

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

Розробка системи управління базою даних гуртожитку №3 КНТЕУ

Студентки 4 курсу, 7 групи,
напряму підготовки 6. 050103
«Програмна інженерія»

підпис студентки

Гаврилюк Яни
Миколаївни

Науковий керівник доктор
технічних наук, професор

підпис керівника

Криворучко
Олена
Володимирівна

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ – 2019

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ОСНОВНІ ВІДОМОСТІ ПРО ДОДАТОК.....	6
1.1 Дослідження проблеми.....	6
1.2 Технічне завдання на розробку системи управління базою даних гуртожитку №3 КНТЕУ.....	7
1.3 Висновки до Розділу 1.....	11
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ ГУРТОЖИТКУ №3 КНТЕУ.....	12
2.1 Мова програмування C#.....	12
2.2 Порівняння мови C# з іншими мовами програмування.....	12
2.3 Системи управління базами даних.....	14
2.4 Висновки до Розділу 2.....	15
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ ГУРТОЖИТКУ №3 КНТЕУ.....	17
3.1 Екран логіну.....	17
3.2 Головний екран.....	18
3.3 Додати студента.....	21
3.4 Деталі кімнати.....	22
3.5 Висновки до Розділу 3.....	24
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	25
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	26
ДОДАТКИ.....	27

					КНТЕУ 6.050103 07-03.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка системи управління базою даних гуртожитку №3 КНТЕУ	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				Зміст	2	26
Керівник		Криворучко О.В				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Гаврилюк Я.М.						
					Зміст			

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КНТЕУ	—	Київський національний торговельно-економічний університет.
БД	—	База даних.
ПК	—	Персональний комп'ютер.
ПЗ	—	Програмне забезпечення.
РСС	—	Рада студентського самоврядування
SQL	—	Structured query language.
СУБД	—	Система управління базами даних
РСУБД	—	Реляційна система управління базами даних.

					<i>КНТЕУ 6.050103 07-03.БР</i>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.		Криворучко О.В			Розробка системи управління базою даних гуртожитку №3 КНТЕУ	Стадія	Аркуш	Акрушів
Керівник		Криворучко О.В				ПС	3	26
Гарант		Цензура М.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив		Гаврилюк Я.М						
					Перелік умовних позначень			

ВСТУП

В кожній установі, організації, на підприємстві документи перебувають в постійному обігу. Для цього потрібно створення оптимальних умов для всіх видів робіт з документами на всіх етапах опрацювання документів – від моменту створення чи одержання документа до його знищення або відправлення на архівне зберігання. Для роботи вищого навчального закладу, зокрема, для гуртожитку важливою складовою є документація, яка містить усю інформацію про мешканця, а також про стан кожної кімнати.

Процес створення такої документації – складний та трудомісткий та вимагає значних людських ресурсів від особи, що займається її веденням. Великі обсяги інформації, особливості її подання – це ті фактори, що підвищують можливість виникнення помилок та суперечностей. Це, та той факт, що у більшості документів використовується однотипна інформація, приводить до думки про можливість створення програмного продукту, що дозволить автоматизувати більшу частину роботи з документами, що розглядаються. Застосування бази даних для збереження інформації допоможе структурувати її, запобігти суперечностям та полегшити роботу персоналу. Інтерфейс користувача дозволить легко редагувати інформацію, що знаходиться у БД без використання відповідних засобів СУБД, зробить роботу з даними більш наочною та інтуїтивно зрозумілою.

Актуальність: полягає у дослідженні роботи гуртожитку київського національного торговельно-економічного університету, та

					КНТЕУ 6.050103 07-03.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка системи управління базою даних гуртожитку №3 КНТЕУ	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				В	4	26
Керівник		Криворучко О.В				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Гаврилюк Я.М			Вступ			

розробки програмного забезпечення для ПК, яке б допомогло персоналу гуртожитку в роботі, зберігаючи всю необхідну інформацію в комп'ютері.

Об'єкт дослідження: система організації функціонування гуртожитку КНТЕУ.

Предмет дослідження: додаток, створений мовою програмування C#.

Мета роботи: дослідження алгоритмів, інструментів та засобів для створення додатку.

Завдання дослідження:

- дослідження роботи гуртожитку №3 КНТЕУ
- розробка додатку для спрощення роботи адміністрації гуртожитку мовою програмування C#;
- робота в середовищі розробки Microsoft Visual Studio;
- розробка додатку з використанням бази даних SQL;
- робота з інструментом для візуального проектування баз даних MySQL Workbench;
- тестування додатку.

Методи і технології розробки: візуальне і об'єктно-орієнтоване програмування, порівняння, спостереження, побудова програмного продукту на основі проаналізованих даних, використання сучасних технологій, аналіз алгоритмів пошуку та сортування та структур даних.

Результат роботи: створено програмне забезпечення «Система управління базою даних гуртожитку №3 КНТЕУ».

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Аркуш	№ Докум	Подпись	Дат		5

РОЗДІЛ 1.

ОСНОВНІ ВІДОМОСТІ ПРО ДОДАТОК

1.1. Дослідження проблеми

Існує чітка система здійснення всіх послідовних операцій з документами: приймання і реєстрація, розгляд керівником, перевірка правильності оформлення, формування справ, підготовка і передача справ до архіву. В гуртожитку №3 проживає 1175 студентів. Їхні дані зберігаються в паперовому вигляді, що ускладнює пошук та роботу з документацією. Також, існує проблема серед мешканців гуртожитку, а саме незнання зауважень з приводу стану кімнати. В кінці року під час підпису заяв на наступний навчальний рік, саме незнання даної інформації призводить до хаосу між студентами та непорозумінь з персоналом гуртожитку.

Додаток «Система управління базою даних гуртожитку №3 КНТЕУ», був розроблений на основі дослідження роботи студмістечка та його потреб. Інколи виникали задачі, які потребували швидкого вирішення. Проте, через нестачу інформації під рукою, вони займали більше часу. Додаток був розроблений для ПК з операційною системою Windows, що є найбільш поширеною серед працівників та мешканців гуртожитку.

Додаток являє собою довідник, в якому зберігається інформація про мешканців. Він дозволяє швидко знайти інформацію про студентів, їх персональні дані, посаду та стан кімнати, в якій вони проживають.

					<i>КНТЕУ 6.050103 07-03.БР</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка системи управління базою даних гуртожитку №3 КНТЕУ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушіє</i>
<i>Зав. Каф.</i>		<i>Криворучко О.В</i>				<i>РІ</i>	<i>6</i>	<i>26</i>
<i>Керівник</i>		<i>Криворучко О.В</i>				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група</i>		
<i>Гарант</i>		<i>Цензура М.О.</i>						
<i>Розробив</i>		<i>Гаврилюк Я.М</i>			<i>Розділ 1</i>			

1.2. Технічне завдання на розробку системи управління базою даних гуртожитку №3 КНТЕУ

1. Загальні положення

1.1. Найменування програмного продукту.

