

**Міністерство освіти і науки України**  
**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з самостійної роботи №2

з дисципліни «Системний аналіз» на тему:

«Автоматизована система купівлі авіаквитків»

**Виконав:**

Студент групи КНТ-122

О. А. Онищенко

**Прийняли:**

Викладач:

П. І. Льовкін

2024

# АВТОМАТИЗОВАНА СИСТЕМА КУПІВЛІ АВІАКВИТКІВ

## Вступ

Ландшафт авіаперевезень суттєво змінився з появою автоматизованих систем продажу авіаквитків, які спростили процес бронювання та покращили якість обслуговування клієнтів. Незважаючи на значний прогрес, що призвів до ефективних процесів бронювання та оплати, залишаються проблеми з інтеграцією цих систем з динамічними моделями ціноутворення та багатоканальними мережами дистрибуції.

Провідні компанії в цій галузі, такі як Amadeus, Sabre і Travelport, продовжують впроваджувати інновації, спираючись на ідеї таких експертів галузі, як Алекс Кремер і Генрі Хартевельдт. Світові тенденції свідчать про перехід до більш персоналізованих і зручних інтерфейсів, використання штучного інтелекту для прогнозування поведінки споживачів та оптимізації продажів.

Актуальність цього дослідження полягає в тому, що воно може заповнити існуючі прогалини, пропонуючи систему, яка не тільки спрощує транзакції, але й пропонує предиктивну аналітику для управління запасами. Метою цієї роботи є розробка комплексного рішення, яке задовольнить потреби як авіакомпаній, так і мандрівників, і може бути застосоване на різних платформах і пристроях.

Ця робота ґрунтується на попередніх дослідженнях у цій галузі, спрямованих на синтез найкращих практик та впровадження нових підходів до дизайну та функціональності системи. Вона є свідченням постійного розвитку технологій авіаперевезень, спрямованих на значний прогрес у цій галузі.

## **Основна Частина**

### **Проектування Системи**

#### **Вимоги до програмного забезпечення**

Вимоги до програмного забезпечення є наступними:

- Середовище розробки: Visual Studio
- Мова програмування: C#
- Фреймворк: .NET для WinForms
- База даних: Текстові файли (flights.txt, tickets.txt, users.txt)

#### **Вхідні та вихідні дані**

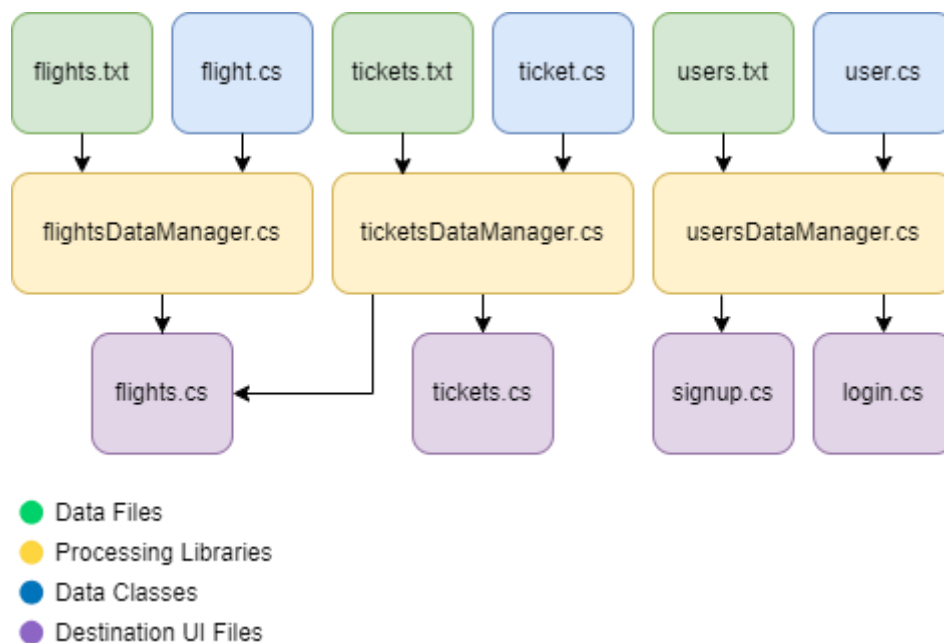
Вхідні та вихідні дані є наступними:

- Вхідні: Дані, введені користувачем через форми графічного інтерфейсу.
- Вихідні: Відображення даних у графічному інтерфейсі, оновлені текстові файли.

#### **Спосіб роботи з даними**

Спосіб роботи з даними є прямолінійним: користувач взаємодіє з графічним інтерфейсом, в залежності від виконаних дій до бази даних надходять запити про створення об'єктів на основі класів даних. Після створення об'єкти заносяться у базу даних, а файли зберігаються. При потребі, в залежності від виконаних дій, до бази даних надходять запити на вибірку даних, а ці дані подаються через форми графічного інтерфейсу до користувача.

Структура бази даних виглядає наступним чином (Рис. 1.1):



Структура бази даних (Рис. 1.1)

## Прототип графічного інтерфейсу користувача

Схематичний прототип графічного інтерфейсу користувача:

- Login Form
- Signup Form
- Menu Form
- Flights Form
- Tickets Form

Нижче наведено графічні форми прототипи графічного інтерфейсу користувача (Рис. 1.2-7):

### Flights

```
graph TD; back[back] --> flights_list[available_flights_list]; flights_list --> options[flight_customization_options]; options --> book_ticket[book_ticket];
```

The Flights form consists of four main components arranged vertically: a 'back' button at the top, followed by a large rectangular area for the 'available\_flights\_list', then a section for 'flight\_customization\_options', and finally a 'book\_ticket' button at the bottom.

Прототип формы Flights (Рис. 1.2)

### Login

```
graph TD; username[user_name] --> password[user_password]; password --> login[login]; login --> signup[sign up];
```

The Login form features two input fields: 'user\_name' and 'user\_password'. Below these are two buttons: 'login' and 'sign up'.

Прототип формы Login (Рис. 1.3)

### Menu

```
graph TD; tickets[tickets] --> flights[flights]; flights --> sign_out[sign out];
```

The Menu form contains three buttons arranged vertically: 'tickets', 'flights', and 'sign out'.

Прототип формы Menu (Рис. 1.4)

Signup

☒ is\_admin

Прототип форми Signup (Рис. 1.5)

Tickets

Прототип форми Tickets (Рис. 1.6)

- Form Name (plain text)
- Buttons (rounded corners)
- Data Fields or Inputs (square corners)

Значення різних фігур на прототипах (Рис. 1.7)

## Система об'єктів

Всі класи даних та обробники даних з їх інтерфейсами знаходяться в окремій бібліотеці класів `Class_Library`, з якої ми імпортуємо скомпільований `.dll` файл, аби потім використовувати їх в інтерфейсі нашої програми.

