ClubScout

Analysis Report

BUS 465 – Business Systems Development
April 27, 2021

Group 4:
Taylor Vaz, Thomas Nguyen, Sanwal Cheema

Table of Contents

Executive Summary

The popularity and usefulness of course selection assistance websites like CourseDiggers and RateMyProf demonstrate a demand towards systems which facilitate decision-making for its users. Such a need has been identified for SFU's extra-curricular programs. Currently, no tool exists which could assist SFU students in choosing one of the many clubs the university offers. The Simon Fraser Student Society has a basic page from which students can search for a club, and simply sign-up. There is no further functionality which may assist in students to better understand the various clubs before joining.

This report outlines the limitations of the existing SFU club facilitating solutions and the need for improvements of these systems. It defines a new system approach and features use cases (UCs), data flow diagrams (DFDs) and entity relationship diagrams (ERDs) to visually represent the reformed system.

The proposed system is designed in consideration of the aforementioned constraints experienced by current processes and aims to improve the quality of user experience, efficiency, reliability, and transparency of its processes.

The system proposes to offer the following functions:

- Search for any existing SFU club with category filtering optionality
- Rating and reviewing joined clubs to assist other students' decision-making
- Creating an account with the site
- Receiving club recommendations based on preferences
- Following/Unfollowing clubs
- Receiving monthly newsletters about new clubs and related club news
- Signing-up as a club admin and managing your club
- Viewing upcoming club events

As with any business, the prospective risks consist of resistance by current systems and its users, less than projected demand for proposed system, unidentified sources of competition, and technical limitations which result in reduction of functionality. The project team intends to identify, evaluate, and plan for mitigating major risks by not creating too ambitious of a project and implementing project-wide reviews upon each significant milestone.

System Request

Project Name	**ClubScout** – All-In-One SFU Club Website
Project Sponsor	N/A
Business Need	This project has been designed to meet the following needs: • Easy and interactive club choosing tool • Get matched with clubs based on personal preferences • Facilitate SFU club choosing decision-making • Get reviews and feedback on existing clubs • Get periodical news updates on new clubs and related club news.
Business Requirements	Our system will provide the following functionalities: • Ability for users to create accounts • User verification on the user's email upon account creation • Ability to log-in to account • Ability to become a club admin if eligible • Ability to search for clubs based on filtered categories • Ability to add new club listings to the site • Ability of club admins to edit or delete clubs from the site • Ability to be able to submit club reviews for joined clubs • Ability of Club admins to add club events to a club webpage • Ability to receive email newsletter with SFU club news
Business Value	With this system, students are better able to find clubs that match their interests. Students are also able to view club reviews to figure out whether or not others enjoyed this club as well. This will save students time and avoid the stress of finding a club for them. It will also alleviate any barriers students have to joining clubs. This site also makes it easier for existing SFU clubs or club admins to reach the SFU student body. Moreover, the site offers an easier way for students to create a new club.
Special Issues or Constraints	Some site functions cannot be implemented without approval and access to the SFU server and database.

Use Cases

The prototype website includes a total of eight use cases: creating a user account, log-in, adding a new club, deleting or editing a club, searching for a club, submitting a club review, sending an email newsletter to account holders, and adding a new club event.

In order to maximize the full capabilities of the site, a user should first create an account after visiting the home page. After, a user account will need to be verified to ensure that the user is in fact an SFU student. Once this process is complete, a user can log-in to their account.

The third use case is triggered by a user going to their account to add a new club to the site. After filling in necessary fields, they will submit the form. After a website admin has verified the club listing and updates its status as verified in the database, the club will be available for view on the user account page. From there if a club admin wishes to delete or edit a club, the fourth use case is triggered. Club admins can edit or delete clubs directly from their account page. Club admins can also add new club events to a club webpage for which they are an admin for.

Users can also search for clubs on the home page by using the search bar to type in keywords, or sorting clubs by category. There are a total of 12 club categories from which users can filter their searches by. Once a user finds the club they are looking for, they can click the link to be taken to the webpage for that particular club.

Users may also submit club reviews for clubs which they have joined. Once a user has searched for the club and is on their webpage, they trigger this use case by clicking “Rate This Club”. After filling out necessary inputs and submitting the review form, their review will be viewable on the club webpage, along with other previous reviews.

The last use case involves a website admin creating a monthly newsletter to be sent to all account holders. The website admin will compile relevant club information or updates for that month, including events and new club additions, from the database to send to all account holders.

Each use case is described in detail below.

Use Case 1: Create a new profile

Use Case Name:	Create a new profile	**ID:**	UC-1	**Priority:**	High

Actor:	Client
Description:	A website user creates a new account using their SFU email. The user can create a student account or a club admin account. The user will verify their account with an activation code.
Trigger:	A user clicks "create account" on home page
Type:	**External**
Preconditions:	1. Site is functional 2. User account database is online

Major Steps:	**Information for Steps:**
1. The user identifies if they would like to create a student account or a club admin account	→Account type
2. The system displays the account sign-up form for the selected account type. User then inputs their information into the required fields such as SFU email, username, password, name and SFU concentration.	←Student/Club admin sign-up form →SFU email →username →Password →Name →SFU Concentration
3. System prompts user to input club category preferences which user can either input at this step, or later on after account creation. a) User inputs a club category preference b) User does not input a club category preference.	←User club category preference
4. User submits form	←Unverified user account
5. System sends verification email to the user's SFU email account including a 4-digit verification code.	←Account verification email
6. The user clicks the link provided in the verification email and enters the 4-digit verification code. System updates the account as verified.	→User verification code ←Verified user account
7. System displays that the user has successfully verified their account	←Account verification message

Postconditions:	1. User is now able to log-in to the website 2. User is now able to save clubs and club preferences to their account
Exceptions:	E1: user decides they no longer wish to sign-up for an account (occurs before step 4) 3. User exits the webpage or clicks "cancel" 4. Use case terminates E2: User is already found in the user account database (occurs at step 4) 1. The website displays error message "existing user account found" 1. Return to step 1

Summary

Inputs	**Source**	**Outputs**	**Destination**
Account type	User	Sign-up form	User
User details (name, email, password)	User	Unverified user account	User account database

User club category preference	User	Account verification email	User
User verification code	User	Verified user account	User account database
		Account verification message	User

Use Case 2: Log-in to the website

Use Case Name:	Log-In to the website	**ID:**	UC-2	**Priority:**	High

Actor:	User
Description:	User logs into the website using their verified user account that has been created. From there, the user can view their user information.
Trigger: **Type:**	A user clicks "log-in" on the home page **External**
Preconditions:	1. Site is functional 2. User account database is online 3. User account has been created.

Major Steps:	**Information for Steps:**
1. User inputs their SFU Email and password that they used to create their account.	→SFU email →Password
2. System verifies that the account exists in the user account database	→User log-in information ←User account database verification
3. System logs in the user and displays a successful login message	←Successful log-in message

Postconditions:	1. User can view their account details 2. User can submit club reviews 3. User can save clubs preferences to their profile
Exceptions:	E1: User is not found in the website database (occurs step 2) 1. The website displays message log-in failure error message 2. Return to step-1

Summary

Inputs	**Source**	**Outputs**	**Destination**
SFU email, password	User	User account verification	User account database
		Successful log-in message	User

Use Case 3: Adding a club to the website

Use Case Name:	Adding a club to the website	**ID:**	UC-3	**Priority:**	High

Actor:	Club Admin
Description:	A club admin can add a new club to the website by providing information such as a club name, description, and category to the database. A website admin will confirm the submission before it is published on the website.
Trigger:	A club admin logs into the website and clicks "add a club" from his/her account page
Type:	**External**
Preconditions:	1. Site is functional. 2. Club admin has been logged in and authenticated 3. Club database is online 4. SFU database is online

Major Steps:	**Information for Steps:**
1. The club admin enters required information fields into the website form.	→Club name →Club description →Club category →Club skills learned
2. The system confirms that the club does not exist in the database.	←New club verification message
3. The system sends the club listing request to be confirmed by a website admin	←Club listing request
4. The website admin confirms the club listing request	→Website admin club listing request confirmation
5. The system adds the club listing to the website	←Club listing
6. Club admin is notified that their club listing request has been verified and is now viewable on the website.	←Confirmation email sent

Postconditions:	1. Club listing is viewable on website 2. Club listing will now be included in quiz 3. Club listing can be edited or removed by a website admin or club admin
Exceptions	E1: Club is already found in the website database (occurs at step 2) 1. The website displays message "Club is already in the system" 2. System prompts club admin to input new info fields or cancel the request 3. Return to step 1

Summary

Inputs	**Source**	**Outputs**	**Destination**
Club name, Club Description, Club Category, Club skills learned	Club Admin	New club verification message	Club Admin
Club listing request confirmation	Website Admin	Club listing request	Club listing request database
		Club listing	Club database
		Confirmation email	Club Admin

Use Case 4: Edit/Delete a club

<table>
<tr><td><b>Use Case Name:</b></td><td>Edit/Delete a club</td><td><b>ID:</b></td><td>UC-4</td><td><b>Priority:</b></td><td>High</td></tr>
<tr><td><b>Actor:</b></td><td colspan="5">Club Admin</td></tr>
<tr><td><b>Description:</b></td><td colspan="5">A club admin logs into their account to view their club listings to edit or delete a club listing</td></tr>
<tr><td><b>Trigger</b>:</td><td colspan="5">Club admin clicks on a club listing and selects "edit listing"</td></tr>
<tr><td><b>Type:</b></td><td colspan="5"><b>External</b></td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="5">1. Site is functional
2. Club listing database is online
3. Club admin is logged in and authenticated</td></tr>
<tr><td colspan="3"><b>Major Steps:</b></td><td colspan="3"><b>Information for Steps:</b></td></tr>
<tr><td colspan="3">1. <b>IF</b> club admin wants to delete the club listing entirely, he/she will select "delete listing" and submit the form</td><td colspan="3">→Club listing deletion request</td></tr>
<tr><td colspan="3"><b>Otherwise,</b> club admin inputs the changes into the required fields of a club listing and submits the form</td><td colspan="3">→Edited changes request</td></tr>
<tr><td colspan="3">2. System displays confirmation message that changes have been successfully made and updates the club listing database</td><td colspan="3">←Edit confirmation message
←Updated club listing</td></tr>
<tr><td><b>Postconditions:</b></td><td colspan="5">1. Club listings database will be updated
2. Edit club listing will now be viewable on the website
3. Club admin is able to view updated listing on their account</td></tr>
<tr><td colspan="6"><b>Summary</b></td></tr>
<tr><td><b>Inputs</b></td><td><b>Source</b></td><td colspan="2"><b>Outputs</b></td><td colspan="2"><b>Destination</b></td></tr>
<tr><td>Order #</td><td>Club Admin</td><td colspan="2">Edit confirmation message</td><td colspan="2">Club Admin</td></tr>
<tr><td>Order verification</td><td>Club Admin</td><td colspan="2">Updated club listing</td><td colspan="2">Club listing database</td></tr>
</table>

