

## EECE 350 Final Project

**Professor:** Dr. Ajami

**Delivered to:** Dr. Jad Matta

**Group 8:** Nourhane Abdel Samad, Omar Al Jaroudi & Mehieddine Zeidan

---

Programming language: Java

IDE: Eclipse

GUI Platform: Javafx (with Scene Builder)

Database: MySQL

Model: Client/Server

OS: Windows 10

### **I. Design:**

#### **1. Client Class (MainControl.java)**

The client class is responsible for initiating contact with the server.

Establishing such a connection consists of creating a socket on a specific IP and port number (in our case during testing we used IP 127.0.0.1 and port 1927).

This is done upon pressing a button.

Scene 1:

##### **a. “login” button**

1. Socket is created on (IP, port)
2. Username and password are passed to output stream (using writeUTF function) separated by “/” (forward slash).
3. Result is read from input stream (using readUTF)
4. If credentials were verified, user gets logged in
5. If credentials are wrong or don’t exist in the database, error message is displayed on the screen. Scene doesn’t change.
6. Socket closed

##### **b. “Sign up” button:**

1. Client wants to create a new account, they are led to another scene to create an account

Scene 2: Sign up scene (optional)

##### **a. “create” button**

1. Socket is created on (IP, port)
2. All entered credentials are passed to output stream separated by “/”

3. Result is read from input stream
4. If credentials are verified, user is taken back to the login scene
5. If not, error message is displayed on the screen

Option 1: "View available gates"

→ The client can view whether each of the 10 gates is available or not

The user is presented with the 10 gates, and prompted to press "Refresh"

This is to make sure that the results are the most up to date

The client is then able to view the status of each gate as "Available" or "Unavailable"

NOTE: A gate is available if it contains at least 1 free time slot

The user can either go back to main menu (lower right corner), or proceed to reserve a time slot(lower left corner)

Option 2: "View time slots for a gate"

→ The client can view the availability of all time slots for a certain gate

The client is prompted to enter a valid gate number and press confirm

Then they will be able to view the status of each time slot for this gate, as "Available" or "Unavailable"

The user can either go back to main menu (lower right corner), or proceed to reserve a time slot(lower left corner)

Option 3: "Search for a specific time slot"

→ The client can search for a specific time slot and view its availability across all gates

The client is prompted to select one of the 12 time slots in front of them

Upon doing so, they will be able to view whether the status of this slot at each gate, as "available" or "Unavailable" after communicating with the server as explained above.

The user can either go back to main menu (lower right corner), or proceed to reserve a time slot(lower left corner)

Option 4: "view history of interactions with server"

→ The client can view their past interactions with server (logins, signups, logouts, reservations, searches)

The client is prompted to confirm their identity by entering their valid password and pressing confirm

Upon doing so, the client is presented with all transactions recorded on the server since they have signed up, after communicating with the server as explained above.

Each transaction has a description, timestamp, and date, which is managed by the Server.

The user can go back to main menu (lower right corner)

Option 5:"cancel a reservation"

→ The client can cancel a reservation they have previously made.

The client is prompted to enter a valid gate number and time slot.

Then the server will verify if the client has previously made a reservation with this info before

If so, the client will have successfully canceled the reservation, and the time slot is made available for new reservations.

If not, the client is told that they haven't reserved this slot before.

The user can go back to main menu (lower right corner).

Option 6:"waiting list"

→ Client can view and update a waiting list for the gates (in case reservations are not possible)

LOGOUT: When in main menu scene, the client has the option to log out (lower right corner)

By doing so, the client will return to the login scene.

#### Server Class: (server.java)

The server is always waiting for the client to establish a connection. It is listening on the socket that was specified above. If the client establishes a connection, the server will read the prepared statement it has received from its input stream. Then it will process it by parsing the strings (initially separated by “/”). Doing so allows it to deal with each string separately.

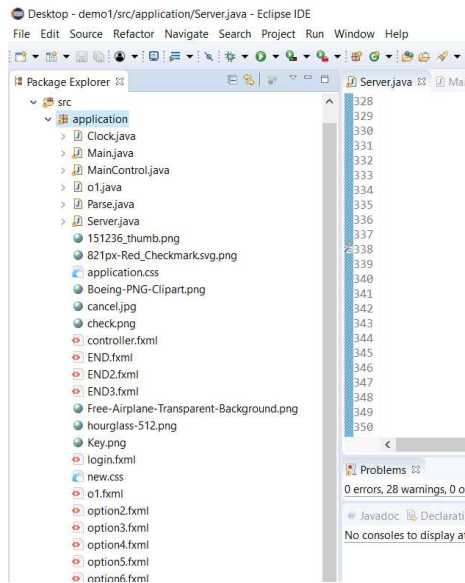
Taking the login example. The input stream will be initially [username +”/” + password]. The server will split this using an array list into username and password. Then it will perform an SQL query to check whether this username and password exist in the database. If the query returns true, the Server passes a confirmation message to the output stream. If not, it passes an error message. The client will then read from the input stream and act accordingly as explained above.

