Python Course. The final project. Lots-Of-Trips: automated public transportation system.

- The system is built as client(s) – server system. Clients send requests to the server and receive a relevant answer. Thus, the request considered as satisfied.
- Objects of the system are magnetic cards that stored as records in SQL table on server-side. Each card has its unique ID number (four digits at most), contract and wallet fields.
- Also, there are 4 types of contracts: North, Center, South and None.
- The value of the wallet is integer, non-negative number.
- Before satisfaction of each potential request, the server checks the card's legality. In case that there is no such card in the dataset a relevant message is shown and request is rejected.
- Client allowed to receive followed services:

    1. **Create a new card.** The client requests it and the server returns back the ID of the newly created card. The default value of the card's wallet is 0 and the contract is None.
    2. **Check the status of an existed card.** The client asks for information about the existed card, mentions its ID and receives the data: the card ID, a wallet and a contract values.
    3. **Pay for a ride.** The client is interested to pay for a ride, mentions card's ID, region of the

destination and receives two possible answers: Done in case that ride is allowed or Fail otherwise.
The payment check done automatically:
at the beginning, the contract field is checked. The client is allowed to have a ride only in region mentioned in value contract field with no payment at all.
If the client decides to take a ride to different that mentioned in contract field region the payment will be automatically redirected to wallet and the amount will be subtracted according to the price list.
There are predefined prices for each one of regions: to the north costs 25, center – 40, south – 30.
In case of both checks fail, the is not ride possible.
4. **Fill/Refill the wallet.** The client wants to increase or refill the value of a wallet. It requires mentioning card's ID and the sum to add to wallet value. In case that the value is logically correct (you can assume that the maximum sum of refill at one time is 4-digit number), the server notifies the client about success by sending Done, otherwise – client sends Fail as a response.
5. **Change the contract.** The client decides to change the contract, mentions card's ID and a name of the contract. In case that the value is logically correct, the server notifies to the client about success, otherwise – the message about a failure is sent back.

- **Notes:**

  If you want it to, you can create a simple (or not) GUI for your project as extra points to the final score.

  You can use any technique, learned during the course, for clientserver connection.

  Pay attention on unforeseen conditions: illegal or incorrect inputs from user, potential disconnections between server and clients. Use any known instruments to prevent unexpected behavior of the system.

- **The final date is: 18.08.2023**
- Send the project as zip, rar or other version of archive, contained server, client(s), GUI (in case you will make it) and other necessary according to your opinion files.
- I recommend send the final version of the project both in two ways: using students portal and my e-mail: grishas@rt-ed.co.il

Best of luck!

I hope you enjoyed during the studying.