

**T.C.**

**MARMARA UNIVERSITY**

**FACULTY of ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

CSE4074 Computer Networks

Socket Programming Assignment

Report

Group Members

Burak ŞENKUŞ – 150115027

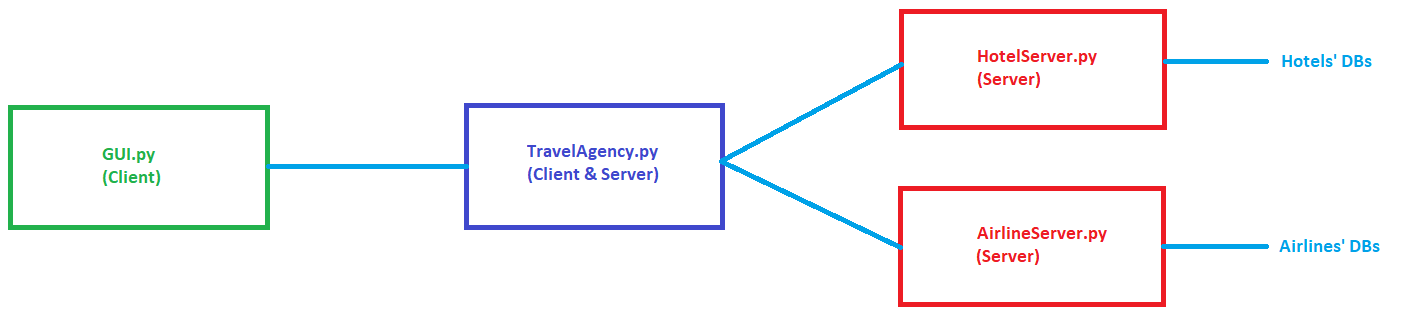
Harun BÜYÜKTEPE – 150115020

Deadline

22.12.2019

In this assignment, we have implemented a trip reservation system. Customer opens program’s GUI, types arrival and departure dates and number of travelers, selects preferred hotel and airline and finally clicks to reserve button. If preferred hotel and airline are available for given inputs, reservation completed successfully and corresponding databases are updated. Any of preferred hotel and airline is not available for given inputs, alternative hotels or airlines are listed to the customer and prompted to select one of them. If customer does not select one of them, reservation process terminated.

Program has been coded with Python and GUI has been created with Tkinter library. There are 4 main Python files and JSON files for airlines’ and hotels’ databases.



**GUI.py**

This file includes graphical user interface and client implementation.

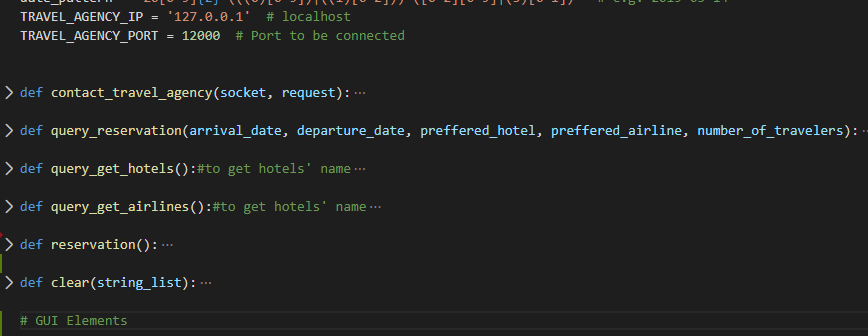


Figure 1: General structure of GUI.py

contanct\_travel\_agency function to send request to Travel Agency.

