Software Requirements Specification

for

P04-09

Version 1.0 approved

Prepared by

- **Benjamin Hatfield**

- **Okasha Varoo**

- **Jannik Ernst**

- **Azman Istiaq Asfi**

- **Nghi Le Hoang Vinh**

- **Abdulahi abdulahi**

RMIT Computing Technologies Students

14-08-2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the functional and non-functional requirements for EventHub, a web-based Campus Event Management System. This document covers version 1.0 of EventHub, which will be the initial minimum viable product release for students, club organisers and administrators at the university.

The scope of this SRS includes the core functionality required for event creation, RSVP's, attendance tracking and administrative moderation. It describes the system as a whole, including all user roles (student, organiser and administrator). This SRS document does not include any details about potential future iterations for the EventHub service.

1.2 Document Conventions

- Bold text is used to highlight important terms, role names, or section headings.
- Italics are used for feature names or user stories.
- All functional requirements are uniquely identified using the format FR-XX.
- All non-functional requirements are uniquely identified using the format NFR-XX. Acceptance criteria are expressed in Gherkin format ("Given… When… Then…") for testability.
- The priority of requirements is explicitly stated as High, Medium, or Low for each requirement, and priorities are not inherited automatically from higher-level requirements.

1.3 Intended Audience and Reading Suggestions

- **Developers:** To understand the detailed functional and non-functional requirements to be implemented.
- **Project managers:** To track project scope, priorities and deliverables.
- **Testers/Quality Assurance Engineers:** To design and execute testing based on defined acceptance criteria.
- **UI/UX Designers:** To design interfaces aligned with the functional flows and usability goals of the system.
- **Stakeholders (Students, Organisers, Administrators):** To validate that the system design adequately captures each stakeholders needs.
- **Technical Writers:** To produce user manuals, help pages as well as development documentation.

1.4 Product Scope

EventHub is a web-based platform designed for university clubs and student organisations to create, publish and manage events. Students can browse upcoming events, RSVP, and receive reminders. Event organisers can create and manage their group/clubs events as well as track attendance and gather feedback for these events. Administrators can moderate event content and manage users to ensure compliance with university guidelines.

The Primary benefits of EventHub include:
- Simplify event creation, publication, and promotion for university groups and clubs.
- Improve event discovery and participation for university students.
- Enable effective moderation and oversight of event content for administrators.
- Provide accurate, real-time information about events, venues and availability university wide.

1.5 References

- IEEE Std 830-1998 — IEEE Recommended Practice for Software Requirements Specifications.
- RMIT University IT Security Guidelines (2023).
- Australian Privacy Principles (APPs), Privacy Act 1988.
- Google Maps Platform Documentation, Google Developers, 2025.
- Spring Boot Reference Documentation, VMware, version 3.x.
- MySQL 8.0 Reference Manual, Oracle Corporation, 2025.
- PostgreSQL 14 Documentation, PostgreSQL Global Development Group, 2025.

2. Overall Description

2.1 Product Perspective

EventHub is a new, self-contained web-based application developed for RMIT University's student community. It is not a follow-on from any existing product but is designed to replace the current fragmented event promotion methods (e.g., email lists, noticeboards, social media posts) with a centralised and accessible platform.

EventHub will operate as a standalone system with its own database and authentication system, but will integrate with selected external services, including:
- Google Maps Static API for displaying event locations.
- Email notification service for RSVP confirmations and reminders.

2.2 Product Functions

At a high level, EventHub will provide the following core functions:

- **Event Management**
  - Create, edit, delete events (organisers).
  - View, edit, delete any event (administrators).
  - List upcoming events with filters and search.
- **User Participation**
  - RSVP to events and view RSVP list.
  - Submit feedback (rating & comments).
  - Share event links.

