AllFence - An interactive database for developing fencing federations to keep track of rankings, competitions, and fencers

An Implementation Project by Quentin Geoffroy

AllFence is an interactive and centralized database aimed to support the needs of emerging fencing federations, with particular relevance to the context of smaller countries like Laos. In many fencing federations, the absence of consistent and accessible systems for storing athlete information, competition results, and tracking national rankings poses challenges to long-term development. As an athlete actively representing Laos in fencing, I have seen the lack of unified data-management infrastructure and the operational difficulties this forms for athletes, coaches, and federation delegates. *AllFence* is envisioned as a practical solution to address these ongoing challenges and provide a foundation for systematic sport development, which may serve as a catalyst for other less-recognized sports.

My objective is to build a structured relational database that registers key information about fencers, clubs, competitions, and ranking history. Then, a user-friendly web interface will allow federation administrators to manage athlete data and competition results, while the public-facing components will make rankings, competition information accessible to the broader community. Features like an automatic ranking system (maybe an ELO system), controlled vocabularies for weapons and event types, and efficient search and filtering tools will allow the platform to remain reliable, transparent, and aligned with the needs of smaller federations. Particular attention will be given to ensure data integrity, and the implementation of intuitive retrieval mechanisms.

This project draws directly from many of the concepts in Information Organization and Retrieval. For instance, database design will reflect principles of structured metadata, classification, normalization, and appropriate indexing strategies to facilitate frictionless information retrieval. Search functions inspired by existing database systems for other sports will be built to support keyword queries, faceted browsing, and role-based access, demonstrating a thorough understanding of retrieval models and user-centered design. The implementation of this will also account for controlled vocabularies and consistent naming conventions to ensure clarity and reliability across the system.

In summary, *AllFence* is an idea that came from a clear need in the sport of fencing. Its objective is to serve as a meaningful real-world system implementing the course concepts while addressing a genuine need within the development of the sport in Laos. By providing a centralized, transparent and accessible platform for tracking athletes and competition results, the system has the potential to support development, enhance administrative tasks, and contribute to the growth of the sport nationally.

The final prototype will be hosted publicly and accompanied by appropriate documentation to ensure accessibility for grading and future iterations.

AllFence: Prototype Design Documentation

Introduction and Overview

AllFence is a centralized, interactive database and web application designed to support the operational and developmental needs of emerging fencing federations, such as the one in Laos. The primary goal is to replace disparate, inconsistent data management methods with a unified, structured system for tracking fencer information, competition results, and national rankings.

Target Users: Federation Administrators, Coaches, Fencers, and the general public

Core Value Proposition: To provide a transparent, reliable, and accessible platform that facilitates systematic sport development, enhances administrative efficiency, and ensures data integrity and consistency.

Key Functionality: Fencer registration, competition management, automatic ranking calculation (e.g., ELO-based system), and public access to official rankings and results

