

Київський національний торговельно-економічний університет

Кафедра програмної інженерії та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

Розробка довідково-інформаційної системи КНТЕУ «Студент»

Студента 4 курсу, 7 групи
спеціальності (6. 050103
«Програмна інженерія»)
спеціалізації Інженерія
програмного забезпечення

підпис студента

Тіхонов Антон Олегович

Науковий керівник

кандидат технічних наук,
доцент

підпис керівника

Цюцюра Миколай Ігорович

Гарант освітньої програми

кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ – 2019

ЗМІСТ

Вступ.....	3
Розділ 1 Інформація щодо використаних інструментів та методів.....	4
1.1. Задіяні мови програмування.....	4
1.2. Патерн програмування MVVM.....	5
1.2.1. Мова розмітки XAML.....	7
1.3. Формування технічного завдання.....	8
1.4 Висновки до розділу 1.....	8
Розділ 2 Проектування системи.....	10
2.1. Проектування бази даних для проекту.....	10
2.1.1. Опис моделі взаємодії у базі даних.....	12
2.2. Систематизація структури проекту згідно патерну MVVM.....	14
2.3. Розробка користувацького інтерфейсу.....	15
2.4. Висновки до розділу 2.....	18
Розділ 3 Програмна реалізація додатку.....	19
3.1. Коннектори до бази даних MySql.....	19
3.2. Реалізація візуальних стилів з допомогою XAML.....	21
3.3. Реалізація логічної моделі.....	22
Висновки до розділу 3.....	23
Висновки та пропозиції.....	24
Список використаних джерел.....	25
Додаток А.....	26

ВСТУП

В сучасному світі, інформаційні технології розвиваються дуже стрімко, та іноді важко залишитись в тренді сучасних методології, патернів, мов програмування, але це необхідно робити щоб залишатись гарним спеціалістом у своїй галузі та мати високу фінансову оцінку на сучасному ринку праці.

На створення будь-якого додатку, програмного забезпечення може наштовхнути ціла низка факторів починаючи від самореалізації своїх інтересів в тій чи іншій сфері і закінчуючи елементарним фінансовим забезпеченням.

Метою цієї випускної кваліфікаційної роботи є демонстрація здобутих навичок та вмінь протягом всього навчального курсу за спеціальність «Програмна інженерія», саме тому тут треба використовувати та реалізувати знання з різних галузей розробки програмного забезпечення.

Завданням до випускної кваліфікаційної роботи є створення довідникової-інформаційної системи КНТЕУ «Студент».

Об'єктом дослідження є освітній процес.

Предметом дослідження є створення інформаційно-довідкової системи університету.

					<i>КНТЕУ 6.050103 07-23.БР</i>			
					<i>Розробка довідково-інформаційної системи КНТЕУ «Студент»</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>В</i>	<i>3</i>	<i>25</i>
Зав.каф.		Криворучко О.В.						
Керівник		Цюцюра М.І.						
Гарант		Цензура М.О.						
Розробив		Тіхонов А.О.			<i>Вступ</i>	<i>Факультет обліку, аудиту та інформаційних систем, 4 курсу, 7 групи</i>		

Розділ 1 Інформація щодо використаних інструментів та методів

1.1. Задіяні мови програмування

Щоб розробити будь який десктопний додаток, необхідно визначитись з мовою програмування та платформою, на базі якого вона повинна буде працювати. Програмне забезпечення завжди має дві сторони: бекенд та фронтенд.

Фронтенд – це користувацька частина програми, за допомогою якої користувач звертається до програмно-апаратної частини, ще можна назвати це інтерфейсом.

Бекенд – це програмно-апаратна частина додатку, те що не бачить користувач, але те через що все працює так як потрібно.

В даній випускній кваліфікаційній роботі застосовано мова програмування C#, мова структурованих запитів SQL(англ. structured query language) та мова розмітки додатків XAML(англ. eXtensible Application Markup Language). До бекенду в цій структурі відноситься мова програмування C# та мова структурованих запитів SQL, а за інтерфейс відповідає мова розмітки додатків XAML. Проект реалізований на базі ядра .NET Framework версії 4.7.2, що дозволяє користуватись свіжими оновленнями платформи та писати сучасний код. База даних розташована на системі керування реляційними базами даними MySql версії 8.0.

Сукупність цих технологій можна назвати стеком, тому що кожен елемент взаємодіє один з одним та на виході ми маємо готовий робочий додаток. Якщо убрати зі стеку одну з технологій, то інша частина розробки буде неспроможна задовольнити вимоги користувача.

					<i>КНТЕУ 6.050103 07-23.БР</i>			
					Розробка довідково-інформаційної системи КНТЕУ «Студент»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		P1	4	25
Зав.каф.		Криворучко О.В.						
Керівник		Цюцюра М.І.						
Гарант		Цензура М.О.						
Розробив		Тіхонов А.О.			Розділ 1	Факультет обліку, аудиту та інформаційних систем, 4 курсу, 7 групи		

1.2. Патерн програмування MVVM

Патерн MVVM (Model-View-ViewModel) дозволяє відокремити логіку програми від візуальної частини (подання). Даний патерн є архітектурним, тобто він задає загальну архітектуру програми.

Даний патерн був представлений Джоном Госсманом (John Gossman) в 2005 році як модифікація шаблону Presentation Model і був спочатку націлений на розробку програм в WPF. І хоча зараз даний патерн вийшов за межі WPF і застосовується в самих різних технологіях, в тому числі при розробці під Android, iOS, тим не менш WPF є досить показовою технологією, яка розкриває можливості даного патерну.

MVVM складається з трьох компонентів: моделі (Model), моделі представлення (ViewModel) і подання (View) (рис.1).



Рис.1 модель взаємодії MVVM

Model описує використовувані в додатку дані. Моделі можуть містити логіку, безпосередньо пов'язану з цими даними, наприклад, логіку валідації властивостей моделі. У той же час модель не повинна містити ніякої логіки, пов'язаної з відображенням даних і взаємодією з візуальними елементами управління.

Нерідко модель реалізує інтерфейси `INotifyPropertyChanged` або `INotifyCollectionChanged`, які дозволяють повідомляти систему про зміни властивостей моделі. Завдяки цьому полегшується прив'язка до подання, хоча знову ж пряма взаємодія між моделлю і поданням відсутня.

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		5

View або подання визначає візуальний інтерфейс, через який користувач взаємодіє з додатком. Стосовно WPF подання-це код в xaml, який визначає інтерфейс у вигляді кнопок, текстових полів та інших візуальних елементів.

Хоча вікно (клас Window) в WPF може містити як інтерфейс в xaml, так і прив'язаний до нього код C#, проте в ідеалі код C# не повинен містити якоїсь логіки, крім хіба що конструктора, який викликає метод InitializeComponent і виконує початкову ініціалізацію вікна. Вся ж основна логіка програми виноситься в компонент ViewModel.

Однак іноді у файлі пов'язаного коду всеж може знаходитися деяка логіка, яку важко реалізувати в рамках патерну MVVM у ViewModel.

Уявлення не обробляє події за рідкісним винятком, а виконує дії в основному за допомогою команд.

ViewModel або модель представлення пов'язує модель і уявлення через механізм прив'язки даних. Якщо в моделі змінюються значення властивостей, при реалізації моделлю інтерфейсу INotifyPropertyChanged автоматично йде зміна даних, які відображаються в поданні, хоча безпосередньо модель і уявлення не пов'язані.

ViewModel також містить логіку по отриманню даних з моделі, які потім передаються в уявлення. І також viewmodel визначає логіку по оновленню даних в моделі.

Оскільки елементи подання, тобто візуальні компоненти типу кнопок, не використовують події, то подання взаємодіє з ViewModel за допомогою команд.

Наприклад, користувач хоче зберегти введені в текстове поле дані. Він натискає на кнопку і тим самим відправляє команду в ViewModel. А ViewModel вже отримує передані дані і відповідно до них оновлює модель.

					КНТЕУ 6.050103 07-23.БР	Аркуш
						6
Изм.	Аркуш	№ докум	Подпись	Дата		

Підсумком застосування патерну MVVM є функціональний поділ програми на три компоненти, які простіше розробляти і тестувати, а також надалі модифікувати і підтримувати. MVVM є функціональний поділ програми на три компоненти, які простіше розробляти і тестувати, а також надалі модифікувати і підтримувати.

1.2.1. Мова розмітки XAML

XAML (eXtensible Application Markup Language) - мова розмітки, яка використовується для ініціалізації об'єктів в технологіях на платформі .NET. Стосовно до WPF (а також до Silverlight) ця мова використовується насамперед для створення користувацького інтерфейсу декларативним шляхом. Хоча функціональність XAML тільки графічними інтерфейсами не обмежується: ця мова також використовується в технологіях WCF і WF, де він ніяк не пов'язаний з графічним інтерфейсом. Тобто його область ширше.

