Go team Squid Squad: Ipsilon Networks

Use Cases and User Stories :

(Please turn in what use cases/ user stories you would like to work on)

Use Cases to create stories on:

- User registration:Jimmy
- Login: Jacob
- Logout: Jimmy
- Password recovery: Kyler
- Connect to server: Tyler
- Disconnect from server: Tyler
- Change password: Jacob
- Admin queries: Kyler

Login cases (Jacob): Successful, Incorrect Username, Incorrect Password, Forgot username, Forgot password
Change password (Jacob):

Use Cases: Login

Successful

- Pre-condition - Connected to server
- Post condition - Ready to use server
- Story
    1) User inputs username
    2) User inputs password
    3) User presses login
    4) Server checks records from database to see if username exists → it does
    5) Server checks to see password matches record in database → it does
    6) Resets incorrect login to 0
    7) End of story

Incorrect Username

- Pre-condition - Connected to server
- Post-condition - Ready to use server
- Story
    1) User inputs username
    2) User inputs password

3) User presses login
4) Server checks records from database to see if username exists → it does not
5) Error message “Incorrect username”
6) End of story

Incorrect Password

- Pre-condition - Connected to server
- Post-condition - Ready to use server
- Story
  1) User inputs username
  2) User inputs password
  3) User presses login
  4) Server checks records from database to see if username exists → it does
  5) Server checks records from database to see if password matches → it does not
  6) Error message “Incorrect password”
  7) End of story

Forgot Username

- Pre-condition - Connected to server
- Post-condition - Ready to use server
- Story
  1) User clicks “Forgot username”
  2) Asks user for email
  3) User types email address
  4) Sends recovery email to user
  5) User changes username
  6) User prompted to login
  7) End of story

Use Case: Change password

Forgot Password

- Pre-condition - Connected to server
- Post-condition - Ready to use server
- Story
  1) User clicks “Forgot password”
  2) Asks user for email
  3) User types email address
  4) Sends recovery email to user

5) User changes and confirms password
6) User prompted to login
7) End of story

Connect to Server cases (for Tyler): Successful, Interrupted/Stopped, Timed Out, Not Found (Or whatever the statement would be if the servers are down, may just be timed out gotta check), check for other client side and server sided issues regarding IP connection

Tyler use cases: Connect to Server, Disconnect to server:

Use Cases: Connect to Server

User times out

Pre-condition – Username & password input (will attempt server connection regardless of input validity)

• Post-condition – Connection made to server

• Story

- 1a) User selects “login”
- 2a) Connection attempt made...
- 3a) Server takes too long to respond/connect, timed out
- End of story

User loses connection

Pre-condition – Username & password input (will attempt server connection regardless of input validity)

• Post-condition – Connection made to server

• Story

- 1b) User selects “login”
- 2b) Connection attempt made...
- 3b) User loses internet/connection
- 4b) Connection failed
- End of story

User encounters firewall related issue

Pre-condition – Username & password input (will attempt server connection regardless of input validity)

• Post-condition – Connection made to server

• Story

- 1c) User selects “login”
- 2c) Connection attempt made...
- 3c) Connection refused (firewall block)
- End of Story

User attempts to connect to unavailable server(s)
Pre-condition – Username & password input (will attempt server connection regardless of input validity)
• Post-condition – Connection made to server
• Story

- 1d) User selects “login”
- 2d) Connection attempt made...
- 3d) Server is unavailable
- 4d) Connection failed
- End of story

Blacklisted IP address attempts to connect
Pre-condition – Username & password input (will attempt server connection regardless of input validity)
• Post-condition – Connection made to server
• Story

- 1e) User selects “login”
- 2e) Connection attempt made...
- 3e) IP address is restricted
- 4e) User’s IP address has been blacklisted, cannot connect.
- End of story

User successfully connects to server
Pre-condition – Username & password input (will attempt server connection regardless of input validity)
• Post-condition – Connection made to server
• Story

- 1f) User selects “login”
- 2f) Connection attempt made...
- 3f) User successfully establishes connection w/ server
- 4f) User is connected to server
- End of story