System Architecture

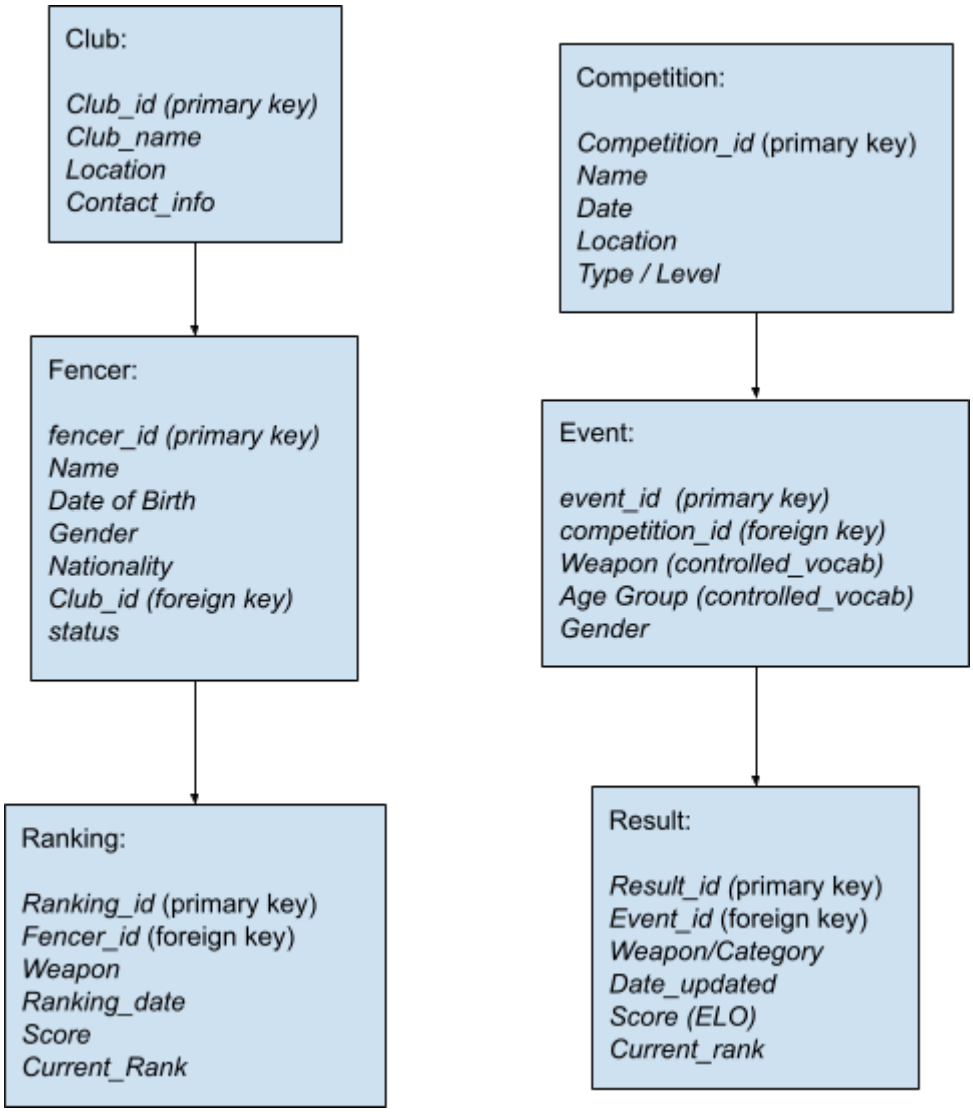
The *AllFence* system will follow a three-tier architecture typical of modern web applications:

Tier	Component	Technology/Role
Client	Web Interface	HTML, CSS, JavaScript. Handles user interaction and displays data
Server / Logic	Web Server and Backend logic	Python (Flask, Django). Implements business logic, handles user authentication/authorization, processes ranking calculations, and serves the API
Data	Relational Database	PostgreSQL or MySQL. Stores all persistent data, including fencer profiles, competition data, and ranking history

Communication: The Presentation Tier communicates with the Application Tier via RESTful API calls

Hosting: The final prototype will be hosted publicly using a service like Heroku, Vercel, or a simple cloud VM for accessibility and testing.

Data Design



To ensure data consistency and reliability, static controlled vocabularies will be implemented for:

1. Weapon: Foil, Épée, Sabre
2. Gender: Male, Female
3. Age Group: Senior, Junior, Cadet, U15, etc.

Interface Design

The Application Tier requires an Application Programming Interface (API) to manage data transactions between the front-end and the database

API Endpoint	HTTP Method	Description	User Role Access
/fencers	GET	Retrieve a list of all fencers (supports search/filter).	Public
/fencers/{id}	GET	Retrieve a specific fencer's profile and history.	Public
/fencers	POST	Register a new fencer.	Admin
/competitions	GET	Retrieve a list of all competitions.	Public
/competitions/{id}/results	GET	Retrieve results for a specific competition.	Public
/results	POST	Submit/record competition results.	Admin
/rankings/{category}	GET	Retrieve current rankings for a specified category.	Public

