleware that is responsible for processing the user’s requests, while the database tier serves as the repository of a pool of thousands of questions. It also consists of other modules for authentication (using User name/Registration Number and Password) and computing results. This is the architecture used by all the e-exams centers visited within Nigeria and it is also the same architecture that was used even in other countries with just little modifications. This type of architecture did not give security issues too much attention and impersonation is very likely.

**Method of Preparing the Questions**

The first step in preparing the e-examination questions is to ask the lecturer in-charge of the course to submit the questions to the administrator at the center via the faculty/school exams officer some days before the commencement of the actual exams. The second step is for the administrator (mostly private operator) to enter the pool of questions into the database. The last step is to set the timing for the exams. The implication here is that, when examination questions passes through so many hands it is likely that the questions may leak, especially when a private individual is involved.

**E-Exams Result Presentation/Checking**

In most of the centers visited in this research work, students don’t get to see their results immediately after the exams. In some cases, the results may take weeks or even months before it is made available to the students. This violates one of the main essence of introducing e-exams (instant access to results). This may give room for alteration of students result. There is also no room for the users to see the correction of their tests if they so wish.

**Methodology**

Six Universities that have been engaging electronic examination were participated in this study across the country, where twenty students (15 males, female) were selected from each University for the interview and questionnaire purposes. Also five Lecturers were selected from each University for the interview on the impacts of electronic examination on their students’ performance. University of Ilorin, University of Lagos, University of Nigeria Nuka, Covenant University Ota, Nigeria Open University of Nigeria and Federal University of Technology Minna.

**Challenges of the existing system**

* **Security**  
  Both existing biometric and non-biometric e-Examination system involved sending examination questions to the e-Exam center from the department, where operator will then enter the questions into the system. The biometric system consists of picture box and fingerprint scanner that collect the biometric data of the candidates. But due to the transferring of the question involved, the security of the system is at risk and there is a need to take care of this by designing intranet and send the question through the internet in encrypting language while the questions will be decrypted at the opening of the questions to be answered by the candidates. With the se operators at the center will not be able to interfere with the questions but just to take care of candidate complains.
* **Human interference**

As long as human being is monitoring the e-Exams; it will certainly be influenced by the invigilator. There should be a data capturing and monitoring machine that can revealed the activities of the examinee during the Examination. Another area where human can interfere is the delay of the Exam result where examinee has to wait for more days to collect his results. The examinee should be able to check their result immediately after the e-Exam so as to prevent the human manipulation of result of whatever kind. Inadequate training for the students and Staff Many candidates engaging the e-Exam do not understand the proper usage of computer system talk less of the system software. There should be adequate training and awareness for the students prior to the period of e-Examination. So also the staff should be well informed the issues concerning e-Examination.

* **Complexity of Software**  
  The software being used in most of the e-center is a little bit cumbersome. The interface of the software to be used should be highly friendly to increase the effectiveness of the system.

# 5. Chapter 3: System Architecture

To make the distribution of our server possible we used the Client-Server Model which is a distributed application structure that partitions tasks or workloads between the providers of a resource or services (Server) and service requesters (Clients). Often Clients and Servers communicate over a computer network on separate hardware.



**Figure 1: Client Server Model**

**Servers:**

* **Database Server**

The Database server connected to three Databases

1. **Question Bank DB**: Contains all questions of the system and all information about Exams and implemented in Oracle.
2. **Users DB**: Contains information about students and teacher and implemented in MS-SQL.
3. **Admin DB**: Contains information about admins and implemented in MySQL.

The Database server is handles all CRUD operation on all three Databases.

And to do the CRUD operation the server use Entity Framework Technology

* **API Server:**

The API Server Contains all the functionalities of our system and handle all client’s requests. The API server use ASP Web API technology.

* **Clients:**
  + ASP MVC Clients:

In this case the ASP server is a client to our API server. there are two types of those servers

1. The Examination Website server: Which is a website on the Intranet that Students Apply for the exam.
2. Student and Teacher Portal: this website allow student to browse their previous exams and show a detailed information about each exam, and also allow teachers to add, delete or edit questions of the question bank.

This a simple Diagram of how our servers and Clients Work together.

DB-Server

API

Server

ASP.NET

Server

Client

Client

Client

MVC Clients

**Figure 2: Online Exam Architecture**

# 5.1. Database Server

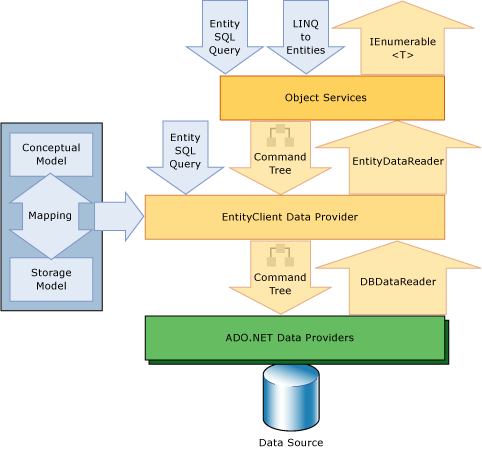
The Database server hold the three databases and preform all CRUD operations on all three databases to use perform this operation we used the Entity Framework.

**5.1.1. Entity Framework**

Entity Framework is data access technology for applications in .NET. It is an object relational mapper that enables .NET developers to work with relational data using domain specific objects. It eliminates the need for the most of the data-access code that developers usually need to write.

**How Entity Framework works?**

The Entity Framework includes the EntityClient data provider. This provider manages connections, translates entity queries into data source-specific queries, and returns a data reader that the Entity Framework uses to materialize entity data into objects. When object materialization is not required, the EntityClient provider can also be used like a standard ADO.NET data provided by enabling applications to execute Entity SQL queries and consume the returned read-only data reader.



**Figure 3: Architecture of Entity Framework**

**Why Entity Framework?**

Entity Framework is an ORM and ORMs are aimed to increase the developer’s productivity by reducing the redundant task of persisting the data used in the applications.

• Entity Framework can generate the necessary database commands for reading or writing data in the database and execute them for you.

• If you're querying, you can express your queries against your domain objects using LINQ to entities.

• Entity Framework will execute the relevant query in the database and then materialize results into instances of your domain objects for you to work within your app.

-There are other ORMs in the marketplace such as NHibernate and LLBLGen Pro. Most ORMs typically map domain types directly to the database schema.



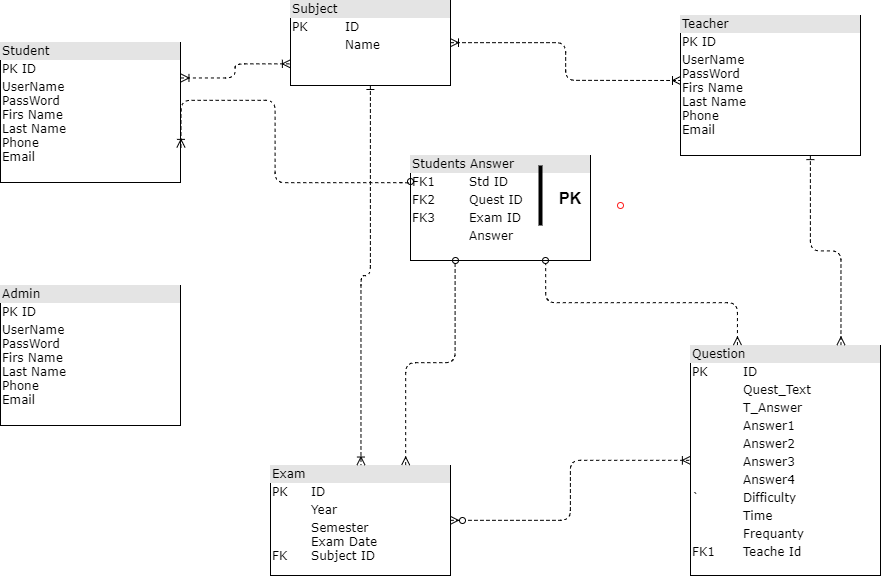
**Figure 4: ORM in Entity Framework**

Entity Framework has a more granular mapping layer so you can customize mappings, for example, by mapping the single entity to multiple database tables or even multiple entities to a single table.



**Figure 5:The Mechanism of Entity Framework**

**5.1.2. Database ERD**



**Figure 6: Database ERD**

# 5.2. API Server

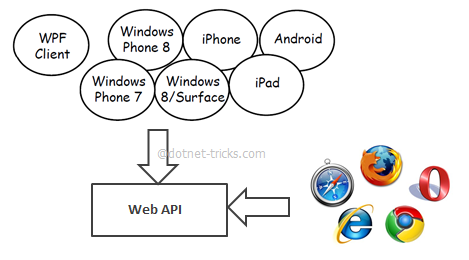
A **Web API** is an application programming interface (API) for either a web server or a web browser, usually limited to a web application's client-side (including any web frameworks being used). Asp.Net Web API is a framework for building HTTP services that can be consumed by a broad range of clients including browsers, mobiles, iPhone and tablets. It is very similar to ASP.NET MVC since it contains the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection. But it is not a part of the MVC Framework. It is a part of the core ASP.NET platform and can be used with MVC and other types of Web applications like Asp.Net Web Forms. It can also be used as a stand-alone Web services application.

**5.2.1. Why Asp.Net Web API (Web API)?**

Today, a web-based application is not enough to reach its customers. People are very smart, they are using iPhone, mobile, tablets etc. devices in its daily life. These devices also have a lot of apps for making the life easy. Actually, we are moving from the web towards apps world.

So, if you like to expose your service data to the browsers and as well as all these modern devices apps in fast and simple way, you should have an API which is compatible with browsers and all these devices.

For example, twitter, Facebook and Google API for the web application and phone apps.



**Figure 7:API Mechanism (Example)**

**Web API Features**

1. It supports convention-based CRUD Actions since it works with HTTP verbs GET, POST, PUT and DELETE.
2. Responses have an Accept header and HTTP status code.
3. Responses are formatted by Web API’s Media Type Formatter into JSON, XML or whatever format you want to add as a Media Type Formatter.
4. It may accept and generates the content which may not be object oriented like images, PDF files etc.
5. It has automatic support for OData. Hence by placing the new [Query able] attribute on a controller method that returns IQueryable, clients can use the method for OData query composition.
6. It can be hosted with in the application or on IIS.
7. It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection that makes it more simple and robust.

