Application starts and presents user with 3 options:

* (1) Create User
* (2) Login
* (3) Exit

If user selects (1) the console, then asks the user to create a username.

* When user hits enter, the application reads that input into a variable, then the console asks the user to enter a password.
* After the user hits enter, the password is stored, and the console asks for a confirmation password.
* The user must hit enter again, and tests if the password and confirmation password match.
  + IF THEY DO NOT MATCH the console asks the user to re-enter the confirmation password until they do match.
  + If they do match, the user can go to the login part of the application.

If the user selects (2) OR if the user is coming from the create user path, the application asks the user for the username.

* After the user has entered their username, the application makes a query to get the password for that username.
* Then the console asks the user for the password.
* After the user has entered a password, the application compares the password from the Database and the password from the user’s input.
* If they do NOT match the application says “Sorry passwords do not match. Renter password.” Until the passwords match.
* If they match the user moves on in the application and is given additional options.

Once the user is logged-in, the application writes additional options to the console:

* (1) View Your Lists.
* (2) Create a List.
* (3) Edit a List.
* (4) Delete a List.
* (5) View all Sharable Lists.
* (6) Logout.

If user selects (1) the application retrieves all Lists from that user, based off that user’s ID.

If user selects (2) the application moves to the Create New List part of the application which will prompt the user to create a new list.

* First the user is prompted to enter a title for their list.
* Then the user is prompted to add a list item. The console also states enter “DONE” to stop adding items to the list.
  + The user enters a string for whatever list item they want. Ex: “Bicep Curl”
  + Then the console tells them to enter details about that list item. Ex: “3 sets 8 reps”
  + After the details, the user can add another list item if they want.
* When the user enters “DONE” the console then asks the user if they want to make their list available for all other users.
  + (1) Yes, I like to share!
  + (2) No, keep it private!
  + If user selects (1) the IsSharable Boolean is set to true.
  + If user selects (2) the IsSharable Boolean is set to false.
* After the user enters (1) or (2) the list is then saved, and the console prints the original options.

If user selects (3) the application asks the user for an ID (OR TITLE?), and then the application will retrieve that list based off that ID. Also tells user that they can input “show” and then the application will retrieve all that user’s lists and will print the titles to console.

* NOTE – the application should test the returned objects to make sure it isn’t empty or didn’t find a list. If it didn’t find a list, then the application should display this to the console.
* If a list is found the console asks, “Which List item do you want to edit?” The console also prints all list items with a respective number.
  + Ex. (1) Bicep Curl
  + (2) Triceps Pushdown
  + Etc.
* Which the user is then prompted to enter a number.
* Then the console will ask, “What is the new list item?”
* The user will then enter a new list item.
* Then the console will ask, “What is the new detail associated with this list item?”
* The user will then enter a new detail about the list item.
* The console will then save this change to the database and then ask the user if they wish to continue editing.
  + (1) Yes
  + (2) No, I am finished.
  + Yes (1), will start the process over again, but using the same list. So, the console will ask, “Which list item do you want to edit?” while printing lists numbers with the list items. This will also show the updated list item.
  + No (2) , will have the user exit this process and then the console will print the original options.

If user selects (4) then the console should ask for an ID (OR TITLE?) to delete that list. First the application will try to retrieve a record from the database.

* If the record comes back empty, the application will say “Invalid list ID (OR TITLE)”
* If it exists, the application will say “Are you sure you want to delete “ + ListTitle + “ list?.”
* Then the applicatin will give the user the following options:
  + (1) Yes, delete already
  + (2) Nah, I changed my mind.
* If (1) the record is then deleted.
* If (2) the console reprints all original options.

If user selects (5) the application selects all lists from the database that have the IsSharable Boolean to true and prints the titles to console. After they are printed the console asks the user if they want to copy one of the lists.

* (1) Yes, I want to copy a list
* (2) No, I do not.
* If user selects (1) the user is prompted for the Title of the list they want to copy, ALSO stating that the title must match exactly.
  + The user enters a title, and if the title matches a list that has the IsSharable Boolean to true, a copy of that list is created but with that user’s ID
  + ASK CHRISTINA ABOUT THIS TOMORROW MAYBE?? Ask if this is a correct way to do many to many, OR should the UserID field be an array of user id’s?
    - If its an array, it will be a varchar object on the other side and will have to do some parsing.
* If the user selects (2) the console prints all original options.

If user selects (6) the user is “Logged out” i.e. the application closes. 😊