- **Event Discovery**
  - Search/filter events by category, keyword, or date.
  - View event details with location preview.
  - Receive event recommendations based on interests.
- **Administrative Tools**
  - Manage all users (activate/deactivate).
  - Moderate and remove inappropriate events.
- **Enhancements (Extension Features)**
  - Calendar view.
  - QR code check-in.
  - CSV export of attendance lists.
  - Badge/achievement system.

2.3 User Classes and Characteristics

User Class	Description	Privileges	Technical Expertise	Importance
Student	Any enrolled student browsing or attending events.	RSVP to events, view event details, submit feedback.	Basic web browsing skills.	High
Event Organiser	Club or society representative creating and managing events.	Create, edit, delete own events, upload event galleries, view RSVPs.	Moderate familiarity with event tools.	High
Administrator	University staff or student services responsible for oversight.	View, edit, delete any event, manage users, moderate content.	Comfortable with admin dashboards.	High
Guest User	Public user without login.	Browse and search events (read-only).	Basic web skills.	Medium

2.4 Operating Environment

- **Client-Side:**
  - Modern web browsers (Google Chrome, Firefox, Safari, Microsoft Edge), latest two versions.
  - Responsive design for desktop, tablet, and mobile screens.
- **Server-Side:**
  - Operating System: Ubuntu Server 22.04 LTS within docker container.
  - Spring Boot web framework
  - Database: MySQL 8.0 or PostgreSQL 14+.
  - Hosting: University server or cloud platform (AWS/GCP).
- **External Dependencies:**
  - Google Maps Static API.
  - SMTP/Email service for notifications.

2.5 Design and Implementation Constraints

- Must comply with **RMIT University IT Security Guidelines**.
- Role-based access control must be implemented for Students, Organisers, and Administrators.
- Must use HTTPS for all communications.
- Must store passwords securely using industry-standard hashing.
- Must be compatible with the university's SSO system in future phases.
- Deployment must be done within a free basis for the purpose of the project presentation.

2.6 User Documentation

The following documentation will be delivered with the product:

- User Manual (PDF + web version) for Students and Organisers, explaining event search, RSVP, and creation workflows.
- Administrator Manual detailing moderation tools and user management.
- Online Help Centre with searchable FAQ and troubleshooting steps.
- Quick Start Guide for first-time users.

2.7 Assumptions and Dependencies

Assumptions:
- Users have stable internet access when using EventHub.
- Organisers and administrators have valid university accounts.
- The university will provide email server access for notifications.

Dependencies:
- Continued availability of Google Maps Static API for location previews.
- Functionality of the SMTP/Email service for sending RSVP confirmations and reminders.

3. External Interface Requirements

3.1 User Interfaces

EventHub will be a **web-based system** that can be accessed via any modern web browser. The design will be clean, minimalist and consistent across all pages.
- Every page will have a header and footer.
- The header will include **Home, Events, Create Event, My Events, Profile/Login.**
- The footer will include the **Help, About and Contact** links.
- Buttons will have a consistent design and color throughout(primary = Create, Save; secondary = Cancel, Back)
- Error messages will directly be shown under the form fields and will be displayed in the color red.
- The interface will adapt according to the device size (responsive design).
- Key pages:
  - **Home Page** (shows overview of the website and upcoming events)
  - **Event Details Page** (title, description, time, location with map, RSPV, gallery, reviews).

- **Create/Edit event** (for organisers).
- **My Events Page** (RSVP'd events for students, created events for organisers and management)
- **Admin Dashboard** (moderation and user management).

3.2 Hardware Interfaces

EventHub will not require any special hardware. It will be able to run smoothly across common client and server environments.

- **Client Side:**
  - Supported on modern web browsers (Google Chrome, Firefox, Safari, Microsoft Edge), latest two versions.
  - Responsive design ensures that it runs on any compatible desktops, laptops, tablets, and mobile devices.
- **Server Side:**
  - Operating System: Ubuntu server 22.04 LTS running inside Docker container.
  - Backend: Spring Boot web framework.
  - Database: MySQL 8.0 or PostgreSQL 14+.
  - Hosting: University server or cloud platform (AWS/GCP)

3.3 Software Interfaces

EventHub will integrate with several software components to deliver its functionality. These includes backend framework, database systems, external APIs, and any supporting tools.

