Examination System

Team Tigers

Table of Contents

Project Overview……………………………………………3

Architecture………………………………………………….4

Database Schema………………………………………....8

User Interface………………………………………………14

Installation Guide………………………………………….18

Usage Guide……………………………………………….20

Conclusion………………………………………………….26

**Project Overview**

The Examination System is a software solution for an e-learning organization that wants to generate, grade, and customize their exams for their students. The software enables the instructors to fill in the questions bank, then generate an exam based on the course type. The exam is generated fairly based on specific criteria given in the business logic. The students can then take exams in the desired courses. On submission, the exam is graded, and the student’s grade is then stored so he can review it whenever. Both the instructor and the student can also view their details via the system UI.

**Architecture**

**Data Layer:**Utilizes a Microsoft SQL Server database to store and manage organization data.

Firstly, the analysis phase, where the requirements were gathered.

Then came the ERD designing phase, where it was determined that there were three main entities (Exam, Instructor and Student).

Based on that ERD, the mapping process took place, after it, a clear schema of the database emerged.

Finally, the implementation phase began, using the MS SQL Server, scripts were formed from the schema, generating stored procedures and detailed tables that will build the software underlying structure.

**Logic Layer:**

At the start of the system, the system administrator creates an account for the user through the database using the email provided by them. The user is then prompted to enter their username and password, from which the system determines whether the logged in person is an instructor or a student. After being logged in, the system redirects each user to either an instructor layout or a student layout.

The instructors can view their details in their layout. They also have the ability to create questions in the questions bank and generate exams in only the courses they are assigned to. They can also view their department’s details, and depending on whether they are the manager or not, they can update the department details. Lastly, they can view their students’ grades.

The students can view and edit their details in their layout accordingly. They can take exams in the courses they are assigned to. Lastly, they can view their grades in the courses they are assigned to.

**Presentation Layer:**

After wrapping up the data and logic layers, comes the presentation layer. The system uses Windows Forms .NET as a user interface to provide accessibility of the system’s functions through a friendly interface. The system also uses MetrosetUI as it provides a more elegant look to the forms.  
  
It consists of 3 parts, Login Form, Instructor Form, and Student Form.

* Login Form is responsible for handling the authentication of the user, passing the type of the current user to the business logic.
* Instructor Form is responsible for the instructor’s actions and details.
* Student Form is responsible for the student’s actions and details.

**Database Schema**

**Main Tables:**

* Course: Stores the courses provided by the organization. Consists of **Cr\_ID**, **Cr\_Name**.
* Topic: Stores topics that go under each course. Consists of

**Topic\_ID**, **Topic\_Name**, **Cr\_ID ->** (Relation between Course and Topic)

* Department: Stores the departments. Consists of

**Dept\_ID**, **Dept\_Name**, **Dept\_Description, Location, Mgr\_ID ->** (Relation between Instructor and Department), **MGR\_HireDate**.

* Instructor: Stores the instructors’ information. Consists of

**Ins\_ID**, **Ins\_Name**, **Ins\_Salary**, **Ins\_Degree**, **Dept\_ID ->**

(Relation between Department and Instructor), **Username**.

* Student: Stores the students’ information. Consists of

**St\_ID**, **St\_Name**, **St\_Phone**, **St\_Age**, **Street**, **City**, **Zip\_Code**, **St\_Email**, **Username**, **Dept\_ID ->** (Relation between Department and Student)

* Exam: Stores the exams generated and whether they are taken or not. Consists of

**Ex\_ID**, **Cr\_ID**, **IsAssigned**.

* Question: Stores information about questions. Consists of  
  **Q\_ID**, **Q\_Head**, **Q\_Grade**, **Q\_Difficulty**, **Type**, **Model\_Answer**, **Cr\_ID->** (Relation between Course and Question)
* Question\_Choices: Stores the choices for the questions. Consists of

**Choice\_Desc**, **Choice Selector, Q\_ID ->** (Relation between Question and Question Choices)

**Relation Tables:**

* Instructor\_Course: Represents the M-M relationship between the two tables. Consists of

**Ins\_ID**, **Cr\_ID**

* Exam\_Questions: Represents the M-M Relationship between the two tables. Consists of  
  **Q\_ID**, **Ex\_ID**
* Student\_Exam\_Questions: Represents the relationship between the Exam, Student and Question tables. Consists of

**St\_ID**, **Exam\_ID**, **Q\_ID**, **St\_Answer**

* Student\_Course: Represents the M-M relationship between the two tables as well as their grade in said course. Consists of

**St\_ID**, **Cr\_ID**, **St\_Grade**

**Login Tables:**

* Logins: Stores the Authentication information for the users. Consists of

**Username**, **Password**, **AccountType**

* For the system’s safety, it was decided that the Login tables would be in a different database to protect it from any possible data breaches or singular database crash with the information database.

