

Vize

PROČ?

Mým cílem je vytvořit systém, který poskytuje jednoduchý a srozumitelný přehled informací o MMA bojovnících, kde je možné zjistit základní informace o bojovnících, jejich aktuální formu, historii zápasů, tréninkové termíny atd. Celý tento systém by také umožňoval správcům efektivně pracovat s daty, například přidávat tréninky, plánovat zápasy a mnoho dalších funkcí.

CO?

Systém propojený s databází slouží k „informování“ uživatelů o bojovnících a správě celé databáze fighterů. Uživatelé budou moci získat přístup k informacím o jednotlivých bojovnících (biografie, historie zápasů, aktuální forma, tréninkové termíny). Správu těchto údajů pomocí CRUD funkcí (vytváření, aktualizace, mazání, čtení) bude moci provádět pouze správce systému. Kromě toho systém umožní trenérům plánovat zápasy, které následně musí bojovníci potvrdit (přijmout nebo odmítnout).

JAK?

Systém bude implementován pomocí agilní metodiky, kde se zaměřujeme na průběžnou správu a zpracování informací. CRUD operace (vytváření, čtení, aktualizace, mazání) budou klíčovými procesy pro správu dat v databázi. Uživatelé budou moci číst informace o bojovnících, zatímco správce systému bude mít přístup ke kompletní správě těchto dat, včetně přidávání tréninků, plánování zápasů atd. Trenéři budou moci plánovat zápasy, které bojovníci následně přijmou nebo odmítnou, ale nebudou mít přístup k ostatním funkcím pro správu systému. Systém bude postaven na frameworku Spring, což umožní snadnou integraci s databázemi, jako je MySQL. Databáze bude strukturovaná tak, aby umožňovala rychlý přístup a efektivní manipulaci s daty.

KDE?

Jedná se o webovou aplikaci vytvořenou v programovacím jazyce Java s využitím frameworku Spring. Aplikace bude nasazena na serveru a přístupná z jakéhokoliv zařízení s internetovým připojením. Pro ukládání a správu dat bude použita databáze MySQL. Systém bude přístupný pro běžné uživatele v tzv. „uživatelském režimu“, kde budou mít přístup k informacím o bojovnících. Trenéři budou mít přístup k funkcím pro plánování zápasů. Bojovníci k jejich potvrzení, odmítnutí, a ke zkouknutí jejich historie zápasů, zatímco správci systému budou moci provádět kompletní operace CRUD a spravovat celý systém.

KDO?

Běžný uživatel → Tato skupina uživatelů bude mít přístup k informacím o bojovnících, jako jsou biografie, aktuální forma, historie zápasů a tréninkové termíny. Běžní uživatelé budou moci pouze číst data a nebudou mít práva na jejich úpravu.

Trenéři → Trenéři budou mít přístup pouze k funkcím, které se týkají plánování zápasů. Budou moci zadat nové zápasy, které následně bojovníci mohou přijmout nebo odmítnout.

Bojovníci → Bojovníci budou mít možnost prohlížet informace o svých zápasech. Bojovníci budou moci potvrzovat nebo odmítat zápasy naplánované trenéry.

Správce systému → Správci budou mít plný přístup k systému, včetně možnosti provádět CRUD operace s daty o bojovnících, trénincích a dalších informacích. Budou zodpovědní za celkovou správu databáze.