Система об'єктів, класів даних та обробників даних схематично виглядає наступним чином:

## Class\_Library

currentUser.cs

```
string name
string password
bool isAdmin
```

flight.cs

```
string name
int price
string date
int seats
```

flightsDataManager.cs

```
interface IFlightsDataManager
List<flight> loadFlights()
flight getFlight(string name)
```

ticket.cs

```
string userName
string flightName
int price
string date
int seatRow
bool isMiddle
bool isWindow
bool isPrivate
bool isBaggage
bool isMeal
```

ticketsDataManager.cs

```
List<ticket> loadTickets()
ticket GetTicket(string flightName)
List<ticket> GetOwnTickets(string userName)
void AddTicket(string userName, string flightName, int price,
string date, int seatRow, bool isMiddle, bool isWindow, bool isPrivate,
bool isBaggage, bool isMeal)
```

user.cs

```
string name
string password
bool isAdmin
```

userManager.cs

```
List<user> loadUsers()
void addUser(string name, string password, bool isAdmin)
user getUser(string name)
bool isAdmin(string name)
bool validateCredentials(string name, string password)
```

## App

```
... all the forms and data files ...
flights.txt
    ... rows of data ...
    flight_name,price,date,seats
tickets.txt
    ... rows of data ...

user_name,flight_name,price,date,seat_row,is_middle,is_window,is_privat
e,is_baggage,is_meal
users.txt
    ... rows of data ...
    name,password,is_admin
flights.cs
    flights.Designer.cs
login.cs
    login.Designer.cs
menu.cs
    menu.Designer.cs
signup.cs
    signup.Designer.cs
tickets.cs
    tickets.Designer.cs
```

У графічному вигляді схема даних виглядає наступним чином (Рис. 1.8):



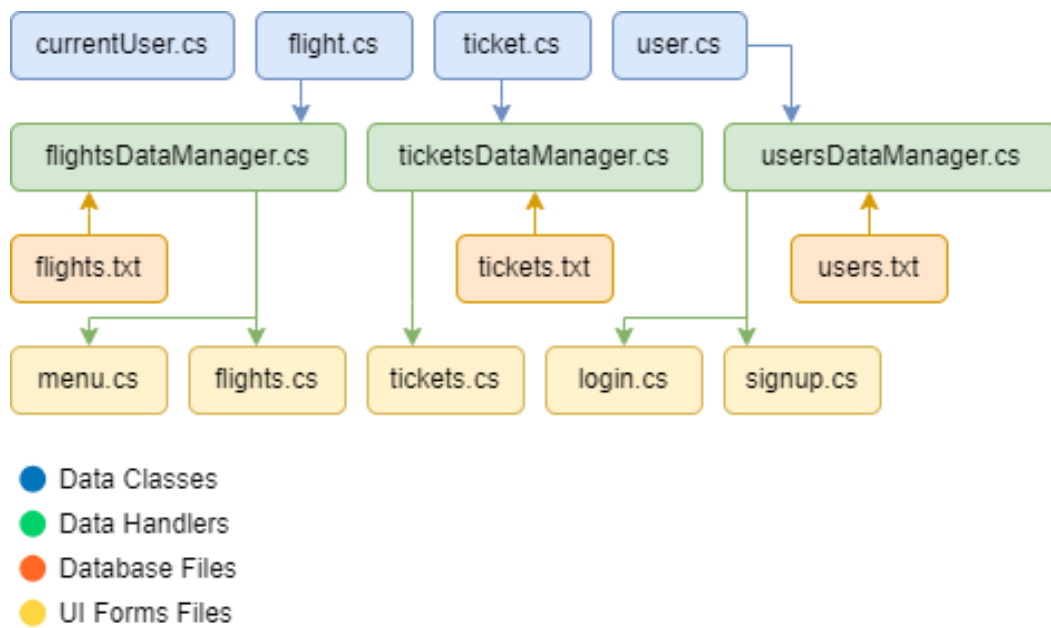
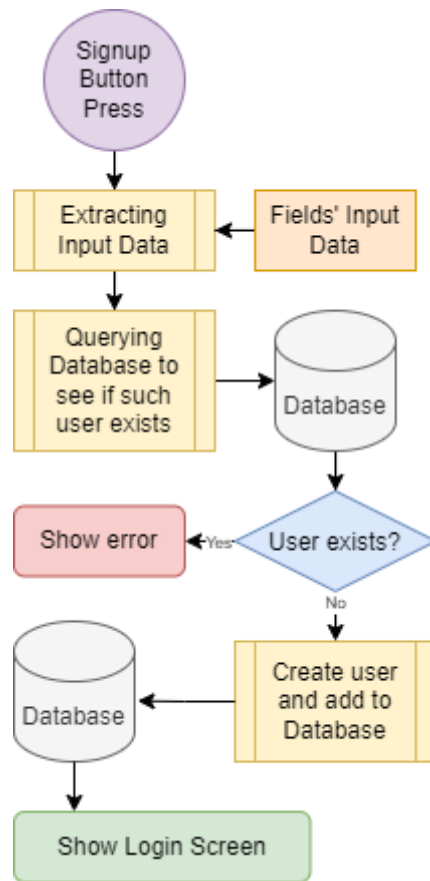


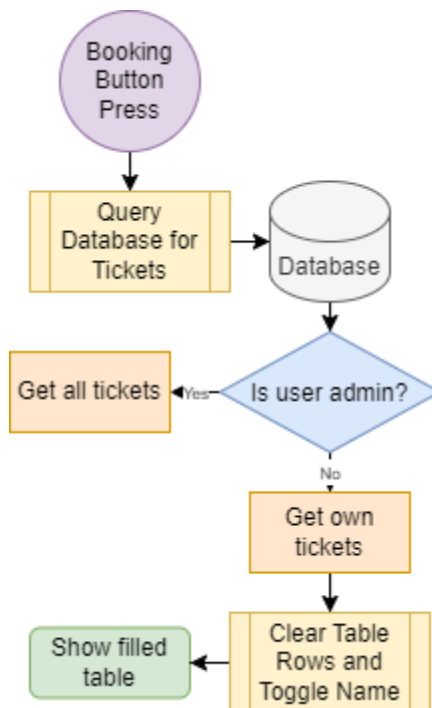
Схема даних у графічному вигляді (Рис. 1.8)

## Функціональні моделі процесів

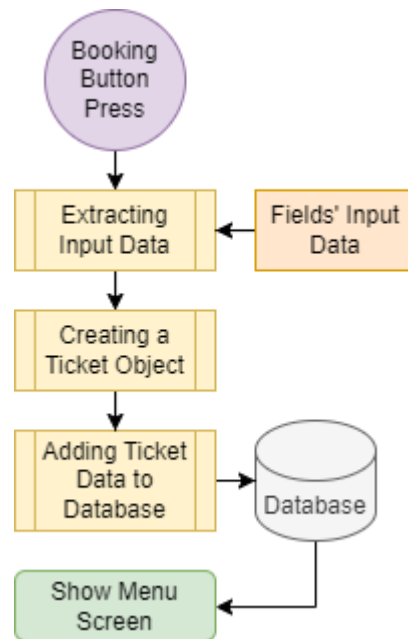
Основні функціональні моделі подано у графічному вигляді нижче (Рис. 1.9-11)



