MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG445 Software Development with Scripting Languages

Assignment 2: A Simple Application for an Apartment Rental Agency in Northern Cyprus

This assignment aims to help you practice multithreading and concurrent programming, network programming, and graphical user interface components in Python. On the successful completion of this assignment, you will also practice Python basics. Your main task in this assignment is to develop a client-server application based on Transmission Control Protocol (TCP) for an apartment rental agency in the northern part of Cyprus. The server should be able to communicate with multiple clients at the same time as illustrated in Figure 1.

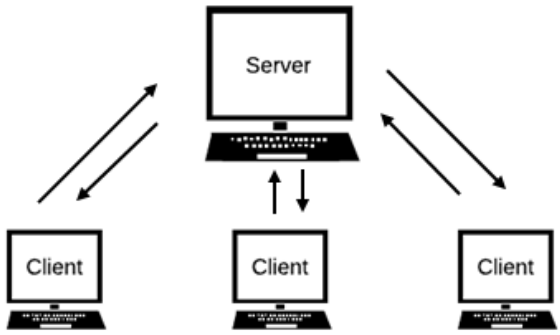

Figure 1: Client-Server Model

Overview

In this assignment, you will develop an application for an apartment rental agency that has multiple employees and managers. It will be a simple client-server application. The main objective of this application is to allow the employees to retrieve the details of the apartments from the server and to make reservations for the customers. This application should also allow the managers to retrieve certain statistics about the apartments and the reservations. The server should manage three text files: **users.txt**, **aparments.txt**, and **reservations.txt**.

1. The **users.txt** file will keep track of the details of the employees and managers as follows (username;password;role):
```
employee1;e123;employee
employee2;k343;employee
manager1;7684;manager
```

2. The **apartments.txt** file will keep track of the details of the apartments as follows (apartment code;street;city;postcode;size;the number of bedrooms):
```
apt1;Sulhi Garan Street;Nicosia;99010;80;2
apt2;Ali Yuzbasi Street;Nicosia;99010;120;3
apt3;Bedrettin Demirel Street;Nicosia;99010;90;2
```

3. The **reservations.txt** file will keep track of the details of the reservations as follows (apartment code;customer name;start date;end date; responsible employee username):
```
apt1;James Brown;01/09/2021;01/06/2022;employee1
apt1;Mary Oscar;01/06/2021;30/08/2021;employee1
apt3;John William;01/09/2021;01/01/2022;employee2
```

After logging in to the application, it should allow the employees to enter an apartment code with a possible start and end dates for a reservation to retrieve the details of the apartment from the server, and also make a reservation for a customer. The application should also allow the managers to request the following statistics as shown in Table 1.

Table 1. Statistics Reports

Code	Statistics
1	Which employee makes the highest number of reservations? If there are multiple of them, all of them will be retrieved.
2	Which apartment is the most popular? If there are multiple of them, all of them will be retrieved.
3	How many apartments are currently available?
4	How many apartments have not been reserved yet?

Requirements

The client application will work either as an employee application or a manager application. When the client application is started and the connection is established with the server, the server will send a confirmation message to the client (message: **`connectionsuccess`**). The client will then show the following login screen to take the username and password from the user (See Figure 2).

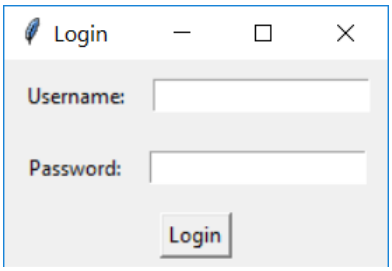

Figure 2: Login screen

Once the client sends the username and password to the server (message format: **`login`**`;username;password`), the server will check the **users.txt** file and send an approval message (message format: **`loginsuccess`**`;username;role`) or rejection message (message format: **`loginfailure`**) back to the client. If the login is not successful, then the following message box will be shown on the client side (See Figure 3). However, if the login is successful, then the appropriate user panel will be shown based on the user role.