**Web API is the best option for our Online Exam System**:

for exposing your data and service to different devices. Moreover, Web API is open source an ideal platform for building REST-full services over the .NET Framework. Unlike WCF Rest service, it uses the full features of HTTP (like URIs, request/response headers, caching, versioning, various content formats) and you don't need to define any extra configuration settings for different devices.

**unlike WCF Rest service:**

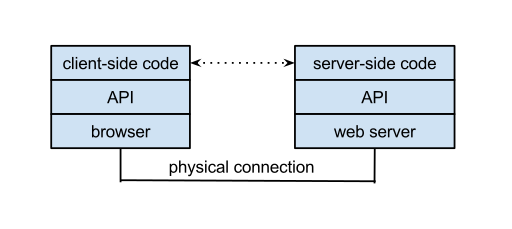
APIs available on both the client side and the server side:

makes Web programming easier allowing programmers to build web application on top of a high-level interface.

**5.2.2. Client & Server in Web API:**

* **Server side:**

A server-side web API is a programmatic interface consisting of one or more publicly exposed endpoints to a defined request–response message system, typically expressed in JSON or XML, which is exposed via the web—most commonly by means of an HTTP-based web server. Mashups are web applications which combine the use of multiple server-side web APIs. Web hooks are server-side web APIs that take as input a Uniform Resource Identifier (URI) that is designed to be used like a remote named pipe or a type of callback such that the server acts as a client to dereference the provided URI and trigger an event on another server which handles this event thus providing a type of peer-to-peer IPC. There are some PHP microframeworks such as Lumen to build a REST API.



**Figure 8:Connection Between Client and Server in Web API**

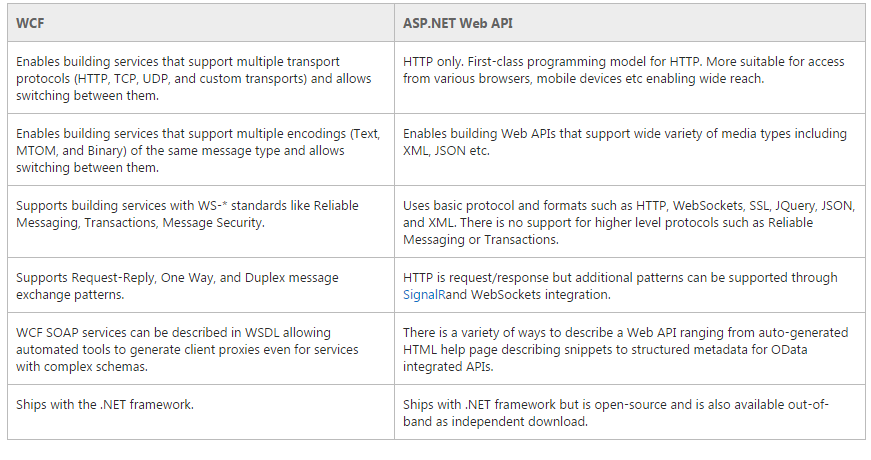
* **Client side:**

A client-side web API is a programmatic interface to extend functionality within a web browser or another HTTP client. Originally these were most commonly in the form of native plug-in browser extensions however most newer ones’ target standardized JavaScript bindings. The Mozilla Foundation created their WebAPI specification which is designed to help replace native mobile applications with HTML5 applications. Google created their Native Client architecture which is designed to help replace insecure native plug-ins with secure native sandboxed extensions and applications. They have also made this portable by employing a modified LLVM AOT compiler.

**5.2.3. ASP.NET Web VS. WCF:**

WCF is Microsoft’s unified programming model for building service-oriented applications. It enables developers to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments

The following table describes the major features of each technology:



**Figure 9:WCF VS. ASP.NET Web API**

**Why to choose Web API over WCF?**

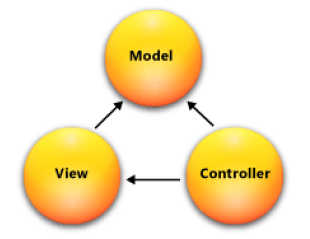
1. If we need a Web Service and don’t need SOAP, then ASP.Net Web API is best choice.
2. It is Used to build simple, non-SOAP-based HTTP Services on top of existing WCF message pipeline.
3. It doesn't have tedious and extensive configuration like WCF REST service.
4. Simple service creation with Web API. With WCF REST Services, service creation is difficult.
5. It is only based on HTTP and easy to define, expose and consume in a REST-
6. It is light weight architecture and good for devices which have limited bandwidth like smart phones.
7. It is open source

**API Server Job:**

The API server handles all HTTP requests of all clients this the types of requests that the API server handles.

1. Get Request: Retrieve data of certain object.
2. Post Request: create new instance of object.
3. Put Request: Modify the data of an object.
4. Delete Request: Delete certain object.

**5.3. ASP.NET MVC Clients**



**Figure 10: MVC Model**

**5.3.1. What is MVC?**

* Model-View-Controller (MVC).
* Standard Architectural Pattern.
* Separation of concerns: model, view, controller.

**ASP .NET MVC Framework Components**

* **Models**
  + Business/domain logic.
  + Model objects, retrieve and store model state in a persistent storage.
* **Views**
  + Display application’s UI.
  + UI created from the model data.
* Controllers
  + Handle user input and interaction.
  + Work with model.
  + Select a view for rendering UI.

There are two MVC projects in our system each one handles certain types of clients

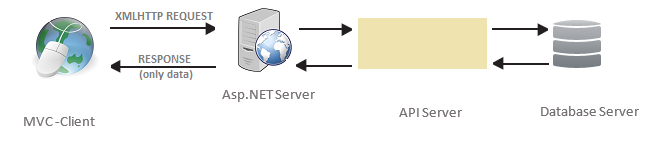
1. **Online Examination:** handles the online examination process and send requests through the intranet
2. **Online Examination Gate & Admin Panel:** consist of three main sections
3. **Students Gate:** Allow students to show their exams and requests detailed report about a certain exam.
4. **Teacher Gate:** Allow Teacher to Add, Modify or Delete question of certain subject.
5. **Admin Panel:** admin can manage accounts and create new exams.

**5.3.2. MVC Request**

To fetch data from the web API server, we must use a technique can bring life into UI.

MVC Endpoint Able to fetch Data from the API server According to the following steps:

1. Build new MVC controller that return view of the wanted Interface.
2. Add new view .in this view we can use html with JavaScript to send and fetch data from the web API server by use jQuery $. ajax that send an XMLHttpRequest (XHR) object to the server. This XHR object (which is a part of Ajax) includes data that tells the server what is being requested. The API server then responds with only the data that was requested for. When the API server responds with the data, the browser uses JavaScript also to receive the data, processes it and updates only a portion of the page that has changed. All this occurs asynchronously in the background without any page reloads; while the user continues working on the other parts of your webpage. This gives the user a more responsive and natural experience. After the web browser sends off a request to the server using the XML Http Request object, it waits for a response from the API server. When the API server responds. Also, we use jQuery callbacks to handle callbacks for success and failure:
   1. done (response, status, jqXHR) - called when the response from the server is successful
   2. fail (jqXHR, status, error) - called when the response from the server fails or the request times out.
   3. complete (response, status, jqXHR) - Always called when a response is received from the server



**Figure 11: How Ajax Request is Made**

**5.3.3. Technologies Used in MVC to call the web API:**

We used the Web Standard Model: HTML & CSS to create our pages and we used the JavaScript to add Animation and to call the API. What is all those technologies.

**HTML & CSS:**

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices.

**JavaScript:**

a high-level, dynamic, untyped, object-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. The two features of JavaScript that we used in our MVC projects are.

**jQuery:**

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: Write less, do more. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery that we used in our system:

* **Event handling:** The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
* **AJAX Support:** jQuery support AJAX technology which helped us to get updates of the Website without the need to refresh the page.
* **Animations:** The jQuery comes with plenty of built-in animation effects which you can use in your websites.

**Ajax:**

Ajax stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

* Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
* Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
* With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
* AJAX is a web browser technology independent of web server software.
* A user can continue to use the application while the client program requests information from the server in the background. Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
* AJAX is based on the following open standards:
  + Browser-based presentation using HTML and Cascading Style Sheets (CSS).
  + Data is stored in XML format and fetched from the server.
  + Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
  + JavaScript to make everything happen.

# 6.Chapter 4: Algorithms

**6.1. Questions Fetching Algorithm**

The purpose to build online exam system is to find an algorithm doing the following tasks:

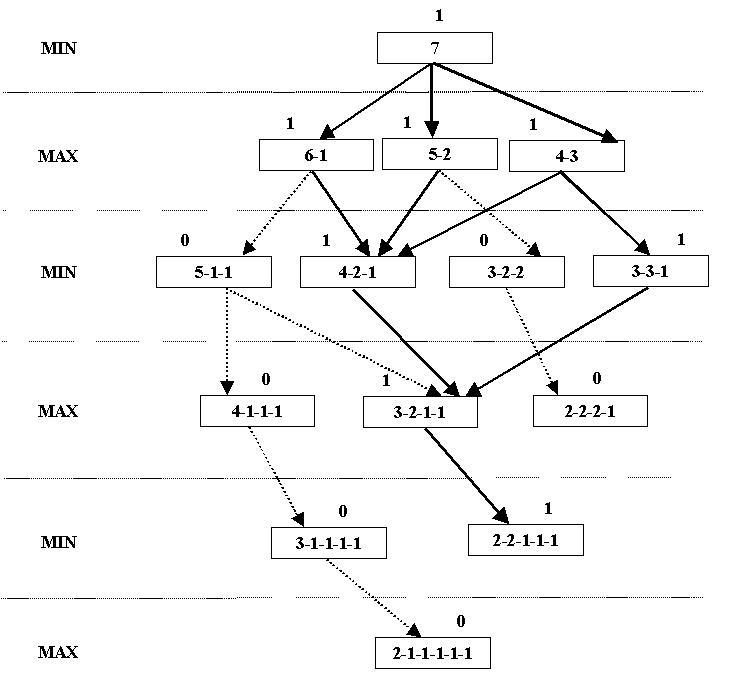
1. Degree of difficulty of the Questions pattern must be fair to all students.
2. The Questions pattern should be Different from one student to another. To ensure that we designed question algorithm using the Grundy’s Numbers.

