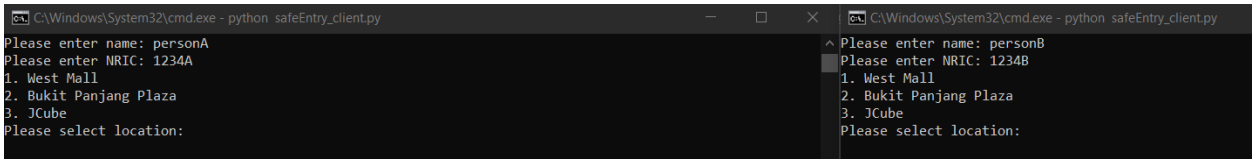
Demonstration

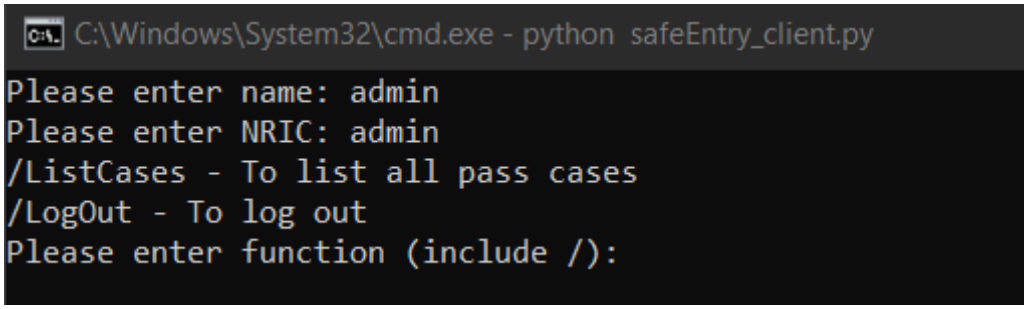
To showcase all functionalities, first start 4 terminals. 3 would run as a client (2 as a user, 1 as an admin).

1) Start 4 independent terminals to run 3 clients '*python safeEntry_client.py*' and 1 server '*python safeEntry_server.py*'

2) The 3 clients would act as 2 different individuals and 1 'MOH officer'

3) Enter any name and NRIC value for both the individual clients, and enter admin/admin





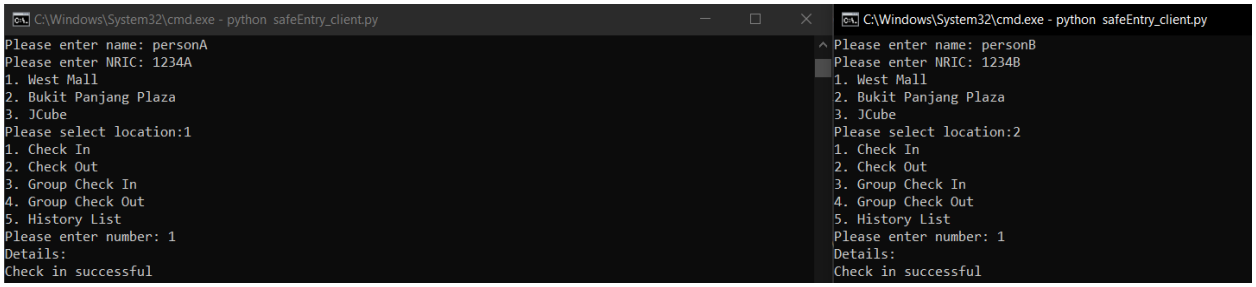
If the original data is used, the dataset should contain 28 rows with all individual flags set to 0.

<u>Individual and Group CheckIn/CheckOut</u>

1) In the first user client select any location and check in followed by checking out

2) In the second user client select a different location and does a check in followed by checking out

3) In both the clients perform a group check in and check out as well.



```
3. Group Check In
4. Group Check Out
5. History List
Please enter number: 3
1234A 26/06/2022 16:38:47
1. Check In
2. Add another user
Please enter number: 2
Please enter name: family1
Please enter NRIC: 1234fam
1. Check In
2. Add another user
Please enter number: 1
Details:
Name: personA
NRIC: 1234A
Location: West Mall
Check In: 26/06/2022 16:38:47
Group Number: 2
```

History Listing

1) On any individual client, select history listing and it should display past check in entries, sorted according to the most recent entry, including those entries that were made by other users using group check in on behalf of the individual.

```
5. History List
Please enter number: 5
message: "[Location: West Mall | Check In: 26/06/2022 16:38:47 | Check Out: -]"

message: "[Location: West Mall | Check In: 26/06/2022 16:37:40 | Check Out: 26/06/2022 16:37:41]"
```

Declaration

1) In the admin terminal, input "/ListCases" and view all cases in the past 14 days (as of local host machine time).

