# Software Requirements Specification

for

# University Programming Club Website

Version 1.0.0

Prepared by Jannatim Maisha

Software Developer

Submitted to Mahfuj Ahmed Jim

Lecturer

September 4, 2023

# Table of Contents

**1. Introduction....3**
1.1 Purpose....3
1.2 Document Conventions....3
Bold Text (Indicating Important Terms and Headings):....3
Monospaced Text (Representing Code, File Names, or Technical References):....3
[Square Brackets (Denoting Optional Features or Elements)]:....3
(Parentheses Providing Additional Clarification or Explanations):....3
1.3 Audience & Reading Suggestions....4
1.4 Project Scope....5
1.5 References....5
The development and implementation of the University Club Website will adhere to industry best practices and standards. The following references will serve as the basis for this project:....5
**2. Overall Description....7**
2.1 Product Perspective....7
2.2 Product Features....7
2.3 User Classes & Characteristics....8
● Members:....8
● Committee:....8
● Admins:....8
2.4 Operating Environment....8
2.5 Design & Implementation Constraints....10
2.6 User Documentation....10
2.7 Assumptions & Dependencies....11
**3. System Features....12**
3.1 User Features....12
Functional Requirements:....12
3.1.2 Event Management....12
Functional Requirements:....12
3.1.3 Membership Management....12
Functional Requirements:....12
3.1.4 Project Collaboration....14
Functional Requirements:....14
3.1.5 Blog and News Section....14
Functional Requirements:....14
3.2 Committee Features....15
Functional Requirements:....15
3.2.2 Event Management....15
Functional Requirements:....15
3.2.3 Email Communication....15

Functional Requirements:.....15
3.3 Admin Features.....17
Functional Requirements:.....17
Functional Requirements:.....17
Functional Requirements:.....17
3.3.4 Analytics and Reporting.....18
Functional Requirements:.....18
**4. External Interface Requirements.....19**
4.1 User Interface.....19
4.2 System Interface.....19
4.3 Hardware Interface.....20
4.4 Communication Interface.....20
**5. Non-Functional Requirements.....21**
5.1 Security.....21
5.2 Performance.....21
5.3 Usability.....22
5.4 Reliability.....22
5.5 Scalability.....22
5.6 Supportability.....22
**6. Other Requirements.....24**

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive outline of the requirements and functionalities for the development of an University Club Website. This document serves as a foundational guide for all stakeholders involved in the project, including developers, designers, testers, and project managers, ensuring a clear and shared understanding of the project's objectives and scope.

## 1.2 Document Conventions

**Bold Text (Indicating Important Terms and Headings):**

An important term in your club website project could be "Landing Page", which represents a critical aspect of enabling potential members to learn about the clubs goal and initiatives.

**Monospaced Text (Representing Code, File Names, or Technical References):**

In your platform's technical documentation, you might reference a specific code snippet like create_blog.cshtml when explaining the blog creation process.

**[Square Brackets (Denoting Optional Features or Elements)]:**

As an optional feature, consider "[Rate Blogs]", where customers have the choice to leave blog reviews and ratings but are not obligated to do so.

**(Parentheses Providing Additional Clarification or Explanations):**

When discussing user authentication, you can clarify it as "User Authentication (verifying user credentials such as username and password during login)".

## 1.3 Audience & Reading Suggestions

This document is intended for various stakeholders involved in the development, implementation, and maintenance of the Club Website. The primary audience includes:

- **Developers:** Developers will use this document as a reference to understand the technical requirements and functionalities needed for the website's development.
- **Designers:** Designers can refer to this document to align the user interface and user experience (UI/UX) design with the website's functional requirements.
- **Testers:** Testers will rely on this document to create test cases and validate that the website meets the specified requirements.
- **Project Managers:** Project managers will use this document to create project plans, allocate resources, and ensure that the project stays on track in terms of scope and objectives.
- **Stakeholders (Members, Committee, Admin):** Members, committee, and administrators may review this document to gain a high-level understanding of the website's capabilities and features.

