OPERATING SYSTEMS
TERM PROJECT

Name Surname: Büşra Medine GÜRAL

Student ID: 20011038

E-mail: medine.gural@std.yildiz.edu.tr

Introduction

Sockets and threads have been employed to address the issue. On the client side (within the main function), it obtains a user identifier (userid) from the user and establishes a connection to the server using this identifier. Subsequently, it presents a menu to the user and awaits the user to choose a specific operation. Depending on the user's selection, it sends an appropriate request to the server and prints the response received from the server to the screen.

On the server side (within the main function), it initializes the server socket (server_socket) and waits for incoming connections. The server socket is created using the socket system call with the arguments AF_INET (IPv4), SOCK_STREAM (TCP), and 0 (protocol). The server's address information, including the IP address and port number is specified using a struct sockaddr_in. The bind system call binds the socket to the server's address. Subsequently, the server listens for incoming connections using the listen system call with a specified connection queue size (5 in this case).

When a client connects, the server accepts the connection, and a separate thread is created for each client using the handle_client function. The server then prints a message indicating that it is listening on the specified port. This multithreaded approach allows the server to handle multiple client connections concurrently.The functions are explained in the next heading. Since the functions selected by the client are processed by the server, the server functions are explained in more detail.

Functions for the Client

sendChoice: Used to send a choice from the client. The choice represents the user's selection in the menu.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - int choice: An integer representing the choice to be sent.

sendUserId: Used to send a user identifier to the server.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

recieveAlertMessage: Used to receive an alert message from the server and print it to the screen.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *response_message: A character array to store the received alert message.

login: Initiates the login process. Sends the user identifier to the server and receives a response indicating whether the user is registered. If not registered, it prompts the user for additional information and sends it to the server.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

listContacts: Sends a request to the server to list the user's contacts and prints the response to the screen.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

addFriend: Sends a request to the server to add a friend, prompting the user for the friend's ID.

- Parameters:

- int client_socket: An integer representing the client socket.
- char *userid: A character array representing the user identifier.

deleteFriend:

- Parameters: Sends a request to the server to delete a friend, prompting the user for the friend's ID.
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

sendMessage: Sends a message to a friend by providing the recipient's ID and the message content.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

checkMessages: Checks and displays the messages for the user. The user selects a friend to check messages with.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

displayMessageHistory: Displays the message history with a specific friend, prompting the user to enter the friend's ID.

- Parameters:
  - int client_socket: An integer representing the client socket.
  - char *userid: A character array representing the user identifier.

Functions for the Server

handle_client: This function manages each client connection. It receives options from the client and performs various functions based on the received choice. It continues processing until the client signals an exit request, at which point it closes the connection.

- Parameters:
  - void *arg: Represents the argument passed to a thread, in this case, it signifies a client socket.

createAppDirectory: Creates the application directory and necessary directories for user folders. It does not recreate existing directories.

recieveUserId: Receives the user ID from the client through the specified socket.

- Parameters:
  - int client_socket: The socket used for communication.
  - char *userid: Buffer to store the received user ID.

sendAlertMessages: Sends alert messages to the client, such as warnings or information.

- Parameters:
  - int client_socket: The socket used for communication.
  - char *message: The message to be sent to the client.

handleListContacts: The function retrieves the user's contact list by reading the contents of the "friends.csv" file associated with the provided user ID. After extracting the user ID from the client, the function constructs the file path and attempts to open the friends file. Upon successful file access, it reads the data into a buffer and subsequently sends it to the client using the established socket connection.

- Parameters:

    - int client_socket: The socket used for communication.

saveUserToCSV: The function is designed to append a new user's information to a CSV file named "allUsers.csv." It takes a pointer to a User structure as a parameter, containing the user's details such as user ID, name, surname, and phone number. The function constructs the file path, opens the CSV file in append mode, and if successful, writes a new line with the user's information using the fprintf function. The information is formatted with commas as separators, aligning with the CSV file format.

- Parameters:
    - User *user: Pointer to a User structure containing user information.

searchUser: The function performs a search for a specific user ID within the "allUsers.csv" file. It reads each line of the file, extracting user IDs, and compares them to the provided userid. If a match is found, the function sets the found flag to 1, indicating that the user exists. The search process terminates once a match is found, or the entire file is traversed. The function returns 1 if the user is found and 0 otherwise.

- Parameters:
    - char *userid: The user ID to be searched.

handleLogin: The function oversees user login in a multi-user chat application. After receiving the user ID from the client, it checks the user's registration status using the searchUser function and communicates the result back to the client. If the user is new, the function creates a directory for them, stores additional information in an "info.csv" file, saves the user's details in the "allUsers.csv" file, and sends a success message. If the user is already registered, it sends a login confirmation message.

- Parameters:
    - int client_socket: The socket used for communication.

handleAddFriend: The function manages the process of adding a friend in the multi-user chat application. Initially, it receives the user ID and the friend's ID from the client. It then searches for the friend in the "allUsers.csv" file, validating their existence. If the friend is found, their details are appended to the user's "friends.csv" file, which maintains the user's list of friends. A success message is sent back to the client, notifying them that the friend has been added. Conversely, if the friend is not found, an alert message is sent to the client.

- Parameters:
    - int client_socket: The socket used for communication.

handleDeleteFriend: The function manages the process of deleting a friend in the multi-user chat application. It receives the user ID and the friend's ID from the client and proceeds to locate and delete the friend from the user's "friends.csv" file. The function reads each line from the original file, excluding the line containing the friend to be deleted, and writes the remaining lines to a temporary file. The original file is then updated by renaming the temporary file to "friends.csv".

- Parameters:
    - int client_socket: The socket used for communication.

handleSendMessage: The function is responsible for managing the process of sending a message between friends. It begins by receiving the message details from the client. The function then checks if the sender and receiver are friends by examining the "friends.csv" file of the sender. If the friends' connection is confirmed, the message is appended to a CSV file specific to the sender-receiver pair, and it is also added to the receiver's message buffer file. If the sender and receiver are friends, the message is successfully sent and recorded; otherwise, a failure message is sent, indicating that the users are not friends.

- Parameters:
    - int client_socket: The socket used for communication.

handleCheckMessage: The function begins by receiving the user ID from the client and subsequently reads the user's message buffer file. It then extracts and separates user IDs and corresponding messages from the buffer, printing the information for debugging. The function constructs a linear buffer containing user IDs and sends it to the client. The client is prompted to choose a message by entering the associated ID. The function searches for the selected ID in the array and, if found, sends the corresponding message back to the client and deletes the message from the user's buffer.

- Parameters:
  - int client_socket: The socket used for communication.

deleteMessageFromBuffer: The function is responsible for removing a selected message identified by its ID (selectedId) from the user's message buffer, associated with the given user ID (userId). The function reads the contents of the buffer file, identifies the target message, creates a modified buffer excluding the selected message, and writes the modified buffer back to the file.

