Software Requirements Specification

for

Rikugan

A Bounty-Based Project Management System

Version 1.0
Printing Date: December 22, 2025

Group ID: Group 9

Group Members:
1155211784 Fok Chun Yin
1155211541 HO Sum Ming
1155211779 Chi Ho KWOK
1155211479 NG Ka Long
1155214295 SIN Lok Man

Revision History:

Version	Date	Author	Description
1.0	December 13, 2025	Group 9	Initial release

Contents

1 Introduction **3**
- 1.1 Purpose . . . 3
- 1.2 Document Conventions . . . 3
- 1.3 Product Scope . . . 3
- 1.4 References . . . 3

2 Overall Description **4**
- 2.1 Product Perspective . . . 4
- 2.2 Product Functions . . . 4
- 2.3 User Classes and Characteristics . . . 4
- 2.4 Operating Environment . . . 4
- 2.5 Design and Implementation Constraints . . . 4
- 2.6 Development Timeline . . . 5
- 2.7 Assumptions and Dependencies . . . 5

3 System Features **6**
- 3.1 User Authentication and Authorization . . . 6
- 3.2 Task Management System . . . 6
- 3.3 User Profile and Reward System . . . 6
- 3.4 Notification System . . . 7

4 Other Non-Functional Requirements **8**
- 4.1 Performance Requirements . . . 8
- 4.2 Security Requirements . . . 8
- 4.3 Software Quality Attributes . . . 8
- 4.4 Business Rules . . . 8

5 Other Requirements **9**
- 5.1 Database Requirements . . . 9
- 5.2 Development Environment . . . 9
- 5.3 Legal and Compliance . . . 9
- 5.4 Testing Requirements . . . 9

1 Introduction

1.1 Purpose

This SRS document specifies requirements for Rikugan v1.0, a bounty-based project management web application combining Kanban functionality with gamified task assignment and rewards. It covers user authentication, task management, role-based access control, and reward tracking.

1.2 Document Conventions

Bold text indicates system components; *italic text* represents actions/responses; `monospace` denotes technical elements. Requirements use REQ-XX format. Priority levels: High (essential), Medium (important), Low (nice-to-have).

1.3 Product Scope

Rikugan is a web-based project management system using gamification and role-based task assignment. It provides task visibility through Kanban status, developer motivation via bounty rewards, and structured role hierarchies (Goons, Hashira, Oyakatasama) for accountability and collaboration.

1.4 References

IEEE Std 830-1998; React, Vite, HeroUI, MySQL, and Docker documentation available at respective official websites.

2 Overall Description

2.1 Product Perspective

Rikugan is a self-contained web application for project management with gamification via a bounty-based reward system. Architecture: React/Vite frontend, Node.js backend, MySQL database, Docker containerization.

2.2 Product Functions

- **Authentication**: Secure login, role-based access control, team license validation
- **Task Management**: Bounty creation, Kanban-like status tracking, deadline penalties
- **User Management**: Profiles, earnings tracking, performance monitoring
- **Notifications**: Task assignments, deadline reminders, achievements
- **Analytics**: Task statistics, performance metrics, progress visualization

2.3 User Classes and Characteristics

- **Goons**: Entry-level programmers; task selection, status updates, profile viewing; basic permissions
- **Hashira**: Senior programmers; all Goon functions plus task creation and team monitoring; elevated permissions
- **Oyakatasama**: Administrators; full system access including user management and configuration; admin privileges

2.4 Operating Environment

Client: Modern browsers (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+), JavaScript enabled, 1024x768 min resolution. **Server**: Node.js v16+, MySQL 8.0+, Docker, Ubuntu 20.04 LTS, 4GB RAM, 20GB storage. **Development**: React 18+, Vite, HeroUI, ES2020+.

2.5 Design and Implementation Constraints

Technical: React/Vite web app, MySQL database, Docker containers, HeroUI components. **Policy**: Secure authentication, basic privacy protection. **Resources**: One-week development timeline, 5-person student team, educational project, development environment only. **Interface**: Responsive design for desktop/tablet, basic accessibility compliance.

2.6 Development Timeline

The project follows a 9-day development schedule using the Waterfall model:

- **Day 1-2**: Planning - Requirement analysis and design planning
- **Day 3-6**: Implementation - Backend and frontend development
- **Day 7-8**: Testing and Validation - Quality assurance and bug fixes
- **Day 9**: User Manual and Delivery - Documentation and project submission

2.7 Assumptions and Dependencies

Assumptions: Users familiar with web apps, team has development tools, MySQL/-Docker properly configured. **Dependencies**: React, HeroUI, MySQL, Docker, Node.js, Vite. **Risks**: Learning curve, Docker setup complexity, time constraints, library compatibility.

