Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “ Об’єктно -орієнтоване програмування”

Ріелтерська контора

Керівник , доцент Бондарєв В. М.

Студентка гр. ПЗПІ-19-2 Погребна Е.

Комісія:

Ст. викл.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Черепанова Ю.Ю.,

Проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Побіженко І. О.

Харків 2020

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Спеціальність: ***121***  ***Інженерія програмного забезпечення***

Освітня програма: ***Програмна інженерія***

Курс 1 .Група *ПЗПІ-19 -2* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студентки***

*Погребної Еліни*

1 Тема проекту: *Ріелтерська контора*

2 Термін здачі студентом закінченого проекту: ***“31” - травня - 2019 р***.

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*. *Основні функції програми.*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача,  висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

*Схема об’єктної моделі, алгоритми, приклади екранних форм* \_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми. | 01-03-2020 р. |
| 2 | Формулювання вимог до програми. | 06-03-2020 – 19-03-2020 р. |
| 3 | Розробка об’єктної моделі. | 01-04-2020 – 15-04-2020 р |
| 4 | Розробка підсистеми зберігання та пошуку спортсменів за критеріями. | 15-04-2020 – 22-04-2020 р. |
| 5 | Розробка функцій додавання та видалення із списку. | 22-04-2020 – 28-04-20120 р. |
| 6 | Розробка функцій зберігання та завантаження даних. | 28-04-2020 – 10-05-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 10-05-2020 – 16-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу. | 16-05-2020 – 31-05-2020 р. |
| 9 | Захист | 01-06-2020 – 07-06-2020 р. |

Студентка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ студентка Погребна Е.К.

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.

«1» березня 2020 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 37 с., 21 рис., 1 додаток, 6 джерел.

КУРСОВИЙ ПРОЕКТ, МОВА ПРОГРАМУВАННЯ C #, ОБ'ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, WINDOWS FORMS, РІЕЛТЕРСЬКА КОНТОРА, НЕРУХОМІСТЬ, АВТОРИЗАЦІЯ, ПОШУК.

Метою роботи є розробка програми “Ріелтерська контора”, використовуючи принципи об'єктно-орієнтованого програмування.

Інструменти, які були використанні для реалізації програми: Microsoft Visual Studio 2019, Windows Forms, платформа .NET Framework 4.7.2, мова програмування C#.

В результаті отримано програму під назвою “Ріелтерська контора”. У програмі реалізовані наступні функції: реєстрація нового покупця в системі, в саме: логін, пароль, так критерії пошуку квартир (фінансові можливості, бажаний стан та інше), авторизація покупця та ріелтора, перегляд існуючих квартир, додавання нових квартир, зміна її даних, видалення квартир з існуючих. Також, реалізована функція пошуку квартири за будь-якою з характеристик. Є можливість запису даних до файлів.

ЗМІСТ

Вступ 6

1 Специфікація програми 7

1.1 Інтерфейс користувача 8

1.2 Окремі функції користувача 10

2 Проектна специфікація. 14

2.1 Архітектура 14

2.2 Структура проекту 14

* 1. Об’єктна модель 14

2.4 Збереження та доступ до даних. Контроль версій. Тестування ....18

3 Інструкція користувача 19

3.1 Запуск програми 19

3.2 Робота програми 20

Висновки 27

Перелік джерел посилання.. 28

Додаток А Код програми.. 29

ВСТУП

Метою даної курсової роботи є освоєння мови програмування C# через написання програми. Реалізація цього проекту базується на заставах об’єктно-орієнтованого програмування, саме тому була використана середа розробки Visual Studio та додатковий інструмент Windows Forms. Завдяки цьому інструменту було створено графічний інтерфейс користувача.

Галуззю застосування можуть стати ріелторські контори, реалізована програма може допомогти з оптимізацією роботи закладу. Також проект можливо представити як допоміжний засіб для втілення більш масштабного проекту цієї тематики на основі навичок, які вже здобуті.

В цьому проекті використан графічний інтерфейс Windows Forms, який містить в собі набір бібліотек, класів та методів, необхідних для розробки графічних програмних додатків для платформ Microsoft Windows, його базові навички є початком для вивчення інших графічних інтерфейсів для розроблення інших проектів. Також техніка застосовування функції запису у бінарний файл та читання даних з бінарного файлу є одною з основоположних, тому це корисні знання щодо серіалізації.

Навички, які я здобула під час написання даної курсової роботи допоможуть мені в майбутньому в опануванні подальшого інструментарію платформи .NET та створенні програмних засобів у WEB, Desktop тощо.

1. **СПЕЦИФІКАЦІЯ ПРОГРАМИ**
   1. Інтерфейс користувача

Інтерфейс користувача складається з вікон. При запуску програми буде відображена головна форма, на якій знаходиться основний інтерфейс користувача (див.рис.1.1).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.1 – Головне меню.

