COMPENG 4DN4: Laboratory 2
Online Grade Retrieval Application

Ashpan Raskar - raskara - 400185326
Dharak Verma - vermad1 - 400114166

1 Member Contributions

1. Ashpan Raskar
   Created the server class and its class fields and methods. Created the main method.

2. Dharak Verma
   Created the client class and its class fields and methods. 3

2 Server Pseudo Code

```
Setup the host based on the current IP address
Configure the port and the file name from the .ENV file
Create a dictionary of commands to easily reference data from the CSV
Bind the socket to the host and port
Run the print course grades function
Run the main server program function

define print course grades function
    Open the CSV file
    Read the CSV file
    Clean the CSV file
    Print the course grades
    Parse the data into a class field dictionary
    Close the CSV file

define clean data function
    Remove the newline character from the data
    Remove the double quotes from the data
    Remove the spaces from the data
    Remove the empty strings from the data
    Return the cleaned data

define parse data function
    Create a dictionary to store the data
    Iterate over the data
        Split the data into a list
        Store the data in the dictionary
    Return the dictionary

define the run command function
    if the command is "GG"
        append all the students grades to a string
    if the command is "GMA"
        append the students average midterm grade to a string
    if the command is "GEA"
        append the students average exam grade to a string
    if the command starts with "GL"
        append the students lab grade based on the command to a string using the dictionary
    return the string

define main server program function
    Listen for connections
    Accept the connection
    while true:
        Receive the data
        Run the run command function
        Encrypt the data using their secret key from the run command function
```

```
        Send the data
    Close the connection
```

3 Client Pseudo Code

```
Setup the host based on the current IP address
Configure the port from the .ENV file
Create a dictionary of student numbers and their secret keys for testing purposes
Create a dictionary of commands to create user readable prints
Create a socket to connect to the server with the host and port
Run the main client program function

define the main client program function
    while the command isn't "bye"
        Take in the student number and the command in from the user
        Print the command being run using the dictionary to the user
        Send the student number and the command to the server
        Receive the data from the server
        Decrypt the data using the secret key from the dictionary
        Print the data to the user
```

4 Explanation

The general workflow of the program is as follows.

1. The client sends their student number and the command they want to run to the server.
2. The server receives the command and runs the appropriate function to compute the value get the data.
3. This data is then encrypted using the secret key of the student.
4. The data is then sent back to the client.
5. The client receives the data and decrypts it using the secret key.
6. The client prints out the data to the user.

To begin the program, the server reads the data from the CSV file and stores it in a dictionary for ease of usage. The server prints all the data, and then binds the socket to the host and port. The server then runs the main server program function. Meanwhile the client also binds the socket to the host and port and is waiting for user input. The parse data function works by creating a dictionary of data in the following format:

```
{
    student_number_1: {
        "key": secret_key,
        "Student Name": student_name,
        "Midterm": midterm_grade,
        "Lab 1": lab_1_grade,
        "Lab 2": lab_2_grade,
        "Lab 3": lab_3_grade,
        "Lab 4": lab_4_grade,
        "Exam 1": exam_1_grade,
        "Exam 2": exam_2_grade,
        "Exam 3": exam_3_grade,
        "Exam 4": exam_4_grade
    },
    student_number_2: {
```

```
            "key": secret_key,
            "Student Name": student_name,
            "Midterm": midterm_grade,
            "Lab 1": lab_1_grade,
            "Lab 2": lab_2_grade,
            "Lab 3": lab_3_grade,
            "Lab 4": lab_4_grade,
            "Exam 1": exam_1_grade,
            "Exam 2": exam_2_grade,
            "Exam 3": exam_3_grade,
            "Exam 4": exam_4_grade
        }, ...
}
```

In the run command function, for the "GLx" commands, the function checks if the command starts with the string "GL" and if it does, it uses the dictionary to check the command with the field in the dictionary to fetch the data. For example if they enter "GL3", then the dictionary value for the key "GL3" will be "Lab 3" and for each student number it will fetch the grade for "Lab 3" and finally take the average of all the grades and return it.

In the encrypt and decrypt string functions, the fernet encryption method is used with a symmetric secret key (the same key for encryption and decryption), this is encoded/decoded using UTF-8 character set.

The main method works by initializing the server or the client based on the flag being passed when running the file using the "-r" or "–role" flag. If it is run with the "-r client" flag, then the client class object is initialized, if it is run with the "-r server" flag, then the server class object is initialized.