

University of Bahrain

College of Information Technology

Department of Computer Engineering

ITNE352

Multithreaded Flight Arrival Client/Server Information System

ITNE352 Project

Group: B4

202102871 | Muna Nawaf Burshaid | Section 2

202108408 | Noor Waleed Almahmood | Section 2

Table of Contents

[Introduction: 2](#_Toc154871794)

[Client: 2](#_Toc154871795)

[Server: 3](#_Toc154871796)

[API (Application Programming Interface): 3](#_Toc154871797)

[Multithreaded Connection: 4](#_Toc154871798)

[Main Elements of the Server: 4](#_Toc154871799)

[Requests: 5](#_Toc154871800)

[Arrived Flights: 5](#_Toc154871801)

[Delayed Flights: 6](#_Toc154871802)

[All Flights Coming from A Specific City: 6](#_Toc154871803)

[Details of a Particular Flight: 6](#_Toc154871804)

[Quit: 6](#_Toc154871805)

[GUI: 7](#_Toc154871806)

[Evaluation while working with python: 8](#_Toc154871807)

[Pros: 8](#_Toc154871808)

[Cons: 8](#_Toc154871809)

[Extension that would like to see made: 9](#_Toc154871810)

[Evaluation on the Design: 9](#_Toc154871811)

[Limitation: 10](#_Toc154871812)

[Conclusion: 11](#_Toc154871813)

[References 12](#_Toc154871814)

# Introduction:

Python programming language is widely used in many countries and companies in the world because it is providing important features for its users such as offering products for both windows and Linux, different packages, less syntax complexity than other programming languages and allow to build GUI for codes.

Through this project, we implemented what we just learned from the course as we wrote the codes in python language. We created a server that accept TCP connections and handle multiple requests from different clients by multithreading. Moreover, we created a client code that sends requests to the server and get the response from it. As a feature, we added the graphical user interface to client’s code. These codes are implemented to have the necessary information about the flights in the communication between the server and the client. The flight information is coming from the Aviation Stack API, and it is stored and retrieved from a JSON file.

# Client:

In the client, we need to be sure that the ICAO code was entered correctly to execute the client’s code, else, the information will not be delivered. First, we enter the username do that the request is sent by the client’s name and the server knows who is sending this request. The client connects with the server by the username, then they can choose their request.

# Server:

In this server, it starts by listening to a client connection in which it gets the username and ICAO of our client and their request of information. The server connects to the AviationStack.com to acquire the needed information for our client, and the data is saved into the “Group\_B4.json” file. Then the server sends this data back to the client for them to read, this keeps happening until the client clicks the quit button to disconnect.

This server can handle multiple clients simultaneously with the use of threading.

The server displays two things:

1. The acceptance of the connection with the client’s name
2. The disconnection of the client with their name.

# API (Application Programming Interface):

API is a server that is used to get and send data that is used in the code, but with API there must be a request in order to get the data. Since we have the API in the software, we get the data from aviationstack.com, and in this way, we make it easy for our program to get, store, and sort through the information. All information is stored in the JSON file.

# Multithreaded Connection:

This basically means that the connection is made up of many threads, these threads in a process can communicate and share information faster than if they were distinct processes due to the fact that they are sharing the data space with the main thread. In this software, this multithreaded connection was added. Whenever a client sends a request, a thread is being started and enables the connection between the client and server. Since we need to handle many clients simultaneously, we need to handle many threads in order to handle many clients, but this might affect the performance of the software.

# Main Elements of the Server:

-Listener Class: Any objects that want to sign up to receive new message alerts utilize an abstract base class named Listener. The Listener can be invoked with a new message and a default listener.

Its purpose is to listen for the incoming connections and creates the threads for each client.

-Thread Class: The threading allows multiple process to happen within a single process, which means that we are able to handle many clients at the same time. The purpose of having a thread class is for the management and generation of threads. After the thread is created, it has to fulfil its purpose, and every thread has a unique name.

Its purpose is to allow multiple processing, which in turn allows us to handle multiple clients.

-While-Read/Write Loop (Server-side): In the primary loop, the server accepts requests continually while creating a thread for each client, reading and processing the client requests and sending the clients back their requested data.

The main purpose is to enable continuous communication between the server and the client while accommodating to the clients’ requests with the relevant flight information.

