Software Requirements Specification

for

Live Scoreboard API

Version 1.0 approved

Prepared by Tran Quoc Tuan

99Tech

21th July 2025

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

2. Module Responsibilities 1

3. Functional Requirements 1

3.1 Increment Score Endpoint (POST /api/score/increment) 1

3.2 Top Scores Endpoint (GET /api/score/top) 1

3.3 Real-time Scoreboard Updates (WebSocket /ws/scoreboard) 2

4. Non-functional Requirements 2

5. Suggested Technologies 2

6. Data Model 2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

The Live Scoreboard API module manages user scores for an interactive website, handling real-time updates to a publicly visible leaderboard. It securely increments individual user scores upon validated action completions and supplies the current top 10 scores for display via a REST API and WebSocket push.

# Module Responsibilities

* Receive and validate requests to increase a user's score following completion of an authorized action.
* Prevent unauthorized or repeated score increments by malicious users.
* Maintain and expose an up-to-date top 10 leaderboard to frontend clients.
* Broadcast real-time leaderboard updates to all connected clients.

# Functional Requirements

## Increment Score Endpoint (POST /api/score/increment)

- **Accepts**:

- Authenticated access token (JWT or session cookie)

- Payload: { actionId }

- **Behaviour**:

- Validates user authentication.

- Verifies actionId has not already been used by this user (to prevent replay attacks).

- Updates user's score for a successful, unique action.

- Returns updated score for the user and position in leaderboard.

**- Error Handling:**

- 401 Unauthorized: If authentication is invalid.

- 409 Conflict: If actionId has already been used or action is invalid.

- 500 Server Error: For unhandled exceptions.

## Top Scores Endpoint (GET /api/score/top)

**- Returns:**

- Array of top 10 users: [{ username, score }]

- User’s own rank and score, if logged in.

**- Behaviour:**

- Queries the current highest 10 scores, sorted descending.

- Includes requesting user's standing, even if not in top 10.

## Real-time Scoreboard Updates (WebSocket /ws/scoreboard)

- **Broadcasts**:

- Whenever top 10 leaderboard changes, sends updated leaderboard to all connected clients.

- Authentication via token during handshake.

# Non-functional Requirements

- **Security**:

- All requests must be authenticated and authorized.

- Only server-validated user actions can increase scores.

- All endpoints must be rate-limited (per user and per IP).

- **Performance**:

- Leaderboard fetch/query and update must complete within 200ms under expected load.

- Leaderboard changes must propagate to subscribed clients in under 1 second.

- **Auditing**:

- All score changes must be logged with userId, actionId, timestamp, and resulting score.

- **Scalability**:

- Support for horizontal scaling.

- Shared leaderboard data store (e.g., Redis, database).

- **Extensibility**:

- Designed to support multiple different types of actions in the future.

- Flexible for batch processing or aggregation if needed.

# Suggested Technologies

- **Backend**: Node.js / Python / Java, Express / FastAPI / Spring Boot.

- **Database**: PostgreSQL, MongoDB, or similar.

- **Realtime**: Redis Pub/Sub or WebSocket server.

- **Caching**: Redis.

# Data Model

**- User**

- id

- username

- score

**- ScoreAction**

- id

- userId

- actionId (unique per user)

- timestamp