- **Operating Systems:**
  - **Server:** Ubuntu Server 22.04 LTS (within Docker)
  - **Client:** Windows, macOS, Linux, iOS, Android via web-browser access.
- **Frontend:**
  - HTML5, CSS, JavaScript
  - Responsive design for various sort of devices
- **Backend Framework:**
  - Spring Boot (Java 17+)
  - Maven build tool for dependency management
- **Database:**
  - My Sql 8.0
  - PostgreSQL 14+
- **External API/ Services:**
  - Google Maps Static API for displaying event location
  - Email Service for RSVP confirmation and reminders
- **Testing and CI/CD:**
  - JUnit 5 for unit and integration testing
  - GitHub Actions for automated testing and deployment
- **Data Exchange:**
  - Rest API calls using JSON for communication between frontend and backend

3.4 Communications Interfaces

EventHub will use standard web communication protocols to ensure a smooth interaction between clients, servers, and external services.

- **Protocols:**
  - All client to server communication will use **HTTPS** (secure HTTP).

    - TLS encryption will be enforced for protecting data between the web-application and the server
- **Notification and Messaging:**
    - **Email** notifications will be sent via SMTP.
- **Data Exchange:**
    - **Rest APIs** will be the main communication method where **JSON** is used for requests and responses.
    - APIs will follow standard REST conventions to fetch, create, update and remove.

4. Nonfunctional Requirements

4.1 Performance Requirements

- *The system must support at least 1,000 concurrent users without noticeable degradation in performance.*
- *Event search queries must return results within 2 seconds when the database contains up to 10,000 events.*
- *RSVP actions (adding or cancelling) must be processed and reflected in the user's "My Events" list within 1 second of submission.*
- *Profile updates must be saved and visible across the system within 3 seconds.*
- *The system must be able to handle peak load conditions, such as the start of semester when multiple events are published simultaneously, without exceeding 80% server CPU utilization for more than 10 minutes.*
- *All pages must load within 3 seconds on a standard campus internet connection (50 Mbps down / 10 Mbps up).*

4.2 Safety Requirements

- *The system must enforce role-based access control (RBAC) for Students, Organisers, and Administrators.*
- *All communication between client and server must use HTTPS with TLS 1.2 or higher.*
- *Passwords must be stored securely using bcrypt hashing with a minimum cost factor of 12.*
- *User sessions must expire after 30 minutes of inactivity and require re-authentication.*
- *Failed login attempts must be limited to 5 consecutive failures before temporarily locking the account for 15 minutes.*

- *The system must comply with RMIT University IT Security Guidelines, ensuring student and staff data is handled according to institutional privacy and data protection standards.*

- *Sensitive data (e.g., user emails, feedback, RSVP lists) must not be exposed through public APIs without authentication.*

- *Audit logging must be implemented to record all administrative actions (event deletions, account bans, role assignments).*

4.3 Security Requirements

EventHub must ensure the confidentiality, integrity, and availability of all user data (students, organisers, administrators) by enforcing strong security and privacy controls.

Authentication and Authorization

- *The system must implement role-based access control (RBAC) to distinguish privileges between Students, Organisers, and Administrators.*
- *User identity must be verified through a secure login process using unique credentials (username/email and password).*

- *Sessions must automatically expire after 30 minutes of inactivity, requiring re-authentication to continue.*

- *Only authenticated users with the correct roles may perform restricted actions (e.g., event creation, editing, deletion, user management).*

Data Protection

- *All client-to-server communication must use HTTPS with TLS 1.2 or higher to prevent interception of sensitive data.*

- Passwords must be stored securely using bcrypt hashing with a minimum cost factor of 12, ensuring they cannot be retrieved in plain text.

