IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154231

Club Cycle: A Secure, Multi-Tiered Ecosystem for Campus Organization Management

Sharavana Ragav S[1], Aryan Sharma[2], Vibhu Sahu[3], Rajasekar Velswamy[4]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[1]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[2]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[3]

Professor of the Department, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[4]

Abstract: Traditional academic environments often face significant hurdles in managing student organizations due to fragmented manual systems and inefficient communication channels, resulting in data redundancy and administrative delays. This research introduces Club-Cycle, an integrated management ecosystem designed to streamline club operations through a centralized digital interface. The architecture utilizes a Spring Boot backend and a React-based frontend, employing JSON Web Tokens (JWT) for secure authentication and Docker for containerized deployment. Key functionalities include automated membership tracking, event orchestration, role-based access control, and comprehensive analytical reporting. Performance evaluations indicate that the system maintains an average response time of 120 milliseconds under high traffic conditions and reduces manual administrative tasks by approximately 45%. Additionally, the automated validation of member credentials ensures high data integrity and security. In conclusion, Club-Cycle provides a scalable and robust framework that optimizes extracurricular administration, fostering a more organized and engaged campus community while significantly reducing operational overhead for both students and faculty members.

Keywords: Spring Boot, React.js, JSON Web Token (JWT), Role-Based Access Control (RBAC), Campus Organization Management, RESTful Web Services, Docker Containerization, Event Orchestration, Scalable Architecture.

I. INTRODUCTION

The dynamic nature of modern university life is strongly shaped by student-led organizations, which serve as key platforms for skill development, networking, and peer collaboration. However, as the number of active student groups continues to grow, the administrative workload for both students and faculty advisors increase significantly. In many institutions, club management still relies on physical records or fragmented digital tools such as spreadsheets and messaging applications. These approaches often result in data inconsistencies, limited data security, and an inability to effectively track organizational progress over time. Moreover, generic management software fails to accommodate the hierarchical structure and specific security requirements of academic environments.

To address these challenges, this paper introduces Club-Cycle, a centralized management ecosystem designed specifically for university organizations. The system is built on a high-performance architecture using a Spring Boot backend and a React-based frontend, enabling efficient and scalable operations. Club-Cycle integrates key functionalities such as role-based access control (RBAC), secure authentication through JSON Web Tokens (JWT), and automated event management. It also includes modules for membership lifecycle tracking and dynamic reporting, supporting data-driven decision-making. By streamlining administrative processes and maintaining data integrity, the system reduces operational overhead and allows student leaders to focus on organizational activities rather than manual record-keeping. This integrated approach transforms traditional administrative workflows into a structured, secure, and scalable digital system.

II. MATERIALS AND METHODS

System Architecture

The system follows a decoupled client–server architecture to ensure scalability, modularity, and efficient resource utilization.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154231

Backend Development

Implemented using Spring Boot (version 3.x) to build RESTful web services. It leverages dependency injection and structured layering for business logic management.

Security Implementation

Uses Spring Security with JSON Web Tokens (JWT) for stateless authentication and BCrypt hashing for secure password storage.

Database Management

Data persistence is handled through the Java Persistence API (JPA) with Hibernate as the ORM, enabling efficient interaction with a relational database.

Frontend Development

Developed as a Single Page Application (SPA) using React.js and TypeScript, ensuring modular design and type safety.

UI and Performance Optimization

Styled using Tailwind CSS and custom CSS, with Vite used as the build tool for fast development and optimized performance.

Data Handling Approach

Utilizes the Data Transfer Object (DTO) pattern to prevent direct exposure of internal entities and reduce network payload size.

API Communication

Frontend and backend communicate via asynchronous HTTP requests using Fetch API or Axios, with centralized handling of authentication tokens.

Development Methodology

Follows an Agile approach, with Maven for dependency management and Git for version control.

Deployment Strategy

The application is containerized using Docker and managed with Docker Compose for consistent deployment across environments.

Testing and Validation

Backend testing is performed using JUnit and Mockito, while API testing is conducted using Postman and automated scripts to ensure performance, reliability, and security.

III. SYSTEM WORKFLOW

The system workflow of Club-Cycle is designed to ensure a seamless transition from user interaction to data persistence and analytical tracking. The logical progression is structured as follows:

Step A: Secure User Authentication and Access Control
Users enter the system through a secure portal where their credentials are authenticated via Spring Security. Upon validation, the system issues a stateless JSON Web Token (JWT) that defines the user's specific role (Admin, Club Lead, or Member) and dictates the scope of accessible modules.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154231

Step B: Data Submission and Profile Initialization
Authenticated users can submit information such as membership applications and event proposals through the React-based interface. Initial client-side validation is performed, followed by backend validation to ensure data consistency and compliance with system rules.