**Grundy’s Numbers**

In 1939, Patrick Michael Grundy wrote a paper in which he defined the following game. Like in the game of NM, the game consists of stones (or matches, marbles, etcetera) distributed over several piles. A turn for one of the players consists of dividing one of these piles into two different-sized piles. For example, a pile con- siting of 7 stones can be divided into a pile of 6 and a pile of 1, or a pile of 5 and a pile of 2 or …. However, it cannot be divided into two piles of 3. Whoever cannot divide any more piles (because all piles contain 1 or 2 stones) loses the game.

**Example:** The graph-representation of Grundy's Game starting with a pile of

7 stones are as follows.



**Figure 12:Grunday's tree for 7**

**Steps of Questions Algorithm**

* when exam start, for each student the system. Randomly select degrees of difficulty from previous degrees.
* Find the next Grandy’s Numbers in the next step and chose the path that have the maximum value.
* For each number the selected stack finds Grandy’s numbers.
* Repeat Until the largest number in the stack is 2.

**Example:**

When student start the exam, the system selects randomly number 5 from this {5,4,3,2,1} and in next select the system selects from {4,3,2,1}, and next select

and each time the system selects a number it uses all its tree.

from

{3,2,1} ……

**Figure 13:Question Algorithm tree for 5**

According to the previous algorithm there are 5 question Will be submitted to the student for an answer:

1) questions with Difficulty >>>5

2) questions with Difficulty >>>4

3) questions with Difficulty>>> 1

4) questions with Difficulty >>>3

5) questions with Difficulty>>> 1

6) questions with Difficulty>>> 1

7) questions with Difficulty>>> 2

Etc.…

For each question, there is a number representing the repetition of choosing this question Each time this question is picked this number increases by one.

The system chooses the least frequent question to ensure that the questions are not repeated in the exam.

**6.2. Questions Correction**

One of the most important functionalities in our system is the auto correction and auto grading. There are two methods to do this functionality.

**First method:**

Operation of Questions correction starts at the end of the exam for each student. But after applying this method, a lot of problems appeared. And that’s because of the high demand for database connections:

1. Overlapping information.
2. System breakdown due to Accumulation of information.

**Second method:**

After each submission, Questions correction starts and save the result individually in the list, and When the exam time ends, the system calculates the final result and show the result directly to the student and store it in the database Correction operation.

we use second method in our examination system, so the process is done as follows:

1. The student enters his answer.
2. The system makes sure the answer is correct or not and stores the result.
3. At the end of time, the final result is calculated as follows let's have a table with student answers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Questions | Mark | True answer | Student answer’s | result |
| Q1… | 3 | one | one | true |
| Q2… | 1 | tow | four | false |
| Q3… | 5 | three | ten | false |
| Q4…. | 2 | four | seven | false |
| Q5…... | 5 | Five | five | true |

Figure 14:Question Correction Algorithm Example

**Result** = (marks of true answer \*100) / all marks.

Calculate the result for the previous student:

**Result** = 3+ 5 \* 100 /16 =50%

**Success Factor** = Number of Answered Questions /Total Number of Questions

Final Result = Result /Success Factor.

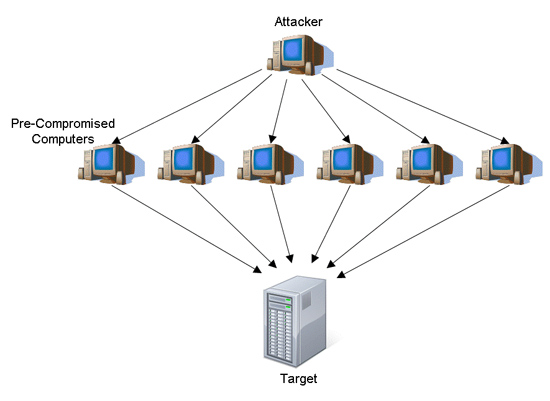
# 7.Chapter 5: Security

As online exam system one of our project special points is security. our system must be secured so that lead us to new question. What are the risks that our system may face in the future??

There are too many cyber-attacks may cause the system failure or (system down) The following attacks are: We protected our system from three types of the attack

**7.1. Denial-of-service**

a denial-of-service attack (DoS attack) is a cyber-attack where the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled. A DoS attack is analogous to a group of people crowding the entry door or gate to a shop or business, and not letting legitimate parties enter into the shop or business, disrupting normal operations.



**Figure 15: Denial of Service attack**

**Counter measures against DoS:**

The only thing you can do to prevent such an attack is to block the response to the attackers. You have no control over the requests, so you have to catch the attacker as early as possible after the request has been received by the web server.

There are two challenges to blocking the attacks

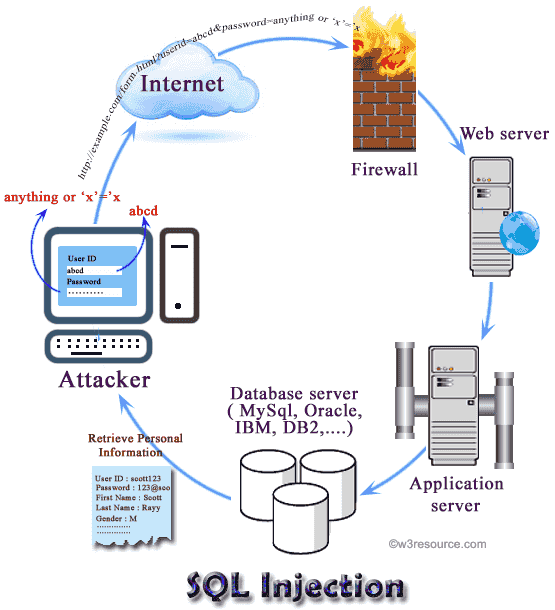
1. Identify the attackers
2. Block the response only to the attackers

To catch the request as early as possible, an HttpModule is the right place. It is executed before any page or any other handler so the impact on the server can be minimized. This HttpModule monitors all requests and block requests coming from IP addresses that make many requests in a short period of time. After a while the attacking IP address gets released from blocking.

The module is a high performance and lightweight protection from DoS attacks and very easy to implement.

**7.2. SQL Injection**

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database. SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.



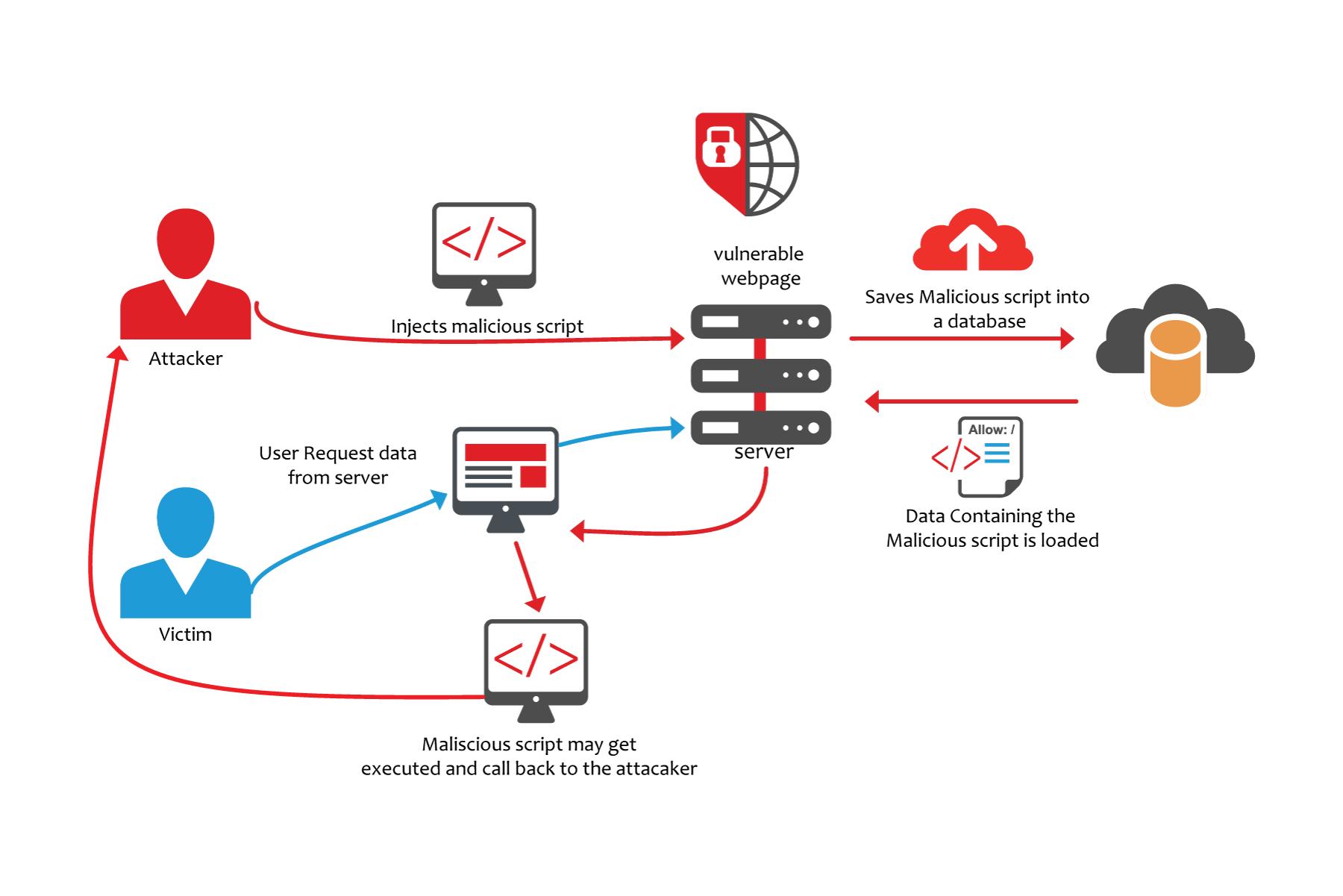
**Figure 16:SQL Injection attack**

**Counter measures against SQL-Injection:**

By using Entity Framework will issue parameterized SQL statements when an operation is executed. Using them will provide protection against SQL Injection attack without an extra effort. Using this data access mechanism can also prevent you from trouble because you can just program against the conceptual application model instead of programming directly against your database. You don’t have to deal with those typo and SQL syntax.