The server executes queries to confirm credentials, and updates the database in case of transactions (registering or canceling) in the same manner, and passes a message to the client accordingly.

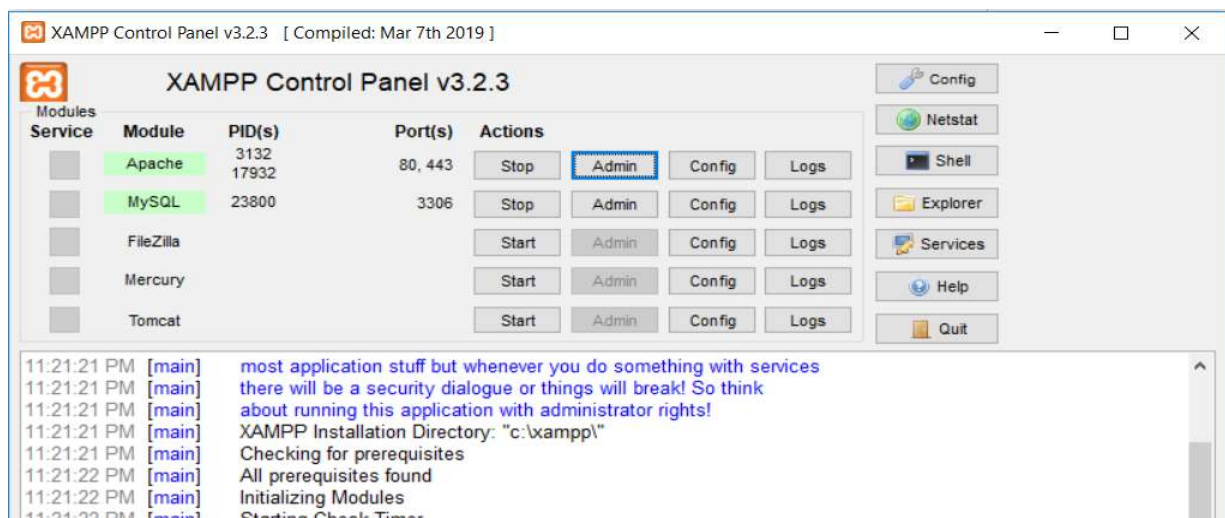
---

## II. Steps to follow by the user:

1. Start the eclipse IDE.
2. Import the project, click on it then click on the application packet.

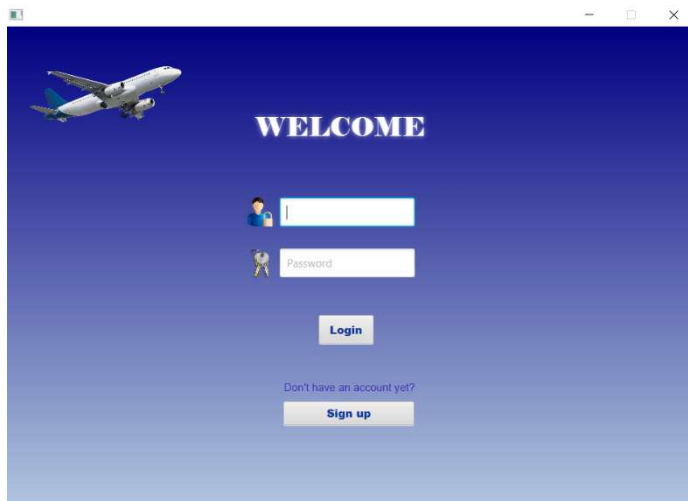


3. Open the XAMPP software, click on the start button next to Apache and MySQL. In order to view the Databases, the user should press on admin next to Apache.  
(This step is crucial to enable the client/server connection)