Use Case 5: Search for a club

Use Case Name:	Search for a club on the home page	**ID:**	UC-5	**Priority:**	High

Actor:	User
Description:	A user can manually search for a club from the home page of the website. A user can click on category links.
Trigger:	User clicks on clicks a club category link on the home page
Type:	**External**
Preconditions:	1. Site is functional 2. Club database is online 3. Club review database is online 4. Club webpage is updated 5. Club category webpage is updated

Major Steps:	**Information for Steps:**
1. User clicks on a club category link on the homepage	→User club category selection
2. The website shows a webpage with a list of clubs under that category.	←Club category webpage
3. User clicks a club link from the shown list	→User club selection
4. The website shows the webpage for that club. The webpage includes the club category, club skills learned, and reviews.	←Club webpage

Postconditions:	1. User has been correctly linked to selected club 2. User is able to view the club category webpage and the club webpages

Summary

Inputs	**Source**	**Outputs**	**Destination**
User club category selection	User	Club category webpage	User
User club selection	User	Club webpage	User

Use Case 6: Write a club review

Use Case Name:	Write a club review	ID:	UC-6	Priority:	High
Actor:	User				
Description:	A user writes and submits a review of a SFU club on the dedicated club page.				
Trigger:	A user clicks on the 'write a review' button under the existing reviews of a club				
Type:	**External**				
Preconditions:	1. Site is functional 2. Review database is online 3. Review system is online and functional				

Major Steps:	Information for Steps:
1. User clicks on the 'write a review' button on the review section of a club page.	→Start review prompt
2. The website displays a review text box and a few questions to answer. a) Questions are time commitment (1-3), enjoyability (1-3), skills gained, and club category b) A Text box is also presented for a written review	←Review box
3. The user inputs his/her review and answers the questions	→Review details and preferences
4. The user clicks 'submit review'	→Submit review
5. The website displays confirmation that the review has been received and will be posted shortly	←Submission confirmation
6. The system submits information to Review Database	←New review

Postconditions:	1. Review database is updated

Summary			
Inputs	**Source**	**Outputs**	**Destination**
User club category selection	User	Club category webpage	User
Review details and preferences	User	Submission confirmation	User
Submit review	User	New review	Review Database

Use Case 7: Sending email newsletter to users

Use Case Name:	Send email newsletter to members	**ID:**	UC-7	**Priority:**	High
Actor:	User				
Description:	A user who is an account holder will receive monthly emails to the SFU email they have used to sign up for their user account. The club admin will create a new newsletter to be sent out on the 1st of every month				
Trigger:	1st of each month				
Type:	**Temporal**				
Preconditions:	1. Site is functional 2. User account database is online 3. User account is created and verified				

Major Steps:	**Information for Steps:**
1. Website admin access the list of club events from the event database for a selected month	→Month selection ←Monthly club events report
2. Website admin inputs these events into the email newsletter. They will also include a small description about each event with a link to the club page on the website	→Club event name →Club event details →Link to club webpage
3. Website admin confirms the email and submits it. The system will access the user account database to send the email to all user account holders' SFU email	→User account SFU email ←Monthly email newsletter
Postconditions:	1. User account holder will receive the email

Summary			
Inputs	**Source**	**Outputs**	**Destination**
Month selection	Website Admin	Monthly club events report	Website Admin
Club event name, details, link to club webpage.	Website Admin	Monthly email newsletter	User
User SFU Email	User account database		

Use Case 8: Adding a club event

Use Case Name:	Adding a club event to the club webpage	**ID:**	UC-8	**Priority:**	High

Actor:	Club Admin
Description:	A club admin can add new events to the club webpage so that they can be viewed on the webpage and sent out in the monthly email newsletter
Trigger:	Club admin clicks "add an event" on the club webpage
Type:	**External**
Preconditions:	1. Site is functional 2. Club admin has been logged in and verified 3. Club event database is online 4. Club database is online

Major Steps:	**Information for Steps:**
1. Club admin fills in details like event name, description and date and submits form.	→Club event name →Club event description →Club event date
2. System confirms that the club event does not exist in club events database	←New club event added message
3. System adds club to the club webpage and stores it in the club events database.	←Club event

Postconditions:	1. Event is viewable on club webpage 2. Event can be accessed for the monthly email newsletter
Exceptions:	E1: Club event is already found in the website database (occurs at step 3) 1. The website displays message "Club event already exists" 2. The system prompts the club admin to input new information fields or cancel the request 3. Return to step 1

Summary

Inputs	**Source**	**Outputs**	**Destination**
Club event name, description, and date.	Club Admin	New club event added message	Club Admin
		Club event	Club Events Database

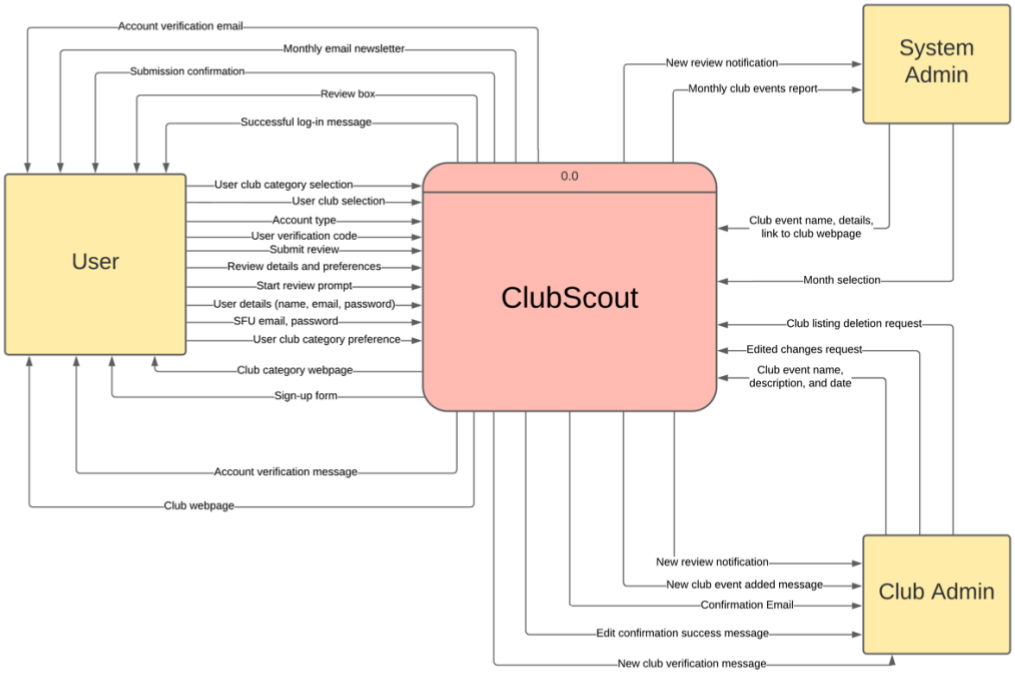
Data Flow Diagrams

To illustrate our system, we created a context, and level zero Data Flow Diagram (DFD).

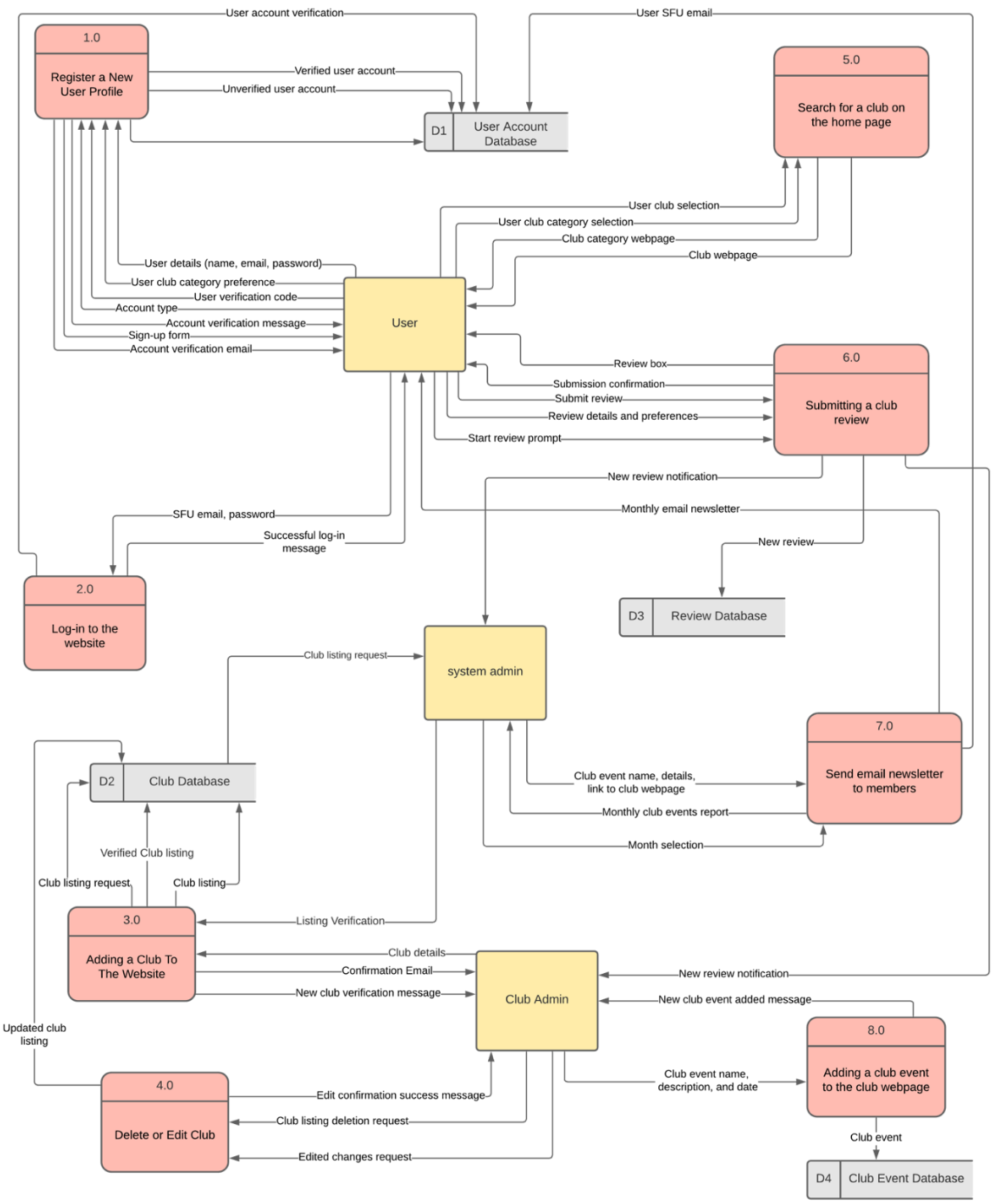
The context level DFD shows our three entities which are the user, club admin, and website admin. These are the same actors performing the use cases listed above.

