Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Bachelor Thesis

Designing an integrated application for online examination

Hazem Suleiman

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

BACHELOR THESIS ASSIGNMENT

abs. v. š. Hazem Suleiman, BA

Systems Engineering and Informatics
Informatics

Thesis title

Designing an integrated application for online examination

Objectives of thesis

The main goal of the thesis is to design an integrated application for conducting online examinations, to ensure the integrity and transparency of the examiners, and to ensure a recommended educational level for the student generation considering the expansion of the popularity of online exams in the last few years.

Methodology

The bachelor thesis consists of two parts – theoretical and practical. The methods used for creating the theoretical part will be based on studying the various information sources.

Based on the synthesis of the gained knowledge the groundwork for the practical part will be formulated. The practical part will consist of the analysis, design and implementation of an online examination platform. The application will be implemented using the .NET Framework. The final application will be deployed and practically tested. Based on the testing results the possible future improvements will be proposed.

The practical part will utilize the standard tools and methods of software engineering.

The proposed extent of the thesis

35-40 pages

Keywords

.NET Framework, C#, XAML, WPF Windows Presentations Foundation, VS API, Online examination

Recommended information sources

BillWagner. ..NET Documentation. https://docs.microsoft.com/en-us/dotnet/. Accessed 06 June 2021

Chowdhury, Kunal. Windows Presentation Foundation Cookbook. 2018. Open WorldCat, http://www.vlebooks.com/vleweb/product/openreader?id=none&isbn=9781788396356.

Sharp, John. Microsoft Visual C� Step by Step. Ninth edition, Microsoft Press ; [Place of publication not identified] : Pearson Education, Inc, 2018.

Yuen, Sheridan. Mastering Windows Presentation Foundation, Packt Publishing, Limited, 2017. ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/czup/detail.action?docID=4812120.

Expected date of thesis defence

2021/22 SS – FEM

The Bachelor Thesis Supervisor

Ing. Jiří Brožek, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 1. 11. 2021	Electronic approval: 23. 11. 2021
Ing. Martin Pelikán, Ph.D.	**Ing. Martin Pelikán, Ph.D.**
Head of department	Dean

Prague on 06. 03. 2022

Declaration

I declare that I have worked on my bachelor thesis titled " Designing an integrated application for online examination" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the bachelor thesis, I declare that the thesis does not break any copyrights.

In Prague on 15th March 2022 ______________________________

Hazem Suleiman

Acknowledgement

I would like to thank Ing. Jiři Brožek Ph.D., for their guidance and encouragement during my work on this thesis.

Designing an integrated application for online examination

Abstract

The main goal of the thesis is to design an integrated application for conducting online examinations, to ensure the integrity and transparency of the examiners, and to ensure a recommended educational level for the student generation considering the expansion of the popularity of online exams in the last few years.

This bachelor thesis deals with the development of a desktop application "Oculus" in the C# programming language and will be implemented using the .NET Framework and WPF.

The application is designed to allow students to take online exams in a monitored environment, providing the typical access for students to attend the online tests while providing the examiners a powerful control of the exam and the examinees.

The introductory part introduces technologies and tools for the development of the parts of the desktop application.

The analysis, design, and implementation of your own application are also described.

All the coding blocks in the practical parts have been tested successfully.

Finally, the deployment, testing, and evaluation of the application are described, along with the design of other improvements that brought testing to the user.

Keywords: .NET Framework, C#, XAML, WPF Windows Presentations Foundation, Visual Studio, MVVM, Azure, SQL Server. Online Examination.

Title of Bachelor Thesis in Czech

Abstrakt

Hlavním cílem práce je navrhnout integrovanou aplikaci pro skládání online zkoušek, zajistit integritu a transparentnost zkoušejicích a zajistit správnou úroveň vzdělání pro generaci studentů, s ohledem na rozšíření a popularitu online zkoušek v posledních několika letech.

Tato bakalářské práce se zabývá vývojem desktopové aplikace „Oculus“ v programovacím jazyku C# a bude implementovaná pomocí .NET frameworku a WPF.

Aplikace je navržena tak, aby studentům umožnila skládat zkoušky online ve sledovaném prostředí a nabízela přístup ke složení testů online a dávala zkoušejicím silnou kontrolu nad zkouškou a zkoušenými.

Úvod představuje technologie a nástroje, které byly použity k vývoji každé části desktopové aplikace.

Analýza, design a implementace vlastní aplikace je také popsána.

Všechen kód z praktické části byl úspěšně testován.

Nakonec je popsáno uvedení do provozu, testování a hodnocení aplikace spolu s designem a ostatními vylepšeními, které přinesl tento typ testování uživatelům.

Klíčová slova: .NET Framework, C#, XAML, WPF Windows Presentations Foundation, Visual Studio, MVVM, Azure, SQL Server. Online zkoušky.

Table of Contents

1 Introduction

In 2019 the post technology world has experienced the first pandemic, the Covid-19 pandemic.

It affected every single aspect of our daily life, Health, Business, Travel, Manufacturing, and Education.

The effects of the epidemic in Europe on the education sector were evident through the necessity to change the pattern of teaching and examination.

The sudden change in education conditions forced educational institutions and universities to use electronic meeting platforms such as MS Team, Google Meet, and Zoom.

Exams conducted online have cut down geographical constraints, allowing for greater reach and participation. Universities and education sectors are trained in global assessment solutions, allowing them to conduct assessments with ease. The ability to conduct examinations with merely an internet connection and devices provides online assessments with great accessibility and reach.

These platforms were sufficient for meetings, consulting, and discussing the business requirements, even were suitable for lectures by sharing the slides and notes. However, when it came to exams and tests, online examination integrity has been dealing with a lot of challenges.

The application is designed to put all the possibilities in the exam coordinator's hands.

Starting with the visual control (Camera), Voice detection (Microphone), and the ability to monitor the examinees' screens.

Oculus - The name of the application- will have easy access designed user interface with a direct path starting from the login using the school/educational institution email, verifying the user students and coordinators, leading to the integrated collaborative screen where the exam will be taken.

There are many technologies are used to design and architect Oculus. They will be explained, as well as the technologies that will make the application a top-notched software.

2 Objectives and Methodology

2.1 Objectives

In my thesis I will present the idea of an online platform for meetings and examination, however, the application, Oculus will focus on the online examination details to reduce the possibility to cheat during the online tests and keep the education level as the same level of honesty and integrity as a real-life situation.

The providing details explain how the application will be presented to students and teachers as well, explaining the technologies that are used to achieve this connection.

2.2 Methodology

The work consists of two parts; the process for developing the theoretical section of the work is based on research into professional information sources. These sources will develop the theoretical framework for processing the practical component of the work based on the synthesis of the findings.

The practical part of the work consists of the design and implementation of a WPF application.

The Windows Presentation Foundation (WPF) is a user interface framework for developing desktop client applications. A wide range of software development features is supported by the WPF development platform, including an application model, resources, controls, graphics, layout, data binding, documents, and security.

Because of the inclusivity of WPF, it will be used to implement the front-end part of the application, however, there will be another technology to develop the UI of the application

From its development and testing the findings will be summarized and the possibilities of further development will be outlined application.

Establishing database queries for exams, creating an access for teachers will be created using SQL tables, using Microsoft Azure.

deploy the database on a connected server, using Microsoft SQL Server Management Studio.

3 Literature Review

3.1 .Net Framework

.Net Framework is a comprehensive software platform designed for the development of a diverse range of types of applications. It offers development from basic window applications, through dynamic web services to mobile applications, not only for Windows Phone but also for the Android2 and IOS3 operating systems The Net Framework itself is the environment that is needed for running the application not only offers a launcher interface but provides a reference to the necessary libraries for a given application The platform offers extensive possibilities of using a programming language.

However, the most popular languages are C # C ++ and VB As a result, when compiling the source code, everything is translated into the CLI4 interlanguage which is executable on any platform if it has access to the Net Framework

The.NET Framework is a run-time execution environment that manages.NET Framework-targeted applications. It comprises a common language runtime that offers memory management and other system functions, as well as a large class library that allows programmers to use solid, dependable code in all main areas of app development.

The architecture of the .NET framework has a simple design with powerful effects.

The two major components of the .NET Framework are the Common Language Runtime and the .NET Framework Class Library.

The Common Language Runtime (CLR) is the execution engine that handles running applications. It provides services like thread management, garbage collection, type-safety, exception handling, and more. (1)

The Class Library provides a set of APIs and types for common functionality. It provides types for strings, dates, numbers, etc. The Class Library includes APIs for reading and writing files, connecting to databases, drawing, and more.

.NET applications are written in the C#, F#, or Visual Basic programming language. Code is compiled into a language-agnostic Common Intermediate Language (CIL). Compiled code is stored in assemblies—files with a .dll or .exe file extension. (2)

When an app runs, the CLR takes the assembly and uses a just-in-time compiler (JIT) to turn it into machine code that can execute on the specific architecture of the computer it is running on.

3.2 The evolution of desktop application development

Microsoft began creating a distributed desktop solution in the early 2000s to transmit information across different branches of the organization and perform efficient operations on centralized units. For their application development, they've picked a brand-new framework called Windows Forms (commonly known as WinForms). With hundreds of thousands of lines of code, the project has matured into a mature, well-tested, and time-proven program. After some time has passed, the.NET Framework 2.0 is no longer the most cutting-edge technology. The programmers who are working on this application are in a pickle. They want to construct their app with the most up-to-date stack of technologies and make it appear and "feel" current.

Desktop programs were the primary method of developing software systems prior to the Internet's arrival. Any programming language, such as COBOL, Fortran, VB6, or C++, might be used by developers. They all produced desktop apps, whether they were simple tools or big networked structures.

Then, with advantages like quick deployment and simpler distribution methods, Internet technologies began to astound the development industry and win over more engineers. The fact that all users received automatic updates once a web application was delivered to production had a significant influence on software agility.