/admin/calculate-rankings	POST	Trigger the automatic ranking calculation process.	Admin

Component Design

Model: A modified ELO rating system or a points-based system (reflecting FIE/local standards will be implemented)

Process: A scheduled or manually triggered backend service will:

1. Fetch all results since the last ranking update
2. Apply a weighting algorithm based on the competition type
3. Calculate the new score/ELO rating for all participating fencers
4. Update the Ranking table

Transparency: The calculation logic will be documented to ensure the ranking system is transparent and easily auditable

Search and Retrieval

Keyword Query: Standard keyword search across fencer names, competition names, and club names

Faceted Browsing: Users will be able to filter search results by:

- Weapon
- Age Group
- Competition Date
- Club

Implementation Optimized SQL queries utilizing the defined indexes will ensure efficient retrieval

User Interface Design

Component	Target User	Key Features	Design Focus
Public Portal	Fencers, Coaches, General Public	View current rankings, search fencer profiles, view competition results, calendar of events.	**Accessibility, Transparency, Simplicity.**

Administrator Dashboard	Federation Delegates	Add/Edit Fencer Profiles, Create/Schedule Competitions, Enter/Verify Competition Results, Trigger Ranking Calculation.	**Data Integrity, Intuitive Data Entry, Security.**

Design Principle: User-Centered Design will be applied, focusing on clean, minimalist design with clear navigation paths, consistent naming conventions, and validation on all data input forms

Assumptions and Dependencies

Assumptions

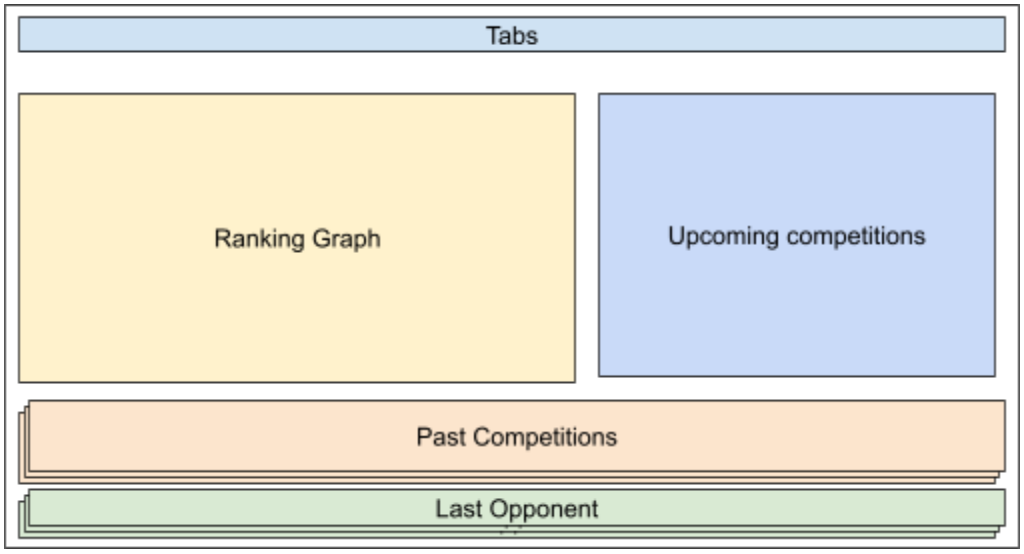
1. **Federation will designate authorized personnel to manage the data**
2. Consistent competition results will be made available to the administrators for entry
3. Chosen hosting environment can support the modest traffic and data storage needs of a prototype for an emerging federation