**7.3. Cross-site scripting**

attacks use known vulnerabilities in web-based applications, their servers, or the plug-in systems on which they rely. Exploiting one of these, attackers fold malicious content into the content being delivered from the compromise zed site. When the resulting combined content arrives at the client-side web browser, it has all been delivered from the trusted source, and thus operates under the permissions granted to that system. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access-privileges to sensitive page content, to session cookies, and to a variety of other information maintained by the browser on behalf of the user. Cross-site scripting attacks are a case of code injection



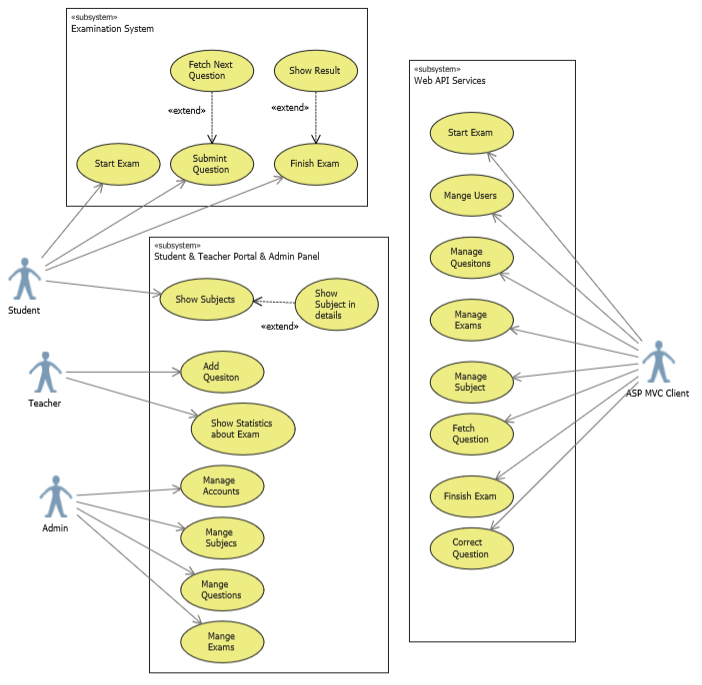
**Figure 17:XSS attack**

**Counter measures against XSS:**

to prevent XSS attacks we used AntiXSS which is an encoding library, designed and developed by CISG team at Microsoft in conjunction with the ACE Team. It is designed to help developers protect their Web-based applications from XSS attacks. This library is very different from most encoding libraries; it uses the principle-of-inclusions technique to provide protection against XSS attacks. This approach works by defining a valid or allowable set of characters, and encoding anything outside this set (invalid characters or potential attacks). It offers several advantages over other encoding schemes.

# 8. Chapter 6: System Diagrams

**8.1. Use-Case Diagram**



**Figure 18:Use-Case Diagram**

# 8.2. Use-Case Specifications

**Start the Exam (UC-A1)**

• **Actor**: Student.

•**Brief description:** This use case describes how the student can start his exam.

•**Pre-conditions:**

There is an active network connection to local server.

sign in as student.

•**Post**-**condition**:

start displays exam time

display the first question.

**Flow of events**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Click button “Start the Exam”. |  |
|  | 2.Request for new exam and get first question. |
|  | 3.Display the first question with his Expected answers. |
|  | 4.Show the total time of the exam and begins to decrease it. |
|  | 5. begins to decrease the specified time of previse question. |

Figure 19: Start Exam flow of events

**Critical scenario:**

1.Error with login information

2.Request to start the exam before the exam time so that the query algorithm is not yet started.

3.It is forbidden to apply for the exam.

**User Interface Prototype:**

**Figure 20: Start Exam Interface Prototype**

Q1) …...  
A) answer1

B) answer2

C) answer3

D) answer4

the remaining time

01:20:00

**Start the exam**

**Fetch the first Question(UC-A2)**

• **Actor:** MVC Client.

• **Brief description:** This use case describes how the system can request for Start new exam and get first question from the web API server.

• **Pre-conditions:** System request for start new exam and get question the first from web API.

**• Post-condition:** web API save exam information in the database and Apple question algorithm.

**• Flow of event:**

|  |  |
| --- | --- |
| **System** | **Web** API |
| 1. Request for new exam”. |  |
|  | 2 Select exam information |
|  | 3. Save the exam information in the database. {student\_id, subject\_id} |
|  | 4.Apply Question Algorithm to select first question. |
|  | 5.send the first question  Refer to the start of the exam and register its information. |

Figure 21: Request New Exam flow of events

**Critical scenario:**

1.The Web API server stopped.

2.No connection to Web API server

No Interface Prototype.

**Submit the answer (UC-A3)**

• **Actor**: Student.

• Brief **description**: This use case describes how the student submit his answer to get next question.

•**Pre-conditions:**

Signed in as student.

Started the exam.

Select his answer.

• **Post-condition:**

Correct the student's answer and display the next question.

**Flow of event:**

|  |  |
| --- | --- |
| Actor | System |
| 1. Click button “submit”. |  |
|  | 2.The system get student answer and request for Correct it. |
|  | 3.Requset for next question. |
|  | 4.Display the question. |

**Figure 22:Submit the answer flow of events**

**Critical scenario:**

No.

**User interface prototype:**

**Q1**) …...  
A) answer1

B) answer2

C) answer3

D) answer4

submit

**Q2**) …...  
A) answer1

B) answer2

C) answer3

D) answer4

submit

**Figure 23: Submit the answer Interface Prototype**

**Correct and fetch the next question(UC-A4)**

• **Actor:** MVC Client.

•**Brief description:** This use case describes how the system can request for next question from the web API server.

• **Pre-conditions:** System request for the next question from web API.

• **Post-condition:** Web API server correct student answer, save it in the database and get the next question.

•**Flow of event:**

|  |  |
| --- | --- |
| **System** | **Web** API |
| 1. Pass student answer with question id. |  |
|  | 2.Connet to database and get the question true answer. |
|  | 3.Apply auto correction algorithm to Correct this answer only |
|  | 4.Save the information in the database |
| 5.Request for next question. |  |
|  | 6.Randomly selects from the Student Questions list, appropriate question Depending on question algorithm. |

Figure 24:Correct and fetch the next question flow of events

**Critical scenario:**

1.The Web API server stopped.

2.No connection to Web API server.

**Finish the exam (UC-A5)**

* **Actor**: Student.
* **Brief description**: This use case describes how the student can finish his exam and get his final result.
* **Pre-conditions**:

sign in as student.

Started the exam.

Answer a number of specific questions.

* **Post-condition**:

Show the student final result.

* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Click button “finish the Exam”. |  |
|  | 2.The system confirm the request from student before execution |
|  | 3.Requset for the final result of this student. |
|  | 4.Display the result with details. |

Figure 25: Finish the exam flow of events

**Critical scenario**:

1. Error with login information.
2. Do not answer the minimum questions that allow student to stop the exam.

**User Interface Prototype**

Finish the exam

Q25) …...  
A) answer1

B) answer2

C) answer3

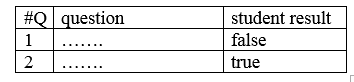
D) answer4

submit

Student username: test1

Student Result: 50%name

Details:



**Figure 26: Finish the exam Interface Prototype**

**Fetch the final Result (UC-A6)**

* **Actor**: MVC Client.
* **Brief description**: This use case describes how the system get student final result from the web API server and stop the exam.
* **Pre-conditions**: System request for the final result from web API.
* **Post-condition**: web API server end the exam and calculate the final result and save it in the database.
* **Flow of event**:

|  |  |
| --- | --- |
| **System** | **Web** API |
| 1. Request for final result”. |  |
|  | 2. Server connect to database to get student individually answers. |
|  | 3. Apply Auto correction algorithm to calculate the final result. |
|  | 4.Save the student final result in the Database. |
|  | 5.Send the final result with details to the system. |

Figure 27: Fetch the final Result flow of events

**Critical scenario**:

1. The web API server stopped.
2. No connection to web API server.

No User Interface Prototype.

**Sign in (UC-A7)**

* **Actors**: Student, Admin, Teacher.
* **Brief description**: This use case describes How to sign in.

redirect the user to their view that Available to them.

* **Pre-conditions**: no Pre-conditions.
* **Post-condition**: redirect the user to view.

.

* **Flow of event**:

|  |  |
| --- | --- |
| Actor | System |
| 1. Enter login information. |  |
| 2.Click Button “Sign in” |  |
|  | 2.Request for view that Available to this user. |
|  | 3.Rediret to the view. |

Figure 28: Sign in flow of events

**Critical scenario**:

1. Error with login information

**User interface prototype:**

**Student View**

**User Name**

or

**Password**

or

**Admin View**

s

**Sign in**

or

Teacher View

**Figure 29: Sign in Interface Prototype**

**Get Available view (UC-A8)**

* **Actor**: MVC Client.
* **Brief description**: This use case describes how the system can request for the view that Customized for each user.
* **Pre-conditions**: System request for View from web API.
* **Post-condition**: Web API return view Determined according to User login information.
* **Flow of event**:

|  |  |
| --- | --- |
| **System** | **Web API** |
| 1. Request for View”. |  |
|  | 2. Check login information |
|  | 3. According to User Role  The API select Appropriate view |
|  | 4.Return view |

Figure 30: Get Available view flow of events

**Critical scenario**:

1. Error with login information

**No User Interface Prototype**

**Show Marks (UC-A9)**

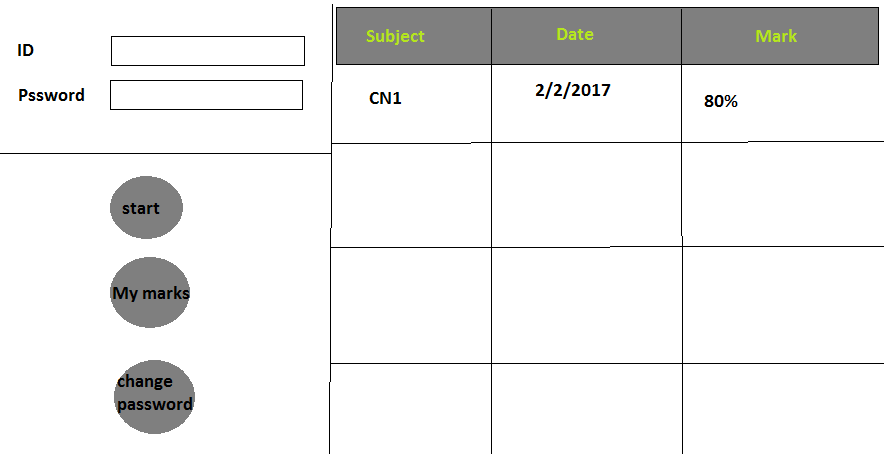
* **Actor:** Student.
* **Brief description:** student can access his records of exam results
* **Pre-condition**: Signed in as student
* **Post-condition**: student can check his detailed marks
* **Flow of events**:

|  |  |
| --- | --- |
| Actor | System |
| 1-Student sign in |  |
| 2-Student click on my mark button |  |
|  | 3-Showing mark page |
| 4-Scrolling up and down |  |