The Internet infrastructure, underlying protocols, and standards like HTTP and HTML, on the other hand, were not meant to support the development of complicated applications. The primary development goal at the time was to provide online apps with the same features as desktop programs, such as quick data entry and state management.

Even though online and mobile apps have expanded at an astonishing rate, desktop applications remain the most efficient and performant for particular jobs. That explains why millions of developers use WPF and WinForms to create their projects, and the number of such apps is always increasing.

3.2.1 Windows Presentation Foundation

WPF emphasizes a clear separation between UI and code, which is based on the XAML language standard. XAML has features like templating, style, and binding that make it ideal for creating massive applications. It's a controlled framework, like Windows Forms, except the architecture is modular and reusable.

The preferences that most developers use nowadays to design from code using declarative XAML, the higher learning curve than Windows Forms, and the long-term support of the Windows version make WPF a preferable choice to design windows applications.

WPF provides developers with a unified programming model for building rich, modern desktop applications on Windows.

It helps the developer to layout the user interface, add images and custom user controls, work with data binding and data templates, and style content. Plus, adding attention-grabbing 3D and UI effects. (3)

The architecture of WPF is multi-layered architecture. It spans across three layers – Managed code, Unmanaged code, and core operating system, we can call these layers a set of assemblies that built up the entire framework. Managed Layer: - The public API exposed is only via this layer.

3.2.2 The behaviour of WPF in .NET Framework

WPF is a subset of.NET types, most of which are found in the System.Windows namespace.

provides to developers as a technology various ability:

- Create classes from scratch.
- Make changes to the properties
- Methods to call

Because appearance-specific markup isn't tightly connected with behavior-specific code, development and maintenance expenses are decreased.

Designers may implement an application's look simultaneously with developers who are creating the application's behavior, making development more efficient.

WPF application globalization and localization have been eased. (4)

3.3 WPF architecture

WPF allows creating an application utilizing both HTML and code-behind, which should be familiar territory for ASP.NET developers. The look of an application is often implemented using XAML markup, while the behaviour is typically implemented using managed programming languages (code-behind).

WPF is mostly written in managed code.

1)PresentationFramework.dll: This file contains the top-level WPF components, such as Windows, panels, controls, styles, and so on. It also provides end-user presentation elements such as data binding and time-dependent, story-based animations.

2) PresentationCore.dll: - Presentation Core is a managed wrapper for MIL that implements core WPF services like UI Element and visual, from which all shapes and controls in PresentationFramework.dll are derived.

3) WindowsBase.dll: Contains additional fundamental features such as Dispatcher objects and Dependency objects that may be utilised outside of the WPF environment.

3.3.1 Layer Unmanaged

1) milCore.dll: Despite the fact that the bulk of WPF is managed, the composition engine that renders the WPF application is a native component. The file milCore.dll contains a Media Integration Layer (MIL). The milord's objective is to give basic support for 2D and 3D surfaces by interacting directly with DirectX. It is WPF's main rendering engine and serves as a bridge between DirectX and CLR.

2) WindowsCodecs.dll (Windows Codecs DLL): WindowsCodecs is another low-level API used in WPF applications for imaging support like as image processing, picture displaying and resizing, and so on. It consists of multiple codecs that encode and decode pictures into vector graphics, which are then rendered on the WPF screen.

3.3.2 Kernel (core operating system layer)

1) DirectX: DirectX is a low-level API that WPF uses to render all of its visuals.
2) User32: This is the main API that all applications use. Memory and process separation are managed by User32.

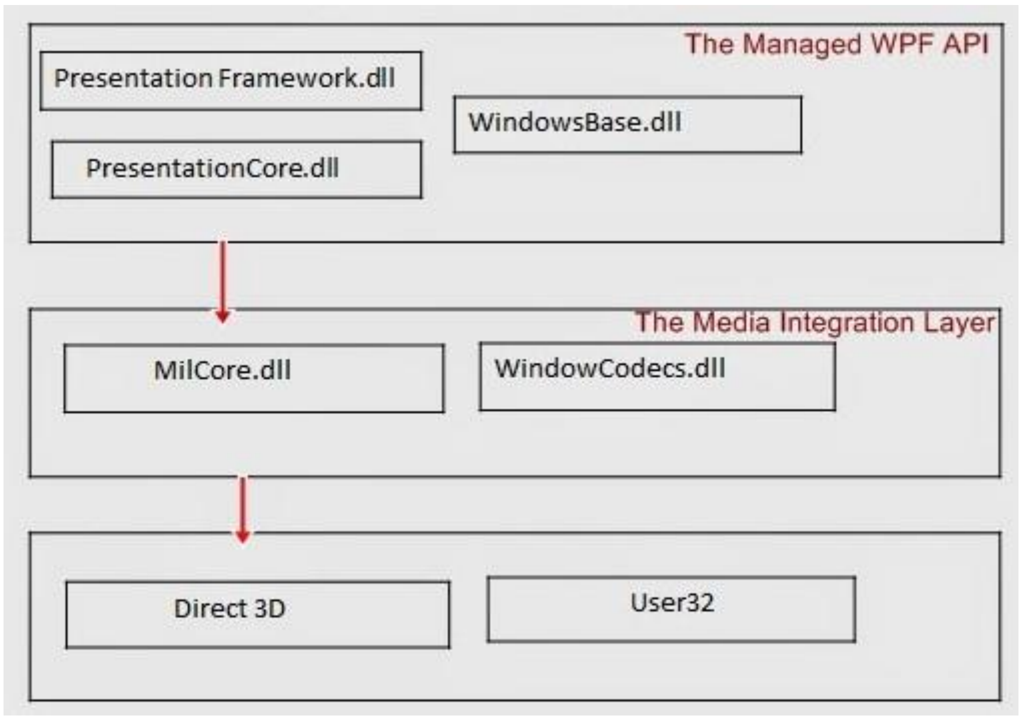


Figure 1: WPF Architecture (5)

In Simple way, The Previous diagram depicts the primary components of WPF architecture:

- Milcore
- Presentation Framework
- Presentation Core

The managed layer contains the Presentation Framework and Presentation Core.

Milcore, on the other hand, considers it to be unmanaged code that allows for close connection with Direct X. (responsible for display and rendering)

CLR improves the productivity of the development process by providing a number of features like as memory management, error handling, and so on. (6)

3.4 WPF User Interface Layout

The shell is the application root object that contains the primary UI content. In a Windows Presentation Foundation (WPF) application, the shell is the Window object.

The shell is the root item of a composite application. The shell serves as the application's master page. There are one or more areas in the shell. Regions serve as placeholders for material that will be loaded later. Regions are used to handle the content of UI components including ContentControls, ItemsControls, TabControls, and custom controls. Depending on the application needs, region material can be loaded automatically or on-demand. (3)

A view is often the content of a region. A view encompasses a component of your UI that you want to keep as separate from the rest of the program as feasible. A view can be defined as a user control, a data template, or a custom control.

A view is often the content of a region. A view encompasses a component of your UI that you want to keep as separate from the rest of the program as feasible. A view can be defined as a user control, a data template, or a custom control.

A region controls how views are displayed and laid up. Regions may be accessible by their names in a disconnected manner, and dynamically adding and deleting views is possible. A hosting control is linked to a region. Consider regions to be containers for dynamically loaded views. (7)

3.5 MVVM (Model-View-ViewModel)

Is an architecture pattern in WPF, WPF has a wide range of data binding, which allows us to use this architecture pattern.

- Model: is an entity that represents customer, product, or any data object in our case students
- View: is the user interface that has been created by WPF including the windows, buttons, any interactive feature between the users (students/Teachers), and the applied logic code behind the WPF design.
- ViewModel: is the program logic. This architecture allows the Model and ViewModel to interact through View by the data binding in WPF using XAML.

I will show the proper design for Model, View, and ViewModel separately.

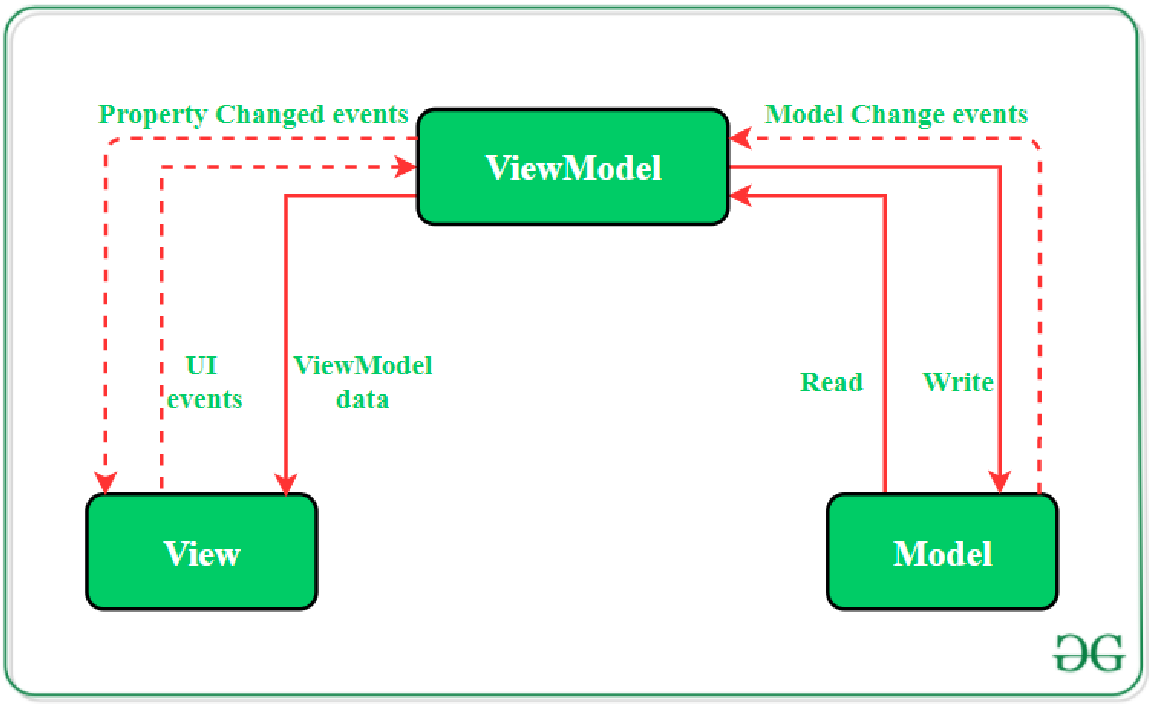


Figure 2: The working component of MVVM

- Model: Model components consist of Models Data and Models Data Providing
- View: the layer where the User interface creates the components that users (students) use.View components consist of UI controlling and Data Binding
- ViewModel: perform actions ViewModel, Loaded from the database of the school server or add manually implementing this code. (8)

3.5.1 The implementation of MVVM in WPF

The Model View ViewModel (MVVM) is an instructional design architectural pattern that originated with Microsoft, a company that specializes in the Presentation Model design pattern. It's built on the Model-View-Controller (MVC) paradigm and is aimed for current UI development platforms (WPF and Silverlight) with a UX developer that has different needs than a "conventional" developer. MVVM is a method of developing client applications that takes advantage of key WPF platform capabilities, allowing for simple unit testing of application functionality, and makes it easier for developers and designers to collaborate with fewer technical hurdles.