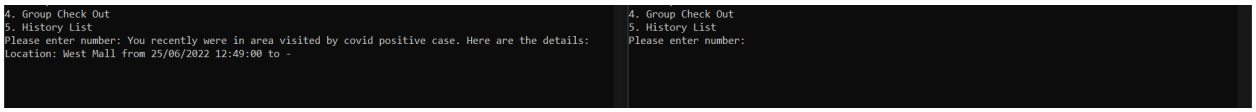
```
/ListCases - To list all pass cases
/LogOut - To log out
Please enter function (include /): /ListCases
```

2) On the left, it displays the indexes that can be chosen to declare as a COVID positive case.

```
(26, ['Jackie', '262422J', 'JCube', Timestamp('2022-06-27 17:33:00'), '27/6/2022 21:33', 0, '237444T 23/
06/2022 17:33:24', 0])
(27, ['James', '23311J', 'JCube', Timestamp('2022-06-28 17:03:00'), '28/6/2022 17:43', 0, '26237A 23/06/
2022 17:03:06', 0])
(28, ['Tom', '237444T', 'West Mall', Timestamp('2022-06-25 12:49:00'), '-', 1, '237444T 25/06/2022 12:49
:14', 0])
```

3) Select a case that has the same location selected in the first client (28 - "West Mall" in this demonstration).

4) Upon selecting, client 1 should receive the notification and client 2 should not receive anything.



Login Notification

1) Terminate the first client and re-run the client and log in as the same person. A notification should be displayed on possible exposure

```
Please enter name: personA
Please enter NRIC: 1234A
You recently were in area visited by covid positive case. Here are the details:
Location: West Mall from 2022-06-25 12:49:00 to -
1. West Mall
2. Bukit Panjang Plaza
3. JCube
Please select location:
```

Check in Possible Exposure Notification

1) In the second client, terminate and re-login.

2) Now select the location(“West Mall" in this demonstration) that was declared an area of possible exposure.

3) Another notification should pop up displaying to the user that they are in an area that was visited by a covid-19 case in the past 14 days.

```
Please enter name: personB
Please enter NRIC: 1234B
1. West Mall
2. Bukit Panjang Plaza
3. JCube
Please select location:1
1. Check In
2. Check Out
3. Group Check In
4. Group Check Out
5. History List
Please enter number: 1
Details:
You have just checked in an area visited by a covid positive case in past 14 days
```

Using Docker

The application has also been packaged inside a docker container. A dockerfile has been created to ease the building of the image. It has been provided under the docker folder.

```
C:\Users\JustP>docker ps
CONTAINER ID   IMAGE       COMMAND               CREATED         STATUS         PORTS     NAMES
6d1f7e8eb506   safeentry   "tail -f /dev/null"   2 minutes ago   Up 2 minutes             zealous_leakey


C:\Users\JustP>docker exec -it zealous_leakey bash
root@6d1f7e8eb506:/# python safeEntry_client.py
Please enter name: personB
Please enter NRIC: 1234B
1. West Mall
2. Bukit Panjang Plaza
3. JCube
Please select location:1
```

```
Run a command in a running container

C:\Users\JustP\OneDrive\Documents\SiT\Y2T3\CSC3004 Cloud and Distributed Computing\Lab\lab_assignment\do
cker>docker exec -it zealous_leakey bash
root@6d1f7e8eb506:/# dir
bin   home   mnt                   root             safeEntry_client.py    sbin  usr
boot  lib    opt                   run              safeEntry_pb2.py       srv   var
dev   lib64  proc                  safeEntry.csv    safeEntry_pb2_grpc.py  sys
etc   media  reproducible_data.csv  safeEntry.proto  safeEntry_server.py    tmp
root@6d1f7e8eb506:/# python safeEntry_server.py
['personB', '1234B', 'West Mall', '27/06/2022 05:10:39', '-', 1, '1234B 27/06/2022 05:10:39', 0]
```

1. Run the command *docker build -t safeentry .* to build the image
2. Run *docker run -d -P safeentry* to run the container
3. Run *docker ps* to obtain the container name
4. To enter the bash shell of the container run *docker exec -it [container name] bash*
5. Repeat step 4 three more times to run 4 different shells to proceed with the demonstration while using containers instead.