**Figure 31: Show Marks flow of events**

* **Critical scenario**:

There is no critical scenario

* **User interface prototype**   
  

**Figure 32: Show Marks Interface Prototype**

**Detailed marks (UC-A10)**

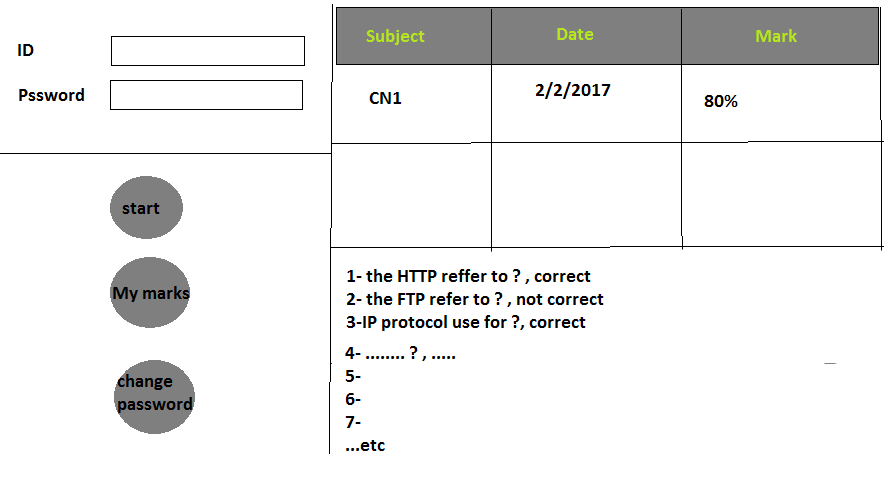
* **Actor:** Student.
* **Brief description**: student can see detailed information about subject result
* **Pre-condition**: Signed in as student and clicked my mark button
* **Post-condition**: can see which questions have right answer and which not
* **Flow of events**:

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Student sign in |  |
| 2-Student click on my mark button |  |
|  | 3-Showing mark page |
| 4-Clicking on specific subject name |  |
|  | 4-Showing new page with correct answered question for this subject |

Figure 33: Detailed marks flow of events

* **Critical scenario:**

There is no critical scenario

* **User interface prototype**   
  

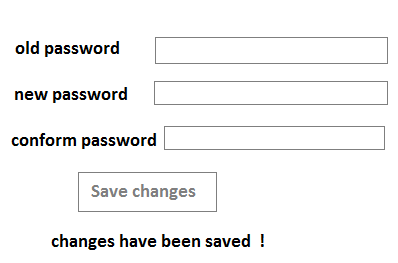
**Figure 34: Detailed marks Interface Prototype**

**Change password (UC-A11)**

* **Actor:** Student, Teacher and Admin.
* **Brief description: user** whatever his roll is can change his account password
* **Pre-condition:** signed in as teacher or student or admin
* **Post-condition:** user has his password changed
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-User sign in |  |
| 2-User click on change password button |  |
|  | 3-Changing password webpage loaded |
| 5-Entering the old password |  |
| 6-Entering the new password |  |
| 7-Conforming the new password |  |
|  | 8-New password matched and conformed |
| 9-Clicking save button |  |
|  | 10-Message said changes have been saved |

**Figure 35: Change password flow of events**

* **Critical scenario:**1- the old password is wrong   
  2- new password is used as old password  
  3-new password doesn’t match

**Figure 36: Change password Interface Prototype**

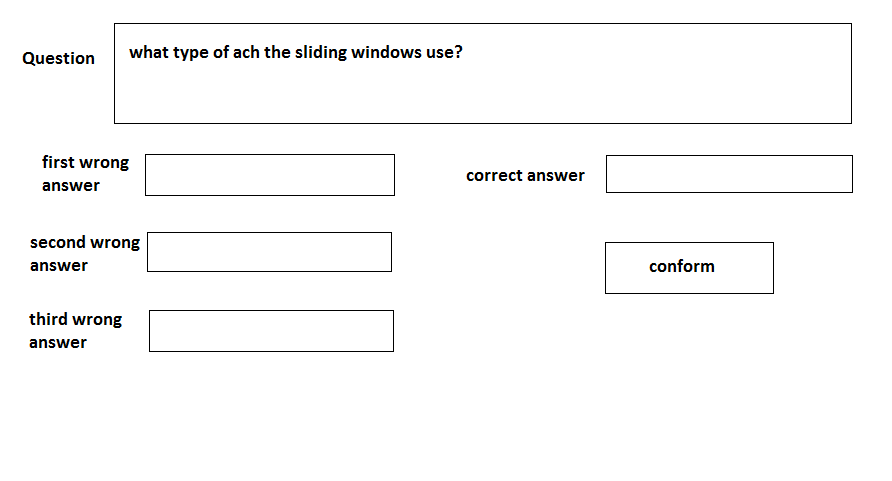
* **User interface prototype**

**Add Question (UC-A12)**

* **Actor:** Teacher.
* **Brief description: teacher** can add question to the question bank
* **Pre-condition:** signed in as teacher
* **Post-condition:** new question added and ready to be selected in exams
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Chose add button |  |
|  | 3- Adding webpage loaded |
| 4-Fill the input box with question and answers and the correct answer |  |
| 5- Conform adding by clicking the conform button |  |
|  | 6-Message said your question has been added |

**Figure 37: Add Question flow of events**

* **Critical scenario:**1-the teacher may click conform button while he is missing one input or more of the answers box  
  2- the teacher might click conform button while missing filling question input box   
  3-the teacher might miss selecting the correct answer
* **User interface prototype**

**Figure 38: Add Question Interface Prototype**

**Managing Questions (UC-A13)**

* **Actors:**  Teacher, Admin.
* **Brief description: teacher** can edit or delete a question from question bank
* **Pre-condition:** signed in as teacher or Admin
* **Post-condition:** question has changed or no more available to be picked
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Clicking edit button |  |
|  | 3-Editing page loaded |
| 4-Making changes on question or answers |  |
| 5-Save changes |  |
|  | 6-Message says your changes have been saved |
| 7-Clicking delete button |  |
| 8-Chose question ID to be deleted |  |
|  | 9-Checking question availability |
| 10-Click delete question button |  |
|  | 10-Meesage box saysare you sure want to delete the question |
| 11- Clicking yes or no |  |
|  | 12-In case yes, message says question has been deleted |
|  | 13- Case no back to main menu |

**Figure 39: Managing Questions flow of events**

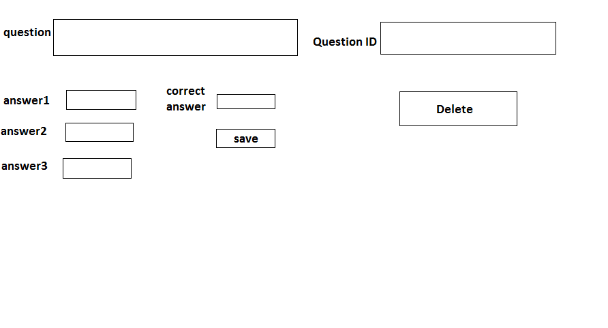
* **Critical scenario:**1- teacher may leave blank input boxes while editing   
  2- teacher may chose unavailable question ID for deleting
* **User interface prototype**

Figure 40: Managing Questions Interface Prototype

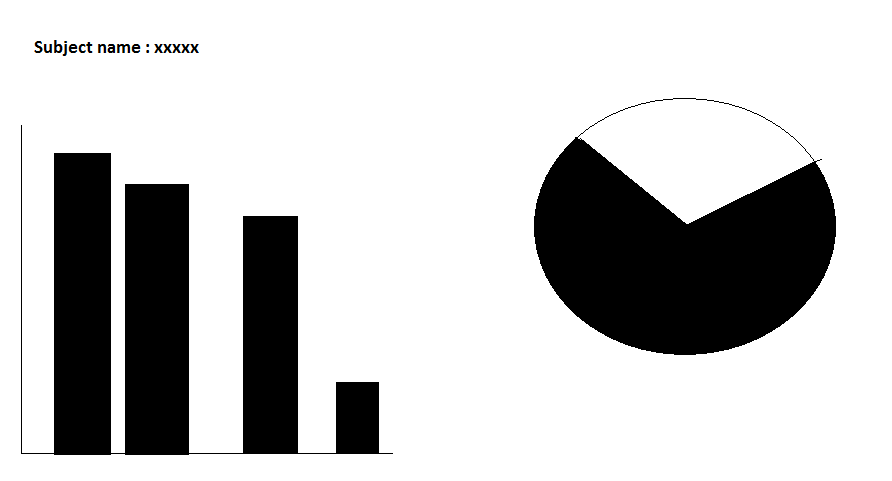
**Statistic (UC-A14)**

* **Actor:**  Teacher.
* **Brief description: graph** represent pass and failed students for the teacher  
  and histogram for average student who get specific marks
* **Pre-condition:** signed in as teacher
* **Post-condition:** showing histogram and graph
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Click statistic button |  |
|  | 4-Subject’s webpage loaded |
| 5-Chose subject |  |
|  | 6- Graph webpage loaded |

**Figure 41:Statistic flow of events**

* **Critical scenario:**there is no critical scenario
* **User interface prototype**



