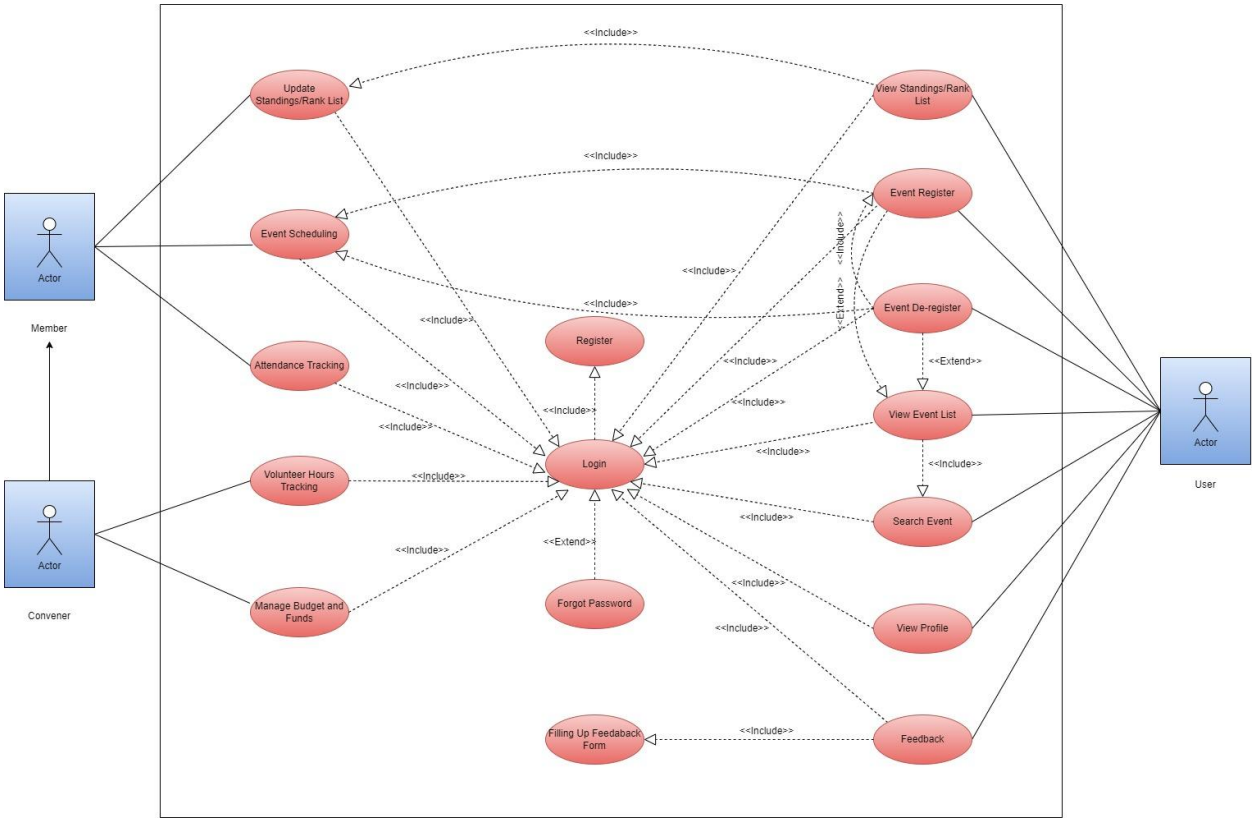
Update Standings/Rank List
View Standings/Rank List
Event Scheduling
Event Register
Event De-register
Attendance Tracking
Register
View Event List
Login
Volunteer Hours Tracking
Search Event
Manage Budget and Funds
Forgot Password
View Profile
Filling Up Feedaback Form
Feedback
Actor
Member
Actor
Convener
Actor
User
<<Include>>
<<Extend>>

Use Case	User Login
Actors	User, System
Goal	To log in to the system in order to access the features of the programming club website.
Pre Conditions	The user must have already registered an account on the College Programming Club website.
Post Conditions	The user is logged in to their account on the College Programming Club website and can access all features and resources available to registered users.
Basic Flow	1. The user navigates to the website's login page and clicks on the "Login" button. 2. The website displays a form prompting the user to enter their username and password. 3. The user enters their username and password into the form and clicks on the "Login" button. 4. The website verifies the user's credentials and checks if the account is active and not suspended or banned. 5. If the user's credentials are valid, the website grants access to the user's account and redirects them to their account dashboard.
Alternative flow	1.1 If the user is not yet registered, they can click on the "Sign Up" button to create a new account. 3.1 If the user forgets their password, they can click on the "Forgot Password" button and enter their registered email address to receive a password reset link. 4.1 If the user's credentials are invalid, the website displays an error message informing the user that their login attempt was unsuccessful and prompts them to try again. 4.2 If the user enters incorrect login credentials multiple times, the website may temporarily suspend or permanently ban their account.

