

## Problem Statement

In this assignment, I needed to develop a city plate number guess program between a server and multiple clients (sequentially). The server should read the Excel file consisting of the city and its corresponding plate number and wait for a client to connect and guess the random city's (which is chosen by the server) plate number. If the user's guess from the client side is not correct, the server should respond with the name of the city that the user has guessed. If the user's guess from the client side is correct, the server should respond to the client, and the client should notify the user and close the connection with the server. After the client closes its connection, the server should wait for another client to connect.

## Algorithm Design

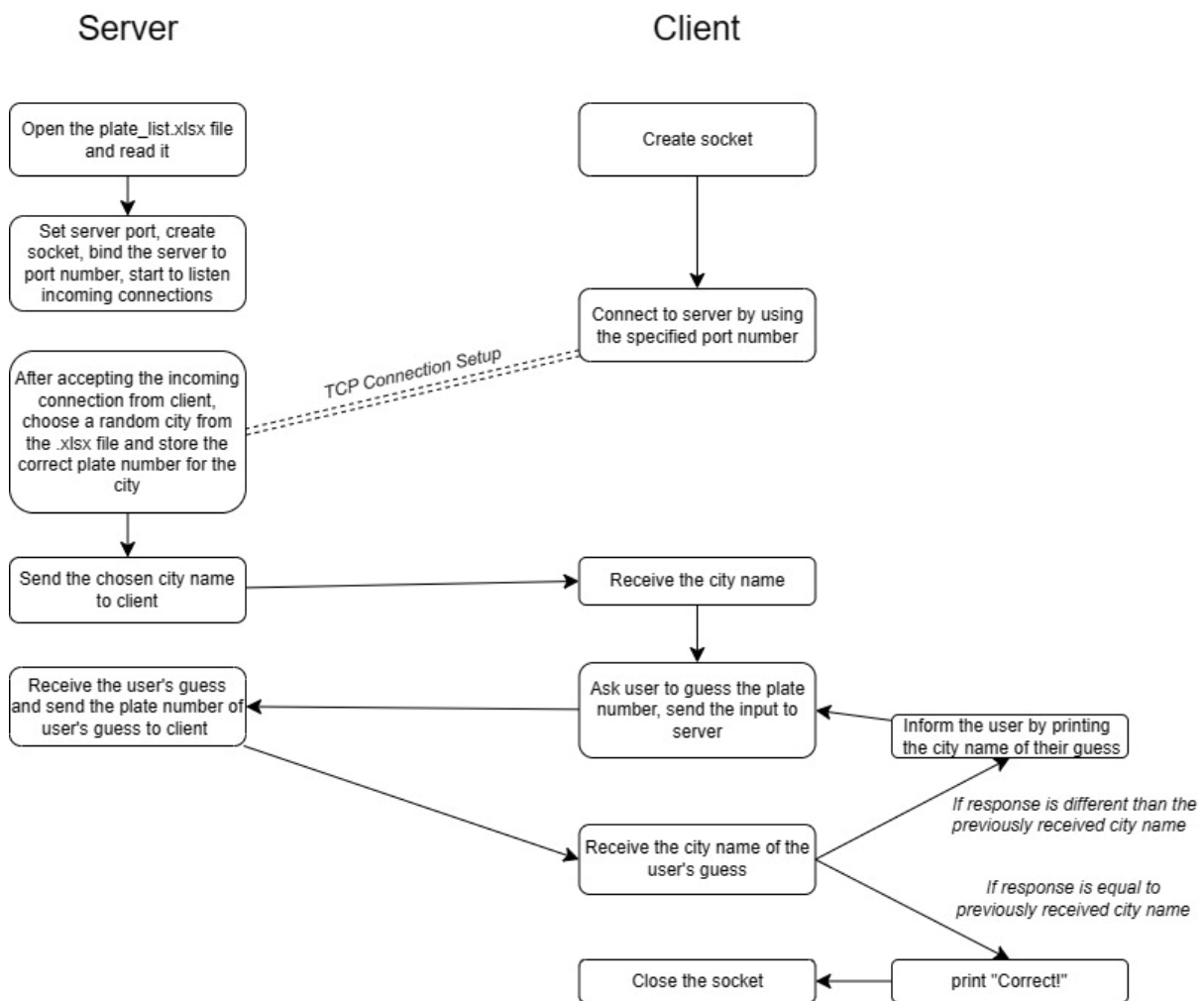


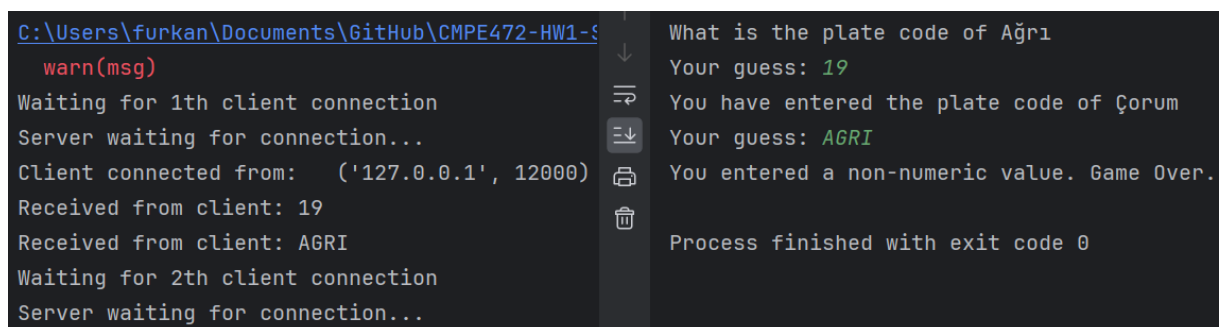
Figure 1

As can be seen from Figure 1, the server should read the Excel file, create the server port, and socket, and bind the server to a specified port number. After the connection has been set, the server should pick a city name from the Excel file randomly and send it to the client. The client should ask the user to guess the plate number of the city that has been received from the server and send the user's input to the server. The server should find the city name corresponding to the plate number the client sends and send that city name to the client. Then, the client should check if the received city name is equal to the one that it gets after setting up the connection and notify the user about the comparison.