- Sensitive data (emails, RSVP lists, feedback) must only be accessible to authorised users and must not be exposed in public APIs.

- Data at rest in the database must be protected with appropriate access controls to prevent unauthorised queries.

System Security Controls

- *Failed login attempts must be limited to five consecutive failures before locking the account for 15 minutes to mitigate brute-force attacks.*
- *Audit logging must be enabled for all administrator-level actions (event deletions, account bans, role assignments) to provide accountability and traceability.*
- *The system must be designed to protect against common web application vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).*

Compliance and Standards

- *EventHub must comply with RMIT University IT Security Guidelines to align with institutional security policies.*
- *The system must adhere to Australian Privacy Principles (APPs) under the Privacy Act 1988, ensuring personal data is handled lawfully and securely.*
- *Any use of third-party APIs (e.g., Google Maps, Email services) must follow their respective security and privacy policies.*

4.4 Software Quality Attributes

Portability: EventHub must run without issues in supported browsers (Chrome, Firefox, Safari) across Windows, Linux, macOS, Android, iOS without making any major changes in the code.

Usability: New users should be able to go through the whole website without having help from any of the documentation. The user should be able to find the RSVP to an event in 3 or less clicks making it more intuitive.

Reusability: Core modules such as RSVP Service and Event Service must be designed to be reusable allowing us to reuse them in the future.

Correctness: All the requirements defined in the SRS document must be fully implemented when making the software. The functional requirements should be implemented as much accurate as possible to ensure that it behaves as intended.

Maintainability: The whole system should be maintainable without any major changes in the code and by not shutting down the system fully. New features should be added without any major modifications to the whole system. System defects must be fixable within 24 hours of identification.

Reliability: Event hub must achieve 99.2% uptime during peak event periods. The system must be able to handle critical failures and recover withing 10 minutes by itself. The system must be able to handle 15,000 users at once.

Efficiency: Pages must load within less than 2 seconds. It should be able to store up to 10,000 events in the database. The payment authentication for the event should be carried out in 3 seconds. The system should not exceed 90% of the server CPU utilization during peak periods.

4.5 Business Rules

User Role-Based Rules
Event Organizers

- Only authenticated users with organizer privileges can create new events
- Event organizers can only edit or delete events they have created
- Organizers can view and manage RSVPs for their own events only
- Organizers can upload photos to event galleries only after the event has concluded
- Organizers can export attendee lists only for events they have created

Students/General Users

- Any authenticated user can RSVP to events
- Users can only view their own RSVP history
- Users can only provide feedback for events they have RSVP'd to
- Users can share any public event regardless of RSVP status
- Users can receive personalized recommendations based on their activity history

Administrators
- Administrators can view and manage all events in the system
- Administrators have the authority to delete inappropriate or policy-violating events
- Administrators can deactivate or ban user accounts for misconduct
- Administrators can access all system data for monitoring and management purposes

Event Management Rules
Event Creation and Modification
- All events must have mandatory fields: title, description, date/time, location, and category
- Events can only be modified by their original creators (organizers)
- Events cannot be deleted if they have existing RSVPs (must be cancelled instead)
- Event dates cannot be set in the past during creation
- Event capacity limits (if set) cannot be exceeded by RSVPs

Event Participation
- Users cannot RSVP to the same event multiple times
- Users can withdraw their RSVP up until the event start time
- RSVPs are automatically closed when event capacity is reached
- Users cannot provide feedback for events that haven't occurred yet
- Event sharing is unrestricted for all public events

5. Other Requirements

Database Requirements:
- All event and user data must be stored in relational tables with foreign key constraints.
- Feedback and reviews must be tied to the event and the user ID.

Internationalization Requirements:
- The system must support English when launched.
- The UI must be designed in such a way to allow extension of additional languages.