The principle should be the separation of logic applications from the user interface. The result is a more readable, clearer structure, with further code development is much easier. The user interface communicates with the application logic through command and binding. unlike the Windows forms where it took place event communication. The principle is based on a simple idea, the "mediator" of the ViewModel class preserves the state of the application, the user interface (View) requests data to be printed to users. When the data in the View changes, the ViewModel is called, which saves the changes.

3.5.2 The structure behind the architecture

• Model describes the data with which the application works

• ViewModel - holds the state of the application and associates the Model with the View The controls are solved by binding and in order for everything to work properly it must be used INotifyPropertyChanged

• View - user interface written in XAML language

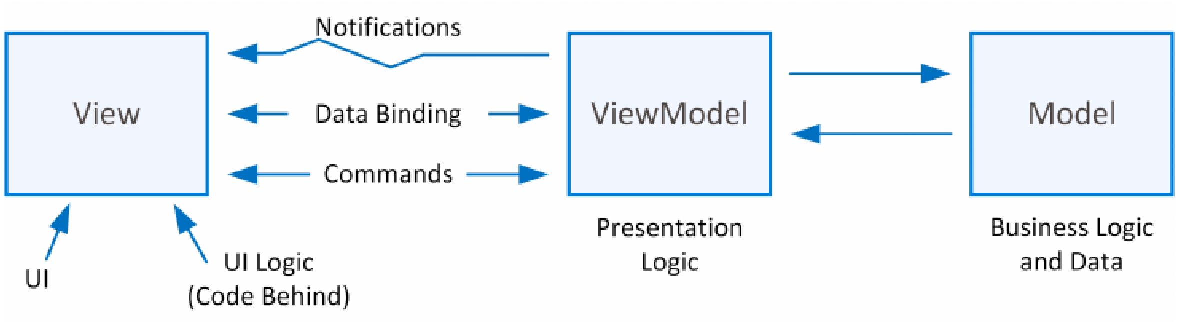


Figure 3: The Architectural Pattern of MVVM

A ViewModel is always associated with it. As a result, the relationship, in this case, is always "one view - one ViewModel." This ViewModel can have one or more Data Binding models, but it can also have one or more Data Binding view models.

3.6 Data Binding

Binding data It is a technical connection of elements and elements to data. Data Binding is used to automate data entry, such as into a table using a Data Template, in which a template is prepared where the data from the collection are called, and thanks to the correct settings they are put in place. [8]

It is essentially the way for us to take a data from one element to another without having to fire some kind of events.

3.7 XAML

Microsoft created XAML as a new descriptive programming language for creating user interfaces for next-generation managed applications. XAML is the user interface language for Windows and Mobile apps that employ Windows Presentation Foundation (WPF), Universal Windows Platform (UWP), and Xamarin Forms.

The goal of XAML is to develop user interfaces with a markup language that resembles XML. XAML provides user interfaces and C# (or VB.NET) is used as code-behind language.

Microsoft created a declarative XML-based language for initializing structured variables and objects. It's free to use according to Microsoft's Open Specification Promise.

WPF, Silverlight, Workflow Foundation (WF), Windows UI Library (WinUI), and the Universal Windows Platform (UWP) all make extensive use of XAML (UWP). XAML is a user interface markup language used in WPF and UWP to design UI components, data binding, and events. XAML, on the other hand, specifies processes in WF. (4)

XAML elements correspond to CLR object instances, and XAML attributes correspond to CLR properties and events on those objects. Microsoft Expression Blend, Microsoft Visual Studio, the hostable WF visual designer, XAMLPad, or Vector Architect may all be used to build XAML files. (9)

3.7.1 XAML features in .NET framework

XAML language for creating graphical user interface11 (presentation layers). XAML is based on the XML markup language, it is very easy to read even for inexperienced users and can be customized.

A XAML document is composed of elements that can contain other elements. A document structured in this way is called a tree structure. We write the element (tag) in angle brackets. We end the paired elements by placing another mark after the slash and odd ones end with a slash. We can insert the content of the paired elements between angle brackets or directly into the element attribute (sample Source Code 1 - Example XAML). The namespace is another part used in XAML. (10)

4 Storing data

4.1 SQL

SQL is a powerful database-interfacing programming language. It operates by deciphering and analysing databases with data fields.

The data is organized and kept in a database, but it must be valuable and accessible, which is where SQL comes in. In this situation, SQL is a platform that connects front-end and back-end databases (computers and databases held on servers).

When it comes to requesting data in any project, it requires storing, representing, implementing data in a way which is easy to provides, deploying database queries on server consists of two main processes: creating the table of the database with the required attributes and connect the database with the server for efficient access.

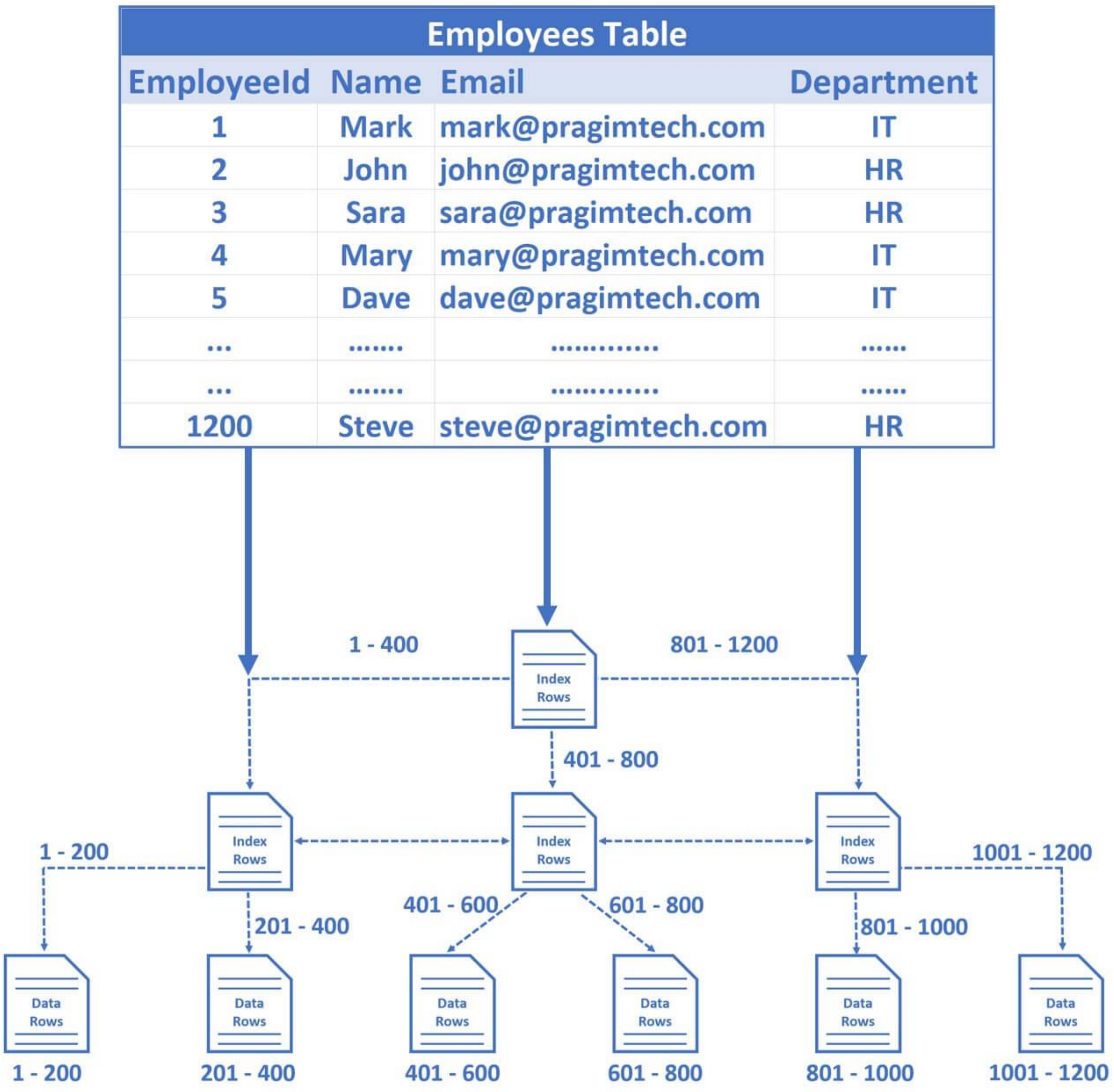
after creating the needed queries by a database programming language, such as MySQL, PostgreSQL, Oracle SQL or any type of database language, it comes the advantage of hosting a set of databases on an easy access, secure, and friendly user server.