Процес реєстрації нового користувача (Рис. 1.9)



Процес перегляду квитків (Рис. 1.10)

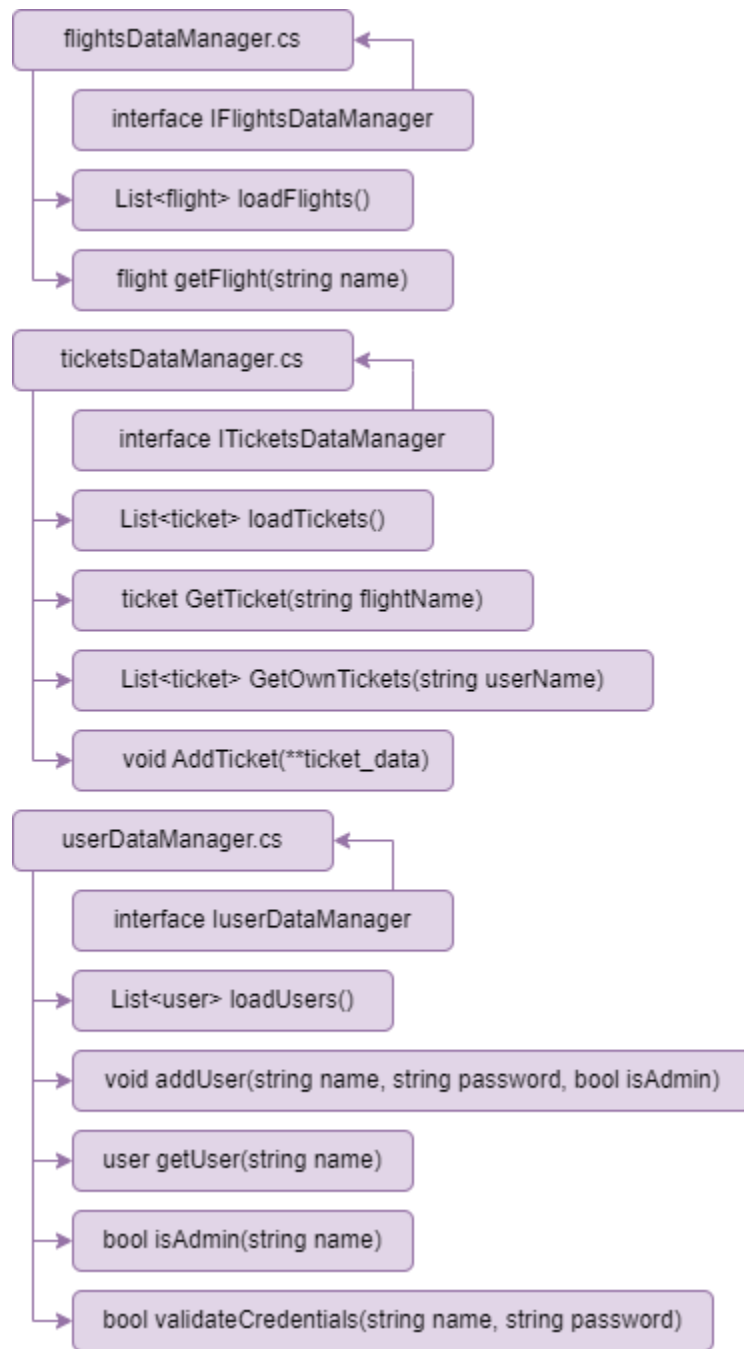


Процес замовлення квитку (Рис. 1.11)

## Опис Програми

### Функціональна схема

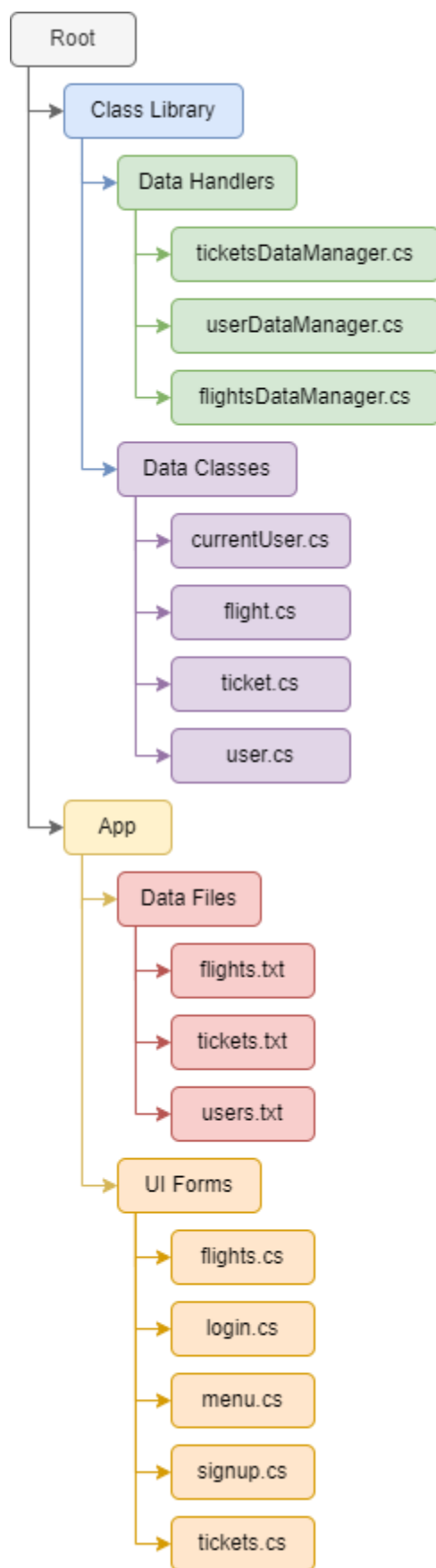
Функціональна схема програмного забезпечення має наступний вигляд (Рис. 2.1):



Функціональна схема програмного забезпечення (Рис. 2.1)

## Структурна схема

Структурна схема програмного забезпечення має наступний вигляд (Рис. 2.2):



Структурна схема програмного забезпечення (Рис. 2.2)

## Опис модулів

Структуру проєкту можна умовно поділити на чотири секції:

- обробники даних
- класи даних
- файли даних
- форми інтерфейсу користувача

Де обробники даних та класи даних належать до окремої бібліотеки Class\_Library, а файли даних та форми користувацького інтерфейсу лежать в основній теці App.

