Projekt Szymon Dyszewski Podstawy Informatyki i Programowania

Cel projektu

Projekt obejmował zaprojektowanie programu, który pozwoli na automatyczne pokładowanie pasażerów samolotu oraz wydawanie kart pokładowych. Komunikacja użytkownika (pasażera) z programem odbywać się ma za pomocą menu, którego polecenia są wyświetlane na ekranie konsoli.

Program miał obejmować:

- Rejestracje pasażera do lotu.
- Weryfikację wprowadzonych danych (zakładamy, że dane pasażerów o zakupionych biletach będą przechowywane w plikach tekstowych imitujących działanie bazy danych).
- Dopuszczalność zmiany miejsca w samolocie.
- Dopuszczalność zmiany klasy lotu z ekonomicznej na biznesową.
- Drukowanie karty pokładowej.
- Zamówienie pomocy asystenta.
- Sprawdzenie informacji o bramce odlotu.
- Sprawdzenie parametrów lotu (typu samolotu, l. miejsc, informacji o przewoźniku).

Opis działania programu

Program składa się z trzech plików:

- Main.py
- Database.py
- Classes.py

Korzysta z dwóch baz danych:

- Passengers.json – zawiera dane pasażerów i ich przelotów.
- Planes.json – zawiera wszystkie niezbędne informacje o samolotach.

Plik main.py obsługuje główne funkcje programu i przeprowadza dialog z użytkownikiem (pasażerem). Po pobraniu danych od użytkownika loguje się na jego konto za pomocą funkcji login bądź w przypadku braku pasażera w bazie najpierw tworzy nowy obiekt klasy Passenger (pasażera) za pomocą funkcji add_passenger następnie się logując. Zawierają się w nim funkcje:

- menu_action – jest to główna funkcja programu, która umożliwia podejmowanie akcji przez użytkownika i uruchamianie poszczególnych funkcji. Konwertuje ona również predefiniowane bazy danych w obiekty poszczególnych klas.
- add_reservation – umożliwia dodanie do konta pasażera nowego lotu z uwzględnieniem samolotu i miejsca, jego dostępności oraz klasy lotu. Ponadto aktualizuje ona wskazany w procesie przez użytkownika obiekt klasy Plane aktualizując status danego miejsca.
- delete_resevation – umożliwia całkowite usunięcie lotu z konta pasażera oraz aktualizację miejsc w maszynie.
- change_resevation – funkcja ta odpowiada za zmienianie zarezerwowanego miejsca w konkretnym locie pasażera. Aktualizując zarówno lot pasażera jak i stan samego samolotu.
- call_advisor – wyświetla numer na infolinię obsługującą system.
- print_ticket – przekuje klasie PDF informacje o pasażerze i jego locie by umożliwić metodom generację biletu.
- check_all_flights – przekazuje do menu informacje o wszystkich dostępnych lotach i ich szczegółach takich jak: nazwa samolotu, ilość dostępnych miejsc, bramka odlotu, czy przewoźnik.
- check_your_flights – wyświetla szczegółowe informacje dotyczące lotu pasażera: zarezerwowane miejsce, jego klasę, nazwę samolotu, bramkę odlotu oraz przewoźnika.

Plik database.py odpowiada za odczyt baz danych z plików json i utworzenie na ich podstawie obiektów klas Passenger i Plane.

Plik classes.py zawiera definicje trzech klas: Passenger, Plane oraz PDF. Dwie pierwsze służą do obsługi działań użytkownika, zaś ostatnia tworzy pdf biletu pasażera.

Instrukcja obsługi

Użytkownik po uruchomieniu programu zostanie poproszony o swoje imię i nazwisko, te stanowią swojego rodzaju „token" niezbędny do zalogowania. Następnie pojawia się menu interakcyjne, gdzie użytkownik ma do dyspozycji osiem funkcjonalności programu. Może on zdecydować się na wybranie:

- 1 – dodanie nowej rezerwacji do konta pasażera. Po wpisaniu nazwy interesującego go samolotu należy podać konkretne miejsce znajdujące się na liście dostępnych by rezerwacja została pomyślnie ukończona. W razie podania niepoprawnych danych użytkownik zostanie poinformowany.
- 2 – usunięcie rezerwacji, jest to funkcja niezwykle zbliżona do poprzedniej, z tą różnicą, że użytkownik widząc dostępne samoloty będzie miał do wyboru jedynie te w których są już zarezerwowane przez niego miejsca. Efektem finalnym będzie usunięcie rezerwacji wskazanej przez pasażera i zwolnienie miejsca.
- 3 – modyfikacja rezerwacji – funkcja pozwalająca na usunięcie rezerwacji z konta pasażera dokładnie jak w funkcji nr 2, oraz podanie nowego numeru miejsca by zmienić nr zarezerwowanego miejsca w obrębie tego samego samolotu.
- 4 – sprawdzenie szczegółów lotów po wybraniu wyświetli wszystkie loty dostępne w systemie i informacje o nich.
- 5 – sprawdzenie szczegółów swoich lotów wyświetli wszystkie rezerwacje posiadane przez pasażera i wszystkie szczegóły o nich.
- 6 – generowanie biletu – tworzy plik w lokalizacji programu, który jest graficznym przedstawieniem informacji o wybranym przez pasażera locie.
- 7 – kontakt z konsultantem ukazuje użytkownikowi numer infolinii.
- 8 – wylogowanie użytkownika z programu.

Aby umożliwić programowi działanie należy utworzyć dwa pliki: Passengers.json oraz Planes.json. Pierwszy z nich może pozostać pusty, gdyż program na bieżąco może utworzyć nowe

konto dla pasażera, natomiast drugi musi zostać uzupełniony o rekordy dostępnych samolotów i miejsc.

Podsumowanie

Podsumowując udało mi się zrealizować wszystkie założenia (cele) projektu, w najprostszy z mojego punktu widzenia sposób. Program rejestruje pasażera i sprawdza możliwość udzielenia mu żądanej rezerwacji. Można zadać sobie pytanie co, jeśli utworzymy niespójne bazy danych (dwaj różni 0pasażerowie posiadać będą to samo miejsce w jednym locie) przy założeniu, że wszelkie rezerwacje odbywać się będą przez system nie dojdzie do tej sytuacji, gdyż program uniemożliwia wszelkie tego typu błędy.

W ramach dalszego rozwoju projektu (ponad postawione mu cele) warto rozważyć umożliwienie zapisywania obiektów klas w formie jsonów oraz osadzenie całego programu na hostingu, który byłby dostępny dla użytkowników przez stronę internetową umożliwiając rezerwację z każdego miejsca na świecie.

Projekt nauczył mnie tworzenia PDFów w pythonie, nigdy wcześniej nie miałem z tym styczności, ale jak się okazało nie był to ciężki orzech do zgryzienia. Charakterystyczną cechą tego języka jest jego elastyczność i prostota, które bardzo pomogły mi w realizacji. Jeśli chodzi o trudności wspomnieć mógłbym tutaj o aktualizacji baz danych na bieżąco i transformacji obiektów klas w bazę json.

Uważam, że wykonanie projektu jest niezwykle proste, co jest dużą zaletą w większych projektach, jak i rzetelne. Wszystkie możliwości „sabotażu” programu przez użytkownika zostały przeze mnie przewidziane, a program uodporniony. Mam nadzieję, że praca zostanie dobrze oceniona.