**Figure 42: Statistic Interface Prototype**

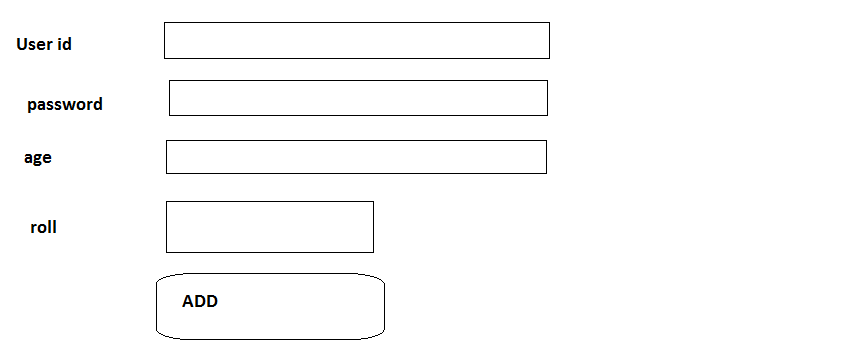
**Add Account (UC-A15)**

* **Actor:**  Admin.
* **Brief description: admin** create a new account for new teacher or student
* **Pre-condition:** signed in as admin
* **Post-condition:** new user as student or teacher can sign in into the system
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Click on adding button |  |
|  | 3-Regestering page loaded |
| 4-Fill the user name, roll and password and other inputs |  |
| 5-Click add button |  |
|  | 6-Message says your account has been added |

**Figure 43: Add Account flow of events**

* **Critical scenario:**1- admin might miss one of the inputs box empty  
  2-admin may miss click the roll and register student as teachers  
  3-admin may miss click the roll and register teacher as student
* **User interface prototype**



**Figure 44: Add Account Interface Prototype**

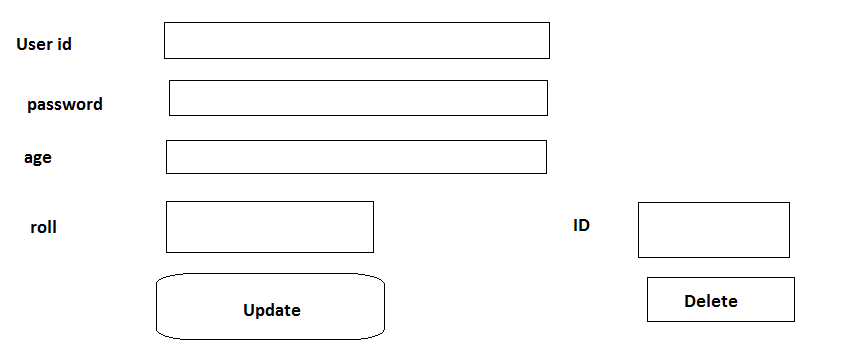
**Manage accounts (UC-A16)**

* **Actor:**  Admin.
* **Brief description: teacher** can edit or delete teacher and student accounts
* **Pre-condition:** signed in as admin
* **Post-condition:** teacher or student account has been changed or no more available
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Clicking the edit button |  |
|  | 3-Edit page loaded |
| 4-Chosing account id to edit |  |
|  | 5-Account information displayed |
| 6-Make changes |  |
| 7-Click save |  |
|  | 8-Your changes have been saved |
| 9-Clicking delete button |  |
| 10-Chose account id |  |
| 11-Click delete |  |
|  | 12-Are you sure you want to delete this account |
| 13- Clicking yes button |  |
|  | 14-Account has been deleted |

**Figure 45: Manage accounts flow of events**

* **Critical scenario:**1- admin may leave empty input boxes while updating   
  2- admin mystically remove student or teacher
* **User interface prototype**



**Figure 46: Manage accounts Interface Prototype**

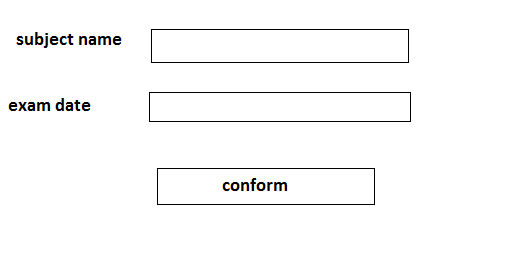
**Add Exam (UC-A17)**

* **Actor:**  Admin.
* **Brief description: admin** can add exam to the schedule
* **Pre-condition:** signed in as admin
* **Post-condition:** exam should be available at specific date for student to apply for it
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Chose creating exam button |  |
|  | 3- Subject webpage loaded |
| 4-Chose subject |  |
| 5- Set exam date |  |
| 6- Click conform button |  |
|  | 7-Message says exam day conformed |

**Figure 47: Add Exam flow of events**

* **Critical scenario:**1- admin may set exam in holiday
* **User interface prototype**



**Figure 48: Add Exam Interface Prototype**

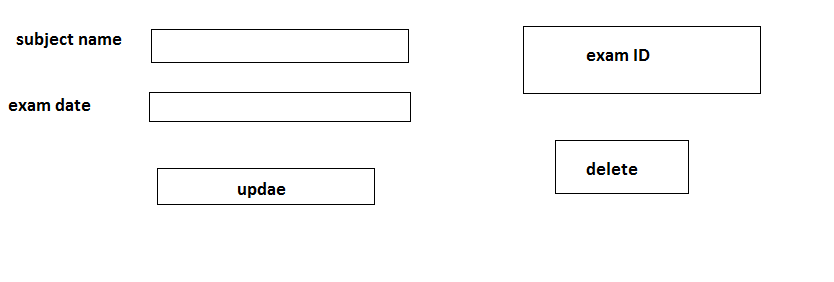
**Manage Exam (UC-A18)**

* **Actor:**  Admin.
* **Brief description:** admin can edit and delete exam
* **Pre-condition:** signed in as admin
* **Post-condition:** exam date or subject has been changed or exam has been canceled
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Click on editing button |  |
| 3-Chose exam |  |
| 4-Change date or subject |  |
| 5-Click save |  |
|  | 6-Message says changes has been saved |
| 7-Click delete button |  |
| 8-Chose exam ID |  |
| 9-Click delete button |  |
|  | 10-Message says are you sure u want to delete this exam |
| 11-Clicking yes |  |
|  | 12-Exam has been deleted |

**Figure 49:Manage Exam flow of events**

* **Critical scenario:**1- admin may change exam date to holiday
* **User interface prototype**



**Figure 50: Manage Exam Interface Prototype**

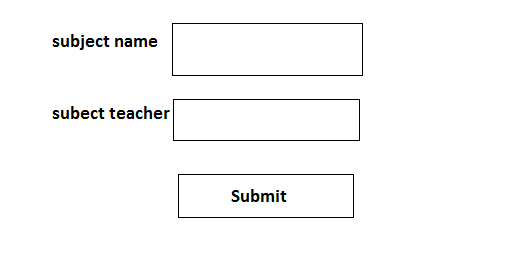
**Add Subject (UC-A19)**

* **Actor:**  Admin.
* **Brief description:** admin can add subject and set the teacher for it
* **Pre-condition:** signed in as admin
* **Post-condition:** we have new subject with supervising teacher
* **Flow of events:**

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Clicking adding subject button |  |
| 3-Set subject name and teacher |  |
| 4-Click on submitting button |  |
|  | 5-Message says your subject has been added |

**Figure 51: Add Subject flow of events**

* **Critical scenario:**1-admin may add unrelated subject to the collage specialization  
  2-admin may set teacher for subject of deferent specialization
* **User interface prototype**



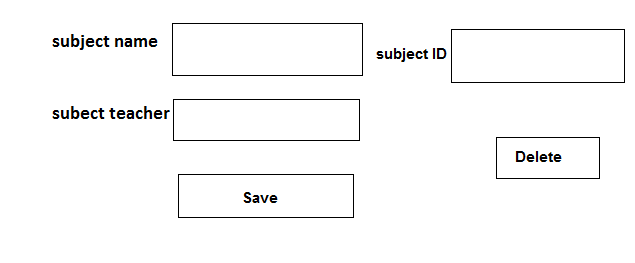
**Figure 52: Add Subject Interface Prototype**

**Manage Subjects (UC-A20)**

* **Actor:**  Admin.
* **Brief description:** admin can edit subject or delete it from the system
* **Pre-condition:** signed in as admin
* **Post-condition:** subject name or teacher changed, subject has been removed from the system
* **Flow of events:**

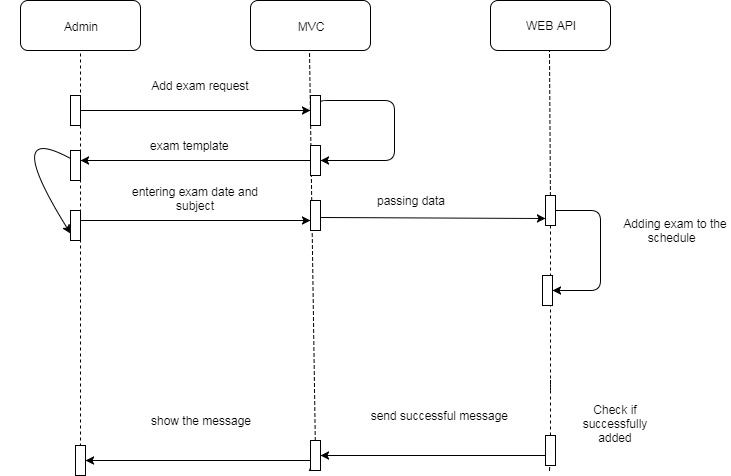
|  |  |
| --- | --- |
| **Actor** | **System** |
| 1-Sign in |  |
| 2-Click one edit subject button |  |
| 3-Chose subject id |  |
| 4-Change name or teacher |  |
| 5-Click save |  |
|  | 6-Conformation message |
| 7-Click yes |  |
|  | 8-Message say changes have been updated |
| 9-Click delete subject button |  |
| 10-Chose subject id |  |
| 11-Click delete button |  |
|  | 12-Conformation message |
| 13-Clicking yes button |  |
|  | 14-Subject has been removed |