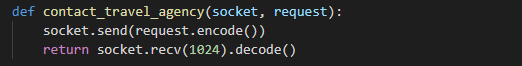


Figure 2: Function to connect Travel Agency

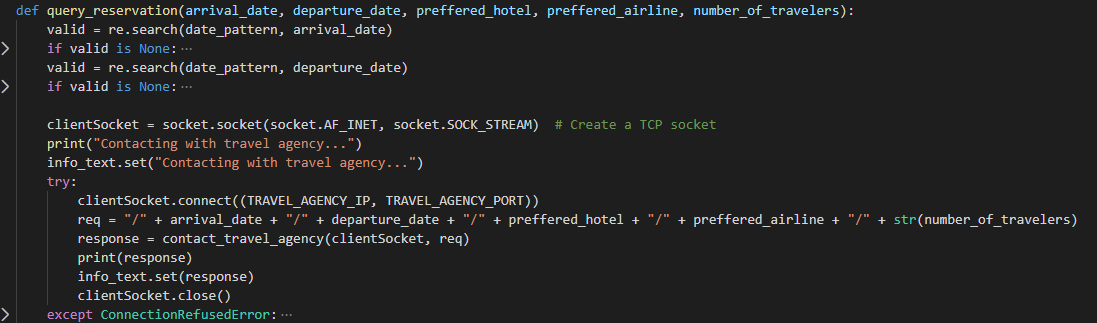


Figure 3: Function to prepare request message

This function examines the GUI’s date inputs. If any wrong format was not occurred, request will be sent. Format of the request message should be “/arrival date/departure date/preffered hotel/preffered airline/number of travelers”. Also query reservation function will get response message from Travel Agency. If message is suitable for each server; GUI display reservation completed successfully, If message is not suitable for any of them; GUI display advise a new reservation to traveler.

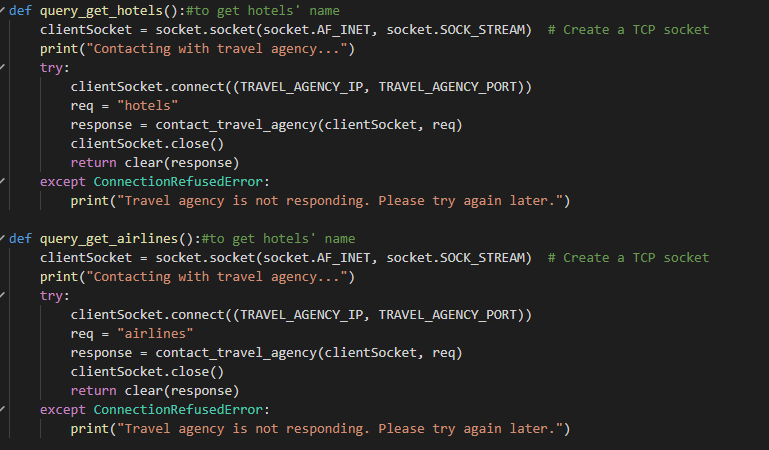


Figure 4: Function to get hotels and airlines

These functions get all hotels and all airlines with sending to request message to Travel Agency.

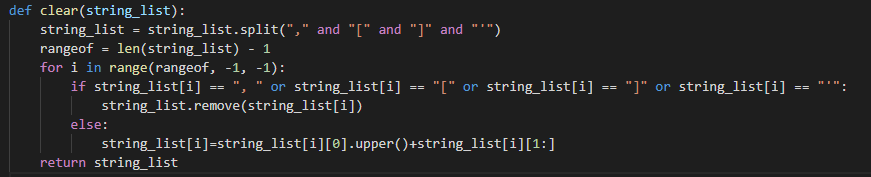


Figure 5: Function to fix response message

Response message comes in string format. We fix the string and convert to list to display hotels and airlines on GUI.

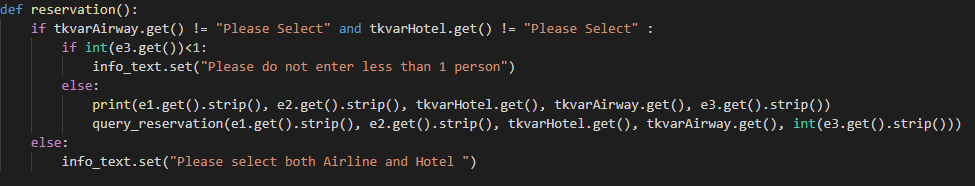


Figure 6: Function to activate query reservation

This function is triggered by the button. It controls hotels and airlines section inside.

**TravelAgency.py**

TravelAgency is actually a buffer between the hotel, airline and GUI(Client) server.