Use Case	User Registration
Actors	User, System
Goal	To create a user account and become a registered user of the college programming club.
Pre Conditions	1. The user has access to an active internet connection and a web browser. 2. The user has not previously registered an account on the website.
Post Conditions	1. The user is successfully registered on the website. 2. The user can log in to their account using their credentials.
Basic Flow	1. The user navigates to the registration page of the college programming club website. 2. The user fills out the required fields, including their name, email address, and password. 3. The user agrees to the terms and conditions of using the website. 4. The user submits the registration form. 5. The system verifies the email address provided by the user. 6. The system creates a new user account with the provided information. 7. The user is redirected to the login page and can now log in to their account.
Alternative flow	3.1 If the user does not agree to the terms and conditions of using the website, the system will not allow them to register an account. 5.1 If the user provides an email address that is already registered on the website, the system will display an error message and ask the user to try again with a different email address.

Use Case	Search Events
Actors	User, System
Goal	The goal of the search event is to allow users to find relevant information about past and upcoming events of the college programming club easily and quickly.
Pre Conditions	1. The user is already logged in to the system. 2. The user is looking for information related to past or upcoming events.
Post Conditions	1. The user is presented with relevant search results related to past and upcoming events of the college programming club. 2. The user is able to navigate to the event page to get more information about the event.
Basic Flow	1. The user types in keywords related to the event they are searching for. 2. The website searches its database for any events that match the keywords. 3. The website displays the search results on the search results page. 4. The user can click on any search result to get more information about the event.
Alternative flow	2.1 If the website does not have any events that match the user's search keywords, the website displays a message that no results were found.

Use Case	Event Scheduling
Actors	Programming club leadership team, System
Goal	The goal of the event scheduling feature for the college programming club website is to allow club members to schedule and manage events related to programming and technology in an organized and efficient way.
Pre Conditions	1. The user must be a registered member of the programming club website. 2. The member must have the necessary permissions to schedule events. 3. The member must have access to the event scheduling feature on the website.
Post Conditions	1. The event details including date, time, location, and description are saved in the database. 2. The event details are visible to all registered club members.
Basic Flow	1. The member logs into the programming club website. 2. The member navigates to the event scheduling feature. 3. The member clicks on the "Add Event" button. 4. The member fills in the event details including the event name, date, time, location, and description. 5. The member selects the target audience for the event (e.g. all members, specific member types, or a custom list). 6. The member reviews the event details and selects the "Create Event" button 7. The event details are saved in the database and are visible to all users.
Alternative flow	1. If the member tries to schedule an event on a date or time slot that is already taken, an error message will appear, and the user will be prompted to select a different date or time. 2. If the member tries to schedule an event without filling in all the necessary details, an error message will appear, and the user will be prompted to fill in the missing information.

Use Case	Event Registration
Actors	User, System
Goal	To register for programming club events through the club website.
Pre Conditions	1. The user must be a registered member of the programming club. 2. The user must be logged in to their account on the club website. 3. The event must be listed on the club website and available for registration.
Post Conditions	1. The user's registration information is saved in the club's database. 2. The event attendance record is updated to reflect the user's registration. 3. The user is added to the event attendee list. 4. The user can view their registration information in their account on the club website.
Basic Flow	1. The user navigates to the event registration page on the club website. 2. The user selects the event they wish to register for. 3. The user confirms their registration details (e.g. name, email address, etc). 3. The user submits their registration. 4. The system verifies the user’s details and saves the user's registration information in the club's database. 5. The user can view their registration information in their account on the club website.
Alternative flow	4.1 If the user is not a registered member of the programming club, they are prompted to register on the website before they can register for events. 4.2 If the user is not logged in to their account on the club website, they are prompted to do so before they can register for events.

Use Case	Event De-registration
Actors	User, Programming club leadership team, System
Goal	to allow users to cancel their registration for a club event and to update the event attendance record accordingly.
Pre Conditions	1. The user is logged in to their account on the programming club website. 2. The user is registered for the event for which they wish to deregister.
Post Conditions	1. The event attendance record is updated to reflect the user's deregistration. 2. The user is removed from the event attendee list.
Basic Flow	1. The user navigates to the event registration page on the programming club website. 2. The user clicks on the "Deregister" button next to the event they wish to deregister from. 3. The user confirms their deregistration. 4. The event attendance record is updated to reflect the user's deregistration. 5. The user is removed from the event attendee list.
Alternative flow	If the user encounters an issue with the event deregistration feature or needs additional assistance: 1. The user can contact the club leadership team or website administrator for assistance. 2. The club leadership team or website administrator can investigate the issue and provide assistance as needed. 3. If necessary, the user can request to be manually removed from the event attendee list.