User attempts to connect during maintenance

Pre-condition – Username & password input (will attempt server connection regardless of input validity)

• Post-condition – Connection made to server

• Story

- 1g) User selects “login”
- 2g) Connection attempt made...
- 3g) Server is under maintenance, cannot connect till over.
- 4g) Connection failed
- End of story

User to server connection routing path is too long, cannot connect/times out

Pre-condition – Username & password input (will attempt server connection regardless of input validity)

• Post-condition – Connection made to server

• Story

- 1h) User selects “login”
- 2h) Connection attempt made...
- 3h) Server routing to desired server is taking too long/path not found
- 4h) Connection failed
- End of story

Use Cases: Disconnect From Server

User exits software/logs off

Pre-condition – User is connected to server

• Post-condition – User is disconnected from server

• Story

- 1a) User logs off and/or closes the software
    - 1b If overriding task on hand… handle and continue or exit out of the disconnect path.
- 2a) Disconnect user from server
- End of story

User is disconnected by Admin, Server restart/disconnect, or loss of internet

Pre-condition – User is connected to server

• Post-condition – User is disconnected from server

• Story

- 1b) User loses connection
    - 1c) Depending on loss of connection type, may wait to reconnect, if fails/too long continue to 2b.
- 2b) Disconnect user from server

- End of story

Kyler Use Cases: Password Recovery, Admin Queries

Use Case: Password Recovery

User Submits A Valid Username

- precondition - Account is locked
- Postcondition - System is ready to deploy email
- Story
  - Input is a Username in the user database
  - Check if username is in user database (it is)
  - Send email to email address linked with the username
  - End of story

User Submits An Invalid Username

- precondition - Account is locked
- Postcondition - System is ready to deploy error message
- Story
  - Input is a Username that is not in the user database
  - Check if username is in the user database (it is not)
  - Formulate error message for wrong username
  - Return error message
  - End of story

Use Case: Admin Queries

Query number of registered users

- precondition - Server is running
- Postcondition - ready for output
- Story
  - Admin inputs call for number of registered users
  - Query user database for all registered users
  - Count amount of registered users
  - Return calculated number of registered users
  - End of Story

Query number of logged in users

- precondition - server is running
- Postcondition - ready for output
- Story
  - Admin inputs call for number of logged in users
  - Query userdata base for all logged in users
  - Count amount of logged in users
  - Return calculated number of logged in users
  - End of Story

Query which users are logged in

- precondition - server is running
- Postcondition - ready for output
- Story
  - Admin inputs call for which users are logged in
  - Query user database for logged in users
  - Organize list of users that are logged in
  - Return formatted list of all users that are logged in
  - End of Story

Query which users locked out

- precondition - server is running
- Postcondition - ready for output
- Story
  - Admin inputs call for which users are locked out
  - Query user database for all users that are locked out
  - Organize list of users that are locked out
  - Return formatted list of all users that are locked out
  - End of Story

Query number of connected users

- precondition - server is running
- Postcondition - ready for output
- Story
  - Admin inputs call for number of connected users
  - Query user database for all users that are connected
  - Count amount of connected users
  - Return calculated number of connected users
  - End of Story

Use case: User registration(Jimmy)

Username is already in use

- precondition - server is running
- Postcondition - ready for output
- Story -
    1. User select registration link
    2. User type of username of his choice
    3. User is request to select another username and password.
    4. End of story

User enter incorrect form of password

- precondition - server is running
- Postcondition - ready for output
- Story -
    1. User select registration link
    2. User type of username of his choice
    3. User types a password
    4. User is request to change password
    5. End of sotry

The two password are different

- precondition - server is running
- Postcondition - ready for output
- Story -
    6. User select registration link
    7. User type of username of his choice
    8. User types a password
    9. User retypes a password
    10. User is request to retype the password
    11. End story

User enter incorrect email address

- precondition - server is running
- Postcondition - ready for output
- Story -

    1. User select registration link
    2. User type of username of his choice
    3. User types a password
    4. User retypes a password
    5. User type email
    6. User is request to retype email