Providers of cloud databases or Database hosting services are available all over the world. Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) are the main hosting providers.

If database hosting is the best option, consider cloud hosting, also known as Database-as-a-Service (DBaaS). DBaaS is a value-added version of database hosting, a subscription-based service in which the provider handles administrative tasks such database migration, installation, configuration, maintenance, and updates for a charge. (11)

The figure below represents the way data stored in the server, and the sequence to reach a row.

The structure here tree-like structure consisting of a sequence of data pages, as seen below. The B-Tree, index B-Tree, or clustered index structure is a tree-like structure.


Employees Table

EmployeeId	Name	Email	Department
1	Mark	mark@pragimtech.com	IT
2	John	john@pragimtech.com	HR
3	Sara	sara@pragimtech.com	HR
4	Mary	mary@pragimtech.com	IT
5	Dave	dave@pragimtech.com	IT
...	....	....	......
...	....	....	......
1200	Steve	steve@pragimtech.com	HR

Figure 4: How data stored in a server (12)

by query SELECT, the server can reach consecutively the requested row where the data is stored.

The prominence of database comes with the necessity of dividing data into subject-based tables to eliminate redundant data, giving access the information it needs to link the information in the tables together as needed, and supporting and ensures the correctness and integrity of your data. (13)

4.2 Cloud computing

The fundamentals of cloud computing The evolution of technology, as well as the expansion of network storage and processing capacity, has resulted in the acme of computing. Cloud computing, or simply cloud, is the term used in this century. What is cloud computing and how does it work? Cloud computing is a technique for organizing, storing, and processing data online over the Internet that allows on-demand network access to pooled computer resources. The following are some of the properties of cloud computing: You utilize an on-demand service when you need it. Using the Internet as a channel for network access. Shared resources: Multiple clients can access and use the same resources. Scalability: Allows for resource elasticity. SaaS, PaaS, and IaaS are three cloud computing delivery methods. Cloud computing provides a variety of services based on three main delivery strategies. When stacked in a pyramid shape.

4.2.1 Cloud services

They are arranged in the following order: SaaS, PaaS, and IaaS.

1- Software as a Service (SaaS)? Software as a service, or SaaS. It is a service that allows customers to pay for application software on a per-use basis. Unlike programs that have been purchased with a license. This service is platform agnostic, and you do not need to install any software on your computer to use it. The cloud hosts a single instance of the program that may be accessed by several users. This lowers the cost of cloud computing. The vendor is solely responsible for all computer resources used to supply SaaS. This service can be accessed by a web browser or with lightweight client software. SaaS is frequently used by end-users. SaaS product and service examples. The following are some of the most popular SaaS products and services. Gmail, Google Docs, and Google Drive are all part of the Google ecosystem. Microsoft Office 365, HR and helpdesk solutions, and CRM services like Salesforce are all available. Advantages: Can be accessed from any platform. There is no need to commute because you can work from anywhere. Excellent for working in a group. The vendor offers a limited set of software tools. Multi-tenancy is possible. Cons: Browser compatibility and portability difficulties. Overall performance may be influenced by internet performance. Restrictions on compliance

2- Platform as a Service (PaaS)? Platform-as-a-service, or PaaS, is an acronym for platform-as-a-service. This service consists of a programming language execution environment, an operating system, a web server, and a database, and is mostly used for development. All of this creates an environment in which users may write, compile, and run their applications without having to worry about the infrastructure. You control data and application resources in this paradigm. The vendor is in charge of all other resources. Who makes use of PaaS? This is a developer's domain. PaaS goods and services examples PaaS goods and services offered by cloud providers include the following: Amazon Web Services elastic Beanstalk, Google App Engine, Windows Azure, Heroku, and force.com are just a few of the services available. Pros: Rapid development at a low cost. It may be scaled up or down. Developers will have access to a more rapid market. Web apps can be deployed quickly. It is possible to install in a private or public environment. Cons: Developers are confined to the languages and tools provided by the vendor. Issues with migration, such as the potential of vendor lock-in.

3- Infrastructure as a Service (IaaS)? Infrastructure-as-a-Service (IaaS) This service provides computer architecture and infrastructure, in the sense that it provides all computing resources in a virtual environment, allowing many users to utilize them.

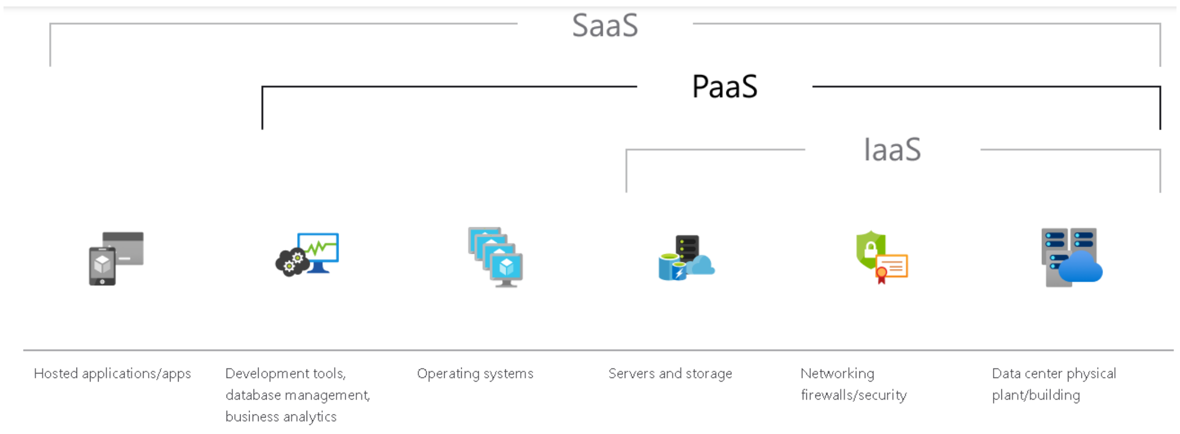

SaaS
PaaS
IaaS
Hosted applications/apps
Development tools, database management, business analytics
Operating systems
Servers and storage
Networking firewalls/security
Data center physical plant/building

Figure 5: The cloud services models among SaaS, PaaS, and IaaS (14)

4.3 Microsoft Azure

The Azure cloud platform includes over 200 products and cloud services that are meant to assist you in developing creative solutions and addressing today's challenges. Create, run, and manage applications using the tools and frameworks of your choice across clouds, on-premises, and at the edge.

The largest cloud computing platform is Amazon Web Services (AWS), while Microsoft Azure is the second largest and fastest-growing. This page is everything about Microsoft Azure, including what it is and what services and applications it offers.

Azure is a web portal and cloud computing platform that allows you to manage and access Microsoft's cloud services and resources. These services and tools may include keeping and changing your data, depending on your needs. (15)

The application Oculus was designed for the cloud it allows Microsoft to manage the platform and the infrastructure part which makes the application in that case (platform as a service) PaaS.

in this model, the cloud provider (Azure) will manage the infrastructure and the platform so besides the hardware Microsoft will need to manage the operating system all the security patches all the middleware, and the runtime to host the application while building and managing the features of the application.

typically, this model is used whenever we want to develop cloud applications such as a state-of-the-art web applications AI machine learning business analytic, and the necessities such as SQL for relational databases app service for web hosting logic apps and for enterprise integration’s function apps.

what distinguish and makes Azure desirable service is the features that Azure provides, such as:

- Azure allows building any type of web application.
- Testing: You may test an application after it has been successfully developed on the platform.
- Azure may assist the customer with application hosting once the testing is completed.
- Build virtual machines: With Azure, you may create virtual machines in any configuration you desire.
- Features for integrating and synchronizing virtual devices and directories: Azure allows you to integrate and synchronize virtual devices and directories.

- Collect and store metrics: Azure provides the ability to collect and store metrics, which can aid in the discovery of what works.

4.4 Controlling the database using SSMS

SQL Server Management Studio makes it simple to perform a variety of database and server-related operations. It enables you to connect to SQL Server instances both on-premises and in the cloud, as well as to a variety of SQL Server versions and editions.

All components of SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server on Azure VM, and Azure Synapse Analytics may be accessed, configured, managed, administered, and developed using SSMS. SSMS is a single, complete application that combines a wide range of graphical tools with several powerful script editors to provide developers and database administrators of all skill levels access to SQL Server.

SSMS provides the easy environment to add, edit, replicate, and delete databases which are connected to the server by Microsoft Azure.

