Client/Server/Database Use Case

# Client-Connect (Use Case)

## Successful Connection

1. Description: The user requests to connect to the Client GUI using an IP address
2. Pre Condition: The user has an internet connection
3. Post Condition: The user has connected to the Client GUI
   1. User enter ip address ‘127.0.0.0’
   2. User connects to Client GUI

## Unsuccessful Connection

1. Description: The user requests to connect to the Client GUI using an IP address
2. Pre Condition: The User has an internet connection
3. Post Condition: The User has not connected to the Client GUI
   1. User enter ip address ‘127.0.0.1’
   2. User does not connect to Client GUI

# Client-Disconnect (Use Case)

## Successful Disconnect

1. Description: The user requests on the Client GUI to disconnect from the server
2. Pre Condition: The user is already connected to the Client GUI
3. Post Condition: The client has disconnected from the Client GUI
   1. User selects the [Disconnect] button on Client GUI
   2. User disconnects from Client GUI

# UserDB - Log in (Use Case)

## Successful Log in

1. Description: The server receives a log in request from a user in the Client GUI and cross checks information to log in the user.
2. Pre Condition: The user has selected the Log In button inside of the Client GUI
3. Post Condition: The user was successfully logged in by the server
   1. Server receives login request with username ‘justinbouse2’ password ‘password1’ and email ‘justinbouse14@gmail.com
   2. Server checks User Database to see if the username justinbouse2 exists
   3. It exists so it will make sure password and email match
   4. Server switches user to a logged in state

## Unsuccessful Log in

1. Description: The server receives a log in request from a user in the Client GUI and cross checks information to log in the user.
2. Pre Condition: The user has selected the Log In button inside of the Client GUI
3. Post Condition: The user was not logged in by the server
   1. Server receives login request with username ‘justinblouse2’ password ‘password1’ and email ‘justinbouse14@gmail.com
   2. Server checks User Database to see if the username justinblouse2 exists
   3. It does not exist
   4. Sever outputs message for client that the information does not match

# UserDB - New User Registration(Use Case)

## Successful New User Registration

1. Description: The server receives a request to Create a new user will check username to see if it already exists and ensure the minimum requirements are met
2. Pre Condition: The user has selected the New User button inside of the Client GUI
3. Post Condition: The server has issued a statement that the new user has been created and stored the username password and email address
   1. Server receives New user request with username ‘justinbouse4’ password ‘password1’ and email ‘[justinbouse14@gmail.com](mailto:justinbouse14@gmail.com)
   2. Sever receives username password and email with minimum requirements
   3. Server checks User Database to see if the username justinbouse4 exists
   4. It does not exist
   5. Server saves username password and email address as a new account and displays a message on client gui that your account has been created

## New User Registration User already exists

1. Description: The server receives a request to Create a new user will check username to see if it already exists and ensure the minimum requirements are met
2. Pre Condition: The user has selected the New User button inside of the Client GUI
3. Post Condition: The server has issued a statement that the username already exists
   1. Server receives New user request with username ‘justinbouse2’ password ‘password1’ and email ‘[justinbouse14@gmail.com](mailto:justinbouse14@gmail.com)
   2. Sever receives username password and email with minimum requirements
   3. Server checks User Database to see if the username justinbouse2 exists
   4. It does exist
   5. Server sends message on Client GUI that the username already exists

## New User Registration User invalid information

1. Description: The server receives a request to Create a new user will check username to see if it already exists and ensure the minimum requirements are met
2. Pre Condition: The user has selected the New User button inside of the Client GUI
3. Post Condition: The server has issued a statement that the information entered does not meet minimum requirements
   1. Server receives New user request with username ‘justinbouse2’ password ‘password1’ and email ‘[justinbouse14@gmail.com](mailto:justinbouse14@gmail.com)
   2. Sever receives username password and email with minimum requirements
   3. Server checks User Database to see if the username justinbouse2 exists
   4. It does exist
   5. Server sends message on Client GUI that the username already exists

