**CA2 External Documentation**

1. Purpose of Program

|  |
| --- |
| Allow users to find and purchase/subscribe to services offered by the company, generate invoice the purchase, show subscribed services and their expiration dates. Allow admin to add and edit services offered by a company. |

1. Features
   1. Basic

|  |
| --- |
| ESP system implemented using sockets ESP\_server.py acting as file system and ESP.py to serve as client  Allow an administrator to add service items, and the prices to the menu (through txt file on server or admin system(admin.py))  Display the list of services for users to subscribe  Allow users to search and add items to subscribe  Modify services added to cart  Tabulate amount to pay and allow users to confirm their selection and pay  Services added to subscribed services in user files (at server) after payment and has timestamps indicating start and end of subscriptions  Additional subscription length can be added to subscribed services in user files (at server)  Used python dictionaries, list  User inputs validated  ESP.py client can read files from server to determine (time stamps indicating start and end of subscription) end date for services currently subscribed  Printout at server end to indicate communication with the client  Allow admin to put the ESP server to maintenance mode or shut it down using admin system |

* 1. Advanced

|  |
| --- |
| Allow admin to put ESP server into maintenance mode using admin system so normal users cannot use ESP.  Sorting of list of services outputs for user to choose from to add to cart. Services sorted according to name or price.  Users can create and login to an account  Services will expire and be removed from user’s account subscribed services  Admin system for staff to update subscriptions offered and send to server  Allow admin to change a user’s password through admin system  Traffic between server and clients are encrypted  User account file(on server) Stores user information such as full name, email, address which would be used for generating invoice that are encrypted  Membership tier implemented for user account depending on how much is spent. Sliver for user’s who spent $100000 and above and gold for user’s who spent $1000000  Discount system for user account with membership tier of sliver(10%) and gold(20%)  Allow user to change user information in their account and their password  Cart is stored for every account and so user can continue where they left off without the cart being emptied every time they quit the program  Invoice can be choose to be downloaded  Invoice emailed to user’s account email if possible  Users can view and download past invoices  Logs for server traffic and interaction with clients Log user and admin out if there is a password change  Server has multi-threading and timeouts |

1. Folder for input/output files = **c:/psec**
2. Accounts
   1. User account

|  |  |
| --- | --- |
| test | pswd = testing |

* 1. Admin account

|  |  |
| --- | --- |
| admin | pswd = admin |

1. Input and Output Files/Folders

|  |
| --- |
| Folder server files:   1. ESP\_server.py (Input) 2. admin folder   adminaccounts.txt (Input and Output)  logs.txt (Input)  rawlogs.txt (Input)   1. data folder   accounts folder  Invoices folder  Invoicenum.txt (Input and Output)  Invoicetracking.txt (Input and Output)  maintenance.txt (Input and Output)  serviceslist.txt (Input and Output)  Folder client files:   1. ESP.py (Input) 2. ESP\_admin.py (Input) 3. Invoice folder |

1. Sample runs

ESP.py:

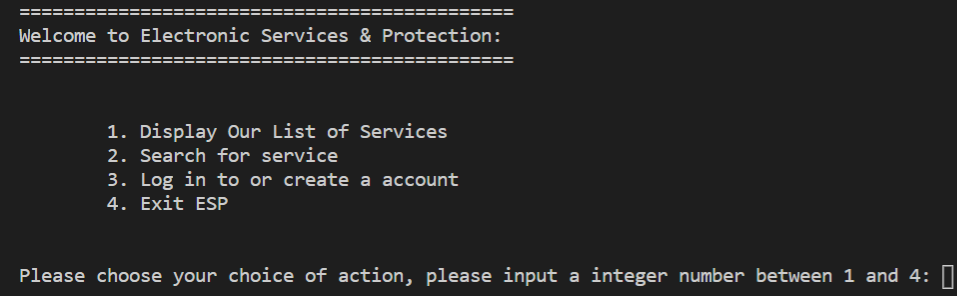
Message if cannot connect to ESP server



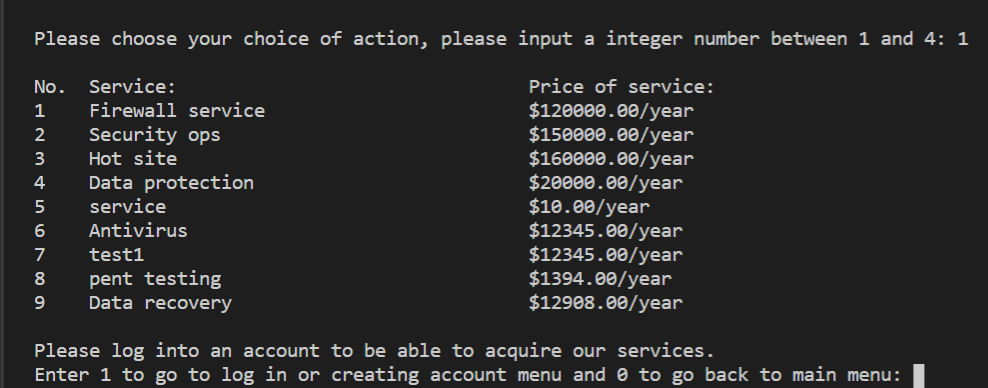
Message if ESP server in maintenance mode