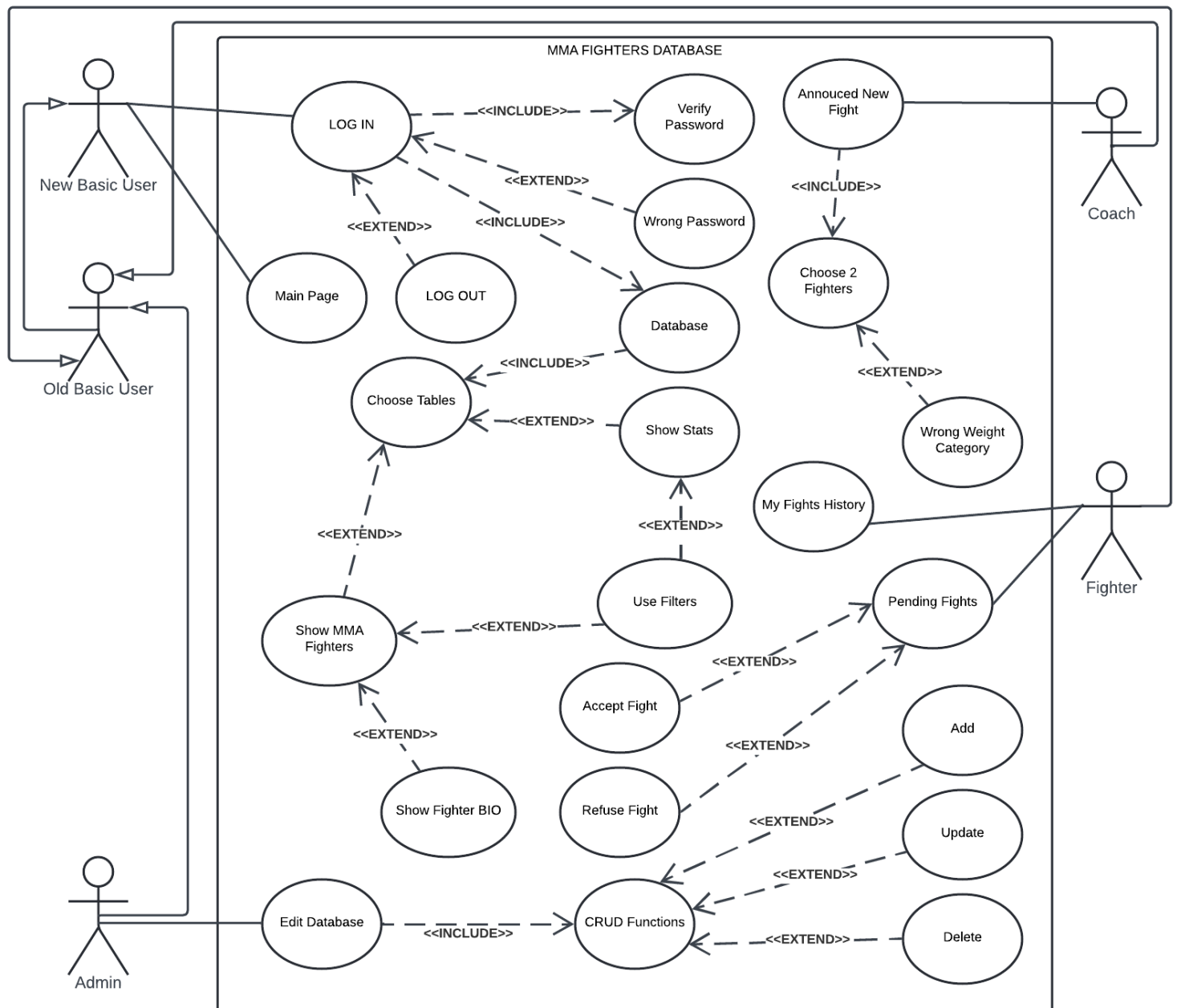
KDY?

Při aktualizaci informací o bojovnících → Správci systému budou provádět aktualizace, jako je přidávání nových bojovníků, aktualizace tréninkových termínů nebo historie zápasů.

Při plánování zápasů → Trenéři budou zadávat nové zápasy, které následně bojovníci potvrdí nebo odmítnou.

Při každodenní práci s informacemi → Běžní uživatelé, bojovníci anebo trenéři, budou systém používat denně k prohlížení informací o nadcházejících zápasech atd.