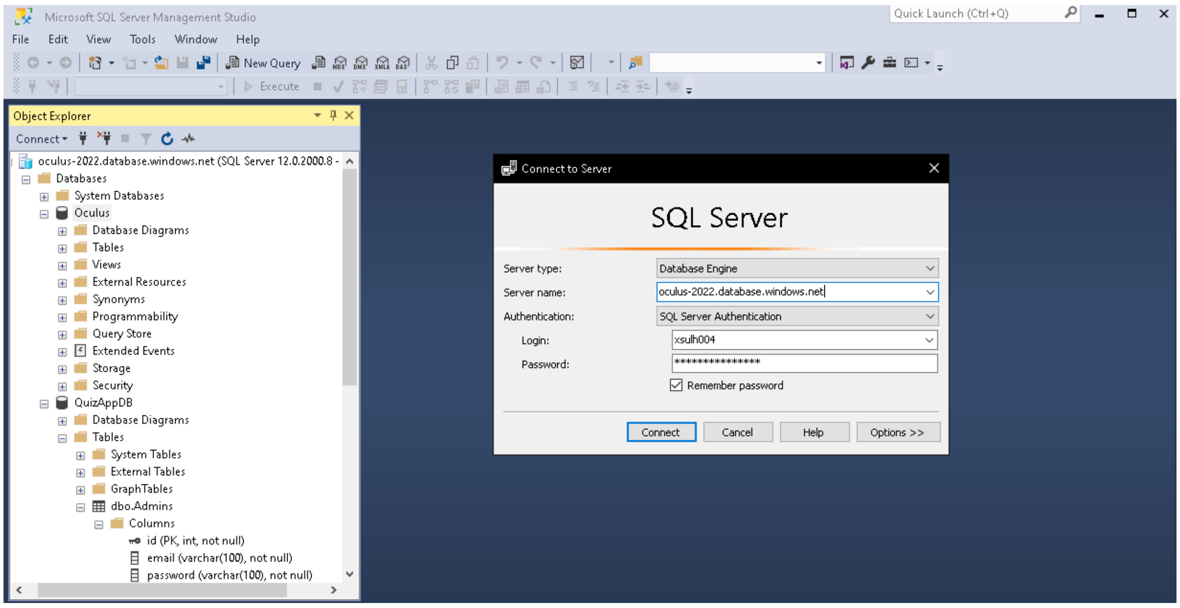
4.4.1 The benefits of SSMS

Many database activities may be performed with SQL Server Management Studio (SSMS), including the following:

- Create or change anything quickly. a database in SQL.
- adding ables, views, and stored procedures are among the database objects.
- Tables, views, and stored procedures are among the database objects that may be modified.
- External testing tools can be used to test database items.
- Deploying database to their appropriate onshore, remote, or cloud-based server environments.
- receiving results, query the database(s).
- Improve response speed by optimizing databases.
- Create database or object scripts by importing or exporting data
- Database backups
- Databases restoring

To access to Azure database using SQL Server Management Studio, the connection must be SQL Server Authentication, to gain the full access of the database.

As the figure below shows the required connection between Azure and SSMS.



oculus-2022.database.windows.net (SQL Server 12.0.2000.8 -
Databases
System Databases
Oculus
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Query Store
Extended Events
Storage
Security
QuizAppDB
Database Diagrams
Tables
System Tables
External Tables
GraphTables
dbo.Admins
Columns
id (PK, int, not null)
email (varchar(100), not null)
password (varchar(100), not null)

Server type:	Database Engine
Server name:	oculus-2022.database.windows.net
Authentication:	SQL Server Authentication
Login:	xsulh004
Password:	***************

[✓] Remember password

Figure 6 Connect SSMS to Azure (Source author)

5 Practical Part

5.1 WPF User Interface Components

Understanding the components of WPF UI is the fundamental of designing any WPF application. Windows application Foundation consists of five major components:

5.1.1 Solution Explorer

This pane will display all project files, code, windows, and resources. Such the data handler, external files, and design patterns.

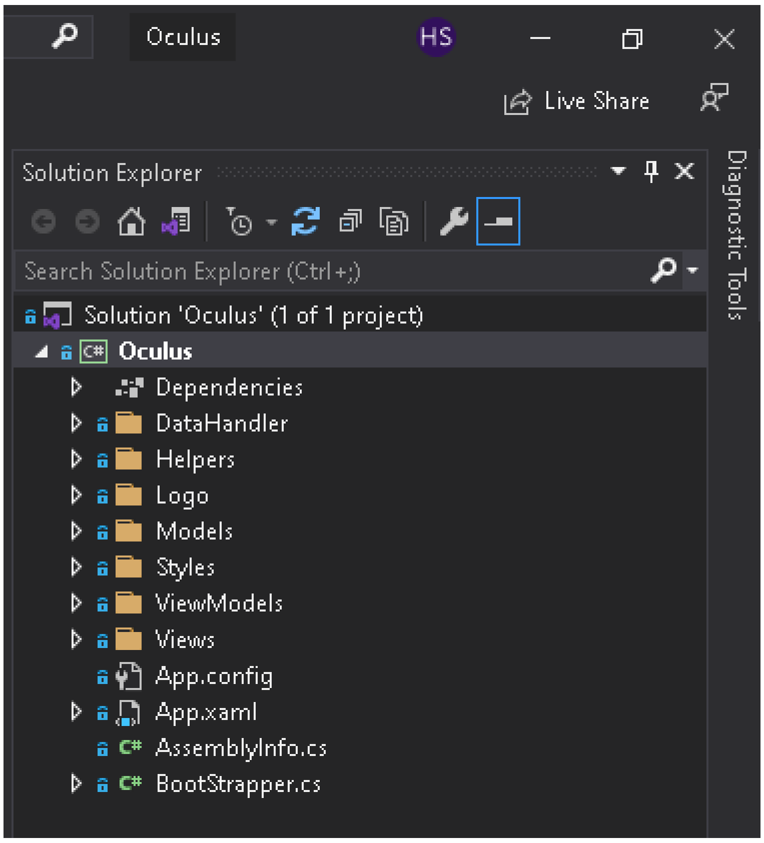
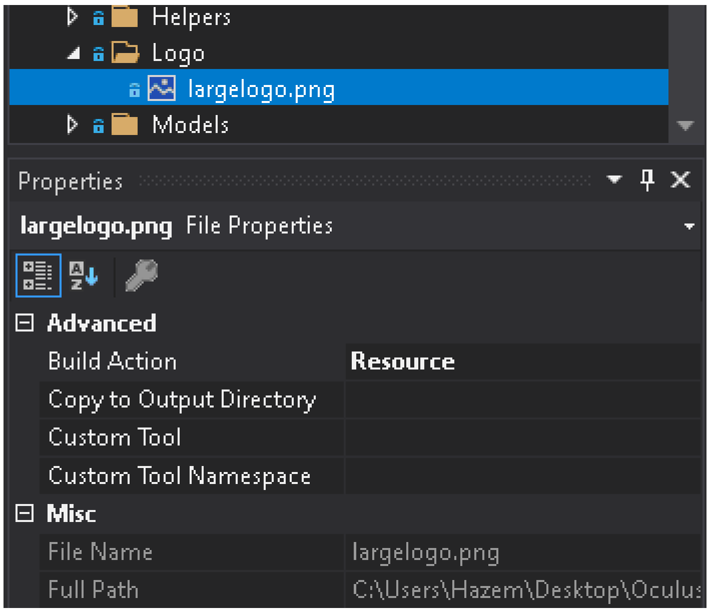


Figure 7: Solution Explorer in Oculus

The Solution Explorer gives the easy navigation between the code parts. It consists also of two main files, the assembly Information file and the application configuration

5.1.2 Properties

This pane displays property settings that we may change based on the object we have chosen., for example, for example we needed to change the file's property settings of an external file such as logo. To implement a picture in box in WPF, the Build Action must be Resource, Not NONE.



Build Action	Resource
Copy to Output Directory	
Custom Tool	
Custom Tool Namespace	
File Name	largelogo.png
Full Path	C:\Users\Hazem\Desktop\Oculus

Figure 8: Property Component in Oculus

5.1.3 Toolbox

All of the controls that may be added to a form are found in the toolbox. Double-click or drag-and-drop a control onto the current form to add it to it.

5.1.4 XAML designer

This is the XAML document's designer. It's interactive, and you can use the Toolbox to drag and drop things. You may visually create the user interface (UI) for your app by choosing and moving components in the designer.

XAML is the design system for building the interface, The designer window builds on XML essentially, written by extensible application Markup Language.

Every button, box, textblock, etc will on the form will be show as XAML code in the designer window will be show in the fifth component, XAML code editor

5.1.5 XAML code editor

For a XAML document, this is the XAML code editor. The XAML code editor allows you to create your UI without the help of a designer.

The editor work side to side with the designer, by clicking on an element on the form, the code will be shown simply and clear to edit and add feature to the design

For example: adjust the margin of the *PasswordBox*., the editor controls the elements attributes with on-time edit shown on the designer, any change on the code in the editor will correspond to the element immediately.

```
<PasswordBox Grid.Row=”1”
             Grid.Column=”1”
             Width=”220”
             Margin=”0,0,0,20”
             Padding=”5”
             VerticalAlignment=”Center”>
```

Source code 1Source Code 1 (By Author) 1

5.2 WPF Controls

5.2.1 TextBox Control

Text block is very basic controls which almost guaranteed to use in every application, and it is very straightforward control. it's a very useful control for displaying text in your application, it’s generally used to display plain text to users more specifically to title controls in your application title pages in your application display warnings alerts in tips to your users or just provide general instructions to users so after this brief explanation.

The class *TextBlock* is one class of many classes of the namespace *System.Windows.Controls.*

```
[System.Windows.Localizability(System.Windows.LocalizationCategory.Text)]
[System.Windows.Markup.ContentProperty("Inlines")] public class TextBlock
: System.Windows.FrameworkElement, IServiceProvider,
System.Windows.IContentHost
```

5.2.2 TextBlock Control

A *TextBlock* can have a string in its Text property or Inline flow content items in its *Inlines* property, such as Bold, Hyperlink, and *InlineUIContainer*.

TextBlock is meant to be compact and lightweight, with the goal of incorporating short bits of flow material into a user interface (UI). *TextBlock* is optimized for single-line display and can display up to a few lines of material with decent speed.

TextBlock is not optimized for scenarios that require more than a few lines of material to be shown; in these cases, a *FlowDocument* combined with an appropriate viewing control is a better solution in terms of performance than *TextBlock*. After *TextBlock*, the next lightest-weight control for showing flow content is *FlowDocumentScrollViewer*, which just provides a scrolling content area with little UI. For flow content, *FlowDocumentPageViewer* is optimized for "page-at-a-time" viewing mode. Finally, *FlowDocumentReader* has the most comprehensive collection of features for reading flow material, but it is also the heaviest. The *TextAlignment* property is used to align text horizontally within a *TextBlock*. The *HorizontalAlignment* and *VerticalAlignment* attributes are used to align the *TextBlock* inside the page's layout.