Reading Suggestions:

- **Developers:** Focus on the technical details, system architecture, and database structure sections to understand the implementation requirements.
- **Designers:** Pay close attention to the user interface and user experience design guidelines mentioned in Section 4 (User Interface Design).
- **Testers:** Concentrate on the functional requirements and the test cases provided in Section 5 (Functional Requirements and Testing).
- **Project Managers:** Review the project scope, timeline, and resource requirements outlined in Section 1.4 (Project Scope).
- **Stakeholders:** Read the executive summary and overview sections (Sections 1 and 2) to grasp the website's purpose and key functionalities.

## 1.4 Project Scope

The project scope encompasses the development of a comprehensive University Club Website catering to the needs of interested students, members, and administrators. The website will facilitate joining the club, streamline member management, and offer a secure and user-friendly environment. It includes but is not limited to:

- User registration and authentication
- Event browsing and registration
- View workshops
- Contest browsing and registration
- Rate blogs (optional)
- Create and browse blogs
- Payment for member registration
- Admin dashboard for system monitoring and management
- Gallery
- About us page
- Newsletter

Out of scope:

- Mobile application development (unless specified otherwise)
- Integration with third-party services not mentioned in this document

## 1.5 References

The development and implementation of the University Club Website will adhere to industry best practices and standards. The following references will serve as the basis for this project:

- **PCI DSS (Payment Card Industry Data Security Standard):** Compliance with PCI DSS for secure handling of payment card data during payment gateway integration.
- **HTTP/HTTPS Protocol Standards:** Adherence to HTTP/HTTPS protocols for secure data transmission over the web.

- **SQL Database Standards:** Utilization of SQL database standards for efficient and secure data storage and retrieval.
- **OWASP (Open Web Application Security Project) Guidelines:** Following OWASP best practices to ensure the platform's security against common web application vulnerabilities.
- **W3C Web Content Accessibility Guidelines (WCAG):** Compliance with WCAG guidelines to ensure the platform is accessible to users with disabilities.
- **Relevant Industry Journals and Publications:** Continuous monitoring of industry journals, publications, and online resources for emerging trends and best practices in e-commerce

# 2. Overall Description

## 2.1 Product Perspective

The Programming Club Website is a vital component of our university's programming club, designed to enhance the club's operations, communication, and engagement with its members and the wider community. This website is intended to serve as a dynamic online platform that facilitates club management, collaboration, and information sharing.

The platform interacts with external systems and services, including but not limited to:

- Payment gateways for secure payment processing.
- Product suppliers for inventory management.
- Shipment and logistics services for order fulfillment.
- User authentication and authorization services for secure access control.

## 2.2 Product Features

The programming club website encompasses a rich set of features tailored to meet the diverse needs of its user classes. Key features include:

- **User Registration and Profiles:** Allow users to create accounts and customize their profiles.
- **Event Management:** Create and manage club events, workshops, seminars, and meetings.
- **Membership Management:** Enable users to join the club, renew memberships, and pay dues (if applicable).
- **Blog and News Section:** Publish club updates, articles, and programming-related news.
- **Newsletter Subscription:** Enable users to subscribe to club newsletters for updates and announcements.
- **Hackathon and Competition Hub:** Feature information on upcoming hackathons, coding competitions, and participation guidelines. Allow members to form teams and register for events.

- **Gallery and Media Library:** Showcase photos, videos, and presentations from past events and activities.
- **Security and Privacy Features:** Implement security measures to protect user data and privacy.

