Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “ Об’єктно -орієнтоване програмування”

ДОВІДНИК ТУРИСТА

Керівник , доцент Бондарєв В. М.

Студент гр. ПЗПІ-19-4 Пальчикова А. В.

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 -4* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

*Пальчикової Анастасії Владиславівни*

1 Тема проекту: *Довідник туриста*

2 Термін здачі студентом закінченого проекту: ***“31” - травня - 2020 р***.

3 Вихідні дані до проекту:

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

4 Зміст розрахунково-пояснювальної записки:

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

5 Перелік графічного матеріалу:

*Діаграма класів, схема об’єктної моделі, алгоритми, приклади екранних форм*\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 21-02-2020 р. |
| 2 | Формулювання вимог до програми | 01-03-2020 – 31-03-2020 р. |
| 3 | Розробка підсистеми зберігання та додавання агентств. | 02-04-2020 – 12-04-2020 р. |
| 4 | Розробка підсистеми зберігання та реєстрації клієнтів. | 12-04-2020 – 22-04-2020 р. |
| 5 | Розробка функцій редагування подорожей та агентств, зберігання та завантаження даних. | 24-04-2020 – 30-04-2020 р. |
| 6 | Розробка функцій сортування та кошику с подорожами. | 05-05-2020 – 11-05-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 10-05-2020 – 15-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 25-05-2020 – 05-06-2020 р. |
| 9 | Захист | 01-06-2020 – 19-06-2020 р. |

Студент Пальчикова А.В.

Керівник Бондарєв В. М.

« 21 »\_лютого\_\_\_\_\_\_\_\_\_\_ 2020 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 53 с., 25 рис., 4 додатки, 6 джерел.

МОВА ПРОГРАМУВАННЯ С#, WINDOWS FORMS, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПОДОРОЖІ, АГЕНТСТВА, КОШИК, ПОВЕРНЕННЯ ЗАМОВЛЕННЯ, ОПРАЦЮВАННЯ ЗАМОВЛЕННЯ, КЛІЄНТ, АДМІНІСТРАТОР.

Метою роботи є розробка програмного забезпечення “Довідник туриста”, базуючись на принципах об’єктно-орієнтованого програмування.

Інструменти, що використовувались для реалізації завдання: Microsoft Visual Studio 2019, Windows Forms, платформа .NET Framework 4.7.2, мова програмування C#.

В результаті створено програму, що має назву «VisitEasy» (букв. з англійської «відвідай легко»), яка є чудовим програмним забезпеченням для агентства, яка спеціалізуються на продажі турів у різні частини світу. Програмне забезпечення складається з двох незалежних програм: для користувача-клієнта та користувача-адміністратора. Цей додаток є особливо корисним в умовах карантину: клієнт може з легкістю обрати найкращий варіант подорожі для себе та використати її у зручний для нього час. Реалізовані наступні функції: реєстрація нового клієнту в системі, реєстрація адміністратора (за наявності секретного паролю «8965»), додавання, видалення, редагування агентств( подорожей), видалення клієнтів за погану поведінку, створення замовлення, видалення елементів з замовлення, додавання «Вподобанок» на подорожі, обробка замовлення, створення звіту про прибуток компанії. Реалізований запис даних до файлів.

ЗМІСТ

[ВСТУП 6](#_Toc532289188)

[1 СПЕЦИФІКАЦІЯ ПРОГРАМИ 8](#_Toc532289189)

[1.1 Бізнес-вимоги 8](#_Toc532289190)

[1.2 Вимоги до користувачів](#_Toc532289191) 8

[1.3 Функціональні вимоги](#_Toc532289193) 8

[1.4 Нефункціональні вимоги  9](#_Toc532289194)

[1.5 Опис інтерфейсу користувача](#_Toc532289194) 9

[1.6 Опис окремих функцій програми 18](#_Toc532289190)

[2 ПРОЕКТНА СПЕЦИФІКАЦІЯ 22](#_Toc532289194)

[2.1 Архітектура програми 2](#_Toc532289195)2

[2.2 Об’єктна структура програми 2](#_Toc532289194)2

[2.3 Об’єктна модель  програми 2](#_Toc532289194)4

[2.4 Діаграма класів 2](#_Toc532289196)6

[2.5 Збереження та доступ до даних. Контроль версій 2](#_Toc532289194)6

[3 ІНСТРУКЦІЯ КОРИСТУВАЧА 2](#_Toc532289196)8

[3.1 Запуск програми 28](#_Toc532289188)

[3.2 Робота програми для клієнта 29](#_Toc532289190)

[3.3 Робота програми для адміністратора 36](#_Toc532289192)

[ВИСНОВКИ 44](#_Toc532289196)

[ДОДАТОК А Клас VisitEasy](#_Toc532289196) 46

[ДОДАТОК Б Клас Order 49](#_Toc532289197)

ДОДАТОК В Клас Agency 50

ДОДАТОК Г Клас Portion 52

ВСТУП­

Даний курсовий проект розроблений з метою полегшення життя людей у таких скрутний для світу час. Реалізація проекту ґрунтується на принципах об’єктно-орієнтованого програмування, тому для роботи була обрана така мова програмування як С# та середа розробки Visual Studio. Наступною задачею була розробка графічного інтерфейсу користувача, що потребувало використання додаткового інструменту – Windows Forms.

Додаток ставить собі на мету зробити життя людей більш кольоровим. Постійні скасування рейсів, туга за пригодами, дефіцит вражень – усе це неабияк знижує дієздатність індивіда, руйнує гармонію емоційного стану, іноді – й жагу до життя та розвитку. Але вихід є. Програма допомагає придбати найбільш привабливий тур , обравши критерій, без осуду можна від нього відмовитися. Адміністратор вчасно обробить замовлення та допоможе клієнтам зробити перший крок назустріч новому життю. Додаток не лише здатен підібрати замовлення, існують спеціальні пропозиції та знижки. Клієнти залишатимуться задоволеними та посміхненими. У дизайні використана палітра таких кольорів:

* + - * + пастельно-жовтий асоціюється з радістю;
        + зелений зі свободою та різноманіттям;
        + помаранчевий з заходом сонця та спокоєм;
        + блідо-бежевий зі спокоєм та надійністю.

Тобто, ця програма є ідеальним рішенням в такий скрутний для туристичної сфери час. Кольорова гамма налаштовує клієнта на пригодницький або відпочиваючий настрій, а надійність наявності та актуальності сертифікатів на подорожі робить пересічного клієнта постійним.

Задачі роботи:

* Розробка системи зчитування та запису даних;
* Створення логіки роботи системи;
* Робота з користувачем;
* Розробка логіки додавання, пошуку та видалення інформації;
* Тестування готового продукту;
* Створення стійкості програми.

**1 СПЕЦИФІКАЦІЯ ПРОГРАМИ**

* 1. Бізнес-вимоги

Серед бізнес-вимог слід зазначити наступні:

* Оформлення замовлення, можливість відмови до його оформлення;
* Оформлення звіту про прибуток для власника;
* Відображення наявних агентств та подорожей.
  1. Вимоги до користувачів

Серед вимог користувачів слід зазначити наступні:

* Перегляд списку агентств та подорожей;
* Пошук агентства за назвою, сортування за рейтингом «Вподобанок»
* Сортування подорожі за місцем, коштовністю, кількістю наявних подорожей;
* Формування замовлення та видалення непотрібних подорожей;
* Додавання «Вподобанок» на подорожі, які сподобались;
* Авторизація користувача;
* Редагування агентств та подорожей, слідкування за зменшенням кількості подорожей;
* Блокування клієнтів за нетолерантну поведінку;
* Обробка замовлення;
* Перегляд звіту про прибуток (у файлі txt).
  1. Функціональні вимоги

Серед функціональних вимог слід зазначити наступні:

* Система має зчитувати з файлу дані про користувачів, агентства, подорожі та замовлення;
* Система має записувати зміни до файлу за необхідності;
* Система має звіряти введений пароль, ім’я та електронну адресу(у клієнта) та перевіряти їх на наявність;
* Система має шифрувати та зберігати змінений пароль у файл;
* Система має зберігати дані про агентство «VisitEasy» та записувати(зберігати) данні про прибуток;
  1. Нефункціональні вимоги

Серед нефункціональних вимог слід зазначити наступні:

* Легкість та простота використання;
* Легкість переміщення;
* Цілісність;
* Ефективність та стійкість до збоїв;
* Взаємодія системи і зовнішнього світу;
* Обмеження дизайну і реалізації.

Серед обмежень дизайну і реалізації слід зазначити наступні:

* + Використання платформи .Net Framework 4.7.2;
  + Використання мови програмування C#;
  + Використання Windows Forms.

1.5 Опис інтерфейсу користувача

Для клієнтської версії додатку інтерфейс користувача складається з 5 вікон. При запуску програми з’являється перше - вікно входу. Можливий перехід на форму для реєстрації у системі. (Див. рис. 1.1 та 1.2)

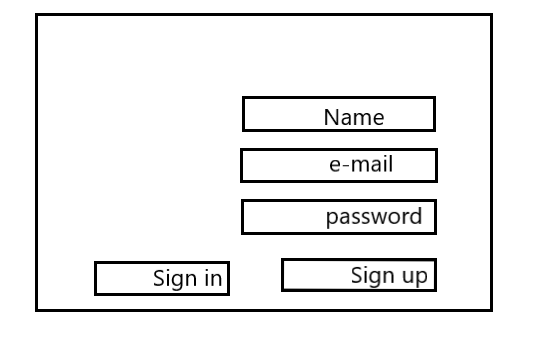


Рисунок 1.1 – Вікно входу

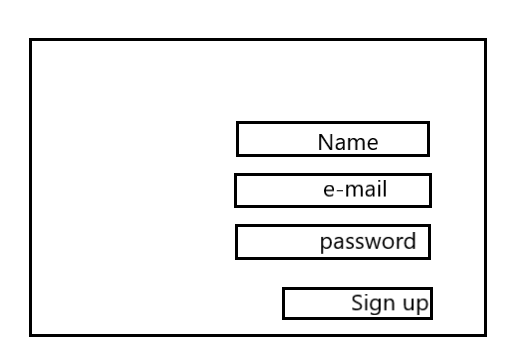


Рисунок 1.2 – Вікно реєстрації

Наступним вікном інтерфейсу користувача є основне вікно програми.