The level zero DFD shows a total of eight processes including: account registration, logging-in, adding a club to the website, editing/deleting a club, searching for a club on our homepage, submitting a club review, sending email newsletters, and adding club events. This level of DFD also displays four main data stores. User Account database holds user information for account management and login purposes. Club database holds all the data surrounding the various clubs we present on our site. The Review database stores all the pending and published user reviews that are posted by students. Finally, the Club Events database stores all the information about club events that we showcase on the site.

Context Diagram


Account verification email
Monthly email newsletter
Submission confirmation
Review box
Successful log-in message
New review notification
Monthly club events report
System Admin
User
User club category selection
User club selection
Account type
User verification code
Submit review
Review details and preferences
Start review prompt
User details (name, email, password)
SFU email, password
User club category preference
Club category webpage
Sign-up form
0.0
ClubScout
Club event name, details, link to club webpage
Month selection
Club listing deletion request
Edited changes request
Club event name, description, and date
Account verification message
Club webpage
New review notification
New club event added message
Confirmation Email
Edit confirmation success message
New club verification message
Club Admin

Level 0 DFD


User account verification
User SFU email
1.0
Register a New User Profile
Verified user account
Unverified user account
D1
User Account Database
5.0
Search for a club on the home page
User club selection
User club category selection
Club category webpage
Club webpage
User details (name, email, password)
User club category preference
User verification code
Account type
Account verification message
Sign-up form
Account verification email
User
Review box
6.0
Submitting a club review
Submission confirmation
Submit review
Review details and preferences
Start review prompt
New review notification
Monthly email newsletter
SFU email, password
Successful log-in message
New review
2.0
Log-in to the website
D3
Review Database
Club listing request
system admin
7.0
Send email newsletter to members
D2
Club Database
Club event name, details, link to club webpage
Monthly club events report
Month selection
Verified Club listing
Club listing request
Club listing
3.0
Adding a Club To The Website
Listing Verification
Club details
Confirmation Email
New club verification message
Club Admin
New review notification
New club event added message
Updated club listing
8.0
Adding a club event to the club webpage
4.0
Delete or Edit Club
Edit confirmation success message
Club listing deletion request
Edited changes request
Club event name, description, and date
Club event
D4
Club Event Database

Entity Relationship Diagram

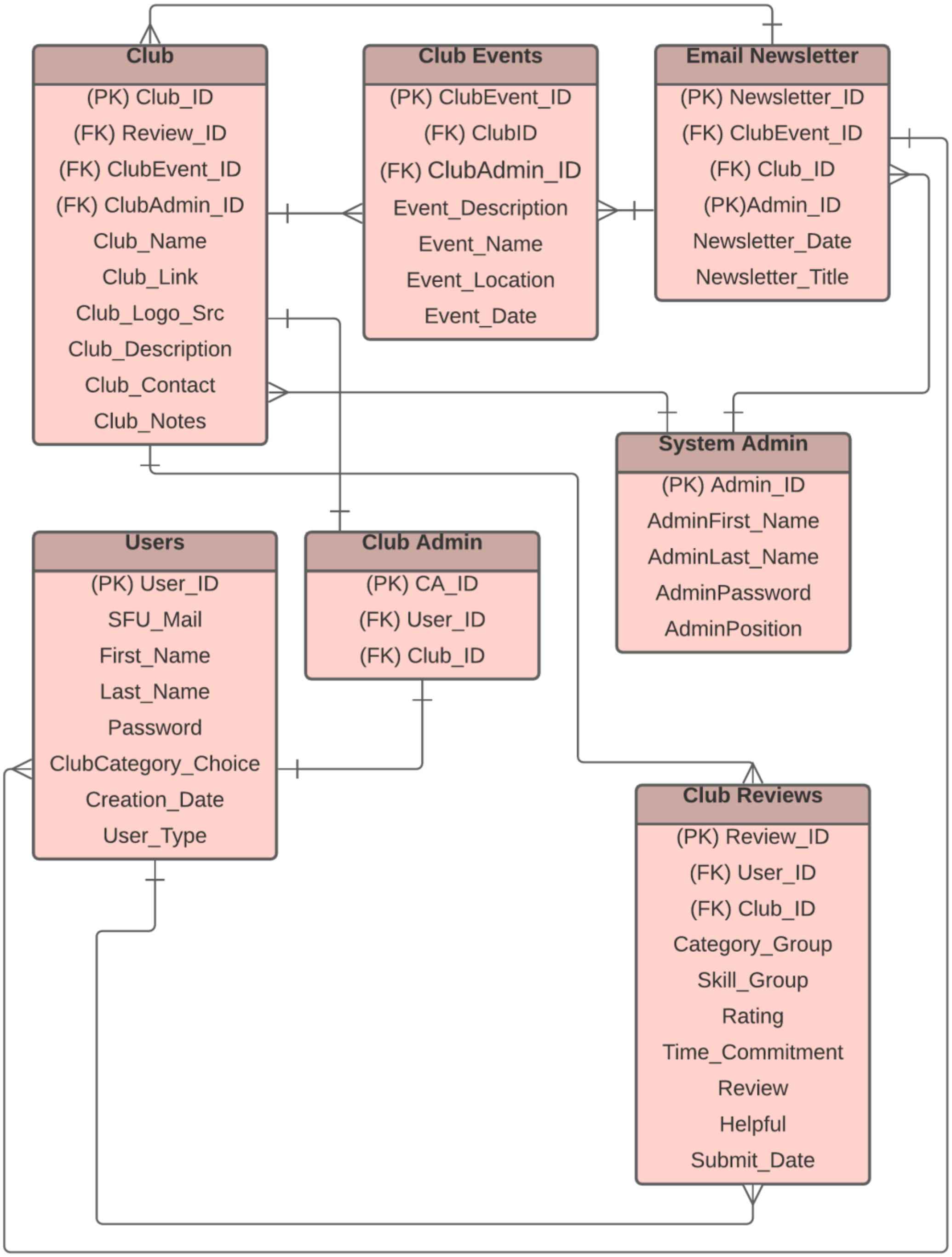
The entity relationship diagram includes the following relationships:

- A user can only be ONE admin
- A user can write many reviews
- One admin can only be the admin of ONE club
- One club can have many club events
- One club can have many club reviews
- One system admin can verify many clubs
- One system can create many email newsletters
- One email newsletter can include information about many clubs
- One email newsletter can include many club events
- One email newsletter will be sent to many users

Some assumptions that were made include the following:

- User information is not recorded into the database until that user creates an account
- All clubs only need to have one person have admin status on the site for a single club
- All clubs will have at least one club event

Our entire ERD is viewable below.

