

IT-314 Software Engineering



Group: 27 College Programming Club

IDs	Name
202001241	RIKI ANILBHAI PARIKH
202001256	TUSHAR DHAR
202001265	SANGANI KISHAN PARESHKUMAR
202001267	BHATT MEET DHAVALBHAI
202001268	MANSARA FENIL LALITBHAI
202001272	ARCHIT SINGH
202001276	OM CHALODIYA
202001426	VADODIA HARSHIT JAGDISHBHAI
202001433	MODIYA DEEP ASHOKBHAI
202001445	PATEL KRUNAL MAHESHBHAI

INDEX

1. Introduction.....	3
1.1 Scope.....	3
1.2 Purpose.....	3
2. Features.....	4
3. Intended audience.....	6
4. Functional Requirements.....	7
5. Non- Functional Requirements.....	8
6. Use Case Descriptions.....	12
7. Operating Environment.....	21
8. Assumptions.....	22
9. Constraints.....	23

1. Introduction:

- Users generally need an online platform related to the programming club in order to perform activities like search events, event registration, event de-registration, etc., while on the other hand, club members usually need such a platform to perform activities like event scheduling, attendance tracking, updating the standings list, etc.
- The programming club website contributes to these scenarios by increasing efficiency, reducing errors, and improving the overall user experience. It also allows members and users to manage their programming-related activities efficiently. Additionally, the system can enhance security and privacy by securely storing user information in a consistent and reliable manner.

1.1 Scope:

- The need for the programming club website is to provide an online platform for club members and other users to organize, maintain, and access programming-related activities, as well as to promote the club and its activities to a wider audience.
- The programming club website can help to build a community of like-minded individuals, provide resources and opportunities for users to learn and grow, and improve the club's visibility and impact.

1.2 Purpose:

- The Programming Club at Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT) is a student-run organization that aims to foster interest and skills in competitive programming and related fields among the students of the college by organizing various events and programming contests throughout the year to provide students with opportunities to learn, practice, and showcase their skills.

2. Features:

1) User registration:

- The system supports the registration facility for its stakeholders (club members) and users.

2) User login:

- The system supports the login facility for its stakeholders (club members) and users.

3) Forgot password:

- The system supports the forgot password feature for the users to reset their password if they forget or want to change their password.

4) User authentication and authorization:

- This system verifies the identity of users who log in to the website and controls their access to different parts of the website based on their authorization level, such as event scheduling, event registration, etc.

5) Event Management:

- The system allows club members to create, modify, and delete events, as well as manage event-related activities such as add event, edit event, delete event, event registration, event participation, etc.

6) Database Management:

- The system stores and manages user account information, event details, and other club-related information.

7) Feedback management:

- The system stores and manages the feedback provided by the users.

8) User profile management:

- The system allows users to update and manage their profiles accordingly.

9) Source management:

- The system provides resources for members, such as links to programming tutorials, coding challenges, and online tools. This can help members improve their programming skills and stay up-to-date with the latest trends and technologies.

10) User-friendly interface:

- A clean, easy-to-use interface that makes it simple for members to navigate the site and find the information they need.

11) Mobile responsiveness:

- The website is optimized for viewing on a variety of devices, including desktops, smartphones, and tablets.

12) Security and Privacy:

- The system ensures security and protection against threats to ensure the safety of the user's personal information.

13) Real-time:

- The system supports the real-time updation of the information.

3. Intended audience:

User categories and their privileges

- **Club members:**
 - Add event
 - Edit event
 - Delete event
 - View participants
 - Track registered users
 - Remove registered user
 - Manage feedback
 - Manage resources

- **Club convener:**
 - Add event
 - Edit event
 - Delete event
 - Track registered users
 - Remove registered user
 - Manage feedback
 - Manage resources
 - Add club member
 - Remove club member

- **College students:**
 - Event register
 - Event de-register
 - View events
 - View resources
 - View profile
 - Edit profile
 - Feedback

4. Functional requirements:

1) User registration:

- The system supports the registration facility for its stakeholders (club members) and users.

2) User login:

- The system supports the login facility for its stakeholders (club members) and users.