### Обробники даних

*ticketsDataManager.cs*

```
List<ticket> loadTickets(); // loads the list of all tickets
ticket GetTicket(string flightName); // returns a ticket by its flight
name
List<ticket> GetOwnTickets(string userName); // return a ticket given a
user name
void AddTicket(string userName, string flightName, int price, string
date, int seatRow, bool isMiddle, bool isWindow, bool isPrivate, bool
isBaggage, bool isMeal); // adds a new ticket given all the data
```

*userDataManager.cs*

```
List<user> loadUsers(); // loads the list of all users
void addUser(string name, string password, bool isAdmin); // adds a new
user given all the data
user getUser(string name); // returns a user by its name
bool isAdmin(string name); // returns true if the user is an admin
bool validateCredentials(string name, string password); // returns true
if the credentials are valid
```

## *flightsDataManager.cs*

```
List<flight> loadFlights(); // loads the list of all flights  
flight getFlight(string name); // returns a flight by its name
```

## **Класи даних**

### *currentUser.cs*

```
string name; // user name  
string password; // user password  
bool isAdmin; // true if the user is an admin
```

### *user.cs*

```
string name; // user name  
string password; // user password  
bool isAdmin; // true if the user is an admin
```

### *flight.cs*

```
string name; // flight name  
int price; // ticket default price  
string date; // flight date  
int seats; // number of seats
```

### *ticket.cs*

```
string userName; // user name  
string flightName; // flight name  
int price; // ticket total price  
string date; // flight date  
int seatRow; // seat row  
bool isMiddle; // true if the seat is in the middle row  
bool isWindow; // true if the seat is near the window  
bool isPrivate; // true if the seat is private  
bool isBaggage; // true if extra baggage is needed  
bool isMeal; // true if extra meal is needed
```

## **Файли даних**

### *flights.txt*

```
flight_name,price,date,seats
```

### *tickets.txt*

user\_name,flight\_name,price,date,seat\_row,is\_middle,is\_window,is\_private,is\_baggage,is\_meal

### *users.txt*

name,password,is\_admin

## **Форми користувачького інтерфейсу**

### *flights.cs*

```
const int collapsedHeight = 190; // the height of the collapsed window
const int expandedHeight = 560; // the height of the expanded window
flight selectedFlight; // the currently selected flight
int totalPrice; // the total price of the selected ticket
int seatTypePrice = 0; // the price of the selected seat type
int mealPrice = 0; // the price of the selected meal
int luggagePrice = 0; // the price of the selected baggage
int privatePrice = 0; // the price of the selected private seat

public flights(); // constructor, collapses the window height and loads
the list of all flights
void updateTotal(); // updates the total price label
renderFlightNames(List<flight> flights); // renders the list of flight
names
menuButton_Click(object sender, EventArgs e); // closes the window and
opens the menu
flightsList_SelectedValueChanged(object sender, EventArgs e); //
handles the selection of a flight, fetched the flight info from the
database, updates all the labels and the total price
middleRadio_CheckedChanged(object sender, EventArgs e); // unchecks
other radio buttons and updates the total price
randomRadio_CheckedChanged(object sender, EventArgs e); // unchecks
other radio buttons and updates the total price
windowRadio_CheckedChanged(object sender, EventArgs e); // unchecks
other radio buttons and updates the total price
mealBox_CheckedChanged(object sender, EventArgs e); // updates the
total price based on the status
luggageBox_CheckedChanged(object sender, EventArgs e); // updates the
total price based on the status
privateBox_CheckedChanged(object sender, EventArgs e); // updates the
total price based on the status
```



```
bookButton_Click(object sender, EventArgs e); // forms a new ticket,  
adds it to the database, shows the success message and closes the  
window
```

### *login.cs*

```
public login(); // constructor  
void loginButton_Click(object sender, EventArgs e); // validates the  
credentials, opens the menu if valid or shows an error if not  
void registerButton_Click(object sender, EventArgs e); // opens the  
signup window and closes the login window
```

### *menu.cs*

```
public menu(); // constructor  
void signoutButton_Click(object sender, EventArgs e); // signs out the  
user and closes the menu  
void flightsButton_Click(object sender, EventArgs e); // opens the  
flights window and closes the menu  
void ticketsButton_Click(object sender, EventArgs e); // opens the  
tickets window and closes the menu
```

### *signup.cs*

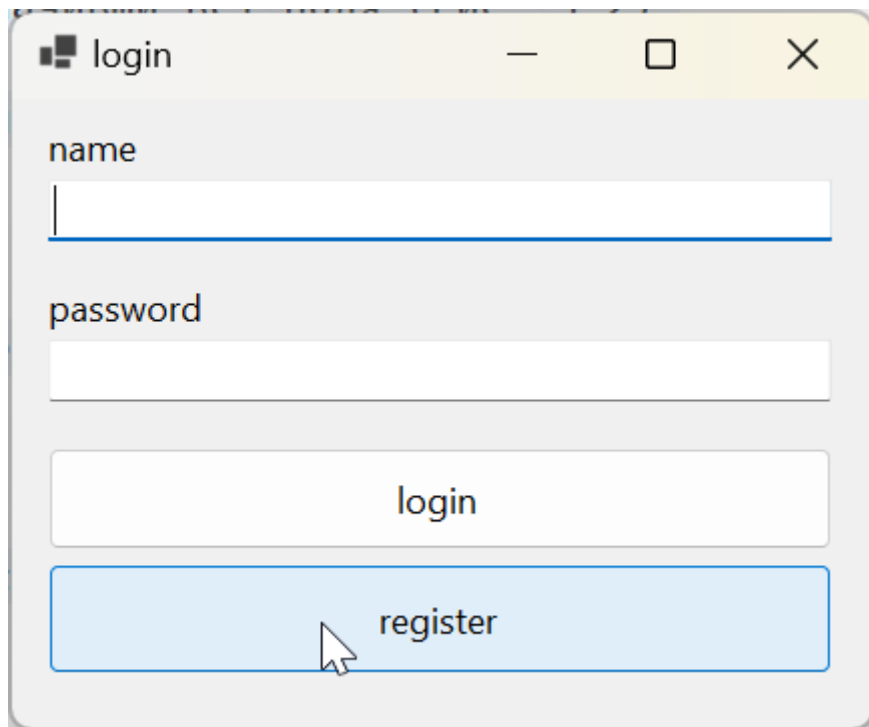
```
public signup(); // constructor  
void loginButton_Click(object sender, EventArgs e); // opens the login  
window and closes the signup window  
void signupButton_Click(object sender, EventArgs e); // creates a new  
user and closes the signup window
```

### *tickets.cs*

```
public tickets(); // constructor, loads the list of tickets  
void loadTickets(); // loads the list of tickets and renders it  
void backButton_Click(object sender, EventArgs e); // closes the window  
and opens the menu
```