Можливий перехід на форму для авторизації для ріелтора у системі. (див. рис. 1.2)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.2 – Вікно авторизації для ріелтора.

На головній формі є кнопка «Add new flat», при натисканні якої відкривається форма для додавання квартири. Також з допомогою цієї форми буде можливість редагувати існуючі квартири. (див. рис. 1.3)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.3– Вікно додавання квартири.

На головній формі є кнопка «Show customers», при натисканні якої відкривається форма зі списком зареєстрованих покупців.(див. рис. 1.4)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.4– Вікно відображення покупців.

У додатку передбачена авторизація та реєстрація для користувачів.(див. рис. 1.5 та 1.6)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.2 – Вікно авторизації для покупця.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.3 – Вікно реєстрації для покупця.

Після авторизації або реєстрації покупцю відкриється форма з можливими для нього варіантами вибору.(див. рис. 1.4)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1.1 – Головне меню покупця.

* 1. Опис окремих функцій користувача.

Програма надає користувачу наступні можливості

* + 1. Вхід в програму як ріелтор за логіном та паролем.

Функція «Вхід в програму».

Ріелтор входить в програму за допомогою імені та пароля. Вхід ріелтора виконується після вибору вікна «Ріелтор».

Основний сценарій:

1. Ріелтор вибирає вікно «Ріелтор» після чого відкривається вікно входу для ріелтора.
2. Ріелтор вводить ім'я і пароль і натискає кнопку "Enter".
3. Введені дані перевіряються, і відкривається адмінпанель - головне вікно ріелтора.
4. Додається функція перегляду покупців.

Додатковий сценарій:

Якщо дані не пройшли перевірку, на формі введення з'являється повідомлення про це і можна спробувати увійти знову. Число спроб не обмежена.

* + 1. Додавання квартир в базу.

Сценарій «Додавання квартири».

1. На адмінпанелі ріелтор натискає кнопку «Додати квартиру».
2. Відкривається форма для заповнення даних про даній квартирі.
3. Заповнює поля форми, вибирає файл зображень.
4. Натискає кнопку «Зберегти».
   * 1. Зміна даних про квартиру.

Сценарій «Зміна квартири».

1. На головній формі вибирає квартиру із загального переліку.
2. Натискає кнопку «Зміна даних».
3. Відкривається форма зміни даних. Поля форми заповнені.
4. Чи змінює поля форми, так само є можливість змінювати файл зображення.
5. Натискає кнопку «Ok».
   * 1. Видалення квартир.

Сценарій «Видалення квартири».

1. Ріелтор вибирає квартиру із загального переліку на головній формі.
2. Натискає кнопку «Видалити».
3. Запитується підтвердження видалення.
4. Після отримання підтвердження квартира видаляється, список оновлюється.
5. Вибраній стає квартира, попередня віддаленої.
   * 1. Перегляд списку потенційних покупців (зареєстрованих користувачів).

Функція «Перегляд потенційних покупців».

Список потенційних покупців відкривається після авторизації користувача як ріелтор. Він містить імена зареєстрованих покупців в програмі.

Основний сценарій:

1. Ріелтор авторизується.
2. Ріелтор вибирає кнопку «Перегляд покупців» на головній формі.
3. Відкривається список зареєстрованих покупців.
   * 1. Реєстрація.

Покупець реєструється.

Основний сценарій:

1. Користувач вибирає вікно «Реєстрація».
2. Відкривається форма для заповнення даних про покупця.
3. Користувач заповнює форму.
4. Натискає кнопку «Ok».
   * 1. Вхід в програму.

Функція «Вхід в програму».

Покупець входить в програму за допомогою імені та пароля.

Імена і паролі покупців зберігаються в даних програми.

Вхід покупця виконується після вибору вікна «Покупець».

Основний сценарій:

1. Покупець вибирає вікно «Покупець» після чого відкривається вікно входу для покупця.
2. Покупець вводить ім'я і пароль і натискає кнопку «Enter».
3. Введені дані перевіряються, і відкривається список квартир підходять під критерії покупця.

Додатковий сценарій:

1. Якщо дані не пройшли перевірку, на формі введення з'являється повідомлення про це і можна спробувати увійти знову. Число спроб не обмежена.
2. Якщо користувач не зареєстрований, він може вибрати вікно «Реєстрація».
   * 1. Заповнення критеріїв.

Основний сценарій:

При реєстрації покупець також заповнює поля критеріїв, задля добірки відповідних варіантів.

* + 1. Перегляд списку добірних.

Після входу або реєстрації покупця, йому відкривається список квартир, які підходять під ним виставлені критерії.