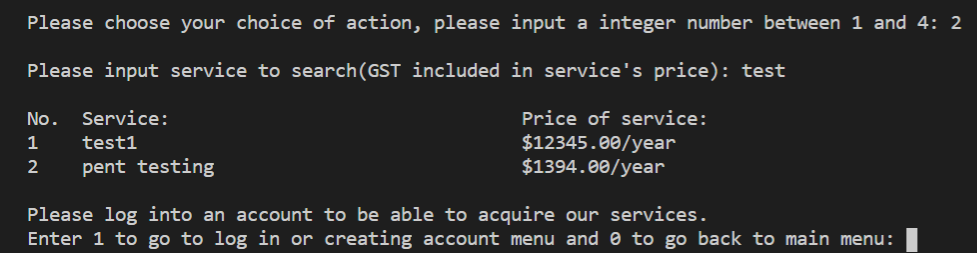


Main menu when not logged in

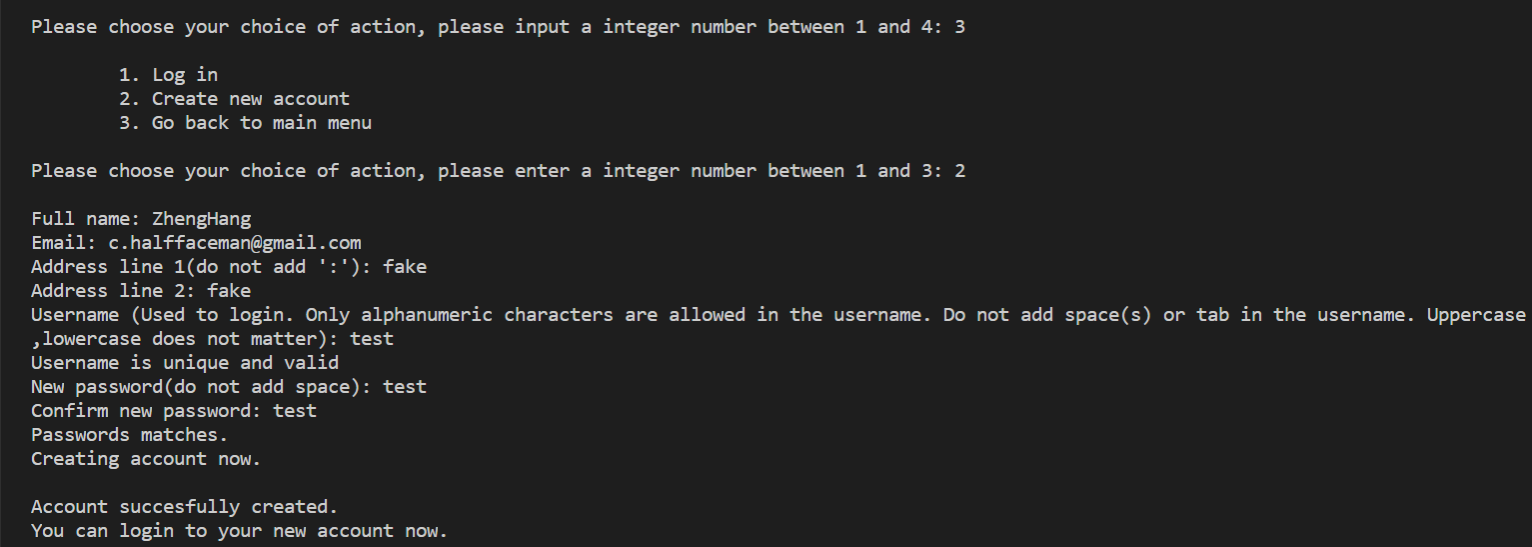


Displaying list of services

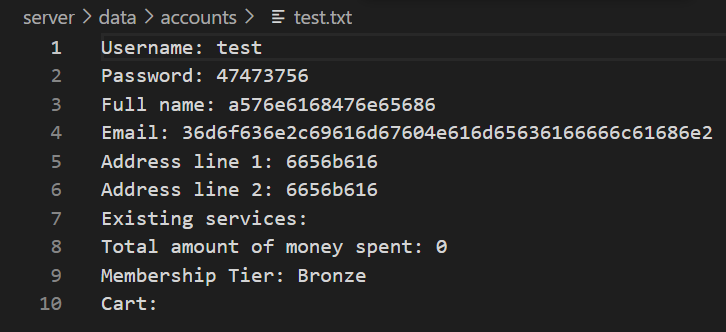


Search and list service

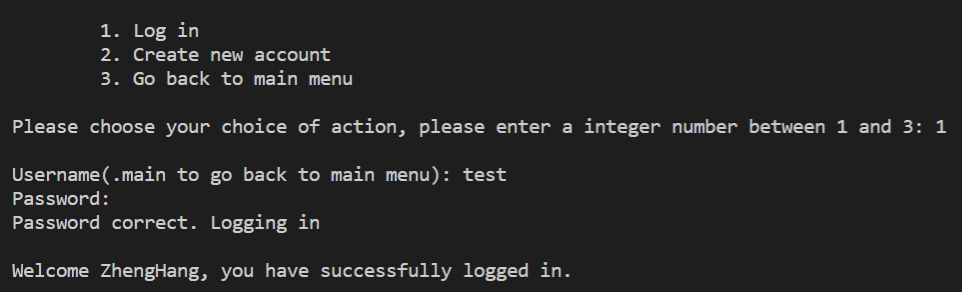
Making new account



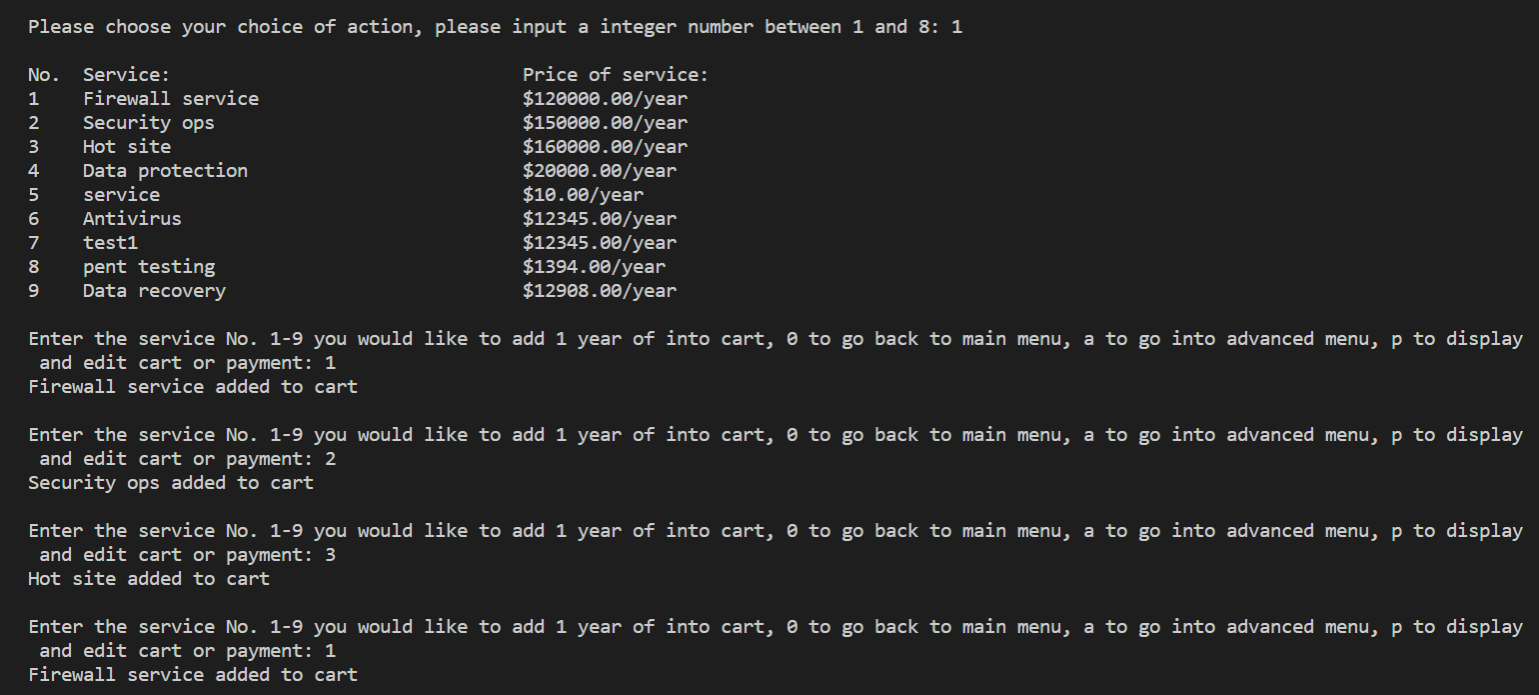
test.txt created in accounts folder when account is made