Стосовно до WPF ми будемо говорити про нього найчастіше саме як про мову розмітки, яка дозволяє створювати декларативним шляхом інтерфейс, зразок HTML в веб-програмуванні.

XAML - не є обов'язковою частиною програми, ми взагалі можемо обходитися без нього, створюючи всі елементи у файлі пов'язаного з ним коду на мові C#. Однак використання XAML все-таки несе деякі переваги:

- Можливість відокремити графічний інтерфейс від логіки програми, завдяки чому над різними частинами програми можуть відносно автономно працювати різні фахівці: над інтерфейсом - дизайнери, над кодом логіки - програмісти.
- Компактність, зрозумілість, код на XAML відносно легко підтримувати.

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		7

При компіляції програми в Visual Studio код в XAML-файлах також компілюється в бінарне уявлення коду xaml, яке називається BAML (Binary Application Markup Language). І тоді код baml вбудовується у фінальну збірку програми-exe або dll-файл.

1.3. Формування технічного завдання

Метою даної випускної кваліфікаційної роботи є демонстрація здобутих навичок автора у сфері програмування. Вміння автора використовувати сучасні методи розробки додатків на платформі .NET та візуальне оформлення цих додатків.

За основну концепцію було взято спрощення ведення реєстру студентів, викладачів та структури у навчальному закладі та інформацію про них. Тому було прийнято рішення створити інформаційну довідкову систему «КНТЕУ». На данному етапі за мету було поставлено формування та програмування основи програмного забезпечення задля подальшого розвитку цього додатку. Під основою цього додатку, автор має на увазі базу даних, первинну логіку додатку та візуальну стилістику всього напрямку.

Задля того, щоб додаток можна було активно розвивати було обрано патерн програмування WMMW, котрий дозволить вносити зміни у додаток на будь якому етапі розробки чи то вже готової версії.

Основними користувачами додатку є методисти навчального закладу.

Висновки до розділу 1

Охоплюючи весь стек технологій, котрий наведено у розділі, та основні цілі цього проекту, можна приступати до розробки додатку. Перелічених мов

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		8

програмування та застосунків вистачить для того, щоб спроектувати та реалізувати програмне забезпечення. Обраний патерн програмування, дозволить вести розробку по сучасним стандартам та методам, котрі в подальшому допоможуть у розвитку цього проекту.

					<i>КНТЕУ 6.050103 07-23.БР</i>	Аркуш
<i>Изм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Подпись</i>	<i>Дата</i>		9

Розділ 2 Проектування системи

2.1. Проектування база даних для проекту

За основну платформу для бази даних було обрано MySQL сервер, котрий установлений на локальному комп'ютері. Доступ до бази даних маємо класично через консоль та через додаток спеціально розроблений для MySQL – MySQL Workbench. Додаток дозволяє нам дуже швидко оперувати базою даних та вносити необхідні зміни до її структури.

Основними принципами проектування баз даних це є дотримання мінімум третьої нормальної форми баз даних, зрозумілі назви таблиць та правильні типи даних, задля того, щоб адмініструвати цю базу даних міг будь-який спеціаліст, не читаючи тони документації.

В нашому випадку, база даних має всього 8 таблиць (рис.2.1), котрих вистачає для покриття структури університету, а саме:

- Таблиця кафедр;
- Таблиця факультетів;
- Таблиця спеціальностей;
- Таблиця викладачів;
- Таблиця предметів;
- Таблиця студентів;
- Суміжна таблиця між спеціальностями та предметами;
- Таблиця с оцінками за екзамени.

					<i>КНТЕУ 6.050103 07-23.БР</i>			
					<i>Розробка довідково-інформаційної системи КНТЕУ «Студент»</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		P2	10	25
Зав.каф.		Криворучко О.В.						
Керівник		Цюцюра М.І.						
Гарант		Цензура М.О.						
Розробив		Тіхонов А.О.			<i>Розділ 2</i>	<i>Факультет обліку, аудиту та інформаційних систем, 4 курсу, 7 групи</i>		

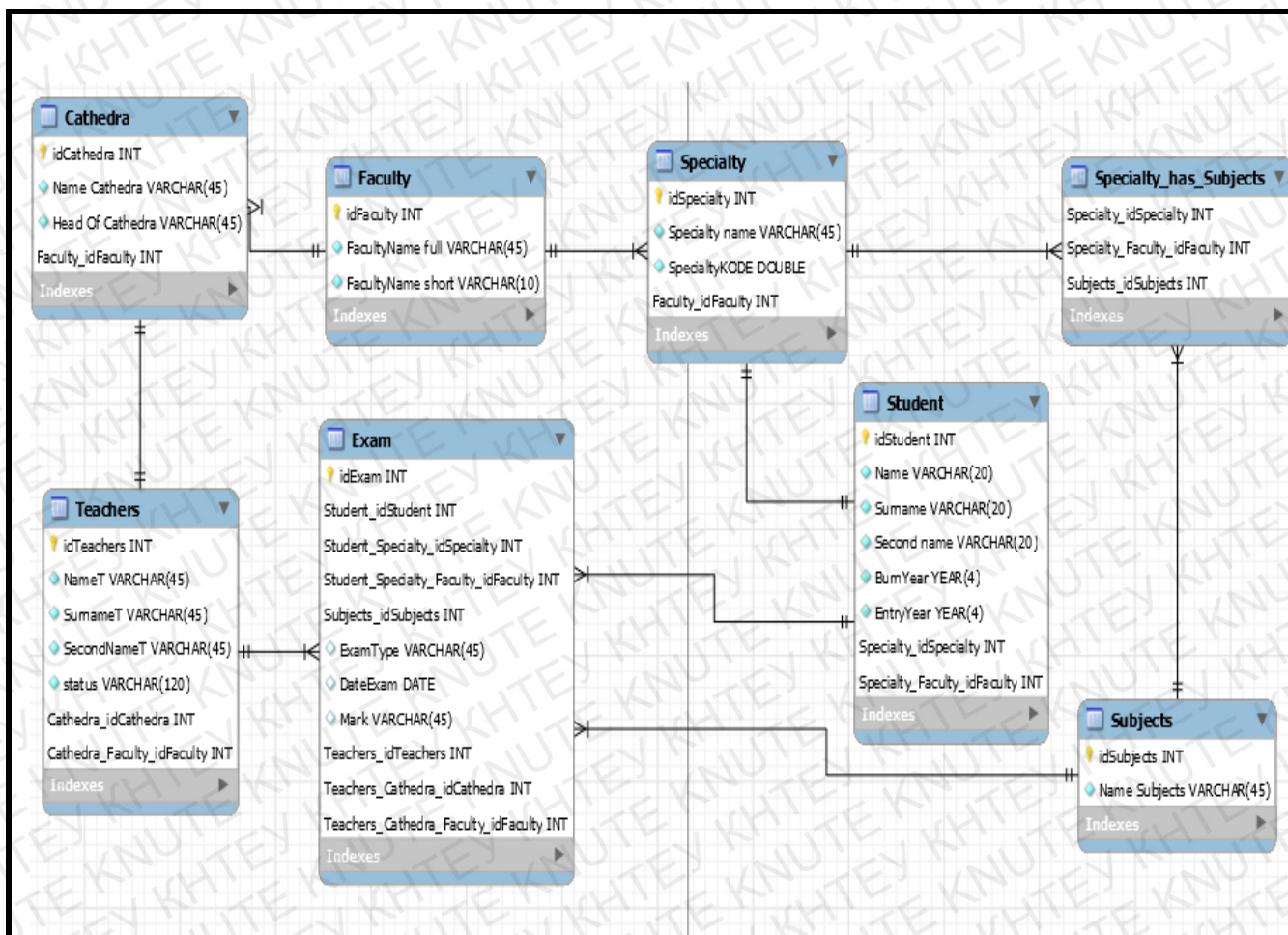


Рис.2.1 Модель бази даних

Всі таблиці взаємодіють між собою тим чи іншим способом, що дозволяють нам уникнути повторення однієї ж і той інформації у різних таблицях бази даних. Виходячи з цього рисунку, ми бачимо, що ми маємо всі необхідні елементи бази даних для створення додатку та наповнення його інформацією. Жодна з таблиць не містить у собі повторну інформацію з інших таблиць, на підставі цього ми можемо стверджувати, що наша база даних приведена до третьої нормальної форми баз даних. Саме цей чинник нам дозволить оперувати базою даних не відчуваючи особливих проблем з тим, що інформація буде повторно записуватись в обидві або більше таблиць.

Фізична модель бази даних побудована таким чином, щоб ми могли з будь якої точки нашої програми отримати необхідну нам інформацію. (рис.2.2.).