(Див. рис. 1.3)

На ньому з меню є доступними такі сторінки:

- Список агентств (див. рис. 1.4), з нього можна перейти на вікно з агентством (див. рис. 1.5), а з нього – на вікно с подорожжю (див. рис. 1.7)

-Список подорожей (див. рис. 1.6) , з нього можна перейти на вікно з подорожжю (див. рис. 1.7)

-Список замовлень (кошик) (див. рис. 1.8),

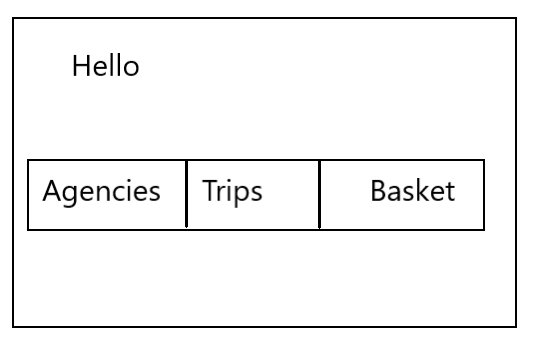


Рисунок 1.3 – Основне вікно

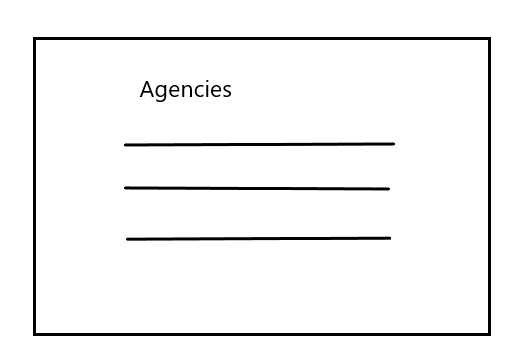


Рисунок 1.4 – Список агентств

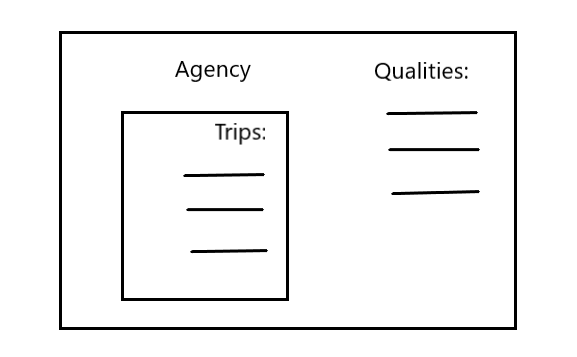


Рисунок 1.5 – Вікно агентства

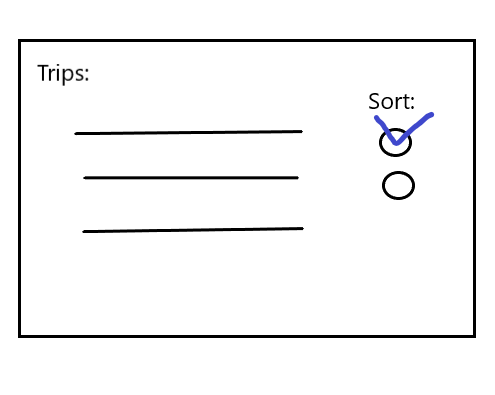


Рисунок 1.6 – Список подорожей

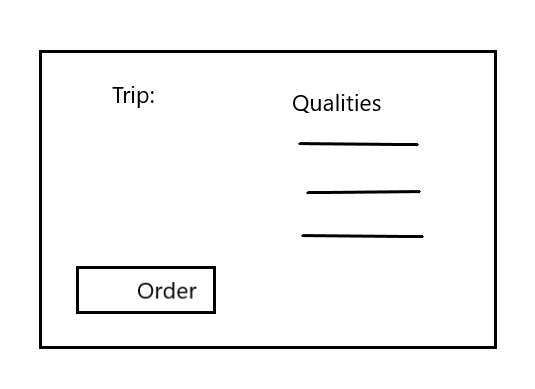


Рисунок 1.7 – Вікно подорожі

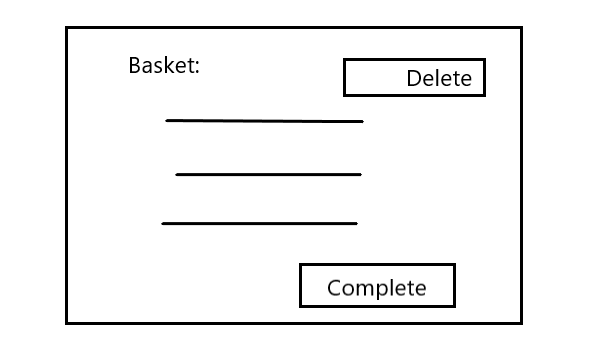


Рисунок 1.8 – Вікно кошику

Для адміністраторської версії додатку інтерфейс складається з шести вікон. Вхід в програму аналогічний(відсутній е-mail) та при реєстрації вводиться спеціальний пароль адміністратора. Наступним вікном є головне вікно (Див. рис. 1.9). З меню доступні сторінка агентств (Див. рис. 1.10), на ній можна відредагувати та додати агентство за допомогою спеціальної форми (Див. рис. 1.11) , на ній можна зробити аналогічні дії с подорожжю на формі редагування подорожі (Див. рис. 1.12), або ж видалити, сторінка подорожей майбутнього (Див. рис. 1.13), подорожей зі знижкою (Див. рис. 1.14), сторінка з клієнтами та їх замовленнями (Див. рис. 1.15). Замовлення можна переглянути на формі перегляду замовлення(Див. рис. 1.16) (При запуску програми відкривається перше – основне вікно програми.