Scenarios, Use Case Model and Activity diagrams



Zobrazení informací o bojovníkovi

Aktéři → Basic User

Vstupní podmínky → Basic User je přihlášen do systému.

Spouštěč → Basic User chce zobrazit BIO konkrétního bojovníka.

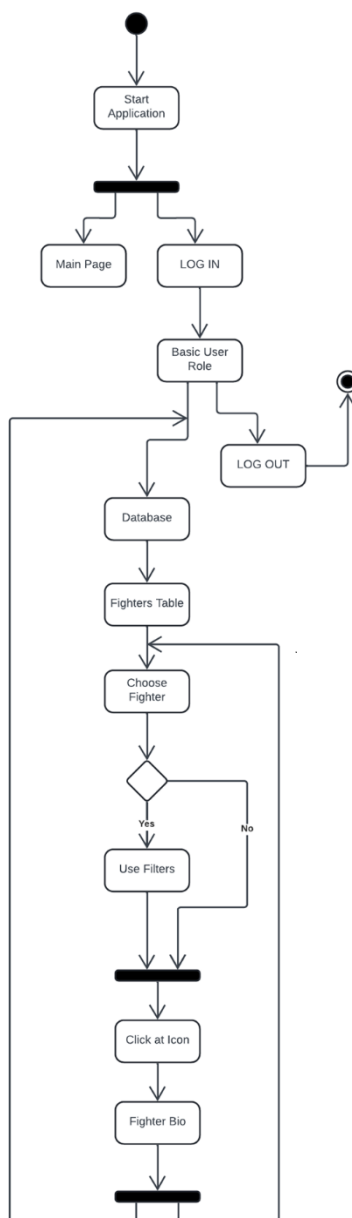
Úspěšný scénář →

1. Basic User se přihlásí do systému přes funkci LOG IN.
2. Po úspěšném přihlášení systém zobrazí Database, která nabídne Basic Userovi možnost ukázat tabulku s bojovníky.
3. Po zakliknutí tlačítka k ukázaní tabulky MMAFighters.
4. Basic User klikne na ikonku lupy u daného bojovníka.
5. Systém zobrazí modál okno s bojovníkovým BIEM (celé jméno, skóre, váhovka a národnost).

Alternativní scénář →

Alternativa 1: Špatné přihlašovací údaje (Wrong Password)

- Pokud Basic User zadá nesprávné přihlašovací údaje, systém zobrazí hlášení o chybě (Wrong Password) a umožní Basic Userovi zadat heslo znovu.



CRUD funkce v databázi

Aktéři → Admin

Vstupní podmínky → Admin je přihlášen do systému

Spouštěč → Admin chce pomocí CRUD funkcí například přidat, upravit nebo smazat bojovníka z celé databázi.