**Figure 53: Manage Subjects flow of events**

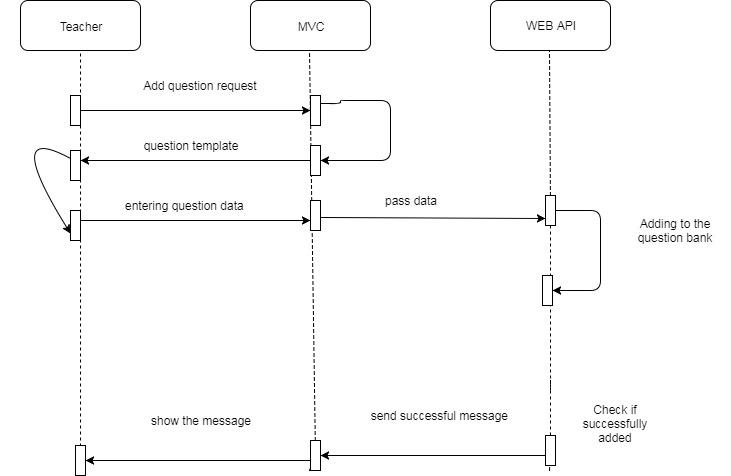
* **Critical scenario:**1-admin may update unrelated subject to the collage specialization  
  2-admin may set teacher for subject of deferent specialization
* **User interface prototype**

**Figure 54:Manage Subjects Interface Prototype**

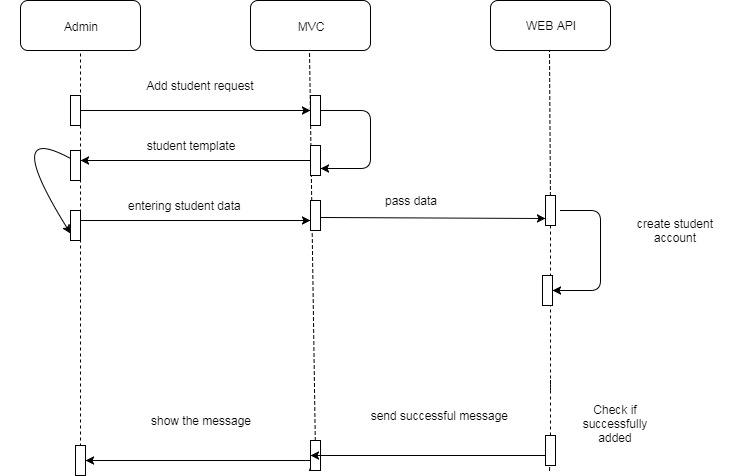
# 8.3. Sequence Diagrams



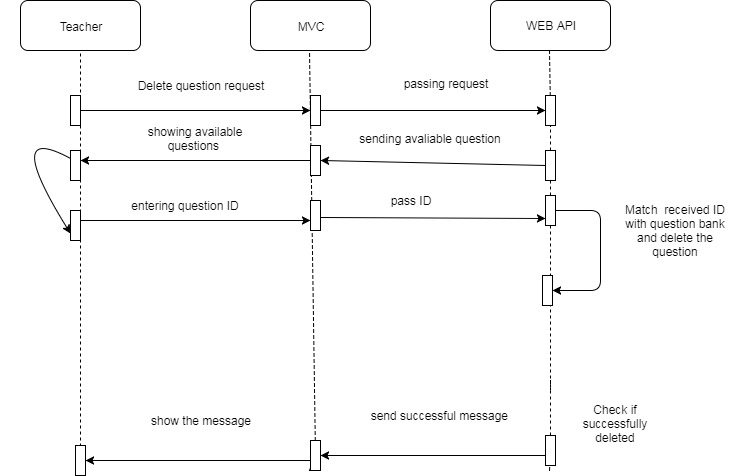
**Figure 55:Add Exam Sequence Diagram**



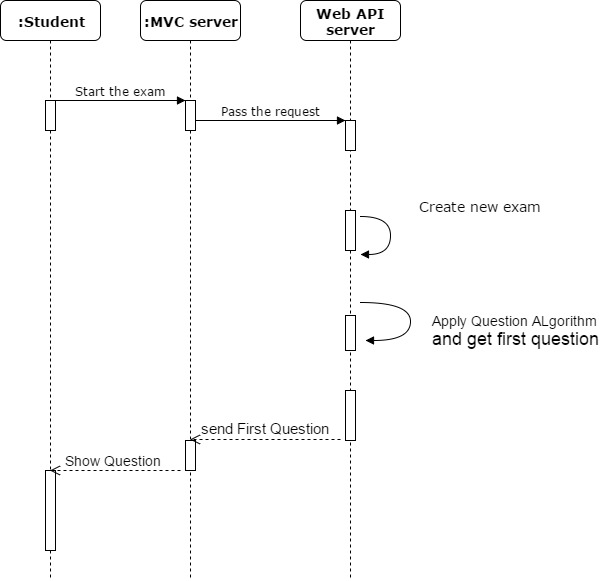
**Figure 56: Add Question Sequence Diagram**



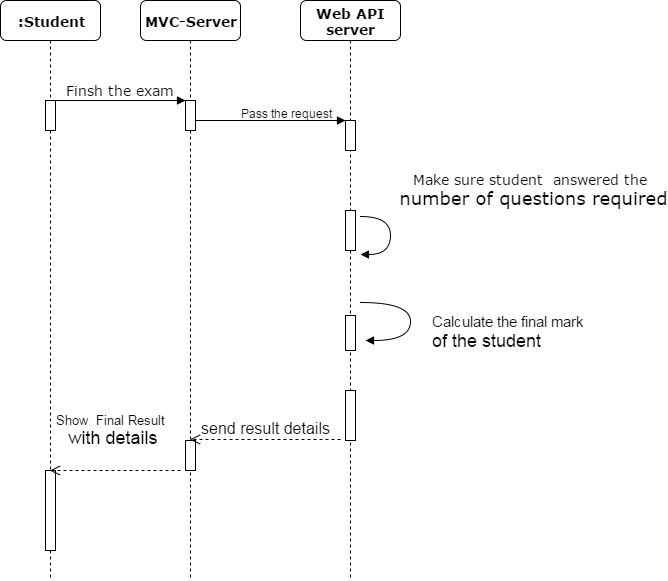
**Figure 57: Add Student Sequence Diagram**



**Figure 58: Delete Question Sequence Diagram**

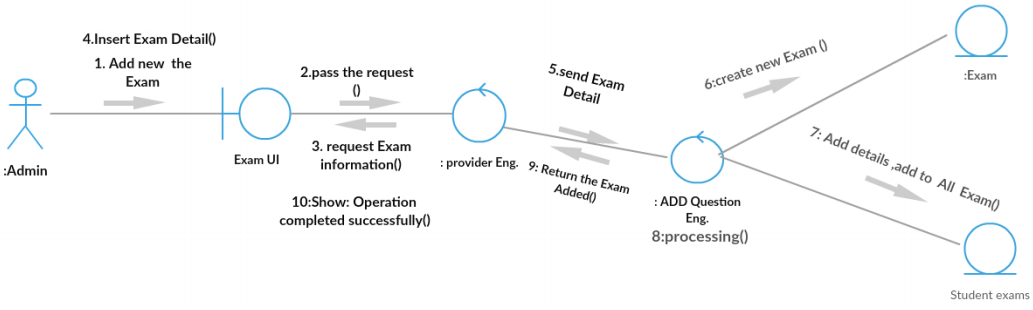


**Figure 59:Start Exam Sequence Diagram**

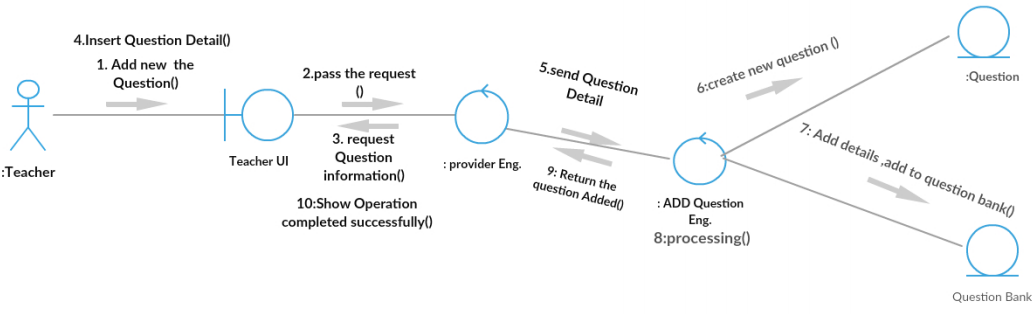


**Figure 60: Finish Exam Sequence Diagram**

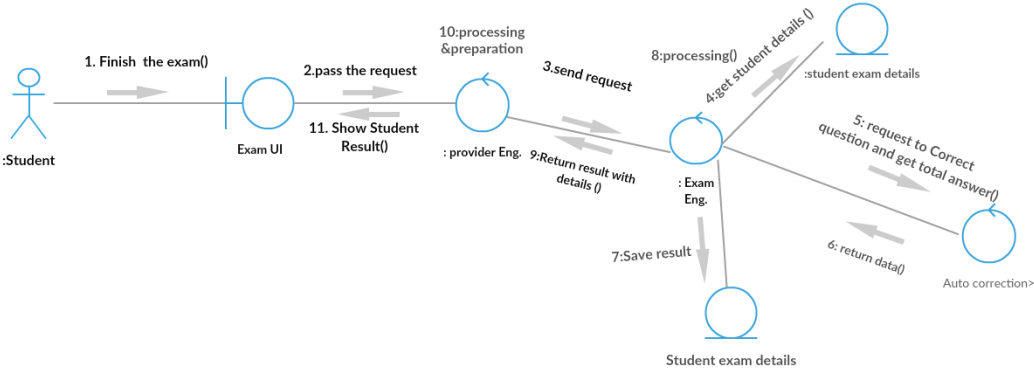
# 8.4. Collaboration Diagrams

****

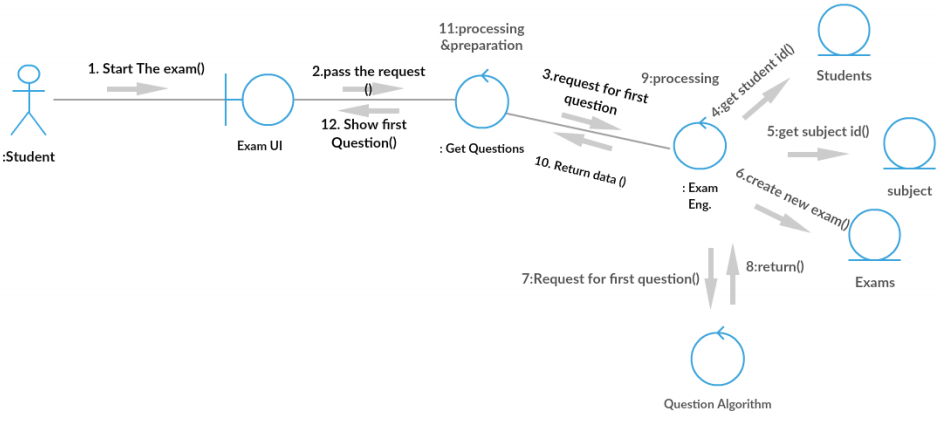
**Figure 61: Add New Exam Collaboration Diagram**