3) Event registration/de-registration:

- The system supports event registration and de-registration facilities for its users.

4) Event scheduling/de-scheduling:

- The system only allows its stakeholders (club members) to schedule and de-schedule the events.

5) View events:

- The system allows all users to view the list of upcoming events and information about the upcoming events, like type, date, time, venue, etc.

6) Edit Event:

- The system only allows its stakeholders (club members) to edit the upcoming event details.

7) View event participation:

- The system only allows its stakeholders (club members) to view the participation list of the events.

8) Resources:

- The system allows club members to provide programming-related instructional materials like coding tutorials, coding challenges, etc.

9) Feedback:

- The system allows the users to give feedback regarding any programming club-related activities.

5. Non-functional requirements:

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 significant volumes of concurrent users and 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 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. 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) Easy Navigation:

- The website should be designed in such a way that users can easily navigate and find what they are looking for, such as club events, membership information, meeting schedules, and resources.

9) Mobile Responsiveness:

- The website should be optimized for different screen sizes and devices, including desktops, laptops, tablets, and smartphones.

10) Attractive and Engaging Design:

- The website design should be visually appealing and engaging, with a consistent layout, typography, and color scheme. The use of graphics, Images and multimedia can help to make the website more attractive and interactive.

6. Use-Case descriptions:

1) User registration

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	<ol style="list-style-type: none">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.3) System must be compatible.
Post Conditions	<ol style="list-style-type: none">1) The user is successfully registered on the website.2) The user can log in to their account using their credentials.
Basic Flow	<ol style="list-style-type: none">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 submits the registration form.4) The system verifies the email address provided by the user.5) The system creates a new user account with the provided information.6) The user is redirected to the login page and can now log in to their account.
Alternative flow	<ol style="list-style-type: none">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.2) 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.

2) User Login

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	<ol style="list-style-type: none">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	<ol style="list-style-type: none">1) If the user is not yet registered, they can click on the "Register" button to create a new account.2) 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.3) 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) If the user enters incorrect login credentials multiple times, the website may temporarily suspend or permanently ban their account.

3) Add Event

Use Case	Add Event
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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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.

4) Delete Event

Use Case	Delete Event
Actors	Club member, System
Goal	To delete the event
Pre Conditions	<ol style="list-style-type: none">1) The club member must be logged into the college programming club website.2) The club member must have the necessary permissions to delete an event.
Post Conditions	<ol style="list-style-type: none">1) The event is removed from the list of events on the website.2) Any associated data or records for the event are also deleted from the website's database.
Basic Flow	<ol style="list-style-type: none">1) The administrator selects the "Events" section of the website.2) The administrator selects the event that they want to delete from the list of events.3) The administrator selects the "Delete" option for the selected event.4) The website displays a confirmation dialog box to confirm the deletion of the event.5) The administrator confirms the deletion of the event.6) The website deletes the event from the list of events and updates the database.
Alternative flow	<ol style="list-style-type: none">1) If the event has already occurred, the website may prompt the administrator with a warning that deleting the event will also delete any associated data or records.

5) Event Registration

Use Case	Event Registration
Actors	User, System
Goal	To register for programming club events through the club website.
Pre Conditions	<ol style="list-style-type: none">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	<ol style="list-style-type: none">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	<ol style="list-style-type: none">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).4) The user submits their registration.5) The system verifies the user's details and saves the user's registration information in the club's database.6) The user can view their registration information in their account on the club website.
Alternative flow	<ol style="list-style-type: none">1) If the user has not registered on the programming club website, they are prompted to register on the website before they can register for events.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.

6) Event De-registration

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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<p>If the user encounters an issue with the event deregistration feature or needs additional assistance:</p> <ol style="list-style-type: none"> 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.

7) User Feedback

Use Case	User Feedback
Actors	User, Programming club leadership team, System
Goal	To allow users to provide feedback on the 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<p>If the user encounters an issue with the feedback feature:</p> <ol style="list-style-type: none"> 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).