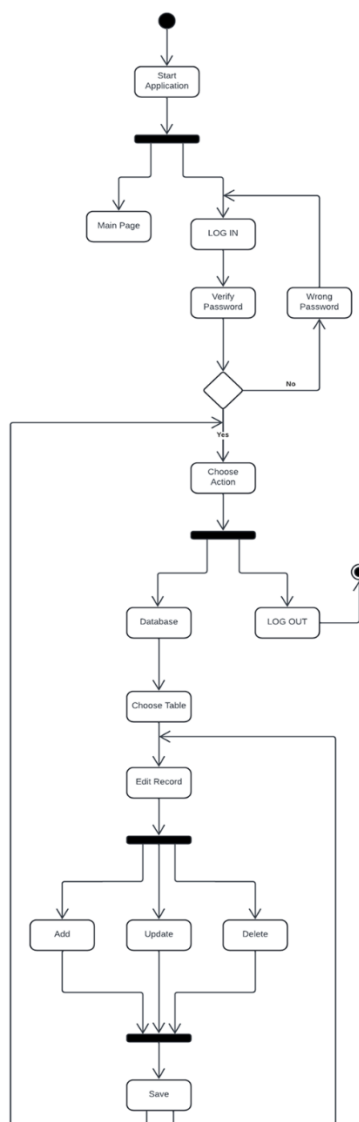
Úspěšný scénář →

1. Admin se přihlásí do systému přes funkci LOG IN.
2. Po úspěšném přihlášení systém zobrazí databázi všech tabulek.
3. Admin si vybere tabulku MMAFighters.
4. Systém k zobrazenému seznamu přidá k databázi jednotlivé ikonky pro vytváření, mazání a editaci.
5. Admin vybere konkrétní akci.
6. Admin potvrdí změny a systém je uloží do databáze.
7. Systém zobrazí aktualizovaný seznam bojovníků.

Alternativní scénář →

Alternativa 1: Špatné přihlašovací údaje (Wrong Password)

- Pokud admin zadá nesprávné přihlašovací údaje, systém zobrazí hlášení o chybě (Wrong Password) a umožní admin zadat heslo znovu.



Plánování zápasu pro bojovníka

Aktéři → Coach, Fighter

Vstupní podmínky → Coach a poté Fighter jsou přihlášení do systému

Spouštěč → Coach chce zapsat Fightery na zápas, který poté Fighteři přijmou nebo odmítnou.

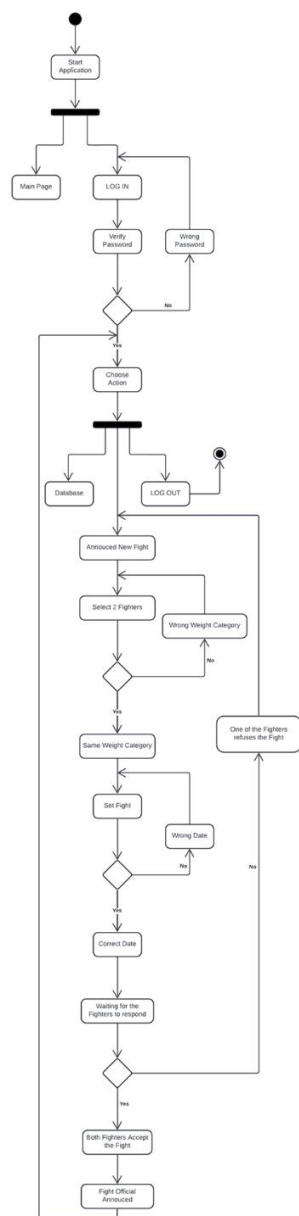
Úspěšný scénář →

1. Coach se přihlásí do systému přes funkci Log In.
2. Po úspěšném přihlášení systém zobrazí Coachovi záložku Annouced New Fight.
3. Coach tuto možnost si vybere a oznámí dvěma fighterům, že jim domluvil zápas mezi sebou.
4. Fighteři si oznámení všimnou a buď zápas přijmou nebo ne.
5. Po potvrzení obou fighterů se zápas oficiálně zapíše do databáze.