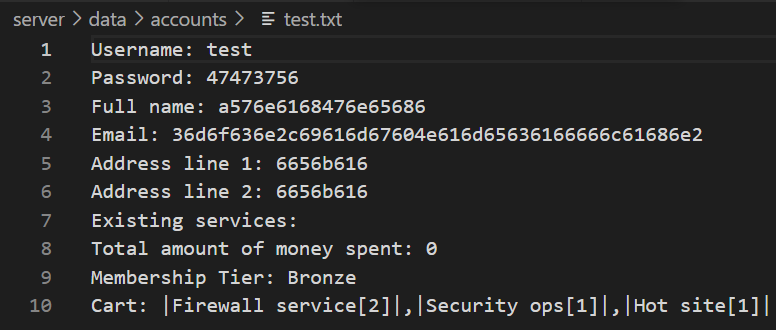
Logging in



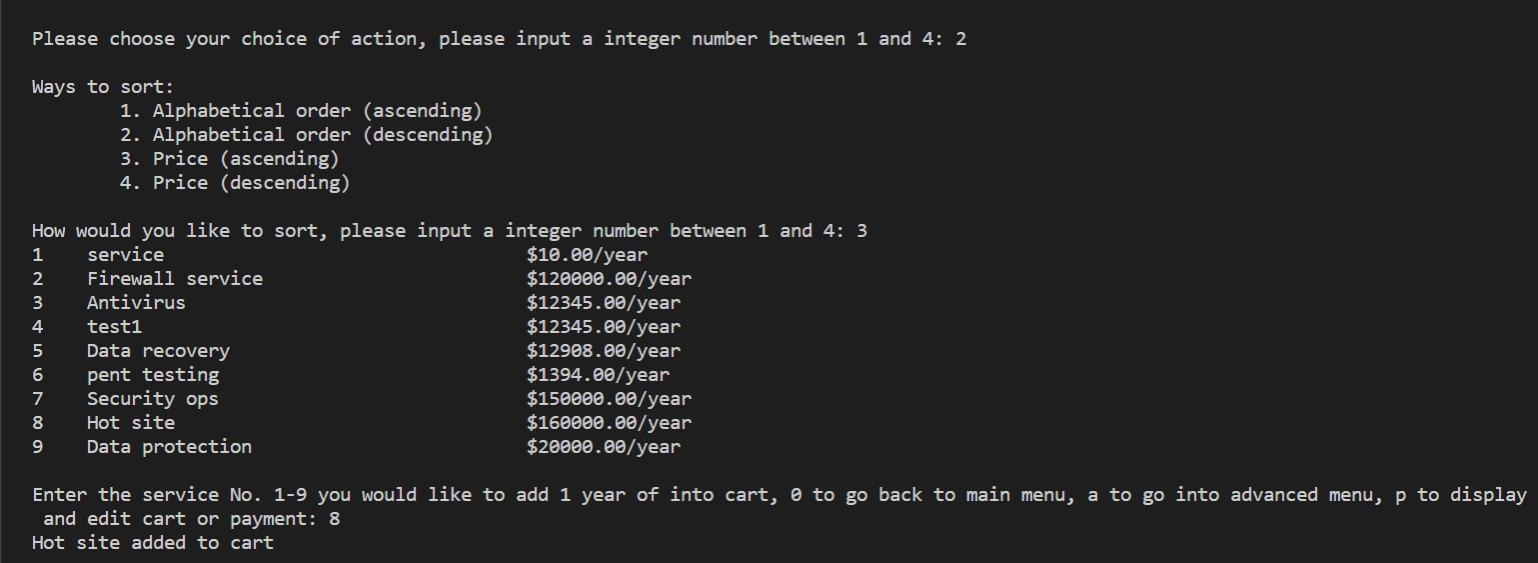
Adding services to cart



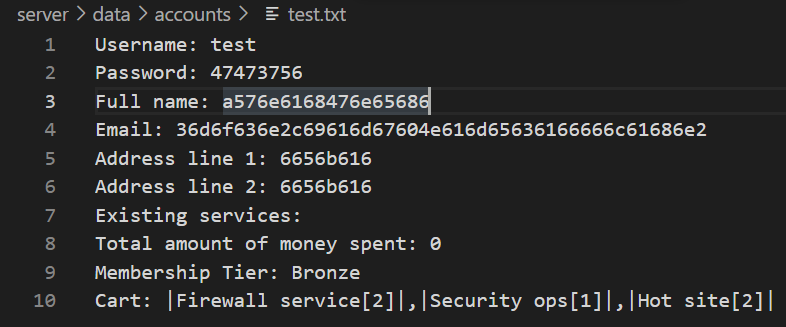
Cart info in test.txt updated



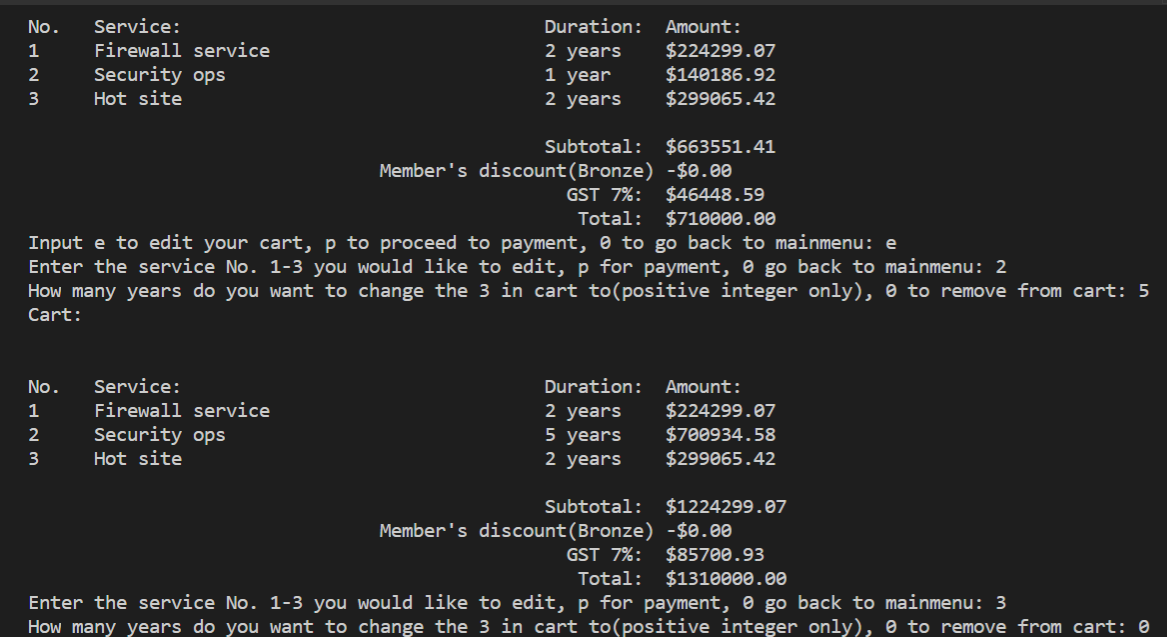