Figure 7: General structure of TravelAgency.py

Contanct\_with\_port function can use to communicate with hotel server and airline server. We define 2 different port to differentiate them.



Figure 8: Ports

Travel Agency always waits for to communicate with GUI. When request comes from GUI, Travel Agency handles the request, then sends proper request to hotel and airline servers.

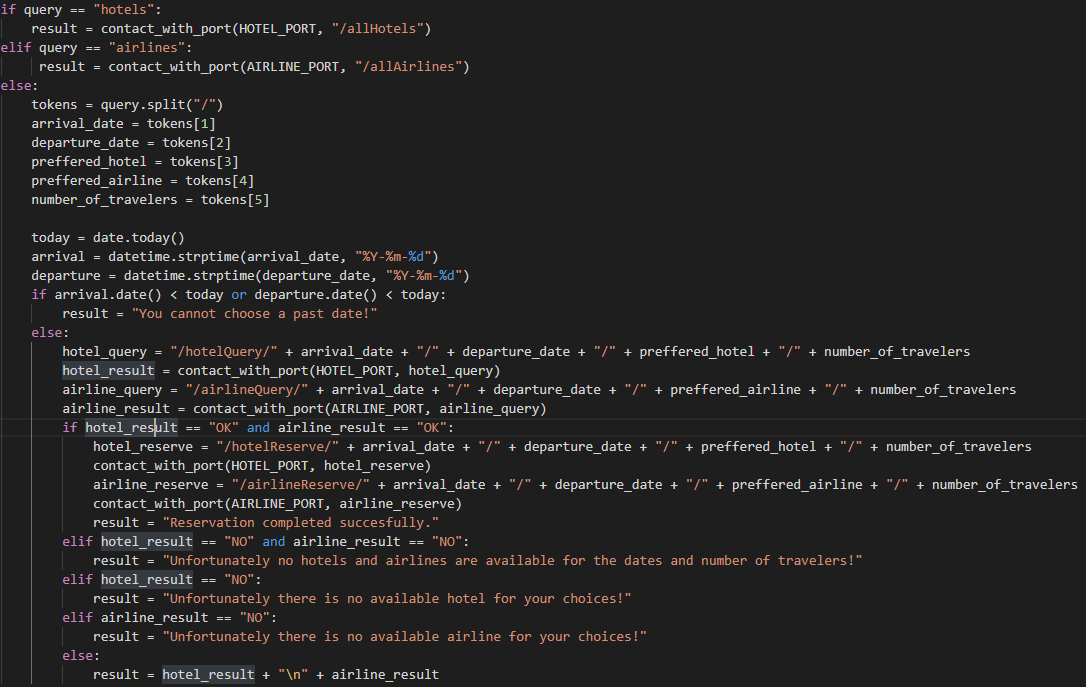


Figure 9: Request extractor

“/allhotels” and “/allairlines” requests handled and creates new requests for getting all hotels and airlines from hotel and airline servers.

“/arrival date/departure date/preffered hotel/preffered airline/number of travelers” request handled and creates new requests for ask to hotels and airlines.

New request messages will send hotel and airline server. After these, Travel Agency return response message to GUI(Client).

**HotelServer.py**

This file is responsible for DB files of all hotels. When a request comes from Travel Agency by using port 33333, Hotel Server gets this request and interacts with hotels’ DB files and returns a result.



Figure 10: General structure of HotelServer.py

There are 3 main GET requests that Hotel Server can respond.

* /allHotels
* /hotelQuery/arrivalDate/departureDate/preferredHotel/numberOfTravelers
* /hotelReserve/arrivalDate/departureDate/preferredHotel/numberOfTravelers



Figure 11: Main requests of HotelServer.py

If preferred hotel is not available for given inputs, find\_all\_hotels\_by\_dates() function is invoked and finds all alternative hotels for given inputs.



Figure 12: Function to find alternative hotels

**AirlineServer.py**

This file is responsible for DB files of all airlines. When a request comes from Travel Agency by using port 44444, Airline Server gets this request and interacts with airlines’ DB files and returns a result.