## Input Validation and Stopping Conditions

The user's guess must go through a validation to ensure that the user entered an integer value. When the program gets a string value instead of an integer value, it notifies the user as "You entered a non-numeric value. Game Over." and closes its socket (Figure 2). Also, the user must enter the input between 1 and 81 because there are no city names assigned exceeding this range. When the program gets a value exceeding the specified range, it notifies the user as "Number exceeds the range. Game Over." and closes its socket (Figure 3).

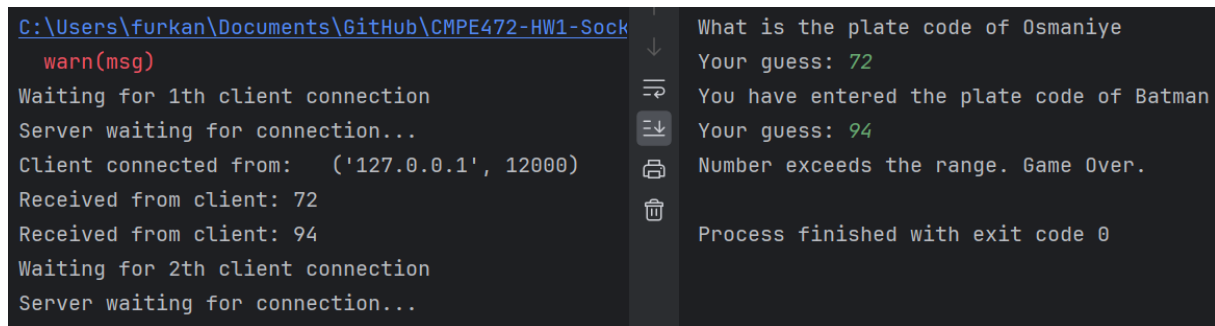
I achieved these validations on the server side after the client sent the user's input to the server. The server checks if the user's guess is not numeric (with `isnumeric()`) and is in range and sends a response if one of the conditions is not satisfied. The server sends "NON\_NUMERIC" when the guess is not numeric and "NOT\_IN\_RANGE" when the guess is not in range. Then the server stops waiting for any more responses from the current client and the client notifies the user about the problem and closes itself.



```
C:\Users\furkan\Documents\GitHub\CMPE472-HW1-9>
warn(msg)
Waiting for 1th client connection
Server waiting for connection...
Client connected from: ('127.0.0.1', 12000)
Received from client: 19
Received from client: AGRI
Waiting for 2th client connection
Server waiting for connection...

What is the plate code of Ağrı
Your guess: 19
You have entered the plate code of Çorum
Your guess: AGRI
You entered a non-numeric value. Game Over.
Process finished with exit code 0
```