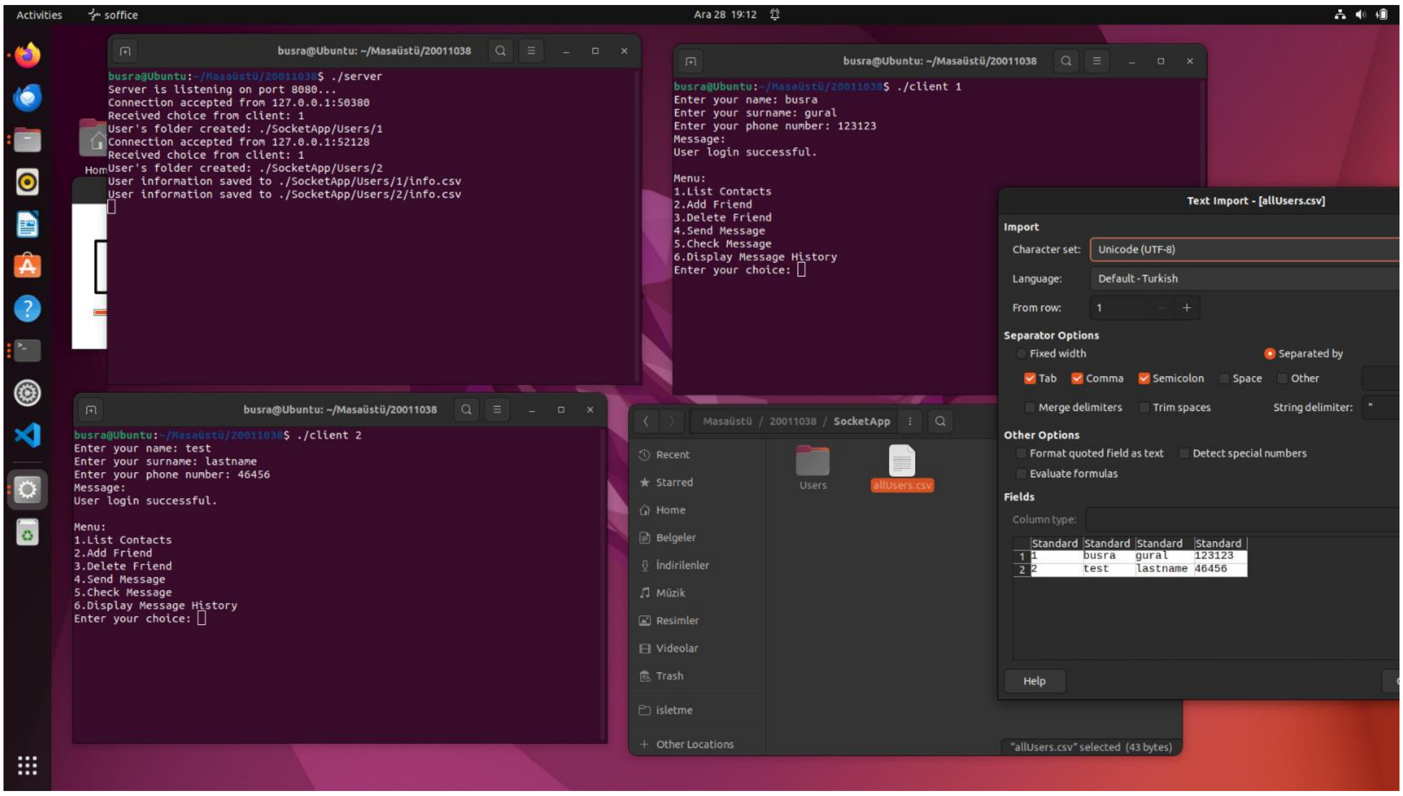
- Parameters:
  - char *userId: The user ID whose buffer is being checked.
  - char *selectedId: The ID of the friend whose message is being deleted.

handleDisplayMessageHistory: The function is designed to retrieve and send the message history between the user and a specific friend, identified by their respective user IDs (userid and friendId). It constructs the file path for the message history file, opens the file in read mode, reads its contents into a buffer (historyBuffer), and then sends this buffer to the client through the specified socket.

- Parameters:
  - int client_socket: The socket used for communication.

ScreenShots

Login:



```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
```

```
busra@Ubuntu:~/Masaüstü/20011038$ ./client 1
Enter your name: busra
Enter your surname: gural
Enter your phone number: 123123
Message:
User login successful.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 
```

```
busra@Ubuntu:~/Masaüstü/20011038$ ./client 2
Enter your name: test
Enter your surname: lastname
Enter your phone number: 46456
Message:
User login successful.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 
```

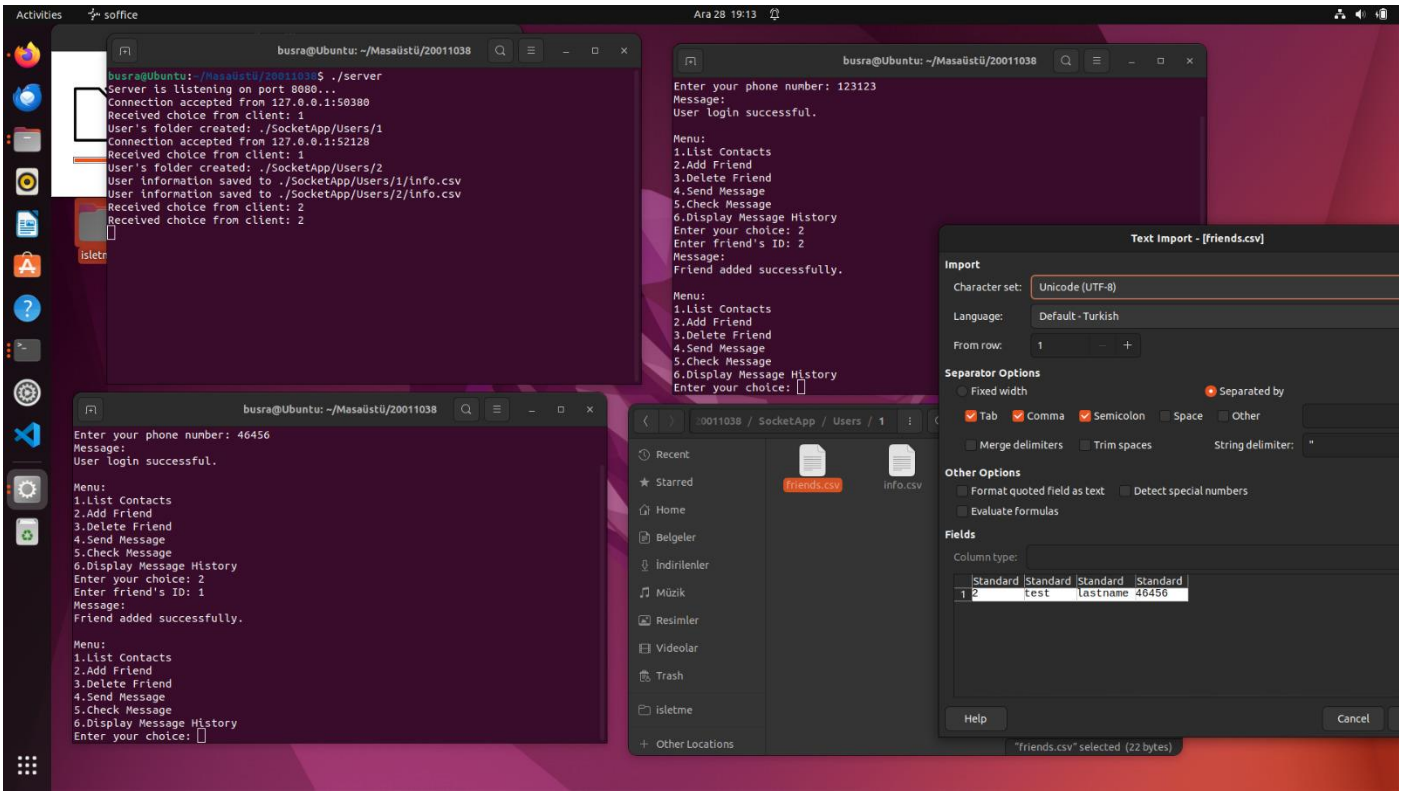
Text Import - [allUsers.csv]

Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1
Separator Options: [✓ Separated by] — [✓ Tab] [✓ Comma] [✓ Semicolon]
String delimiter: "

	Standard	Standard	Standard	Standard
1	1	busra	gural	123123
2	2	test	lastname	46456

"allUsers.csv" selected (43 bytes)

Adding Friend:



```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
```

```
Enter your phone number: 123123
Message:
User login successful.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 2
Enter friend's ID: 2
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
Enter your phone number: 46456
Message:
User login successful.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 2
Enter friend's ID: 1
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

Text Import - [friends.csv]

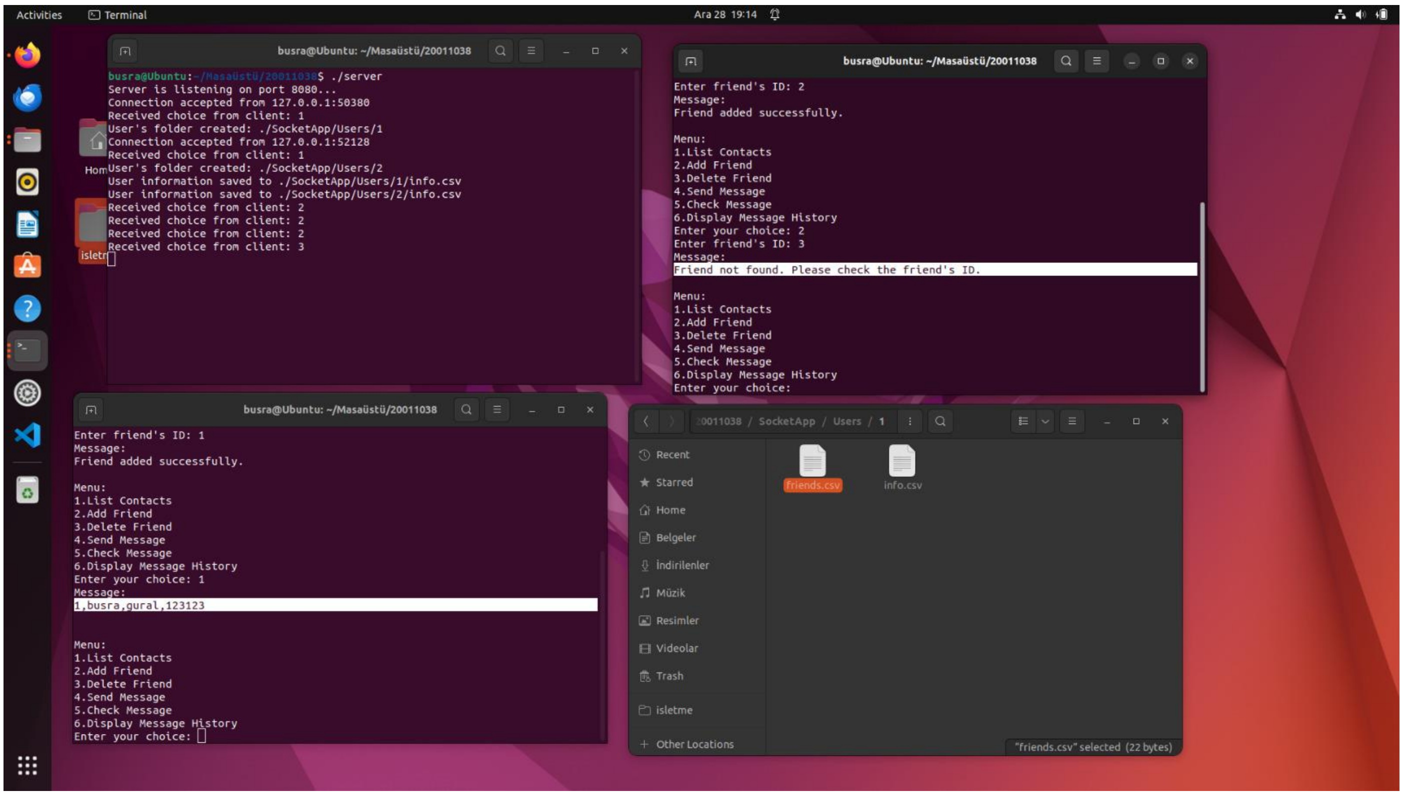
Import
Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1

Separator Options
Separated by: Tab ✓, Comma ✓, Semicolon ✓

Standard	Standard	Standard	Standard
2	test	lastname	46456

"friends.csv" selected (22 bytes)

Client 2 is listing its contacts. Client 1 is trying to add someone who is no in the system.



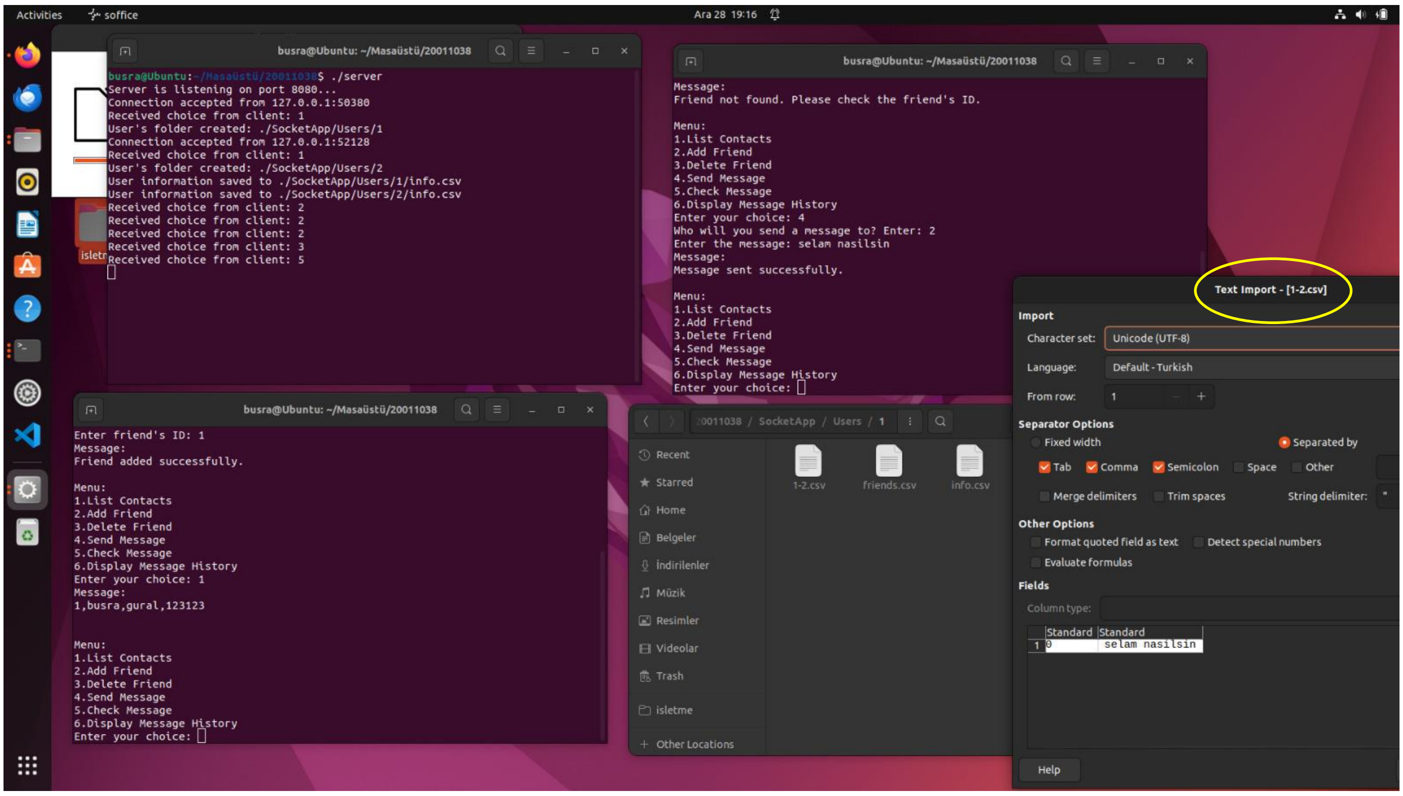
```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
```