User email can not verify

- precondition - server is running
- Postcondition - ready for output
- Story -
    1. User select registration link
    2. User type of username of his choice
    3. User types a password
    4. User retypes a password
    5. User type email
    6. Send a confirmation number to the email
    7. Request rend another confirmation number
    8. End story

User enter correct information

- precondition - server is running
- Postcondition - ready for output
- Story -
    1. User select registration link
    2. User type of username of his choice
    3. User types a password
    4. User retypes a password
    5. User type email
    6. Send a confirmation number to the email
    7. Verify the email address
    8. Submit
    9. System checks if the user name is not already in use
    10. System checks if the two passwords are identical
    11. System registers the new player with the given parameters (user name, password,email)

Use case: Log out

User successfully log out

- precondition - server is running
- Postcondition - ready for output
- Story -

  1. User select the menu button
  2. User select the log out
  3. End of story

User enter wrong password

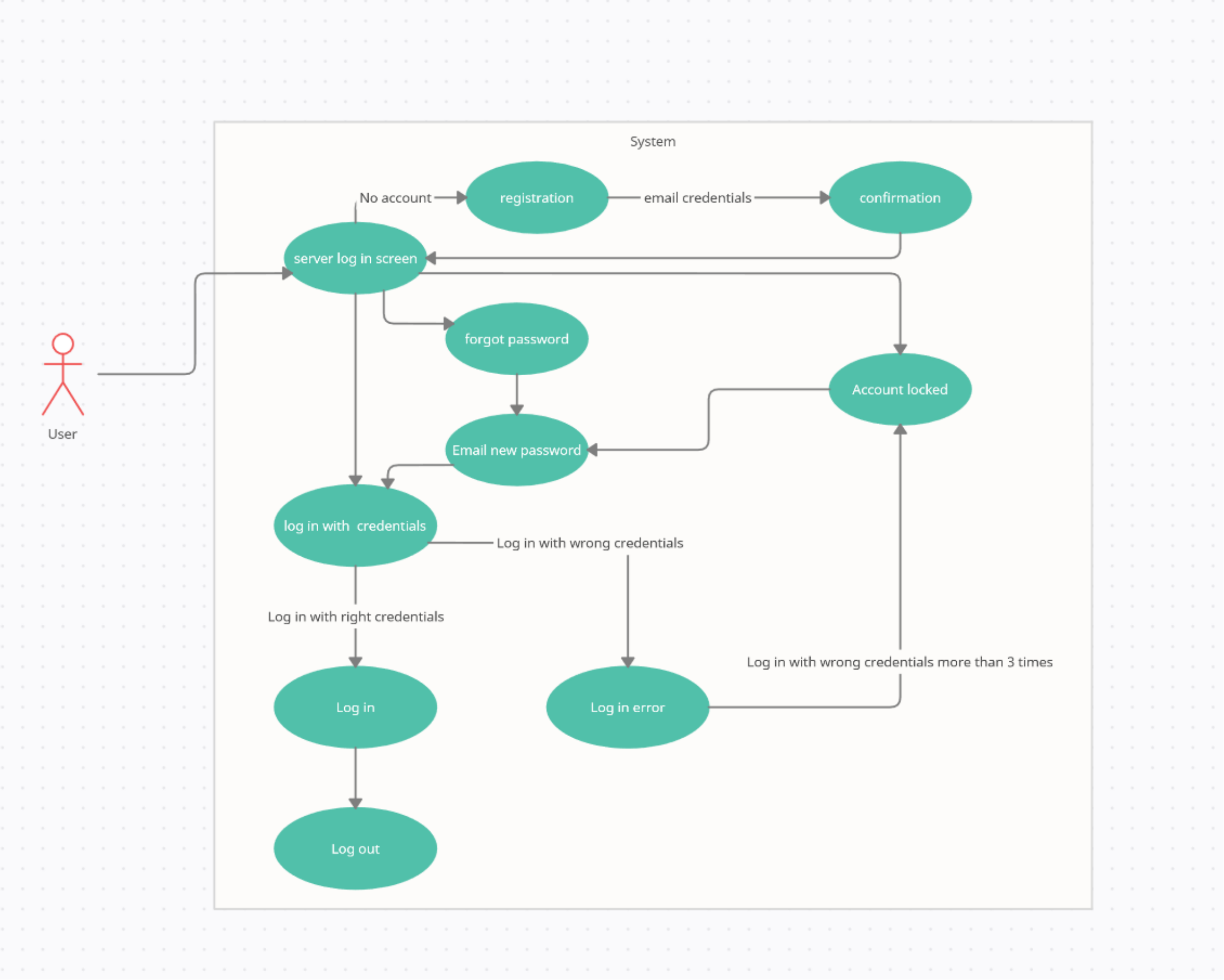
- precondition - server is running
- Postcondition - ready for output
- Story -
  1. User enter the user name
  2. User enter the incorrect password
  3. System automatically log out the account
  4. End of story

