|  |  |
| --- | --- |
|  | **AKADEMIA GÓRNICZO-HUTNICZA** |

Dokumentacja do projektu

**Airline Reservation System**

z przedmiotu

**Programowanie obiektowe**

Elektronika, 2 rok

*Łukasz Piasecki*

prowadzący: Rafał Frączek

19.01.2021

# Opis projektu

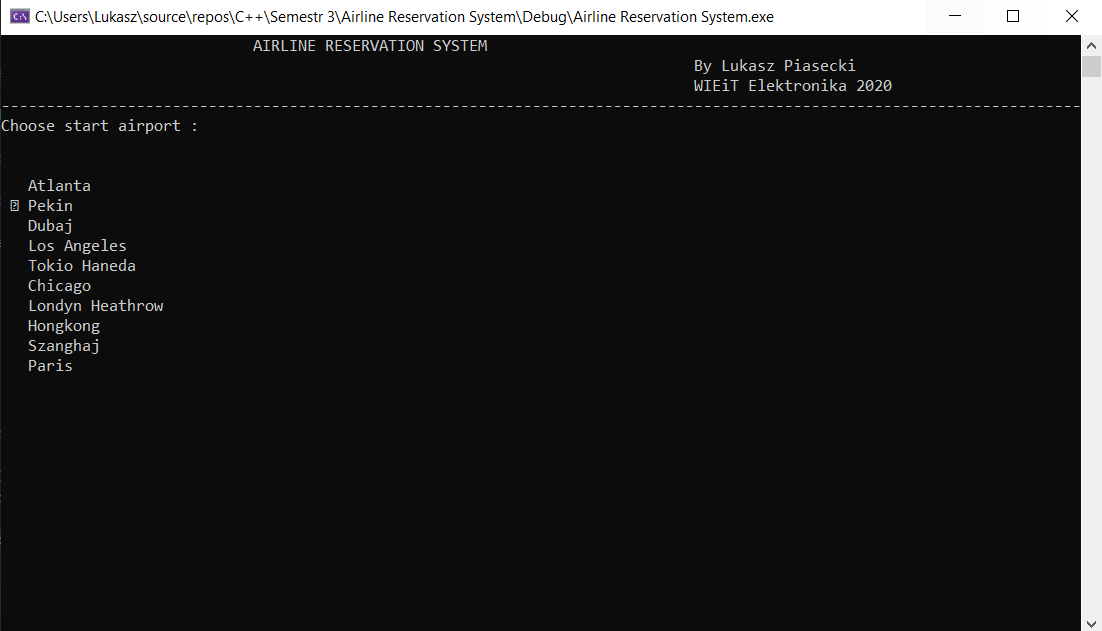
Program służy do rezerwacji miejsc w samolocie. Wykorzystywany może być na platformach zajmujących się sprzedażą biletów zarówno lotniczych jak i biletów innych środków komunikacyjnych.

# Project description

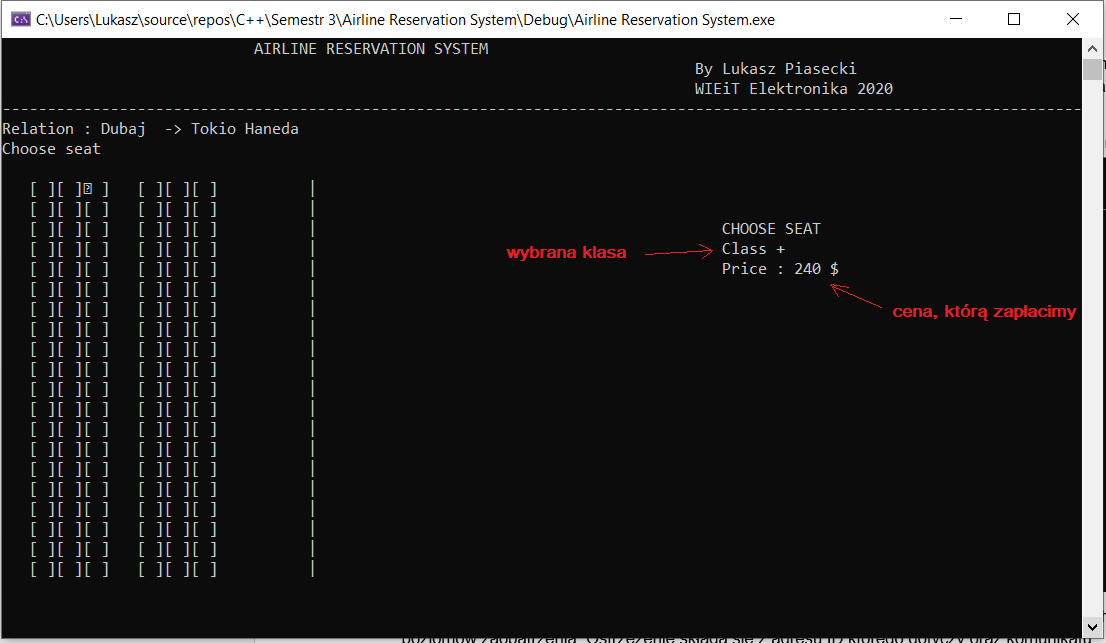
The software is used to reserve seats on the plane. The program may be used on platforms selling airline tickets and tickets by other means of communication.