Sorting by price(ascending) and adding services



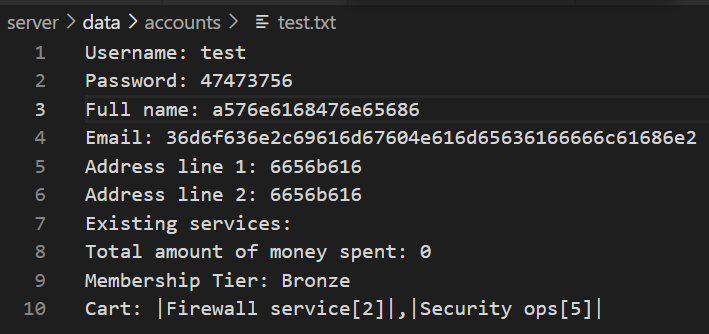
Cart info in test.txt updated



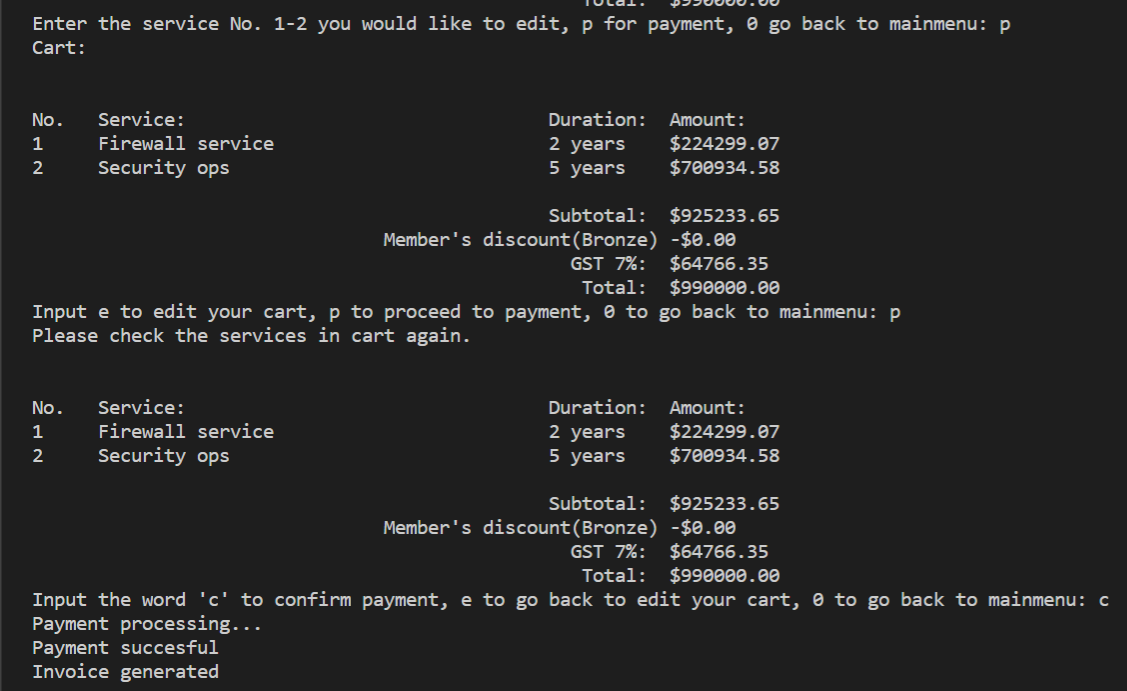
Editing cart, changing years of security ops to be subscribed to from 1 to 5 and removing Hot site from cart