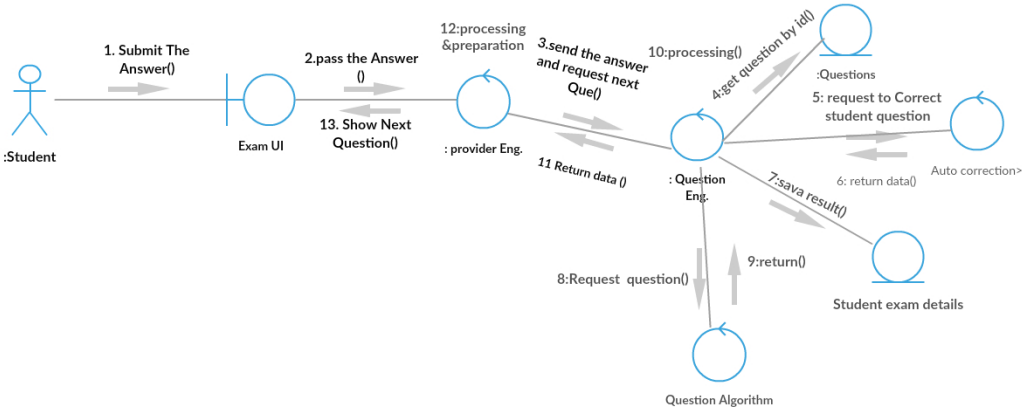
**Figure 62:Add new Question Collaboration Diagram**



**Figure 63: Finish The Exam Collaboration Diagram**



**Figure 64:Start The Exam Collaboration Diagram**



**Figure 65: Submit Answers Collaboration Diagram**

# 9. Chapter 7: Testing

# 9.1. Test Suits

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A test suit to verify the “Fetch Next Question” feature | | | | | | |
|  | | TSI | | | | |
| Title | | Verify “Fetch Next Question” functionality | | | | |
| Description | | To test the different scenarios that might arise while a user is trying to login | | | | |
|  | | | | | | |
| # | Summary | Dependency | Pre-condition | Post-Condition | Execution Steps | Executed Output |
| TC1 | Verify that user already logged in to the system can Fetch Question |  | Student who logged in type a message and click send | The Question is Fetched and Viewed | 1. Login as student 2. Click next Question | New Question appeared to the student |
| TC2 | Verify That Student Can’t Fetch Question after the end of time |  | Student who logged in Click Fetch Next Question after the end of exam time | The Question is not Fetched and Error message appeared | 1. Login as student 2. Wait until the time of Exam is finished 3. Click Fetch Question | Pop out Error Message tell the user that the exam ended and can’t get another question |
| TC3 | Verify that question  Difficulties is reasonable |  | Student Click Next Question | The Fetched Question difficulty is reasonable | 1. Login as student 2. Click next question | New question appeared to the student |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A test suit to verify the “Finish Exam Functionality” feature | | | | | | |
|  | | TSI | | | | |
| Title | | Verify “Finish Exam Functionality” functionality | | | | |
| Description | | To test the different scenarios that might arise while a user is trying to login | | | | |
|  | | | | | | |
| # | Summary | Dependency | Pre-condition | Post-Condition | Execution Steps | Executed Output |
| TC1 | Verify that exam is finished after clicking finish exam button |  | Student who logged in click finish exam | The exam is finished | 1. Login as student 2. Click finish exam | Pop out screen tell the Student that the exam is finished and His result |
| TC4 | Verify the exam is Automatically finished after the time is ended |  | The time of the exam is ended | The exam is finished | 1. Login as student 2. Wait until the time of Exam is finished | Pop out screen tell the Student that the exam is finished and His result |
| TC3 | Verify that the result is showed after the exam is finished | TC1 or TC2 | The time of the exam is ended **OR**  Student who logged in click finish exam | The exam is finished and the result is showed | 1. Login as student 2. Wait until the time of Exam is finished   **OR** the Finish exam button is clicked | Pop out screen tell the Student that the exam is finished and His result |

# 10. Conclusion

We build Online Exam System which is a distributed system. Our system is consisting of three main servers Database Server, API Server and MVC Server. The main server is the Web API server. And the MVC server is the endpoint of our system.

**Technologies we used are:**

ASP.NET Web API, ASP MVC, Entity Framework, MS-SQL DB, Oracle DB, MySQL DB, HTML & CSS, JavaScript, JQuery, Ajax.

**We Developed Two algorithms:**

* Question Fetching Algorithm
* Auto Correction & Auto Grading Algorithm

**We secured our System against the following attacks:**

* Denial of Service
* SQL injection
* Cross site scripting

# 11. References

**Books:**

**J. Kurtz & B. Wortman, ASP.NET Web API 2, Second Edition, Apress.**

**B. Lakshmiraghavan & D. Baier, Pro ASP.NET Web API 2 Security, First edition, Apress.**

**Jonas Fagebrerg, ASP.NET MVC 5 Building A Website, First Edition, csharpschool.**

**Julia Lerman, Rowan Miller, Programming Entity Framework, 2011, O'Reilly Media.**

# 12. Index

[**Figure 1: Client Server Model** 18](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869502)

[**Figure 2: Online Exam Architecture** 19](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869503)

[**Figure 3: Architecture of Entity Framework** 20](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869504)

[**Figure 4: ORM in Entity Framework** 21](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869505)

[**Figure 5:The Mechanism of Entity Framework** 21](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869506)

[**Figure 6: Database ERD** 22](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869507)

[**Figure 7:API Mechanism (Example)** 23](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869508)

[**Figure 8:Connection Between Client and Server in Web API** 25](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869509)

[**Figure 9:WCF VS. ASP.NET Web API** 26](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869510)

[**Figure 10: MVC Model** 28](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869511)

[**Figure 11: How Ajax Request is Made** 29](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869512)

[**Figure 12:Grunday's tree for 7** 32](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869513)

[**Figure 13:Question Algorithm tree for 5** 33](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869514)

[Figure 14:Question Correction Algorithm Example 35](#_Toc487869515)

[**Figure 15: Denial of Service attack** 36](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869516)

[**Figure 16:SQL Injection attack** 38](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869517)

[**Figure 17:XSS attack** 39](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869518)

[**Figure 18:Use-Case Diagram** 40](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869519)

[Figure 19: Start Exam flow of events 41](#_Toc487869520)

[**Figure 20: Start Exam Interface Prototype** 42](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869521)

[Figure 21: Request New Exam flow of events 43](#_Toc487869522)

[**Figure 22:Submit the answer flow of events** 44](#_Toc487869523)

[**Figure 23: Submit the answer Interface Prototype** 45](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869524)

[Figure 24:Correct and fetch the next question flow of events 46](#_Toc487869525)

[Figure 25: Finish the exam flow of events 47](#_Toc487869526)

[**Figure 26: Finish the exam Interface Prototype** 48](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869527)

[Figure 27: Fetch the final Result flow of events 49](#_Toc487869528)

[Figure 28: Sign in flow of events 50](#_Toc487869529)

[**Figure 29: Sign in Interface Prototype** 51](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869530)

[Figure 30: Get Available view flow of events 52](#_Toc487869531)

[**Figure 31: Show Marks flow of events** 53](#_Toc487869532)

[**Figure 32: Show Marks Interface Prototype** 53](#_Toc487869533)

[Figure 33: Detailed marks flow of events 54](#_Toc487869534)

[**Figure 34: Detailed marks Interface Prototype** 54](#_Toc487869535)

[**Figure 35: Change password flow of events** 55](#_Toc487869536)

[**Figure 36: Change password Interface Prototype** 55](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869537)

[**Figure 37: Add Question flow of events** 56](#_Toc487869538)

[**Figure 38: Add Question Interface Prototype** 56](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869539)

[**Figure 39: Managing Questions flow of events** 57](#_Toc487869540)

[Figure 40: Managing Questions Interface Prototype 57](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869541)

[**Figure 41:Statistic flow of events** 58](#_Toc487869542)

[**Figure 42: Statistic Interface Prototype** 58](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869543)

[**Figure 43: Add Account flow of events** 59](#_Toc487869544)

[**Figure 44: Add Account Interface Prototype** 59](#_Toc487869545)

[**Figure 45: Manage accounts flow of events** 60](#_Toc487869546)

[**Figure 46: Manage accounts Interface Prototype** 60](#_Toc487869547)

[**Figure 47: Add Exam flow of events** 61](#_Toc487869548)

[**Figure 48: Add Exam Interface Prototype** 61](#_Toc487869549)

[**Figure 49:Manage Exam flow of events** 62](#_Toc487869550)

[**Figure 50: Manage Exam Interface Prototype** 62](#_Toc487869551)

[**Figure 51: Add Subject flow of events** 63](#_Toc487869552)

[**Figure 52: Add Subject Interface Prototype** 63](#_Toc487869553)

[**Figure 53: Manage Subjects flow of events** 64](#_Toc487869554)

[**Figure 54:Manage Subjects Interface Prototype** 64](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869555)

[**Figure 55:Add Exam Sequence Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869556)

[**Figure 56: Add Question Sequence Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869557)

[**Figure 57: Add Student Sequence Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869558)

[**Figure 58: Delete Question Sequence Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869559)

[**Figure 59:Start Exam Sequence Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869560)

[**Figure 60: Finish Exam Sequence Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869561)

[**Figure 61: Add New Exam Collaboration Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869562)

[**Figure 62:Add new Question Collaboration Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869563)

[**Figure 63: Finish The Exam Collaboration Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869564)

[**Figure 64:Start The Exam Collaboration Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869565)

[**Figure 65: Submit Answers Collaboration Diagram** 65](file:///C:\Users\FNI\Desktop\Project%20Report.docx#_Toc487869566)