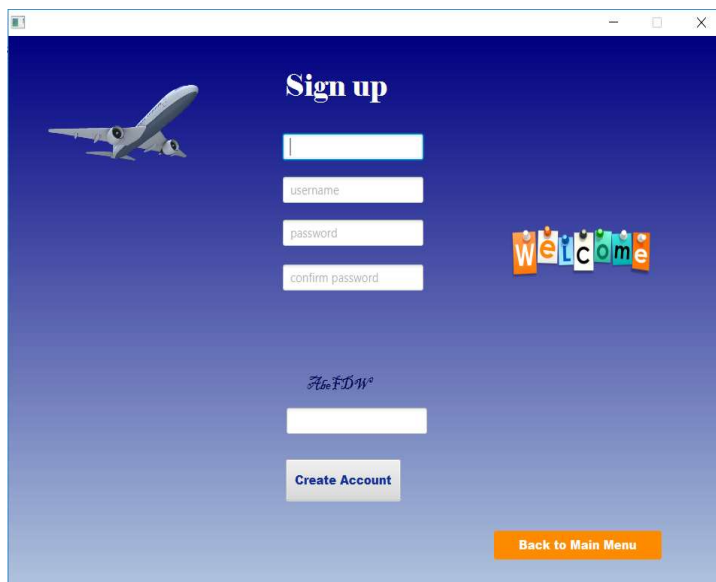
4. Click on **Server.java** then click on the run button (This is done to run the server).
5. Click on **Main.java** then click on the run button (This is done to run the application).

The user should see the scene below:



6. If the user has already an account, he should enter his username and password, then press on the login button.

If not, he/she should press on the **“Create Account”** button. The user should see the scene below:

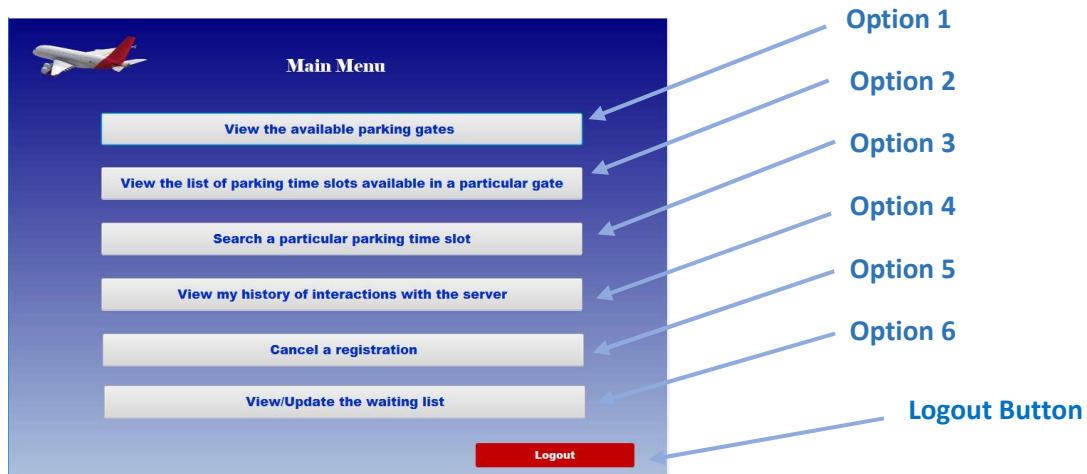


The user should enter a valid email (if the email was not valid, an error message will pop up), a username (that was not used before), a password, and he/she should type the shown characters to prove that she/he is not a robot.