# Instrukcja użytkownika

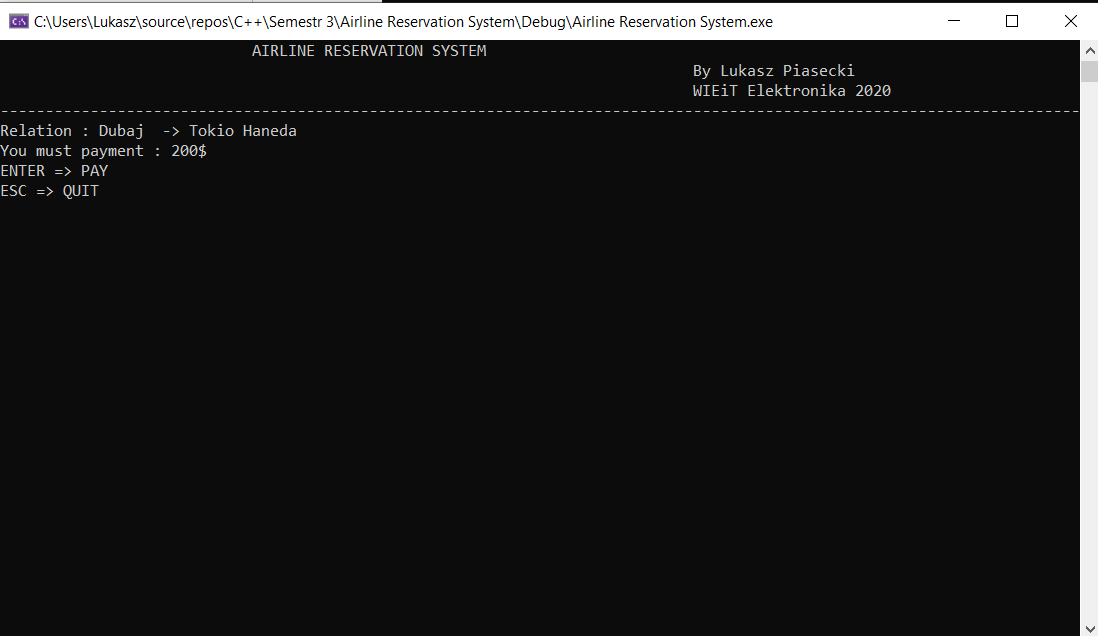
* Pierwszym czynnością, którą musimy wykonać po uruchomieniu programu jest wybranie lotniska z którego startujemy i na którym lądujemy. Sterowanie jest bardzo intuicyjne. Kursor przesuwamy za pomocą strzałek, a swój wybór zatwierdzamy klawiszem Enter.



* Kolejną czynnością, którą musimy wykonać jest wybór miejsca. Za pomocą strzałek wybieramy miejsce, które chcemy zarezerwować. Miejsca mają różną cenę ze względu na swoje ułożenie. Wybór akceptujemy klawiszem Enter. Możemy wrócić do Menu Głównego wciskając klawisz Esc.



* Ostatnim etapem jest zaakceptowanie rezerwacji oraz zapłata.





# Kompilacja

Do uruchomienia programu wystarczy standardowa kompilacja. Program bezproblemowo uruchamia się w systemie Windows.

# Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

* *Plane.h*, *Plane.cpp*– deklaracja oraz implementacja klasy Plane,
* *File\_Services.h*, *File\_Services.cpp*– deklaracja oraz implementacja klasy File\_Services,
* Airline Reservation System*.cpp –* klasa główna programu,
* *Menu.h*, *Menu.cpp*– deklaracja oraz implementacja klasy Menu,
* *Global\_Function.h*, *Global\_Function.cpp*– deklaracja oraz implementacja funkcji globalnych,

# Zależności

brak

# Opis klas

W projekcie utworzono następujące klasy publiczne:

* Plane - reprezentuje samolot
* Plane(int a); - Inicjalizacja klasy
* int price{}; - zwraca cenę miejsca
* int Choose\_seat(string air\_a, string air\_b); - funkcja odpowiedzialna za wybieranie rezerwowanego miejsca
* void Change\_Seat\_Status(int seat); - funkcja zmieniająca status zarezerwowanego miejsca
* Menu – klasa odpowiedzialna za obsługę wszystkich okien menu.
* Menu(int list\_size); - Inicjalizacja klasy
* void SetStock(int stock\_min, int stock\_max); - funkcja odpowiedzialna za przypisanie zmiennym prywatnym wartości.
* int GetStock\_min();- zwraca wartość stock\_min
* int GetStock\_max();- zwraca wartość stock\_max
* int MainMenu(map<int, string> airport\_list, string tekst); - funkcja odpowiedzialna za obsługę głównego menu
* File\_Services – klasa odpowiedzialna za odczyt danych z pliku
* File\_Services(string patch); - Inicjalizacja klasy
* map<int, string> Read\_File(); - zwraca odczytane dane z pliku
* void Write\_File(map<int, string> tekst ); - zapisuje dane do pliku
* Global\_Function – klasa zawierająca funkcje statyczne
  + static void Clean(); - funkcja odpowiedzialna za czyszczenie ekranu i rysowanie tytułu.
  + static void GoToxy(const int x, const int y); - funkcja odpowiedzialna za przesuwanie kursora we wskazane miejsce

# Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

* airports.txt– plik zawierający nazwy portów lotniczych.
  + każda kolejna linijka to osobne lotnisko.
* seats.txt– plik zawierający nazwy portów lotniczych.
  + każda kolejna linijka to status osobnego miejsca.

# Dalszy rozwój i ulepszenia

* Utworzenie dla programu interfejsu graficznego znacznie poprawiłoby nawigację.
* Dodanie innych samolotów/lotnisk
* Modyfikacja programu, aby tworzyć nowe konta użytkownika,
* Dodanie możliwości zakupu przewiezienia dodatkowego bagażu
* Dodanie innych standardowych elementów używanych przy rezerwacji biletów

# Inne

Brak.