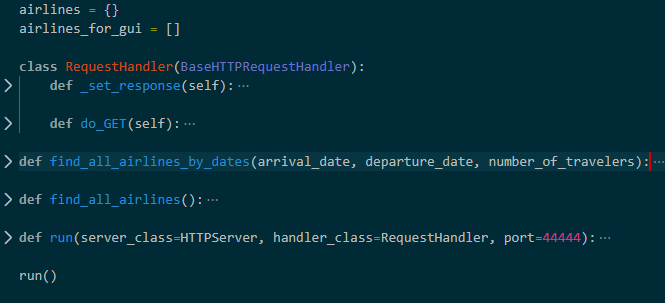


Figure 13: General structure of AirlineServer.py

There are 3 main GET requests that Airline Server can respond.

* /allAirlines
* /airlineQuery/arrivalDate/departureDate/preferredAirline/numberOfTravelers
* /airlineReserve/arrivalDate/departureDate/preferredAirline/numberOfTravelers

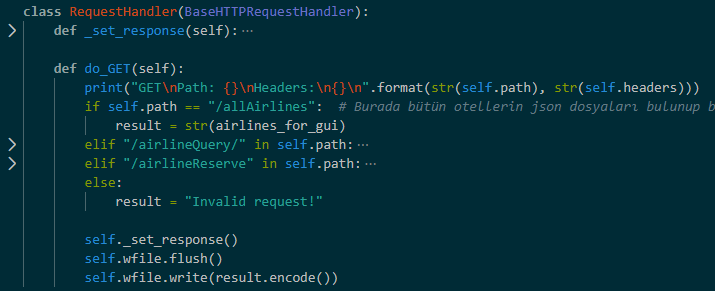


Figure 14: Main requests of AirlineServer.py

If preferred airline is not available for given inputs, find\_all\_airlines\_by\_dates() function is invoked and finds all alternative airlines for given inputs.

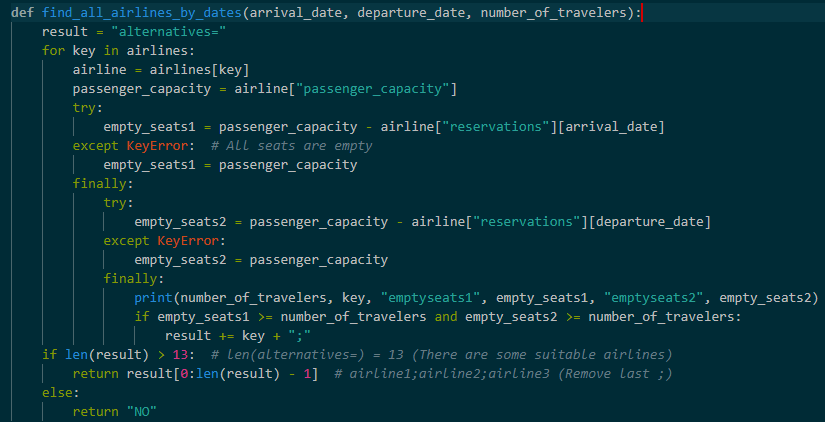


Figure 15: Function to find alternative airlines

**Database Files**

Database files for all hotels and airlines are in JSON format. Names of airline files start with a\_ prefix and hotel files start with h\_ prefix. Each hotel and airline has its own DB file. In the file, max capacity of daily customer and up-to-date daily customer counts are specified. The availability for each hotel or airline on any given day is calculated by subtracting the number of customers available on that day from the maximum capacity. When a new database file added to the directory, GUI automatically finds it on next startup.

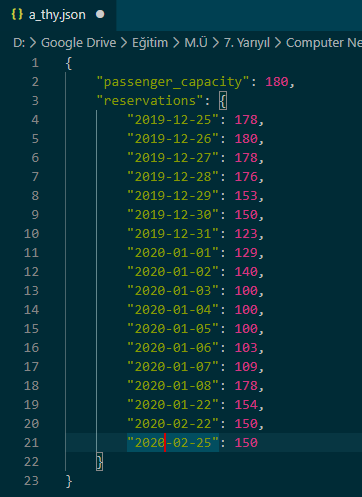


Figure 16: Sample DB file structure