Основний сценарій:

Користувачеві доступні дані у вигляді списку. Відображення критеріїв і зображень в кожному елементі. Перегляд списку за допомогою скролінгу мишею.

1. **ПРОЕКТНА СПЕЦИФІКАЦІЯ**
   1. Архітектура проекту

Програма буде написана на мові С # з використанням середовища розробки Microsoft Visual Studio Community 2019 на платформі .NET Framework 4.7.2, тип програми: desktop, Windows Forms.

 Оскільки завдання до написання курсового проекту має на увазі об'єктно-орієнтований підхід автор використовує мову програмування C # для написання даної програми. Мова С # є одним з найбільш розвинених об'єктно-орієнтованих і тіпобезопасних мов програмування.

В процесі проектування розглядалися різні платформи, які мають найкраще взаємодію з мовою С #. Вибір платформи .NET Framework для написання даного курсового проекту заснований на зручному інтерфейсі і високий рівень розвитку даної платформи. Мінусом використання даної платформи є обмеження роботи програми (додаток буде працювати тільки на комп'ютерах з операційною системою Windows).

Для того, щоб реалізувати користувальницький графічний інтерфейс цього додатка буде використовуватися інтерфейс Windows Forms, який є однією з частин платформи .NET Framework.

* 1. Структура проекту

Оскільки програма для ріелтора є базовим варіантом з усім функціоналом, а для покупця доступна дуже обмежена функціональність на основі базової програми - ухвалено рішення створити єдину програму для обох користувачів. Програма буде тестуватися з використанням декількох тестових даних.

Модель даного проекту буде виглядати наступним чином (Див. рис. 2.1):

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 2.1 – Модель проекту

* 1. Об’єктна модель

2.3.1 Опис

База квартир включає в себе список всіх доступних квартир.

Ріелтор додає квартири, переглядає базу, вносить зміни, видаляє квартири з бази.

Покупець вводить критерії вибору, переглядає підібрані програмою квартири, залишає свої координати.

Про квартиру відомо: найменування, зображення, адреса, район, кількість кімнат, стан, вартість, координати продавця.

Про ріелторів відомо: логін (ім'я та прізвище), пароль.

Про покупця відомо: логін (ім'я та прізвище), пароль, критерії вибору (а саме фінансові можливості, бажану кількість кімнат, район і стан).

До простих типів не належать:

База квартир, квартира, користувач (ріелтор, покупець), база користувачів, список відповідних квартир.

З цих типів відокремлюємо ті, які не є колекціями і не містять колекцій:

Квартира, ріелтор, покупець.

Перші призначені для користувача типи:

Квартира - це найменування + зображення + адреса + район + кількість кімнат + стан + вартість + координати продавця.

Покупець - це логін + пароль + критерії вибору.

Ріелтор - це логін + пароль.

Решта типів:

База квартир - це колекція квартир.

Список користувачів - це колекція користувачів.

Список відповідних квартир - це колекція обраних програмою квартир.

Ріелтерська контора - це кілька колекцій: користувачів, квартир, відповідних квартир.

Зауваження: множина позначатиме колекцію.

* + 1. Методи

До головних методів належать:

1. Реєстрація нових користувачів програми (покупець, список користувачів, квартира).
2. Додавання квартири (ріелтор, база квартир, квартира).
3. Видалення квартири (ріелтор, база квартир, квартира).
4. Перегляд відповідних квартир (покупець, список відповідних квартир).
5. Перегляд потенційних покупців (ріелтор, список користувачів, покупець).

Зв’язки між класами представлені на рисунку 2.2

Изображение выглядит как дисплей, сидит, знак, черный

Автоматически созданное описание

Рисунок 2.2 – Модель класів

* 1. Збереження та доступ до даних. Контроль версій. Тестування.

У проекті використовується серілізація , завдяки якій дані переводяться з текстових в потік байтів з використанням вбудованої бібліотеки ВinaryFormatter. Захист даних в рамках даного проекту не розглядається.

Тестування при розробці здійснювалось за допомогою вбудованого у Visual Studio тестового проекту.

Контроль версій проекту здійснюється за допомогою системи контроля

версій GitHub.

**3 ІНСТРУКЦІЯ КОРИСТУВАЧА**

3.1 Запуск програми

Для використання даного курсового проекту, перш за все, слід завантажити папку з файлами проекту. Надалі слід виконувати такі дії:

Якщо користувач не має програми Visual Studio для детальної перевірки програми, йому слід виконувати дії 1-4.

1. Відкрити папку з проектом, в якій будуть присутні два файли для завантаження проекту через програму Visual Studio та папка з ідентичною назвою проекту.
2. Відкрити наступну папку, визначивши її вміст, знайти папку с назвою «bin».
3. Надалі відкрити папку під назвою «Debug», визначивши її вміст, слід запустити файл формату .exe, який має тип«Application». Цей файл обов’язково повинен називатися так само як і курсовий проект.