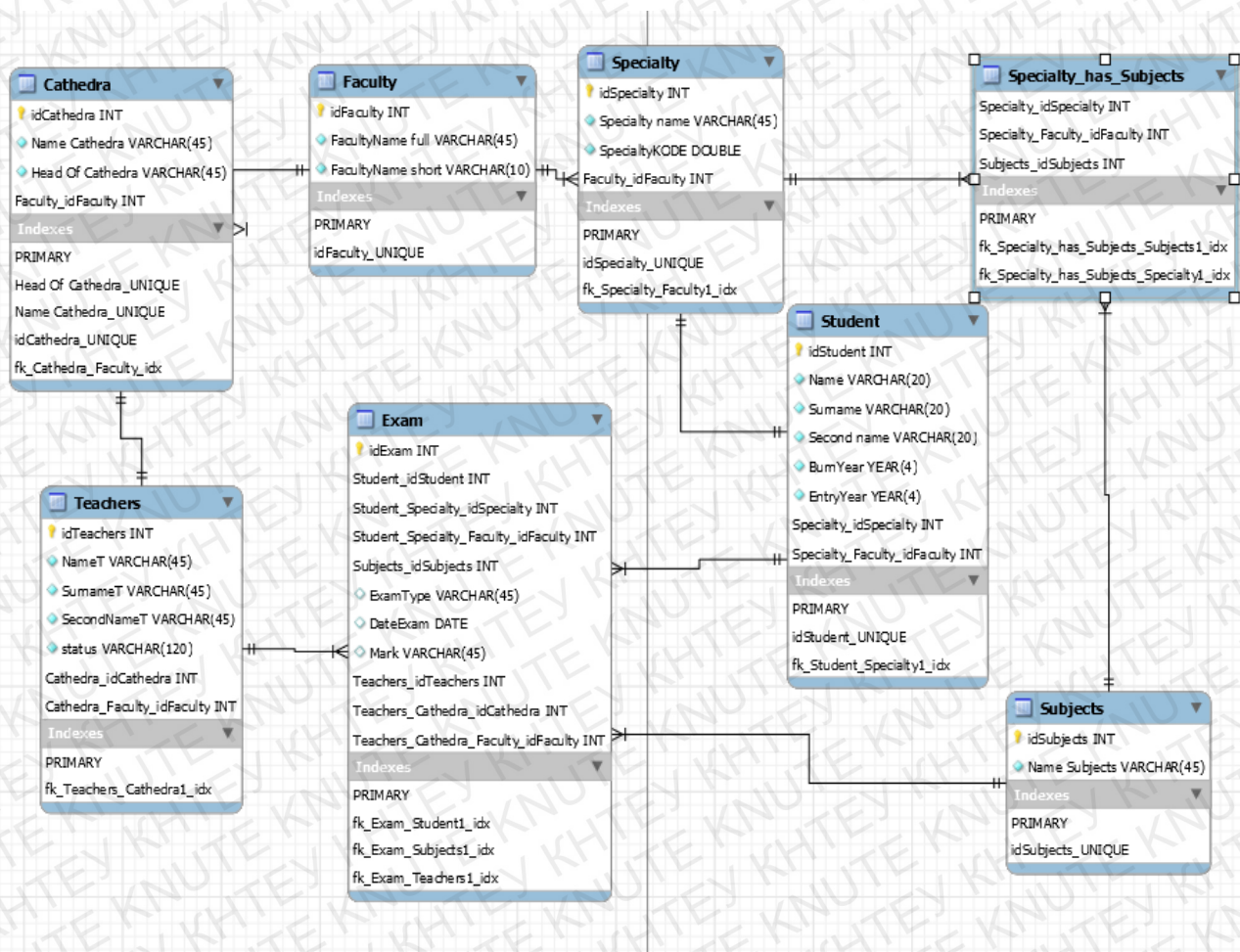


Рис.2.2 Фізична модель бази даних

Виходячи з цього рисунку, можемо побачити, що маємо безліч другорядних ключів, котрі допомагають підтримувати нормалізовану форму бази даних та відбирати інформацію з необхідних нам таблиць.

Також, в базі даних створено безліч готових «представлень» з метою оптимізації програмного коду та прискореного вибору необхідної інформації

2.1.2. Опис моделі взаємодії у базі даних

Для прикладу розберемо взаємодію двох таблиць, з котрих ми можемо створити суміжну таблицю для подання інформації у зручному нам вигляді. До деталей ми візьмемо представлення викладачів, де буде зазначено повне

ім'я викладача, його вчений ступінь, факультет до якого він відноситься та кафедра. Ця інформація знаходиться в трьох різних таблицях, а саме:

- Таблиця викладачів;
- Таблиця факультетів;
- Таблиця кафедри.

Для того щоб отримати дане подання (рис.2.3), нам необхідно застосувати оператор JOIN об'єднавши головні ключі трьох таблиць між собою. Також ми використовуємо оператор AS задля зручного подання заголовків.

Цей метод дозволяє нам формувати будь які таблиці подання з можливих, та значно прискорювати розробку додатку використовуючи подання.

	Прізвище	Ім'я	По-Батькові	Вчений рівень	Факультет	Кафедра
►	Криворучко	Олена	Володимирівна	Доктор технічних наук, професор	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Харченко	Олександр	Анатолійович	Кандидат технічних наук, доцент кафедри програмної інженерії та кібербез...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Пашорін	Валерій	Іванович	Кандидат технічних наук, професор кафедри програмної інженерії та кіберб...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Рзаєва	Світлана	Леонідівна	Кандидат технічних наук, професор кафедри програмної інженерії та кібербез...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Цензура	Микола	Олександрович	Кандидат технічних наук, доцент кафедри програмної інженерії та кібербез...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Рассмакін	Володимир	Якович	Кандидат технічних наук, доцент кафедри програмної інженерії та кібербез...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Палагута	Катерина	Олексіївна	Кандидат економічних наук, доцент кафедри програмної інженерії та кіберб...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Савченко	Тетяна	Віталіївна	Кандидат технічних наук, доцент кафедри програмної інженерії та кібербез...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Жирова	Тетяна	Олександрівна	Кандидат педагогічних наук, старший викладач кафедри програмної інжене...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Котенко	Наталія	Олексіївна	Кандидат педагогічних наук, старший викладач кафедри програмної інжене...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Десятко	Альона	Миколаївна	Старший викладач кафедри програмної інженерії та кібербезпеки	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Гнатченко	Дмитро	Дмитрович	Асистент кафедри програмної інженерії та кібербезпеки	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Шестак	Ярослав	Іванович	Директор ІОЦ ЦІТТ КНТЕУ, старший викладач кафедри програмної інженерії ...	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Козік	Олександр	Іванович	Інженер кафедри програмної інженерії та кібербезпеки	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Степашкіна	Катерина	Володимирівна	Спеціаліст кафедри програмної інженерії та кібербезпеки	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Шевченко	Ліна	Анатоліївна	Лаборант	ФОАІС	КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
	Фоміна	Олена	Володимирівна	Завідувач кафедри, Доктор економічних наук, доцент	ФОАІС	КАФЕДРА ОБЛІКУ ТА ОПОДАТКУВАННЯ
	Сопко	Валерія	Василівна	Професор кафедри, Доктор економічних наук, професор	ФОАІС	КАФЕДРА ОБЛІКУ ТА ОПОДАТКУВАННЯ

Рис.2.3 Подання викладацького складу факультету

					КНТЕУ 6.050103 07-23.БР	Аркуш
Ізм.	Аркуш	№ докум	Підпись	Дата		13

З точки зору мови SQL цей запит буде виглядати так.(рис.2.4)

```

5  VIEW `systemstudent`.`teacher` AS
6  SELECT
7      `systemstudent`.`teachers`.`SurnameT` AS `Прізвище`,
8      `systemstudent`.`teachers`.`NameT` AS `Ім'я`,
9      `systemstudent`.`teachers`.`SecondNameT` AS `По-Батькові`,
10     `systemstudent`.`teachers`.`status` AS `Вченний рівень`,
11     `systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,
12     `systemstudent`.`cathedra`.`NameCathedra` AS `Кафедра`
13  FROM
14     ((`systemstudent`.`teachers`
15  JOIN `systemstudent`.`cathedra` ON ((`systemstudent`.`cathedra`.`idCathedra`
16  = `systemstudent`.`teachers`.`Cathedra_idCathedra`)))
17  JOIN `systemstudent`.`faculty` ON ((`systemstudent`.`faculty`.`idFaculty`
18  = `systemstudent`.`cathedra`.`Faculty_idFaculty`)))

```

Рис.2.4 Структурований запит до бази даних

З даного запиту ми бачимо, що ми просимо базу даних дати нам представлення з 3 таблиць, а для достовірної передачі даних, тобто щоб вони були зв'язані між собою, ми прив'язуємо значення з таблиці кафедри до таблиці вчителів за відповідним ключем, так само ми робимо і з таблицею факультетів та кафедри. Це дозволяє нам отримати необхідне подання з трьох таблиць. При чому ми вказуємо ті дані, котрі не відображаються у поданні, але слугують для фізичної та логічної зв'язки даних та дотримання цілісності.

