



Date handed out: 14 April 2020, Tuesday

Date submission due: 28 April 2020, Tuesday 23:59

Assignment 2

This assignment aims to help you practice multithreading and concurrent programming, network programming, graphical user interface components and database application development in Python. Your main task in this assignment is to develop a client-server application based on Transmission Control Protocol (TCP) for storing and reporting the details of the visitors of the historical places 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.

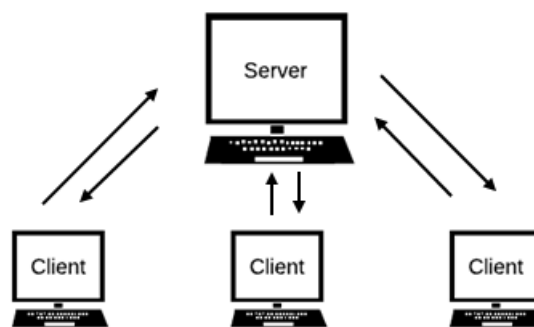


Figure 1: Client-Server Model

Application Overview

In this assignment, you are expected to develop a client-server application to keep track of the details of the visitors of the historical places in different cities in the northern part of Cyprus and to generate necessary reports. Each historical place has a manager who is responsible for sending the details of the visitors to the central server on a daily basis. There are also a number of administrators who can request different kinds of reports from the central server.

Clients

In this application, there will be two types of clients. You are expected to provide a different Python script for each client. For each client, you are expected to design and provide a single graphical user interface to be used to send the username, password and other required details to the server (GUI).

- 1. Managers of Historical Places:** Each historical place has a manager and the manager is responsible for sending the daily statistics to the central server. These statistics should include (1) the total number of visitors, (2) the number of male visitors, (3) the number of female visitors, (4) the number of local visitors, (5) the number of tourists.
- 2. Administrators:** Administrators can request the following reports from the central system. You should use radio buttons in the GUI to allow the administrators to select only one of these reports at a time.
 - (a) The historical place with the maximum number of visitors
 - (b) The city with the maximum number of visitors
 - (c) The number of visitors, the number of male visitors, the number of female visitors and the number of local visitors and the number of tourists for each city

- (d) The number of visitors, the number of male visitors, the number of female visitors and the number of local visitors and the number of tourists for each historical place in a given city
- (e) The number of visitors, the number of male visitors, the number of female visitors and the number of local visitors and the number of tourists for a given historical place on a given date

When the necessary details are retrieved from the server, the details are formatted and then shown in a message box. Figure 2 shows an example of a report for the item (d) above where the given city is Famagusta.

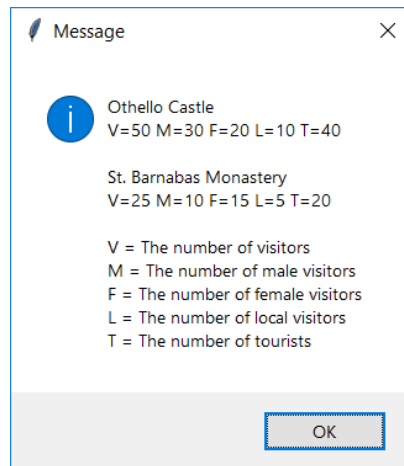


Figure 2: An example of a report of the visitor statistics of the historical places in Famagusta

Server

You are expected to develop a single server and the server will be responsible for communicating with both types of clients. The server does not need to have a graphical user interface (GUI). When these clients communicate with the server, they need to send their usernames and passwords. In case of any invalid username and password, the server will send an error message and the message will be shown in a message box as shown in Figure 3.