Програма запущена

Якщо користувач здійснює відкриття проекту через програму Visual Studio. Йому слід виконати наступні дії:

1. Відкрити програму Visual Studio, на головній панелі натиснути «Файл» та знайти папку з назвою проекту на системному диску.
2. У папці з проектом потрібно знайти файл з ідентичною назвою формату .sln та відкрити його.
3. Натиснути «F5» або «CTRL + F5» для того щоб запустити проект.

3.2 Робота з програмою

При запуску додатку відкривається головне вікно, в якому знаходиться списком позицій, наявних у базі, а також деякі функції. (див.рис.3.1)

1. Кнопка «Add new flat», після натискання якої відкривається форма для додавання нової квартири.(1)
2. Іконка «Інфо» для відкриття повідомлення з поясненнями для користувача.(2)
3. Панель з кнопками приховування, масштабування та закриття додатку.(3)

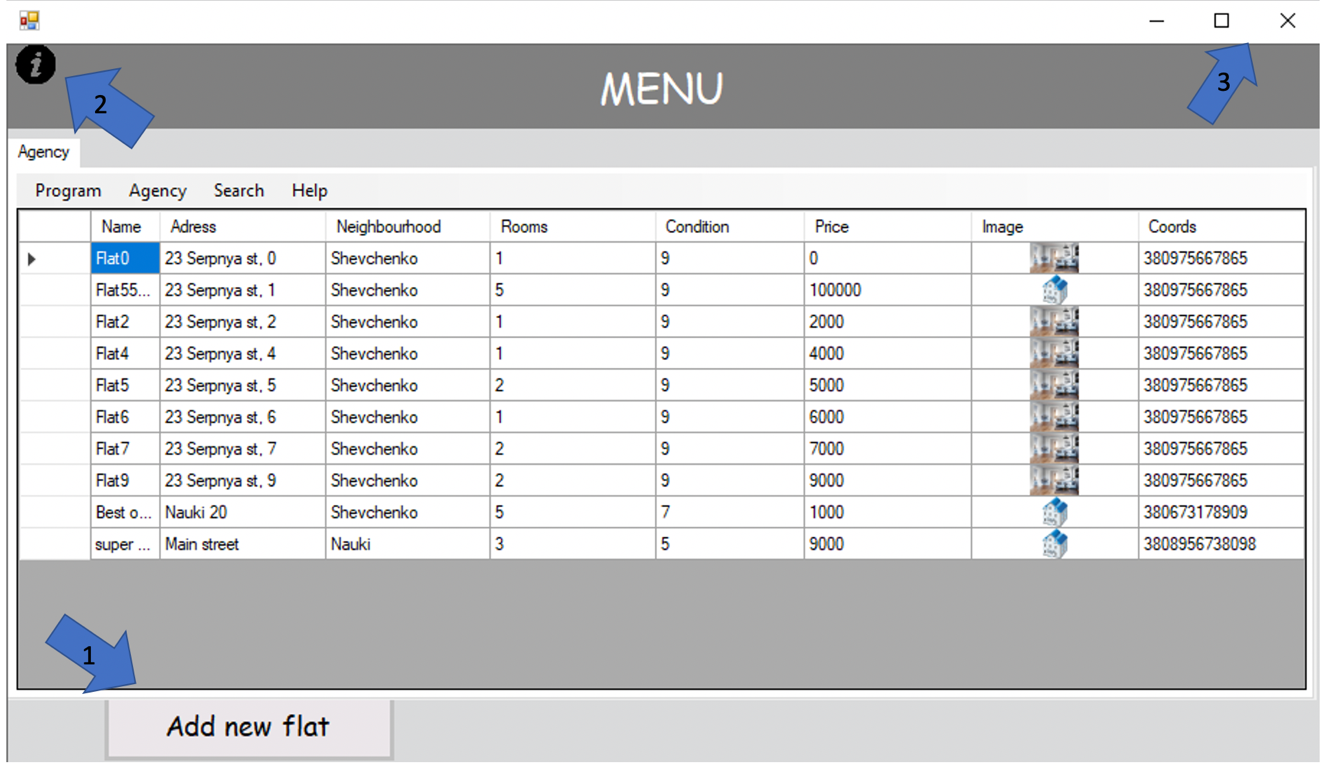


Рисунок 3.1 – Головна форма

1. Меню з кнопками для реалізації додаткових функцій.(див. рис. 3.2, 3.3 та рис. 3.4)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.2 – Меню додаткових функцій («Program»)

* 1. Кнопка «Load» для завантаження збережених даних.
  2. Кнопка «Save» для зберігання даних.
  3. Кнопка «Authorization» для відкриття форми авторизації.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.3 – Меню додаткових функцій («Agency»)

* 1. Кнопка «Edit flat» для відкриття форми редагування даних.
  2. Кнопка «Delete flat» для видалення квартири зі списку.
  3. Кнопка «Help» для відкриття посилання з інструкцією для користувача.
  4. Кнопка «Search» для пошуку квартир.