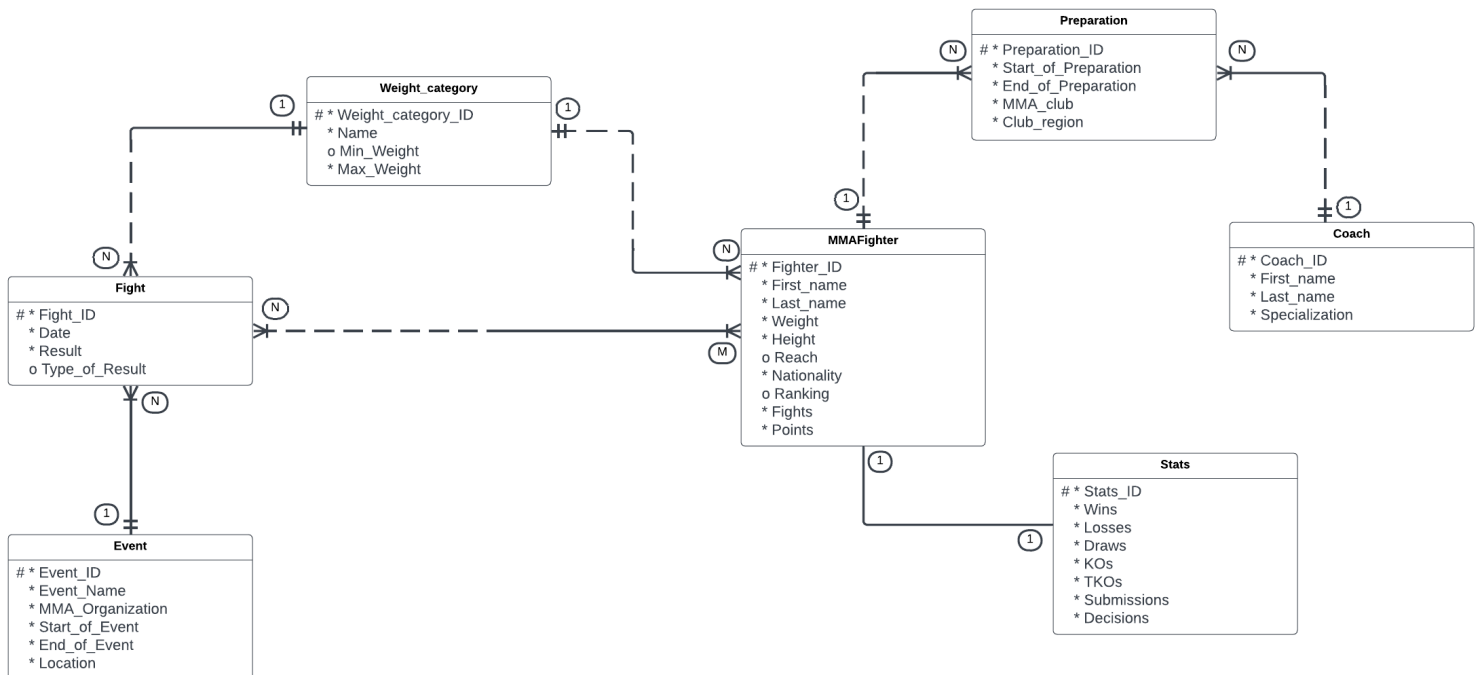
Alternativní scénář →

Alternativa 1: Špatné přihlašovací údaje (Wrong Password)

- Pokud coach zadá nesprávné přihlašovací údaje, systém zobrazí hlášení o chybě (Wrong Password) a umožní coach zadat heslo znovu.



Conceptual Domain Model



Odhady velikostí entit a jejich množství:

- **MMAFighter** → 1 kB, 1000 záznamů, 1 mB.
- **Fight** → 500 B, 500 záznamů, 300 kB.
- **Event** → 250 B, 200 záznamů, 150 kB.
- **Preparation** → 2 kB, 2000 záznamů, 1-1.5 mB.
- **Stats** → 500 B, 1000 záznamů, 1 mB.
- **Coach** → 350 B, 300 záznamů, 200 kB.
- **Weight_category** → 80 B, 50 záznamů, 35 kB.
- **MMAFight** → 800 B, 1000 záznamů, 500 kB.

Odhad počtu uživatelů současně pracujících se systémem:

- **Průměrný počet uživatelů** → 1000 – 3000 uživatelů.
- **Peak** → 10000 – 15000 uživatelů.