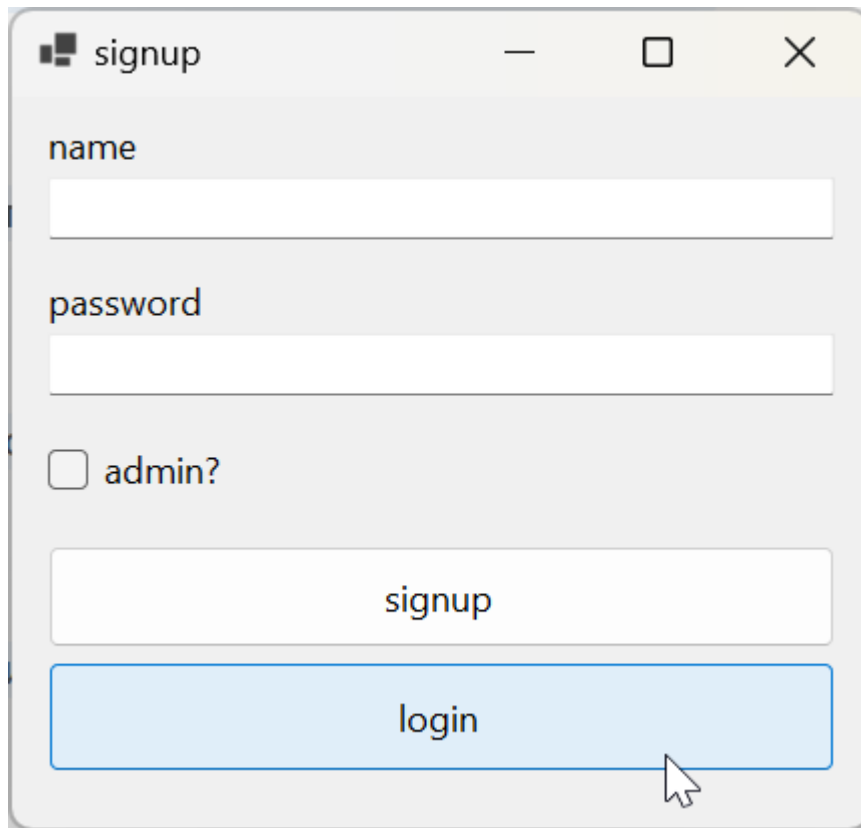
## **Опис роботи**

Робота з програмою завжди починається з вікна входу (Рис. 3.1):

A screenshot of a login window titled 'login'. The window has a standard title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains two text input fields. The first field is labeled 'name' and the second is labeled 'password'. Below the password field is a white button with the text 'login'. At the bottom is a larger, light blue button with the text 'register'. A mouse cursor is pointing at the 'register' button.

Вікно входу до програми (Рис. 3.1)

У вікні входу користувач може зайти у свій обліковий запис. Якщо користувач облікового запису не має, він може перейти до вікна реєстрації (Рис. 3.2)



signup

name

password

☐ admin?

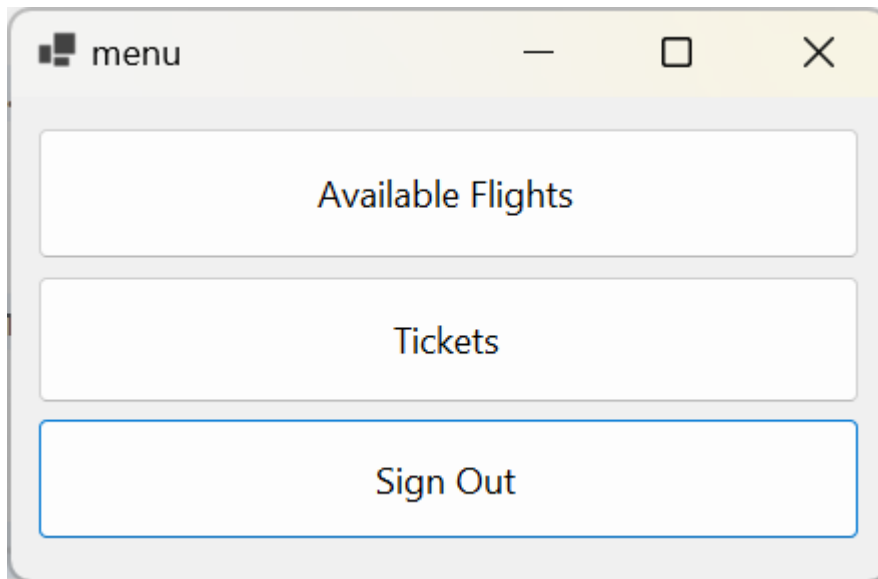
signup

login

Вікно реєстрації нового користувача (Рис. 3.2)

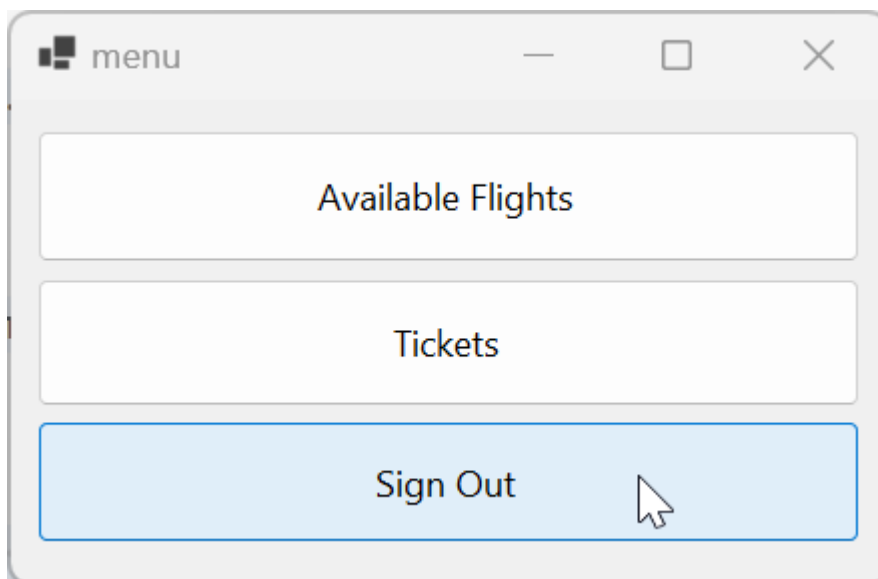
У вікні реєстрації користувач може створити обліковий запис, зазначивши всі поля (Рис. 3.2)

Після входу або реєстрації, користувач потрапляє до головного меню (Рис. 3.3)