Legal Requirements:
- All customer data must be private and confidential.
- All email must be able to have the unsubscribe option for notifications and promotions.

Reuse Objectives:
- EventHub components must be modular so that they can be used within any other university systems.

6. System Architecture

The goal of the project is to create a simple and maintainable platform, the simplicity required by the project will shape the architectural decisions made. Many decisions are mandated and thus choice is restricted in how some decisions are made.

6.1 Architecture Overview

6.2 Architectural Decisions

ADR-001: Use GitHub Projects for Project Management

Status: Accepted (Mandated)

Context: A software project management tool is needed to track progress, as well as plan and execute sprints throughout the development cycle. Although most team members have not yet used GitHub Projects in the past, it is mandated by the specifications of the project.

Decision: We will use GitHub Projects.

Consequences: As the GitHub Project is directly linked to code base repository, it has the benefit of avoiding additional coordination between a third-party software development management tool. As such, the ability to keep the Project board accurate and up to date should become easier.

ADR-002: Use Spring Boot as Java Web Framework

Status: Accepted (Mandated)

Context: A web framework for Java is needed in order to simplify the process of serving the web application. Spring Boot is the mandated framework and is what most team members have experience with.

Decision: We will use Spring Boot as the Java web framework.

Consequences: Spring Boot has the benefit of being a simple and widely used platform to use, meaning that the creation and the maintenance of the project will be manageable.

ADR-003: Use MySQL for Database

Status: Accepted (Mandated)

Context: A database is needed for this project, and a relational database has been selected for this project. MySQL is the mandated database.

Decision: We will use MySQL as the relational database for this project.

Consequences: A relational database is ideal for this project, in which the data stored will not be of a high complexity.

ADR-004: Use Model View Controller (MVC) Design Pattern

Status: Accepted (Mandated)

Context: A set design pattern is needed to define the direction of the project and create a plan on how the project is structured. The use of the MVC design pattern is mandated and is one that team members have experience with.

Decision: We will use the MVC design pattern as mandated.

Consequences: Having a set design pattern is critical to guiding the structure of the project. The MVC design pattern also creates a testable project – a key aspect of the final design. On the other hand, the restricted design pattern limits the flexibility of the structure of the project, not allowing on-the-fly adjustments that may be needed.

ADR-005: Use JUnit5 for Unit Testing

Status: Accepted (Mandated)

Context: A unit testing framework is needed for the project. JUnit5 is mandated and is commonly used for Java projects.

Decision: We will use JUnit5 as mandated.

Consequences: JUnit5 is widely used and simple to implement for unit testing, and the experience team members have in using JUnit5 will add to the ease of use throughout the project.

7. User Interface Design

7.1 General User Interfaces

7.2 Event Manager Interfaces

7.3 Admin Pages

7.4 Shared Pages

8. Appendix A: Glossary

- **API:** Application Programming Interface.
- **APP:** Australian Privacy Principles.
- **CSRF:** Cross-Site Request Forgery.
- **ER Diagram:** Entity-Relationship Diagram.
- **FR:** Functional Requirement.
- **MVC**: Model View Controller
- **MVP:** Minimal Viable Product.
- **NFR:** Non-Functional Requirement.
- **RBAC:** Role-Based Access Control.
- **RSVP:** Répondez s'il vous plaît (Event attendance confirmation).
- **SMTP:** Simple Mail Transfer Protocol.
- **SSO:** Single Sign-On.
- **XSS:** Cross-Site Scripting.

9. Appendix B: Analysis Models

- **Use Case Diagrams:** Shows the interaction between Students, Organizers, Administration System, and the EventHub System.

- **Entity Relationship Diagram:**
  Entities: User, Event, RSVP, Feedback

Relationships: User - Event**, User –** RSVP, Event - Feedback, RSVP – Event.

10. Appendix C: To Be Determined List

- Final selection of hosting platform
- Choice of email service provider
- Final UI design