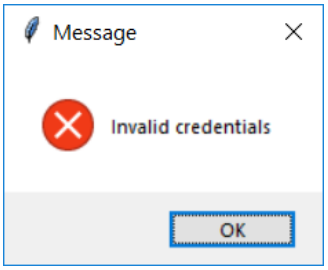

Figure 3: Message box for invalid credentials

Figure 4 shows the user panel for an employee. When an employee wants to take the details of a specific apartment, s/he should enter the apartment code, and the possible start date and end date for a reservation and then click on the "Show" button. When the button is clicked, a request is sent to the server (message format: **`apartment`**`;apartment code;startdate;enddate`) and the server sends the details of the apartment and its availability (message format: **`apartment`**`;apartment code;address;city;postcode;size;the number of bedrooms;availability`[1]), and these details are shown in a message box on the employee side. If the apartment code is not valid, then the

[1] Availability is a Boolean value where True means not occupied and False is occupied for the entered dates.

server will send an invalid apartment code message (message: **`invalidapartmentcode`**), and it should be shown in a message box on the employee's side.

When an employee wants to make a reservation for a particular customer, then s/he should also enter the customer name, in addition to the apartment code, start date and end date, and click on the "Reserve" button. When the button is clicked, a request is sent to the server (message format: **`reservation`**`;`*`apartment code;customer name;start date;end date;responsible employee username`*) and the server makes a reservation by adding the record into the **reservations.txt** file and sends an acknowledgement message for a successful reservation (message format: **`successfulreservation`**), or an unsuccessful reservation (message: **`invalidapartmentcode`** or **`notavailable`**). The acknowledgement message should be shown in a message box on the employee side.

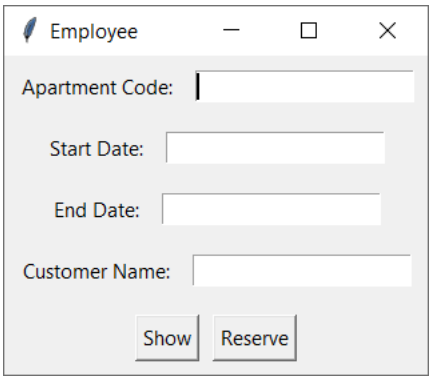

Figure 4: The user panel for an employee

Figure 5 shows the user panel for a manager. When the manager selects one of the available statistics reports, the report code will be sent to the server such as "**`report2`**". The server will then generate and send back (message format: **`report2`**`;`*`answer`* *or* **`report2`**`;`*`answer1;answer2;...`* if there are multiple answers). The answer should be shown in a message box appropriately on the manager's side.

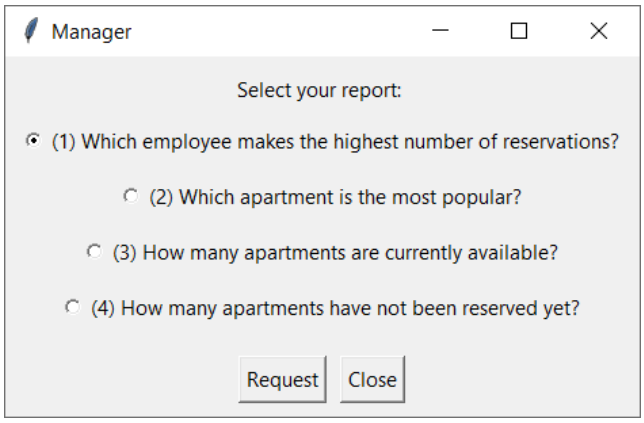

Figure 5: The user panel for a manager

When the "Close" button is clicked, the connection between the client and the server will be terminated. You should consider the RLock thread synchronization technique to deal with any problems caused by attempting to access the shared data at the same file.

Rules

- You need to write your program by using **Python 3.x**.
- You can **only** use all built-in functions and modules.

- You also need to create a file called ReadMe.txt which contains the following items. Please note that **if you do not submit ReadMe.txt, your submission will not be evaluated.**
    - Team members
    - Which version of Python 3.x you have used
    - Which operating system you have used
    - How you have worked as a team, especially how you have divided the tasks among the team members (who was responsible for what?), how you have communicated, how you have tested the program, etc.
- You should name your server as **server.py** and name your client as **client.py**.
- You should not forget to submit your **users.txt**, **apartments.txt**, **reservations.txt** files.
- You need to put all your files into a folder that is named with your student id(s) and submit the compressed version of the folder in the **.zip** format.
- **Only one team member** should submit the assignment.
- **Code quality, modularity, efficiency, maintainability, and appropriate comments** will be part of the grading.

Grading Policy

The assignment will be graded as follows:

Grading Item	**Mark (out of 100)**
Login	10
Apartment Details	15
Reservation	15
Report 1	10
Report 2	10
Report 3	10
Report 4	10
Employee Screen	10
Manager Screen	10