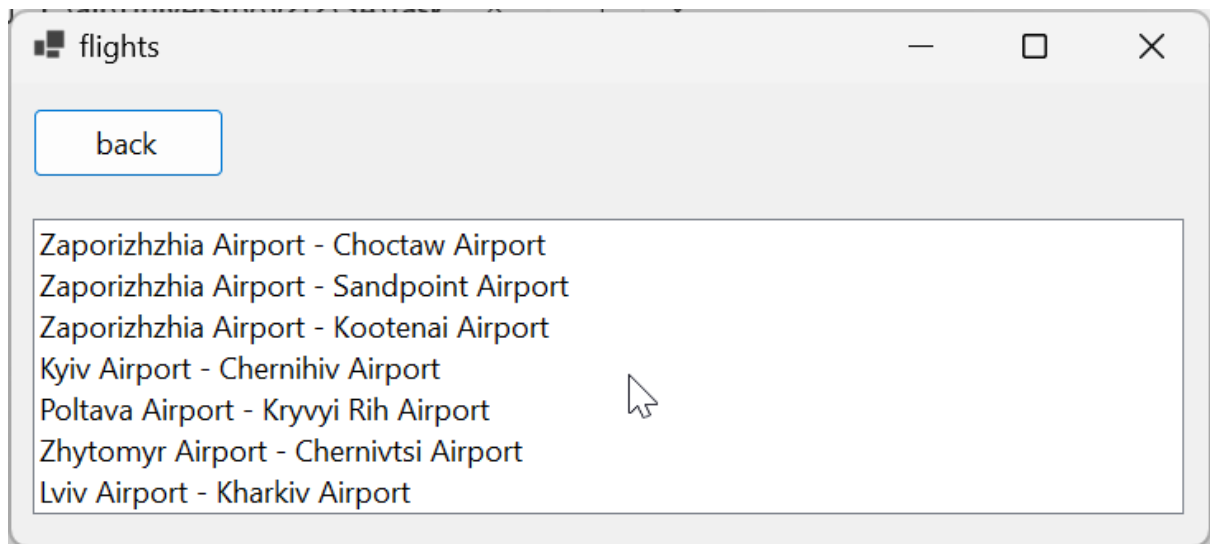
Головне меню програми (Рис. 3.3)

У головному меню користувач може вийти з облікового запису, що перенесе його назад до вікна входу (Рис. 3.4)



Кнопка виходу з облікового запису (Рис. 3.4)

У головному меню користувач може переглянути список польотів (Рис. 3.5)



Список доступних польотів (Рис. 3.5)

У списку польотів користувач може обрати підходящий польот, змінити параметри та замовити квиток (Рис. 3.6)

flights

—□×

back

Zaporizhzhia Airport - Choctaw Airport

Zaporizhzhia Airport - Sandpoint Airport

Zaporizhzhia Airport - Kootenai Airport

Kyiv Airport - Chernihiv Airport

Poltava Airport - Kryvyi Rih Airport

Zhytomyr Airport - Chernivtsi Airport

Lviv Airport - Kharkiv Airport

### Zaporizhzhia Airport - Kootenai Airport

Ticket price: \$30

Flight date: 13/05/2024

Available seats: 77

#### Select seat

Select seat row

7

Select seat type

☐ Random seat | +\$0

☒ Middle row | +\$10

☐ Window seat | +\$30

Select optional extra services

☐ Meal on board | +\$15

☐ Extra luggage space | +\$30

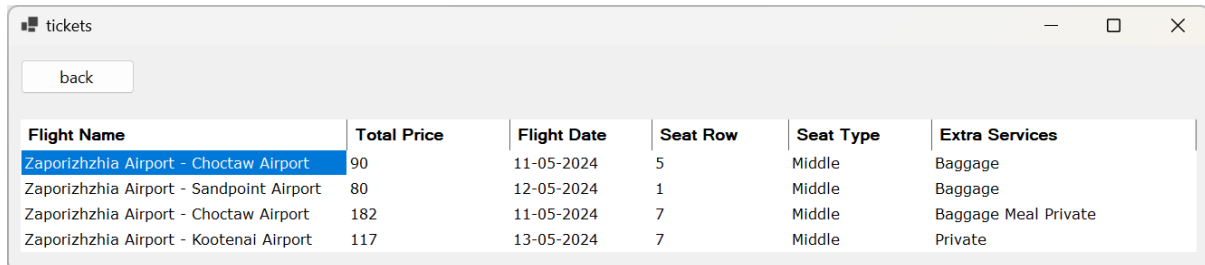
☒ Private compartment | +\$77

**In total: \$117**

Book Ticket

Кнопка замовлення квитка (Рис. 3.6)

Після замовлення квитка користувача переносить до головного меню, де він може переглянути свої квитки (Рис. 3.7)

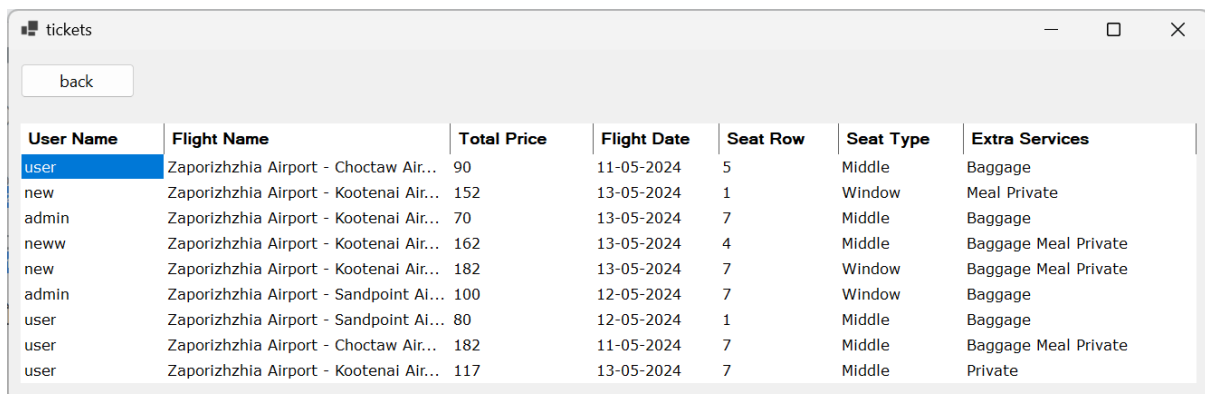


The screenshot shows a window titled 'tickets' with a 'back' button. Below it is a table with the following data:

Flight Name	Total Price	Flight Date	Seat Row	Seat Type	Extra Services
Zaporizhzhia Airport - Choctaw Airport	90	11-05-2024	5	Middle	Baggage
Zaporizhzhia Airport - Sandpoint Airport	80	12-05-2024	1	Middle	Baggage
Zaporizhzhia Airport - Choctaw Airport	182	11-05-2024	7	Middle	Baggage Meal Private
Zaporizhzhia Airport - Kootenai Airport	117	13-05-2024	7	Middle	Private

Список квитків користувача (Рис. 3.7)

Якщо користувач є адміністратором, йому будуть доступні квитки всіх користувачів (Рис. 3.8)