2.2. Систематизація структури проекту згідно патерну MVVM

Патерн розробки MVVM чітко змушує нас розділяти структуру проекту таким чином, щоб вони мінімально впливали одне на одного з точки зору розробки коду, відповідно до цього є і вимоги щодо

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		14

структурування папок при розробці, щоб під час розробки легко можна було звертатися до необхідних форм чи класів.(рис.2.5)

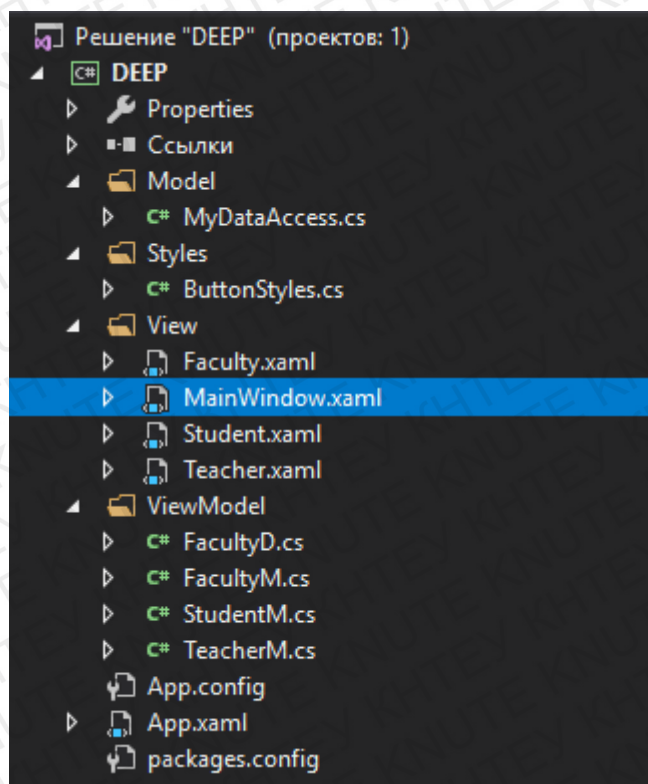


Рис.2.5 Структура проекту.

Ми бачимо, що всі елементи знаходяться на своїх місцях та викликаються та стають активними, тільки тоді коли це потрібно з точки зору програмної реалізації. Це дозволяє економити ресурси комп'ютера на більш важливі завдання.

2.3. Розробка користувацького інтерфейсу

Дизайн інтерфейсу користувача додатку відіграє дуже важливу роль у доведенні до нього великого обсягу інформаційної системи. Добре продуманий користувацький інтерфейс не тільки збільшує зручність використання додатку, алей й забезпечує плавне завершення будь-якої задачі,

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		15

роблячи таким чином враження від використання приємним та відповідним до вимог користувача.

З точки зору дизайну додатку, він має бути сприятливим для користувача, щоб йому одразу було зрозуміло, де знаходяться необхідні функції. У додатку повинні бути присутніми підказки та запобіжники для того, щоб інформація, з якої працює користувач була коректною та правильно відображалась або оброблювалась.

Патерн розробки MVVM та WPF(Windows Presentation Form) дозволяють нам реалізувати будь який дизайн у нашому додатку, ми не обмежуймось стандартними властивостями WF(Windows Forms), а маємо в своєму розпорядженні мову розмітки XAML, про яку автор згадував раніше. За допомогою цих технологій обрамлювати будь які елементи дизайну, від звичайних кнопок, роблячи їх будь якої форми та кольору або ж анімації, до головних вікон, елементів таблиці, та будь яких елементів дизайну, котрі доступні нам у WPF – формах.

Мова розмітки XAML дає можливість нам розробляти, власні стилі та прив'язувати їх за певним індикатором. Так наприклад ми можемо створити окремий стиль для елемента кнопки та вже в коді форми зазначити, що певна кнопка наслідує цей стиль. Наведемо приклад створення такого стилю (рис.2.7), де відобразимо такі кнопку, для перегляду контенту(рис.2.6).



Рис.2.6 Приклад створення стилю для елемента Button

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		16

```

<Style TargetType="Button" x:Key="btnBlue">
  <Setter Property="Background" Value="#fff"/>
  <Setter Property="Foreground" Value="#000"/>
  <Setter Property="FontSize" Value="15"/>
  <Setter Property="Margin" Value="5"/>
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="Button">
        <Border Background="{TemplateBinding Background}"
          CornerRadius="5"
          BorderThickness="1"
          Padding="5"
          BorderBrush="#000">
          <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center">
            </ContentPresenter>
          </Border>
        </ControlTemplate>
      </Setter.Value>
    </Setter>
  <Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Background" Value="Blue"/>
      <Setter Property="Foreground" Value="#fff"/>
    </Trigger>
  </Style.Triggers>
</Style>

<Style TargetType="Button" x:Key="btnGreen">
  <Setter Property="Background" Value="#fff"/>
  <Setter Property="Foreground" Value="#000"/>
  <Setter Property="FontSize" Value="15"/>
  <Setter Property="Padding" Value="5"/>
  <Setter Property="Margin" Value="5"/>
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="Button">
        <Border Background="{TemplateBinding Background}"
          CornerRadius="5"
          BorderThickness="1"
          Padding="5"
          BorderBrush="#000">
          <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center">
            </ContentPresenter>
          </Border>
        </ControlTemplate>
      </Setter.Value>
    </Setter>
  <Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Background" Value="Green"/>
    </Trigger>
  </Style.Triggers>
</Style>

```

Рис.2.7. Фрагмент коду XAML по створенню стилю для елементу Button

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		17

З фрагменту коду, ми бачимо, що розробник програмує кожну деталь елементу від зовнішнього вигляду, до ефекту натискання на кнопку (Рис.7). Програміст може позначити стандартний розмір кнопки, кути загострення, фон, горизонтальне та вертикальне вирівнювання та безліч інших властивостей даного елементу форми.

З точки зору виклику цієї кнопки на формі та передання необхідної інформації можна побачити тут(Рис.2.8)

```
<StackPanel Orientation="Vertical">
<TextBlock Text="WPF Button Design" Foreground="Blue" FontSize="20" FontWeight="Bold" Margin="10,0,0,0"/>
<StackPanel Orientation="Horizontal" Margin="10">

<Button Content="Blue" Width="100" Style="{StaticResource btnBlue}" HorizontalAlignment="Left"/>
<Button Content="Green" Width="100" Style="{StaticResource btnGreen}" HorizontalAlignment="Left"/>
<Button Content="Gray" Width="100" Style="{StaticResource btnGray}" HorizontalAlignment="Left"/>
<Button Content="Light Coral" Width="100" Style="{StaticResource btnLightRed}" HorizontalAlignment="Left"/>
</StackPanel>
</StackPanel>
</Window>
```

Рис.2.8 Виклик елементу Button на форму.

2.4.Висновки до розділу 2

Використовуючи всі перелічені технології, а саме:

- Мова розробки C#;
- Мова розмітки XAML;
- Патерн розробки десктоп додатку MVVM.

Автор може реалізувати програмний додаток, та зробити його інтерфейс дружелюбним до користувача. Можливостей в даному стеку більш ніж достатньо для цього.

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Підпись	Дата		18

Розділ 3 Програмна реалізація додатку

3.1.Коннектори до бази даних MySQL

Для взаємодії програмного коду додатку, нам необхідно підключатись до бази даних MySQL, котра знаходиться на локальному комп'ютері. Відповідно до цього в програмному коді С# ми маємо сталий об'єкт типу String, який містить в собі дані для підключення до бази даних MySQL(рис.3.1)

```
string connstr = "server = localhost; " +  
"user = root;" +  
" database = systemstudent; " +  
"password = !QAZpassSQL1";
```

Рис.3.1 рядок підключення до бази даних MySQL.

Цей рядок при кожному з'єднанні буде однаковим, тому ми виносимо його до файлу «App.config», тим самим робимо його доступним з будь-якої точки нашого коду(Рис.3.2)

```
</configSections>  
<connectionStrings>  
  <add name="DbConnectionString" connectionString="server = localhost; user = root; database = systemstudent; password = !QAZpassSQL1  
    Initial Catalog=Student; Integrated Security=True;" providerName="System.Data.SqlClient"/>  
</connectionStrings>
```

Рис.3.2 рядок підключення до бази даних в App.config

Для того, щоб компілятор С# розумів команди, котрі ми передаємо в сторону серверу баз даних, нам необхідно підключити бібліотеку «MySQL.Data.MySqlClient», котра дозволить нам звертатись до простору імен бібліотеки та використовувати функції і методи звідти.