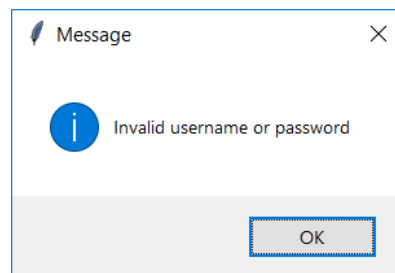


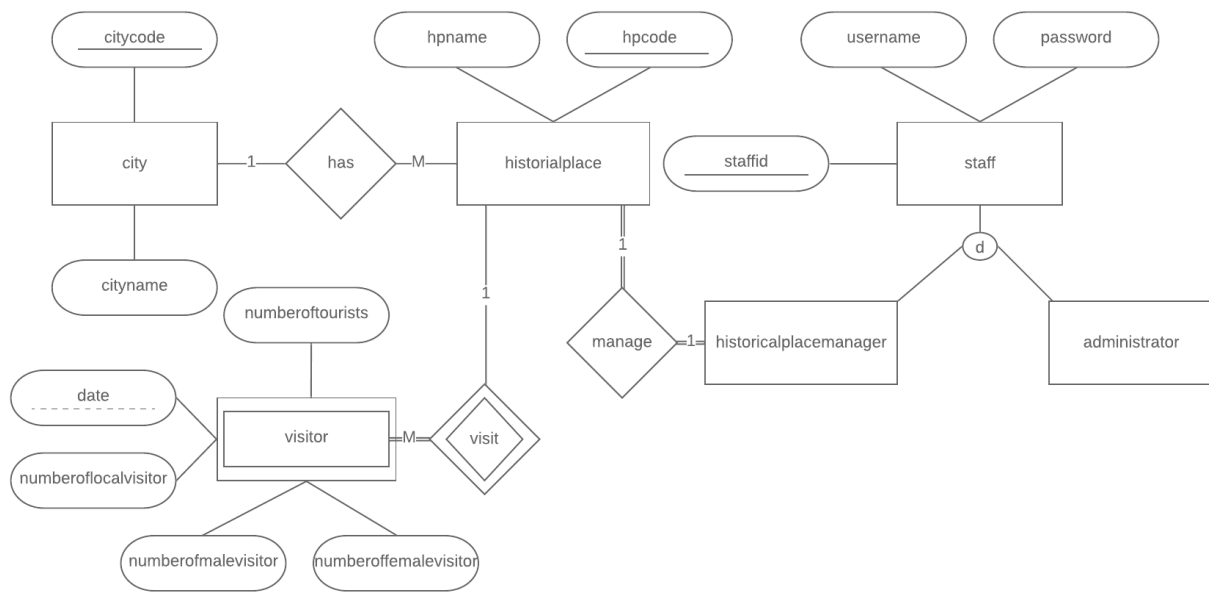
Figure 3: A message box for invalid username and password

1. **Communication with Managers of Historical Places:** When the server receives the daily statistics from the historical place manager, the system should check whether the number of total visitors is equal to the sum of the number of male visitors and the number of female visitors and also the number of total visitors is equal to the sum of the number of local visitors and the number of tourists. If so, the server should send an acknowledgement to the historical place manager. Otherwise, the system will send a message that there is a problem in the statistics and request the correct statistics. The server will be responsible for storing the statistics in the application database. When the historical place manager sends the termination message, then the connection between the server and the historical place manager will be terminated.

2. **Communication with Administrators:** When the server receives the request form the administrator, then it will query the application database and sends the results to the administrator. When the administrator sends the termination message, then the connection between the server and the historical place manager will be terminated.

Database

You need to create a SQLite database for this application based on the following entity-relationship diagram. You are expected provide a separate Python script to create necessary tables and inserts necessary records into these tables.



You will need to add the following details into the tables.

City Code	City Name
1	Gazimagusa
2	Girne
3	Guzelyurt
4	Iskele
5	Lefke
6	Lefkosa

Historical Place Code	Historical Place Name	City Code	Historical Place Manager
1	Othello Castle	1	1001
2	St. Barnabas Monastery	1	1002
3	St. Hilarion Castle	2	1003
4	Bellapais Abbey	2	1004
5	Guzelyurt Museum	3	1005
6	St. Mamas Monastery	3	1006
7	Apostolos Andreas Monastery	4	1007
8	Kantara Castle	4	1008
9	Soli	5	1009
10	Vouni Palace	5	1010
11	St. Sophia Cathedral	6	1011
12	Dervis Pasa Mansion	6	1012

Staff ID	Username	Password	Role
1001	1001HPM	1234	Historical Place Manager
1002	1002HPM	5678	Historical Place Manager
1003	1003HPM	9123	Historical Place Manager
1004	1004HPM	4567	Historical Place Manager
1005	1005HPM	8912	Historical Place Manager
1006	1006HPM	3456	Historical Place Manager
1007	1007HPM	7891	Historical Place Manager
1008	1008HPM	2345	Historical Place Manager
1009	1009HPM	6789	Historical Place Manager
1010	1010HPM	1234	Historical Place Manager
1011	1011HPM	5678	Historical Place Manager
1012	1012HPM	9123	Historical Place Manager
1013	1013A	4567	Administrator
1014	1014A	8912	Administrator
1015	1015A	3456	Administrator

Rules

- You need to write your program by using Python 3.x.
- You can only use all built-in functions and modules.
- You need to put all your files into a folder which is named with your student id(s) and submit the compressed version of the folder.
- Only one team member should submit the assignment.
- The code quality, modularity, efficiency and appropriate comments will be part of the grading.

Grading Policy

The assignment will be graded as follows:

Grading Item	Mark (out of 100)
Database Creation File	10
Client: Historical Place Manager	20
Client: Administrator	30
Server	40