In Oculus UI, Email and Password elements are *TextBlock*.

```
<TextBlock
                Grid.Row="0"
                Grid.Column="0"
                Margin="0,0,15,20"
                VerticalAlignment="Center"
                FontSize="14"
                FontWeight="DemiBold"
                Text="Email" />
```

Source Code 2 (By Author) 1

5.2.3 Button Control

By simply adding a Button tag to Window, it may show a Button. If we add text (or another control) between the tags, it will serve as the Button's content. The Button doesn't do anything yet but hovering over it will produce an effect; it comes with a lovely hover effect out of the box. Giving action to its Click event, on the other hand.

Such the sign in *Button* in Oculus

```
<Button
                    x:Name="SignInCommand"
                    cal:Message.Attach="[Event Click] = [Action
RunOperation($source)]"
                    materialDesign:ButtonAssist.CornerRadius="5"
                    BorderThickness="0"
                    Content="Sign in" />
```

Source Code 3 (By Author) 1

5.2.4 PasswordBox

WPF has a built-in *PasswordBox* widget for handling and managing passwords in WPF applications. A *TextBox* control with the Masking functionality activated as a *PasswordBox* control. You may conceal characters and restrict the number of characters that can be input in the editable area using the *PasswordBox* control.

The code below shows how to use XAML and C# to build and utilize a *PasswordBox* control in WPF.

A *PasswordBox* control is represented by the *PasswordBox* element in XAML. The following code sample creates a *PasswordBox* and customizes its Height, Width, Foreground, and Background attributes.

By using the CheckVBox functionality, the application should display a message box to the student if the password is incorrect.

Near the bottom of the *MessageBoxEx*.xaml.cs file is the *MsgBoxExCheckBoxData* class. It appears as follows:

```
cal:Message.Attach="[Event PasswordChanged] = [Action
OnPasswordChanged($source)]"
                materialDesign:TextFieldAssist.UnderlineBrush="Transparent"
                Background="LightGray" />
```

Source Code 4 (By Athor) 1

5.3 Implementing the logo

Oculus is the Latin word for eye. The Oculus logo refers to the interrelationship between the eye of the observer and the eye of the camera.

Implementing a picture in a class Image in WPF requires creating folder in Solution Explorer and import the images we need to insert as a logo or wallpaper.

5.4 MVVM in WPF

Two parts have been created during the process

- ❖ XAML File (*MainWindow*.xaml)
- ❖ CS File (*MainWindow*.xaml.cs)

First part is the user interface part *MainWindow*.xmal consists of:
the login window contains details of:

1. App Titles
2. Text to provide the examinees with the important information
3. Email (student’s school email)
4. Password (student’s password)
5. Admin Access
6. Login button (to start the exam)

Second part is the logic part *MainWindow.xaml*.cs where the C# code will be implemented to create the login window by using buttons, opening tabs, and other windows

VIEW: A View is defined in XAML and should not include any code-behind logic. It solely uses data binding to connect to the view-model. For the purpose of the visual presentation, the View comprises the visual controls that will be presented to the user, as well as animations, navigation elements, themes, and other interactive features. The Model is directly data linked to the View. One-way data binding is used to display parts of the Model in the view. By directly tying controls two-way to the data, other sections of the model may be modified. It serves as a link between your program and the people who use it.

MODEL: The Model is in charge of delivering data in a WPF-friendly format. As applicable, it must implement INotifyPropertyChanged and/or INotifyCollectionChanged.

When retrieving data is time-consuming, it separates the time-consuming processes away from the user interface thread. The data or business logic that saves the state and processes the issue domain is fully independent of the user interface. The Model is either programmed or pure data recorded in relational tables or XML.

VIEWMODEL: A *ViewModel* is an application's model for a view, or an abstraction of the view. It commonly uses Commands to expose data related to the view and actions for the views. The *ViewModel* connects the View to the rest of the world. The *ViewModel* is the object to which the View is attached. It offers a Model specialization that the View may utilize to bind data.

WPF's powerful binding features are largely used by MVVM to eliminate some of the plumbing code, allowing us to focus on the logical model of our view, or the *ViewModel*, and our application. The purpose of MVVM is to isolate the application's design from its data and functionality, while also supporting several programming languages (such as C#/XAML) and utilizing extensive databinding.

In the order of making the entities that go with the database the following command must run in Package Manager Console (PM):

```
Scaffold-DbContext "Server=<NameServer>; Database=<NameDB>;
Trusted_Connection=True;"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Entities
```

5.5 Solution Explorer Structure

Beside *Model*, *View*, *ViewModel*, external files, Style (design UI folder), the explorer consists of *DataHandler* which used to provide access to data used in reports.

The structure must have parts such as md5Hash a cryptographic mechanism for message authentication, information verification, and user validation.

The user (student) Model must have references as they are classes have been mentioned in another projects.

```
class UserModel
    {
        public string email { get; set; }
        public string password { get; set; }
    }
```

Source Code 5 (By Author) 1

The *app.confg* has the new connection string which created by SSMS

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>

 <connectionStrings>

   <add name="QuizDB" connectionString="Server=tcp:oculus-2022.database.windows.net,1433;Initial Catalog=QuizAppDB;Persist Security Info=False;User ID=xsulh004;Password=23-35@Ta;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;" providerName="System.Data.SqlClient.SqlConnection" />