Use Case	Update Standings/Rank List
Actors	Programming club leadership team, System
Goal	To update the standings/rank list of the programming club's website with the latest competition results.
Pre Conditions	1. The club member updating the standings/rank list has authorized access to the website's backend. 2. The competition results have been collected and verified. 3. The current standings/rank list is accessible.
Post Conditions	1. The standings/rank list has been updated with the latest competition results. 2. The updated standings/rank list is visible and accessible to all users.
Basic Flow	1. Log in to the website's backend with authorized credentials. 2. Navigate to the standings/rank list page. 3. Select the option to update the standings/rank list. 4. Enter the latest competition results, including the names of participants and their scores. 5. Review the updated standings/rank list for accuracy. 6. Save and publish the updated standings/rank list.
Alternative flow	If the authorized member encounters technical issues or errors while updating the standings/rank list: 1. The authorized member can contact the technical support team/system administrator for assistance. 2. The technical support team/system administrator can investigate the issue and provide assistance as needed, in order to resolve the issue.

Use Case	Attendance Tracking
Actors	Programming club member, System
Goal	To provide an easy and efficient way for club members to track attendance for various events, meetings, and activities organized by the club, as well as to provide attendance reports to the club's members and advisors.
Pre Conditions	1. The college programming club website is up and running. 2. Member has signed up for the club and has login credentials. 3. The club has scheduled meetings and events that members are expected to attend.
Post Conditions	1. The club members can easily track their attendance at club meetings and events. 2. The club members have access to attendance records for each meeting and event.
Basic Flow	1. The member navigates to the attendance tracking page. 2. The member selects the meeting or event for which they want to track attendance. 3. The member clicks on the “Track Attendance” button corresponding to that meeting or event. 4. The system displays records of attendance for the selected meeting or event.
Alternative flow	If the authorized member encounters technical issues or errors while tracking attendance: 1. The authorized member can contact the technical support team/system administrator for assistance. 2. The technical support team/system administrator can investigate the issue and provide assistance as needed, in order to resolve the issue.

Use Case	Volunteer Hours Tracking
Actors	Programming club member, System
Goal	The goal of the volunteer hours tracking feature on the college programming club website is to provide a mechanism for members to record and track the number of hours they contribute to the club through various activities such as organizing events, participating in meetings, and mentoring other members.
Pre Conditions	1. The member is logged in to their account on the programming club website. 2. The member has participated in one or more activities related to the programming club.
Post Conditions	1. The member's recorded volunteer hours are updated and displayed on their profile page. 2. The total volunteer hours contributed by all members are displayed on a dedicated page on the website.
Basic Flow	1. The member logs in to their account on the programming club website. 2. The member navigates to the volunteer hours tracking feature. 3. The member selects the type of activity they participated in (e.g. event organization, meeting participation). 4. The member enters the number of hours they spent on the activity. 5. The member submits the information, and the volunteer hours are recorded and updated on their profile page. 6. The member can view their recorded volunteer hours on their profile page.
Alternative flow	If the member encounters an error or enters incorrect information: 1. An error message is displayed, indicating the issue. 2. The member can correct the error and resubmit the information. 3. If the issue persists, the member can contact the system administrator for assistance.

Use Case	User Feedback
Actors	User, Programming club leadership team, System
Goal	to allow users to provide feedback club's website and its activities, and to enable the club leadership team to review and respond to feedback to improve the club's offerings.
Pre Conditions	1. The user is logged in to their account on the programming club website. 2. The user has interacted with the club website or has attended one or more club activities.
Post Conditions	1. The user's feedback is recorded and saved in the feedback database. 2. The club leadership team can review and respond to the feedback. 3. If necessary, the club leadership team can take actions to address any issues raised in the feedback.
Basic Flow	1. The user navigates to the feedback section of the programming club website. 2. The user selects the category of feedback they wish to provide (e.g. website feedback, event feedback, club activities feedback). 3. The user types their feedback in the provided form. 4. The user submits the feedback. 5. The feedback is recorded and saved in the feedback database.
Alternative flow	If the user encounters an issue with the feedback feature: 1. The user can contact the club leadership team or system administrator for assistance. 2. The club leadership team or system administrator can investigate the issue and provide assistance as needed. 3. If necessary, the user can provide feedback through alternative means (e.g. email, in-person).

Non-functional requirements specify criteria or constraints that the system must meet but do not describe specific functionalities. Some non-functional requirements that could be applicable to a programming club website are:

1. Performance:
    - The system should be quick and responsive, with minimal lag times for a better user experience.
    - The system should be able to handle high volumes of concurrent users and high peak loads, with minimal downtime.