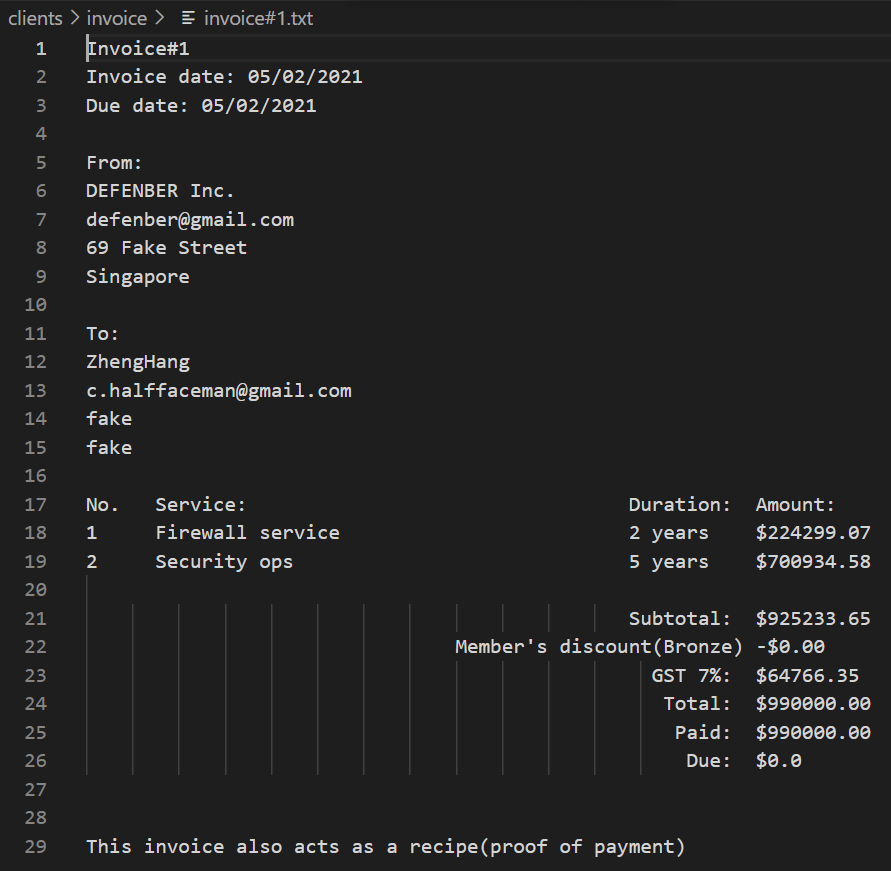
Cart info in test.txt updated



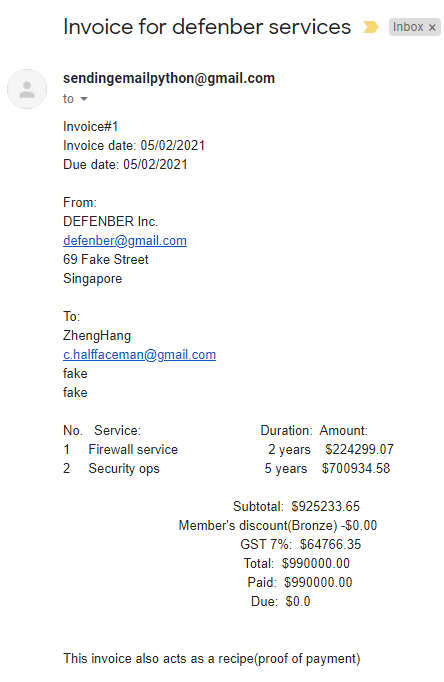
Payment for services in cart



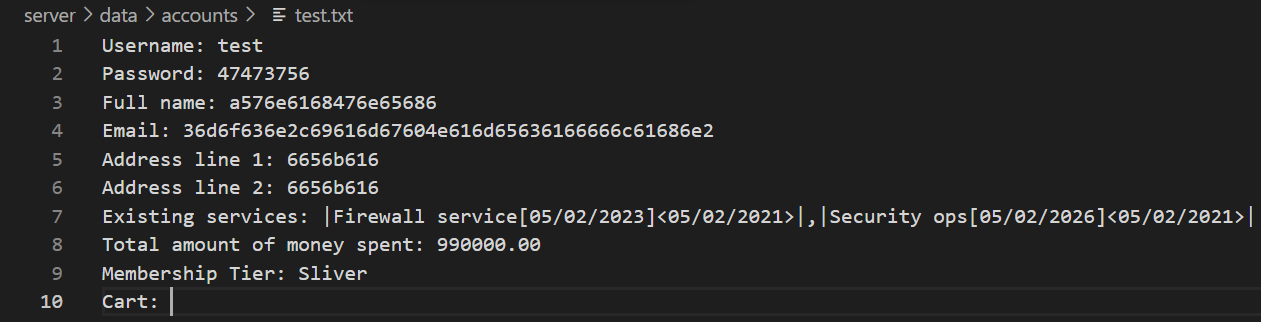
Invoice if choose to save



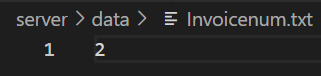
Invoice sent to email



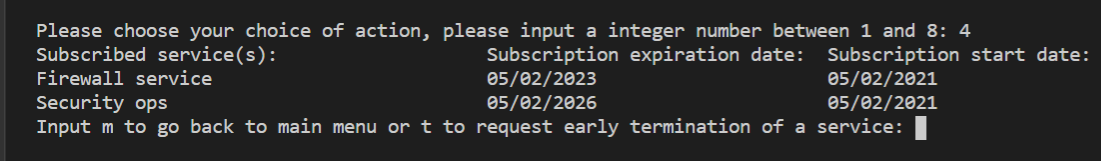
Existing services, membership tier and cart info updated in test.txt



Invoice number in Invoicenum.txt go up from 1 to 2



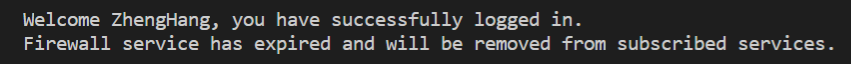
Show subscription services and their start and expiration date

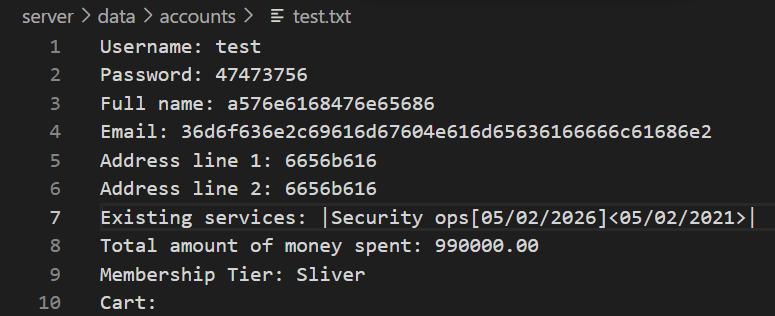


Change Firewall service expiration date in test.txt so firewall service will expire