					КНТЕУ 6.050103 07-23.БР			
					Розробка довідково-інформаційної системи КНТЕУ «Студент»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		РЗ	19	25
Зав.каф.		Криворучко О.В.			Розділ 3	Факультет обліку, аудиту та інформаційних систем, 4 курсу, 7 групи		
Керівник		Цюцюра М.І.						
Гарант		Цензура М.О.						
Розробив		Тіхонов А.О.						

Для того щоб, отримати певні дані з бази даних до наших елементів, як наприклад «DataGrid», нам необхідно передавати команду SQL до бази даних, саме для цього ми і використовуємо бібліотеку імен «MySQL» та методи, які вона додає, а саме:

- MySqlConnection;
- MySqlCommand;
- MySqlDataReader;
- MySqlDataAdapter.

Звичайно, методів на порядок більше, але це основні методи, котрі дозволяють нам отримувати необхідну інформацію з бази даних. Для прикладу розглянемо коннектор до бази даних, котрий передає інформацію до нашого елементу «DataGrid» (Рис.3.3)

```
string connstr = "server = localhost;user = root;database = systemstudent;password = !QAZpassSQL1";
string sqlFS = " SELECT " +
""systemstudent`.`student`.`Surname` AS `Прізвище`,`" +
""systemstudent`.`student`.`Name` AS `Ім'я`,`" +
""systemstudent`.`student`.`SecondName` AS `По - батькові`,`" +
""systemstudent`.`student`.`BurnYear` AS `Рік народження`,`" +
""systemstudent`.`student`.`EntryYear` AS `Рік вступу`,`" +
""systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`,`" +
""systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,`" +
""systemstudent`.`student`.`group` AS `Група`" +
" FROM(`systemstudent`.`student`"+
"JOIN `systemstudent`.`specialty` ON((`systemstudent`.`student`.`Specialty_idSpecialty` = `systemstudent`.`specialty`.`idSpecialty`)))"+
"JOIN `systemstudent`.`faculty` ON((`systemstudent`.`specialty`.`Faculty_idFaculty` = `systemstudent`.`faculty`.`idFaculty`)))"+
"WHERE(`systemstudent`.`faculty`.`idFaculty` = '2')";

MySqlConnection conn = new MySqlConnection(connstr);
conn.Open();
MySqlCommand comF = new MySqlCommand(sqlFS,conn);
MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);
DataTable dt = new DataTable();
dataAdapter.Fill(dt);
FacultyData.ItemsSource = dt.DefaultView;
conn.Close();
```

Рис.3.3. Коннектор до бази даних.

На цьому прикладі маємо складний запит, котрий формує таблицю з трьох різних таблиць реляційної бази даних.

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		20

3.2. Реалізація візуальних стилів з допомогою XAML

XAML це потужний інструмент для створення унікального дизайну, але автор не відносить себе до дизайнерської групи, тому дизайн матиме строгий вигляд, без шалених візуальних ефектів. Для прикладу розглянемо реалізацію елементу «StackPanel», на якому розміщені кнопки для виклику тієї чи іншої інформації до таблиці.(Рис.3.4)

```
<StackPanel>
  <Button Grid.Column="0" Margin="3"
    Width="140" Height="80"
    Content="ΦΟΑΙC"
    Background="Aqua"
    Click="FOAISfac"/>
  <Button Grid.Column="0" Margin="3"
    Width="140" Height="80"
    Content="ΦΕΜΠ"
    Background="PaleVioletRed"
    Click="FEMPfac"/>
  <Button Grid.Column="0" Margin="3"
    Width="140" Height="80"
    Content="ΦΜΤΠ"
    Background="Violet"
    Click="FMTPfac"/>
  <Button Grid.Column="0" Margin="3"
    Width="140" Height="80"
    Content="ΦΒΒC"
    Background="Yellow"
    Click="FFBSfac"/>
  <Button Grid.Column="0" Margin="3"
    Width="140" Height="80"
    Content="ΦΡΓΤΒ"
    Background="Pink"
    Click="FRGTBfac"/>
  <Button Grid.Column="0" Margin="3"
    Width="140" Height="80"
    Content="ΦΤΜ"
    Background="Green"
    Click="FTMfac"/>
</StackPanel>
```

Рис.3.4. Розмітка XAML відносно кнопок

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		21

3.3.Реалізація логічної моделі

Як приклад реалізації логічної моделі, ми розглянемо процес заповнення елемента «ComboBox» інформацією з баз даних, для подальшого сортування інформації у таблиці.

Мовою розмітки, ми створюємо елемент та прив'язуємо до нашої сітки, задаємо розміри, ім'я по якому до нього можна буде звернутись з моделі та ініціюємо обробник події (рис.3.5).

```
<ComboBox Grid.Row="0"
            Width="140" Height="30"
            Margin="10"
            x:Name="enter"
            SelectionChanged="Enter_SelectionChanged"
            />
```

Рис.3.5 Елемент ComboBox

А вже в моделі взаємодії створюємо метод, котрий буде підключатися до бази даних, створюємо скрипт, для того, щоб отримати необхідні дані з бази даних та наповнюємо масив об'єктів списку нашого елемента «ComboBox», перед цим реалізуємо конструкцію «try - catch», для того щоб наша програма не вилітала під час ініціалізації форми, на якій знаходиться наш елемент.(Рис.3.6)

```
void FillCombo()
{
    string connstr = "server = localhost; user = root; database = systemstudent; password = !QA2passSQL1";
    string combobox = "SELECT DISTINCT student.EntryYear FROM systemstudent.student";
    MySqlConnection conn = new MySqlConnection(connstr);
    MySqlCommand com = new MySqlCommand(combobox, conn);
    MySqlDataReader myReader;
    try
    {
        conn.Open();
        myReader = com.ExecuteReader();

        while (myReader.Read())
        {
            string sname = myReader.GetString("EntryYear");
            enter.Items.Add(sname);
        }
    }
    catch (Exception )
    {
    }
}
```

Рис.3.6 Наповнення масиву об'єктів.

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		22

3.4. Висновки до розділу 3

Реалізація такої програми, з усіма стилями потребує безліч строчок коду та використаних елементів. В даному розділі, автором, було наведено лише деякі приклади реалізації патерну MVVM та взаємодію між додатком та базою даних. Подібним чином і реалізується весь програмний продукт.

Автор, вважає, що наведених прикладів вистачає для того, щоб показати рівень освіченості в даній темі та в даному випускному кваліфікаційному проекті.

					КНТЕУ 6.050103 07-23.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		23

Висновки та пропозиції

Автор, вважає, що тема випускної кваліфікованої роботи, ним була успішно розглянута на прикладах та реалізовано у програмному додатку для десктопу. Додаток має великий потенціал для подальшої розробки за рахунок того, що там чітко відмежовуються сфери діяльності програміста та дизайнера інтерфейсу. Також додаток є легко масштабованим, в будь який час можна додати нові функції, методи, котрі не будуть заважати основній концепції.

Додаток дає можливість працювати з базою даних університету та швидко її оновлювати. Звичайно, для вдосконалення бази даних, треба ввести політику безпеки в середі MySQL серверу та на рівні програмного забезпечення.

Також автор бачить можливим додати модулі розрахунку бюджетників по групі, введення індивідуальної картки студента з оцінками за весь навчальний курс але всі ці доповнення потребують великих затрат у вигляді часу та з фінансової сторони.

					<i>КНТЕУ 6.050103 07-23.БР</i>			
					<i>Розробка довідково-інформаційної системи КНТЕУ «Студент»</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		ВП	24	25
Зав.каф.		Криворучко О.В.			<i>Висновки та пропозиції</i>	<i>Факультет обліку, аудиту та інформаційних систем, 4 курсу, 7 групи</i>		
Керівник		Цюцюра М.І.						
Гарант		Цензура М.О.						
Розробив		Тіхонов А.О.						

Список використаних джерел

1. А. Ю. Берко, О. М. Верес; Організація баз даних: практичний курс : Навч. посіб. для студ. Нац. ун-т "Львів. політехніка". - Л., 2003. - 149 с.
2. Абрамян М.; Visual C# на прикладах . - СПб.: БХВ-петербург, 2008. – 496с
3. Д. Албахари.; C# 6.0. Довідник. Повний опис мови, 2017.-1040с.
4. А. Стилмен.; Вивчаємо C#, 2012.-704с.