Dependencies

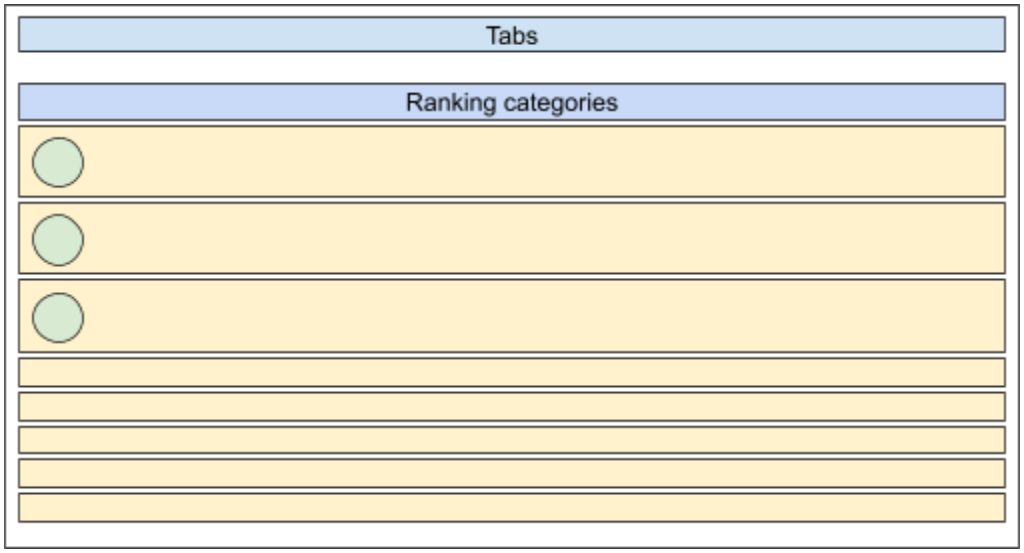
1. Backend: (Flask/Django, Express.js) must be correctly configured
2. Database Server: PostgreSQL instance must be running and accessible to the Application Tier
3. Authentication Library: A standard library is required for securing the Administrator Dashboard and implementing role-based access

Main pages:

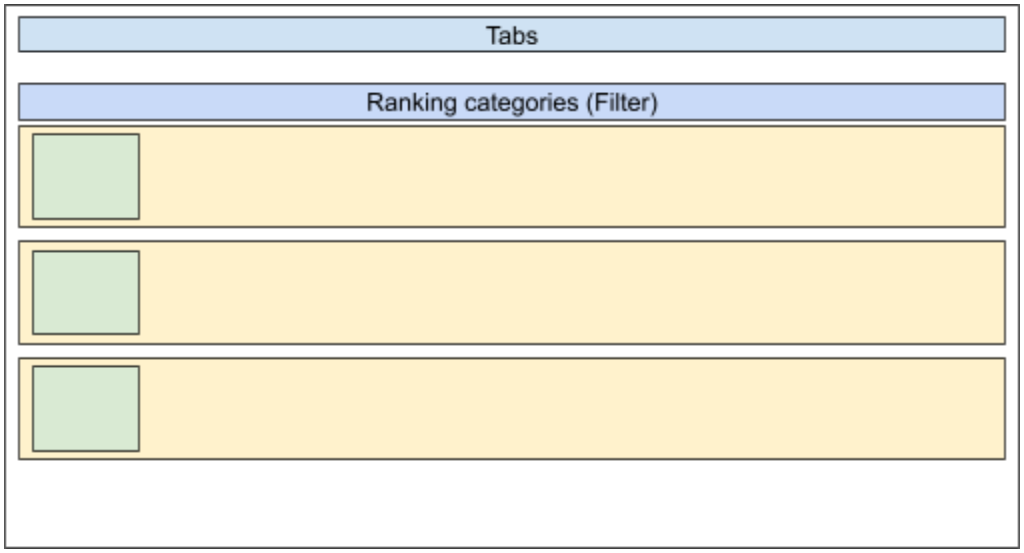
1. Dashboard (Also Profile):



2. Rankings



3. Competitions



Database design