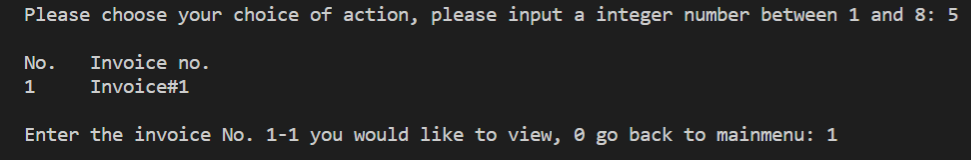


Firewall service expired when logged in and test.txt is updated

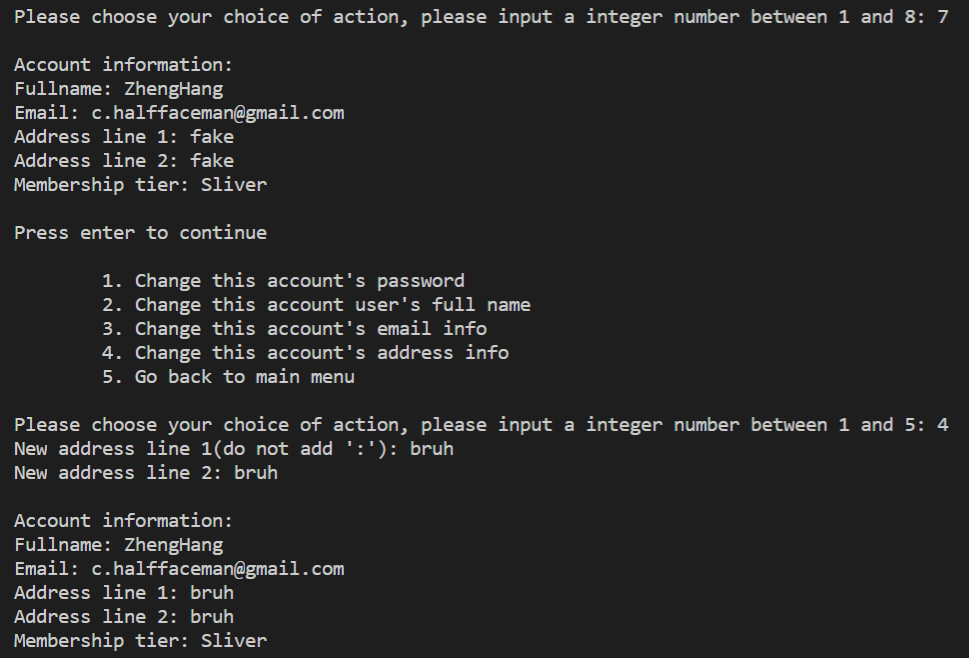




Choose invoice to view and save if needed

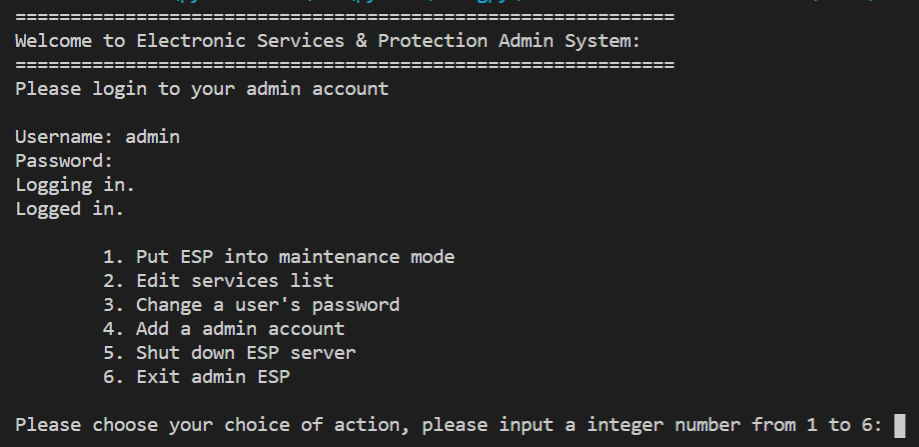


Change account address information

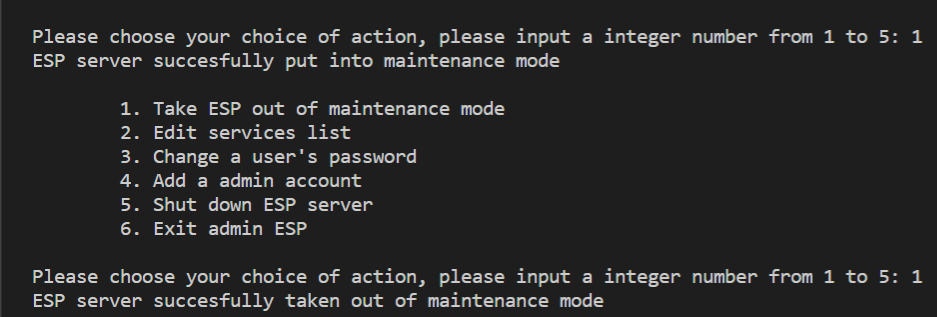


ESP\_admin.py:

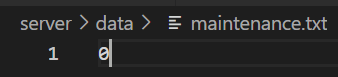
Admin login and main menu



Putting and taking ESP out of maintenance mode



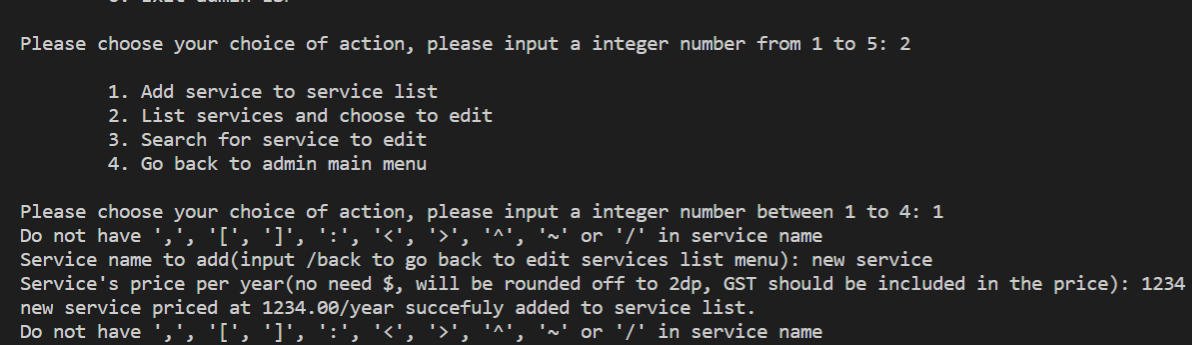
Put into maintenance mode

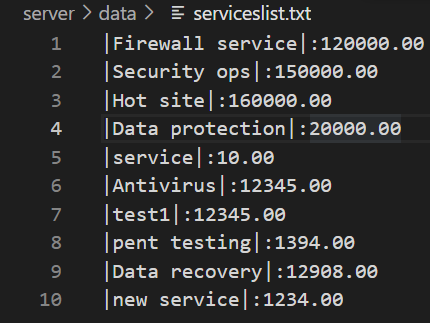


Take out of maintenance mode

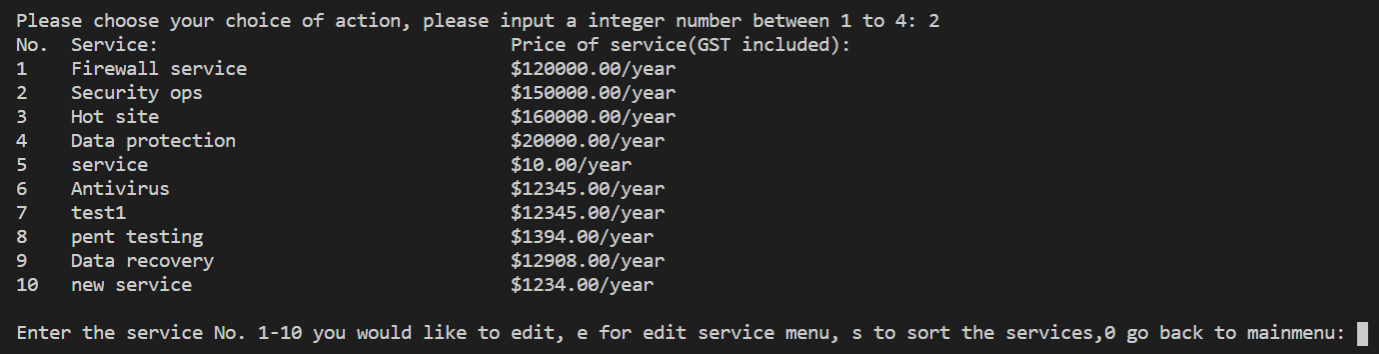


Add service and price to services list and updated serviceslist.txt

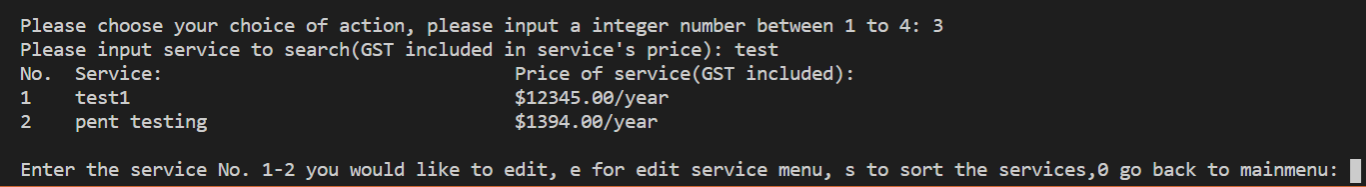




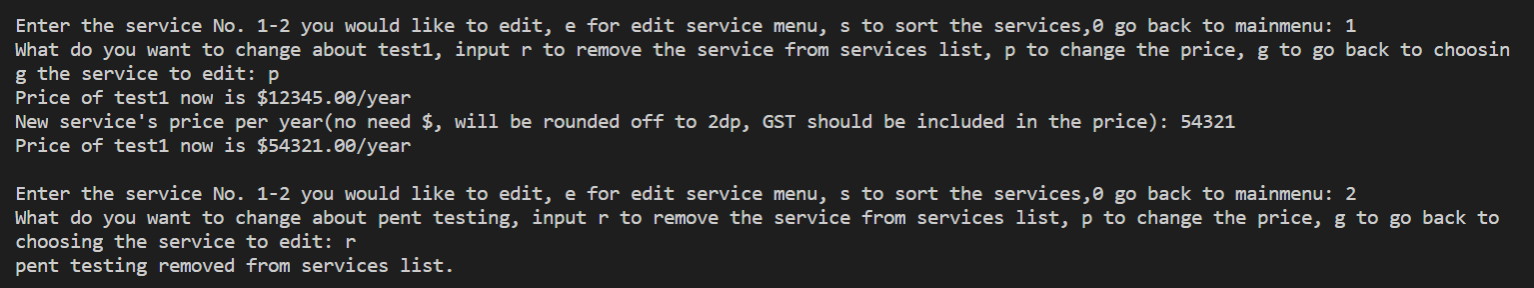
Listing services to choose and edit



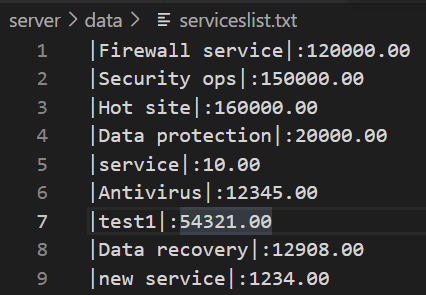
Search for services to choose and edit



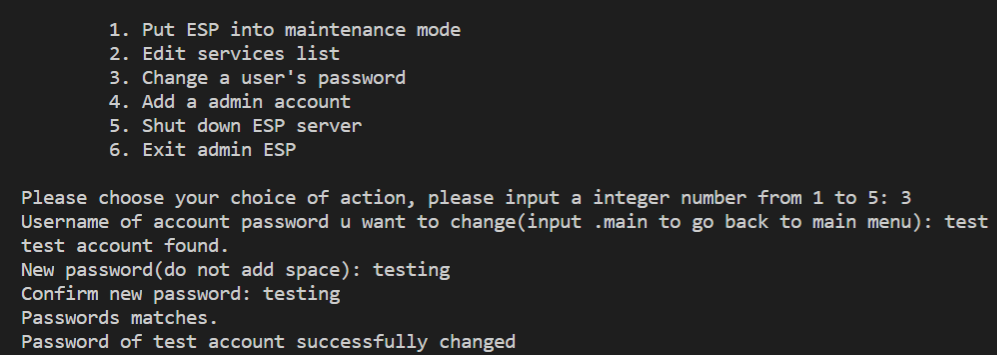
Edit service price test1 to $54321 and remove pent testing service