## 2.3 User Classes & Characteristics

The programming club website caters to three distinct user classes, each with specific characteristics:

- **Members:**
    - Current students, faculty, or staff of the university.
    - Varied levels of programming experience, from beginners to experts.
    - Interested in attending club events, collaborating on projects, and accessing resources.
- **Committee:**
    - Current club officers, administrators, and moderators.
    - Proficient in programming and club management.
    - Responsible for organizing events, managing memberships, and maintaining website content.
- **Admins:**
    - Web developers, IT staff, or administrators responsible for website maintenance.
    - Proficient in web development and server administration.
    - Ensure the technical functionality, security, and performance of the website.

## 2.4 Operating Environment

The operating environment for a programming club website encompasses the technical infrastructure required for its functionality. This includes web servers, databases, programming languages, content management systems, frontend technologies, security measures, and more. It also involves considerations such as scalability, cross-browser compatibility, and compliance with legal regulations. By carefully configuring and maintaining these components, the website can deliver a secure, responsive, and reliable user experience while supporting the club's activities and community engagement.

- **Web Application Framework:** The platform will be built using the popular Node.js and Express.js framework for efficient and scalable backend development.
- **Programming Languages:** Backend development will be primarily done in JavaScript, and the frontend will be built using HTML5, CSS3, and React.js for a responsive and interactive user interface.
- **Database Management System (DBMS):** PostgreSQL, a powerful open-source relational database, will be used for efficient data storage and retrieval.
- **Cloud Infrastructure:** The platform will be hosted on Amazon Web Services (AWS) to leverage its scalability, reliability, and wide range of services, such as AWS Elastic Beanstalk for application deployment.
- **Payment Gateway Integration:** Secure payment processing will be achieved through integration with Stripe, a widely used payment gateway known for its developer-friendly APIs and strong security features.
- **Security Measures:** SSL/TLS encryption will be implemented for secure data transmission, and a Web Application Firewall (WAF) like AWS WAF will be used for protection against common web application attacks.
- **Content Delivery Network (CDN):** Cloudflare, a globally distributed CDN, will be employed to optimize content delivery, reduce latency, and enhance overall performance.
- **Server Operating System:** Servers will run on Ubuntu Server, a stable and secure Linux distribution, ensuring reliable system operation.
- **Version Control System:** Git will be used for version control and code management, enabling collaborative development and code tracking.
- **Continuous Integration/Continuous Deployment (CI/CD):** The CI/CD pipeline will be established using Jenkins to automate testing and deployment processes, ensuring rapid and reliable software releases.
- **Monitoring and Analytics:** Prometheus and Grafana will be used for monitoring system performance, while Google Analytics will provide insights into user behavior for system optimization.

## 2.5 Design & Implementation Constraints

The development of the programming club website is subject to several design and implementation constraints, which include:

- **Cross-Browser Compatibility:** The platform must be compatible with major web browsers, including but not limited to Chrome, Firefox, Safari, and Edge, to ensure a consistent user experience.
- **Responsive Design:** The user interface must be responsive, adapting seamlessly to various screen sizes and devices, including smartphones, tablets, and desktops.
- **Performance Optimization:** Efforts will be made to optimize platform performance, reducing page load times and ensuring smooth navigation, even during high traffic periods.
- **Security Compliance:** The platform must adhere to industry-standard security practices, including encryption of sensitive data, protection against SQL injection, and user authentication best practices.
- **Regulatory Compliance:** Compliance with applicable laws and regulations, such as GDPR for data protection and PCI DSS for payment card data, is a mandatory constraint.
- **Scalability:** The system architecture should be designed to allow for scalability to accommodate increasing numbers of users, products, and transactions.
- **Budget Constraints:** The project must operate within the allocated budget, and any additional expenses or scope changes will be evaluated and approved accordingly.
- **Resource Limitations:** Availability of human and technical resources may impact project timelines and development capabilities.

## 2.6 User Documentation

User documentation for a programming club website is essential to help users, including club members, administrators, and other stakeholders, navigate and utilize the website effectively.

- **User Manuals:** We'll provide user manuals for different user groups, covering registration, navigation, and relevant tasks to ensure a smooth experience on our website.