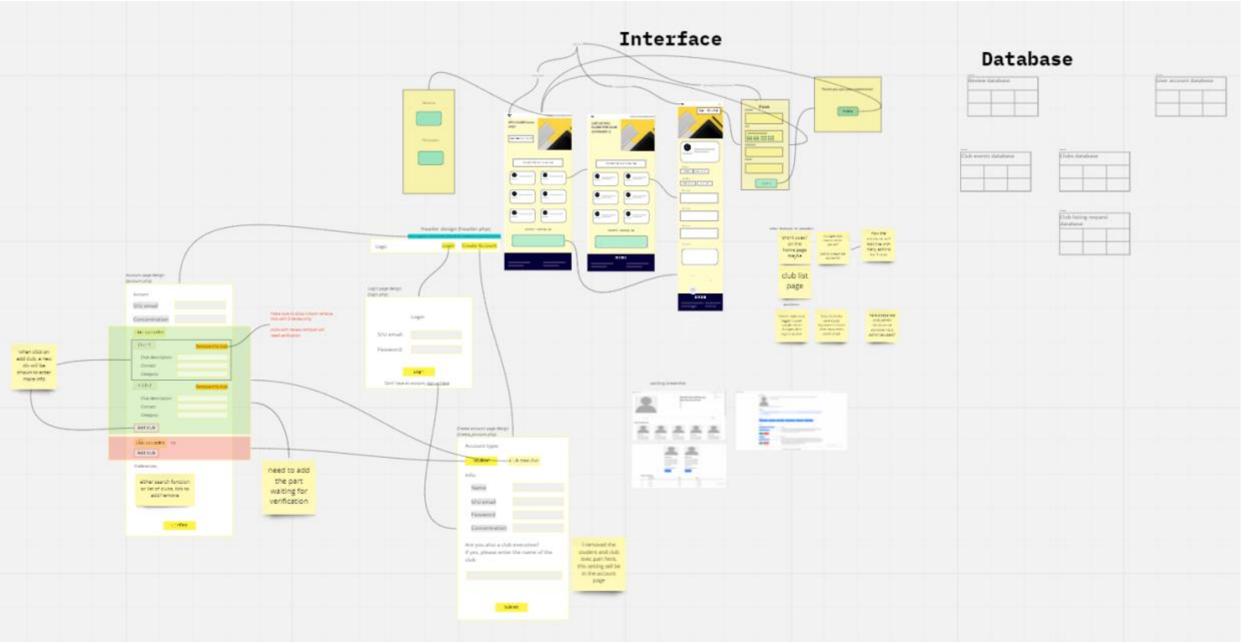

Club
(PK) Club_ID
(FK) Review_ID
(FK) ClubEvent_ID
(FK) ClubAdmin_ID
Club_Name
Club_Link
Club_Logo_Src
Club_Description
Club_Contact
Club_Notes
Club Events
(PK) ClubEvent_ID
(FK) ClubID
(FK) ClubAdmin_ID
Event_Description
Event_Name
Event_Location
Event_Date
Email Newsletter
(PK) Newsletter_ID
(FK) ClubEvent_ID
(FK) Club_ID
(PK)Admin_ID
Newsletter_Date
Newsletter_Title
System Admin
(PK) Admin_ID
AdminFirst_Name
AdminLast_Name
AdminPassword
AdminPosition
Users
(PK) User_ID
SFU_Mail
First_Name
Last_Name
Password
ClubCategory_Choice
Creation_Date
User_Type
Club Admin
(PK) CA_ID
(FK) User_ID
(FK) Club_ID
Club Reviews
(PK) Review_ID
(FK) User_ID
(FK) Club_ID
Category_Group
Skill_Group
Rating
Time_Commitment
Review
Helpful
Submit_Date

Implementation

To create the website, we used HTML, CSS, Bootstrap, JS, Python, and class materials to construct the final product (of course with the help of Stack Overflow too).

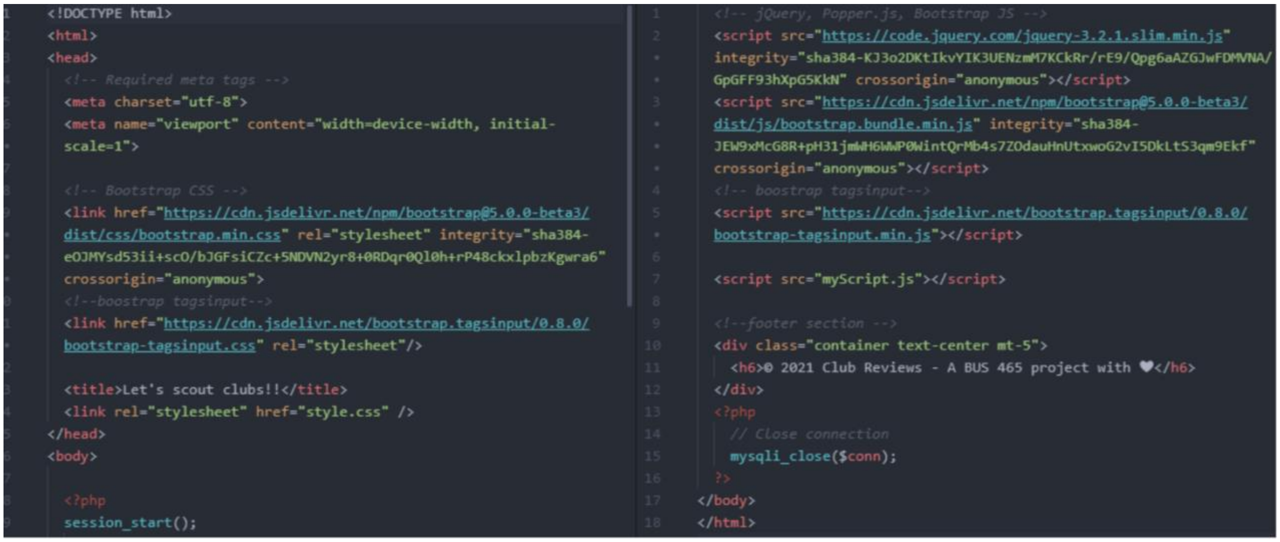
Some concepts & features that we learned and applied during the project:

Project Planning: Miro App


Interface
Database
when click on add club, a new div will be shown to enter more info
need to add the part waiting for verification
club list page

We used the Miro app as a collaboration platform to design our interfaces and plan out our project.

Reusable Code

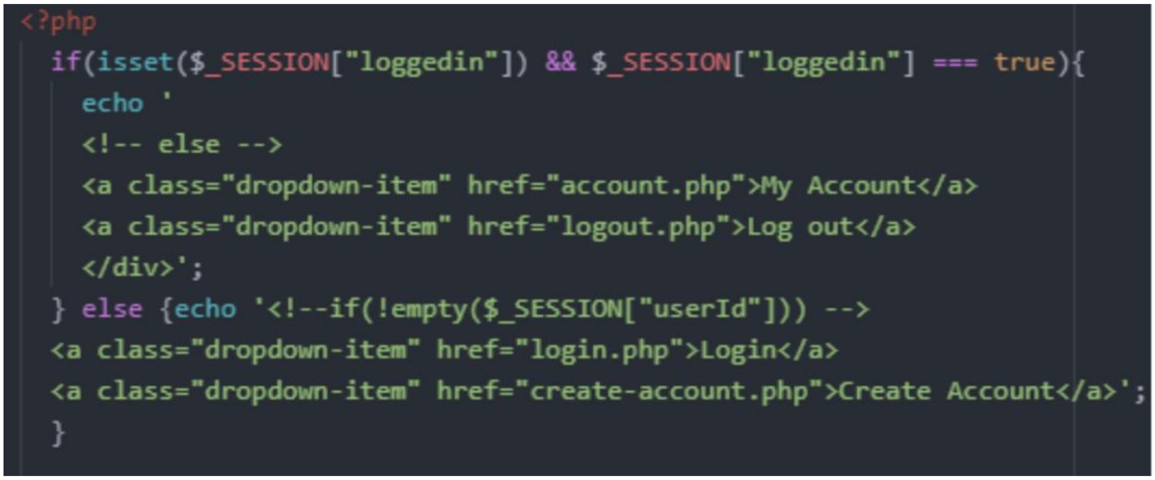


```html
<!DOCTYPE html>
<html>
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
  scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/
  dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
  eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
  crossorigin="anonymous">
  <!--boostrap tagsinput-->
  <link href="https://cdn.jsdelivr.net/bootstrap.tagsinput/0.8.0/
  bootstrap-tagsinput.css" rel="stylesheet"/>

  <title>Let's scout clubs!!</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>

  <?php
  session_start();
```

```html
  <!-- jQuery, Popper.js, Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
  integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/
  GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/
  dist/js/bootstrap.bundle.min.js" integrity="sha384-
  JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
  crossorigin="anonymous"></script>
  <!-- boostrap tagsinput-->
  <script src="https://cdn.jsdelivr.net/bootstrap.tagsinput/0.8.0/
  bootstrap-tagsinput.min.js"></script>

  <script src="myScript.js"></script>

  <!--footer section -->
  <div class="container text-center mt-5">
    <h6>© 2021 Club Reviews - A BUS 465 project with ♥</h6>
  </div>
  <?php
    // Close connection
    mysqli_close($conn);
  ?>
</body>
</html>
```

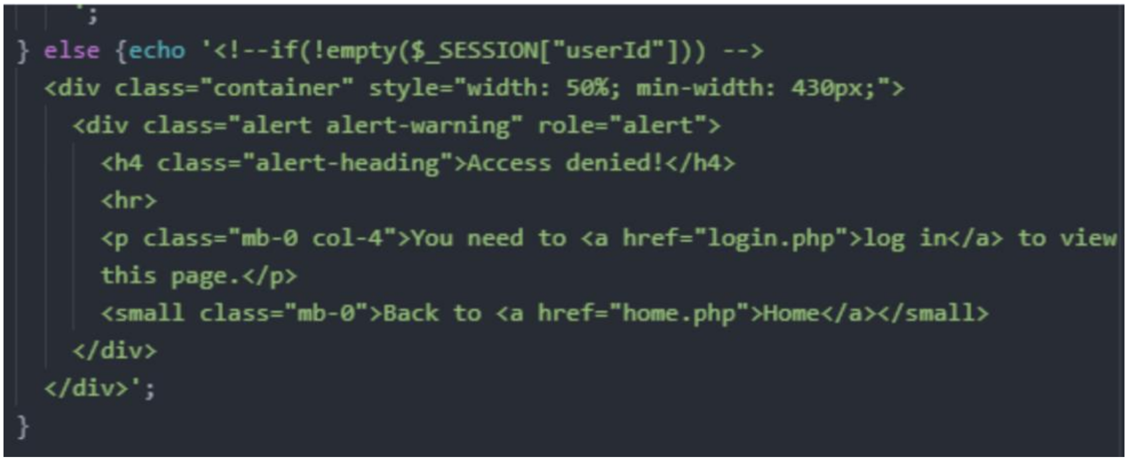
Since we have different settings for logged in users and anonymous users, creating *header.php* and *footer.php* as reusable pieces across pages

Sessions



```
<?php
  if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
    echo '
    <!-- else -->
    <a class="dropdown-item" href="account.php">My Account</a>
    <a class="dropdown-item" href="logout.php">Log out</a>
    </div>';
  } else {echo '<!--if(!empty($_SESSION["userId"])) -->
  <a class="dropdown-item" href="login.php">Login</a>
  <a class="dropdown-item" href="create-account.php">Create Account</a>';
  }
```