User logged in after 5 hours

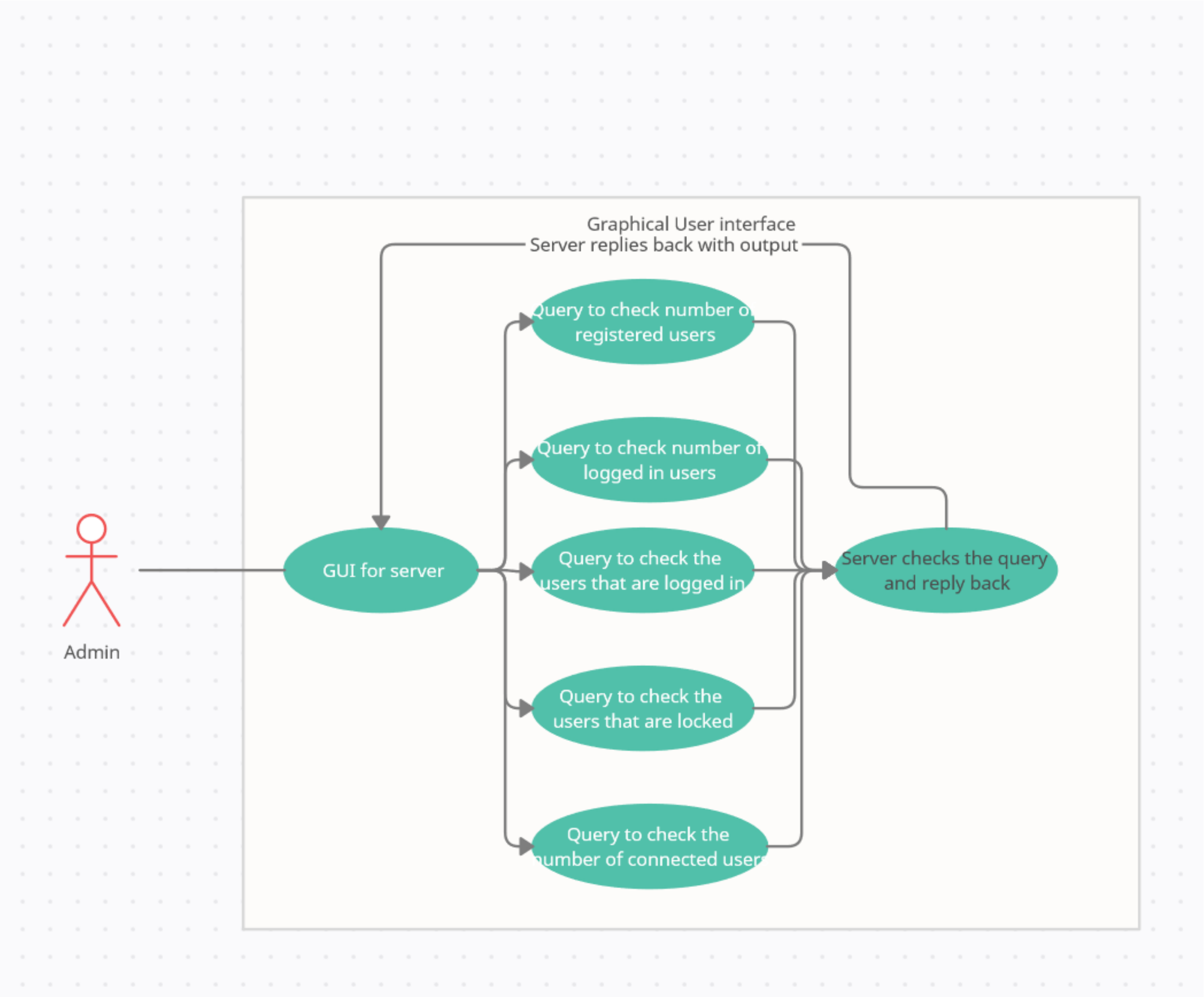
- precondition - server is running
- Postcondition - ready for output
- Story -