Для того, щоб знайти квартиру потрібно:

* + 1. Натиснути кнопку «Search».
    2. Вписати у поле дані.
    3. Вибрати за яким критерієм здійснюється пошук.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.4 – Меню додаткової функції («Search»)

Для додавання нової квартири у список необхідно натиснути кнопку «Add new flat». Після натискання відкривається форма з полями для заповнення. Необхідно вказати: ім’я ( не більше 40 символів), адресу, район, ціну, кількість кімнат, ціну, стан (від 1 до 10), координати ріелтора або продавця та завантажити фотографію. Після натискання кнопки «Complete» дані зберігаються, квартира з’являється у списку на головній формі. Після натискання кнопки «Back» дані не змінюються, форма закривається. Іконка «Інфо» відповідає за показ повідомлення з інструкцією для користувача при натисканні. (див. рис. 3.5)

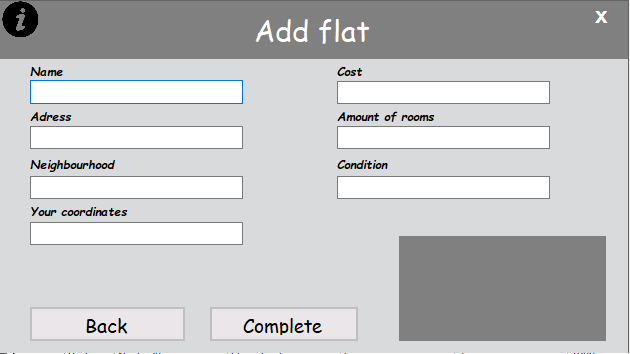


Рисунок 3.5 – Форма створення нової квартири

Для відкриття списку з покупцями необхідно авторизуватися як ріелтор, після чого на головній формі натиснути на кнопку «Show Customers», що з’явилась на формі(1). Після натискання відкривається форма з усіма зареєстрованими користувачами та їх паролями. Для повернення на головну форму потрібно натиснути кнопку «Cancel». (див. рис. 3.6 та 3.7)