 </connectionStrings>


</configuration>
```

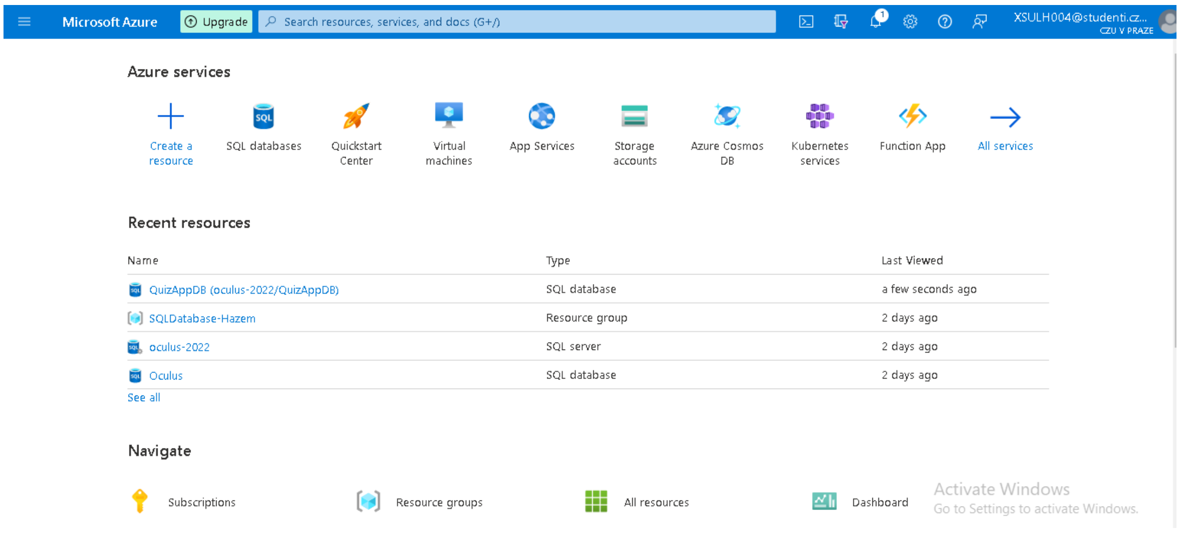
Source Code 6 (By Author) 1

5.6 Connect database with WPF

Microsoft Azure offers two types of cloud services

Public clouds, which provide computing resources such as servers and storage through the Internet, are owned and operated by third-party cloud service providers. One example is a public cloud like Microsoft Azure. In a public cloud, the cloud provider owns and operates the hardware, software, and other supporting infrastructure. To utilize these services and manage your account, you'll need a web browser.

A private cloud is a set of cloud computing services that are only used by one enterprise or organization. A private cloud might be physically located at a business's datacentre. Some companies also pay for third-party service providers to host their private clouds. A private cloud's services and infrastructure. (16)



Azure services

Create a resource · SQL databases · Quickstart Center · Virtual machines · App Services · Storage accounts · Azure Cosmos DB · Kubernetes services · Function App · All services

Recent resources

Name	Type	Last Viewed
QuizAppDB (oculus-2022/QuizAppDB)	SQL database	a few seconds ago
SQLDatabase-Hazem	Resource group	2 days ago
oculus-2022	SQL server	2 days ago
Oculus	SQL database	2 days ago

See all

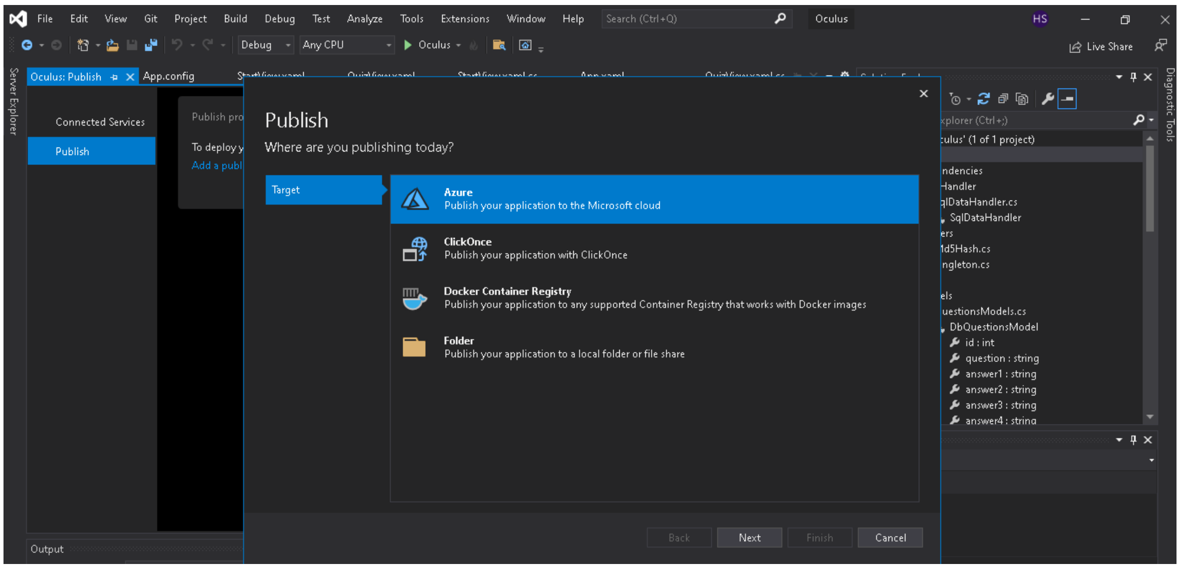
Navigate

Subscriptions · Resource groups · All resources · Dashboard

Activate Windows
Go to Settings to activate Windows.

Figure 9 Creating Azure account

The next step is publishing the project from Visual Studio to Azure



Publish

Where are you publishing today?

Target

Azure
Publish your application to the Microsoft cloud

ClickOnce
Publish your application with ClickOnce

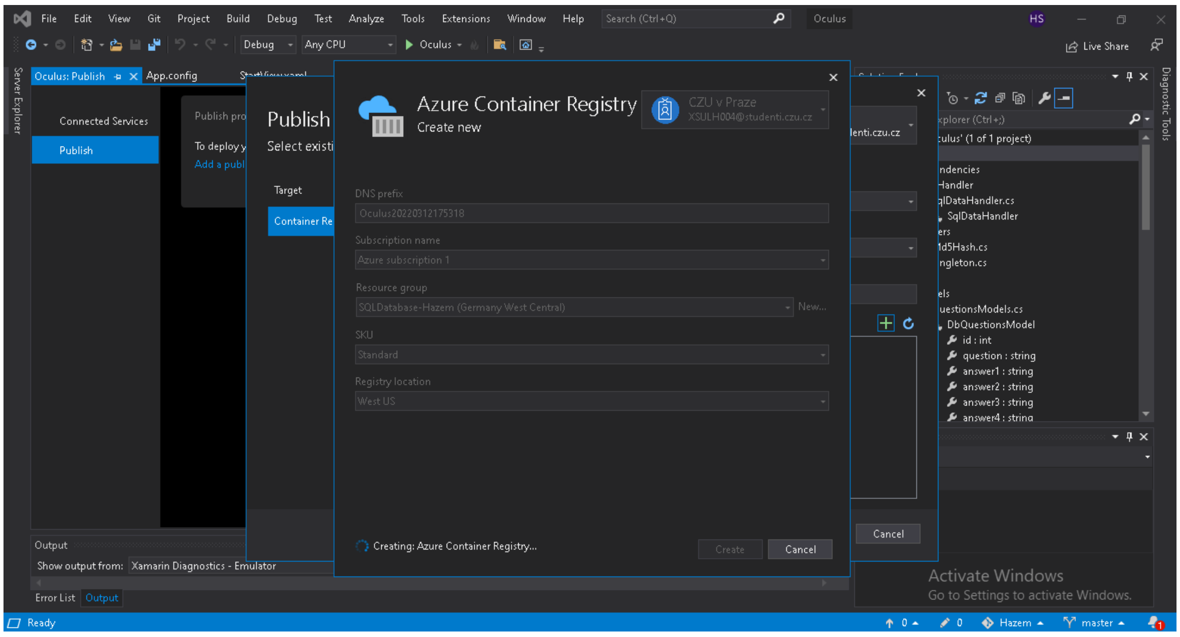
Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images

Folder
Publish your application to a local folder or file share

Back · Next · Finish · Cancel

Figure 10 Publishing Oculus on Azure

After choosing the right service – Azure



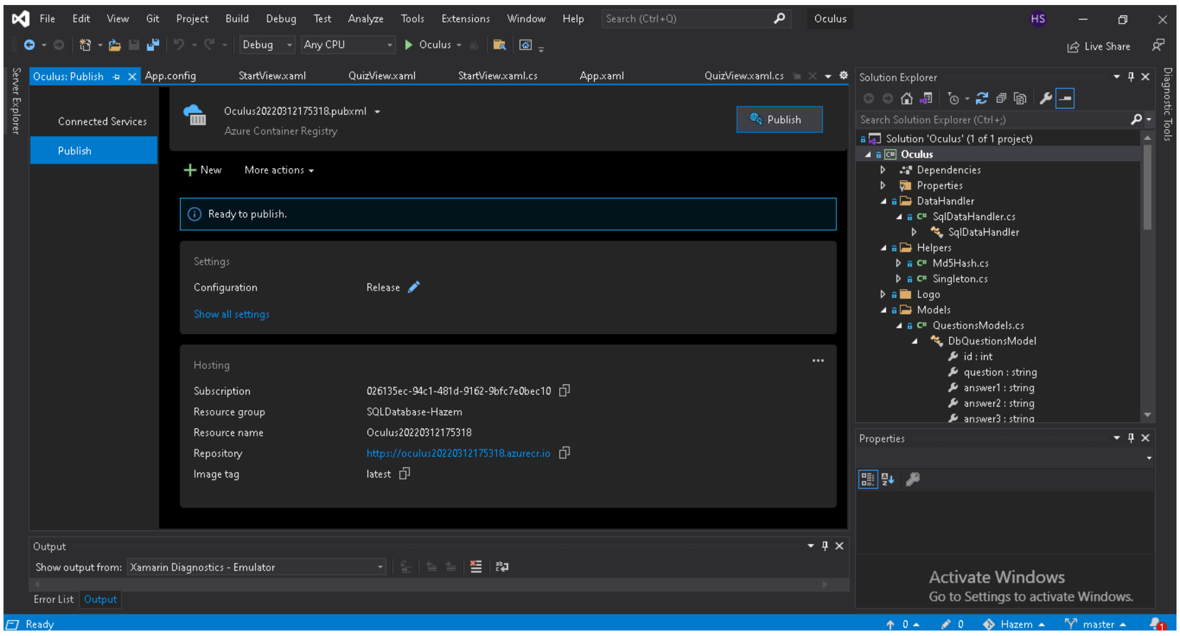
Azure Container Registry
Create new
CZU v Praze
XSULH004@studenti.czu.cz

DNS prefix
Oculus20220312175318
Subscription name
Azure subscription 1
Resource group
SQLDatabase-Hazem (Germany West Central)
SKU
Standard
Registry location
West US

Creating: Azure Container Registry...

Figure 11Connect to Azure resource

The resource that has been created during the registration in the first step.



Oculus20220312175318.pubxml
Azure Container Registry

Ready to publish.

Settings
Configuration Release

Hosting
Subscription 026135ec-94c1-481d-9162-9bfc7e0bec10
Resource group SQLDatabase-Hazem
Resource name Oculus20220312175318
Repository https://oculus20220312175318.azurecr.io
Image tag latest

Figure 12Successful connection with Azure

After adding the correct information, Visual studio will be connected to Microsoft Azure With unique sever URL: [`https://oculus20220312175318.azurecr.io`](https://oculus20220312175318.azurecr.io)

6 The aspiring idea of the application

For starter, the application meant to increase the application meant to increase the accessibility of the teachers during the exams all come to mind after witnessing the crooked ways that have been used by some student in the last two years, taking advantage of the unique circumstances which affected the educational institutions methods.

To achieve such control, it must be built first as client server application first and improve it later with more technologies to achieve the needed control during the use of the application.

The technologies that have been use in this application so far are:

1. C# as the OOP language that has been used as the logic of the application.
2. XAML as the descriptive language that has been used for designing.
3. MVVM as the architecture pattern that has been used in this application.
4. SQL the language that has been used to communicate with the database.
5. Microsoft Azure as the technology for storing the application data.

6.1 Analysis the process

The function of the application consists of many parts, some of them have been designed and programmed, and some of them using the features of existed technologies.

6.1.1 UI Specification:

This stage of the process required coding part, using WPF is going to be used to create graphical user interfaces like I said and Xaml which is going to be used to create the UI using sort of controls that allow to define buttons boxes animation 2d 3d graphics and more.

After creating a WPF project, the first thing to do is to set the controls we need: Including Images, TextBlocks, TextBoxs, PasswordBoxs, and Buttons, etc
Some of the control has events.

6.1.2 Sign in and Sign up commands

The sign In design in Xaml as we explainded, but the has been written in the code behind, as a logic of the event.

```
public void SignInCommand()
        {
            if (string.IsNullOrEmpty(Password) || string.IsNullOrEmpty(Email))
            {
                MessageBox.Show("Fields Empty");
                return;
            }

            var sqlDataHandler = new SqlDataHandler();
            var status = sqlDataHandler.VerifyUser(new Models.UserModel { email =
Email, password = Md5Hash.MD5Hash(Password) });

            if (status)
            {
                var conductor = this.Parent as IConductor;
                conductor.ActivateItemAsync(new QuizViewModel());
            }
            else
                MessageBox.Show("user not found");
        }
    }
}
```

Source Code (By author) 1

The attribute event will run the code above. Which explain the following:

Return the value if the email and the password were added as a blank.

Check the validation of the user email and password by sending request to the database.

With return value “User not found” if the user isn’t created in the database system.

The implementation of the following code:

```
public void SignUpCommand()
        {
            var conductor = this.Parent as IConductor;
            conductor.ActivateItemAsync(new UserRegViewModel());
        }
```

source Code (By Author) 2

Will set request to register a new user in the database. The event sets an argument to create a new object in the database.

Considering returning an error message in case of insert invalid email address.

The following code sample to create a new username in SQL:

```
SqlConnection con = new SqlConnection("Data Source=TESTPURU;Initial Catalog=Data;User ID=sa;Password=wintellect");
```

```
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * from Registration where Email='" + email + "' and password='" + password + "'", con);
                cmd.CommandType = CommandType.Text;
                SqlDataAdapter adapter = new SqlDataAdapter();
                adapter.SelectCommand = cmd;
                DataSet dataSet = new DataSet();
```

Source Code 5 (By Author) 2

6.1.3 Admin Command

The same methods like the previous commands, but the logic behind is different since the Admin.View.Model is the program logic action loaded from database.

The sample of the following code sent a request to the database to retrieve the admin table in the database.