- **Video Tutorials:** Video tutorials will be created to visually guide users through common actions, enhancing the learning experience.
- **Frequently Asked Questions (FAQs):** A set of FAQs will be available to address common queries and issues, ensuring users can find quick solutions to their problems.
- **Online Help Center:** An online help center with a searchable knowledge base will be accessible, offering in-depth guidance on platform features and troubleshooting.
- **Contextual Tooltips:** Context-sensitive tooltips and hints will be incorporated within the user interface to provide on-demand assistance.

## 2.7 Assumptions & Dependencies

Several assumptions and dependencies are inherent to the development and deployment of the programming club website, including:

- **Third-Party Services:** Assumption that third-party services, such as payment gateways and CDN providers, will be available and compatible with the platform.
- **Internet Connectivity:** Dependency on users having access to a stable internet connection for platform usage.
- **Data Integrity:** Assumption that data provided by buyers, merchants, and administrators will be accurate and complete.
- **Compliance with Regulations:** Dependency on adherence to legal and regulatory requirements, such as tax laws and privacy regulations.
- **Project Timeline:** Assumption that the project will be executed within the specified timeline, accounting for any potential delays or changes in scope.
- **Resource Availability:** Dependency on the availability of development and testing resources as per project planning.

# 3. System Features

## 3.1 User Features

### 3.1.1 User Registration and Authentication

**Description and Priority:** Users can create accounts and log in to access the programming club's features. Priority: High.

**Functional Requirements:**

- **FR-01:** Users can register by providing a unique username, valid email, and password.
- **FR-02:** User authentication is required during login to verify the username and password.

### 3.1.2 Event Management

**Description and Priority:** Users can view, register for, and participate in programming club events, workshops, and meetings. Priority: High.

**Functional Requirements:**

- **FR-03:** Users can search for events by category, keywords, or filters.
- **FR-04:** Users can view detailed event listings, including descriptions, dates, times, and registration options.
- **FR-05:** Users can register for events and receive event-related notifications.

### 3.1.3 Membership Management

**Description and Priority:** Users can join or renew programming club memberships and manage their membership status. Priority: Medium.

**Functional Requirements:**

- **FR-06:** Users can join the club by following the membership registration process.
- **FR-07:** Users can renew their club memberships.

- **FR-08:** Users can view and manage their membership status and history.

### 3.1.4 Project Collaboration

**Description and Priority:** Users can create and collaborate on programming projects within the club. Priority: High.

**Functional Requirements:**

- **FR-09:** Users can create new programming projects and invite other members to join.
- **FR-10:** Users can collaborate on projects by assigning tasks, sharing code, and tracking progress.
- **FR-11:** Users can utilize version control and code sharing features.

### 3.1.5 Blog and News Section

**Description and Priority:** Users can read and contribute programming-related articles and stay updated on club news. Priority: High.

**Functional Requirements:**

- **FR-12:** Users can access blog articles and news posts by category or date.
- **FR-13:** Users can submit guest posts and articles for publication.
- **FR-13:** Users can comment on blog posts and engage in discussions.

## 3.2 Committee Features

### 3.2.1 Content Management

**Description and Priority:** Committee members can oversee and moderate product listings and user-generated content. Priority: Medium.

**Functional Requirements:**

- **FR-14:** Committee members can review and moderate product listings, removing or flagging inappropriate content.
- **FR-15:** Committee members can review and moderate user-generated content, such as forum posts, project contributions, and comments.

### 3.2.2 Event Management

**Description and Priority:** Committee members can create, edit, and manage programming club events. Priority: High.

**Functional Requirements:**

- **FR-16:** Committee members can add new events, including event details, descriptions, dates, and locations.
- **FR-17:** Committee members can edit existing event information and update event statuses.
- **FR-18:** Committee members can delete events from the calendar.

### 3.2.3 Email Communication

**Description and Priority:** Committee members can send club-wide announcements, newsletters, and event notifications to club members. Priority: Medium.