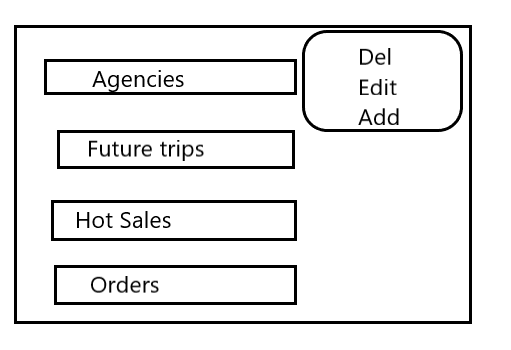


Рисунок 1.9 – Головна панель адміністратора



Рисунок 1.10 – Сторінка агентств

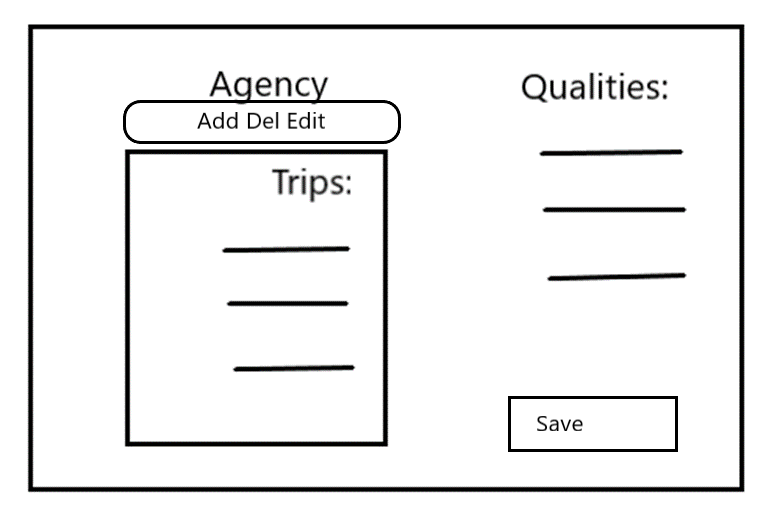


Рисунок 1.11 – Вікно редагування агентств

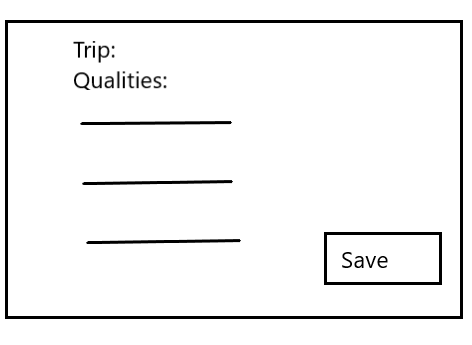


Рисунок 1.12 – Вікно редагування подорожі

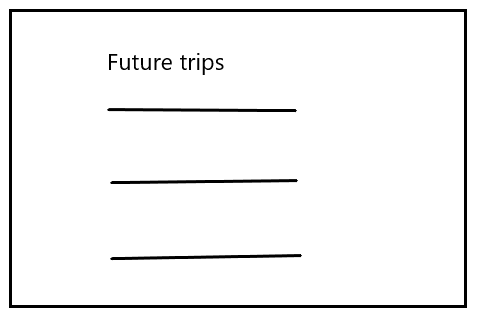


Рисунок 1.13 – Вікно подорожей майбутнього

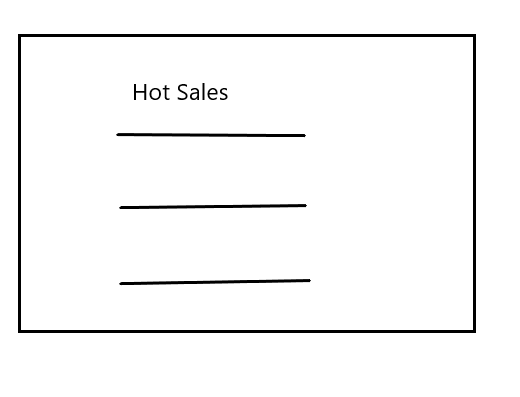


Рисунок 1.14 – Вікно подорожей зі знижкою

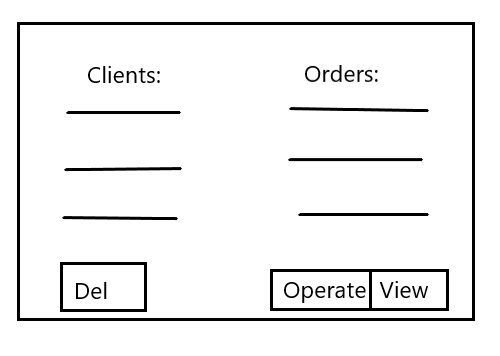


Рисунок 1.15 – Сторінка адміністратора з клієнтами та замовленнями

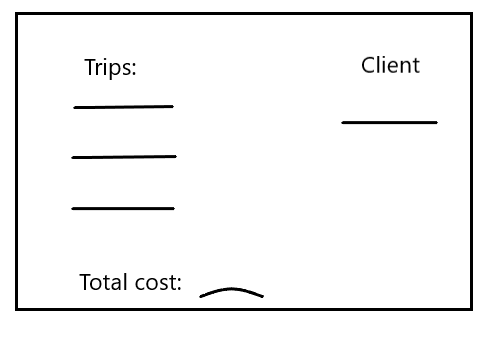


Рисунок 1.16 – Вікно перегляду окремого замовлення

* 1. Опис окремих функцій

Користувачу доступні такі функції:

* Вхід до системи за логіном та паролем

Клієнт входить в програму за допомогою логіна і пароля.

Імена і паролі клієнтів зберігаються в даних програми.

При правильному введенні пароля клієнт потрапляє в меню клієнта.

Основний сценарій

1. Клієнт потрапляє на форму з авторизацією (рис. 1).

2. Клієнт вводить логін, пошту та пароль і натискає кнопку “Sign in”.

3. Якщо дані введені вірно - відбувається перехід в меню клієнта.

Додатковий сценарій:

1. Якщо дані введені невірно - на формі введення з'являється повідомлення про помилку і можливий повторний вхід. Число спроб не обмежене.

2. Якщо користувач не зареєстрований, він може вибрати вікно «Sign up».

-Реєстрація у системі за логіном, паролем та поштою

Клієнт реєструється і вводить свої дані.

Основний сценарій:

1. Користувач вибирає вікно «Sign up».

2. Відкривається форма для заповнення даних про клієнта.

3. Користувач заповнює форму.

4. Натискає кнопку «Sign up».

-Перегляд наявних у VisitEasy агентств. Про турагентства відомо: назва, логотип, кількість видів подорожей, кількість «Вподобанок» ;про подорож відомо: локація, ціна, додаткові послуги, умови проживання та проїзду, сервіс приймаючої сторони;

-Купівля подорожі. Для цього необхідно подорож, її кількість, якщо все

влаштовує – надіслати на обробку адміністратору.

Сценарій:

1. Клієнт заходить на подорож.
2. Клієнт обирає кількість та натискає «Order».

3. Якщо замовлення влаштовує - то на сторінці кошика натискається «Complete», клієнт чекає, доки адміністратор виконає замовлення.

4. Якщо якась подорож не влаштовую – її можна видалити за допомогою кнопки «Delete».

5. Адміністратор обробляє замовлення.

-Перегляд прибутку VisitEasy. Для цього необхідно перейти до файлу «CompletedOrders» за адресою: AdminApp\bin\Debug\CompletedOrders.txt

У процесі роботи користувач змінює дані, які знаходяться в оперативній пам'яті.

Якщо дані змінені, але ще не введено постійної пам'яті, то перед закриттям програми користувач отримує пропозицію зберегти дані.

Адміністратору доступні такі функції:

-Перегляд агентств та подорожей. Можливість їх редагування, додавання, видалення.

1. На адмін. панелі вибирає агентство.

2. Натискає кнопку edit.

3. Відкривається форма зміни агентств. Поля форми заповнені.

(Можна відредагувати подорожі агентства аналогічним чином).

4. Змінює поля форми.

5. Натискає кнопку «Check».

6.Валідує данні.

7.Якщо валідація пройшла успішно – з’являється кнопка «Finish».

8.При натисканні цієї кнопки додавання або редагування агентства (подорожі) завершено.

- Додавання товарів до бази

1. Відкривається аналогічна форма.

2. За виконання умов – додається агентство(подорож).

-Видалення агентства (подорожі)

1. Адміністратор вибирає агентство із на адмін. панелі ( якщо видалення подорожі - то аналогічно на формі редагування агентства) на адмін. панелі.

2. Натискає кнопку Del.

3. Відкривається панель із запитом на видалення.

4. Після отримання підтвердження агентство (подорож) видаляється, список оновлюється.

-Перегляд користувачів, можливість блокування(видалення), схвалення або перегляд замовлення.

Якщо дані змінені, але ще не введено постійної пам'яті, то перед закриттям програми користувач отримує пропозицію зберегти дані.

1. **ПРОЕКТНА СПЕЦИФІКАЦІЯ**

2.1 Архітектура проекту

Програма буде створюватися на мові С # з використанням середовища розробки Visual Studio 2019 на платформі .NЕТ Framework 4.7.2 з використанням інтерфейсу програмування додатків WindowsForms.

Вибір мови обумовлений об'єктно-орієнтованим підходом до написання програми. На даний момент мова С # є одним з найпопулярніших, зручних і повноцінних серед об'єктно-орієнтованих мов програмування.

В процесі проектування розглядалися 2 платформи, з якими мова С # має найкраще взаємодію, а саме .NЕТ Core і .NЕТ Framework. Наш вибір припав на останню, через збільшеної кількості керуючих елементів і більш високого рівня розвитку платформи. На жаль, це означає що додаток буде працювати виключно на машинах з операційною системою Windows.

Для реалізації призначеного для користувача інтерфейсу було вирішено використовувати графічний інтерфейс так як він простий для сприйняття і інтуїтивно зрозумілий. Для реалізації інтерфейсу ми будемо використовувати інтерфейс WindowsForms який є частиною платформи .NЕТ Framework. З його допомогою у нас буде можливість створити повноцінний інтерфейс для користувачів програми.

2.2 Oб’єктна структура проекту:

Буде розумно поділити проект, описаний в специфікації, на два окремих додатки: для адміністратора і для клієнтів турагентства, оскільки функції цих типів користувачів між собою не перетинаються.

Крім того, для тестування програми на етапі програмування доречно створити кілька тестових даних для перевірки правильності роботи програми.

На цій підставі, можемо скласти наступну структуру проекту: (Див. рис. 2.1)

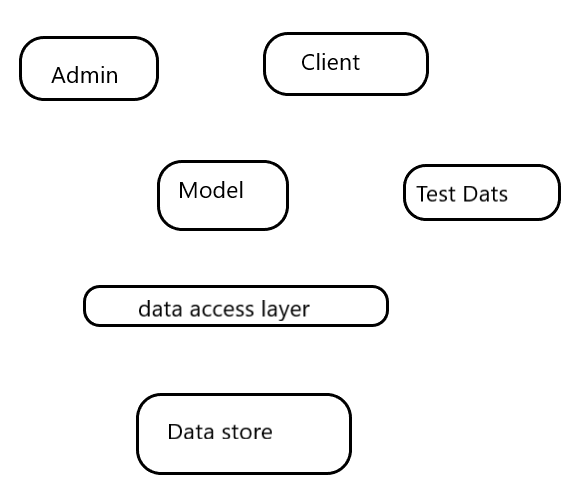


Рисунок 2.1 – Модель проекту

2.3 Об’єктна модель

2.3.1 Опис

Клієнт обирає подорож(подорожі). Після чого, наповнює кошик і робить замовлення.

Адміністратор додає турагентства, редагує їх, також вносить зміни до подорожей турагентств, відзначає продані подорожі і отримує документ-звіт продажів.

Про турагентства відомо: назва, логотип, коротка інформація (опціонально), кількість путівок, кількість «Вподобанок».

Про подорож відомо: локація, ціна, дод. послуги, проживання, сервіс приймаючої сторони (ім’я та фото).

Про клієнта відомо - ім'я, пароль, поштову адресу.

Про адміністратора відомо - ім'я, пароль.

Класи

Додаток VisitEasy, турагентство, подорож, клієнт, порція, замовлення, адміністратор, абстрактний користувач.

Типи, які не містять колекцій:

(Подорож, клієнт , адміністратор)

Подорож – локація , ціна, дод. послуги, умови проживання, сервіс приймаючої сторони (ім’я та фото), фото подорожі ;

Клієнт - ім'я, пароль, адреса.