**ERD**

A diagram of a network

Description automatically generated

**Mapping**

A screenshot of a computer

Description automatically generated

**A computer screen shot of a computer flow chart

Description automatically generatedDiagram**

**User Interface**

* The user interface is designed using C#’s Windows Forms .NET and enhanced by the [Metroset UI](https://github.com/N-a-r-w-i-n/MetroSet-UI) to give it a fresher look aligning with the new versions of windows.
* A screenshot of a login box

  Description automatically generatedIt consists of 3 main forms:  
  **1- Login Form:**
* A screenshot of a computer

  Description automatically generated**2- Instructor Form:**
* **3- Student Form:**

**A screenshot of a computer

Description automatically generated**

**Installation Guide**

* To install the examination system you need the following items:
  + Visual Studio and Sql Server Management Studio (SSMS)
  + Project files ([GitHub](https://github.com/AhmedFawzi98/ExaminationSystemProject))
  + Databases .bak
* Steps:
  1. Install Visual Studio and SSMS
  2. Download the project files (clone if using GitHub)
  3. Download the database restore files (ExaminationSystem.bak, ExaminationSystemLogin.bak)
  4. Restore the databases through SSMS.
  5. Configure the context connection string through the App.Config file.

<connectionStrings>

    <add name="ConnectionString" connectionString="Data Source=.;Initial Catalog=Examination\_System\_Final;TrustServerCertificate=True;Integrated Security=True"/>

    <add name="LoginConnectionString" connectionString="Data Source=.;Initial Catalog=Examination\_System\_Logins;TrustServerCertificate=True;Integrated Security=True"/>

</connectionStrings>

* 1. Build the application and run.

**Usage Guide**

1. A screenshot of a login box

   Description automatically generatedLogin through the login form using username and password provided by database administrator.
2. Navigate through the popup form
   1. **Instructor Form**

A screenshot of a computer

Description automatically generated2.1.1 Profile

2.1.2 Generate Exam

A screenshot of a computer

Description automatically generated

2.1.3 Student Grades

A screenshot of a computer

Description automatically generated

2.1.4 Department Details

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated2.1.5 Questions Bank: used for storing new questions

* 1. **Student Form**

A screenshot of a computer

Description automatically generated2.2.1 Profile

2.2.2 Take Exam

A screenshot of a computer

Description automatically generated2.2.3 Grades: only the current student’s grade

**Conclusion**

In conclusion, the Examination System represents a robust and comprehensive solution tailored for e-learning organizations seeking efficient exam generation, grading, and customization capabilities. Throughout the development process, **Team Tigers** accurately analyzed requirements, designed an appropriate database schema, and implemented a multi-layered architecture to ensure scalability, reliability, and security.

The software's architecture, building distinct layers for data management, business logic, and presentation, facilitates seamless interaction between users and the system. Leveraging Microsoft SQL Server for data storage ensures the integrity and accessibility of organization data, while Windows Forms .NET, augmented with Metroset UI, provides a user-friendly interface for instructors and students alike.

With features such as exam generation based on predefined criteria, question bank management, and grade tracking, the system empowers instructors to efficiently manage courses and assessments. Similarly, students benefit from streamlined exam-taking experiences and easy access to their grades and profile information.

The installation process, outlined in the documentation, ensures a smooth setup for administrators, while the usage guide offers clear instructions for users to navigate the system's functionalities effectively.

In essence, the Examination System exemplifies **Team Tigers'** commitment to delivering a tailored software solution that meets the evolving needs of e-learning organizations, enhancing productivity, and fostering an enriched learning experience for both instructors and students.