**Functional Requirements:**

- **FR-19:** Committee members can compose and send email communications to club members.

- **FR-20:** Committee members can manage email templates and recipient lists.

## 3.3 Admin Features

### 3.3.1 Admin Dashboard

**Description and Priority:** Admins have access to a comprehensive administrative dashboard for system monitoring and management. Priority: High.

**Functional Requirements:**

- **FR-21:** Admins can log in and access the admin dashboard.
- **FR-22:** The admin dashboard provides options for system monitoring and management.

### 3.3.2 User Management

**Description and Priority:** Administrators can manage user accounts, assign roles, and control access permissions. Priority: High.

**Functional Requirements:**

- **FR-23:** Administrators can view user account information, including roles and permissions.
- **FR-24:** Administrators can modify user account details, reset passwords, or deactivate accounts as needed.

### 3.3.3 Security Management

**Description and Priority:** Administrators can monitor and enforce security measures, including data encryption and access control. Priority: High.

**Functional Requirements:**

- **FR-25:** Administrators can configure and monitor security settings, including data encryption and access control measures.
- **FR-26:** Administrators can access security logs to review and respond to security-related events.

### 3.3.4 Analytics and Reporting

**Description and Priority:** Administrators can access analytics and generate reports for club activities and member engagement. Priority: Medium.

**Functional Requirements:**

- **FR-27:** Administrators can view analytics and performance reports, including event attendance, resource usage, and user engagement.
- **FR-28:** Administrators can generate customized reports based on specific criteria and timeframes.

# 4. External Interface Requirements

## 4.1 User Interface

The user interface (UI) of our programming club website is designed to be intuitive, user-friendly, and responsive, catering to the needs of club members, committee members, and administrators. Key aspects of the UI include:

- **Member Interface:** Club members interact with the platform through a web-based UI accessible from desktop and mobile devices. The UI includes features for event registration, project collaboration, and resource access.
- **Committee Interface:** Committee members access their management tools via a separate UI tailored for event management, membership oversight, and content moderation.
- **Admin Interface:** Administrators utilize an administrative dashboard with tools for system monitoring, user management, content moderation, and security configuration.

## 4.2 System Interface

The programming club website interacts with various software components and external services to provide its functionality. These software interfaces include:

- **Database Management System (DBMS):** The platform communicates with the selected DBMS (e.g., PostgreSQL) for data storage and retrieval, ensuring efficient and secure access to product information, user data, and order records.
- **Payment Gateway APIs:** Integration with payment gateway APIs (e.g., Stripe API) enables secure payment processing during checkout, allowing users to make online payments for their orders.
- **Third-Party Services:** The platform may interface with third-party services for additional functionality, such as content delivery networks (CDNs) for optimized content delivery and analytics services for performance tracking.

## 4.3 Hardware Interface

The Programming Club Website is designed to operate on standard hardware configurations. Hardware interface requirements include:

- **Server Infrastructure:** The platform operates on cloud-based servers (e.g., Amazon Web Services) equipped with suitable hardware resources, including CPU, RAM, and storage, to support the application's scalability and performance needs for hosting the club's website, databases, and associated services.
- **Client Devices:** Club members, committee members, and administrators can access the platform via standard web browsers on desktop and mobile devices. The user interface (UI) is optimized to be compatible with a wide range of hardware configurations and screen sizes, ensuring accessibility and usability across various devices.

## 4.4 Communication Interface

Efficient communication between the platform and external systems is essential. Communication interfaces encompass:

- **HTTP/HTTPS Protocols:** The platform uses HTTP/HTTPS protocols for secure data transmission between clients and servers, ensuring the confidentiality and integrity of user data during interactions.
- **API Integration:** The platform may integrate with external APIs, such as payment gateway APIs and shipping APIs, to enable seamless communication and data exchange with third-party services.
- **Email Notifications:** Email communication with users, including order confirmations, password resets, and notifications, relies on SMTP (Simple Mail Transfer Protocol) for reliable email delivery.