Адміністратор - ім'я, пароль.

Колекції:

Замовлення – список порцій , покупець, дата замовлення, чи було надіслане покупцем, чи було перевірене адміністратором.

Порція - це подорожі, їх кількість, чи є ці подорожі акційними, майбутнього або звичайними (Для комфортної демонстрації користувачу дублюється також локація подорожі, ціна, агентство).

Агентство - це путівки, їх кількість, назва, опис, кількість «Вподобанок», зображення.

VisitEasy - це кілька колекцій: турагентств, клієнтів, адміністраторів, замовлень.

* + 1. Методи

Основні методи:

* Реєстрація нових клієнтів програми (клієнт, список користувачів).
* Реєстрація нових адміністраторів програми (адміністратор, список адміністраторів).
* Відправка запиту на замовлення (клієнт, замовлення).
* Додавання «Вподобанок» на подорожі (клієнт, подорож)
* Схвалення заявки (адміністратор, замовлення).
* Перегляд списку агентств , в них подорожей, які можна купити (клієнт, список доступних агентств(подорожей).
* Додавання нового агентства(подорожі) (адміністратор, список агентств(подорожей)).
* Зміна агентства(подорожі) (адміністратор, список агентств(подорожей)).
* Видалення агентства(подорожі) (адміністратор, список агентств(подорожей)).

Зв’язки між класами представлені на рисунку 2.2

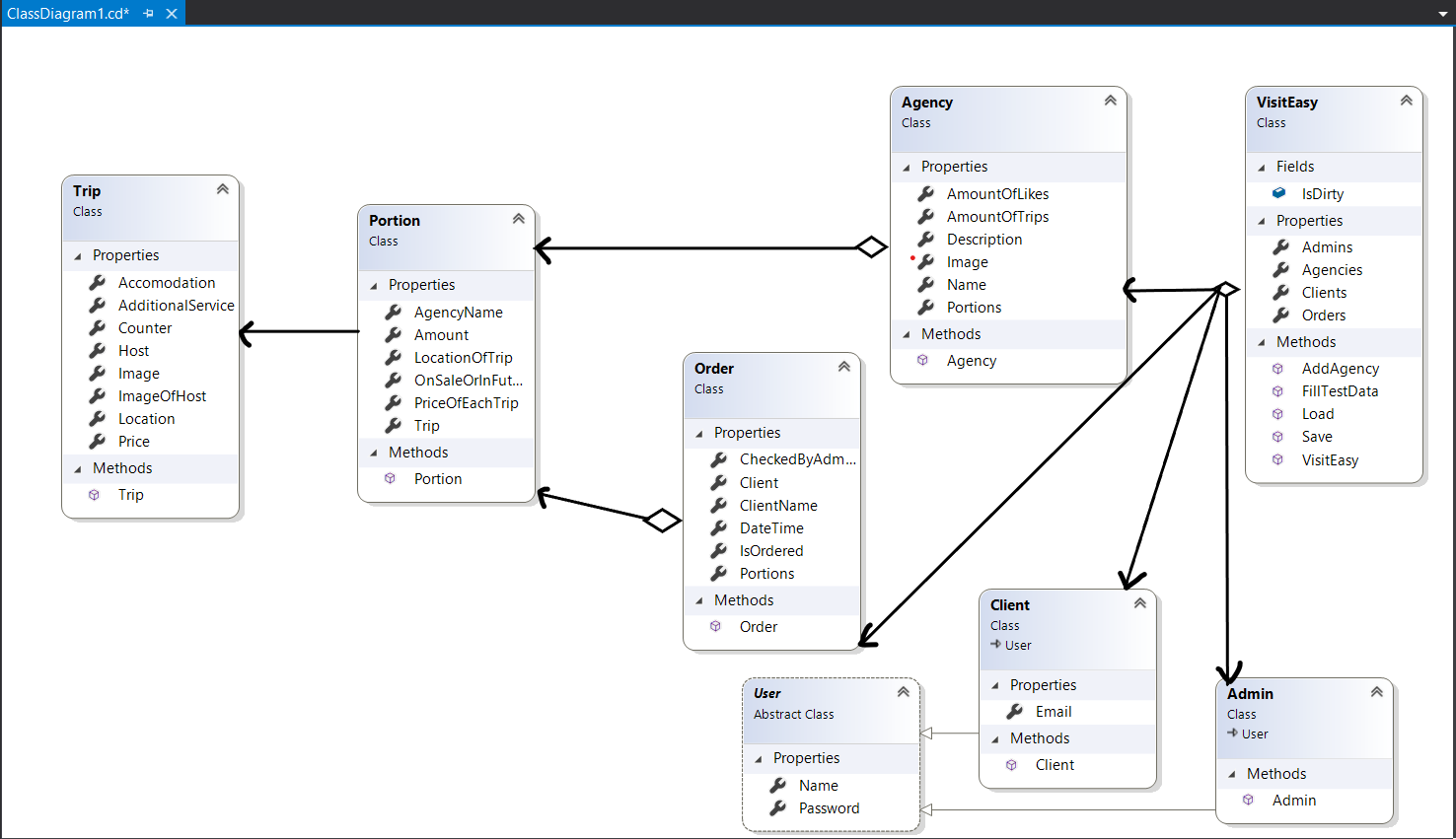


Рисунок 2.2 – Діаграма класів

З метою спрощення коду та запобігання його дублювання був використаний принцип поліморфізму у створенні класу «Клієнт» та «Адміністратор», що були успадковані від спільного класу «Користувач».

* 1. Збереження та доступ до даних. Контроль версій

Дані переводяться з текстових в потік байтів за допомогою серіалізації з використанням вбудованої бібліотеки ВinaryFormatter. Захист даних в рамках даного проекту не розглядається.

Контроль версій проекту здійснюється за допомогою системи контроля версій GitHub.

**3 ІНСТРУКЦІЯ КОРИСТУВАЧА**

3.1 Запуск програми

Для запуску даної програми необхідна програма Visual Studio. Після інсталяції цієї програми на свій ПК слід виконати такі дії:

1. Завантажити директорію з файлами курсового проекту(Всі папки, разом з папкою «Images»)
2. У директорії проекту запустити файл формату .sln з назвою, що співпадає з назвою самої директорії.
3. Натиснувши кнопку «Start» запустити проект у режимі «Release» попередньо вибравши «AdminApp» або «ClientApp» для додатку адміністратора або клієнта відповідно/
4. Програма запущена.
   1. Робота програми для клієнта:
      1. Вхід у програму

При запуску програми з’ являється вікно із входом до системи. Клієнту необхідно ввести свій логін , пошту та пароль, або натиснути на кнопку «Sign up», якщо він не є зареєстрованим у системі. У разі введення некоректних даних, по виводиться повідомлення про вид помилки. (Див. рис. 3.1 та 3.2)

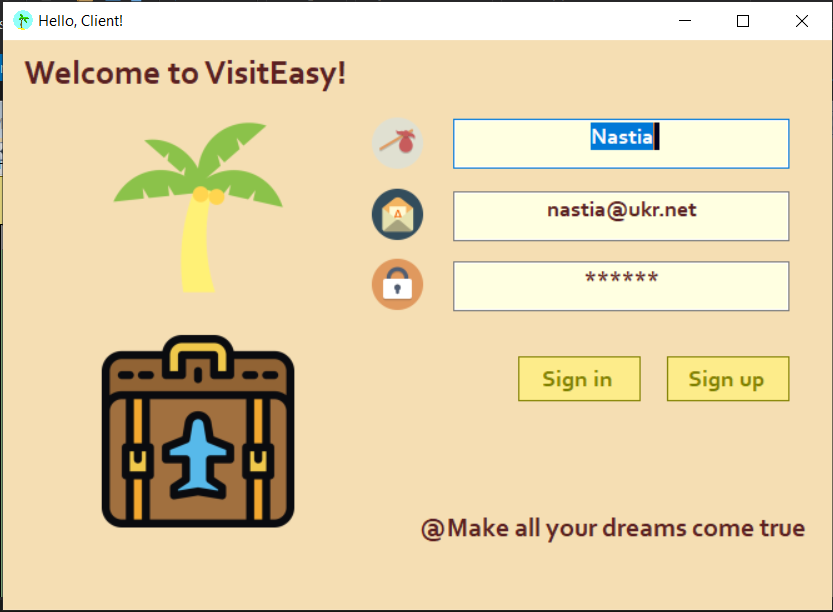


Рисунок 3.1 – Вікно входу

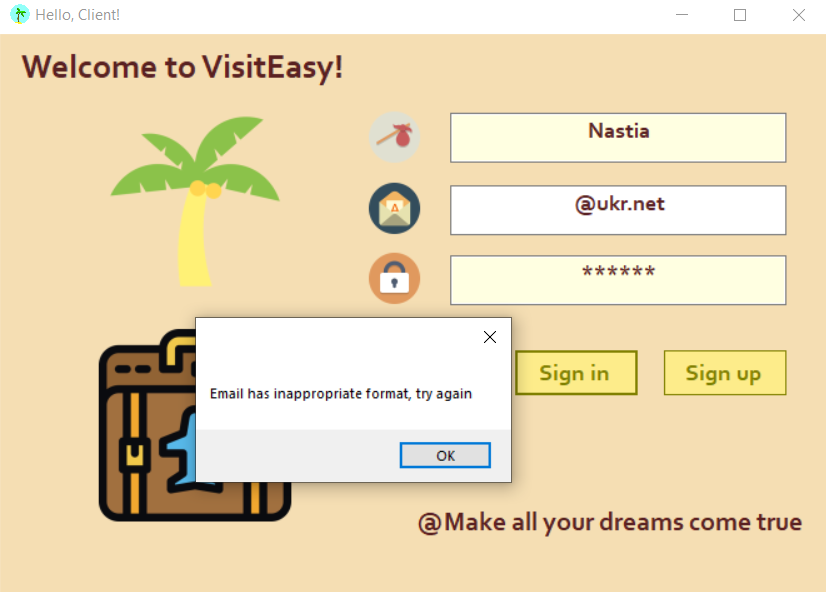


Рисунок 3.2 – Некоректні дані при вході

У разі реєстрації, відкривається форма для заповнення. Необхідно ввести логін, пароль пошту. Вимоги вказані при натисканні кнопки допомоги. У разі недотримання вимог з’являється діалогове вікно із повідомленням про помилку та поле підсвічується зеленим кольором. (Див. рис. 3.3)

