**Join Channel**

User: Registered User/Guest

Input: the channel name or the streamer name

Output: “Channel found” and the exact match or “No result found”

Base Case: The exact match is found and the user joins in the channel

Exceptions: the channel is not found or the channel is banned, then “No result found” is returned. The user fails to join the channel he/she is looking for.

**Follow Another User**

User: Registered User

Input: the name of another user

Output: “Successfully follow XXX” or “XXX is not found”

Base Case: The user with the input name is found and the related follow relationship is added to the Follows table.

Exceptions: The user with the input name is not found and the message “XXX is not found” is returned.

**Check for Followers**

User: Registered User

Input: None

Output: list of registered users that are following the user.

Base Case: The list of users that are following you is returned and the possible count of followers

Exceptions: none

**Send a Message**

User: Registered User

Input: content

Output: None, “Please join a channel first” or “You are muted”

Base Case: The system will first check if the user is in a channel and the user is not muted in that channel. If this is the case, the message with the input content is created and added to the message table including the sender’s username and the current time.

Exception: If the user is not in a channel, “Please join a channel first” is returned. If the user is muted in the current channel, “You are muted” is returned.

**Search for Channels of a Specific Type**

User: Registered User/Guest

Input: target type

Output: “Not Found” or a list of channels

Base Case: A list of channels having the input type

Exception: If there is no user with the input type exists, “Not Found” is returned.

**Search for a User**

User: Registered User/Guest

Input: target’s username

Output: “Not Found” or the exact match

Base Case: The exact match is found and returned

Exception: If there is no user with the input type exists, “Not Found” is returned

**Register**

User: Guest

Input: username, password

Output: OK or Denied

Base Case: The system will check if the username is in the database. The system will update the RegisteredUser table with a new RegisteredUser.

Exceptions: If username exists in the database, then Denied is returned.

**Donate**

User: RegisterdUser

Input: RegisteredUser.username, amount

Output: OK or Cancelled

Base Case: The system will add a log record in the donate table.

Exceptions: If Streamer does not exist in the database, donation is aborted and “Cancelled” is returned.

**Mute**

User: Moderator

Input: Moderator.username, RegisteredUser.username, startTime, duration

Output: OK or Cancelled

Base Case: The system will check if RegisteredUser.username exists in the database. If it exists, the user will not be able to send chat messeges to the channel chat from startTime for duration. The system will also update the mute table.

Exceptions:

If RegisteredUser.username does not exist in the database, the action is cancelled.

If RegisteredUser is not watching a Channel that the Moderator is in, the action is cancelled.

**Check Donation Made**

User: Registered User

Input: None

Output: A list of tuples of the form (toUser, amountDonated) and the total amount donated

Base Case: The system will check the donate table if the user made any donations

Exceptions: If the user does not make any donations, an empty list and a 0 total is returned.

**Check Donations Received**

User: Streamer

Input: None

Output: A list of tuples of the form (fromUser, amountReceived) and the total amount received

Base Case: The system will check the donate table if Streamer received any donations

Exceptions: If Streamer does not receive any donations, an empty list and a 0 total is returned.

**Grant Moderator Privileges**

User: Streamer

Input: RegisteredUser.username

Output: OK or Denied

Base Case: The system will check if RegisteredUser.username exists in the database. It will flag RegisteredUser as a Moderator in the Channel of Streamer.

Exceptions: If the username does not exist, then Denied is returned.

**Start/End Streaming**

User: Streamer

Input: None

Output: OK or Denied

Base Case: The system will toggle the channel status to online/offline if it is not banned by the admin.

Exceptions: If the channel is banned by the admin, then Denied is returned.

**Change Channel Info**

User: Streamer

Input: title, description, language, Type.name

Output: OK or Denied

Base Case: The system will update the channel information with the given parameters if the channel status is online or offline.

Exceptions:

If the channel is banned, then Denied is returned.

If Type.name does not exist in the database, then Denied is returned.

**Add Type**

User: Admin

Input: Type.name

Optional Input:

Game.platform

Show.theme

Output: OK or Denied

Base Case: The system will check if the type exists in the database. It will update the type table.

Exceptions: If the type exists in the database, then Denied is returned.

**Add User**

User: admin

Input: User Name, User Gender, User Password

Output: “User already existed” or “User Added”

Basic Case: The system will first check if the User Name already exists in the database. If doesn’t, then the system will generated a unique ID for the user and update the user ID, user name and user gender to the user table. If the user is registered, then the system will update the “registered user table”. Otherwise, the system will update the “ guest table “. Finally the system will return “User Added”

Exception: If the user name already exists, then the system will ask the user to enter another name until the name is not being used by others.

**Delete**

User: admin

Input: User ID

Output:” User deleted” or “User does not exist’

Basic Case: The system will first check if the User Name already exists in the database. If it exists, then the user account associated with that user ID is removed from the user table , and “User deleted” is returned.

Exception: If the user name doesn’t exist, then the operation is aborted and “User does not exist” is returned.

**Create Channel**

User: admin

Input: Streamer.username, Channel title, Channel type.name

Optional input:

If the type is game : gaming platform

If the type is shows: theme

Output: “Channel Created” or “Failed” or “Streamer unqualified”

Basic Case: The system will update the Channel table. Then update the Game or Show tables according to the channel type and return “Channel created”

Exceptions:

Streamer ID is not in the streamer table, or the streamer is currently banned, then the system will return “Streamer unqualified”

If the type doesn’t exist, the user will be asked to correct it. If still not correct, then the operation is aborted and “Failed” is returned.

If the gaming platform or shows theme does not exist, then the user will be asked to correct it. If still not correct, transaction is aborted and “Failed“ is returned.

**Allow registered user to be a streamer**

User: admin

Input: user ID

Output: “Enabled” or “User does not exist” or “ User is non-registered”

Basic Case: The system will first check if the user ID exists in the database. Then the system will update the streamer table and returns “Enabled”

Exception: If the user ID does not exist in the user table, then “User does not exist” is returned.

If the user ID does not exist in the registered user table, then “User is non-registered” is returned.

**Ban channel**

User: admin

Input: Channel ID

Output: “ Channel Banned” or “ Channel does not exist”

Basic Case: The system will first check if the channel ID exists in the database. Then update the Banned channel table and returns “ Channel Banned”.

Exception: If the Channel doesn’t exist in the channel table, then the operation is aborted and “Channel does not exist” is returned.

**Reactive banned channels**

User: admin

Input : Channel ID

Output:”Channel is reactive” or “Channel does not exist” or “Channel is not banned”

Basic Case: The system will first check if the channel ID exists in the database. Then move the channel from banned channel table to Channel table and returns “ Channel is reactive”.

Exception: If the Channel doesn’t exist in the channel table, then the operation is aborted and “Channel does not exist” is returned.

If the Channel ID doesn’t exist in the banned channel table , then the operation is aborted and “Channel is not banned” is returned.

**get top channels**

User: admin

Input: null

Output: channel IDs which has the top most viewers or “Currently no channels”

Basic Case: The system looks through the Channel table, returns the top 5 channel IDs with the most viewers.

Exception: If the channel table is empty, the system will return “Currently no channels”

**get top channel types**

User: admin

Input: null

Output: channel types which has the top most viewers in total or “Currently no channels”

Basic Case: The system looks through all the different types of Channel table, adding up the count of viewers of each element in the tables, and returns the channel types which have the top most total number of viewers.

Exception: If the channel table is empty, the system will return “Currently no channels”