Найменування програми — «Система управління базою даних гуртожитку №3 КНТЕУ»

1.2. Коротка характеристика області застосування.

Програма призначена для внесення інформації про мешканців гуртожитку та їхні кімнати в базу даних.

2. Підстави для розробки

2.1. Підстава для проведення розробки

Підставою для розробки є проведена робота з базою даних в гуртожитку №3. І для забезпечення швидкого доступу до інформації та полегшеної роботи з нею, було розроблено даний додаток.

2.2. Найменування та умовне позначення теми розробки.

Найменування теми розробки - розробка системи управління базою даних гуртожитку №3 КНТЕУ. Умовне позначення теми розробки – «Система управління базою даних гуртожитку №3 КНТЕУ».

3. Призначення розробки

3.1. Функціональне призначення

Полягає в фіксації наступних даних: ПІБ мешканця, його факультет, курс, групу, посаду, кімнату та її санітарний стан.

3.2. Експлуатаційне призначення

Програма повинна експлуатуватися в гуртожитку №3 завідувачем гуртожитку, кастелянкою, радою студентського самоврядування та мешканцями даного гуртожитку.

4. Вимоги до програми або програмного виробу

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Аркуш	№ Докум	Подпись	Дат		7

4.1.Вимоги до функціональних характеристик;

4.1.1.Вимоги до складу виконуваних функцій

Програма повинна забезпечувати можливість виконання перерахованих нижче функцій:

- Система повинна володіти набором зручних можливостей щодо введення інформації в систему;
- ПЗ повинно зберігати введену інформацію в базу даних, та при необхідності відображати її.
- Система повинна мати зручний інтерфейс.
- База даних не має обмежень по кількості введених записів.
- При необхідності, користувач «Адміністрація» може видалити запис про мешканця або редагувати його, якщо інформація потребує змін.
- Користувач «РСС» може додавати, редагувати та видаляти інформацію лише щодо санітарного стану.
- Користувач «Мешканець» може переглядати дані.

4.1.2.Вимоги до організації вхідних даних

- Форма введення: таблиця з полями для внесення даних.
- Інформація швидко зберігається в базу та не потребує зайвого часу.
- Записи зберігаються доти, доки додаток не буде видалено з пристрою.

4.2.Вимоги до надійності;

4.2.1.Вимоги до забезпечення надійного функціонування програми

1. Передбачити надійний захист даних від несанкціонованого доступу:

- На додаток, для забезпечення безпеки даних, встановлено вхід під користувачами «Адміністрація» та «РСС» після введення паролю.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Архив	№ Докум	Подпись	Дат		8

2. Забезпечити цілісність зберігання в базі даних інформації.

4.3.Умови експлуатації;

4.3.1.Кліматичні умови експлуатації

Залежить від пристрою, на якому встановлено дане ПЗ

4.3.2.Вимоги до видів обслуговування

Не передбачені

4.3.3.Вимоги до чисельності та кваліфікації персоналу задіяного в штаті

Вводити дані може завідувач гуртожитку, кастиелянка та РСС.

4.4.Вимоги до складу і параметрів технічних засобів;

Вимоги до технічних засобів: ПК з операційною системою Windows.

4.5.Вимоги до інформаційної та програмної сумісності;

4.5.1.Вимоги до інформаційних структур

Введення даних підтримує будь-яку мову.

4.5.2.Вимоги до вихідного коду і мов програмування

ПЗ розроблено на мові програмування C# в середовищі розробки Microsoft Visual Studio.

4.6.Спеціальні вимоги

Програма повинна забезпечувати взаємодію з користувачем (оператором) за допомогою графічного інтерфейсу користувача.

5.Техніко-економічні показники

Техніко-економічні показники додатку для гуртожитку не визначаються.

ПЗ надається безкоштовно.

6.Стадії і етапи розробки

6.1.Стадії розробки

Розробка повинна бути проведена в три стадії:

— технічне завдання;

— технічний (і робочий) проекти;

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		9

— впровадження.

6.2.Етапи розробки

На стадії «Технічне завдання» повинен бути виконаний етап розробки, узгодження і затвердження цього технічного завдання.

На стадії «Технічний (і робочий) проект» повинні бути виконані перераховані нижче етапи робіт:

- розробка програми;
- розробка програмної документації;
- випробування програми.

На стадії «Впровадження» повинен бути виконаний етап розробки «Підготовка та передача програми».

6.3. Зміст робіт по етапах

На етапі розробки техзавдання повинні бути виконані перераховані нижче роботи:

- постановка задачі;
- визначення та уточнення вимог до технічних засобів;
- визначення вимог до програми;
- визначення стадій, етапів і термінів розробки програми і документації на неї;
- вибір мов програмування;
- узгодження і затвердження технічного завдання.

На етапі розробки програми повинна бути виконана робота по програмуванню (кодування) і налагодженні програми.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		10

На етапі розробки програмної документації повинна бути виконана розробка програмних документів відповідно до вимог ГОСТ 19.101-77.

На етапі випробувань програми повинні бути виконані перераховані нижче види робіт:

- розробка, узгодження та затвердження програми в ГОСТ
- проведення приймально-здавальних випробувань;
- коригування програми і програмної документації за результатами випробувань.

На етапі підготовки і передачі програми повинна бути виконана робота по підготовці і передачі програми і програмної документації в експлуатацію на об'єктах замовника.

1.3. Висновки до Розділу 1

Провівши аналіз системи зберігання даних в гуртожитку №3 Київського національного торговельно-економічного університету, були визначені вимоги, яким має відповідати створене програмне забезпечення.

Основною задачею стало розробка програмного забезпечення, в якому усі дані про мешканців гуртожитку буде впорядковано, що дає можливість швидкого доступу до даних.

Розроблене технічне завдання допоможе створити хороше програмне забезпечення, яким без проблем зможуть користуватися співробітники та мешканці гуртожитку. Дотримання всіх стандартів технічного завдання, допоможе створити зручний та справний додаток.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		11

РОЗДІЛ 2.

ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ ГУРТОЖИТКУ №3 КНТЕУ

2.1. Мова програмування C#

Дослідженням інфраструктури КНТЕУ, зокрема студмістечка, було з'ясовано, що адміністрація гуртожитку користується ПК з операційною системою Windows. Виходячи з цього, було обрано мову програмування C#.

C # є об'єктно-орієнтованою мовою, але підтримує також і компонентно-орієнтоване програмування. Розробка сучасних додатків все більше тяжіє до створення програмних компонентів у формі автономних і самоопісательних пакетів, що реалізують окремі функціональні можливості. Важлива особливість таких компонентів - це модель програмування на основі властивостей, методів і подій. Кожен компонент має атрибути, які надають декларативні відомості про компоненті, а також вбудовані елементи документації. C # надає мовні конструкції, безпосередньо підтримують таку концепцію роботи. Завдяки цьому C # відмінно підходить для створення і застосування програмних компонентів.