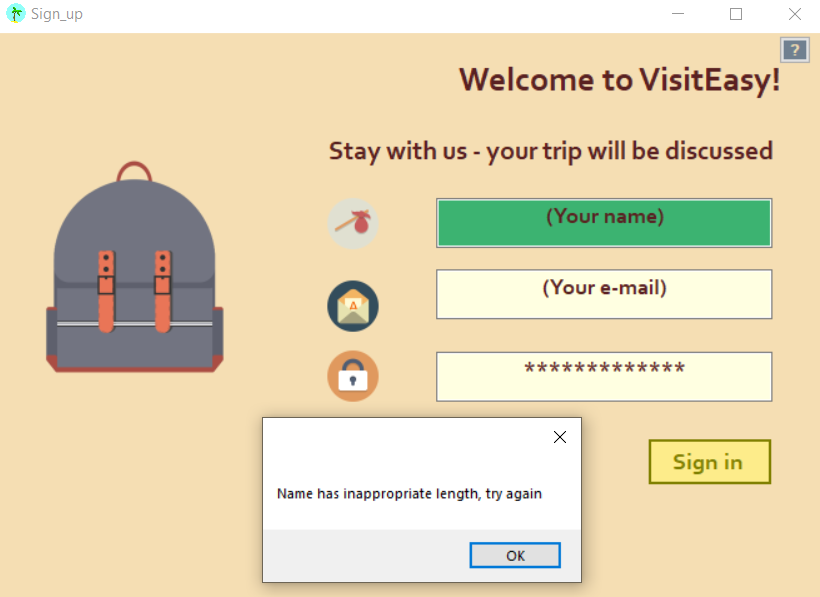


Рисунок 3.3 – Вікно реєстрації

Після входу/реєстрації відкривається вікно з короткою інформацією(Див. рис. 3.4). Можна перейти на сторінку агентств(Див. рис. 3.5). Можливий пошук за назвою, сортування за рейтингом кількості «Вподобанок» та повернення до дефолтного вигляду. При натисканні відкривається форма с інформацією про нього та подорожами(Див. рис. 3.6).

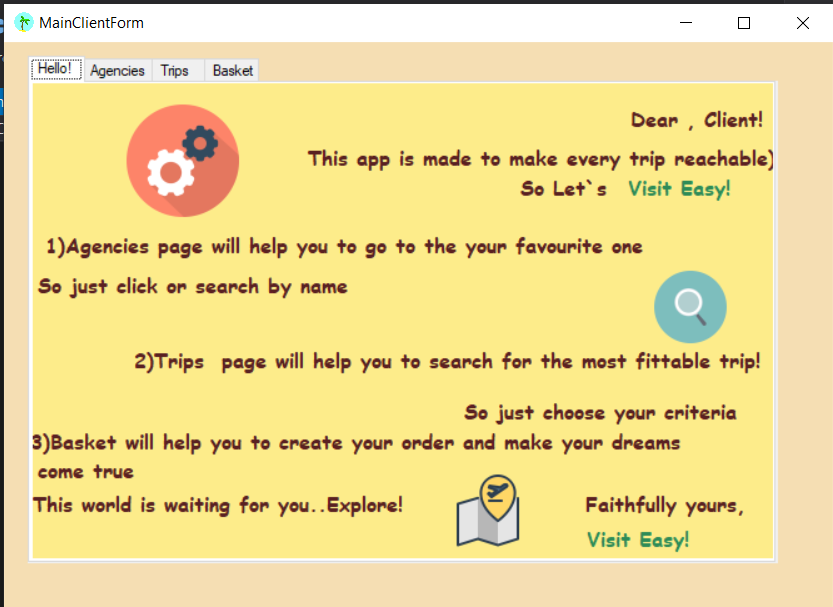


Рисунок 3.4 – Головна сторінка

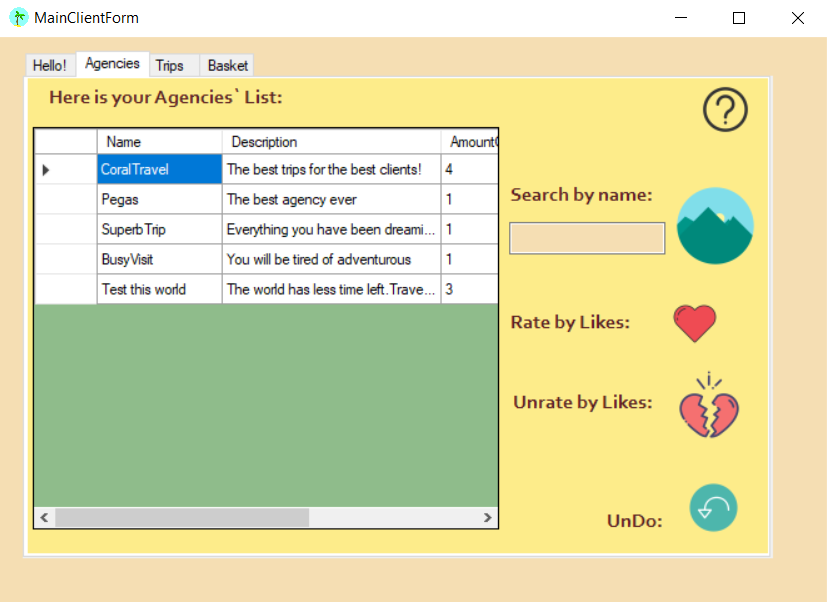


Рисунок 3.5 – Сторінка агентств

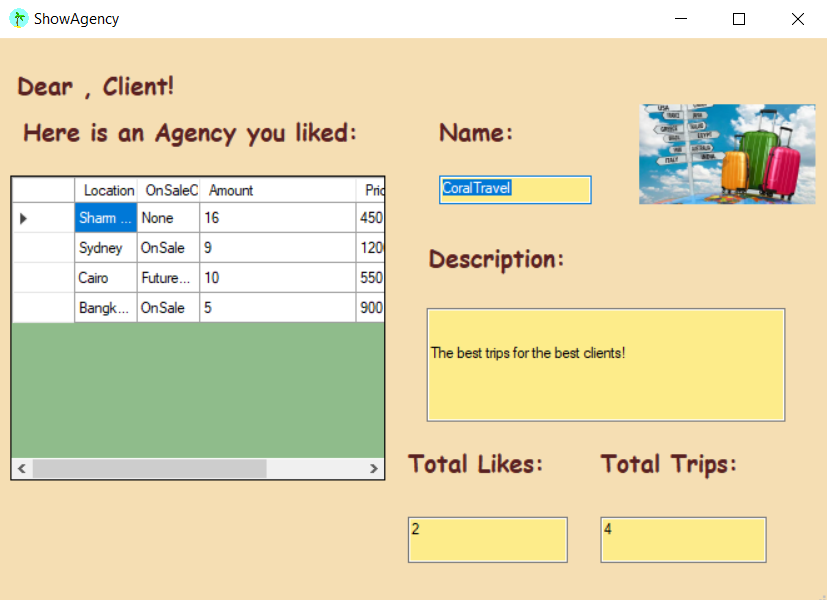


Рисунок 3.6 – Вікно обраного агентства

При натисканні на подорож виводиться вікно з інформацією про подорож(Див. рис. 3.7). На ньому можна обрати та замовити кількість подорожей. Поставити «Вподобанку» натисканням на сердечко.

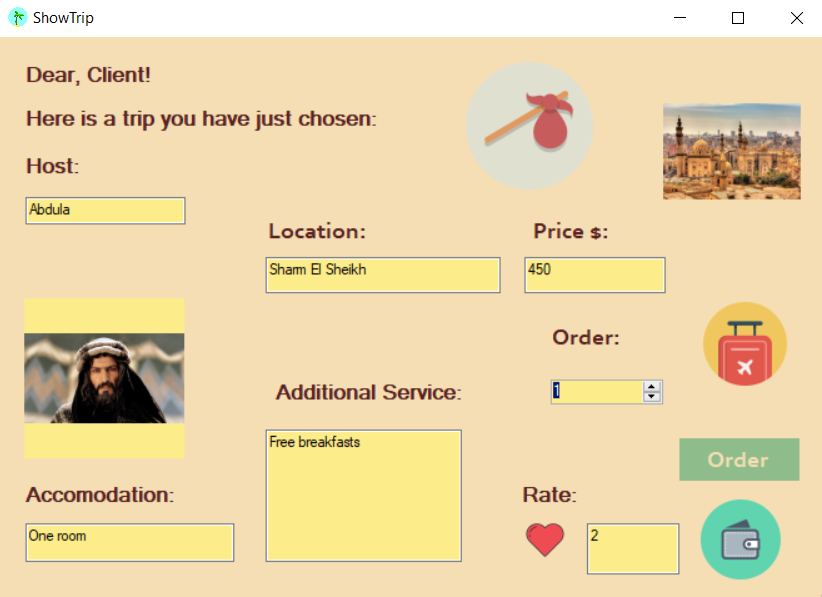


Рисунок 3.7 – Вікно обраної подорожі

При переході на сторінку з подорожами (Див. рис. 3.8) можна відсортувати за локацією. Натиснувши на гаманець або купу грошей, за ціною. Також можна відсортувати за кількість наявних місць. При натискання на подорож відкривається вікно (Див. рис. 3.7).