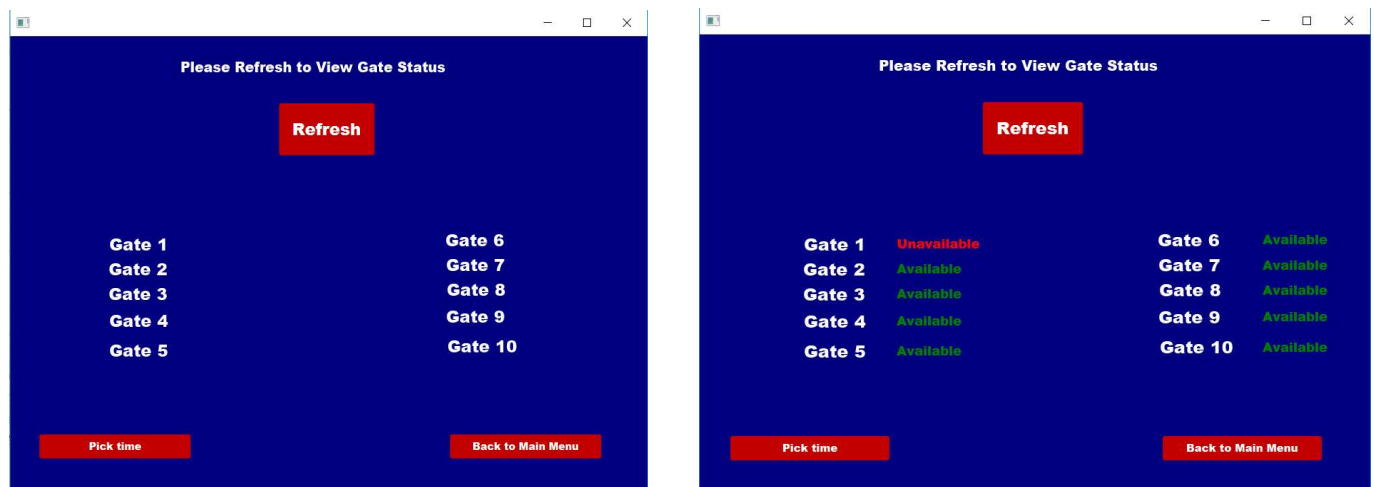
**Note:** If the user leave any field empty, a message will pop up to tell him/her to fill all the fields.

The button **“Back to Main Menu”** can be used to go back to the main frame.

7. After signing up, the user can now login, he should see the scene below if the login was successful:



8. In order to view the available gates, the user should press on the first button. The scene below (to the left) will be opened. After pressing on "Refresh", the scene below (to the right) will be displayed:



In the example above, all the gates except gate 1 have available time slots. The user now can choose to pick a time or go back to main menu by pressing the right button.

9. In order to view the available time slots in a particular gate, the user should press on the second button of the image in part 7. The below scene below (left) will be displayed, after entering a gate number, the scene (right) will be displayed:

Please enter a gate number then press on the Ok button.

Ok

Time Slot 1 (00:00 to 02:00)	Time Slot 7 (12:00 to 14:00)
Time Slot 2 (02:00 to 04:00)	Time Slot 8 (14:00 to 16:00)
Time Slot 3 (04:00 to 06:00)	Time Slot 9 (16:00 to 18:00)
Time Slot 4 (06:00 to 08:00)	Time Slot 10 (18:00 to 20:00)
Time Slot 5 (08:00 to 10:00)	Time Slot 11 (20:00 to 22:00)
Time Slot 6 (10:00 to 12:00)	Time Slot 12 (22:00 to 24:00)

Reserve Back to Main Menu

Please enter a gate number then press on the Ok button.

Ok

Time Slot 1 (00:00 to 02:00) Available	Time Slot 7 (12:00 to 14:00) Available
Time Slot 2 (02:00 to 04:00) Available	Time Slot 8 (14:00 to 16:00) Available
Time Slot 3 (04:00 to 06:00) Available	Time Slot 9 (16:00 to 18:00) Available
Time Slot 4 (06:00 to 08:00) Unavailable	Time Slot 10 (18:00 to 20:00) Available
Time Slot 5 (08:00 to 10:00) Available	Time Slot 11 (20:00 to 22:00) Available
Time Slot 6 (10:00 to 12:00) Available	Time Slot 12 (22:00 to 24:00) Unavailable

Reserve Back to Main Menu

10. The user can now press on “Reserve” to reserve an available slot.

**Reservation**

Please enter the gate number and the time slot that you have chosen.

Gate Number Time Slot Number

Reserve

Back to main menu

The user should enter a valid gate number and an available time slot number, then click on “Reserve”.

In case the user entered an invalid gate number, an error message will pop up.

In case the user entered an unavailable time slot, an error message will pop up:

**Reservation**

Please enter the gate number and the time slot that you have chosen.

Gate Number Time Slot Number

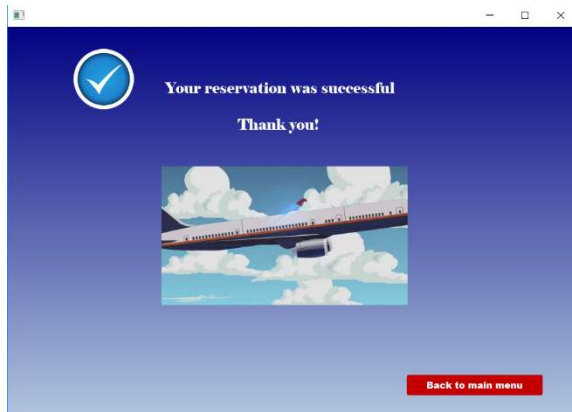
4 12

Reserve

unavailable

Back to main menu

In case of successful registration:



11. If the user wants to check at which gates a particular time slot is available, he/she should choose option 3 and then click on that time slot.