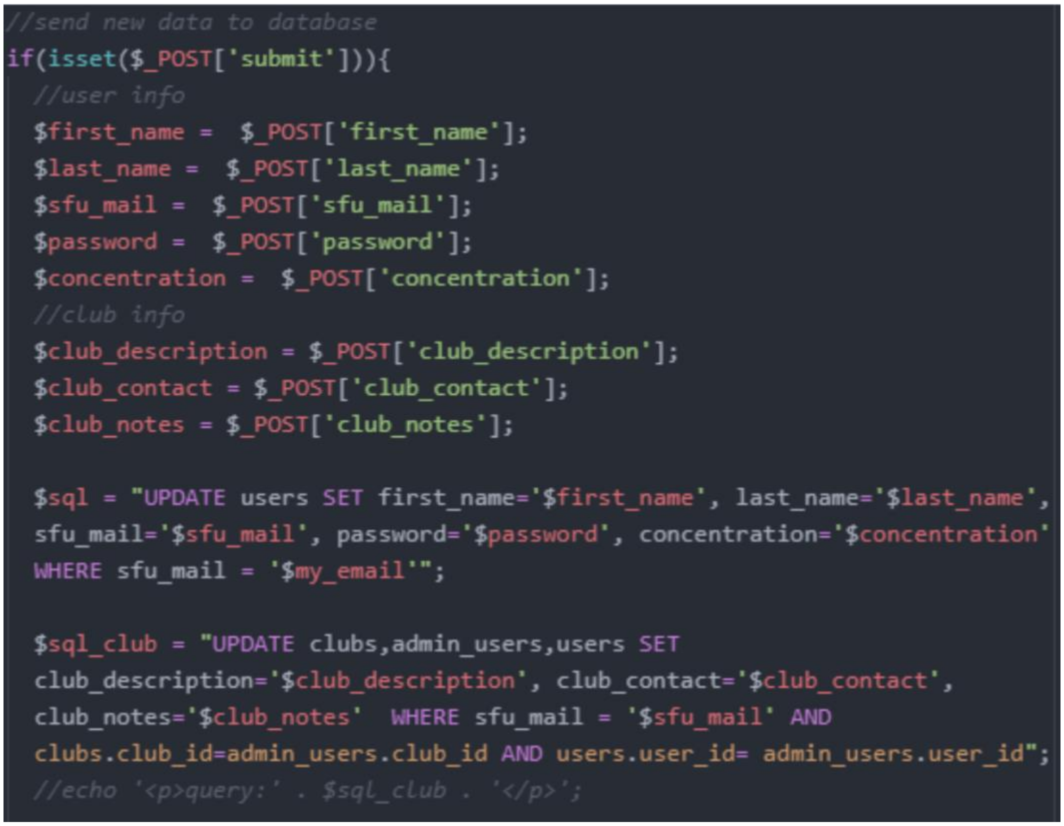
To keep user data after login, we use SESSION to retain data such as user info and the most recent club visited even when a user loads a new page.



```
      ';
} else {echo '<!--if(!empty($_SESSION["userId"])) -->
  <div class="container" style="width: 50%; min-width: 430px;">
    <div class="alert alert-warning" role="alert">
      <h4 class="alert-heading">Access denied!</h4>
      <hr>
      <p class="mb-0 col-4">You need to <a href="login.php">log in</a> to view
      this page.</p>
      <small class="mb-0">Back to <a href="home.php">Home</a></small>
    </div>
  </div>';
}
```

Some features only available to logged in user, thus we need to block anonymous access.

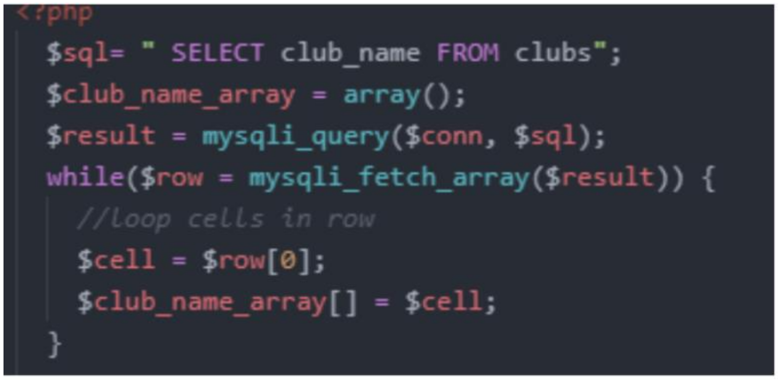
Update User Profile on the Same Page



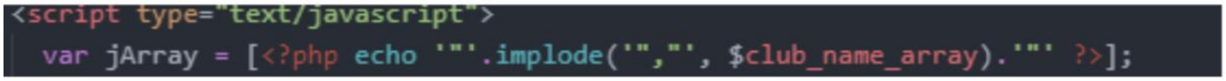
```
//send new data to database
if(isset($_POST['submit'])){
  //user info
  $first_name =  $_POST['first_name'];
  $last_name =  $_POST['last_name'];
  $sfu_mail =  $_POST['sfu_mail'];
  $password =  $_POST['password'];
  $concentration =  $_POST['concentration'];
  //club info
  $club_description = $_POST['club_description'];
  $club_contact = $_POST['club_contact'];
  $club_notes = $_POST['club_notes'];

  $sql = "UPDATE users SET first_name='$first_name', last_name='$last_name',
  sfu_mail='$sfu_mail', password='$password', concentration='$concentration'
  WHERE sfu_mail = '$my_email'";

  $sql_club = "UPDATE clubs,admin_users,users SET
  club_description='$club_description', club_contact='$club_contact',
  club_notes='$club_notes'  WHERE sfu_mail = '$sfu_mail' AND
  clubs.club_id=admin_users.club_id AND users.user_id= admin_users.user_id";
  //echo '<p>query:' . $sql_club . '</p>';
```

By sending form data to the same page, we were able to update the database with new user information

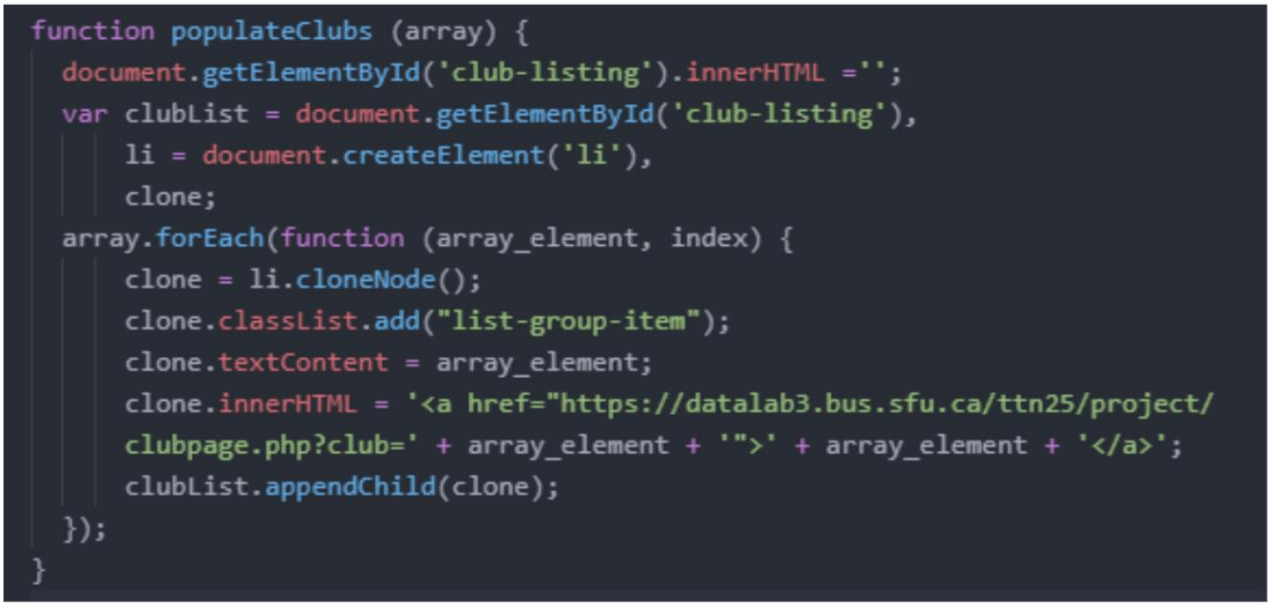
Populate Club Lists from Array Using JS:



```
<?php
  $sql= " SELECT club_name FROM clubs";
  $club_name_array = array();
  $result = mysqli_query($conn, $sql);
  while($row = mysqli_fetch_array($result)) {
    //loop cells in row
    $cell = $row[0];
    $club_name_array[] = $cell;
  }
```



```
<script type="text/javascript">
  var jArray = [<?php echo '"'.implode('","', $club_name_array).'"' ?>];
```