Typy interakcí uživatelů se systémem a odhad jejich náročnosti:

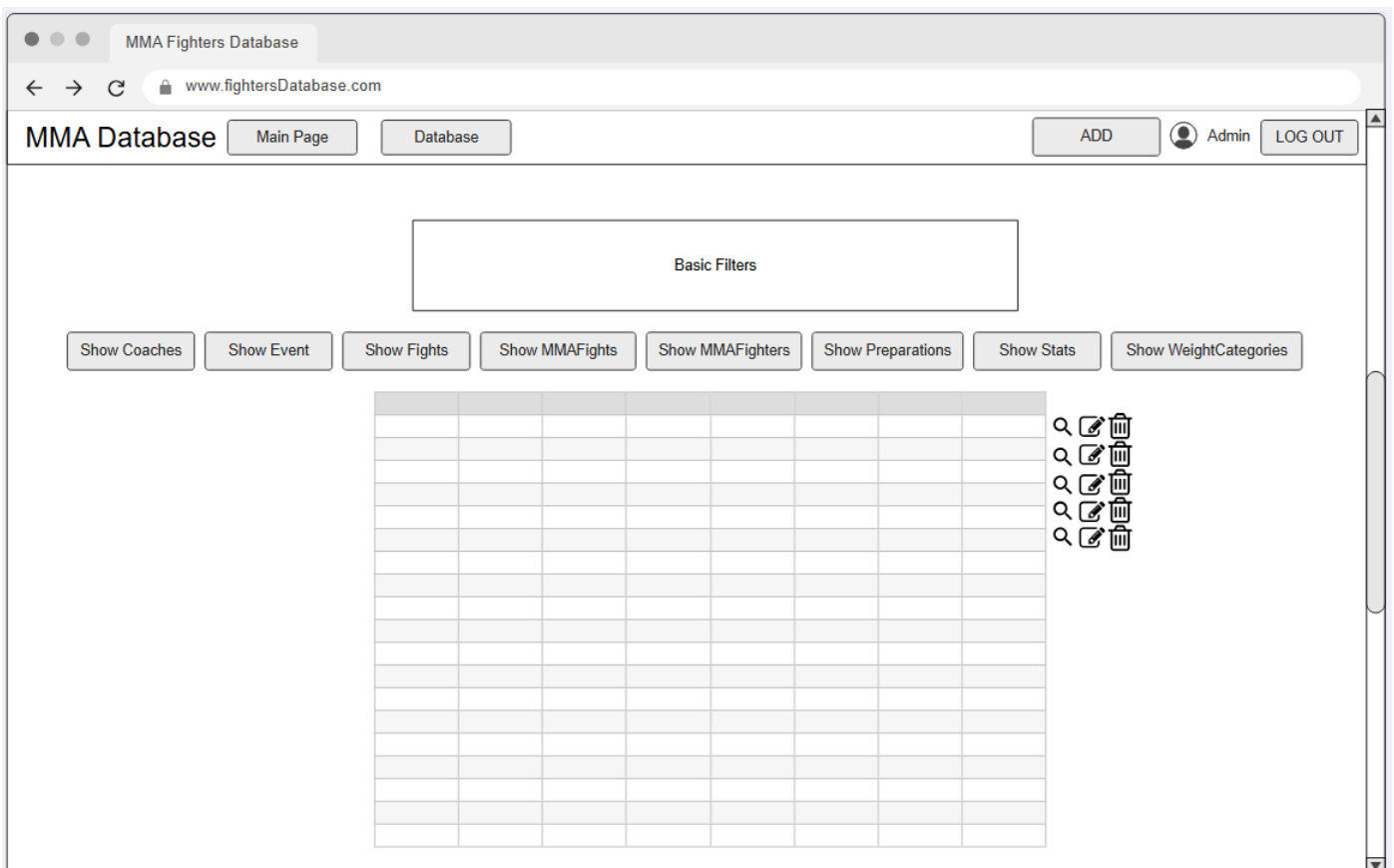
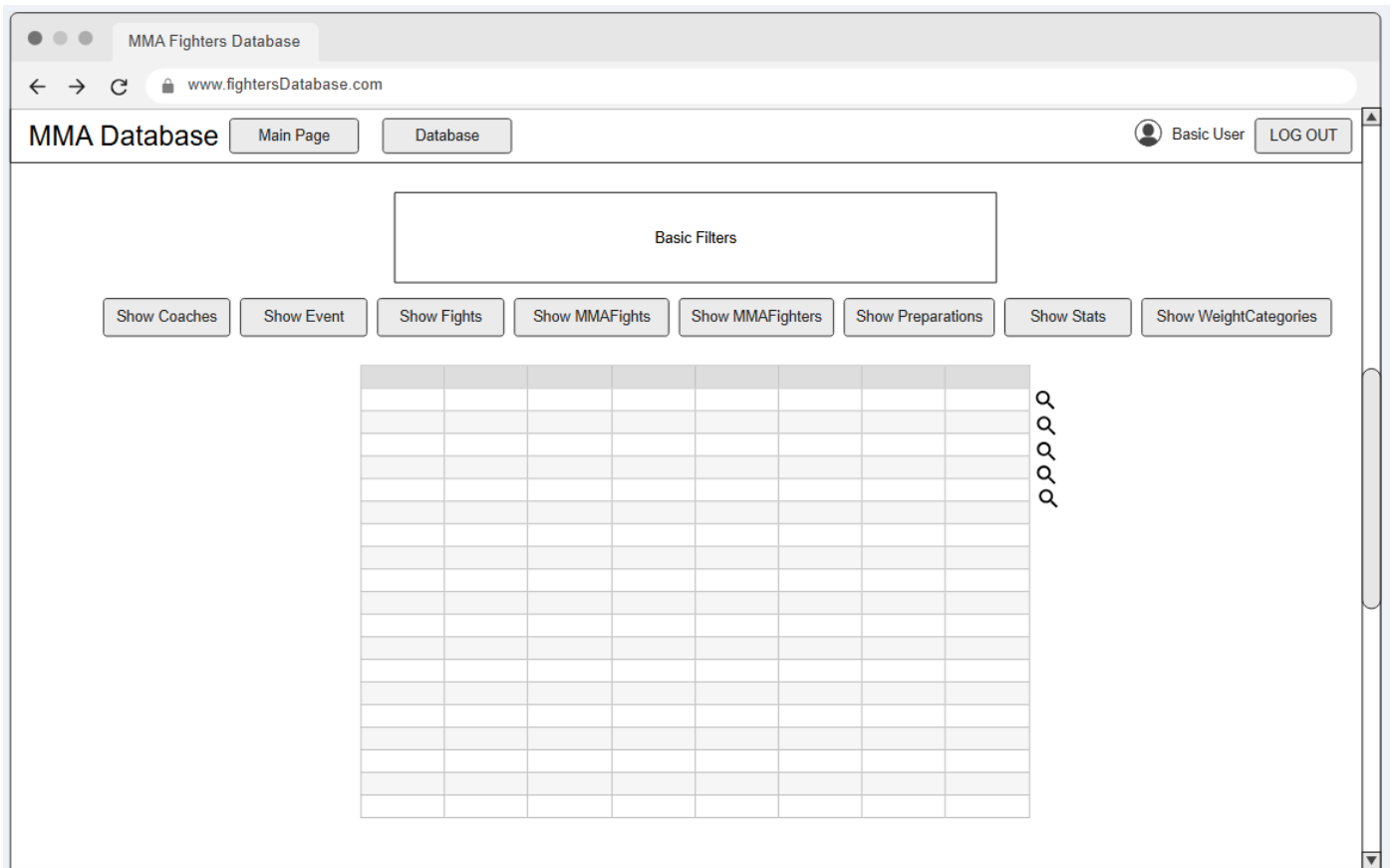
- **Čtení dat z tabulek** → Nízká náročnost.
- **CRUD funkce** → Střední náročnost.
- **Plánování a potvrzování zápasů** → Vysoká náročnost.