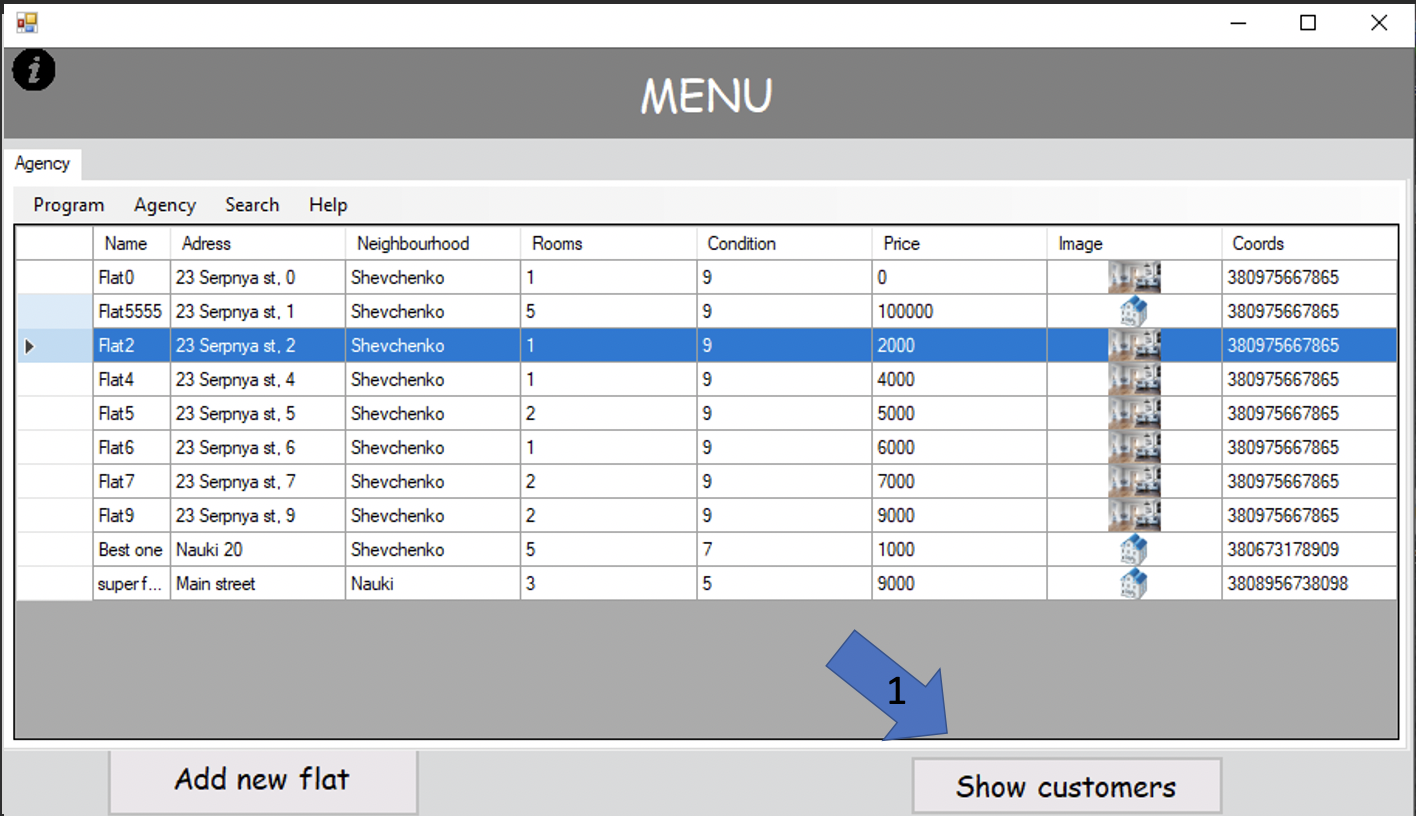
****

Рисунок 3.6 –Головна форма після авторизації.

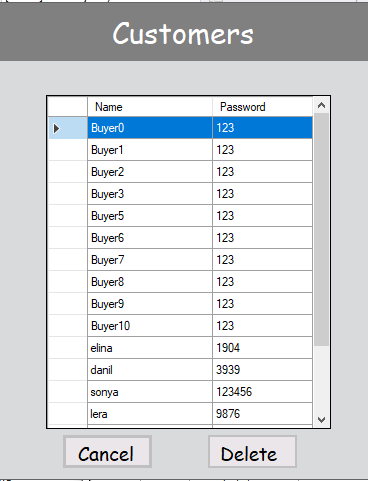


Рисунок 3.7 – Форма відображення всіх покупців.

Для того, щоб авторизуватись у програмі як ріелтор необхідно у меню натиснути «Program», після чого натиснути «Authorization». Відкриється форма авторизації. Ріелтору необхідно ввести свій логін та пароль. Для тестування додатку використовується логін «elina» та пароль «1904». У разі введення некоректних даних виводиться повідомлення. Кількість спроб необмежена. Після вдалого вводу даних відкриється головна форма з доступом до додаткової кнопки «Show Customers». Іконка «Інфо» відповідає за показ повідомлення з інструкцією для користувача при натисканні.(див. рис. 3.8)

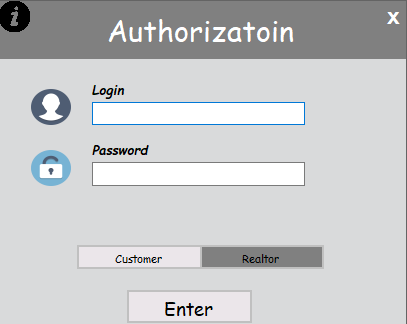


Рисунок 3.8 – Форма авторизації ріелтора.

Для того, щоб авторизуватись у програмі як покупець необхідно у меню натиснути «Program», після чого натиснути «Authorization». Відкриється форма авторизації. Далі треба натиснути кнопку «Customer»(1). Покупцю необхідно ввести свій логін та пароль, або натиснути на кнопку «Registration» (2), якщо він не є зареєстрованим у системі. У разі введення некоректних даних, поля підсвічуються червоним кольором. Після вдалого вводу даних відкриється форма з варіантами квартир для цього користувача. Іконка «Інфо» відповідає за показ повідомлення з інструкцією для користувача при натисканні. (див. рис. 3.9)