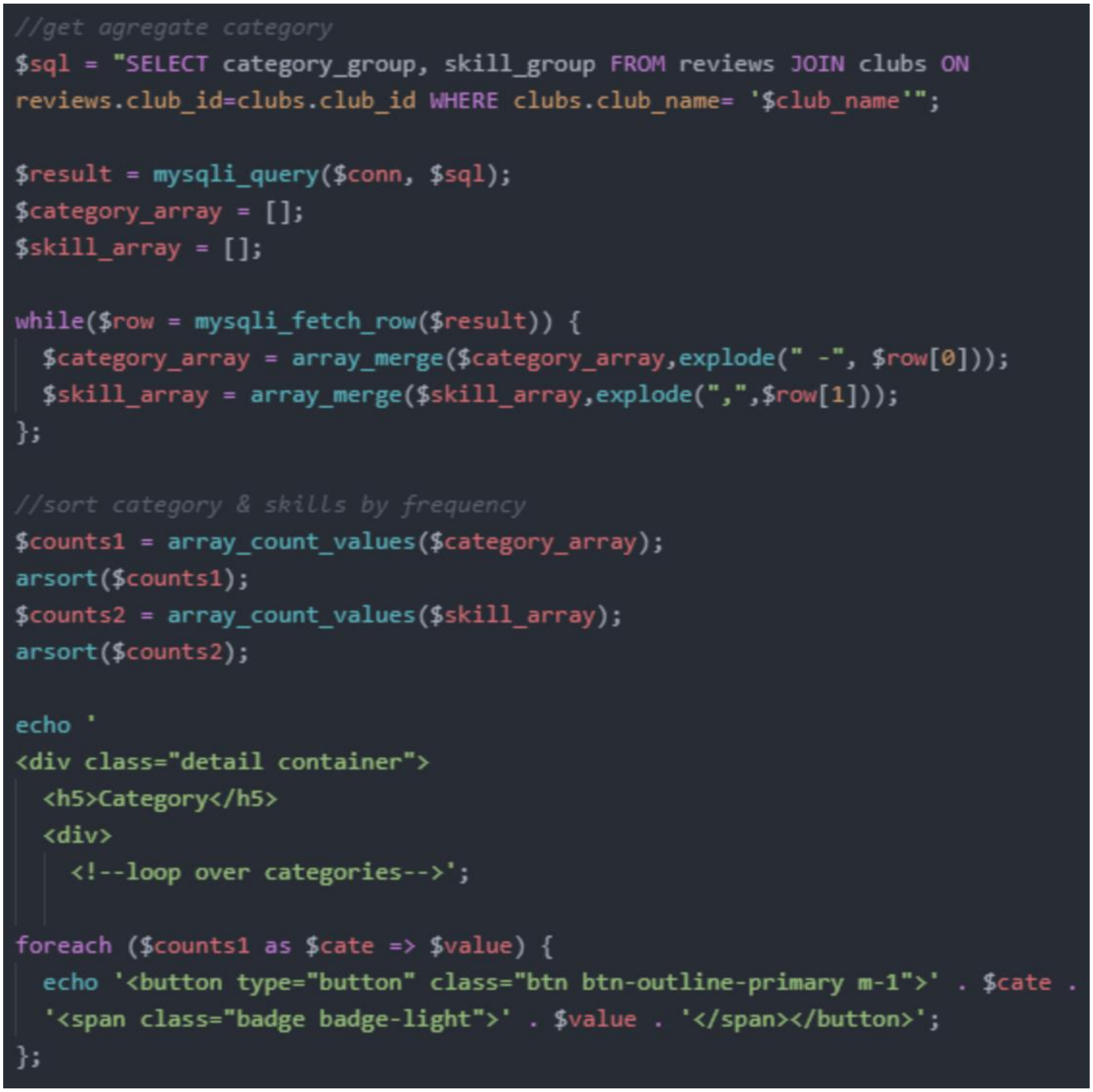


```
 function populateClubs (array) {
   document.getElementById('club-listing').innerHTML ='';
   var clubList = document.getElementById('club-listing'),
       li = document.createElement('li'),
       clone;
   array.forEach(function (array_element, index) {
       clone = li.cloneNode();
       clone.classList.add("list-group-item");
       clone.textContent = array_element;
       clone.innerHTML = '<a href="https://datalab3.bus.sfu.ca/ttn25/project/
       clubpage.php?club=' + array_element + '">' + array_element + '</a>';
       clubList.appendChild(clone);
   });
 }
```

1) We first used a piece of php code (shown in first screenshot) to get a list of clubs in the form of an array variable

2) We then passed this to store as a JavaScript array (shown in second screenshot) to be used as parameters for the populate clubs function, which changes the content of an empty <ul> element.

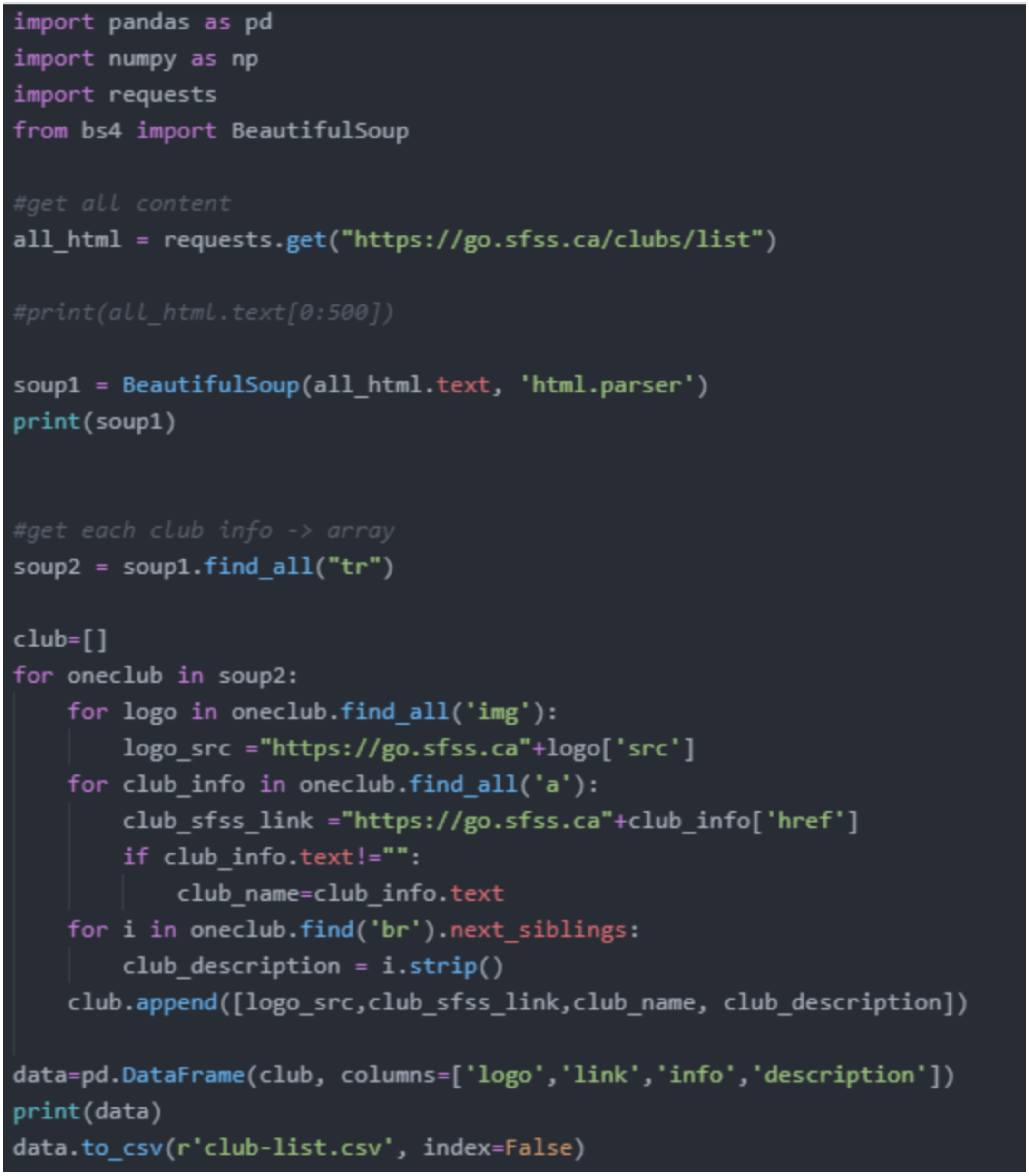
Dealing with Input Data Having Multiple Values:



```php
//get agregate category
$sql = "SELECT category_group, skill_group FROM reviews JOIN clubs ON
reviews.club_id=clubs.club_id WHERE clubs.club_name= '$club_name'";

$result = mysqli_query($conn, $sql);
$category_array = [];
$skill_array = [];

while($row = mysqli_fetch_row($result)) {
  $category_array = array_merge($category_array,explode(" -", $row[0]));
  $skill_array = array_merge($skill_array,explode(",",$row[1]));
};

//sort category & skills by frequency
$counts1 = array_count_values($category_array);
arsort($counts1);
$counts2 = array_count_values($skill_array);
arsort($counts2);

echo '
<div class="detail container">
  <h5>Category</h5>
  <div>
    <!--loop over categories-->';

foreach ($counts1 as $cate => $value) {
  echo '<button type="button" class="btn btn-outline-primary m-1">' . $cate .
  '<span class="badge badge-light">' . $value . '</span></button>';
};
```

Even though not ideal, we want to keep things simple by converting array input (checkbox and tags input) into string to store in the database and manipulate it again to show on the website. If we have time, we might either create a new table to store them or convert them to JSON objects.