2.2. Порівняння мови C# з іншими мовами програмування

Варто підтримувати C# при розробці настільних додатків і, можливо, ігор, але не для вбудованих систем (таких як мікроконтролерів), в режимі реального часу додатків або ядер, так що наступні переваги будуть відноситися тільки до настільних додатків.

					КНТЕУ 6.050103 07-03.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка системи управління базою даних гуртожитку №3 КНТЕУ	Стадія	Аркуш	Акрушіє
Зав. Каф.		Криворучко О.В				Р2	12	26
Керівник		Криворучко О.В				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Гаврилюк Я.М			Розділ 2			

Переваги C# в порівнянні з C і C ++:

- Класи можуть бути визначені всередині класів
- Можливості відображення
- Не потрібно турбуватися про заголовні файли “.h”
- Визначення класів і функцій можуть бути зроблені в будь-якому порядку
- Декларація функцій і класів не вимагається
- Неіснуючі циклічні залежності
- Автоматичний збір сміття
- Немає глобальних функцій або змінних, все належить до класу
- Всі змінні ініціалізуються значеннями за замовчуванням, перш ніж використовувати (це автоматично, за замовчуванням, але може бути зроблено вручну, використовуючи статичні конструктори)
- Не можна використовувати не логічні змінні (цілі), як умови. Це набагато чистіше і менш схильно до помилок
- Переваги в порівнянні з C++ і Java
- Формалізоване поняття GET-набір методів, так що код стає більш розбірливим
- Більш чисте управління подіями (за допомогою делегатів)

Переваги C# в порівнянні з Java

- Зазвичай набагато ефективніший, ніж Java і працює швидше
- Є стандартною мовою
- Має більш примітивні типи (типи значень)
- Індексатори дозволяють отримати доступ до об'єктів, як якщо б вони були масивами
- Умовна компіляція
- Спрощена багатопоточність

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Архивш	№ Докум	Подпись	Дат		13

- Перевантаження операторів. Це може зробити розробку трохи складнішою, але вони не є обов'язковими, а іноді й дуже корисними
- Використання показників — обмежене, якщо вони вам дійсно потрібні, а при дзвінку некеровані бібліотеки, які не працюють на верхній частині віртуальної машини.

2.3. Системи управління базами даних

Робота з великою кількістю документації є дуже трудомісткий процес. Для того, щоб систематизувати, обробляти та модифікувати великий об'єм інформації, було обрано комерційну систему керування базами даних Microsoft SQL Server

SQL Server підтримує дзеркалювання та кластеризацію баз даних. Кластер серверу SQL — це сукупність однаково конфігурованих серверів, така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server 2005 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Архив	№ Докум	Подпись	Дат		14

Для реалізації DbHelper використано Singleton. Патерн Singleton є одним з найбільш відомих шаблонів в розробці програмного забезпечення. По суті Singleton - це клас, який дозволяє створювати тільки один екземпляр і зазвичай надає простий доступ до цього примірника.

Найчастіше Singleton не дозволяє вказувати будь-які параметри при створенні екземпляра, оскільки в іншому випадку повторний запит для створення екземпляра з іншими параметрами може бути проблематичним! Якщо до того ж екземпляру потрібно отримати доступ для всіх запитів з одним і тим же параметром, більш відповідним є шаблон фабрики.

Існують різні способи реалізації Singleton в C #. Більшість реалізацій мають чотири загальні характеристики:

- Єдиний конструктор, який є закритим (модифікатор private) і без параметрів. Це запобіжить створення інших примірників (що було б порушенням паттерна).
- Клас повинен бути запечатаним (модифікатор sealed). Строго кажучи це є необов'язковою умовою, виходячи з вищевикладених концепцій Singleton, але дозволяє JIT-компілятор поліпшити оптимізацію.
- Змінна, яка містить посилання на створений екземпляр, повинна бути статичною.
- Необхідно відкрите (public) статичне властивість, яке буде містити посилання на створений екземпляр.

2.4. Висновки до Розділу 2

C # — це повнофункціональна об'єктно-орієнтована мова, що підтримує всі три «стовпи» об'єктно-орієнтованого програмування: інкапсуляцію, успадкування і поліморфізм. Вона має прекрасну

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		15

підтримку компонентів, надійний і стійкий завдяки використанню «складання сміття», обробки винятків, безпеки типів. Код буде більш надійний, чистий. Його написання буде швидшим і комфортнішим, ніж в інших мовах. Завдяки цьому, C# найбільш підходить для написання системи управління базою даних гуртожитку №3 КНТЕУ.

SQL Server реляційною системою управління базами даних, тобто РСУБД. Усі дані в таких базах даних зберігаються в плоских двовірних таблицях. В основі реляційних систем лежить математичний апарат. MS SQL Server відмінно підходить для систематизування, нагромадження і раціонального використання інформації про мешканців.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		16

РОЗДІЛ 3.

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ ГУРТОЖИТКУ №3 КНТЕУ

3.1. Екран логіну

Логін – це головний екран, який користувач бачить першим після запуску додатка.

На даній формі знаходяться елементи полів вводу паролю або номеру кімнати та кнопка входу. На цьому етапі користувач може увійти за допомогою паролю, як «Адміністрація» або «РСС», або ввести номер кімнати.