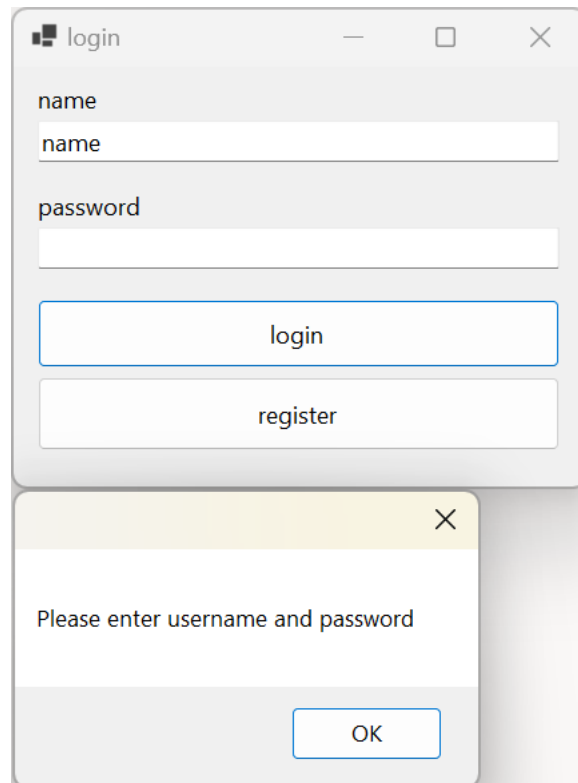
The screenshot shows a window titled 'tickets' with a 'back' button. Below it is a table with the following data:

User Name	Flight Name	Total Price	Flight Date	Seat Row	Seat Type	Extra Services
user	Zaporizhzhia Airport - Choctaw Air...	90	11-05-2024	5	Middle	Baggage
new	Zaporizhzhia Airport - Kootenai Air...	152	13-05-2024	1	Window	Meal Private
admin	Zaporizhzhia Airport - Kootenai Air...	70	13-05-2024	7	Middle	Baggage
neww	Zaporizhzhia Airport - Kootenai Air...	162	13-05-2024	4	Middle	Baggage Meal Private
new	Zaporizhzhia Airport - Kootenai Air...	182	13-05-2024	7	Window	Baggage Meal Private
admin	Zaporizhzhia Airport - Sandpoint Ai...	100	12-05-2024	7	Window	Baggage
user	Zaporizhzhia Airport - Sandpoint Ai...	80	12-05-2024	1	Middle	Baggage
user	Zaporizhzhia Airport - Choctaw Air...	182	11-05-2024	7	Middle	Baggage Meal Private
user	Zaporizhzhia Airport - Kootenai Air...	117	13-05-2024	7	Middle	Private

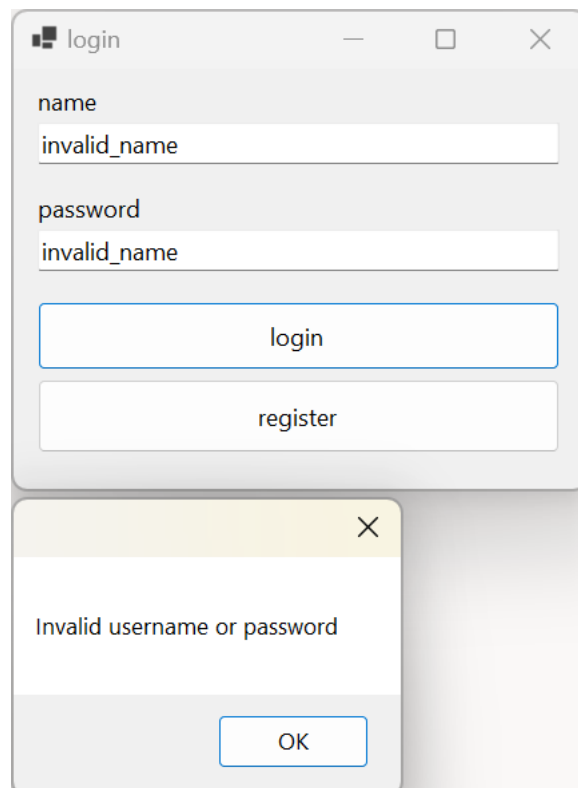
Список квитків для адміністратора (Рис. 3.8)

## Опис повідомлень

Програма містить наступні повідомлення (Рис. 4.1-4):

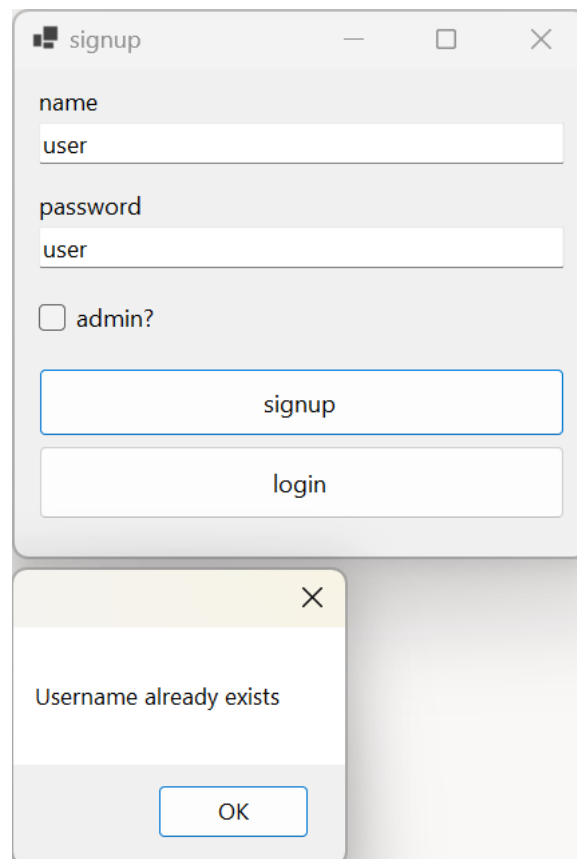


Помилка Відсутній пароль або ім'я (Рис. 4.1)