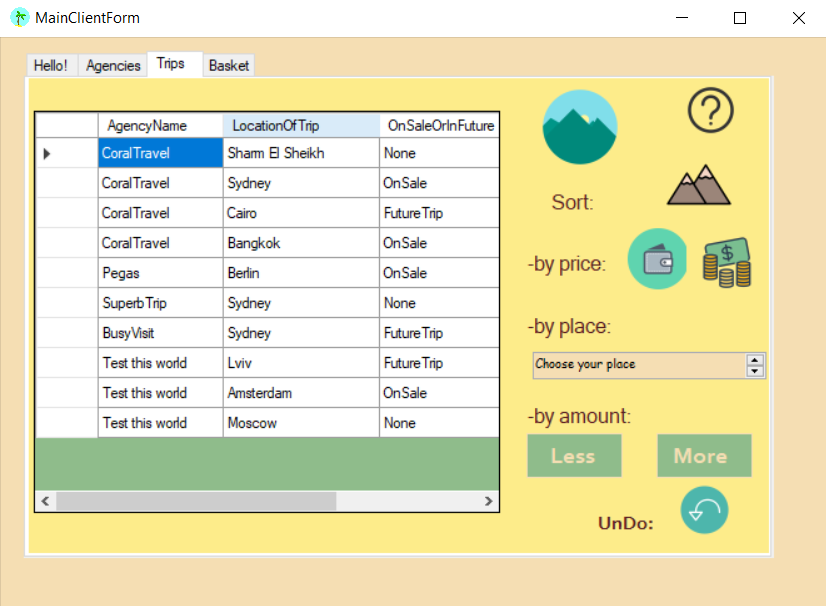


Рисунок 3.8 – Сторінка подорожей

На сторінці з кошиком(Див. рис. 3.9) розміщене замовлення, можна видалити ту подорож, яка не потрібна. Або замовити обрані подорожі натиском на кнопку «Complete». Наступна серія замовлень буде неможлива, доки адміністратор не обробить замовлення.

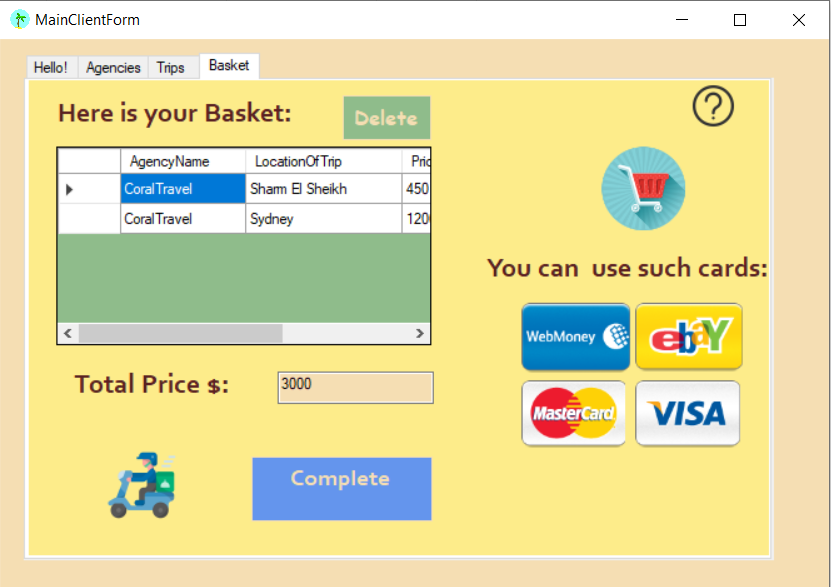


Рисунок 3.8 – Сторінка з кошиком

* 1. Робота програми для адміністратора

При запуску програми з’ являється вікно із входом до системи (Див. рис. 3.9). Адміністратору необхідно ввести свій логін , та пароль, або натиснути на кнопку «Sign up», якщо він не є зареєстрованим у системі. У разі введення некоректних даних, по виводиться повідомлення про вид помилки. (Див. рис. 3.9 та 3.10)

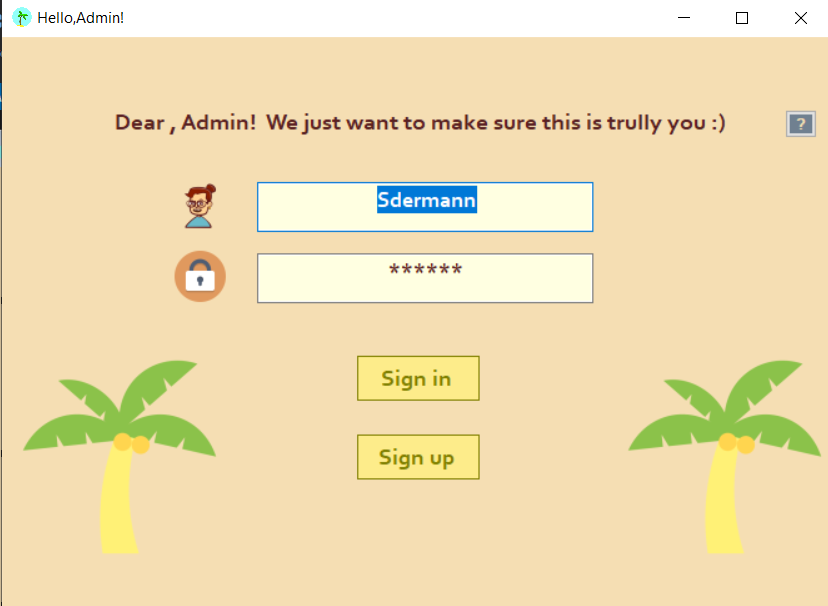


Рисунок 3.9 – Вікно входу для адміністратора

Якщо адміністратор новий, необхідно зареєструватися за допомогою секретного паролю. Якщо данні введені невірно, з’явиться повідомлення.

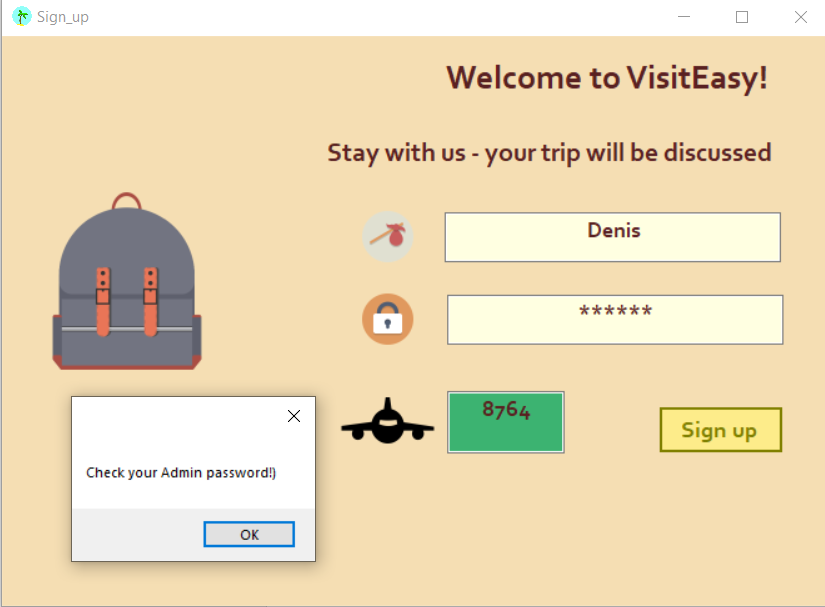


Рисунок 3.10 – Вікно реєстрації для адміністратора

Після входу/реєстрації відкривається основна панель адмінітсратора (Див. рис. 3.11).На ній можна натиснути на агентство та обравши певну опцію можна відредагувати, видалити, додати або створити нове, за допомогою форми редагування агентства (Див. рис. 3.12). Аналогічні дії можна зробити з подорожами агентства за допомогою форми редагування подорожі. Всі данні необхідно заповнити без допуску помилок. Агентства без подорожей (або їх кількість 0) необхідно видалити або відредагувати. Колір подорожей позначає статус (блакитний-подорожі майбутнього, червоний – зі знижкою, жовтий-звичайні).