8) View participants list

Use Case	View participants list
Actors	Club member, System
Goal	To view the participants list of the event.
Pre Conditions	1) The club member must be logged into the club website.
Post Conditions	1) The club member has successfully viewed the participants list. 2) The event for which the club member wishes to view the participants list has been scheduled.
Basic Flow	1) To view the participants list, the club member navigates to the events section. 2) The club member then navigates to the specific event whose participants list had to be viewed. 3) The club member then clicks on the "view participants" button. 4) The system displays the list of participants who had registered for that specific event.
Alternative flow	1) If a club member is not currently logged in to the club's website, they will need to log in before being able to view the list of registered participants.

9) Forgot Password

Use Case	Forgot Password
Actors	User, System
Goal	To reset the password.
Pre Conditions	<ol style="list-style-type: none"> 1) The user has previously registered an account with the college programming website. 2) The user has forgotten their password and cannot log in to their account.
Post Conditions	<ol style="list-style-type: none"> 1) The user has successfully reset their password and can log in to their account. 2) The user's account password has been updated in the system.
Basic Flow	<ol style="list-style-type: none"> 1) User navigates to the login page of the college programming website. 2) User clicks on the "Forgot Password?" 3) The system displays a page with a form for the user to enter their email address. 4) User enters their email address and clicks on the "Submit" button. 5) The system verifies that the email address entered is associated with a registered account. 6) The system sends an email to the user's email address with a link to reset their password. 7) User receives the email and clicks on the link. 8) The system displays a page with a form for the user to enter a new password. 9) User enters a new password and confirms it by entering it again. 10) The system verifies that the new password meets the website's password requirements. 11) The system updates the user's account with the new password and logs the user in.

	12) The system displays the user's account dashboard.
Alternative flow	1) If the email address entered is not associated with a registered account, the system displays an error message and asks the user to try again. 2) The new password does not meet the website's password requirements, the system displays an error message and asks the user to try again.

7. Operating environment:

- **Hardware Platform:**

- Processor: Intel Core i3 or higher
- RAM: 4GB or higher
- Storage: At least 10GB of free disk space

- **Operating System:**

- Windows 10 or higher
- macOS 10.14 or higher
- Linux (Ubuntu, Fedora, or any other popular distribution)

- **Web Server:**

- Apache or Nginx web server
- PHP (version 7.2 or higher)
- MySQL or MariaDB database

- **Web Browser:**

- The latest version of Google Chrome, Mozilla Firefox, Apple Safari, or Microsoft Edge

- **Other software components:**

- Git version control system
- Code editor (Sublime Text, Visual Studio Code, Atom, etc.)
- FTP client (FileZilla, Cyberduck, WinSCP, etc.).

8. Assumptions:

It is assumed that the users of this website will have all the necessary hardware and software required for running this application.

- **Target audience:** The website's target audience is likely to be college students interested in programming, coding, and software development.
- **Technical proficiency:** The website's audience is likely to have some level of technical proficiency, so the website can assume a basic understanding of programming concepts.
- **Content:** The website is likely to feature content related to programming, such as tutorials, articles, and links to external resources.
- **Design:** The website's design is likely to be clean, modern, and easy to navigate. It should also have a professional look and feel to reflect the club's goals and values.
- **Responsiveness friendly:** The website is likely to be optimized for viewing on a variety of devices, including desktops, smartphones, and tablets.
- **Real-time:** It is assumed that the information stored in the database gets updated in real-time.
- **Stable internet:** It is assumed that the users of this website will have a stable internet connection.
- **Consistency:** It is assumed that the database is consistent.
- **Integrity:** It is assumed that the integrity of the data is maintained throughout the database.
- **Security and Privacy:** The website is likely to ensure security and protection against threats to ensure the safety of the user's personal information.

9. Constraints:

- **Time:** The club may have a limited amount of time to develop and maintain the website, especially if members are busy with coursework and other activities.
- **Resources:** The club may have limited resources, such as technical expertise, to develop and maintain the website.
- **Software and hardware:** The amount of software and hardware required for the website is limited.
- **Computing power:** The amount of computing power available for the website is also limited.
- **Capacity:** The capacity of the website to handle simultaneous users is also limited.