Інтернет ресурси:

1. Документація редактору коду Sublime Text [Електронний ресурс]. – [Режим доступу]: <https://www.sublimetext.com/support>
2. Офіційна документація Microsoft [Електронний ресурс]. – [Режим доступу]: <https://msdn.microsoft.com/ru-ru/>

					<i>КНТЕУ 6.050103 07-23.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка довідково-інформаційної системи КНТЕУ «Студент»	Стадія	Аркуш	Аркушів
Зав.каф.		Криворучко О.В.				СД	25	25
Керівник		Цюцюра М.І.						
Гарант		Цензура М.О.						
Розробив		Тіхонов А.О.			Список використаних джерел	Факультет обліку, аудиту та інформаційних систем, 4 курсу, 7 групи		

Додаток А

Реалізація програмного коду

MainWindow.xaml

```
<Window x:Class="DEEP.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:DEEP"
  xmlns:viewmodels="clr-namespace:DEEP.ViewModel"
  xmlns:view="clr-namespace:DEEP.View"
  mc:Ignorable="d"
  Title="MainWindow" Height="720" Width="1280">
  <Window.Resources>
    <DataTemplate x:Name="WelcomeView" DataType="{x:Type
view:WelcomeView}">
      <view:WelcomeView DataContext="{Binding}" />
    </DataTemplate>
    <DataTemplate x:Name="FacultyViewTemplate" DataType="{x:Type
viewmodels:FacultyM}">
      <view:Faculty DataContext="{Binding}" />
    </DataTemplate>
    <DataTemplate x:Name="TeacherViewTemplate" DataType="{x:Type
viewmodels:TeacherM}">
      <view:Teacher DataContext="{Binding}" />
    </DataTemplate>
    <DataTemplate x:Name="StudentViewTemplate" DataType="{x:Type
viewmodels:StudentM}">
      <view:Student DataContext="{Binding}" />
    </DataTemplate>
  </Window.Resources>
  <Grid Background="OldLace" >
    <Grid.RowDefinitions>
      <RowDefinition Height="30"/>
      <RowDefinition Height="*" />
    </Grid.RowDefinitions>

    <StackPanel Orientation="Horizontal" >
      <Button Content="Факультети" Margin="5"
Width="auto" Click="FacultyClick">
```

```

        />
        <Button Content="Викладачі" Margin="5"
            Width="0" Click="TeacherClick"
            HorizontalAlignment="Center"/>
        <Button Content="Студент" Margin="5"
            Width="auto" Click="StudentClick"
            HorizontalAlignment="Right"/>
    </StackPanel>
    <ContentControl Grid.Row="1" Content="{Binding}"/>

</Grid>
</Window>

```

MainWindow.xaml.cs

```

using DEEP.ViewModel;
using DEEP.View;
using System.Windows;

namespace DEEP
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            DataContext = new WelcomeView();
        }
        private void FacultyClick(object sender, RoutedEventArgs e)
        {
            DataContext = new FacultyM();
        }
        private void TeacherClick(object sender, RoutedEventArgs e)
        {
            DataContext = new TeacherM();
        }
        private void StudentClick(object sender, RoutedEventArgs e)
        {
            DataContext = new StudentM();
        }
    }
}

```

Faculty.xaml.cs

```
using System.Windows;
using System.Windows.Controls;
using System.Data.SqlClient;
using System.Data;
using MySql.Data.MySqlClient;
using DEEP.Model;
using System;

namespace DEEP.View
{
    /// <summary>
    /// Логика взаимодействия для UserControl1.xaml
    /// </summary>
    public partial class Faculty : UserControl
    {
        public Faculty()
        {
            InitializeComponent();
            FillCombo();
            FillComboGr();
        }

        void FillCombo()
        {

```

```
string connstr = "server = localhost; user = root; database = systemstudent;  
password = !QAZpassSQL1";
```

```
string combobox = "SELECT DISTINCT student.EntryYear FROM  
systemstudent.student";
```

```
MySqlConnection conn = new MySqlConnection(connstr);
```

```
MySqlCommand com = new MySqlCommand(combobox, conn);
```

```
MySqlDataReader myReader;
```

```
try
```

```
{
```

```
    conn.Open();
```

```
    myReader = com.ExecuteReader();
```

```
    while (myReader.Read())
```

```
    {
```

```
        string sname = myReader.GetString("EntryYear");
```

```
        enter.Items.Add(sname);
```

```
    }
```

```
}
```

```
catch (Exception )
```

```
{
```

```
}
```

```
}
```

```
void FillComboGr()
```

```
{
```

```
string connstr = "server = localhost; user = root; database = systemstudent;  
password = !QAZpassSQL1";
```

```
string combobox = "SELECT DISTINCT `student`.`group` FROM  
`systemstudent`.`student`";
```

```
MySQLConnection conn = new MySQLConnection(connstr);
```

```
MySQLCommand com = new MySQLCommand(combobox, conn);
```

```
MySQLDataReader myReader;
```

```
try
```

```
{
```

```
    conn.Open();
```

```
    myReader = com.ExecuteReader();
```

```
    while (myReader.Read())
```

```
    {
```

```
        string sname = myReader.GetString("group");
```

```
        gropcb.Items.Add(sname);
```

```
    }
```

```
}
```

```
catch (Exception)
```

```
{
```

```
}
```

```
}
```

```
public void FOAISfac(object sender, RoutedEventArgs e)
```

```
{
```

```
string connstr = "server = localhost;user = root;database =  
systemstudent;password = !QAZpassSQL1";  
  
string sqlFS = " SELECT" +  
""systemstudent`.`student`.`Surname` AS `Прізвище`,`" +  
""systemstudent`.`student`.`Name` AS `Ім'я`,`" +  
""systemstudent`.`student`.`SecondName` AS `По - батькові`,`" +  
""systemstudent`.`student`.`BurnYear` AS `Рік народження`,`" +  
""systemstudent`.`student`.`EntryYear` AS `Рік вступу`,`" +  
""systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`,`" +  
""systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,`" +  
""systemstudent`.`student`.`group` AS `Група`" +  
" FROM((`systemstudent`.`student`"+  
"JOIN `systemstudent`.`specialty`  
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =  
`systemstudent`.`specialty`.`idSpecialty`)))"+  
"JOIN `systemstudent`.`faculty`  
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =  
`systemstudent`.`faculty`.`idFaculty`)))"+  
"WHERE(`systemstudent`.`faculty`.`idFaculty` = '2')";  
  
MySqlConnection conn = new MySqlConnection(connstr);  
conn.Open();  
  
MySqlCommand comF = new MySqlCommand(sqlFS,conn);  
  
MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);  
  
DataTable dt = new DataTable();  
  
dataAdapter.Fill(dt);
```

```

FacultyData.ItemsSource = dt.DefaultView;

conn.Close();

}

private void FEMPfac(object sender, RoutedEventArgs e)
{
    string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

    string sqlFS = " SELECT" +
        ""systemstudent`.`student`.`Surname` AS `Прізвище`,`" +
        ""systemstudent`.`student`.`Name` AS `Ім'я`,`" +
        ""systemstudent`.`student`.`SecondName` AS `По - батькові`,`" +
        ""systemstudent`.`student`.`BurnYear` AS `Рік народження`,`" +
        ""systemstudent`.`student`.`EntryYear` AS `Рік вступу`,`" +
        ""systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`,`" +
        ""systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,`" +
        ""systemstudent`.`student`.`group` AS `Група"" +
        " FROM((`systemstudent`.`student`" +
        "JOIN `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =
`systemstudent`.`specialty`.`idSpecialty`)))" +
        "JOIN `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =
`systemstudent`.`faculty`.`idFaculty`)))" +
        "WHERE(`systemstudent`.`faculty`.`idFaculty` = '1')";

```

```

MySQLConnection conn = new MySqlConnection(connstr);
conn.Open();
MySQLCommand comF = new MySqlCommand(sqlFS, conn);
MySQLDataAdapter dataAdapter = new MySQLDataAdapter(comF);
DataTable dt = new DataTable();
dataAdapter.Fill(dt);
FacultyData.ItemsSource = dt.DefaultView;
conn.Close();
}

private void FMTPfac(object sender, RoutedEventArgs e)
{
    string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

    string sqlFS = " SELECT" +
        "`systemstudent`.`student`.`Surname` AS `Прізвище`,`" +
        "`systemstudent`.`student`.`Name` AS `Ім'я`,`" +
        "`systemstudent`.`student`.`SecondName` AS `По - батькові`,`" +
        "`systemstudent`.`student`.`BurnYear` AS `Рік народження`,`" +
        "`systemstudent`.`student`.`EntryYear` AS `Рік вступу`,`" +
        "`systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`,`" +
        "`systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,`" +
        "`systemstudent`.`student`.`group` AS `Група`" +
        " FROM((`systemstudent`.`student`" +