2. Scalability:
    - The system should be able to handle an increasing number of users or requests without losing performance or functionality. Scalability is an important aspect of the programming club website, as it ensures that the website can grow and adapt to changing user needs and usage patterns.

    - Scalability is concerned with how well the website can handle an increasing load of users or requests, without experiencing performance degradation or downtime. A scalability requirement for a programming club website could be to support a certain number of concurrent users or requests, with the ability to scale up or down as needed.

    - To ensure scalability, the website must be designed with a scalable architecture that can handle increased traffic and usage. The website architecture should be able to distribute traffic evenly across multiple servers and handle spikes in traffic without slowing down or crashing.

    - Caching, load balancing, database optimization, cloud hosting, etc are some of the techniques that can be used to improve the scalability of a programming club website.

3. Availability:
    - The system should remain operational and accessible to users at all times, providing them with a reliable and consistent user experience. Availability is a critical aspect of the programming club website, as it directly affects user satisfaction, usability, and overall website performance.

    - The availability aspect is concerned with the percentage of time that the website is operational and accessible to users, often referred to as the website's uptime. The system should have a high availability rate(24x7) and should be designed to minimize downtime.

    - The availability of a programming club website can be affected by various factors, such as hardware or software failures, network outages, maintenance or upgrades, and

security breaches. Therefore, to ensure high availability, the website must be designed with redundancy and fault-tolerance mechanisms. For example, the website can be hosted on multiple servers in different locations, and load balancers can distribute traffic to the available servers to ensure uninterrupted service.

4. Security:

- The system should be able to protect sensitive information, prevent unauthorized access, and ensure the confidentiality, integrity, and availability of the website's data and resources in order to provide a secure and trustworthy user experience. Security is a critical aspect of a programming club website, as it directly affects user trust and privacy.

- The security aspect is concerned with protecting the website from various security threats, such as hacking, phishing, malware, and denial of service attacks. A security requirement for a programming club website could be to ensure that user data is encrypted in transit and at rest, access control mechanisms are in place to restrict access to sensitive data and resources, and regular security audits and vulnerability assessments are conducted.

- To ensure security, the website architecture should incorporate security measures, such as firewalls, intrusion detection systems, and data encryption, to protect against security threats.

5. Usability:

- The usability aspect of a programming club website refers to the ease with which users can interact with the website and perform desired tasks. Usability is an important aspect as it directly affects user satisfaction and the overall user experience.

- The usability aspect is concerned with making the website easy to use and intuitive, with minimal user effort and cognitive load required to complete tasks. A usability requirement for a programming club website could be to ensure that the website is easy to navigate, with clear and consistent labeling and organization of content, and that user feedback is provided in a timely and useful manner.

- To ensure usability, the website must be designed with user-centered design principles in mind. The website should be designed based on user needs and preferences. Consistent design, clear labeling, organization, user feedback, accessibility, and user testing are some of the techniques that can be used to improve the usability of a programming club website.

6. Reliability:
   - The reliability aspect of a programming club website refers to the ability of the website to function correctly and consistently without errors or downtime. Reliability is an important aspect as it affects user trust and confidence in the website.

   - To ensure reliability, the website must be designed and implemented in a way that minimizes errors and downtime.

   - Some techniques that can be used to improve the reliability of a programming club website include using reliable hardware and software components, testing the website thoroughly to identify and handle bugs and errors, and implementing backup and recovery mechanisms in case of system failures.

7. Maintainability:
   - The maintainability aspect of a programming club website refers to the ease with which the website can be maintained and updated over time. Maintaining and updating a website is important to ensure that it remains relevant and functional for users and that it can adapt to changing user needs and technological advancements.

   - To ensure maintainability, the website must be designed and implemented in a way that allows for easy and efficient updates and maintenance.

   - Some techniques that can be used to improve the maintainability of a programming club website include using modular and well-organized code, implementing version control systems, providing clear documentation, and ensuring code quality.

8. Interoperability
   - The interoperability aspect of a programming club website refers to the ability of the website to integrate and communicate with other systems or platforms. In other words, it is the ability of the website to work with other systems or platforms in a seamless and efficient manner.

   - Interoperability is important for a programming club website because it may need to communicate with other systems or platforms, such as external APIs, third-party services, or other websites. For example, a programming club website may need to integrate with a payment gateway or a social media platform to enable users to pay for events or share information about the club.

   - To ensure interoperability, the website must be designed and implemented in a way that allows it to communicate and integrate with other systems or platforms. Some techniques that can be used to improve the interoperability of a programming club website include using standardized protocols and data formats such as HTTP or JSON,

and providing well-documented APIs for external systems to interact with the website while ensuring security.