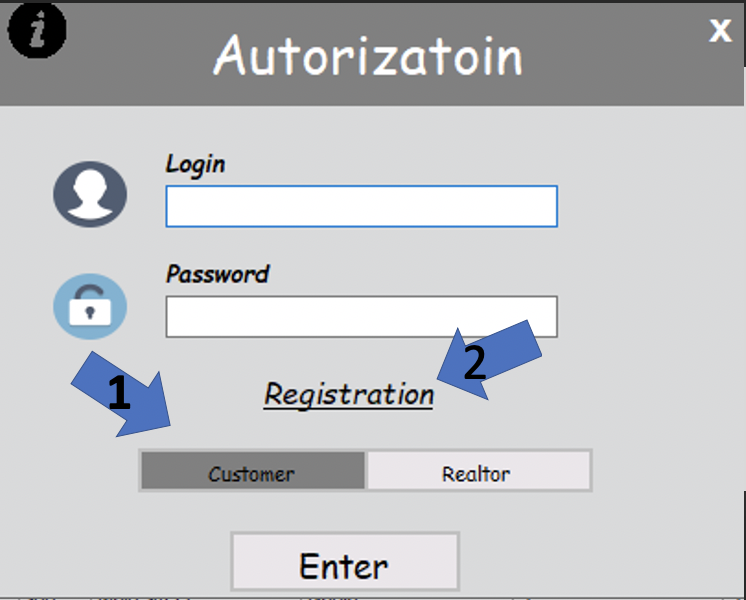


Рисунок 3.9 – Форма авторизації покупця.

Для реєстрації нового покупця потрібно натиснути кнопку «Registration» на формі авторизації покупця. Після натискання відкривається форма для заповнення. Необхідно ввести логін, а також критерії свого вибору, а саме бажану ціну, кількість кімнат, стан та район. Після натискання кнопки «Complete» дані зберігаються. Після натискання кнопки «Back» дані не змінюються, форма закривається. Іконка «Інфо» відповідає за показ повідомлення з інструкцією для користувача при натисканні. (див. рис. 3.10)

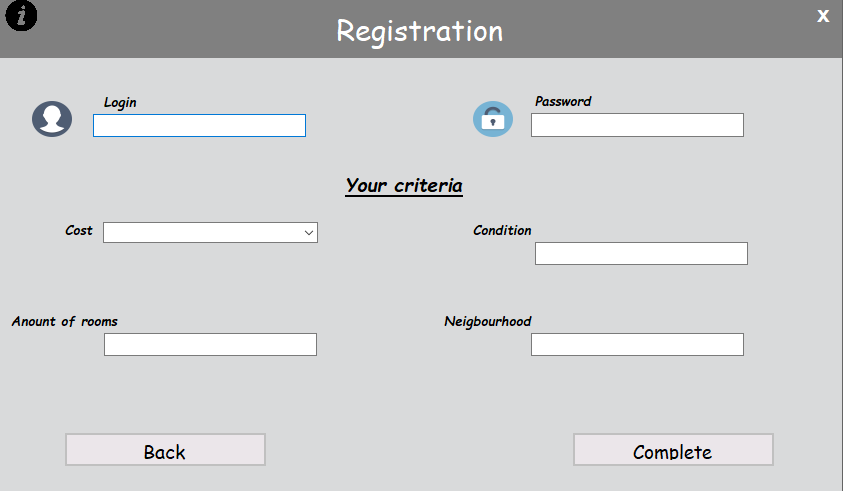


Рисунок 3.10 – Форма реєстрації.

Після реєстрації або авторизації покупцю відкривається форма з квартирами, які підходять цьому користувачу. (див. рис. 3.11)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.10 – Форма вибору покупця.

Для того, щоб отримати підказку користувачу необхідно натиснути на іконку «Інфо» на будь-якій формі. Після цього з’являється повідомлення, яке слід уважно прочитати.(див. рис. 3.12)

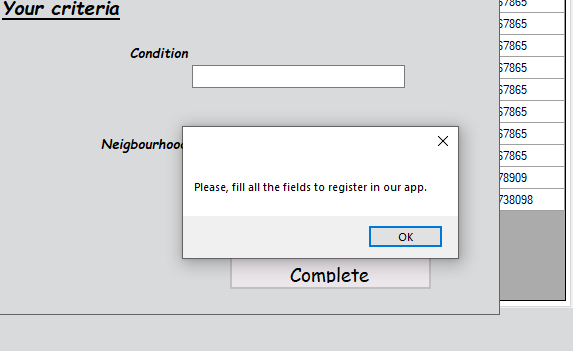


Рисунок 3.12 – Приклад вікна повідомлення з підказками для користувача.

**ВИСНОВКИ**

В ході виконання курсової роботи було розроблено програму «Ріелтерська контора », яка зберігає, оновлює та надає доступ до бази квартир у зручному для користувача форматі, дозволяє проводити пошук за інформацією будь-якого елементу бази (назва, ціна, стан і т.д.), фільтрувати, змінювати дані квартири, переглядати інформацію про них, можливість додавати, видаляти інформацію про квартири. Також дана програма дає можливість користувачам реєструватись та авторизуватись у системі.

В результаті написання курсового проекту був створений повноцінний програмний продукт. Булa розглянути специфiка мови С# та вивчені аспекти створення і розробки віконних додатків в середовищі .NET Framework. За період написання курсового проекту склалося комплексне поняття про процес створення програмного продукту: формулювання завдання, виділення чітких умов та критеріїв, візуалізація компонентів програми, розробка моделі класів та принципів взаємодії між ними, розробка інтерфейсу користувача.