```
Enter friend's ID: 2
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 2
Enter friend's ID: 3
Message:
Friend not found. Please check the friend's ID.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
Enter friend's ID: 1
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 1
Message:
1,busra,gural,123123

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

"friends.csv" selected (22 bytes)

Client 1 is sending a message to the Client 2.



```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
```

```
Message:
Friend not found. Please check the friend's ID.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 2
Enter the message: selam nasilsin
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
Enter friend's ID: 1
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 1
Message:
1,busra,gural,123123


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

Text Import - [1-2.csv]

Import
Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1

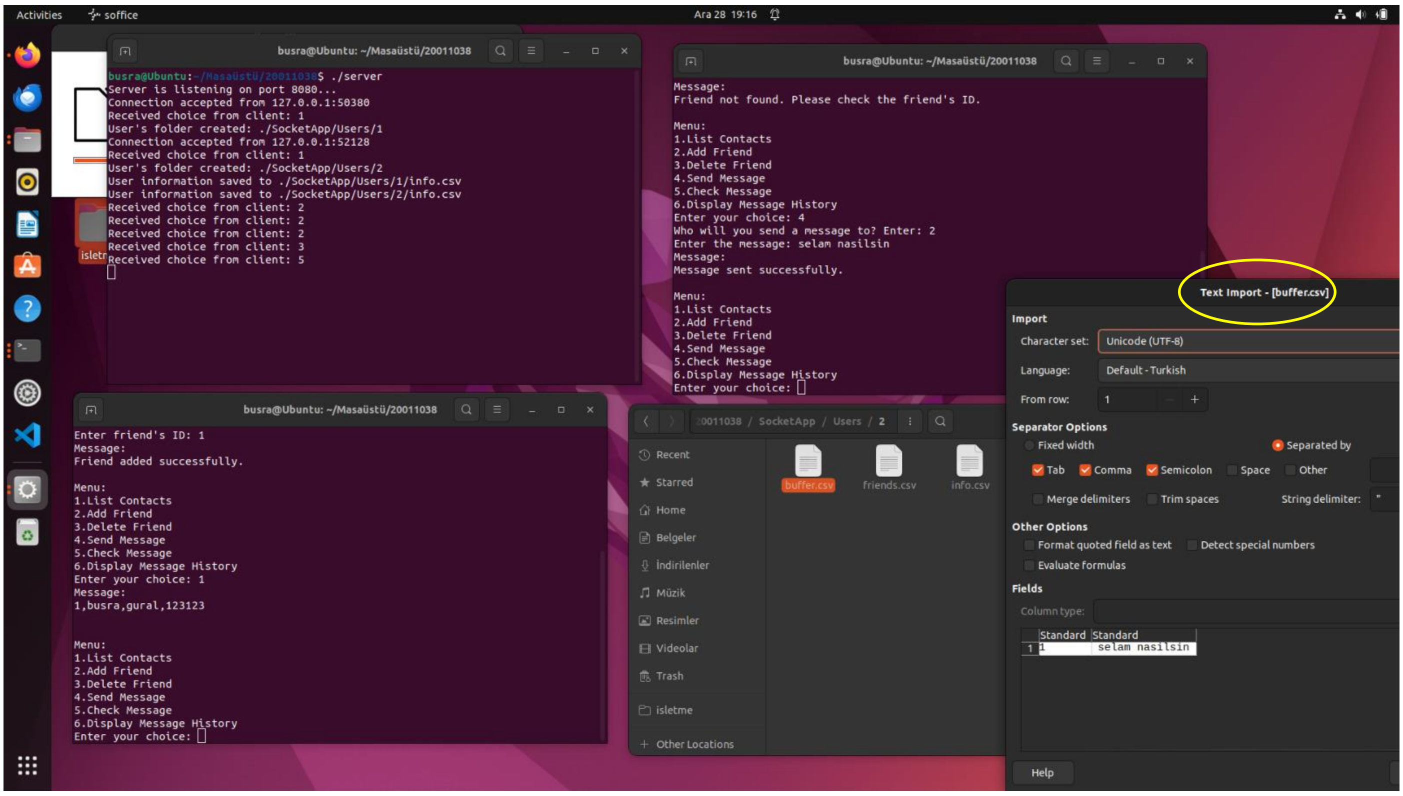
Separator Options
Fixed width [✓ Separated by]
[✓ Tab] [✓ Comma] [✓ Semicolon] Space Other
Merge delimiters Trim spaces String delimiter: "

Other Options
Format quoted field as text Detect special numbers
Evaluate formulas

Fields
Column type:

	Standard	Standard
1	0	selam nasilsin

First, the message is in the Client 2's buffer.