# UserDB - Password Recovery(Use Case)

## Password Recovery request

1. Description: The server receives a password recovery request from a user in the Client GUI and sends a new generated password to that email and changes the one associated with that username
2. Pre Condition: The user has selected the Password recovery button inside of the Client GUI
3. Post Condition: A recovery password was sent to the User’s email
   1. Server receives password recovery request with username ‘justinbouse2’
   2. Server checks User Database to see if the username justinbouse2 exists
   3. It exists so it generates a new password and sends it to the email address associated with that username
   4. Server sends message to be displayed in Client GUI ‘If a username exists your new password has been sent to the associated email’

# Client - Log in(Use Case)

## Successful Log in

1. Description: The User requests to Log in to an account and enters a username, password, and email.
2. Pre Condition: The User has successfully connected to the server and a thread has started for that socket connection
3. Post Condition: The User has successfully logged in to the Client GUI with an account and is in a [logged in state]
4. Story:
   1. User clicks button [Log in]
   2. User enters username: ‘justinbouse2’
   3. User enters password: ‘password1’
   4. User enters email: justinbouse14@gmail.com
   5. User receives message on Client GUI log in successful
   6. User GUI is in a logged in state

## Unsuccessful Login

1. Description: The User requests to Log in to an account and enters a username, password, and email.
2. Pre Condition: The User has successfully connected to the server and a thread has started for that socket connection
3. Post Condition: The User not logged into the Client GUI with a user account remains in Non-Logged in state
4. Story:
   1. Client clicks button [Log in]
   2. Client enters username: ‘justinbouse2’
   3. Client enters password: ‘password1234’
   4. Client enters email: justinbouse14@gmail.com
   5. Client receives message on Client GUI Invalid username and password combination

## Login attempts with Lock out

1. Description: The client requests to Log in to an account and enters a username and password
2. Pre Condition: The client has successfully connected to the server and a thread has started for that socket connection
3. Post Condition: The user has has been locked out
4. Story:
   1. Client joins server using server ip address
   2. Client clicks button [Log in]
   3. Client enters username: ‘justinbouse2’
   4. Client enters password: ‘password’
   5. Clients reads output saying invalid username and password combination
   6. Client enters username: ‘justinbouse2’
   7. Client enters password: ‘passworD’
   8. Clients reads output saying invalid username and password combination
   9. Client enters username: ‘justinbouse2’
   10. Client enters password: ‘password2’
   11. Client enters username: ‘justinbouse2’
   12. Client enters password: ‘password3’
   13. Client reads output saying your username has been locked out use password recovery to access your account

# Client-New User Registration(Use Case)

## Account creation success

1. Description: The user requests to register an account and enters a username, password and email address. Client sees output successful account creation
2. Pre Condition: The user has successfully connected to the CLIENT GUI and a thread has started for that socket connection
3. Post Condition: An account with the users information has successfully been created and the information stored
4. Story:
   1. User joins server using server ip address
   2. User clicks button [Create Account]
   3. User enters username: ‘justinbouse2’
   4. User enters password: ‘password1’
   5. User enters emails: justinbouse14@gmail.com
   6. User reads output on “Account creation successful”

## Unsuccessful Account creation

1. Description: The client requests to create an account and enters a username, password and email address. Client is not able to create an account
2. Pre Condition: The user has successfully connected to the server and a thread has started for that socket connection
3. Post Condition: An account with the information entered by the user has not been created due to minimum requirements not being met
4. Story:
   1. User joins server using server ip address
   2. User clicks button [Create Account]
   3. User enters username: ‘justinbouse2’
   4. User enters password: ‘pass’
   5. User enters emails: justinbouse14@gmail.com
   6. User reads output ‘minimum requirements for account information not met’