Figure 2



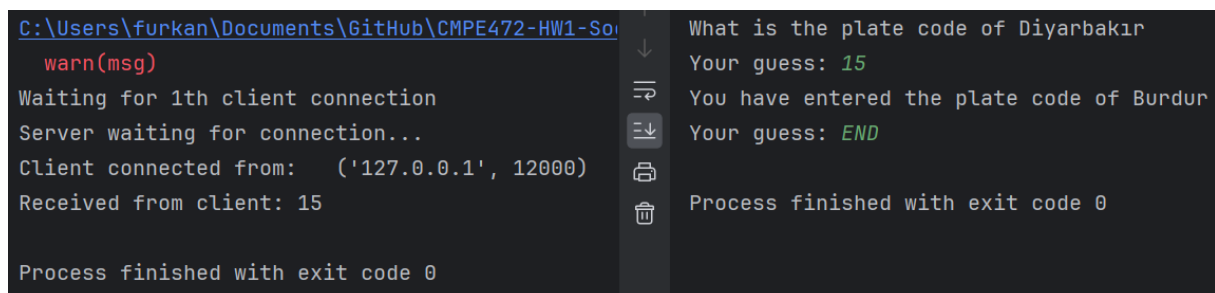
```
C:\Users\furkan\Documents\GitHub\CMPE472-HW1-Sock
warn(msg)
Waiting for 1th client connection
Server waiting for connection...
Client connected from: ('127.0.0.1', 12000)
Received from client: 72
Received from client: 94
Waiting for 2th client connection
Server waiting for connection...

What is the plate code of Osmaniye
Your guess: 72
You have entered the plate code of Batman
Your guess: 94
Number exceeds the range. Game Over.

Process finished with exit code 0
```

Figure 3

To stop both client and server, the user must input “END” when they are asked to guess. After the user enters “END”, the client sends the string “END” and closes its socket. Then, the server checks if the response from the client is “END” instead of a plate number guess and breaks the outer infinite loop to close its socket. A sample run for this condition can be seen in Figure 3.



```
C:\Users\furkan\Documents\GitHub\CMPE472-HW1-Sock
warn(msg)
Waiting for 1th client connection
Server waiting for connection...
Client connected from: ('127.0.0.1', 12000)
Received from client: 15
Received from client: END
Process finished with exit code 0

What is the plate code of Diyarbakır
Your guess: 15
You have entered the plate code of Burdur
Your guess: END

Process finished with exit code 0
```

Figure 4

## Sample Runs

Stopping conditions and invalid inputs are covered in Figures 2, 3, and 4. Here are some of the successful runs.

```
C:\Users\furkan\Documents\GitHub\CMPE472-HW1-SocketProgra
warn(msg)
Waiting for 1th client connection
Server waiting for connection...
Client connected from: ('127.0.0.1', 12000)
Received from client: 12
Received from client: 27
Received from client: 23
Received from client: 45
Received from client: 76
Received from client:
Waiting for 2th client connection
Server waiting for connection...

What is the plate code of Iğdır
Your guess: 12
You have entered the plate code of Bingöl
Your guess: 27
You have entered the plate code of Gaziantep
Your guess: 23
You have entered the plate code of Elazığ
Your guess: 45
You have entered the plate code of Manisa
Your guess: 76
Correct!

Process finished with exit code 0
```

```
Waiting for 1th client connection
Server waiting for connection...
Client connected from: ('127.0.0.1', 12000)
Received from client: 12
Received from client: 27
Received from client: 23
Received from client: 45
Received from client: 76
Received from client:
Waiting for 2th client connection
Server waiting for connection...
Client connected from: ('127.0.0.1', 12000)
Received from client: 45
Received from client: 50
Received from client: 54
Received from client: 58
Received from client:
Waiting for 3th client connection
Server waiting for connection...

C:\Users\furkan\Documents\GitHub\CMPE472-HW1
What is the plate code of Sivas
Your guess: 45
You have entered the plate code of Manisa
Your guess: 50
You have entered the plate code of Nevşehir
Your guess: 54
You have entered the plate code of Sakarya
Your guess: 58
Correct!

Process finished with exit code 0
```