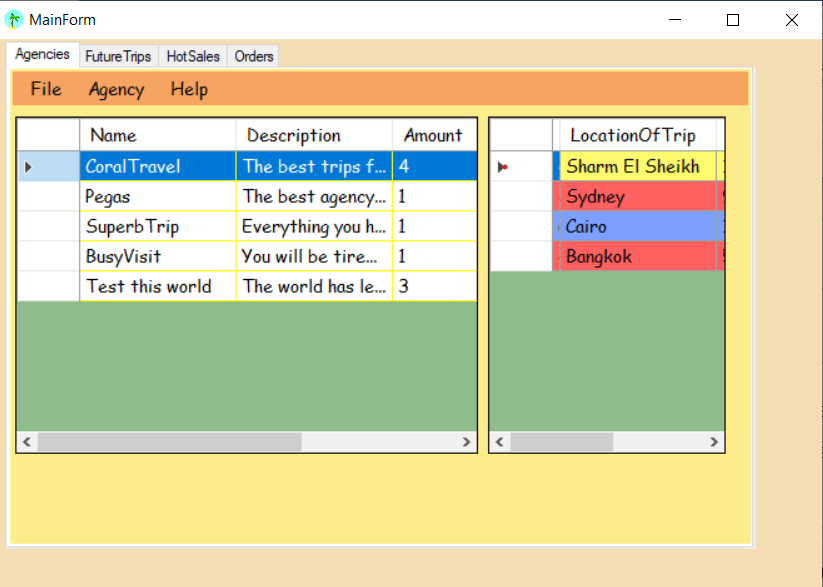


Рисунок 3.12 –Головна панель адміністратора

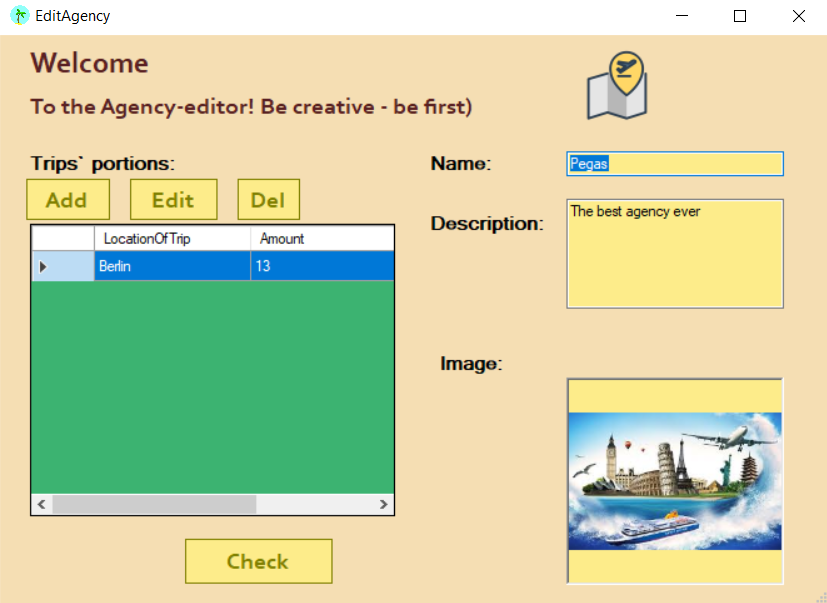


Рисунок 3.13 – Вікно редагування агентства

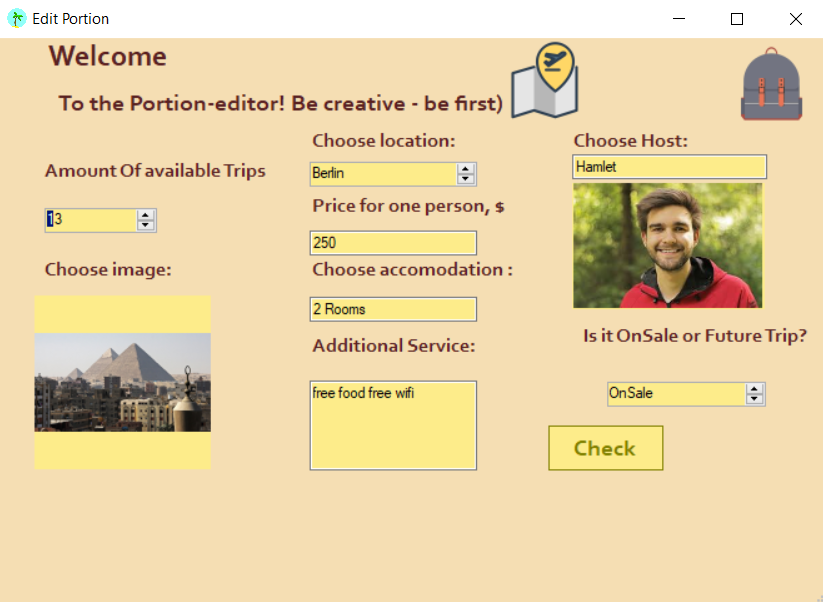


Рисунок 3.14 – Вікно редагування подорожі

На сторінці «Future trips» (Див. рис. 3.15). та «Hot Sales» (Див. рис. 3.16). відображено списки подорожей за такими статусами відповідно. Це задля комфорту адміністратора.

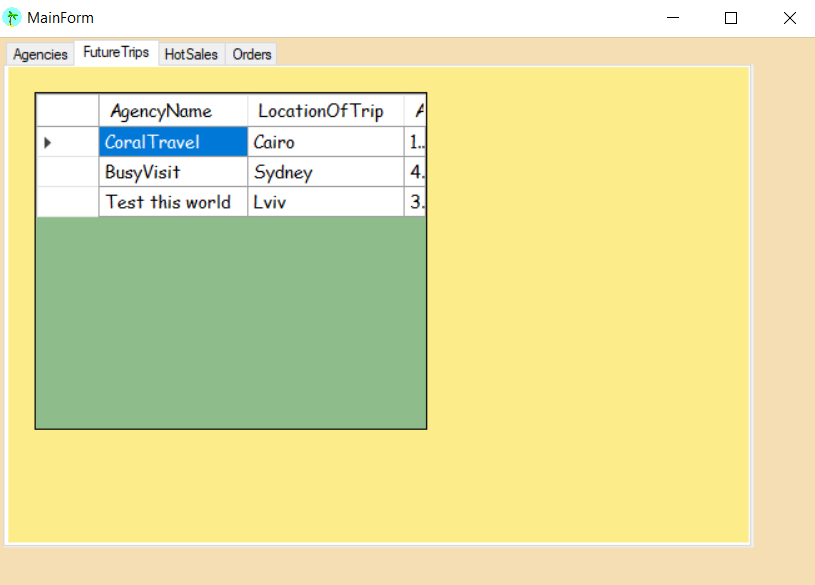


Рисунок 3.15 – Сторінка подорожей майбутнього

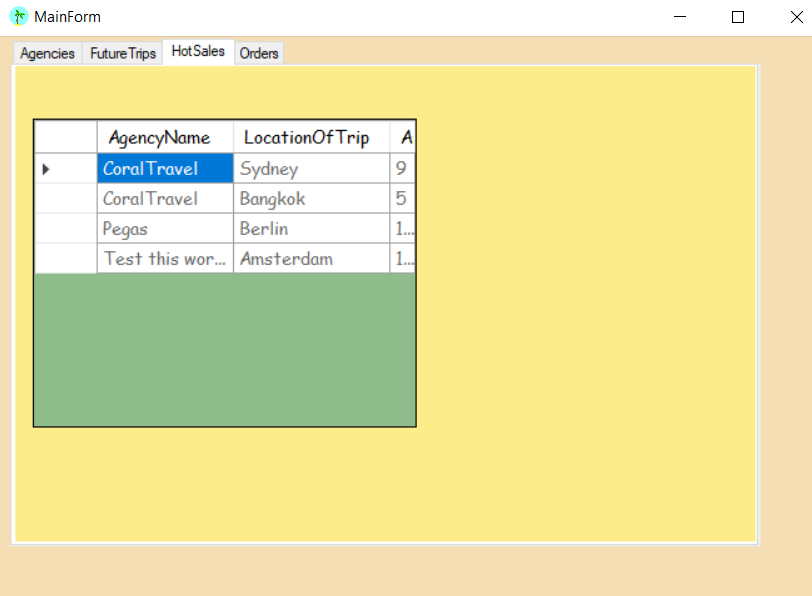


Рисунок 3.16 – Сторінка подорожей зі знижкою

На сторінці з замовленням є клієнти (Див. рис. 3.17), при натискання на яких можна переглянути замовлення або заблокувати клієнта. При натисканні на кнопку «View» можна переглянути замовлення за допомогою форми перегляду замовлення (Див. рис. 3.18). При натисканні на кнопку «Operate» можна його опрацювати.