Example:





12. In order to see his/her history of transactions with the server, the user should pick option 4.  
The following will be displayed:

**Please Confirm By Entering Your Password**

..

**Confirm**

Action: logout  
Date: Saturday, April 27, 2019  
Time: 1:10 AM

Action: login  
Date: Saturday, April 27, 2019  
Time: 1:11 AM

Action: force\_logout  
Date: Saturday, April 27, 2019  
Time: 1:11 AM

Action: login  
Date: Saturday, April 27, 2019  
Time: 1:12 AM

**Back to Main Menu**

Use this to scroll up and down

13. In order to cancel a registration, the user should go for option 5.

**Cancel a reservation**

Please enter the gate number and the time slot.

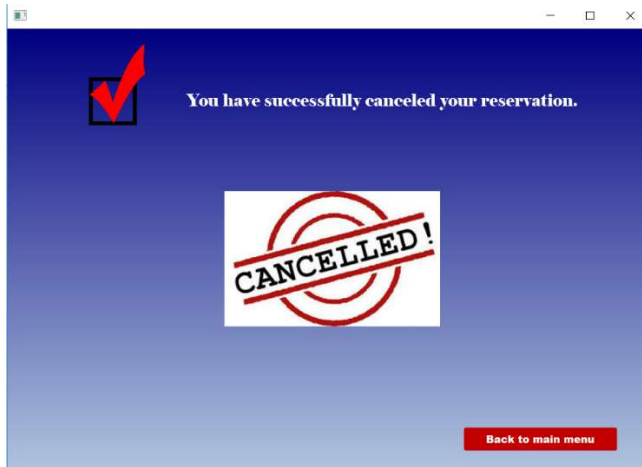
**Gate Number**

**Time Slot Number**

**Cancel**

**Back to main menu**

If the cancelation was successful, the user should see this scene:



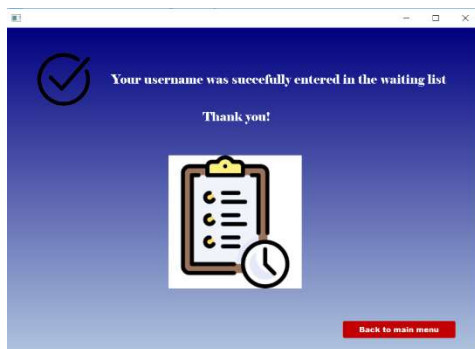
14. If a gate is totally reserved (all time slots are unavailable), the user can insert his ID in the gate waiting list by pressing on option 6 in the image shown in part 7.



In this example, gate 1 has a waiting list of size 7.

In order to be added to the waiting list, the client must make sure the gate is fully reserved, then insert the gate number and press on "Update the waiting list"

In case of successful update, the user will see:



### III. Databases at the server:

Table 1: clientinfo

+ Options			
username	email	password	ID
Nourhane	nma99@mail.aub.edu	magic	0
Nourhane	nma99@mail.aub.edu	magic	0
athena	nourhane.a.s@hotmail.com	magic	2529
n	n@gmail.com	magic	5260
noura	nour.a.s@hotmail.com	magic	3797
Client1	client1@hotmail.com	client1	1
Client2	client2@gmail.com	client2	2
Client3	Client3@gmail.com	client3	0
Client3	Client3@gmail.com	client3	3
client4	client4@hotmail.com	client4	4
client5	client5@hotmail.com	client5	5
client5	client5@hotmail.com	client5	5
client6	client6@gmail.com	client6	6
client7	client7@gmail.com	client7	7
marwan	marwan@mail.com	1964	5475
ne	nma@mail.aub	ne	3122
me	me@mail.aub	me	2387
Rim	Rim@hotmail.com	Rim	2053
nour	nour@mail.aub.rdu	n	5908
hadilas	hadil.a.s@outlook.com	marcoreus1989	4162
Sara	Sara@outlook.com	s	3072