```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
```

```
Message:
Friend not found. Please check the friend's ID.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 2
Enter the message: selam nasilsin
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
Enter friend's ID: 1
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 1
Message:
1,busra,gural,123123


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

Text Import - [buffer.csv]

Import
Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1

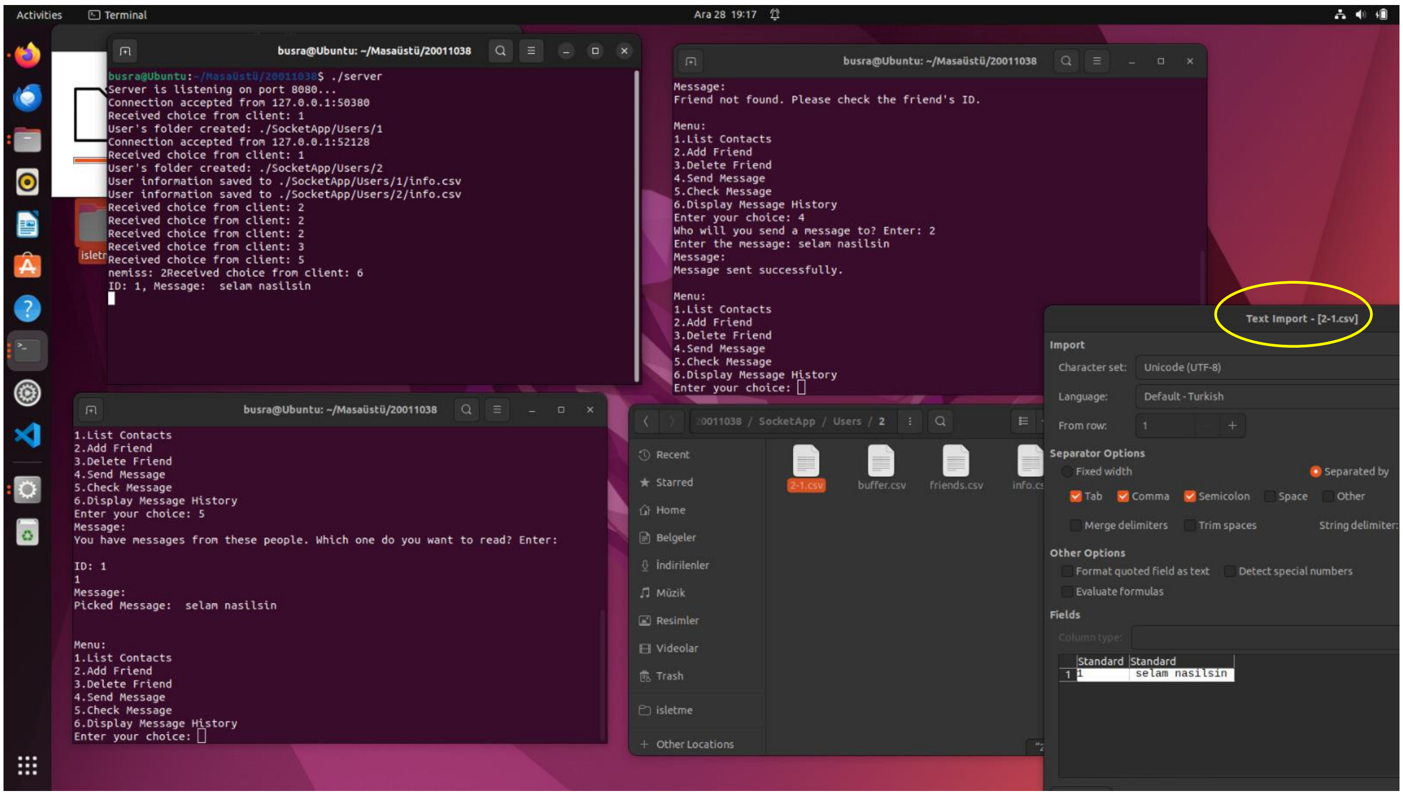
Separator Options
Fixed width [✓ Separated by]
[✓ Tab] [✓ Comma] [✓ Semicolon] Space Other
Merge delimiters Trim spaces String delimiter: "

Other Options
Format quoted field as text Detect special numbers
Evaluate formulas

Fields
Column type:

	Standard	Standard
1	1	selam nasilsin

Second, if Client 2 checks its message, the message is deleted from the buffer. It is transferred to the file containing the conversation history.



```
busra@Ubuntu: ~/Masaüstü/20011038

busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
nemiss: 2Received choice from client: 6
ID: 1, Message:  selam nasilsin
```

```
busra@Ubuntu: ~/Masaüstü/20011038

Message:
Friend not found. Please check the friend's ID.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 2
Enter the message: selam nasilsin
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
busra@Ubuntu: ~/Masaüstü/20011038

1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 5
Message:
You have messages from these people. Which one do you want to read? Enter:

ID: 1
1
Message:
Picked Message:  selam nasilsin


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

Text Import - [2-1.csv]

Import
Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1

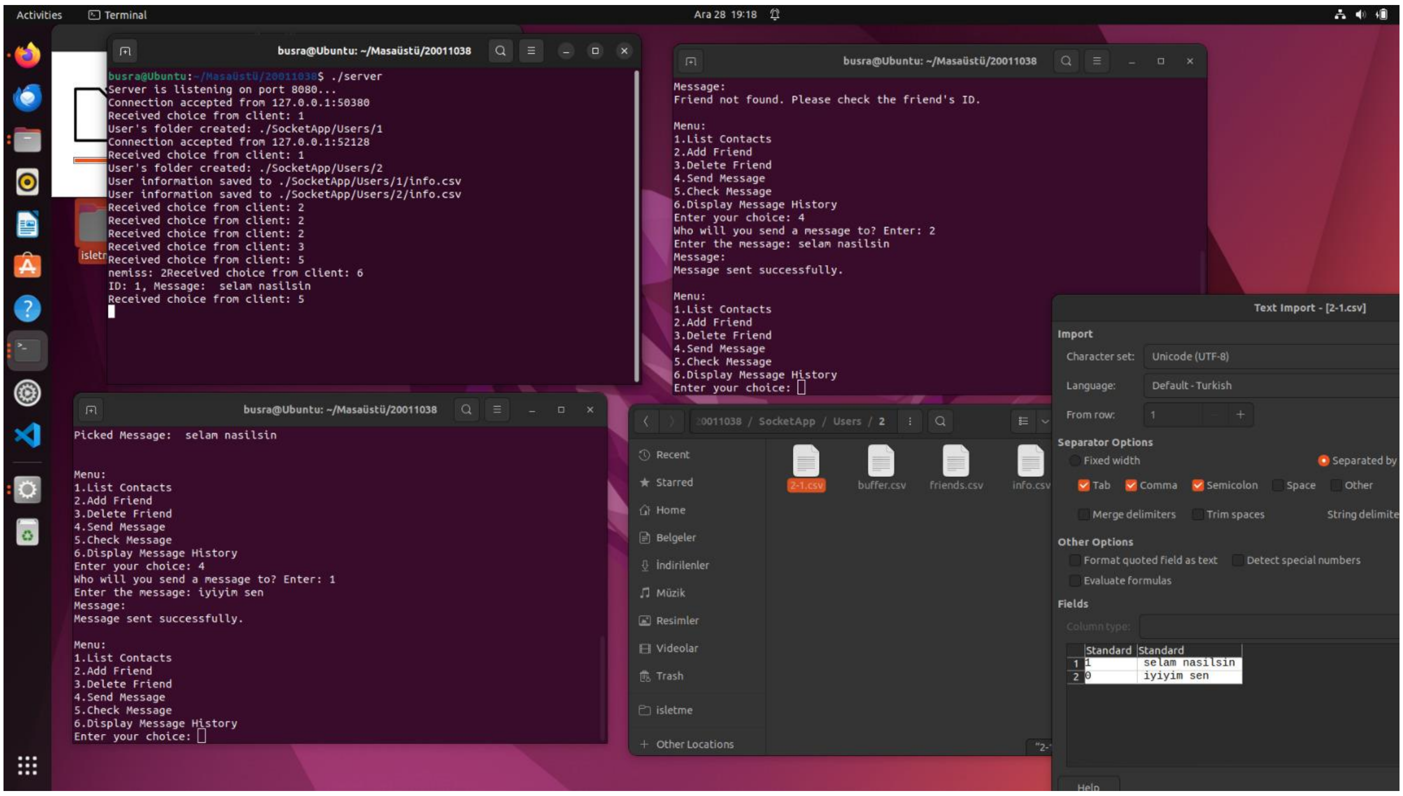
Separator Options
Fixed width
Separated by
Tab [✓] Comma [✓] Semicolon [✓] Space Other
Merge delimiters Trim spaces String delimiter

Other Options
Format quoted field as text Detect special numbers
Evaluate formulas

Fields
Column type:

	Standard	Standard
1	1	selam nasilsin

Client 2 is sending a message to the Client 1.