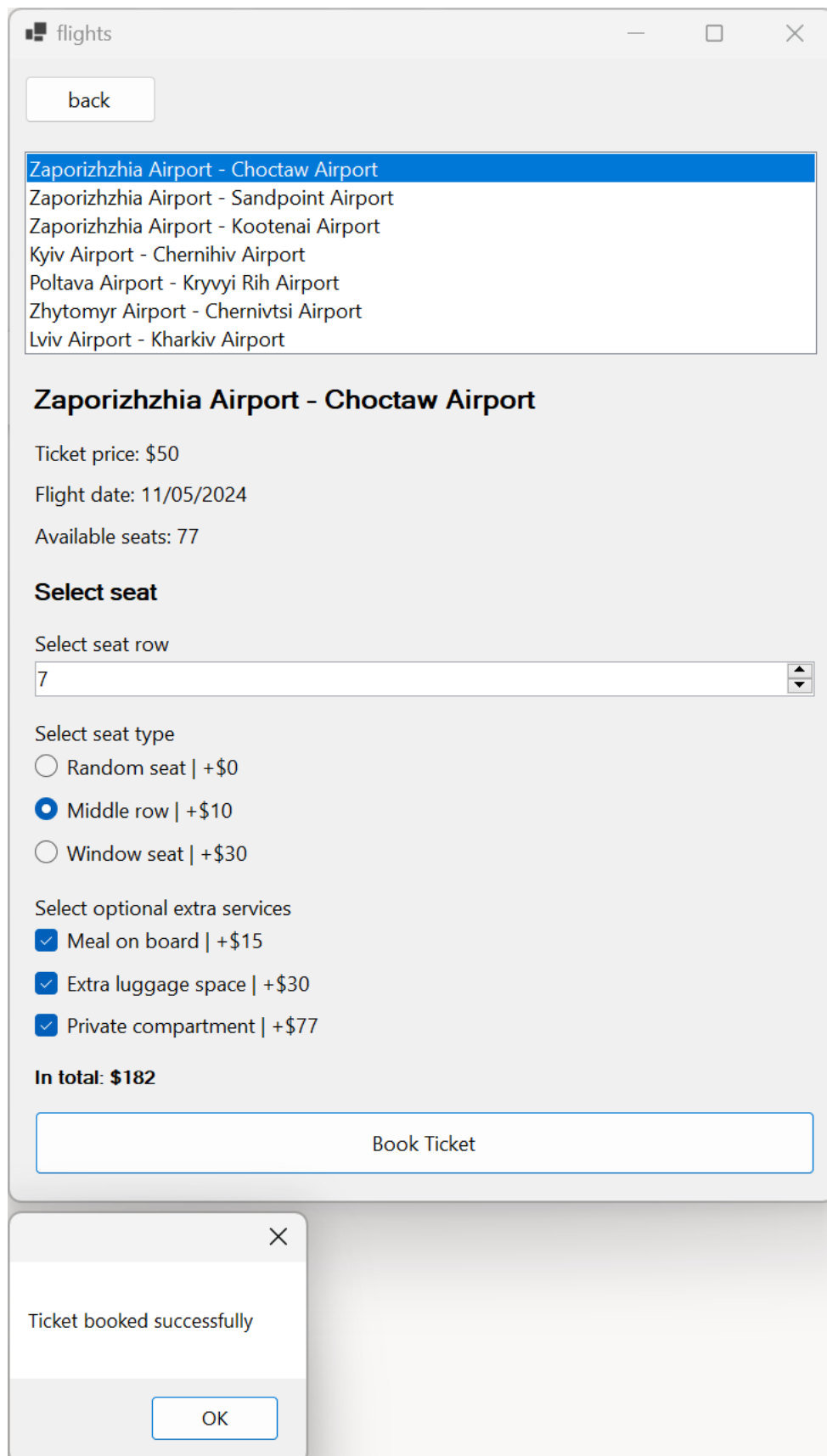


Помилка Некоректні облікові дані (Рис. 4.2)





Помилка Користувач вже існує (Рис. 4.3)



Підтвердження покупки квитка (Рис. 4.4)

## **Управління Ризиками**

### **Виявлення ризиків**

Виявимо ризики у проєкті:

- Корупція бази даних:

- Опис: База даних, що зберігає інформацію про рейси та квитки, може бути пошкоджена через апаратні збої, програмні помилки або інші проблеми.

- Сфери впливу: Цілісність даних, доступність системи.

- Зміни обсягу робіт:

- Опис: Зміни у вимогах або обсязі проєкту можуть призвести до додаткової роботи, затримок або невідповідності початковим цілям.

- Сфери впливу: План-графік проєкту, зусилля з розробки.

- Економічна нестабільність:

- Опис: Економічні коливання (наприклад, інфляція, девальвація валюти) можуть вплинути на витрати та фінансування проєкту.

- Сфери впливу: Бюджет проєкту, фінансова стабільність.

### **Ймовірність ризиків**

Оцінимо ймовірність кожного ризику:

- Корупція бази даних:

- Якісний: Помірний (завдяки регулярному резервному копіюванню та надійному управлінню базою даних).

- Зміни обсягу робіт:

- Якісний: Високий (зміни обсягу робіт є поширеним явищем у програмних проєктах).

- Економічна нестабільність:

- Якісний: Дуже високий (зовнішні фактори поза контролем людей).

### **Вплив ризиків**

Оцінимо вплив на аспекти проєкту:

- Корупція бази даних:
  - Якісний: Помірний (зусилля з відновлення даних, простої системи).
- Зміни обсягу робіт:
  - Якісний: Високий (доопрацювання, затримки).
- Економічна нестабільність:
  - Якісний: Високий (коригування бюджету, розподіл ресурсів).

### **Оцінка до управління**

Поєднаємо ймовірності та вплив:

- Корупція бази даних: Важливий (помірна ймовірність, помірний вплив).
- Зміни обсягу робіт: Критичний (висока ймовірність, високий вплив).
- Економічна нестабільність: Критичний (дуже висока ймовірність, високий вплив).

### **Оцінка після управління**

Оцінимо ризики після управлінню:

- Корупція бази даних: Мінімізований (стратегії резервного копіювання, моніторинг).
- Зміни обсягу робіт: Заходи в разі непередбачуваних ситуацій (процес управління змінами).
- Економічна нестабільність: Моніторинг та адаптація (фінансові резерви).

## **Висновки**

*Бо заплата за гріх смерть, а благодатний дар  
Божий вічне життя в Христі Ісусі, Господі нашім  
([Римляни 6:23](#))*

На закінчення, розробка передової автоматизованої системи продажу квитків являє собою значний стрибок вперед в індустрії авіаперевезень. Система, розроблена в цій роботі, використовуючи описану структуру проекту, пропонує надійне рішення, яке вирішує поточні проблеми у сфері бронювання та ціноутворення, забезпечуючи при цьому предиктивну аналітику для управління запасами.

Інтеграція зручного інтерфейсу з можливостями прогнозування на основі штучного інтелекту гарантує, що авіакомпанії зможуть пропонувати персоналізований досвід для мандрівників, оптимізуючи продажі та задоволеність клієнтів. Наукова та соціальна значущість цієї роботи полягає в тому, що вона може революціонізувати спосіб взаємодії авіакомпаній та мандрівників, зробивши авіаперевезення більш доступними та ефективними.

Якісні та кількісні показники успіху цієї системи включають покращення простоти транзакцій, підвищення точності управління запасами та покращення користувацького досвіду. Надійність цих

результатів ґрунтується на комплексному підході, застосованому при розробці системи, що відображено в структурі проекту.

Рекомендації щодо впровадження включають поетапне розгортання системи на різних платформах з постійним моніторингом та механізмами зворотного зв'язку для забезпечення адаптивності до мінливих потреб ринку. Ця робота не тільки сприяє розширенню знань у галузі технологій повітряних перевезень, але й створює прецедент для майбутніх інновацій, спрямованих на покращення глобального досвіду подорожей.