Додаток було розроблено в повній відповідності до складеного плану. У ході роботи були закріплені знання з дисципліни, отримано досвід в розробці програмного продукту, вивчені стандарти оформлення звітної документації.

Одним з шляхів поліпшення запропонованої програми є розширення функціоналу. Також, шляхом поліпшення можна вважати під’єднання до бази даних (для можливості перегляду списку продуктів з будь-якого ПК) .

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Бондaрев В.М. Объектно-ориентированное программирование на С#: учебное пособие. Харьков: СМИТ, 2009г. 221 с.
2. Троелсен, Э. C# и платформа .NET. Библиотека программиста: учебное пособие. Питер: СПб, 2005. 796с.
3. Документaція по .NET. URL: https://www.udemy.com/ (дата звернення: 24.03.2019).
4. Документація по Windows Forms. URL: https://docs.microsoft.com/en-us/dotnet/framework/winforms/ (дата звернення: 24.03.2019).
5. Документація по C# від Microsoft. URL: https://docs.microsoft.com/en-us/dotnet/csharp/ (дата звернення: 25.05.2020).
6. Объектно-ориентированное проектирование: концепции и программный код: учебное пособие. Киев: Диалекта-Вильямс, 2019, 1089 с.

**ДОДАТОК А**

Код програми

using RealtAgency.Models;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Serialization.Formatters.Binary;

using System.Threading;

namespace RealtAgency.Data

{

public class Dao

{

// Клас для здійсненя збереження файлу та серілазації

Contora store;

const string filePath = "store.bin";

public Dao(Contora store)

{

this.store = store;

}

public void Save()

{

using (Stream stream = File.Create(filePath))

{

var serializer = new BinaryFormatter();

serializer.Serialize(stream, store);

}

}

public void Load()

{

using (Stream stream = File.OpenRead(filePath))

{

var serializer = new BinaryFormatter();

Contora st = (Contora)serializer.Deserialize(stream);

Copy(st.Flats, store.Flats);

Copy(st.Buyers, store.Buyers);

}

void Copy<T>(List<T> from, List<T> to)

{

to.Clear();

to.AddRange(from);

}

}

}

}

using RealtAgency.RealAgencyLibrary.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using RealtAgency.Models;

namespace RealtAgency.Models

{

//Покупець - це ім'я + пароль.

[Serializable]

public class Buyer : User

{

public List<Criteria> Criterias;

public Buyer(string name, string password)

{

Name = name;

Password = password;

Criterias = new List<Criteria>();

}

}

}

using RealtAgency.RealAgencyLibrary.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using RealtAgency.Models;

namespace RealtAgency.Models

{

//Покупець - це ім'я + пароль.

[Serializable]

public class Buyer : User

{

public List<Criteria> Criterias;

public Buyer(string name, string password)

{

Name = name;

Password = password;

Criterias = new List<Criteria>();

}

}

}

using RealtAgency.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace RealtAgency.RealAgencyLibrary.Models

{

[Serializable]

public class Criteria

{

public string Price { get; set; }

public string Rooms { set; get; }

public string Condition { set; get; }

public string Neighbourhood { set; get; }

public List<Buyer> Buyers;

public Criteria (string price, string rooms, string condition, string neighbourhood)

{

Price = price;

Rooms = rooms;

Condition = condition;

Neighbourhood = neighbourhood;

Buyers = new List<Buyer>();

}

public Criteria()

{

}

}

}

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using RealtAgency.Models;

namespace RealtAgency.Models

{

[Serializable]

// Квартира - це найменування + зображення + адреса + район + + кількість кімнат + вартість + координати продавця

public class Flat

{

public string Name { set; get; }

public string Adress { set; get; }

public string Neighbourhood { set; get; }

public string Rooms { set; get; }

public string Condition { set; get; }

public string Price { set; get; }

public Image Image { set; get; }

public string Coords { set; get; }

// public List <Potencial> Potencials { private set; get; }

public Flat( string name ="", string adress = "", string neighbourhood = "", string rooms = "", string condition = "", string price = "", string coords="", Image image = null)

{

Name = name;

Adress = adress;

Neighbourhood = neighbourhood;

Rooms = rooms;

Condition = condition;

Price = price;

Image = image;

Coords = coords;

//Potencials = potencials;

}

public Flat()

{

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace RealtAgency.Models

{

// Ріелтор - це ім'я + пароль.

[Serializable]

public class Realtor : User

{

public Realtor(string name, string password)

{

Name = name;

Password = password;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace RealtAgency.Models

{

[Serializable]

public abstract class User

{

// Абстрактний користувач від якого наслідується рвелтор та користувач: ім'я + пароль

public string Name { set; get; }

public string Password { set; get; }

}

}