Web Scraping:



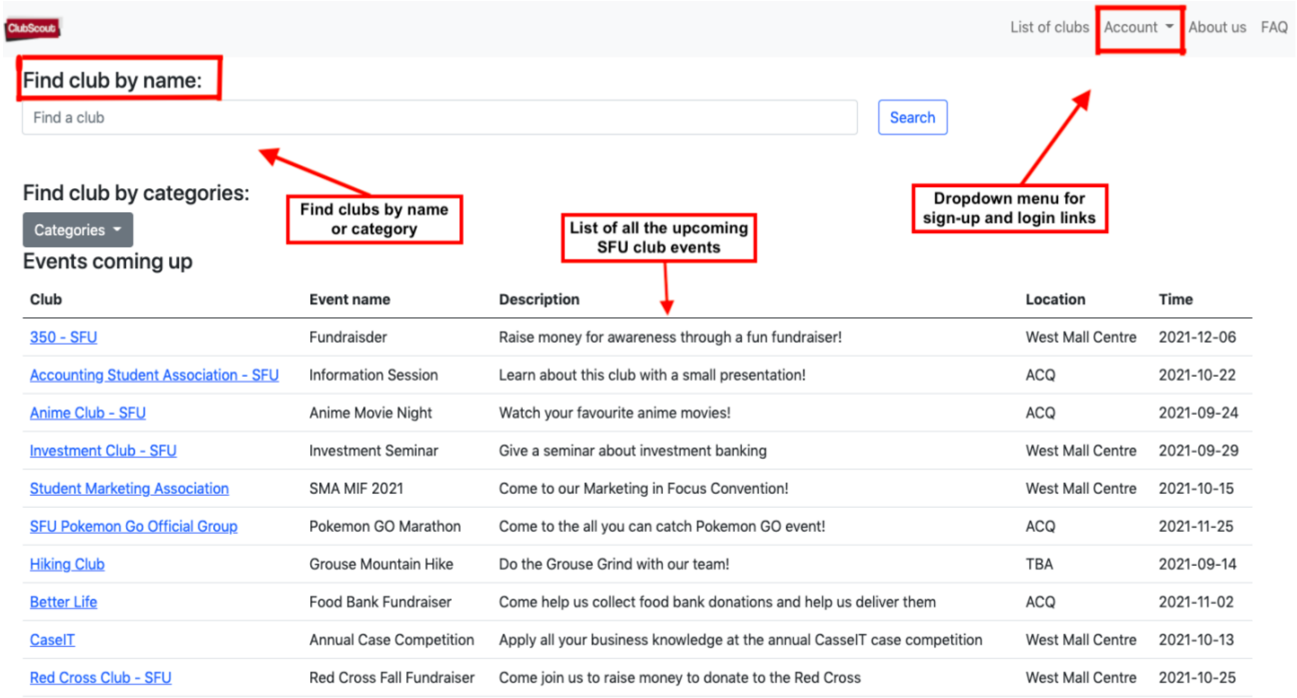
```python
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup

#get all content
all_html = requests.get("https://go.sfss.ca/clubs/list")

#print(all_html.text[0:500])

soup1 = BeautifulSoup(all_html.text, 'html.parser')
print(soup1)


#get each club info -> array
soup2 = soup1.find_all("tr")

club=[]
for oneclub in soup2:
    for logo in oneclub.find_all('img'):
        logo_src ="https://go.sfss.ca"+logo['src']
    for club_info in oneclub.find_all('a'):
        club_sfss_link ="https://go.sfss.ca"+club_info['href']
        if club_info.text!="":
            club_name=club_info.text
    for i in oneclub.find('br').next_siblings:
        club_description = i.strip()
    club.append([logo_src,club_sfss_link,club_name, club_description])

data=pd.DataFrame(club, columns=['logo','link','info','description'])
print(data)
data.to_csv(r'club-list.csv', index=False)
```

This was outside of this course's context, however, we learned how to retrieve data from the original SFSS club listing using Beautifulsoup instead of populating our database with random data.

System Prototype

Below are the screenshots highlighting the use cases and other functionalities provided by the system.

Homepage



Find club by name:

Find club by categories:

Events coming up

Find clubs by name or category

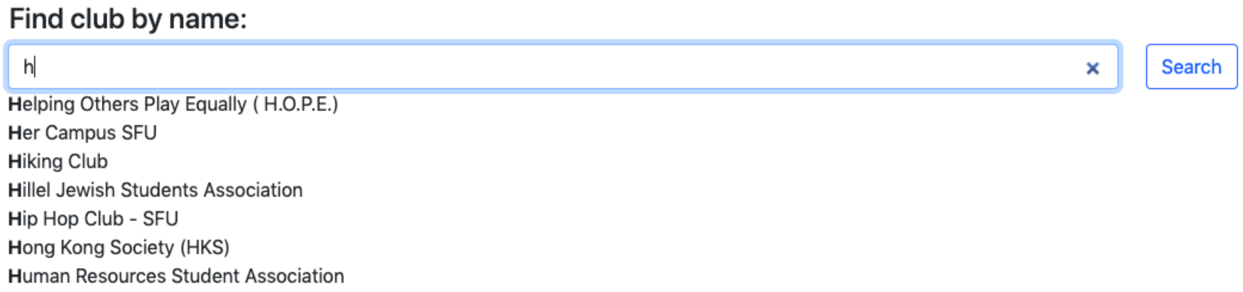
List of all the upcoming SFU club events

Dropdown menu for sign-up and login links

Club	Event name	Description	Location	Time
350 - SFU	Fundraisder	Raise money for awareness through a fun fundraiser!	West Mall Centre	2021-12-06
Accounting Student Association - SFU	Information Session	Learn about this club with a small presentation!	ACQ	2021-10-22
Anime Club - SFU	Anime Movie Night	Watch your favourite anime movies!	ACQ	2021-09-24
Investment Club - SFU	Investment Seminar	Give a seminar about investment banking	West Mall Centre	2021-09-29
Student Marketing Association	SMA MIF 2021	Come to our Marketing in Focus Convention!	West Mall Centre	2021-10-15
SFU Pokemon Go Official Group	Pokemon GO Marathon	Come to the all you can catch Pokemon GO event!	ACQ	2021-11-25
Hiking Club	Grouse Mountain Hike	Do the Grouse Grind with our team!	TBA	2021-09-14
Better Life	Food Bank Fundraiser	Come help us collect food bank donations and help us deliver them	ACQ	2021-11-02
CaseIT	Annual Case Competition	Apply all your business knowledge at the annual CasseIT case competition	West Mall Centre	2021-10-13
Red Cross Club - SFU	Red Cross Fall Fundraiser	Come join us to raise money to donate to the Red Cross	West Mall Centre	2021-10-25

On the homepage of ClubScout, users can search for specific clubs by name, find clubs based on category filters, or view upcoming club events. Users can also access quick links, such as login, from the very top of the page.

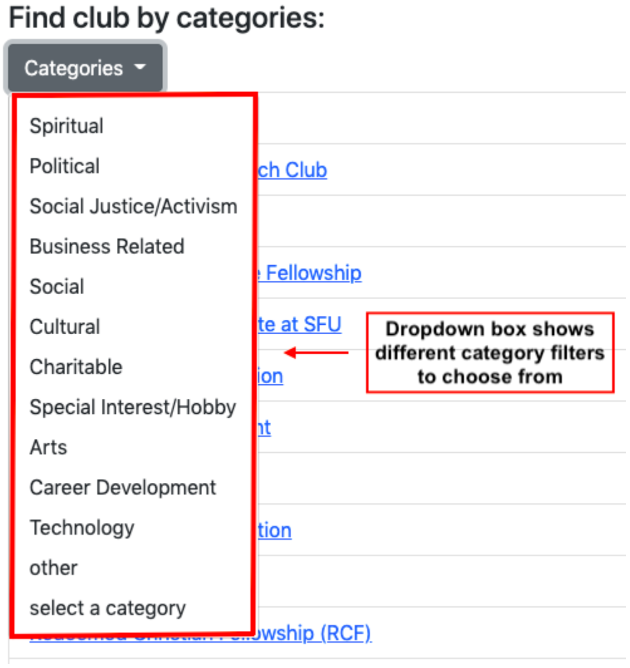
Search for a Club by Name


Find club by name:

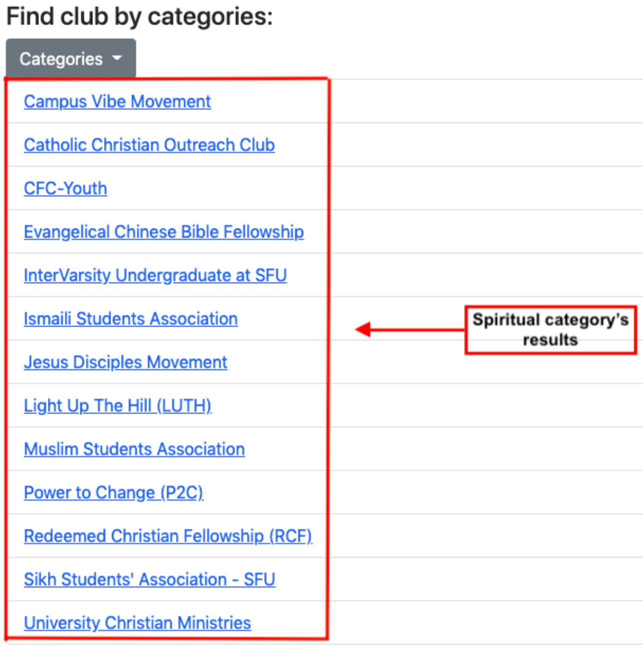
Helping Others Play Equally (H.O.P.E.)
Her Campus SFU
Hiking Club
Hillel Jewish Students Association
Hip Hop Club - SFU
Hong Kong Society (HKS)
Human Resources Student Association

Users can type in letters or keywords to search for clubs. Search results will show up underneath the search bar based on the typed input. To reset the search bar, you can click the "x" on the far-right of the search bar or by pressing the backspace key.

Search for a Club by Category


Find club by categories:

Dropdown box shows different category filters to choose from

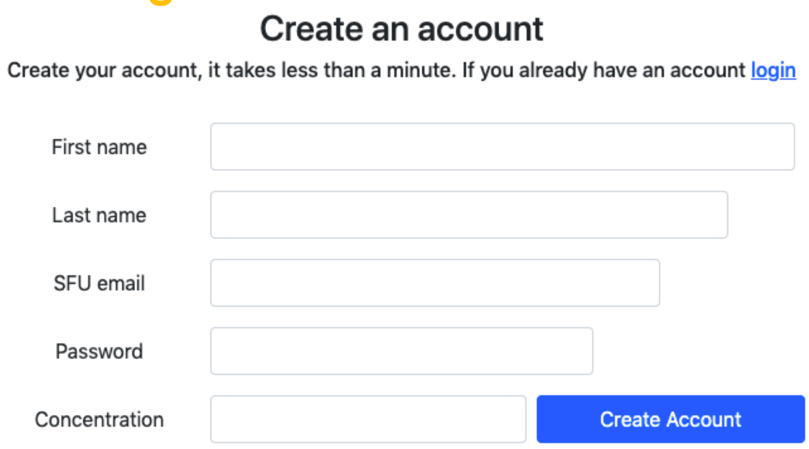

Find club by categories:

Campus Vibe Movement
Catholic Christian Outreach Club
CFC-Youth
Evangelical Chinese Bible Fellowship
InterVarsity Undergraduate at SFU
Ismaili Students Association
Jesus Disciples Movement
Light Up The Hill (LUTH)
Muslim Students Association
Power to Change (P2C)
Redeemed Christian Fellowship (RCF)
Sikh Students' Association - SFU
University Christian Ministries

Spiritual category's results

Clubs can also be searched by looking at a particular club category. Above shows a list of pre-determined SFU club categories that can be selected from a drop-down list. Once a selection has been made, a list of all clubs within that category will be displayed. Users can click a club to be directed to the link for that club webpage.