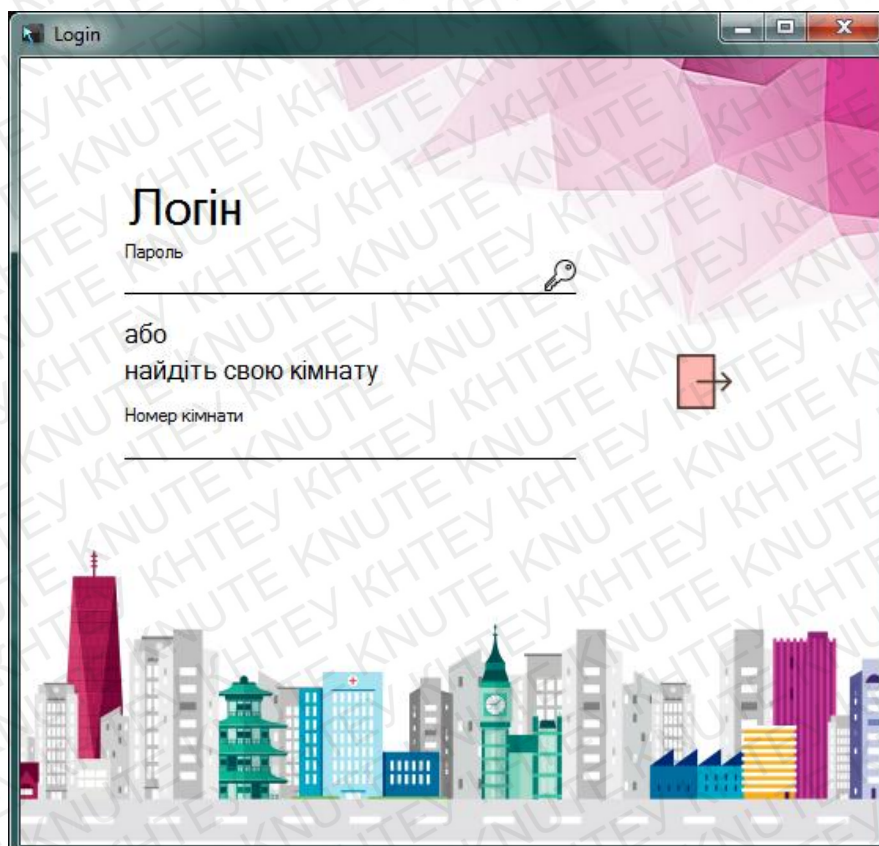


Рис 3.1.1 Логін

					<i>КНТЕУ 6.050103 07-03.БР</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.		Криворучко О.В			Розробка системи управління базою даних гуртожитку №3 КНТЕУ Розділ 3	Стадія	Аркуш
Керівник		Криворучко О.В				РЗ	17
Гарант		Цензура М.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група	
Розробив		Гаврилюк Я.М					
						Акрушів	26

Якщо пароль введено невірно, на екрані з'являється MessageBox з відповідним повідомленням.

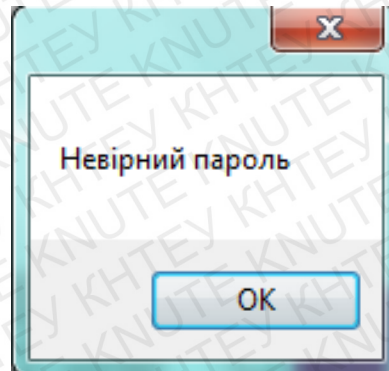


Рис 3.1.2 MessageBox пароль

При введенні некоректного значення номеру кімнати, виводиться MessageBox з наступним повідомленням.

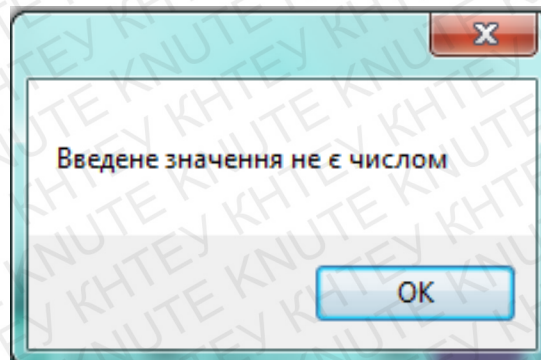


Рис 3.1.3 MessageBox кімната

3.2. Головний екран

Вхід до головного ерану відбувається за різним типом доступу, а саме:

- «Адміністрація»;
- «РСС»;
- «Гість».

При введенні невірної кімнати, користувачу потрібно перевірити коректність написання номеру.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Аркуш	№ Докум	Подпись	Дат		18

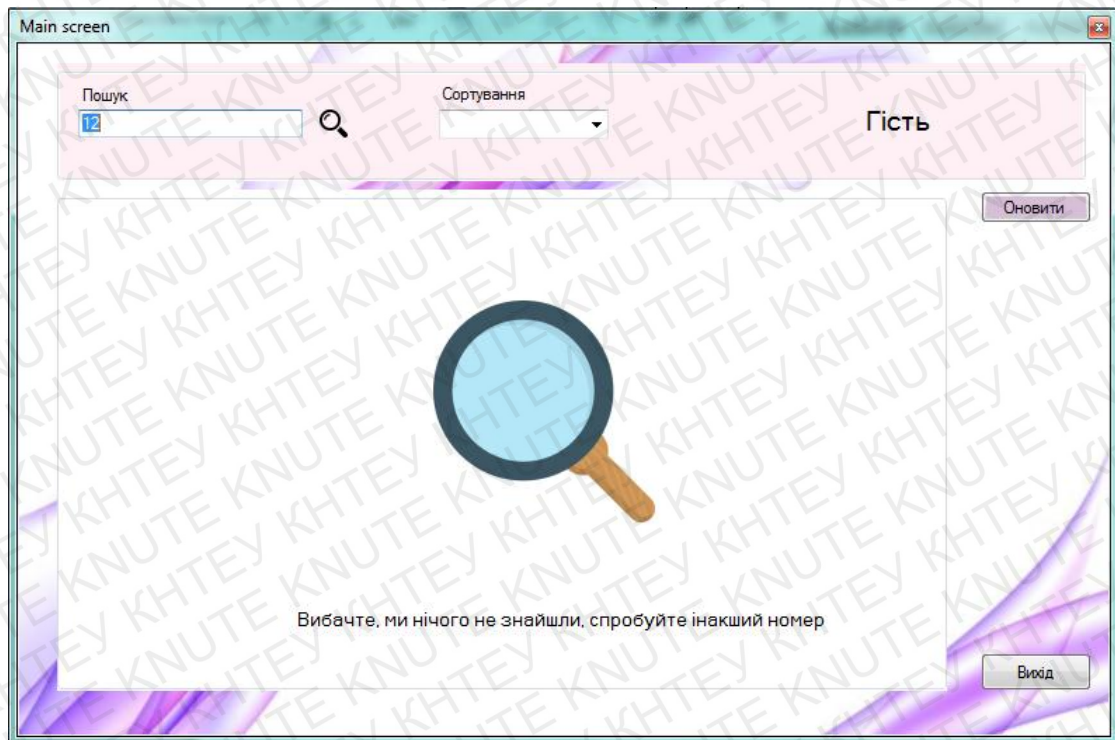


Рис 3.2.1 Головний екран (некоректний номер кімнати)

Користувач «Гість» має право переглядати усі дані, виконувати пошук мешканців за їхньою кімнатою. Також, можливе сортування мешканців за номером кімнати або по імені. Користувач може встановити вибірку по факультетах та курсах.

За допомогою кнопки «Оновити» усі дані оновлюються. Після натискання на кнопку «Деталі кімнати» користувач переходить на форму «Кімната».

При виборі кнопки «Вихід» відбувається перехід до форми «Логін».