```
busra@Ubuntu: ~/Masaüstü/20011038

busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
nemiss: 2Received choice from client: 6
ID: 1, Message:  selam nasilsin
Received choice from client: 5
```

```
busra@Ubuntu: ~/Masaüstü/20011038

Message:
Friend not found. Please check the friend's ID.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 2
Enter the message: selam nasilsin
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
busra@Ubuntu: ~/Masaüstü/20011038

Picked Message:  selam nasilsin


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 1
Enter the message: iyiyim sen
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

Text Import - [2-1.csv]

Import
Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1

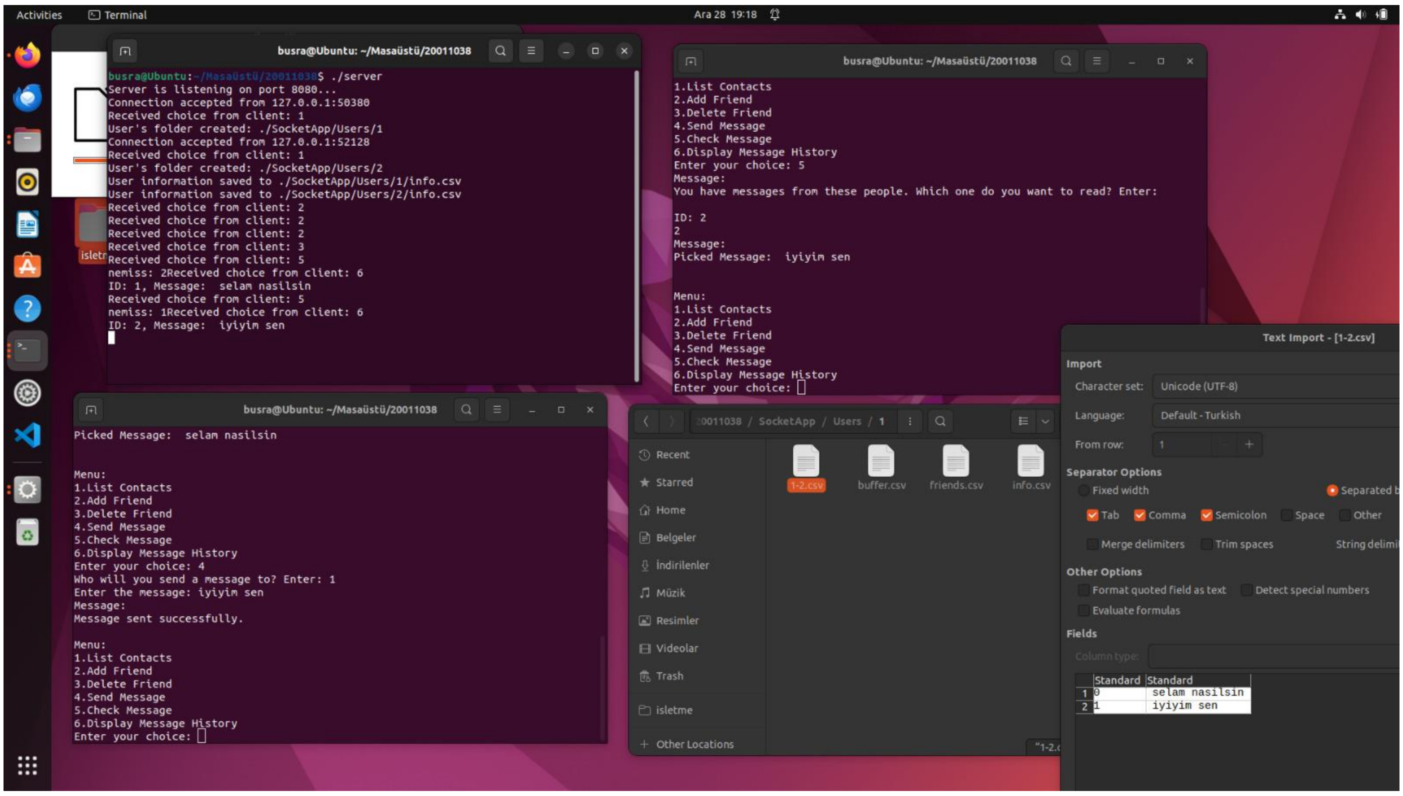
Separator Options
Fixed width
Separated by
Tab [✓] Comma [✓] Semicolon [✓] Space Other
Merge delimiters Trim spaces String delimite

Other Options
Format quoted field as text Detect special numbers
Evaluate formulas

Fields
Column type:

	Standard	Standard
1	1	selam nasilsin
2	0	iyiyim sen

Client 1 is checking its message. The message is transferred to the 1-2.csv from the buffer.



```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
nemiss: 2Received choice from client: 6
ID: 1, Message:  selam nasilsin
Received choice from client: 5
nemiss: 1Received choice from client: 6
ID: 2, Message:  iyiyim sen
```

```
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 5
Message:
You have messages from these people. Which one do you want to read? Enter:

ID: 2
2
Message:
Picked Message:  iyiyim sen


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
Picked Message:  selam nasilsin


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 1
Enter the message: iyiyim sen
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

:0011038 / SocketApp / Users / 1

1-2.csv buffer.csv friends.csv info.csv

Text Import - [1-2.csv]

Import
Character set: Unicode (UTF-8)
Language: Default - Turkish
From row: 1

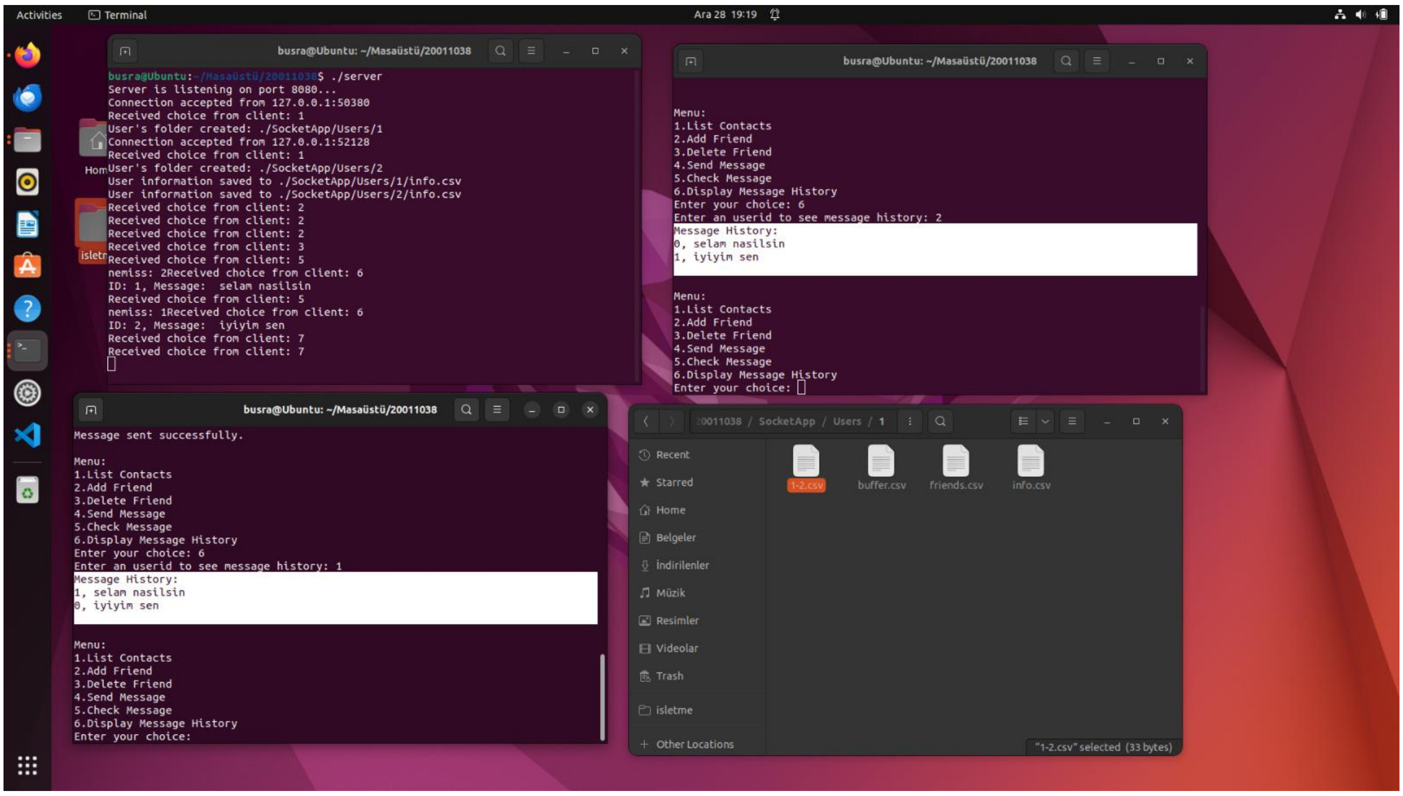
Separator Options
Fixed width   [✓ Separated b]
[✓ Tab] [✓ Comma] [✓ Semicolon] Space Other
Merge delimiters   Trim spaces   String delimi

Other Options
Format quoted field as text   Detect special numbers
Evaluate formulas

Fields

	Standard	Standard
1	0	selam nasilsin
2	1	iyiyim sen

Their message history (0 is sender, 1 is receiver):



```
busra@Ubuntu:~/Masaüstü/20011038$ ./server
Server is listening on port 8080...
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
nemiss: 2Received choice from client: 6
ID: 1, Message:  selam nasilsin
Received choice from client: 5
nemiss: 1Received choice from client: 6
ID: 2, Message:  iyiyim sen
Received choice from client: 7
Received choice from client: 7
```

```

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 6
Enter an userid to see message history: 2
Message History:
0, selam nasilsin
1, iyiyim sen