## Account already exists with username entered

1. Description: The user requests to create an account and enters a username, password and email address. Client sees output that account with username already exists
2. Pre Condition: The client has successfully connected to the server and a thread has started for that socket connection
3. Post Condition: An account with the user information has not been created
4. Story:
   1. User joins server using server ip address
   2. User clicks button [Create Account]
   3. User enters username: ‘justinbouse2’
   4. User enters password: ‘password1’
   5. User enters emails: justinbouse14@gmail.com
   6. User reads output saying that the username already exists

# Client-Logout (Use Case)

## Client Logout

1. Description: The user requests to to Logout of the CLIENT GUI
2. Pre Condition: The user is connected to the Client GUI and in a logged in state
3. Post Condition: The user is in a logged out state
4. Story:
   1. User clicked [log out] button
   2. User is returned to a logged out state

# Client-Disconnect(Use Case)

## Client Disconnect

1. Description: The user requests to to Logout of the CLIENT GUI
2. Pre Condition: The user is connected to the Client GUI
3. Post Condition: The Client has logged out of the CLIENT GUI
4. Story:
   1. User clicked [log out] button
   2. User is returned to a logged out state

# Client-Password Recovery(Use Case)

## Password Recovery Request

1. Description: The user requests toRecover a Password enters in their username and a randomly generated password is sent to the usernames’ associated email
2. Pre Condition: The user has selected the Recover my password button and entered a username
3. Post Condition: An account with the users information has successfully been created and the information stored
4. Story:
   1. User joins server using server ip address
   2. User clicks button [Create Account]
   3. User enters username: ‘justinbouse2’
   4. User enters password: ‘password1’
   5. User enters emails: justinbouse14@gmail.com
   6. User reads output on “Account creation successful”

# Server- Query (Use Cases)

## Query Registered Users

1. Description: The server will receive a request to check the amount of users that are registered.
2. Pre Condition: Server receives total registered users request.
3. Post Condition: Server returns a message with number of registered users.
4. Story:
   1. Admin logs on to the server.
   2. Admin sends the server a request to count the number of registered users and return the number.
   3. Server connects to the database and gets the number of registered users.
   4. Server sends the admin a message with the number.

## Query Logged-in Active Users

1. Description: The server will receive a request to check the amount of users that are currently logged in.
2. Pre Condition: Server receives total logged-in users request.
3. Post Condition: Server returns a message with the number of active logged-in users.
4. Story:
   1. Admin logs on to the server.
   2. Admin sends the server a request to count the number of active logged-in users and return the number.
   3. Server connects to the database and gets the number of active logged-in users.
   4. Server sends the admin a message with the number.

# 

## Query Logged-in inactive Users

1. Description: The server will receive a request to check the amount of users that are currently logged in.
2. Pre Condition: Server receives total logged-in users request.
3. Post Condition: Server returns a message with the number of inactive logged-in users.
4. Story:
   1. Admin logs on to the server.
   2. Admin sends the server a request to count the number of inactive logged-in users and return the number.
   3. Server connects to the database and gets the number of inactive logged-in users.
   4. Server sends the admin a message with the number.

# 

## Query Users Locked Out

* 1. Description: The server will receive a request to check the amount of users that are have been locked out.
  2. Pre Condition: Server receives total locked-out users request.
  3. Post Condition: Server returns a message with the number of locked-out users.
  4. Story:
     1. Admin logs on to the server.
     2. Admin sends the server a request to count the number of locked-out users and return the number.
     3. Server connects to the database and gets the number of locked-out users.
     4. Server sends the admin a message with the number.

## Query Connected Users

* 1. Description: The server will receive a request to retrieve information about users currently connected to the system.
  2. Pre Condition: Server receives a request for connected users information.
  3. Post Condition: Server returns a detailed message with specific information about the users currently connected.
  4. Story:
     1. Admin logs on to the system.
     2. Admin initiates request to the server to query detailed information about users who are currently connected.
     3. Server receives the request and validates the user credentials.
     4. The server connects to the database upon successful validation and collects usernames of users connected.
     5. Server compiles the information into a message and the admin receives information about connected users.