```

```

        "JOIN `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =
`systemstudent`.`specialty`.`idSpecialty`)))" +

        "JOIN `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =
`systemstudent`.`faculty`.`idFaculty`)))" +

        "WHERE(`systemstudent`.`faculty`.`idFaculty` = '6')";

        MySqlConnection conn = new MySqlConnection(connstr);
        conn.Open();

        MySqlCommand comF = new MySqlCommand(sqlFS, conn);
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);
        DataTable dt = new DataTable();
        dataAdapter.Fill(dt);

        FacultyData.ItemsSource = dt.DefaultView;
        conn.Close();
    }

    private void FRGTBfac(object sender, RoutedEventArgs e)
    {
        string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

        string sqlFS = " SELECT" +

        "`systemstudent`.`student`.`Surname` AS `Прізвище`,`" +

        "`systemstudent`.`student`.`Name` AS `Ім'я`,`" +

        "`systemstudent`.`student`.`SecondName` AS `По - батькові`,`" +

        "`systemstudent`.`student`.`BurnYear` AS `Рік народження`,`" +

```

```

        "systemstudent`.`student`.`EntryYear` AS `Рік вступу`," +
        "systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`," +
        "systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`," +
        "systemstudent`.`student`.`group` AS `Група`" +
        " FROM((`systemstudent`.`student`" +
        "JOIN `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =
`systemstudent`.`specialty`.`idSpecialty`)))" +
        "JOIN `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =
`systemstudent`.`faculty`.`idFaculty`)))" +
        "WHERE(`systemstudent`.`faculty`.`idFaculty` = '3')";

        MySqlConnection conn = new MySqlConnection(connstr);
        conn.Open();

        MySqlCommand comF = new MySqlCommand(sqlFS, conn);
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);
        DataTable dt = new DataTable();
        dataAdapter.Fill(dt);
        FacultyData.ItemsSource = dt.DefaultView;
        conn.Close();
    }

    private void FFBSfac(object sender, RoutedEventArgs e)
    {
        string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

```

```

        string sqlFS = " SELECT" +
        ""systemstudent`.`student`.`Surname` AS `Прізвище`," +
        ""systemstudent`.`student`.`Name` AS `Ім'я`," +
        ""systemstudent`.`student`.`SecondName` AS `По - батькові`," +
        ""systemstudent`.`student`.`BurnYear` AS `Рік народження`," +
        ""systemstudent`.`student`.`EntryYear` AS `Рік вступу`," +
        ""systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`," +
        ""systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`," +
        ""systemstudent`.`student`.`group` AS `Група`" +
        " FROM((`systemstudent`.`student`" +
        "JOIN `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =
`systemstudent`.`specialty`.`idSpecialty`)))" +
        "JOIN `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =
`systemstudent`.`faculty`.`idFaculty`)))" +
        "WHERE(`systemstudent`.`faculty`.`idFaculty` = '5')";

```

```

        MySqlConnection conn = new MySqlConnection(connstr);
        conn.Open();
        MySqlCommand comF = new MySqlCommand(sqlFS, conn);
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);
        DataTable dt = new DataTable();
        dataAdapter.Fill(dt);
        FacultyData.ItemsSource = dt.DefaultView;

```

```

        conn.Close();
    }

    private void FTMfac(object sender, RoutedEventArgs e)
    {
        string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

        string sqlFS = " SELECT" +
        ""systemstudent`.`student`.`Surname` AS `Прізвище`,`" +
        ""systemstudent`.`student`.`Name` AS `Ім'я`,`" +
        ""systemstudent`.`student`.`SecondName` AS `По - батькові`,`" +
        ""systemstudent`.`student`.`BurnYear` AS `Рік народження`,`" +
        ""systemstudent`.`student`.`EntryYear` AS `Рік вступу`,`" +
        ""systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`,`" +
        ""systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,`" +
        ""systemstudent`.`student`.`group` AS `Група`" +
        " FROM((`systemstudent`.`student`" +
        "JOIN `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =
`systemstudent`.`specialty`.`idSpecialty`)))" +
        "JOIN `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =
`systemstudent`.`faculty`.`idFaculty`)))" +
        "WHERE( `systemstudent`.`faculty`.`idFaculty` = '4')";

        MySqlConnection conn = new MySqlConnection(connstr)
        conn.Open();
    }

```

```

MySQLCommand comF = new MySQLCommand(sqlFS, conn);
MySQLDataAdapter dataAdapter = new MySQLDataAdapter(comF);
DataTable dt = new DataTable();
dataAdapter.Fill(dt);
FacultyData.ItemsSource = dt.DefaultView;
conn.Close();
}

private void data_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    //surname.Text = FacultyData.SelectedCells[0].Item.ToString();
    //FacultyData.SelectedCells[0].Item.ToString();
    //surname.Text = FacultyData.SelectedItems[0].ToString();
    DataRowView odrv = enter.SelectedItem as DataRowView;
    string svalue = "";
    if (odrv != null)
    {
        svalue = odrv.Row["Surname"] as string;
    }
}

private void FacultyData_CurrentCellChanged(object sender, EventArgs e)
{
}

private void Enter_SelectionChanged(object sender, SelectionChangedEventArgs
e)

```

```

{
    string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

    string sqlFS = " SELECT" +

        "`systemstudent`.`student`.`Surname` AS `Прізвище`," +
        "`systemstudent`.`student`.`Name` AS `Ім'я`," +
        "`systemstudent`.`student`.`SecondName` AS `По - батькові`," +
        "`systemstudent`.`student`.`BurnYear` AS `Рік народження`," +
        "`systemstudent`.`student`.`EntryYear` AS `Рік вступу`," +
        "`systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`," +
        "`systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`," +
        "`systemstudent`.`student`.`group` AS `Група`" +

        " FROM((`systemstudent`.`student`" +

        "JOIN `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty` =
`systemstudent`.`specialty`.`idSpecialty`)))" +

        "JOIN `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty` =
`systemstudent`.`faculty`.`idFaculty`)))" +

        "WHERE(`systemstudent`.`faculty`.`idFaculty` = '2'AND
(`systemstudent`.`student`.`EntryYear` = '" + enter.SelectedItem.ToString() + "'))";

    MySqlConnection conn = new MySqlConnection(connstr);
    conn.Open();

    MySqlCommand comF = new MySqlCommand(sqlFS, conn);

    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);

```

```

        DataTable dt = new DataTable();

        dataAdapter.Fill(dt);

        FacultyData.ItemsSource = dt.DefaultView;

        conn.Close();
    }

    private void Gropcb_SelectionChanged(object sender,
    SelectionChangedEventArgs e)
    {
    }
}

```

Faculty.xaml

```

<UserControl x:Class="DEEP.View.Faculty"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:DEEP.View"
    xmlns:model="clr-namespace:DEEP.Model"
    mc:Ignorable="d"
    d:DesignHeight="690" d:DesignWidth="1280">
    <Grid >
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="150"/>
            <ColumnDefinition Width="*/"/>
        </Grid.ColumnDefinitions>

        <StackPanel>
            <Button Grid.Column="0" Margin="3"
                Width="140" Height="80"
                Content="ΦΟΑΙC"
                Background="Aqua"
                Click="FOAISfac"/>
            <Button Grid.Column="0" Margin="3"

```

```

Width="140" Height="80"
Content="ΦΕΜΠ"
Background="PaleVioletRed"
Click="FEMPfac"/>
<Button Grid.Column="0" Margin="3"
Width="140" Height="80"
Content="ΦΜΤΠ"
Background="Violet"
Click="FMTPfac"/>
<Button Grid.Column="0" Margin="3"
Width="140" Height="80"
Content="ΦΒΒC"
Background="Yellow"
Click="FFBSfac"/>
<Button Grid.Column="0" Margin="3"
Width="140" Height="80"
Content="ΦΡΓΤΒ"
Background="Pink"
Click="FRGTBfac"/>
<Button Grid.Column="0" Margin="3"
Width="140" Height="80"
Content="ΦΤΜ"
Background="Green"
Click="FTMfac"/>
</StackPanel>

<Grid Grid.Column="1" Background="Bisque">
<Grid.ColumnDefinitions>
<ColumnDefinition Width="*"/>
<ColumnDefinition Width="*"/>
<ColumnDefinition Width="*"/>
<ColumnDefinition Width="110"/>
<ColumnDefinition Width="200"/>
</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
<RowDefinition Height="40"/>
<RowDefinition Height="*"/>
<RowDefinition Height="*"/>
<RowDefinition Height="*"/>
<RowDefinition Height="*"/>
</Grid.RowDefinitions>
<StackPanel Orientation="Horizontal"
Grid.Row="0" Grid.Column="0"