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Аркуш	№ Докум	Подпись	Дат		19

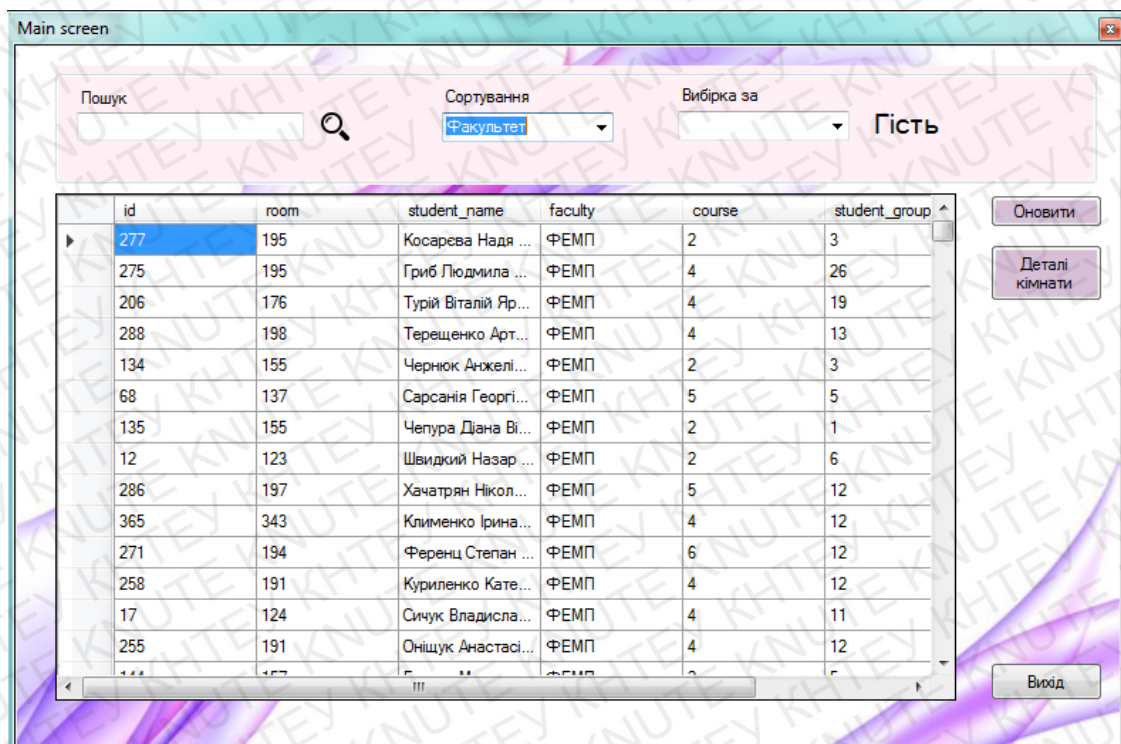


Рис 3.2.2 Головний екран(«Гість»)

На даному етапі інтерфейс та право доступу «РСС» нічим не відрізняється від типу «Гість».

Користувач «Адміністрація» окрім вищеперерахованих функцій, має право додавати або видаляти студентів відповідними кнопками.

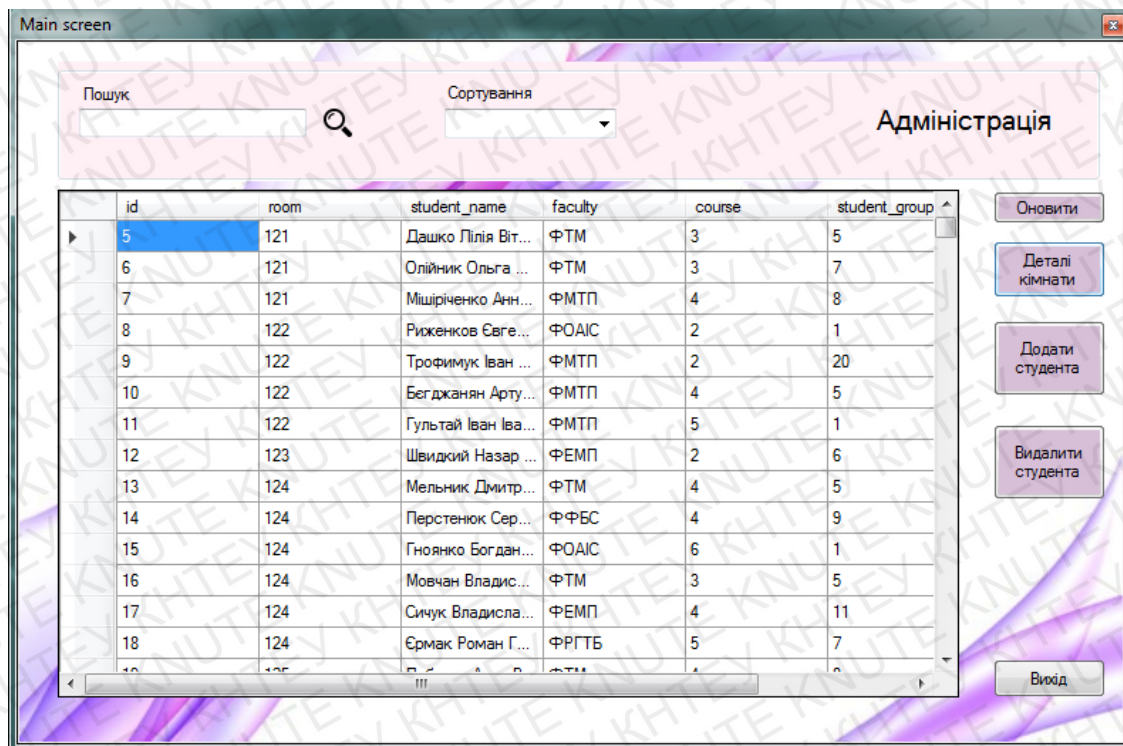


Рис 3.2.3 Головний екран(«Адміністрація»)

3.3 Додати студента

Екран додавання мешканця доступний лише користувачу «Адміністрація». Він дає можливість коректно заповнити дані мешканця гуртожитку.

Рис 3.3 Додавання студента

3.3 Деталі кімнати

Після натискання на відповідну кнопку на формі «Головний екран», користувачу виводиться екран з відповідним номером кімнати, дані про її стан та перелік жителів.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Аркуш	№ Докум	Подпись	Дат		22

Room

236

room_number	comment
236	

student_name	faculty	course	student_group
Лисяна Олекса...	ФТМ	4	11
Карпенко Надія ...	ФТМ	4	11
Звержівська К...	ФТМ	1	19
Бортник Світлана	ФЕМП	2	7

+

Рис 3.4.1 Деталі кімнати

Типи доступу «Адміністрація» та «РСС» мають право додавати, редагувати та видаляти інформацію про санітарний стан кімнати та інші коментарі. «Гість» може лише переглядати дану інформацію.

Натиснувши на «+», з'являється вікно для введення коментаря.

Comment

Коментар

Змінити

Рис 3.4.2 Коментар

Після редагування поля «Коментар», користувач натискає на кнопку «Змінити»

Кожний елемент має кнопку «Додатково», при натисканні якої з'являється діалогове вікно з інформацією про конкретний об'єкт.