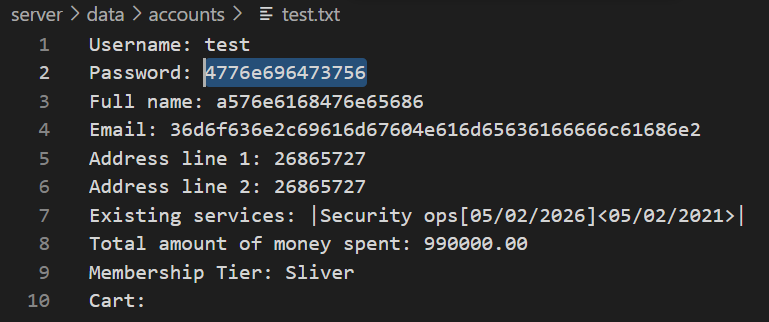
serviceslist.txt updated



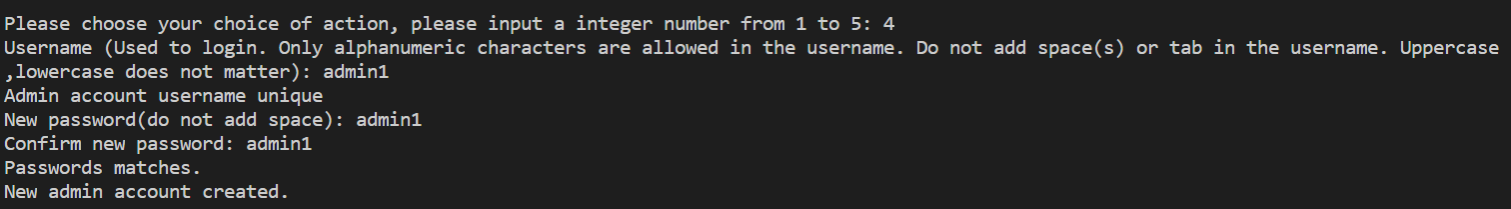
Changing password of test account



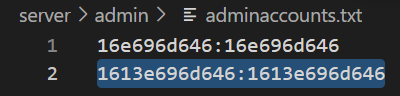
Password changed in test.txt



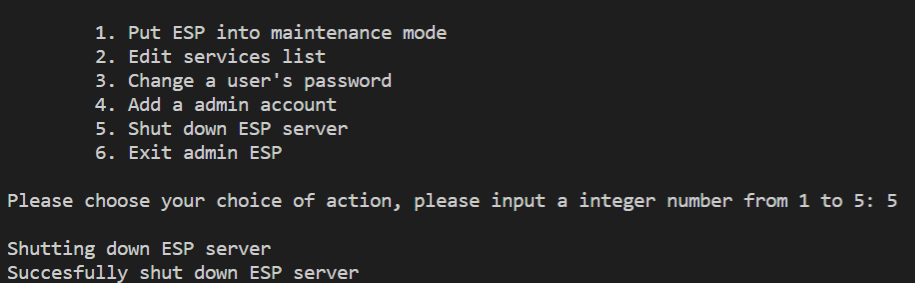
Add new admin account



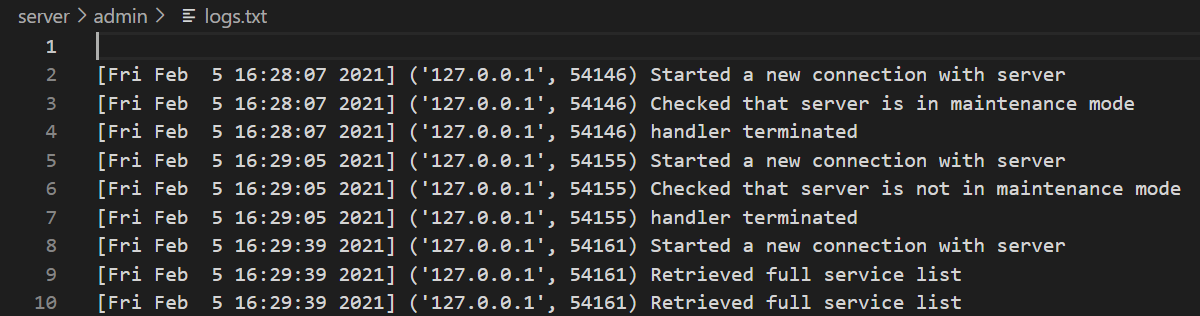
adminaccounts.txt updated



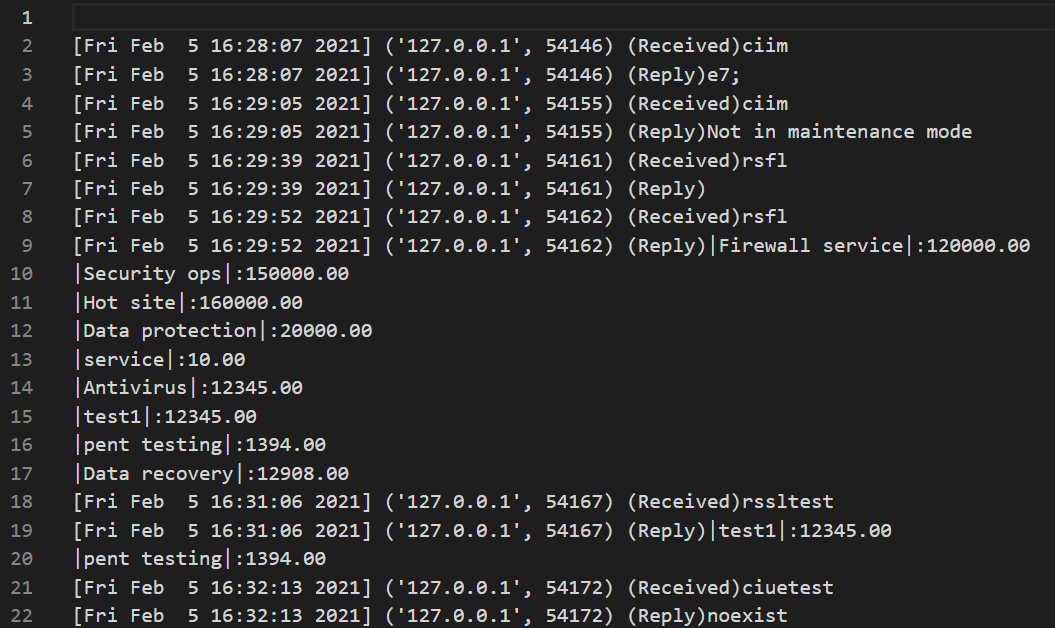
Shut down ESP server



Server logs



Server raw logs



**Limitations:**

Service name cannot include ',', '[', ']', ':', '<','>', '~' or '/'

Admin should put ESP server into maintenance mode before changing service list

Cannot input character that cannot be written into txt files

Input or resulting strings cannot be so long, it does not fit into txt files

Username must be alphanumeric

txt files cannot be tampered with incorrectly and must follow a fixed format as it would mess up the program, use ESP\_admin.py unless tampering with txt files is necessary

an admin account’s username and password must be already added to adminaccounts.txt for ESP admin system to be usable

**What is needed and what to do:**

Python 3.7.2 and higher

Use command prompt to run the programs:

cd C:/PSEC

run server:

py -3.7-64 server/ESP\_server.py

running clients:

py -3.7-64 clients/ESP.py

py -3.7-64 clients/ESP\_admin.py