

UFC Manager System Requirements

1. Functional requirements

1.1. Fighter management

CRUD operations for fighters (create, read, update, delete)

Possibility to deactivate/return a fighter to the roster

Accounting for weight category, country, statistics (record in UFC and total in MMA)

Displaying the flag of the fighter's country

1.2. Tournament management

Creating, editing and viewing tournaments

Adding fights to a tournament with participants, weight category and result

Importing tournament history from a TXT file

1.3. Ratings (ELO system)

Automatic recalculation of rating after each battle

Manual recalculation of ratings (for administrators)

Filtering rating by weight categories

1.4. Authorization and roles

Guest: viewing fighters, tournaments, ratings

Manager: editing fighters and tournaments

Admin: managing users, forced recalculation of ratings

1.5. Additional functions

Statistics on fighters (wins/losses, knockouts, submissions)

Search and filter fighters (by name, country, weight category)

2. Non-functional requirements

2.1. Performance

API response time ≤ 500 ms

Support up to 1000 concurrent users

2.2. Security

Authentication via JWT

SQL injection and XSS protection

Role-based access model (RBAC)

2.3. Scalability

Possibility of horizontal scaling of the server part

Caching of frequently requested data (Redis)

2.4. Reliability

Database backup

Critical operations logging

2.5. Usability

Adaptive interface (Bootstrap)

Mobile device support

Use-Cases

ID	Name	Actor	Description
UC-01	View list of fighters	Guest	User sees a table of fighters with filtering option
UC-02	Add new fighter	Manager	Manager fills out the form and saves fighter data
UC-03	Create tournament	Manager	Manager creates a tournament, adds fights and saves
UC-04	Import tournaments from file	Manager	Loads a TXT file with tournament history
UC-05	Rating recalculation	Admin	Admin starts forced recalculation of ELO ratings
UC-06	User management	Admin	Admin adds/removes users and assigns roles

Objects, classes and their relationships

1. Basic entities (classes)

1.1. Fighter

id: Long

name: String

country: String

weightClass: Enum

ufcRecord: String (for example, "12-3-0")

mmaRecord: String

eloRating: Double

isActive: Boolean

1.2. Tournament

id: Long

name: String

date: LocalDate

fight: List<Fight>

1.3. Fight

id: Long

fighter1: Fighter

fighter2: Fighter

winner: Fighter

weightClass: Enum

rounds: Integer

1.4. User

id: Long

username: String

password: String

role: Enum (GUEST, MANAGER, ADMIN)

2. Relationships between classes

Tournament → Fight (1:N) – The tournament contains several fights.

Fight → Fighter (N:2) – Two fighters are involved in the fight.

User → Roles (N:1) – The user has one role