Присутній функціонал пошуку по даних, який доступний по натисканню на кнопку «Пошук».

3.4 Висновки до Розділу 3

Розроблений додаток побудований відповідно до технічного завдання, використовуючи новітні технології, ресурси та інструменти.

Описана розробка, функціонал та алгоритм програми. Робота виконана об'єктно-орієнтованою мовою програмування C# в середовищі Microsoft Visual Studio 2015 за допомогою Microsoft SQL Server. Програма була успішно протестована та скомпільована та готова до повноцінної роботи.

					КНТЕУ 6.050103 07-03.БР	Лист
Изм.	Аркуш	№ Докум	Подпись	Дат		24

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У даній роботі проводилася розробка системи управління базою даних гуртожитку №3 КНТЕУ – бази даних та інтерфейсу користувача.

Для виконання поставленого завдання на першому етапі роботи було досліджено проблему та актуальність додатку.

На основі отриманих результатів першої частини роботи, а саме проаналізованих обсягів даних, які містяться у відповідних документах. Проведено дослідження наявних моделей баз даних, та обрано найбільш зручну для реалізації поставленого завдання – реляційну модель. Спираючись на це, було проведено аналіз доступних для використання в розробці систем управління базами даних. Обрана СУБД MySQL задовольняє своїм функціоналом, допоміжним програмним забезпеченням, а також доступністю довідкових матеріалів потреби, що виникають при вирішенні даної задачі. Також дана СУБД є безкоштовною, тобто не накладає фінансового навантаження на гуртожиток при її використанні.

Останнім етапом розробки системи була побудова інтерфейсу користувача. Створений інтерфейс відповідає клієнтській частині клієнт-серверної архітектури. При розробці використані сучасні інструменти веб-розробки, що дозволило створити простий та зрозумілий інтерфейс користувача.

Отже, усі поставлені технічні завдання були виконані. Після затвердження програми керівництвом університету, її буде введено в подальшу експлуатацію.

					КНТЕУ 6.050103 07-03.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка системи управління базою даних гуртожитку №3 КНТЕУ Висновки та пропозиції	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				ВП	25	26
Керівник		Криворучко О.В				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Гаврилюк Я.М.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основний:

1. Microsoft SQL Server 2005. / Nelsen Pol – O`Reilly, 2007. - 1232 с.
2. Програмування баз даних MS SQL Server 2005. Базовий курс/ Роберт Виейра – К. : Ентроп, 2007. – 832 с.
3. С# в короткому викладенні. / Бішоп Дж., Хорспул Н. – К. : Лабораторія знань, 2005. – 472с.
4. Повний довідник по С#. / Шилд Г. – К. : Вільямс, 2004. – 752с.
5. С# і платформа.NET. Бібліотека програміста. / Троелсен Е. – К. : Ентроп, 2004. – 796с.
6. Visual С# .NET. Повне керівництво. / Прайм Д., Гандерлой М. – Х. : НТІ, Ентроп, 2004. – 960с.
7. Базы данных. Учебный курс / Глушаков С.В., Ломотько Д.В. – К. : Фолио, Х. : Абрис, 2000. – 504 с.

Інтернет ресурси:

1. Microsoft Visual Studio [Електронний ресурс]. – Режим доступу: URL: http://uk.wikipedia.org/wiki/Microsoft_Visual_Studio.
2. Краткий обзор языка С# [Електронний ресурс]. – Режим доступу: URL: <http://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>.
3. Одиночка (Singleton) [Електронний ресурс]. – Режим доступу: URL: <http://metanit.com/sharp/patterns/2.3.php>

					<i>КНТЕУ 6.050103 07-03.БР</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.		Криворучко О.В			Розробка системи управління базою даних гуртожитку №3 КНТЕУ Список використаних джерел	Стадія	Аркуш
Керівник		Криворучко О.В				СД	26
Гарант		Цензура М.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група	
Розробив		Гаврилюк Я.М.					

DbHelper

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Threading.Tasks;
using MySql.Data;
using MySql.Data.MySqlClient;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Collections.Generic;

namespace Diploma
{
    public class DbHelper
    {
        private static DbHelper dbHelper;

        public static DbHelper getInstance()
        {
            if(dbHelper == null)
            {
                dbHelper = new DbHelper();
                return dbHelper;
            }
            return dbHelper;
        }

        public MySqlConnection connection;

        public DbHelper()
        {
            connection = new MySqlConnection("Server=localhost; database=diploma; userId=YY;
password=Sql343YY2019");
        }

        public DataTable executeSqlQuery(string query)
        {
            connection.Open();
            MySqlDataAdapter adapter = new MySqlDataAdapter(query, connection);
            DataTable table = new DataTable();
            adapter.Fill(table);
            connection.Close();
            return table;
        }

        public int executeSqlForLogin(string password)
        {
            string query = "SELECT * FROM users";
            MySqlCommand cmd = new MySqlCommand(query, connection);
            connection.Open();
            MySqlDataReader reader = cmd.ExecuteReader();
            int userType = 2;
            while (reader.Read())
            {
                string passwordFromDataBase = passwordFromDataBase =
reader.GetString("password");
```

```

        if (passwordFromDataBase.Equals(password))
        {
            userType = reader.GetInt32("isAdmin");
        }
    }
    connection.Close();
    return userType;
}

public void executeSqlForAddStudent(Dictionary<string, string> data)
{
    string name = "";
    data.TryGetValue(Constants.NAME, out name);
    string room = "";
    data.TryGetValue(Constants.ROOM, out room);
    string faculty = "";
    data.TryGetValue(Constants.FACULTY, out faculty);
    string course = "";
    data.TryGetValue(Constants.COURSE, out course);
    string group = "";
    data.TryGetValue(Constants.GROUP, out group);
    string role = "";
    data.TryGetValue(Constants.ROLE, out role);
    string query = "SELECT * FROM users";

    connection.Open();
    MySqlCommand cmd3 = new MySqlCommand("SELECT room_number FROM rooms WHERE
room_number = " + room, connection);
    MySqlDataReader reader = cmd3.ExecuteReader();
    string roomFromDatabase = null;
    while (reader.Read())
    {
        roomFromDatabase = reader.GetString("room_number");
    }
    connection.Close();
    if (roomFromDatabase == null || roomFromDatabase.Equals(""))
    {
        connection.Open();
        MySqlCommand cmd2 = new MySqlCommand("INSERT INTO rooms(room_number,
comment) VALUES (@ROOM, @COMMENT)", connection);
        cmd2.Parameters.AddWithValue("@ROOM", room);
        cmd2.Parameters.AddWithValue("@COMMENT", "nothing");
        cmd2.ExecuteNonQuery();
        connection.Close();
    }
    connection.Open();
    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = connection;
    cmd.CommandText = "INSERT INTO students(room, student_name, faculty, course,
student_group, roles) VALUES (@ROOM, @NAME, @FAC, @COURSE, @GROUP, @ROLE)";
    cmd.Parameters.AddWithValue("@ROOM", room);
    cmd.Parameters.AddWithValue("@NAME", name);
    cmd.Parameters.AddWithValue("@FAC", faculty);
    cmd.Parameters.AddWithValue("@COURSE", course);
    cmd.Parameters.AddWithValue("@GROUP", group);
    cmd.Parameters.AddWithValue("@ROLE", role);
    cmd.ExecuteNonQuery();