Creating an Account



Create an account

Create your account, it takes less than a minute. If you already have an account login

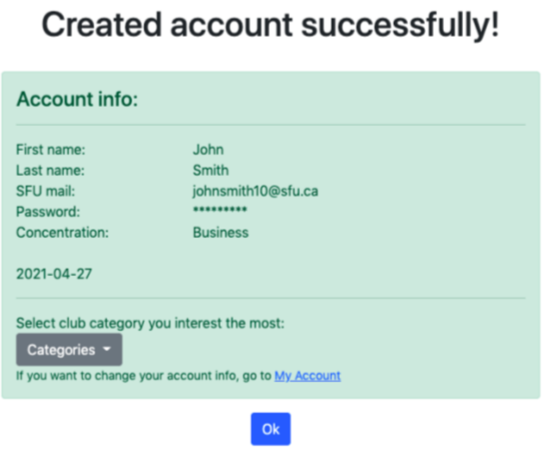
First name

Last name

SFU email

Password

Concentration



Created account successfully!

Account info:

First name: John
Last name: Smith
SFU mail: johnsmith10@sfu.ca
Password: *********
Concentration: Business

2021-04-27

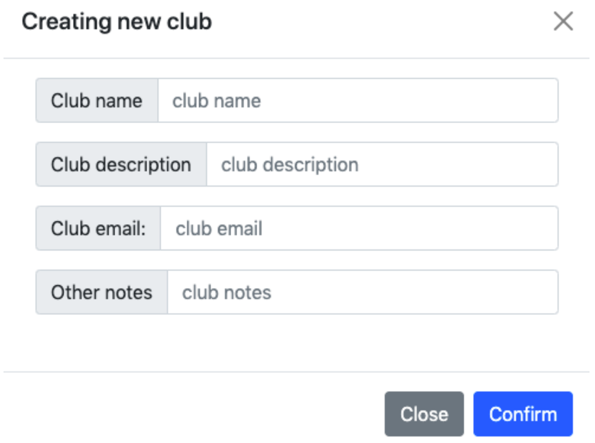
Select club category you interest the most:

If you want to change your account info, go to My Account

Once a user has clicked "Create Account" on the home page, they will fill in their first name, last name, SFU email, password, and SFU concentration.

Once they submit their form, a confirmation containing all the users account information inputs will appear. From there, users can select a club category which interests them the most.

Creating a New Club



Creating new club

Club name: club name

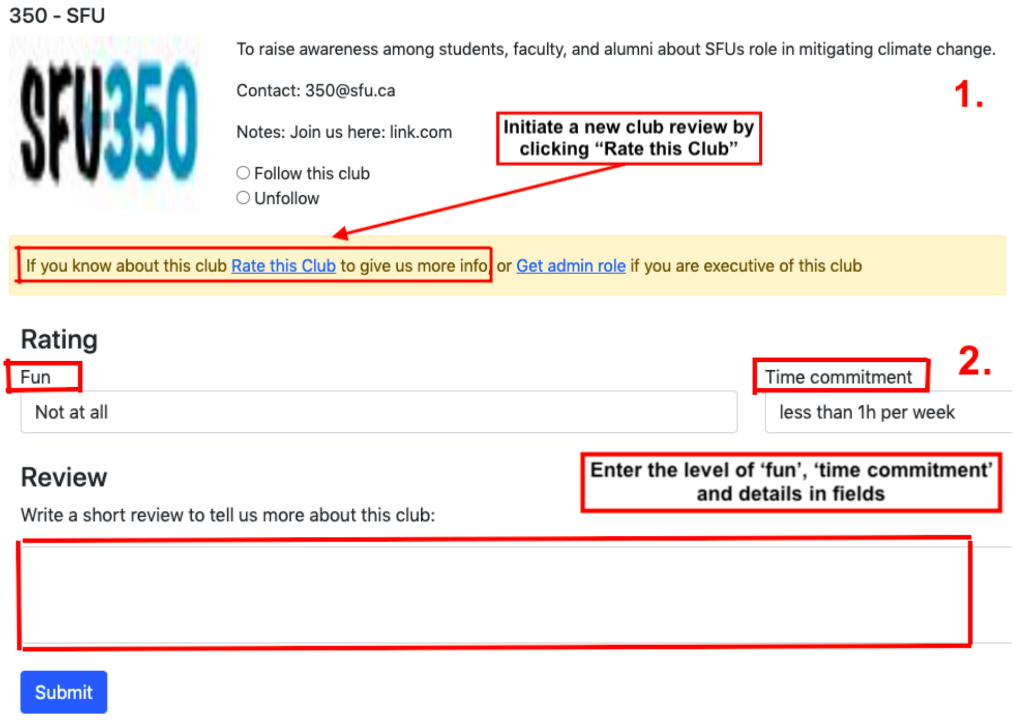
Club description: club description

Club email: club email

Other notes: club notes

Once an account is created, new clubs can be added from the "My Account" page. A user will input the club's name, description, email, and other extra notes if needed. Once the form is submitted, the club will be pending until a website admin verifies that club listing. The user who creates a club listing will become the admin for that club.

Writing a Club Review



350 - SFU

To raise awareness among students, faculty, and alumni about SFUs role in mitigating climate change.

Contact: 350@sfu.ca

Notes: Join us here: link.com

○ Follow this club
○ Unfollow

If you know about this club Rate this Club to give us more info or Get admin role if you are executive of this club

1.

Initiate a new club review by clicking "Rate this Club"

Rating

Fun: Not at all

Time commitment: less than 1h per week

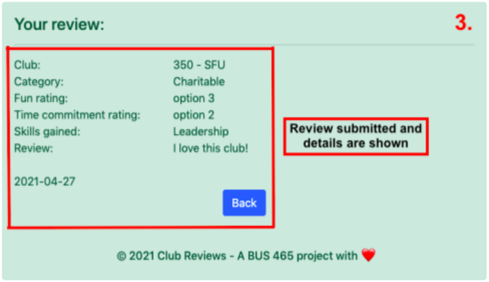
2.

Enter the level of 'fun', 'time commitment' and details in fields

Review

Write a short review to tell us more about this club:

Thank you for your input!



3.

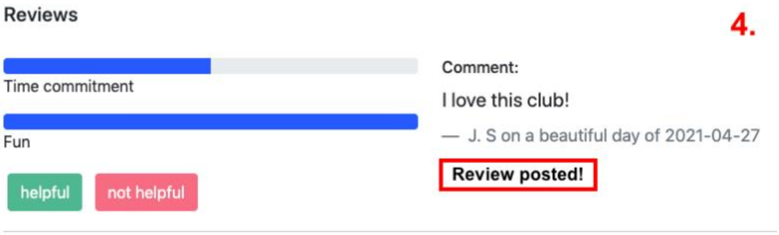
Your review:

Club:	350 - SFU
Category:	Charitable
Fun rating:	option 3
Time commitment rating:	option 2
Skills gained:	Leadership
Review:	I love this club!

2021-04-27

Review submitted and details are shown

© 2021 Club Reviews - A BUS 465 project with ❤️



4.

Reviews

Comment:

I love this club!

— J. S on a beautiful day of 2021-04-27

Review posted!

1) When a user clicks a club webpage, they can write a review for that club based on their experience by clicking "Rate This Club".

2) From there, users will input what club category they associate the club with, what skills they learned, how fun they found the club, and the amount of time commitment per week they needed to dedicate to it.

3) Once the form has been submitted, a message will appear summarizing the user inputs. After a review has been submitted.

4) Finally, the review will be viewable on the club webpage as pictured above.

Conclusion

The inspiration for our system came after we realized how useful sites like CourseDiggers and RateMyProf are in students' course selection process each term. We believe there is a similar need in SFU's extra-curricular programs. The current page that the SFSS offers is very limited in functionality, as it only allows you to search and sign-up for clubs. This is similar to SFU's course enrollment system. It does not provide any insights or decision-making assistance like the two aforementioned sites offer for course selection. Our system solves this problem by giving students a hub where they could get recommendations on what clubs could be of interest to them, and what current members are saying about any clubs they may be interested in. This will all help students make a more informed decision on which club is right for them.

Through the course of making this system, a variety of programming tools we used. The combination of these tools allowed us to construct a prototype system for ClubScout that allows visitors to search for existing SFU clubs, write club reviews, as well as register for an account. Registered users are also allowed to request club admin roles and are made into club admins after verification. Through the system, users can see clubs which are recommended to them and can also follow/unfollow clubs.

While we have created a solid foundation for ClubScout, the system still has some limitations which should be enhanced to offer the best user experience. First off, the verification process needs to be connected with SFU and fully implemented to ensure only SFU students become users. Secondly, a more thorough club recommendations system which takes multiple variables into account when recommending clubs to users. Finally, the introduction of new functionalities like a site forum, or a way to message and converse with club admins on the site will enhance user experience.

Overall, the prototype of this system will allow ClubScout to gauge market interest in the idea behind this system. This is crucial, as well will get a better sense of the demand and scope that this project can become. As we move forward, we believe that maintaining communication with our users throughout the project and lifespan of our system is necessary to meet changing user demands. If our system succeeds in providing a seamless experience to its users and the demand is there, ClubScout has the potential to expand beyond just SFU, spanning multiple Universities across Canada.

Now feel free to browse ClubScout's prototype website for yourself!
→ https://datalab3.bus.sfu.ca/ttn25/project/home.php

For club admin access input the following details at log-in:
→ Username: ttn25@sfu.ca Password: 123456