```
class AdminLoginViewModel : Screen
    {
        private string _email { get; set; }
        public string Email { get => _email; set { _email = value; 
NotifyOfPropertyChange(); } }

        private string Password = string.Empty;
        public AdminLoginViewModel()
```

Source Code 5 (By Author) 3

6.2 Database Specification

6.2.1 Prerequisites

Under the namespace QuizApplication Model.

Implementing the following code sets the functionality of generating questions from database

```
class DbQuestionsModel
    {
        public int id { get; set; }
        public string question { get; set; }
        public string answer1 { get; set; }
        public string answer2 { get; set; }
        public string answer3 { get; set; }
        public string answer4 { get; set; }
        public string correctAns { get; set; }
        public bool IsMatch { get; set; } = false;
        public int? SelectedIndex { get; set; }
    }

}
```

Source Code 6 (By Author) 2

Implementing AddQuestions Class inserts new questions into the database

```
public AddQuestionViewModel()
        {
            Iscorrect1 = true;
        }

        public AddQuestionViewModel(DbQuestionsModel dbQuestionsModel)
        {
            isUpdate = true;
            id = dbQuestionsModel.id;
            Question = dbQuestionsModel.question;
            Answer1 = dbQuestionsModel.answer1;
            Answer2 = dbQuestionsModel.answer2;
            Answer3 = dbQuestionsModel.answer3;
            Answer4 = dbQuestionsModel.answer4;
```

Source Code 7 (By Author) 1

And the following class sets the correct answer:

```
            SqlDataHandler sqlDataHandler = new SqlDataHandler();
            DbQuestionsModel dbQuestionsModel = new DbQuestionsModel
            {
                question = Question,
                answer1 = Answer1,
                answer2 = Answer2,
                answer3 = Answer3,
                answer4 = Answer4,
                correctAns = CorrectAnswer,
```

Source Code 8 1

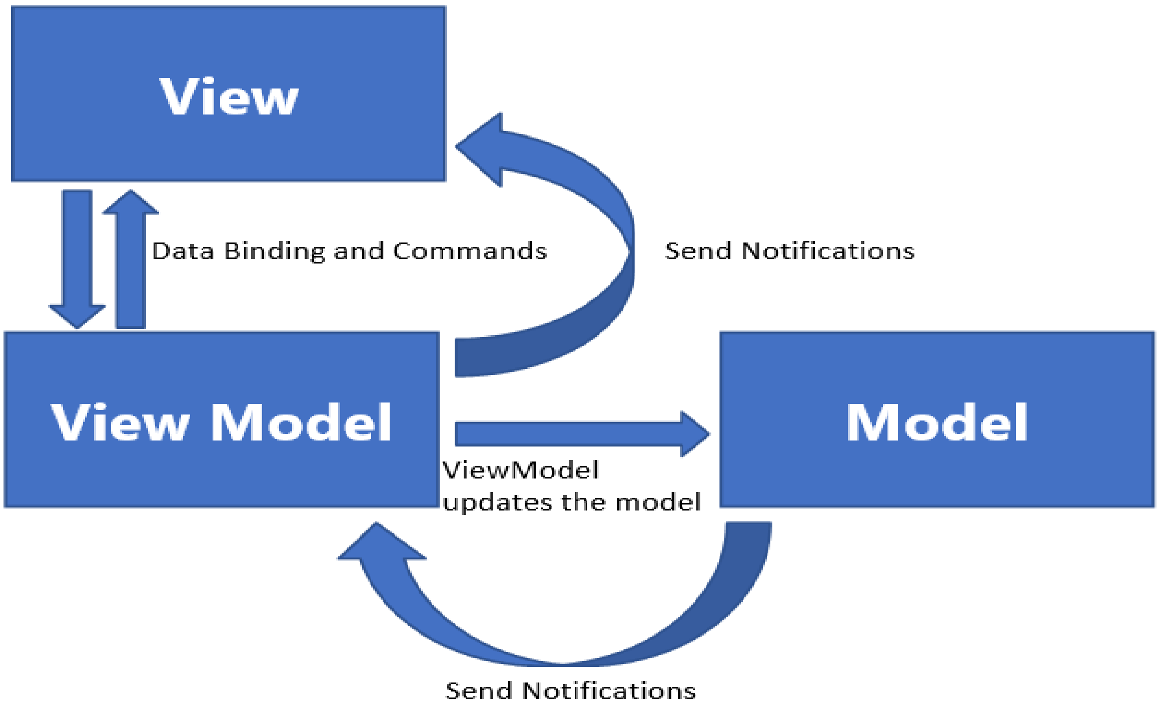
Using Microsoft SQL Server Management Studio, we can easily import/Export Database from the server.

6.3 The function of the architecture

The biggest hurdle for reusability is that the complete code is tied up with the UI Technology, how do we solve this Reusability problem How do we make the code Behind reusable

the answer Move the Behind Code to the Class Library The way to solve the problem is if you want to make something reusable, we must create a DLL In other word you have to create a Class.

The solution is basically whatever is you XAML we will create a view out of it whatever is a Behind Code that is a XAML.CS will create a View Model Class out of this Why the name is View Model because it has the logic with binds The View and The Model All the Glue Code will move it into a class library Once the code is moved it to a class library we can reuse it anywhere we have to just add a reference add the DLL reference and start using it.



MVVM design pattern in WPF (17)

7 Results and Discussion

7.1 Results

The application has been tested, all the features that has been mentioned are functional and logically correct.

The testing methods consists of two stages, user experience and hypothetical process.

The UX was proved by following the procedure with the functional buttons, opening the target windows, having the right authorization, and creating the requested data in the database.

The hypothesis centres around the experience that will be given to the user after inserting the available technologies to add the targeted functionality of the application

7.2 Discussion

The ultimate possibility to improve the application comes with the idea of implementing technologies such as real-time monitoring by C# or another programming languages.

In addition, implementing artificial intelligence methods to detection such as monitoring camera, microphone, keyboard, etc.

Improving the project will be supported by real-life testing and reporting, using set of testers from both sides, students, and teachers.

8 Conclusion

The aim of the thesis is to create desktop application using WPF for online examination, adding the required features to create the exam environment.

And introduce .Net Framework and the implementation of the architecture of Model-View-ViewModel. Databinding, and the design patterns.

Deploying the database on cloud server using Microsoft Azure, connecting the cloud database to the application.

The application is set for improvement, as structure able to be edited to handle the required technologies to reach the ultimate objectives.

In other hand, the application is ready to be tested and be used through the functionality that has been mentioned through the thesis.

9 Bibliography

1. **Microsoft.** https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview. *Microsoft Developer Network.* [Online]
2. **Sharp, John.** *Microsoft Visual C# Step by Step, 9th Edition.* CA, USA : Microsoft Press, 2018.
3. **MacDonald, Matthew.** *Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5.* Toronto, Ontario, Canada. : Apress, 2012. 9781430243656.
4. **Lalonde, Buddy James and Lori.** *Pro XAML with C#: Application Development Strategies (covers WPF, Windows 8.1, and Windows Phone 8.1) 1st ed. Edition.* New York City, USA : Apress; 1st ed. edition (July 1, 2015), 2015.
5. **MY.NET Tutorials. *MY.NET Tutorials.* [Online] https://www.msdotnet.co.in.**
6. https://www.c-sharpcorner.com/article/wpf-architecture/. C# corner. [Online]
7. https://prismlibrary.com/docs/wpf/view-composition.html. Prism Documentation. [Online]
8. Griffiths, Ian. *Programming C# 8.0: Build Cloud, Web, and Desktop Applications 1st Edition.* London, United Kingdom : O'Reilly Media; 1st edition (January 14, 2020), 2020.
9. https://devblogs.microsoft.com. Microsoft Blog. [Online]
10. https://docs.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-6.0. [Online]
11. https://www.mongodb.com/database-hosting. [Online]
12. SQL Server performance tuning | How is data stored in SQL database. *Pragim Tech | Software Training institutes in Bangalore.* [Online] Pragim Tech. https://www.pragimtech.com/blog/sql-optimization/how-is-data-stored-in-sql-database/.
13. https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5. [Online]
14. What is PaaS? Platform as a Service | Microsoft Azure. *Cloud Computing Services | Microsoft Azure.* [Online] Microsoft. https://azure.microsoft.com/en-us/overview/what-is-paas/.
15. https://azure.microsoft.com/en-us/overview/what-is-azure/. [Online]
16. Cloud Computing Services | Microsoft Azure. *What is cloud computing? A beginner's guide | Microsoft Azure.* [Online] https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/#cloud-deployment-types.
17. Design Pattern, MVVM. *C# Corner - Community of Software and Data Developers.* [Online] https://www.c-sharpcorner.com/article/design-pattern-2-mvvm/.
18. Johannsen, Joseph Albahari and Eric. *C# 8.0 in a Nutshell.* United States of America. : O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2020.
19. https://www.msdotnet.co.in/2014/06/the-architecture-of-wpf.html#.YirLBeiZPIU. [Online]
20. TextBlock Class (System.Windows.Controls) | Microsoft Docs. *Developer tools, technical documentation and coding examples | Microsoft Docs.* [Online]

10 List of pictures, tables, graphs and abbreviations

10.1 List of figures

Appendix

Part of the required appendix for testing.

The existed registered emails in the database:

Student emails:

Xsulh004@studenti.czu.cz Password: wpf-123

hazem@gmail.com Password: wpf-123

Admin email:

brozek@gmail.com Password: test-123

Cloud database:

Microsoft Azure created with the email xsulh004@studenti.czu.cz

Resource name: SQLDatabase-Hazem

Server name Oculus

Login: xsulh004

Password: 23-35@Ta

Access to the database through SSMS

SQL server authentication:

Oculus-2022.database.windows.net

With the same login and password as Azure