Table 2: data

+ Options												
Gate Number	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4	Time Slot 5	Time Slot 6	Time Slot 7	Time Slot 8	Time Slot 9	Time Slot 10	Time Slot 11	Time Slot 12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	1	1	1	1	1	1	1	1	1
3	1	0	1	1	1	0	1	1	1	1	1	1
4	1	1	1	0	1	1	1	1	1	0	1	1
5	1	0	1	0	1	1	1	1	0	1	1	0
6	1	1	1	1	1	1	1	1	1	1	1	1
7	1	0	1	1	1	1	1	1	1	1	1	1
8	1	1	0	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1

Table 3: gates waiting list size

+ Options	
Gate_Number	Waiting_list_size
1	8
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0

Table 4: History

+ Options				
username	ID	Action	Date	Time
ne	3122	login	Friday, April 26, 2019	9:57 PM
ne	3122	search_gates	Friday, April 26, 2019	9:58 PM
ne	3122	reserve_gate_3_timeSlot_6	Friday, April 26, 2019	9:58 PM
ne	3122	search_slots	Friday, April 26, 2019	9:59 PM
ne	3122	search_gates	Friday, April 26, 2019	9:59 PM
ne	3122	logout	Friday, April 26, 2019	9:59 PM
ne	3122	login	Friday, April 26, 2019	10:27 PM
ne	3122	login	Friday, April 26, 2019	10:27 PM
ne	3122	login	Friday, April 26, 2019	10:30 PM
ne	3122	force_logout	Friday, April 26, 2019	10:30 PM
Nourhane	0	login	Friday, April 26, 2019	10:56 PM
Nourhane	0	force_logout	Friday, April 26, 2019	10:56 PM
Nourhane	0	login	Friday, April 26, 2019	10:58 PM
Nourhane	0	view_all_gates	Friday, April 26, 2019	10:58 PM
Nourhane	0	search_gates	Friday, April 26, 2019	10:58 PM
Nourhane	0	reserve_gate_5_timeSlot_9	Friday, April 26, 2019	10:59 PM
Nourhane	0	search_gates	Friday, April 26, 2019	10:59 PM
Nourhane	0	search_slots	Friday, April 26, 2019	11:00 PM
Nourhane	0	search_slots	Friday, April 26, 2019	11:00 PM
Nourhane	0	search_slots	Friday, April 26, 2019	11:00 PM
Nourhane	0	force_logout	Friday, April 26, 2019	11:02 PM
Nourhane	0	login	Friday, April 26, 2019	11:26 PM
Nourhane	0	logout	Friday, April 26, 2019	11:27 PM
Nourhane	0	login	Friday, April 26, 2019	11:29 PM
Nourhane	0	login	Friday, April 26, 2019	11:31 PM

Table 5: Reservation History

+ Options		
ClientUsername	GateReserved	TimeSlot
ne	3	6
Nourhane	5	9
ne	4	1
ne	1	1
ne	1	2
ne	1	3
ne	1	4
ne	1	5
ne	1	6
ne	1	7
ne	1	8
ne	1	9
ne	1	10
ne	1	11
ne	1	12
ne	2	1
ne	3	2
ne	4	10
ne	2	10
ne	7	10
ne	5	8
ne	10	8
ne	6	1
ne	6	3
ne	6	4

Table 6: Waiting list content

+ Options		
Username	ID	Gate
ne	3122	1
ne	3122	1
ne	3122	1
Nourhane	0	1
ne	3122	1
ne	3122	1

#### IV. Utility classes and functions:

1. Clock class (clock.java)

This class is responsible for formulating the date and time information.

It uses java's calendar class (java.util.calendar) and parses information using SimpleDateFormat (java.text.SimpeDateFormat).

It contains two methods that are called several times by the server class, which are getTimeSstring() and getDateString(). The former returns the hours and minutes and AM/PM as a single string, whereas the latter returns the day, month, and year (e.g. Friday April 20, 2019) as a single string. The serve uses these two strings to set a timestamp for the client's transactions and allocating them in the client's history database, to be viewed later on.

2. Server.ValidEmail(String email)

This utility method is found in the server class.

After parsing the client's email when attempting to create an account, the Server calls this function to validate the format of the email address.

This function check to see if "email" contains an '@' symbol, and a '.' symbol after it, and that the length of the email address is bigger than a certain threshold.

3. Server.RandomString()

This utility method is found in the server class.

It generates a random string of characters with random capitalization.

The server calls this function when the user is trying to create an account, to verify that the client is not a bot. (CAPTCHA)

If the client cannot read and understand the CAPTCHA, the server will generate another one sing this function, and will ask the client to write it correctly, and will keep doing so until the client is successful.