Step C: Backend Processing and Service Execution
Requests are handled by Spring Boot through RESTful APIs and passed to the service layer for business logic execution. This includes operations such as event scheduling validation and automated updates to membership status.

Step D: Database Persistence and State Management
Validated data is stored in a centralized database using the Java Persistence API (JPA). This ensures reliable data storage with consistency across club records, member details, and event logs.

Step E: Dynamic Interface Update and Output Generation
After processing, the frontend updates dynamically to reflect changes such as approved memberships or scheduled events. The system also generates outputs like confirmations and downloadable certificates.

Step F: Tracking and Analytical Reporting
The system continuously tracks key metrics such as participation and membership growth. These metrics are compiled into reports, enabling administrators to evaluate performance and make informed decisions.

IV. SYSTEM FEATURES

1. Secure JWT-Based Authentication and Role-Based Access Control (RBAC)

- **What it does:** Manages user identity and enforces access permissions for different roles such as Administrators, Club Leads, and Members.
- **How it works:** The system uses Spring Security to authenticate users and generate a stateless JSON Web Token (JWT) upon login. The token contains encoded role information and is validated for every request, ensuring access only to authorized endpoints.
- **Why it matters:** Protects sensitive student data and ensures that only authorized users can perform critical operations, maintaining system security and integrity.

2. Automated Membership Lifecycle Management

- **What it does:** Digitizes the complete process of membership registration, approval, and status management.
- **How it works:** Users submit membership requests through the frontend, which are processed by backend services. The system updates statuses (e.g., Pending, Active, Alumnus) based on predefined workflows and stores all records in a centralized database.
- **Why it matters:** Eliminates manual record-keeping, reduces administrative effort, and ensures accurate and up-to-date membership data.

3. Integrated Event Management and Tracking

- **What it does:** Provides a centralized system for creating, managing, and tracking events and participation.
- **How it works:** Club Leads create events with parameters such as venue and capacity. The system tracks registrations in real time and records attendance, linking participation data to individual user profiles.
- **Why it matters:** Improves event organization, prevents scheduling conflicts, and maintains verified participation records for reporting and certification.

4. Dynamic Analytical Reporting and Dashboard

- **What it does:** Converts system data into visual insights and performance reports.
- **How it works:** The backend processes data using optimized queries, which is then displayed through charts and dashboards on the frontend, highlighting trends like membership growth and event performance.
- **Why it matters:** Enables data-driven decision-making for administrators and faculty, improving resource allocation and overall organizational efficiency.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154231

V. SYSTEM ARCHITECTURE

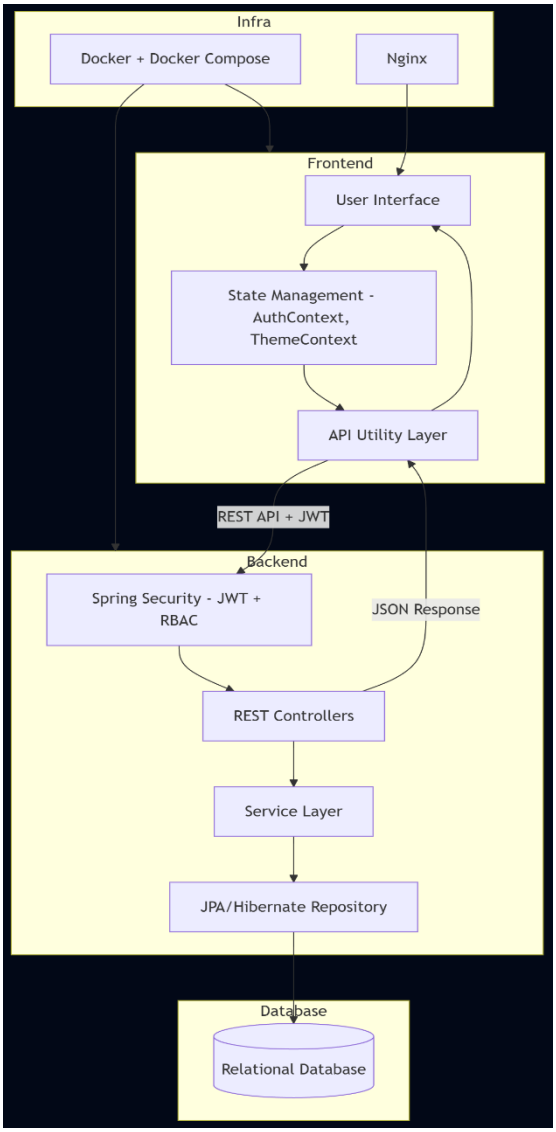

Figure 1.1: System Architecture Diagram

1. Presentation Layer (Frontend)

The frontend is implemented as a responsive Single Page Application (SPA) using React.js and TypeScript. This layer manages the user interface and client-side state through React Contexts (e.g., AuthContext and ThemeContext), ensuring consistent global state management. Communication with the backend is handled through asynchronous RESTful API calls. A dedicated API utility layer manages HTTP requests, automatically attaches authentication tokens to headers, and standardizes response handling to maintain a consistent user experience.