  1. User log in the account
  2. User timer ready for 5 hours
  3. User count automatically logged out

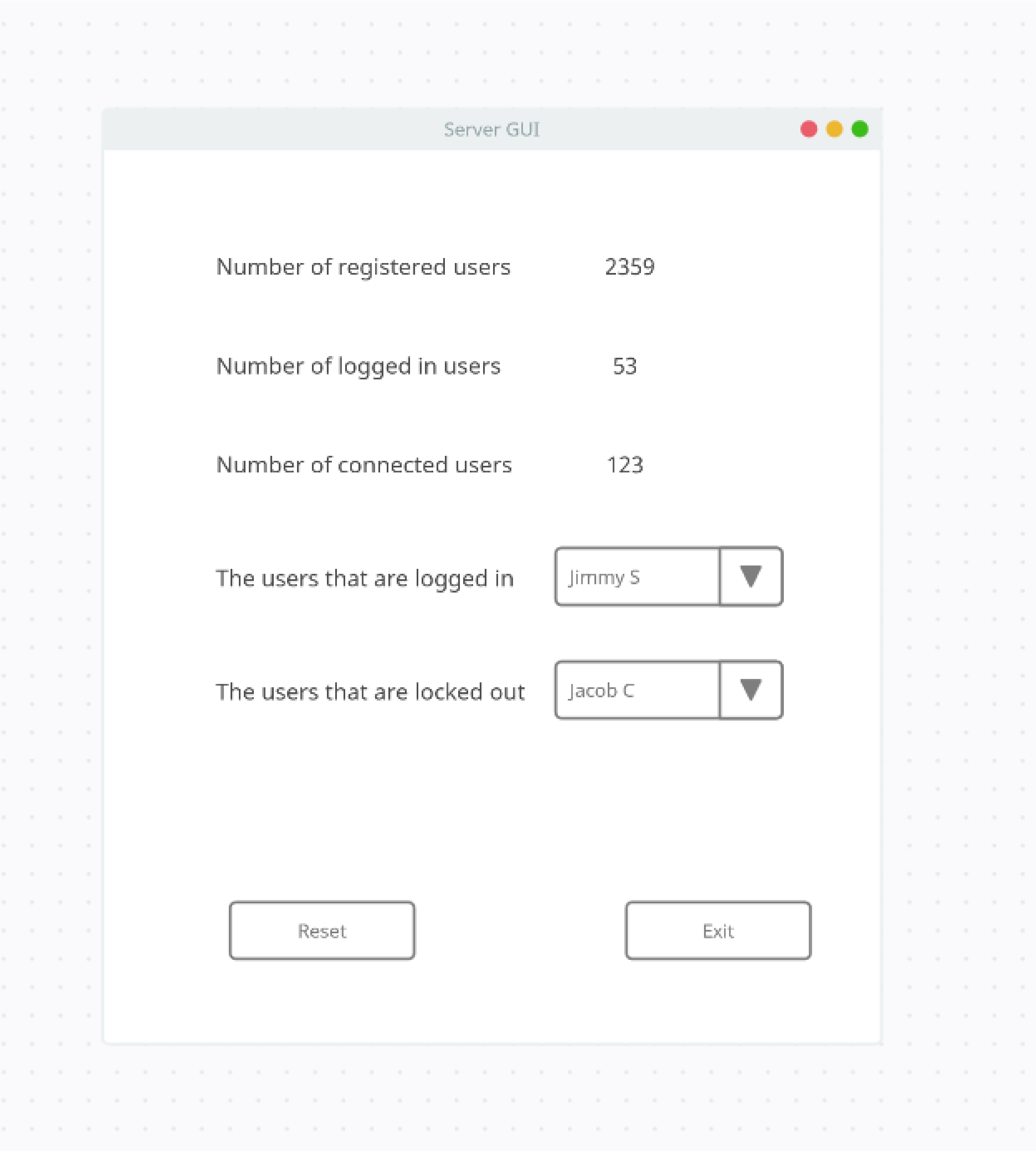
UML diagram for users:



UML diagram for Admin


Graphical User interface
Server replies back with output
Query to check number of registered users
Query to check number of logged in users
Query to check the users that are logged in
Query to check the users that are locked
Query to check the number of connected users
GUI for server
Server checks the query and reply back
Admin

Initial GUI Design:



Number of registered users 2359

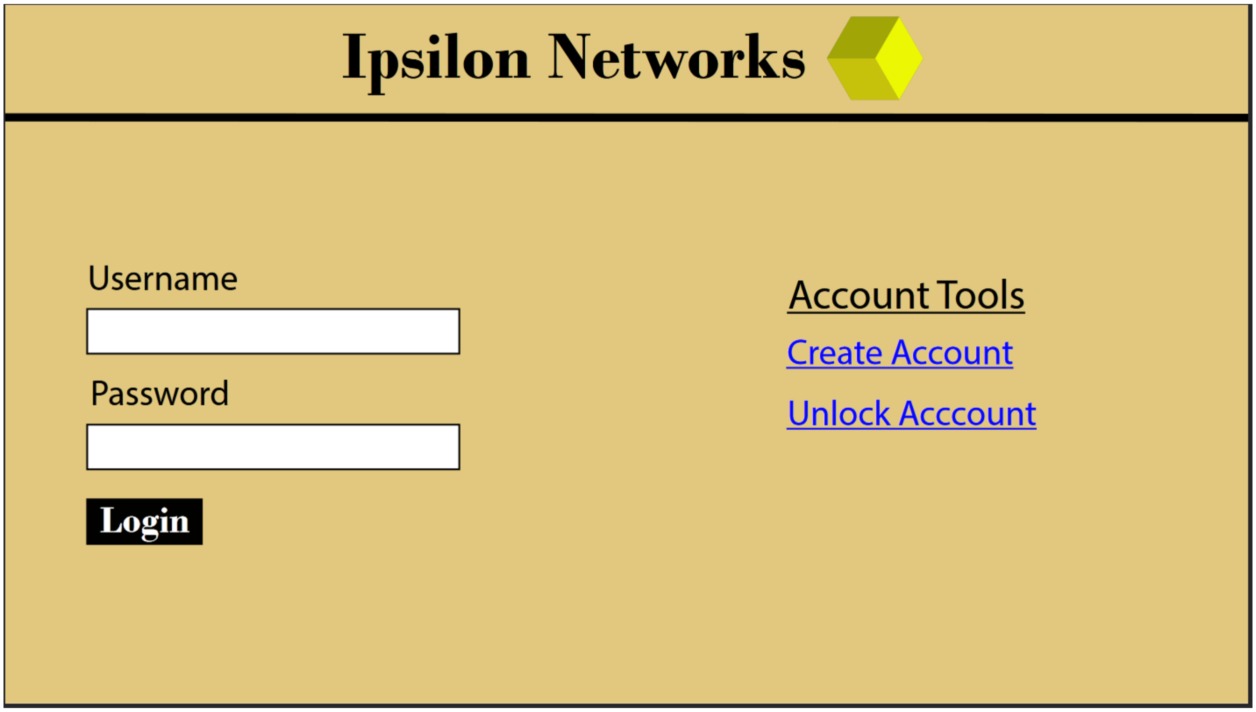
Number of logged in users 53

Number of connected users 123

The users that are logged in: Jimmy S ▼

The users that are locked out: Jacob C ▼

Server graphical user interface



User Interface