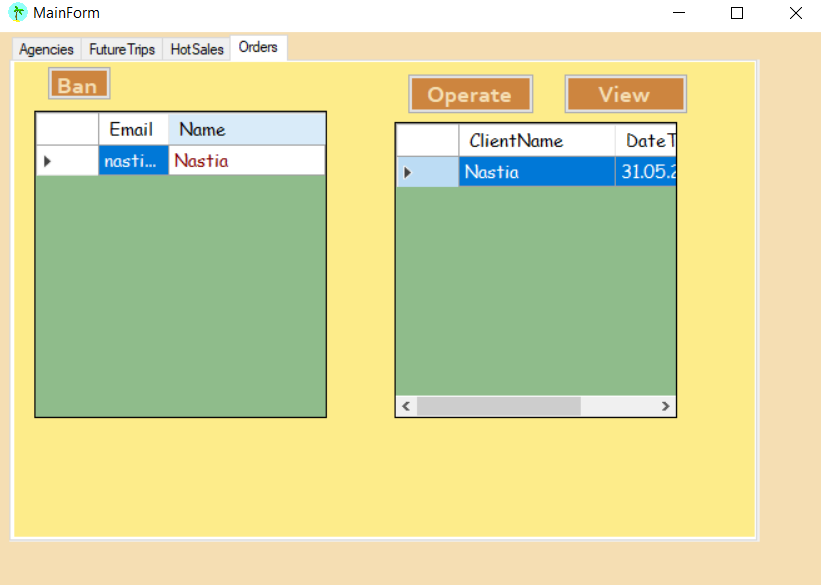


Рисунок 3.17 – Сторінка замовлень

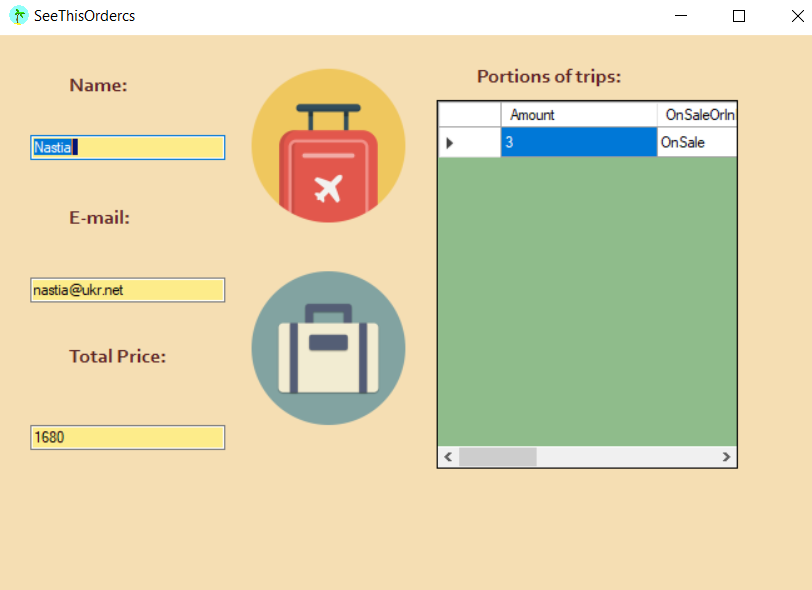


Рисунок 3.18 – Вікно для перегляду замовлення

Після опрацювання замовлення дані про прибуток з’являються в файлі

«Сompletedorders.txt» (Див. рис. 3.19).

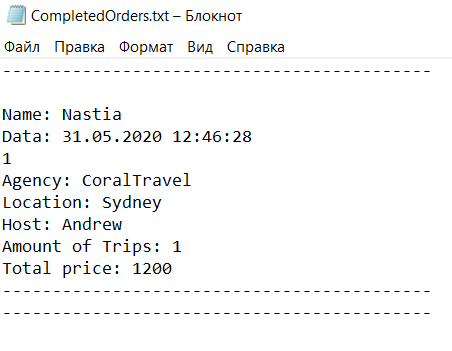


Рисунок 3.19 – Приклад звіту

**ВИСНОВКИ**

За результатами виконання курсового проекту було створену програмне забезпечення «Довідник туриста», що зберігає та автоматично оновлює інформацію про доступні агентства та подорожі, надає змогу клієнтам реєструватись у системі, робити покупки, робити свої мрії реальністю. Відповідно адміністратор має змогу редагувати інформацію про існуючі агентства та подорожі, блокувати клієнтів, переглядати замовлення та затверджувати їх.

Під час створення проекту було глибоко вивчено особливості та принципи об’єктно-орієнтованого програмування: взаємодію між класами, поняття наслідування та композиції, поліморфізм. Було розглянуто специфіку мови програмування С#. Також було детально вивчено особливості створення інтерфейсу користувача, взаємодію програмних класів із вікнами  Windows Forms. Усе це дало комплексне розуміння про процес створення програмного продукту від початку та до кінця: формулювання завдання, виділення чітких умов та критеріїв, візуалізація компонентів програми, розробка моделі класів та принципів взаємодії між ними.

Шляхами поліпшення запропонованої програми є розширення функціоналу, під’єднання до бази даних (для можливості перегляду списку продуктів з будь-якого ПК), розширення мережі VisitEasy, додавання кольорів подорожей і для клієнта.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Бондарев В.М. Объектно-ориентированное программирование на С#: учебное пособие. Харьков: СМИТ, 2009г. 221 с.
2. Документація по C# від Microsoft. URL: https://docs.microsoft.com/en-us/dotnet/csharp/ (дата звернення: 25.05.2020).
3. Документація по Windows Forms. URL: https://docs.microsoft.com/en-us/dotnet/framework/winforms/ (дата звернення: 25.05.2020).
4. Троелсен, Э, Джепикс, Ф. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. СПб. : ООО “Диалектика”, 2018 — 1328 с. 978-5-6040723-1-8
5. Документація по .NET та C#. URL: https://metanit.com/sharp/ (дата звернення: 25.05.2020).
6. ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. К.: ДП «УкрНДНЦ», 2016. 26 с.

**ДОДАТОК А**

Клас VisitEasy

using TravelAgency.DAl;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TravelAgency.Models

{

//Store VisitEasy is a collection of agencies + clients + admins + orders.

[Serializable]

public class VisitEasy

{

public List<Agency> Agencies { private set; get; }

public List<Client> Clients { private set; get; }

public List<Admin> Admins { private set; get; }

public List<Order> Orders {private set; get; }

public VisitEasy()

{

Agencies = new List<Agency>();

Clients = new List<Client>();

Admins = new List<Admin>();

Orders = new List<Order>();

}

// If any data changed.

public bool IsDirty;

public void FillTestData(int n)

{

// Agencies.

Agencies = new List<Agency>();

var noImage = new Bitmap(Path.GetFullPath("palm.png"));

for (int i = 0; i < n; i++)

{

List<Trip> Trips = new List<Trip>();

for (int o = 0; o < 5; o++)

{

Trips.Add(new Trip()

{

Location = $"{i}",

Price = i,

AdditionalService = $"{i}",

Accomodation = $"{i}",

Host = $"{i}",

});

}

var ps = new List<Portion>();

for (int j = 0; j < 5; j++)

{

ps.Add(new Portion { Trip = Trips[(j)], Amount = j,OnSaleOrInFuture = "OnSale"});

}

Agencies.Add(new Agency($"Name{i}","Good",0,i,ps,noImage)) ;

}

// Clients

Clients = new List<Client>();

for (int i = 1; i <= n; i++)

{

Clients.Add(new Client ($"Client{i}", 123,"E-mail" ));

}

// Orders

Orders = new List<Order>();

for (int i = 0; i < n - 5; i++)

{

List<Trip> Trips = new List<Trip>();

for (int o = 0; o < 5; o++)

{

Trips.Add(new Trip()

{

Location = $"Location{i}",

Price = i,

AdditionalService = $"{i}",

Accomodation = $"{i}",

Host = $"{i}",

});

}

var ps = new List<Portion>();

for (int j = 0; j < 5; j++)

{

ps.Add(new Portion { Trip = Trips[(j)], Amount = j });

}

Orders.Add(new Order(ps, Clients[i],DateTime.Now + TimeSpan.FromDays(i)));

}

}

public void AddAgency(Agency agency)

{

Agencies.Add(agency);

}

//////////////////////////////////////////////

public void Save()

{

new Dao(this).Save();

IsDirty = false;

}

public void Load()

{

new Dao(this).Load();

IsDirty = false;

}

}

}

**ДОДАТОК Б**

Клас Order

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TravelAgency.Models

{

// Order is a collection of potions + date + Client + status.

[Serializable]

public class Order

{

public Order(List<Portion> portions, Client client, DateTime? t = null)

{

Portions = portions == null ? new List<Portion>() : portions ;

ClientName = client.Name;

Client = client;

if (t == null)

{

DateTime = DateTime.Now;

}

else

{

DateTime = (DateTime)t;

}

IsOrdered = false;

CheckedByAdmin = false;

}

public List<Portion> Portions { set; get; }

public Client Client { private set; get; }

public string ClientName { set; get; }

public DateTime DateTime { private set; get; }

public bool IsOrdered { set; get; }

public bool CheckedByAdmin { set; get; }

}

}

**ДОДАТОК В**

Клас Agency

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TravelAgency.Models

{

//Agency is a collection of portions + name + description + amount of likes + amount of trips + image.

[Serializable]

public class Agency

{

public Agency(string name = "Hello", string description = "I`m going to hell", int amountOfTrips = 0, int amountOflikes = 0, List<Portion> portions = null, Image image = null)

{

Name = name;

Description = description;

Portions = portions == null ? new List<Portion>():portions ;

AmountOfTrips = amountOfTrips;

AmountOfLikes = amountOflikes;

Image = image;

if (portions != null)

{

foreach (Portion p in Portions)

{

p.AgencyName = name;

AmountOfLikes += p.Trip.Counter;

}

}

}

public string Name { set; get; }

public string Description { set; get; }

public int AmountOfTrips { set; get; }

public int AmountOfLikes {set; get; }

public List<Portion> Portions {set; get; }

public Image Image { set; get; }

}

**ДОДАТОК Г**

Клас Portion

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TravelAgency.Models

{

// Portions is a trip and its amount.

[Serializable]

public class Portion

{

public Trip Trip { set; get; }

public int Amount { set; get; }

public string OnSaleOrInFuture { set; get; }

public string AgencyName { set; get;}

public string LocationOfTrip { set; get; }

public decimal PriceOfEachTrip { set; get; }

public Portion(Trip trip = null, int amount = 0,string onSaleOrInFuture = "None")

{

Trip = trip==null? new Trip(): trip;

Amount = amount;

OnSaleOrInFuture = onSaleOrInFuture;

if (trip != null)

{

LocationOfTrip = trip.Location;

PriceOfEachTrip = trip.Price;

}

}

}

}

Ссылка на гитхаб: <https://github.com/sdermann/travel-agency-app>

Просьба скачивать вместе с папкой Images