    connection.Close();
}

```



```

public void executeSqlForDeleteStudent(String id)
{
    connection.Open();
    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = connection;
    cmd.CommandText = "DELETE FROM students WHERE id='" + id + "'";
    cmd.ExecuteNonQuery();
    connection.Close();
}

public string executeSqlForRoomById(String id)
{
    connection.Open();
    MySqlCommand cmd = new MySqlCommand("SELECT * FROM rooms WHERE id='" + id + "'",
connection);
    MySqlDataReader reader = cmd.ExecuteReader();
    string room = "";
    while (reader.Read())
    {
        room = reader.GetString("room_number");
    }
    return room;
}

public DataTable executeSqlForStudentsByRoom(String room)
{
    connection.Open();
    MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT student_name, faculty,
course, student_group, roles FROM students WHERE room='" + room + "'", connection);
    DataTable table = new DataTable();
    adapter.Fill(table);
    connection.Close();
    return table;
}

public DataTable executeSqlForCommentsByRoom(String room)
{
    connection.Open();
    MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT * FROM rooms WHERE
room_number='" + room + "'", connection);
    DataTable table = new DataTable();
    adapter.Fill(table);
    connection.Close();
    return table;
}

public string executeSqlForCommentStringByRoom(string room)
{
    connection.Open();
    MySqlCommand cmd = new MySqlCommand("SELECT comment FROM rooms WHERE
room_number='" + room + "'", connection);
    MySqlDataReader reader = cmd.ExecuteReader();
    string comment = "";
    while (reader.Read())
    {
        comment = reader.GetString("comment");
    }
    connection.Close();
    return comment;
}

```



```
public string executeSqlForUpdatingComment(string comment, string roomNumber)
{
    connection.Open();
    MySqlCommand cmd = new MySqlCommand("UPDATE rooms SET comment = '" + comment +
    "' WHERE room_number='" + roomNumber + "'", connection);
    cmd.ExecuteNonQuery();
    connection.Close();
    return comment;
}
}
```

Login

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void pictureBox3_Click(object sender, EventArgs e)
        {
            if (!password.Text.Equals(""))
            {
                int userType = Program.dbHelper.executeSqlForLogin(password.Text);
                if (userType == 2)
                {
                    MessageBox.Show("Невірний пароль");
                    return;
                }
                else
                {
                    Form2 f = new Form2(userType);
                    this.Hide();
                    f.Show();
                }
            }
            else
            {
                int roomInt = 0;
                if (!Int32.TryParse(room.Text, out roomInt))
                {
                    MessageBox.Show("Введено значення не є числом");
                    return;
                }
            }
        }
    }
}
```



```
if (!room.Text.Equals(""))
{
    Form2 f = new Form2(room.Text);
    this.Hide();
    f.Show();
}
```


Main screen

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Odbc;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace Diploma
{
    public partial class Form2 : Form
    {
        private string selectedSortItem;
        private int userType = 2;

        public Form2(string room)
        {
            InitializeComponent();
            init(room);
            initUi();
        }

        public Form2(int type)
        {
            InitializeComponent();
            this.userType = type;
            init();
            initUi();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
        }

        private void init(string room)
        {
            search.Text = room;
            string query = "SELECT * FROM students WHERE room LIKE " + room;
            setDataIntoGridView(query);
        }

        private void init()
        {
            string query = "SELECT * FROM students";
            rooms.DataSource = Program.dbHelper.executeSqlQuery(query);
        }
    }
}

```

```

private void initUi()
{
    if (userType == 1)
    {
        userName.Text = "Адміністрація";
        add.Show();
        delete.Show();
        return;
    }
    else if (userType == 0)
    {
        userName.Text = "РСС";
        add.Hide();
        delete.Hide();
        return;
    }
    else
    {
        userName.Text = "Гість";
        add.Hide();
        delete.Hide();
        return;
    }
}

private void rooms_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}

private void label1_Click(object sender, EventArgs e)
{
}

private void provideEmptySearchViewVisibility(bool isVisible)
{
    if (isVisible)
    {
        emptySearch.Show();
        details.Hide();
        rooms.Hide();
        return;
    }
    else
    {
        emptySearch.Hide();
        rooms.Show();
        details.Show();
        return;
    }
}

private void setDataIntoGridView(string query)
{
    DataTable table = Program.dbHelper.executeQuery(query);
    provideEmptySearchViewVisibility(table.Rows.Count == 0);
    rooms.DataSource = table;
}

```



```

private void sort_SelectedIndexChanged(object sender, EventArgs e)
{
    selectedSortItem = sort.GetItemText(sort.SelectedItem);
    StringBuilder builder = new StringBuilder();
    builder.Append("SELECT * FROM students ORDER BY ");
    if (selectedSortItem.Equals(Constants.NAME))
    {
        builder.Append("student_name");
        provideSelectorVisibility(false, null);
    }
    if (selectedSortItem.Equals(Constants.COURSE))
    {
        builder.Append("course");
        provideSelectorVisibility(true, Constants.getListOfCourses());
    }
    if (selectedSortItem.Equals(Constants.FACULTY))
    {
        builder.Append("faculty");
        provideSelectorVisibility(true, Constants.getListOfFaculties());
    }
    if (selectedSortItem.Equals(Constants.ROOM))
    {
        builder.Append("room");
        provideSelectorVisibility(false, null);
    }
    string query = builder.ToString();
    if (query != null)
    {
        setDataIntoGridView(query);
    }
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    if (search.Text.Equals("")){
        MessageBox.Show("Поле не пустое");
        return;
    }
    string query = "SELECT * FROM students WHERE room LIKE " + search.Text;
    setDataIntoGridView(query);
}

private void provideSelectorVisibility(bool isVisible, List<string> items)
{
    if (isVisible)
    {
        select.DataSource = items;
        select.SelectedIndex = -1;
        label14.Show();
        select.Show();
        return;
    }
    else
    {
        label14.Hide();
        select.Hide();
        return;
    }
}

private void select_SelectedIndexChanged(object sender, EventArgs e)
{
    if(select.SelectedIndex == -1)

```