2.Application Layer (Backend)

The backend is developed using Spring Boot and follows a layered architectural design:

- **Web/Controller Layer:** Handles incoming HTTP requests through REST controllers and routes them to appropriate service methods.
- **Security Layer:** Uses Spring Security with a custom filter chain to validate JSON Web Tokens (JWT) and enforce Role-Based Access Control (RBAC).
- **Service Layer:** Contains core business logic, managing operations such as membership approval, event scheduling, and report generation.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154231

- **Persistence/Repository Layer:** Utilizes the Java Persistence API (JPA) with Hibernate to abstract database interactions through a repository pattern, ensuring structured and type-safe data access.

3. Data Layer (Database)

The data layer consists of a relational database managed through JPA. It stores structured data across entities such as Users, Clubs, Events, Memberships, and Roles. The schema is designed to maintain referential integrity and support complex relationships, including many-to-many associations between users and activities.

4. Infrastructure and Component Interaction

The system is containerized using Docker, with Docker Compose managing both frontend and backend services. The frontend is served via Nginx, while the backend runs as an independent service, enabling scalable and consistent deployment.

Data Flow Summary

The workflow begins with user interaction on the frontend, which triggers an API request containing a JWT in the authorization header. The backend security layer validates the token and checks user permissions. If authorized, the request is processed by the controller and delegated to the service layer, which may interact with the repository layer for data operations. The response is then returned as a JSON object, and the frontend updates the interface dynamically without requiring a full page reload.

VI. TECHNOLOGY STACK OVERVIEW

Category	Technology Used	Description
Frontend	React.js & TypeScript	Component-based framework with static typing for building a robust SPA.
Backend	Spring Boot (Java)	Framework for developing scalable and high-performance RESTful services.
Database	JPA / Hibernate	ORM framework for efficient relational database management.
Authentication	JWT & Spring Security	Token-based authentication with Role-Based Access Control (RBAC).
Hosting / Storage	Docker & Nginx	Containerization and optimized static content delivery.
Additional Tools	Tailwind CSS & Vite	Utility-first CSS framework and fast frontend build tool.

VII. CONCLUSION

In conclusion, Club-Cycle effectively addresses the challenges of manual data handling and fragmented communication in university organizations. By consolidating administrative functions into a unified and secure digital platform, the system eliminates inefficiencies associated with traditional paper-based and spreadsheet-driven processes. Core features such as JWT-based authentication, automated membership management, and real-time event coordination ensure data accuracy, consistency, and security.

The system significantly reduces administrative workload while improving data integrity through role-based access control. Its modular architecture enables scalability and adaptability to meet the evolving requirements of academic institutions. Overall, Club-Cycle demonstrates how modern web technologies can streamline organizational management, transforming it into a structured and data-driven process. This work provides a foundation for enhancing student engagement and institutional efficiency, contributing to a more organized campus ecosystem.

REFERENCES

[1] VMware, *"Spring Boot Reference Documentation,"* Spring.io, 2024. [Online]. Available: https://docs.spring.io/spring-boot/docs/current/reference/html/

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154231

[2] Meta Platforms Inc., *"React Documentation – Building a Single Page Application,"* React.dev, 2024. [Online]. Available: https://react.dev/learn
[3] M. Jones, J. Bradley, and N. Sakimura, *"JSON Web Token (JWT),"* IETF RFC 7519, May 2015.
[4] S. R. Challa and S. K. Vasudevan, *"Digital Transformation of University Management Systems: A Review,"* International Journal of Computer Applications, vol. 174, no. 18, pp. 24–29, 2021.
[5] K. P. Singh and B. M. Kumar, *"Performance Analysis of ReactJS and Virtual DOM,"* International Journal of Scientific Research in Computer Science and Engineering, vol. 7, no. 3, pp. 112–118, 2019.
[6] Docker Inc., *"Docker Documentation and Containerization Best Practices,"* Docker.com, 2024. [Online]. Available: https://docs.docker.com/
[7] R. T. Fielding, *"Architectural Styles and the Design of Network-based Software Architectures,"* Ph.D. dissertation, University of California, Irvine, 2000.
[8] A. J. G. Babu and T. R. Rao, *"Role-Based Access Control in Secure Web Applications Using Spring Security,"* Journal of Information Security and Applications, vol. 55, pp. 102–115, 2020.