3 System Features

3.1 User Authentication and Authorization

Priority: High - Essential for security and role-based access control.

REQ-1: User Registration Oyakatasama can create accounts with username (3-50 chars), email, password (8+ chars), and role.

REQ-2: User Login Authenticate via username/email and password; 8-hour session timeout.

REQ-3: Role-Based Access **Goons**: Task selection, status updates, profile viewing. **Hashira**: All Goon functions plus task creation and team monitoring. **Oyakatasama**: All functions plus user/system administration.

REQ-4: License Management Hard-coded team license key via environment config; invalid license blocks all access.

3.2 Task Management System

Priority: High - Core project management functionality.

REQ-5: Task Creation Hashira create tasks with name (max 100 chars), description (max 1000 chars), bounty amount (positive), deadline (future date), priority, and skill tags.

REQ-6: Task Assignment Goons select available tasks; automatic assignment to first qualified user.

REQ-7: Kanban-Like Status Display tasks : Available, In Progress, Review, Completed.

REQ-8: Task Status Updates Assigned users update status and add progress comments.

REQ-9: Deadline Management Track deadlines; apply monetary penalties for missed deadlines.

REQ-10: Task Deletion Hashira/Oyakatasama delete tasks only when Available or unassigned.

3.3 User Profile and Reward System

Priority: Medium - Gamification and engagement.

REQ-11: User Profile Maintain username, email, role, balance, task history, and performance statistics.

REQ-12: Bounty Rewards Auto-credit user accounts upon task completion.

REQ-13: Penalty System Apply configurable penalties for missed deadlines.

REQ-14: Performance Tracking Display tasks completed, avg completion time, success rate, total earnings.

3.4 Notification System

Priority: Medium - User engagement and communication.

REQ-15: Task Notifications Notify on: task assignment, 24hr deadline reminders, status updates, completion/cancellation.

4 Other Non-Functional Requirements

4.1 Performance Requirements

- Page load: ¡3s; API response: 95% ¡500ms; Database queries: ¡2s; Login: ¡1s
- Support 50 concurrent users, 100 task updates/min, notifications within 5s
- Capacity: 200 users, 1000 active tasks
- Frontend ¡100MB RAM/tab, DB ¡1GB, server CPU ¡70%

4.2 Security Requirements

- **Authentication**: 8+ char passwords, bcrypt hashing, JWT tokens, role-based access, secure logout
- **Data Security**: HTTPS, encrypted credentials, parameterized queries, input sanitization, XSS prevention
- **System**: Rate limiting, security logging, no sensitive info in errors

4.3 Software Quality Attributes

- **Usability**: Intuitive interface, consistent navigation, clear errors, responsive design, WCAG 2.1 Level A
- **Reliability**: 95% uptime, graceful degradation, automatic error recovery, data consistency
- **Maintainability**: Modular React code, API docs, separation of concerns, ESLint standards, 70% test coverage
- **Portability**: Cross-browser (Chrome/Firefox/Safari/Edge), Docker containers
- **Testability**: Unit testing, automated API testing, test data procedures

4.4 Business Rules

- **User Roles**: Only Oyakatasama manage accounts; Hashira/Oyakatasama create tasks; Goons/Hashira work on tasks; users update only assigned tasks
- **Tasks**: One user per task, positive bounties, future deadlines, completed tasks immutable, status flow: Available → In Progress → Review → Completed
- **Financial**: Auto bounty payment on completion, auto penalties on missed deadlines, -$100 balance minimum, all transactions logged, admin-only manual adjustments
- **License**: One hard-coded key per team via environment config; invalid license blocks all access
- **Notifications**: 24hr deadline reminders, immediate assignment alerts, milestone achievements, admin priority

5 Other Requirements

5.1 Database Requirements

MySQL 8.0+, InnoDB, UTF-8, daily backups. Core tables: users (id, username, email, password_hash, role, balance, timestamps), tasks (id, title, description, bounty_amount, deadline, status, created_by, assigned_to, timestamps), notifications (id, user_id, type, message, read_status, created_at), licenses (id, license_key, expiry_date, status, max_users, created_at). Indexes on user_id, task_status, created_at.

5.2 Development Environment

Docker 20.0+ with compose, Node.js 16+, npm/yarn, Git, Vite build, hot reload, environment config (.env), automated testing.

5.3 Legal and Compliance

Educational use only, no commercial transactions, university policy compliance, open source libraries, MIT license. Privacy: minimal data collection (username/email), local environment only, user consent, data deletion on account termination.

5.4 Testing Requirements

Jest for unit tests, 70% coverage for critical logic, API endpoint testing, React component integration testing, E2E for critical workflows. Test database with seed data, cleanup procedures.