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

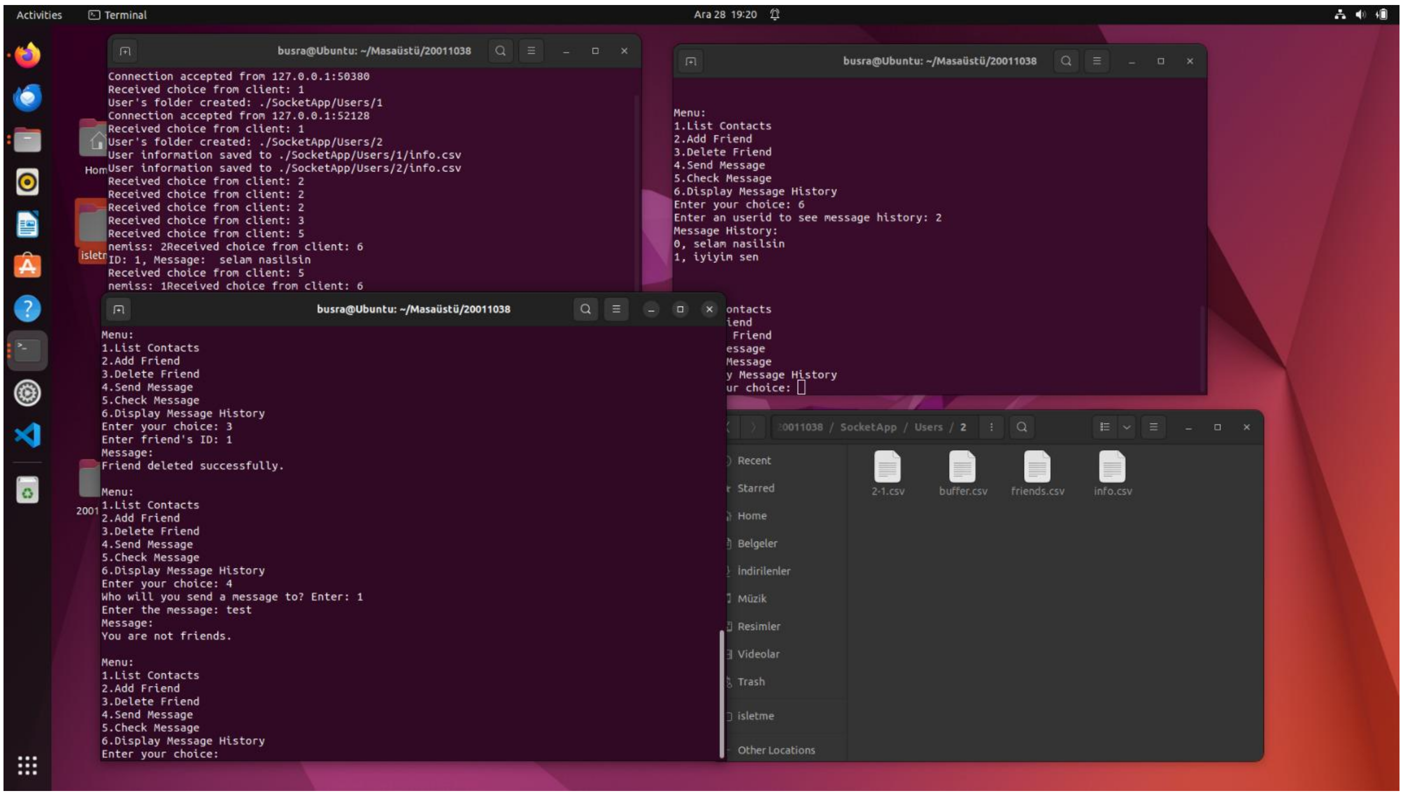
```
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 6
Enter an userid to see message history: 1
Message History:
1, selam nasilsin
0, iyiyim sen


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

:0011038 / SocketApp / Users / 1

1-2.csv buffer.csv friends.csv info.csv

"1-2.csv" selected (33 bytes)

Client 2 is deleting Client 1 and trying to send a message.

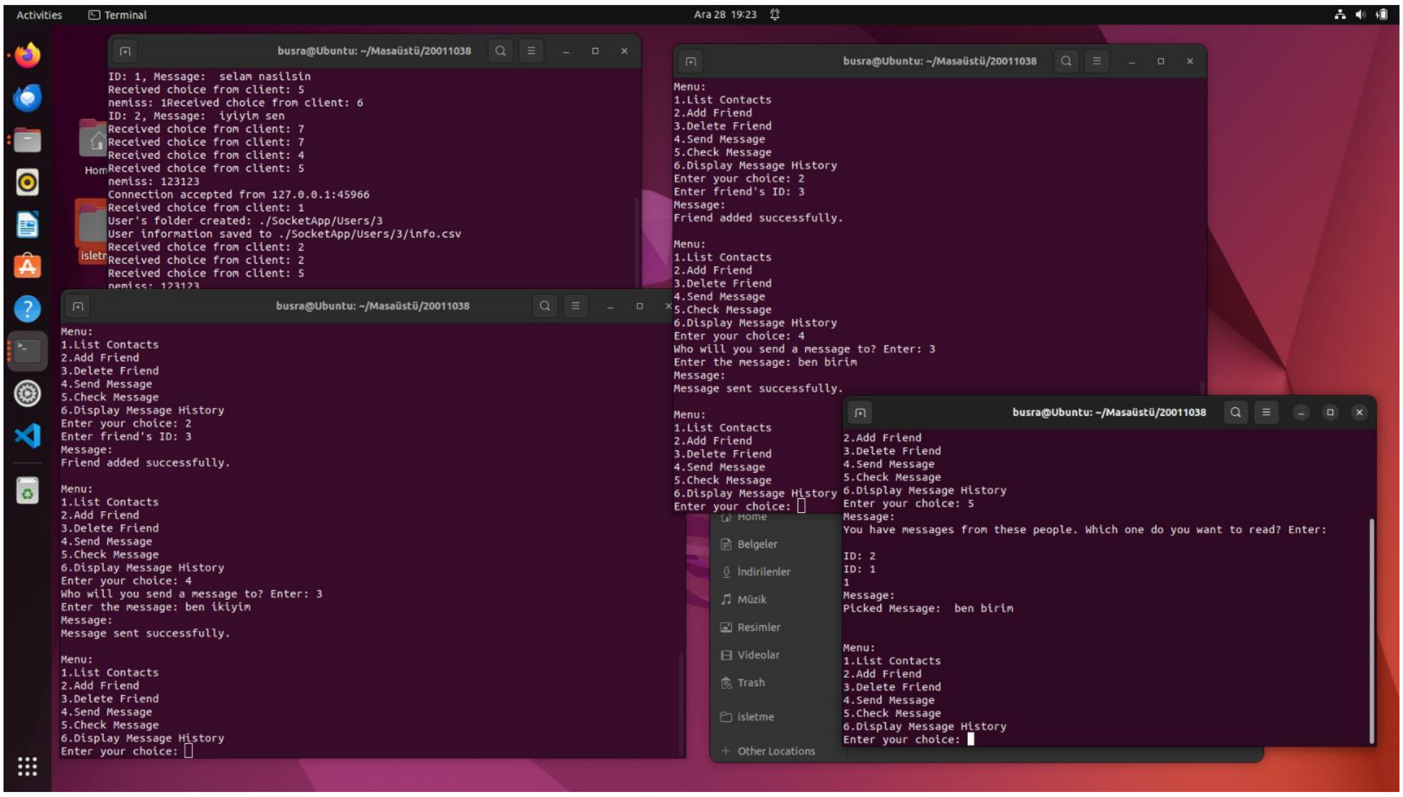


```
Connection accepted from 127.0.0.1:50380
Received choice from client: 1
User's folder created: ./SocketApp/Users/1
Connection accepted from 127.0.0.1:52128
Received choice from client: 1
User's folder created: ./SocketApp/Users/2
User information saved to ./SocketApp/Users/1/info.csv
User information saved to ./SocketApp/Users/2/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 2
Received choice from client: 3
Received choice from client: 5
nemiss: 2Received choice from client: 6
ID: 1, Message:  selam nasilsin
Received choice from client: 5
nemiss: 1Received choice from client: 6
```

```
Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 6
Enter an userid to see message history: 2
Message History:
0, selam nasilsin
1, iyiyim sen
```

```
Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 3
Enter friend's ID: 1
Message:
Friend deleted successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 1
Enter the message: test
Message:
You are not friends.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

Client 3 is the new user. Client 1 and Client 2 is adding the new user and sending message. Client 3 is checking and reading the message.



```
ID: 1, Message:  selam nasilsin
Received choice from client: 5
nemiss: 1Received choice from client: 6
ID: 2, Message:  iyiyim sen
Received choice from client: 7
Received choice from client: 7
Received choice from client: 4
Received choice from client: 5
nemiss: 123123
Connection accepted from 127.0.0.1:45966
Received choice from client: 1
User's folder created: ./SocketApp/Users/3
User information saved to ./SocketApp/Users/3/info.csv
Received choice from client: 2
Received choice from client: 2
Received choice from client: 5
nemiss: 123123
```

```
Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 2
Enter friend's ID: 3
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 3
Enter the message: ben ikiyim
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 2
Enter friend's ID: 3
Message:
Friend added successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 4
Who will you send a message to? Enter: 3
Enter the message: ben birim
Message:
Message sent successfully.

Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```

```
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice: 5
Message:
You have messages from these people. Which one do you want to read? Enter:

ID: 2
ID: 1
1
Message:
Picked Message:  ben birim


Menu:
1.List Contacts
2.Add Friend
3.Delete Friend
4.Send Message
5.Check Message
6.Display Message History
Enter your choice:
```