-Removing Dead Connections: Whenever the client disconnects from the server, the server removes this disconnected client.

Its purpose is to have an updated active clients list and remove disconnected clients.

# Requests:

The client can request five different types of requests to the server in order to get the data

# Arrived Flights:

When the client chooses the request “Arrived Flights”, it will display the data of the arrived flights. If in case of no data found, it will display a message that there are no flights found.

The server sends the flight code (IATA), departure airport, arrival time, arrival terminal, and arrival gate.

# Delayed Flights:

When the client chooses the request “Delayed Flights”, it will display the delayed fights.

The server sends the flight code (IATA), departure airport, departure time, estimated time arrival, delay, terminal, and gate.

# All Flights Coming from A Specific City:

When the client chooses the request “Flights from a City”, it will request a city IATA code from the client, and then displays the flights from this specified city.

The server sends the flight code (IATA), departure airport, departure time, estimated time arrival, departure gate, arrival gate, and status.

# Details of a Particular Flight:

When the client chooses the request “Flight Details”, it will request the flight code (IATA) and then displays the details of this flight.

The server sends the flight code (IATA), the departure airport, gate, and terminal, the arrival airport, gate, and terminal, status, scheduled departure time, and scheduled arrival time.

# Quit:

When the client clicks in the Quit button, the client will be disconnected.

The server will send a message about the client disconnecting by name.

# GUI:

In the client code, we assigned the GUI in order to make communication easier between the client and server in a way that is visualized and with a single click the client gets what information they need.

# Evaluation while working with python:

Python is a programming language that is widely accepted in software development. Throughout this course we have been using python and learning many commands and ways to interact with it.

# Pros:

Two main advantages that Python holds over any other languages is:

1. Simplicity and ease of coding: the python language is very simple compared to other languages that has a specific way of writing the code, but with python, we can simply get to the point, run the code, and simply fix it.
2. Compatibility: The python language can be used through many platforms such as: Linux, Windows, and MacOS.

# Cons:

Two disadvantages of Python include:

1. Being Slow at Runtime: since python is a high-level language, it is slow compared to C/C++ or Java. The execution of python takes place with the help of an interpreter and executes line by line which makes it slow down.
2. Runtime Errors: one of the main issues with python is that it needs more testing due to the fact that it has errors that show up at runtime which is kind of frustrating.

# Extension that would like to see made:

I would like to see an extension that scans and detect any error that might happen during runtime, this would help programmers in finding their errors and fix them easily rather than running their code for the millionth time hoping that it will work smoothly.

# Evaluation on the Design:

-Message formats: In both Server and Client we used simple string messages to communicate with both ends that are encoded and decoded by ascii.

-Concurrent Connection Requests: The threading is there to make it able for the Server to accept and handle multiple Clients simultaneously.

-Handling Client Failures: In any case of an exception occurring while communicating with a client, the Server takes action and disconnects the Client.

-Handling Chat Server Failures: In case of Server failures, the Client shouldn’t be responsible in handling this failure, so the Client wouldn’t receive proper notice about the situation. Keeping that into consideration, in case of Server termination, there would be a notice about it.

-Storing Data: All fight information is stored and retrieved from the JSON file named “Group\_B4.json”

# Limitation:

-System shows data for one airport: Since this is an optional feature that could be worked on, it will definitely come in handy since the client can choose which airport to retrieve the data from.

-Using free API: As we used free API, we had to do limited requests and have limited features to be used in our project’s code.

-Traffic building up on the Server: While handling many Clients at once seems great, it does have some drawbacks and that is causing the Server to slow down or in the worst-case crash.

# Conclusion:

With that being said, throughout this project we have been testing our knowledge about python while working and fixing this framework, we faced many errors and fixed them to the best of our abilities learning more and more while correcting our mistakes all for the sake of perfecting our framework. In this code we included multithreading, sockets, how to accept connections from Clients, understanding about API and working with it, and overall bring our programmer minds to the test.

# References

*Disadvantages of Python*. (2023, Nov 02). Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/disadvantages-of-python/

Lutz, M. (2001). *Programming Python, Second Edition.* O'Reilly & Associates, Inc.

*Top Advantages of Python Over Other Programming Languages*. (2023, November 14). Retrieved from edoxi.com: https://www.edoxi.com/studyhub-detail/advantages-of-python-over-other-programming-languages