# 5. Non-Functional Requirements

## 5.1 Security

Ensuring data and system security is paramount to protect user information and prevent unauthorized access.

- **Data Security:** The platform shall implement robust data security measures, including encryption of sensitive user data, to ensure the confidentiality and integrity of information.
- **Authentication and Authorization:** User authentication shall be enforced for all user types (club members, committee members, admins), with role-based access control (RBAC) to protect against unauthorized access to sensitive club resources and administrative functions.
- **Payment Security:** Payment processing shall comply with industry standards such as PCI DSS (Payment Card Industry Data Security Standard) to safeguard payment card information during transactions.

## 5.2 Performance

Efficient system performance is essential for providing a seamless user experience.

- **Response Time:** The platform shall maintain a responsive user interface with an average page load time of under 2 seconds to enhance user satisfaction.
- **Scalability:** The system shall be designed to handle increased traffic during peak periods, ensuring consistent performance and response times.

## 5.3 Usability

An intuitive user interface promotes user satisfaction and ease of use.

- **Intuitive User Interface:** The user interface (UI) shall be meticulously designed with a strong emphasis on usability, ensuring an intuitive experience for club members, committee members, and administrators. The UI shall feature clear and logical navigation, straightforward labels, and user-friendly interactions. This will facilitate seamless engagement and user satisfaction throughout the website, fostering a positive user experience.

## 5.4 Reliability

Reliable system operation ensures minimal disruptions and error handling.

- **System Uptime:** The platform shall aim for 99.9% system uptime, with scheduled maintenance communicated in advance to minimize disruption.
- **Error Handling:** Robust error handling and logging mechanisms shall be implemented to identify, report, and resolve issues promptly.

## 5.5 Scalability

Scalability is essential to support growth and accommodate increasing membership and activities within the programming club.

- **Horizontal Scaling:** The system shall be designed to support horizontal scaling, allowing it to seamlessly accommodate a growing number of club members, events, and collaborative projects without compromising performance. This ensures that the website remains responsive and efficient as the programming club community expands.

## 5.6 Supportability

Supportability measures, including documentation, enhance user and administrator support.

- **Documentation:** Comprehensive user and technical documentation shall be provided to support users and system administrators.

# 6. Other Requirements

This section covers miscellaneous requirements that do not fall directly into the categories of functional, non-functional, or external interface requirements. These may include:

- **Legal Requirements:** Document any legal or compliance requirements specific to the E-commerce Platform, such as privacy regulations (e.g., GDPR) and tax laws.
- **Third-Party Integration:** Specify any third-party integrations or services required for the platform to function effectively, such as external payment gateways, shipping providers, or analytics tools.
- **Localization and Internationalization:** Outline requirements for supporting multiple languages, currencies, and regional variations, if applicable.
- **Data Backup and Recovery:** Describe data backup and recovery procedures to safeguard against data loss and system downtime.
- **Customization and Branding:** If the platform allows for customization by merchants, define requirements related to branding, templates, and design options.
- **User Training:** If user training is necessary, detail the scope and delivery method for training materials and resources.
- **Performance Metrics:** Specify key performance indicators (KPIs) to be monitored regularly to assess system performance and user satisfaction.
- **Reporting and Analytics:** Describe requirements for generating and accessing reports and analytics data for business insights.
- **System Maintenance and Updates:** Specify how system maintenance, bug fixes, and updates will be handled, including maintenance windows and version control.
- **Backup and Disaster Recovery:** Outline procedures and requirements for data backup, disaster recovery, and business continuity planning.
- **Change Management:** Describe the process for handling change requests, including evaluation, approval, and implementation.
- **Performance Testing:** Specify any performance testing requirements, including load testing and stress testing.
- **Audit Trails and Logs:** Describe requirements for maintaining audit trails and logs for security and compliance purposes.