```

```

        Grid.ColumnSpan="4">
<Label Content="Пік Бєрєнь"
        Margin="5"/>
<ComboBox Grid.Row="0"
        Width="140" Height="30"
        Margin="10"
        x:Name="enter"
        SelectionChanged="Enter_SelectionChanged"
        />
<Label Content="Група"
        Margin="5"/>
<ComboBox Grid.Row="0"
        Width="140" Height="30"
        Margin="10"
        x:Name="gropcb"
        SelectionChanged="Gropcb_SelectionChanged"/>

</StackPanel>
<DataGrid x:Name="FacultyData"
        Grid.Column="0" Grid.Row="1"
        Grid.RowSpan="4" Grid.ColumnSpan="5"
        SelectionChanged ="data_SelectionChanged"
        >
</DataGrid>
</Grid>
</Grid>
</UserControl>

```

Student.xaml

```

<UserControl x:Class="DEEP.View.Student"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:local="clr-namespace:DEEP.View"
        mc:Ignorable="d"
        d:DesignHeight="450" d:DesignWidth="800">
<Grid Background="Gray">
<Grid.RowDefinitions>
<RowDefinition Height="50"/>
<RowDefinition Height="*/>
<RowDefinition Height="*/>

```

```

        <RowDefinition Height="40"/>
        <RowDefinition Height="40"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="100"/>
        <ColumnDefinition Width="200"/>
        <ColumnDefinition Width="*/>
        <ColumnDefinition Width="*/>
        <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>

    <DataGrid Grid.Row="1" Grid.RowSpan="4"
        Grid.Column="2" Grid.ColumnSpan="4"
        x:Name="studentTable"
        CurrentCellChanged="StudentTable_CurrentCellChanged"
        />

    <StackPanel Grid.Row="3" Orientation="Horizontal"
        HorizontalAlignment="Center"
        Grid.ColumnSpan="2">
        <Button x:Name="updateB" Content="Оновити дані" Grid.Row="4"
            HorizontalAlignment="Left" Margin="4" Click="UpdateB_Click"/>
        <Button x:Name="AddB" Content="Додати дані " Grid.Row="4"
            HorizontalAlignment="Left" Margin="4" Click="AddB_Click"/>
        <Button x:Name="DeletB" Content="Видалити дані" Grid.Row="4"
            HorizontalAlignment="Left" Margin="4" Click="DeletB_Click"/>
    </StackPanel>

    <StackPanel Grid.Row="1" Grid.RowSpan="2"
        Grid.Column="1"
        Orientation="Vertical">

        <TextBox x:Name="surTXT" Margin="5" Height="20"
            Text="{Binding SelectedItem.Прізвище,
            ElementName=studentTable}"/>
        <TextBox x:Name="NameTXT" Margin="5" Height="20"
            Text="{Binding SelectedItem.Імя,
            ElementName=studentTable}" />
        <TextBox x:Name="SecondTXT" Margin="5" Height="20"
            Text="{Binding SelectedItem.Побатькові,
            ElementName=studentTable}"/>
        <TextBox x:Name="BurnTXT" Margin="5" Height="20"

```

```
Text="{Binding SelectedItem.РікНародження,
ElementName=studentTable}"/>
<TextBox x:Name="EntryTXT" Margin="5" Height="20"
Text="{Binding SelectedItem.РікВступу,
ElementName=studentTable}"/>
<ComboBox x:Name="FacCB" Margin="5" Height="20"
SelectionChanged="FacCB_Selected"
Text="{Binding SelectedItem.Факультет,
ElementName=studentTable}"/>
<ComboBox x:Name="SpecCB" Margin="5" Height="20"
SelectionChanged="SpecCB_SelectionChanged"
Text="{Binding SelectedItem.Спеціальність,
ElementName=studentTable}"/>
<TextBox x:Name="GrCB" Margin="5" Height="20"
Text="{Binding SelectedItem.Група,
ElementName=studentTable}"/>
<TextBox x:Name="idT" Margin="5" Height="0"
Width="0"
Text="{Binding SelectedItem.ID,
ElementName=studentTable}"/>
</StackPanel>

<StackPanel Grid.Column="0" Grid.Row="1"
Grid.RowSpan="2"
Orientation="Vertical" HorizontalAlignment="Right">
<Label Content="Прізвище" Margin="2"/>
<Label Content="Ім`я" Margin="2"/>
<Label Content="По-батькові" Margin="2"/>
<Label Content="Рік народження" Margin="2"/>
<Label Content="Рік вступу" Margin="2"/>
<Label Content="Факультет" Margin="2"/>
<Label Content="Спеціальність" Margin="2"/>
<Label Content="Група" Margin="2"/>
</StackPanel>

</Grid>
</UserControl>
```

Student.xaml.cs

```
public partial class Student : UserControl
{
    public Student()
    {
        InitializeComponent();
        student();
        FillComboFac();
    }

    void student()
    {
        string connstr = "server = localhost; user = root; database = systemstudent; password = !QAZpassSQL1";
        string sqlFS = " SELECT" +
            "`systemstudent`.`student`.`idStudent` AS `ID`,`" +
            "`systemstudent`.`student`.`Surname` AS `Прізвище`,`" +
            "`systemstudent`.`student`.`Name` AS `Імя`,`" +
            "`systemstudent`.`student`.`SecondName` AS `Побатькові`,`" +
            "`systemstudent`.`student`.`BurnYear` AS `РікНародження`,`" +
            "`systemstudent`.`student`.`EntryYear` AS `РікВступу`,`" +
            "`systemstudent`.`specialty`.`Specialtyname` AS `Спеціальність`,`" +
            "`systemstudent`.`faculty`.`FacultyNameshort` AS `Факультет`,`" +
            "`systemstudent`.`student`.`group` AS `Група`" +
            " FROM((`systemstudent`.`student`" +
```

```

        "JOIN
        `systemstudent`.`specialty`
ON((`systemstudent`.`student`.`Specialty_idSpecialty`
`systemstudent`.`specialty`.`idSpecialty`)))" +

        "JOIN
        `systemstudent`.`faculty`
ON((`systemstudent`.`specialty`.`Faculty_idFaculty`
`systemstudent`.`faculty`.`idFaculty`)))";

        MySqlConnection conn = new MySqlConnection(connstr);
        conn.Open();

        MySqlCommand comF = new MySqlCommand(sqlFS, conn);
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(comF);
        DataTable dt = new DataTable();
        dataAdapter.Fill(dt);
        studentTable.ItemsSource = dt.DefaultView;
        conn.Close();
    }

    private void AddB_Click(object sender, RoutedEventArgs e)
    {
        int x = 0;
        int m = SpecCB.SelectedIndex;
        int n = FacCB.SelectedIndex;
        try
        {
            .....
        }
    }

```

```

    }

    string connstr = "server = localhost; user = root; database = systemstudent;
password = !QAZpassSQL1";

    string sqlST = "INSERT INTO systemstudent.student (`Name`, Surname,
SecondName, BurnYear, EntryYear, Specialty_idSpecialty, `group`)" +
        " VALUES ('"+NameTXT.Text+"', '"+surTXT.Text + "', '" + SecondTXT.Text
+ "', '" + BurnTXT.Text+"', '"+EntryTXT.Text+"', '"+x+"', '"+GrCB.Text+"'); ";

    MySqlConnection conn = new MySqlConnection(connstr);
    conn.Open();

    MySqlCommand comSt = new MySqlCommand(sqlST, conn);
    comSt.ExecuteNonQuery();
    conn.Close();

    NameTXT.Clear();
    surTXT.Clear();
    SecondTXT.Clear();
    BurnTXT.Clear();
    EntryTXT.Clear();
    FacCB.Items.Clear();
    SpecCB.Items.Clear();
    GrCB.Clear();

    MessageBox.Show("Дані успішно додані");
    student();
    FillComboFac();
}

```

```
string connstr = "server = localhost; user = root; database = systemstudent;  
password = !QAZpassSQL1";  
  
string sqlST = "UPDATE systemstudent.student SET `Name` =  
\""+NameTXT.Text+"\", `Surname` = \""+surTXT.Text+"\", `SecondName` =  
\""+SecondTXT.Text+"\", \" +  
    `BurnYear` = \""+BurnTXT.Text+"\", `EntryYear` = \""+EntryTXT.Text+"\",  
`Specialty_idSpecialty` = \""+x+"\", `group` = \""+GrCB.Text+"\" WHERE idStudent =  
\""+idT.Text+"\";";  
  
MySqlConnection conn = new MySqlConnection(connstr);  
conn.Open();  
  
MySqlCommand comSt = new MySqlCommand(sqlST, conn);  
comSt.ExecuteNonQuery();  
conn.Close();  
  
NameTXT.Clear();  
surTXT.Clear();  
SecondTXT.Clear();  
BurnTXT.Clear();  
EntryTXT.Clear();  
FacCB.Items.Clear();  
SpecCB.Items.Clear();  
GrCB.Clear();  
MessageBox.Show("Дані успішно оновлені");  
student();  
FillComboFac();
```