```

    {
        return;
    }
    string selectedSelectedItem = select.GetItemText(select.SelectedItem);
    StringBuilder builder = new StringBuilder();
    builder.Append("SELECT * FROM students WHERE ");
    if (selectedSelectedItem.Equals(Constants.FIRST_COURSE))
    {
        builder.Append("course='" + Constants.FIRST_COURSE);
    }
    if (selectedSelectedItem.Equals(Constants.SECOND_COURSE))
    {
        builder.Append("course='" + Constants.SECOND_COURSE);
    }
    if (selectedSelectedItem.Equals(Constants.THIRD_COURSE))
    {
        builder.Append("course='" + Constants.THIRD_COURSE);
    }
    if (selectedSelectedItem.Equals(Constants.FOURTH_COURSE))
    {
        builder.Append("course='" + Constants.FOURTH_COURSE);
    }
    if (selectedSelectedItem.Equals(Constants.FIFTH_COURSE))
    {
        builder.Append("course='" + Constants.FIFTH_COURSE);
    }
    if (selectedSelectedItem.Equals(Constants.SIXTH_COURSE))
    {
        builder.Append("course='" + Constants.SIXTH_COURSE);
    }
    if (selectedSelectedItem.Equals(Constants.FOAI))
    {
        builder.Append("faculty='" + Constants.FOAI);
    }
    if (selectedSelectedItem.Equals(Constants.FTM))
    {
        builder.Append("faculty='" + Constants.FTM);
    }
    if (selectedSelectedItem.Equals(Constants.FFBS))
    {
        builder.Append("faculty='" + Constants.FFBS);
    }
    if (selectedSelectedItem.Equals(Constants.FEMP))
    {
        builder.Append("faculty='" + Constants.FEMP);
    }
    if (selectedSelectedItem.Equals(Constants.FMTP))
    {
        builder.Append("faculty='" + Constants.FMTP);
    }
    builder.Append("");
    setDataIntoGridView(builder.ToString());
}

private void logout_Click(object sender, EventArgs e)
{
    Form1 form1 = new Form1();
    form1.Show();
    this.Hide();
}

private void update_Click(object sender, EventArgs e)
{
    string query = "SELECT * FROM students";
    sort.SelectedIndex = -1;
}

```

```

        select.SelectedIndex = -1;
        provideEmptySearchViewVisibility(false);
        rooms.DataSource = Program.dbHelper.executeSqlQuery(query);
    }

    private void add_Click(object sender, EventArgs e)
    {
        Form3 form = new Form3();
        form.Show();
    }

    private void delete_Click(object sender, EventArgs e)
    {
        if (rooms.CurrentCell.ColumnIndex != 0)
        {
            MessageBox.Show("Виділіть Id студента, якого бажаєте видалити");
            return;
        }
        foreach (DataGridViewCell cell in rooms.SelectedCells)
        {
            Program.dbHelper.executeSqlForDeleteStudent(cell.Value.ToString());
        }
        rooms.DataSource = Program.dbHelper.executeSqlQuery("SELECT * FROM students");
    }

    private void details_Click(object sender, EventArgs e)
    {
        if (rooms.CurrentCell.ColumnIndex != 1)
        {
            MessageBox.Show("Виділіть номер кімнати");
            return;
        }
        Form4 form = new Form4(rooms.CurrentCell.Value.ToString(), userType);
        form.Show();
    }
}

```


Add student

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
            initUi();
        }

        private void initUi()
        {
            role.DataSource = Constants.getListOfRoles();
            role.SelectedIndex = -1;
            faculty.DataSource = Constants.getListOfFaculties();
            faculty.SelectedIndex = -1;
            course.DataSource = Constants.getListOfCourses();
            course.SelectedIndex = -1;
        }

        private void add_Click(object sender, EventArgs e)
        {
            if (isFieldEmpty())
            {
                MessageBox.Show("Поля повинні бути заповнені");
            }
            else
            {
                Dictionary<string, string> data = new Dictionary<string, string>();
                data.Add(Constants.NAME, name.Text);
                data.Add(Constants.ROOM, room.Text);
                data.Add(Constants.FACULTY, faculty.GetItemText(faculty.SelectedItem));
                data.Add(Constants.COURSE, course.GetItemText(course.SelectedItem));
                data.Add(Constants.GROUP, group.Text);
                data.Add(Constants.ROLE, role.GetItemText(role.SelectedItem));
                Program.dbHelper.executeSqlForAddStudent(data);
                this.Hide();
            }
        }
    }
}
```

```

private bool isEmptyField()
{
    bool result = false;
    List<TextBox> boxes = new List<TextBox>();
    boxes.Add(room);
    boxes.Add(name);
    boxes.Add(group);
    List<ComboBox> comboBoxes = new List<ComboBox>();
    comboBoxes.Add(faculty);
    comboBoxes.Add(course);
    comboBoxes.Add(role);
    foreach (TextBox box in boxes)
    {
        if (box.Text.Equals(""))
        {
            result = true;
            return result;
        }
    }
    foreach (ComboBox box in comboBoxes)
    {
        if (box.SelectedIndex == -1)
        {
            result = true;
            return result;
        }
    }
    return result;
}

private void Form3_Load(object sender, EventArgs e)
{
}
}

```


Room

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form4 : Form , CommentCallback
    {
        string roomNumber = "";

        public Form4(string roomNumber, int userType)
        {
            InitializeComponent();
            this.roomNumber = roomNumber;
            init(roomNumber, userType);
        }

        public void onCommentEntered(string comment)
        {
            Program.dbHelper.executeSqlForUpdatingComment(comment, roomNumber);
            comments.DataSource = Program.dbHelper.executeSqlForCommentsByRoom(roomNumber);
        }

        private void init(string roomNumber, int userType)
        {
            room.Text = roomNumber;
            comments.DataSource = Program.dbHelper.executeSqlForCommentsByRoom(roomNumber);
            students.DataSource = Program.dbHelper.executeSqlForStudentsByRoom(roomNumber);
            if (userType == 1 || userType == 0)
            {
                add.Show();
                return;
            }
            else
            {
                add.Hide();
                return;
            }
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            if(comments.CurrentCell == null || comments.CurrentCell.ColumnIndex != 1)
            {
                MessageBox.Show("Виділіть строку коментаря");
                return;
            }
            string comment = Program.dbHelper.executeSqlForCommentStringByRoom(roomNumber);
            Form5 form = new Form5(comment, this);
            form.Show();
        }
    }
}

```

}

Додаток E

Comment

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form5 : Form
    {
        CommentCallback callback;
        string commentStr;

        public Form5(string comment, CommentCallback callback)
        {
            InitializeComponent();
            this.comment.Text = comment;
            commentStr = comment;
            this.callback = callback;
        }

        private void add_Click(object sender, EventArgs e)
        {
            callback.onCommentEntered(comment.Text);
            this.Hide();
        }
    }
}
```