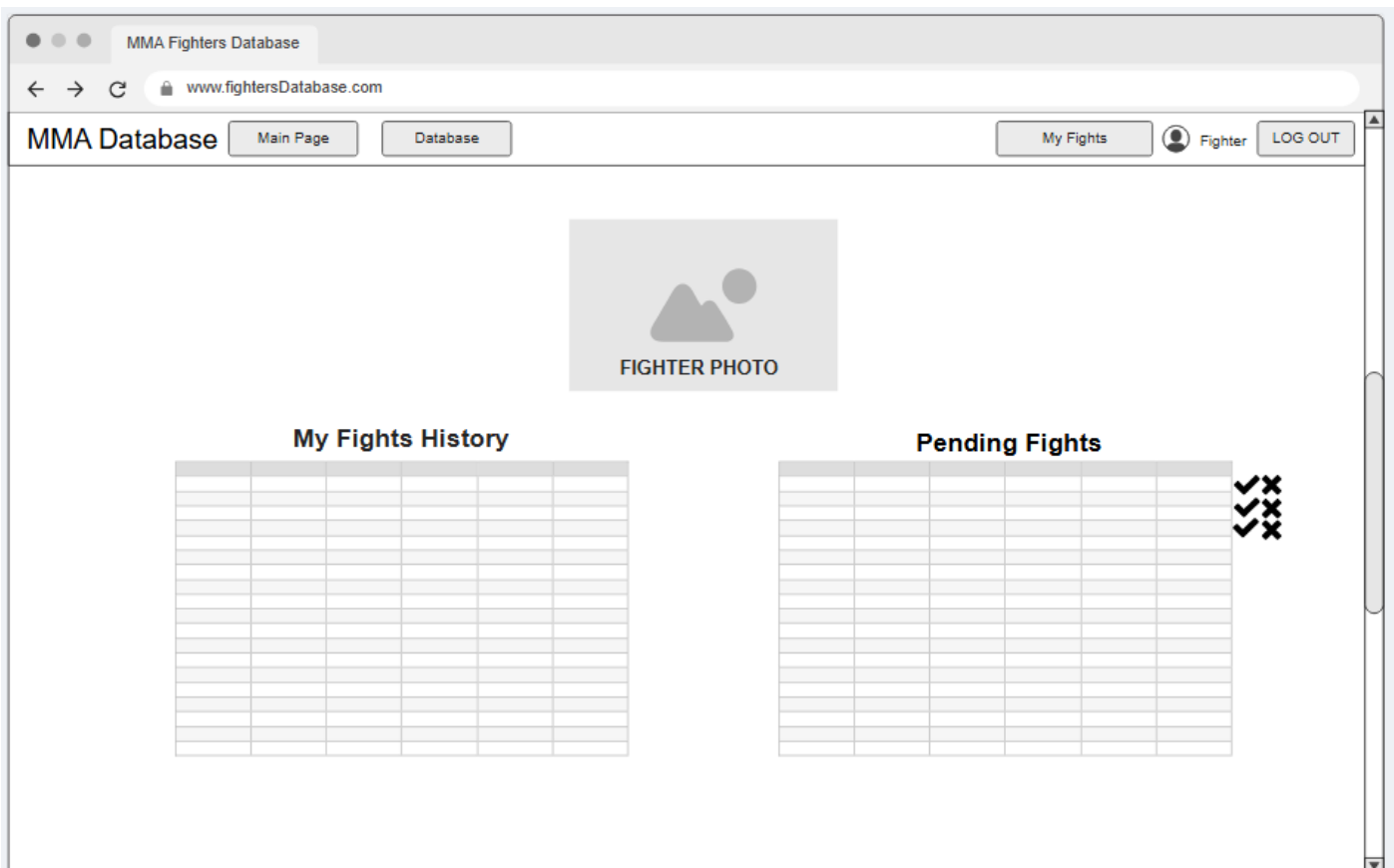
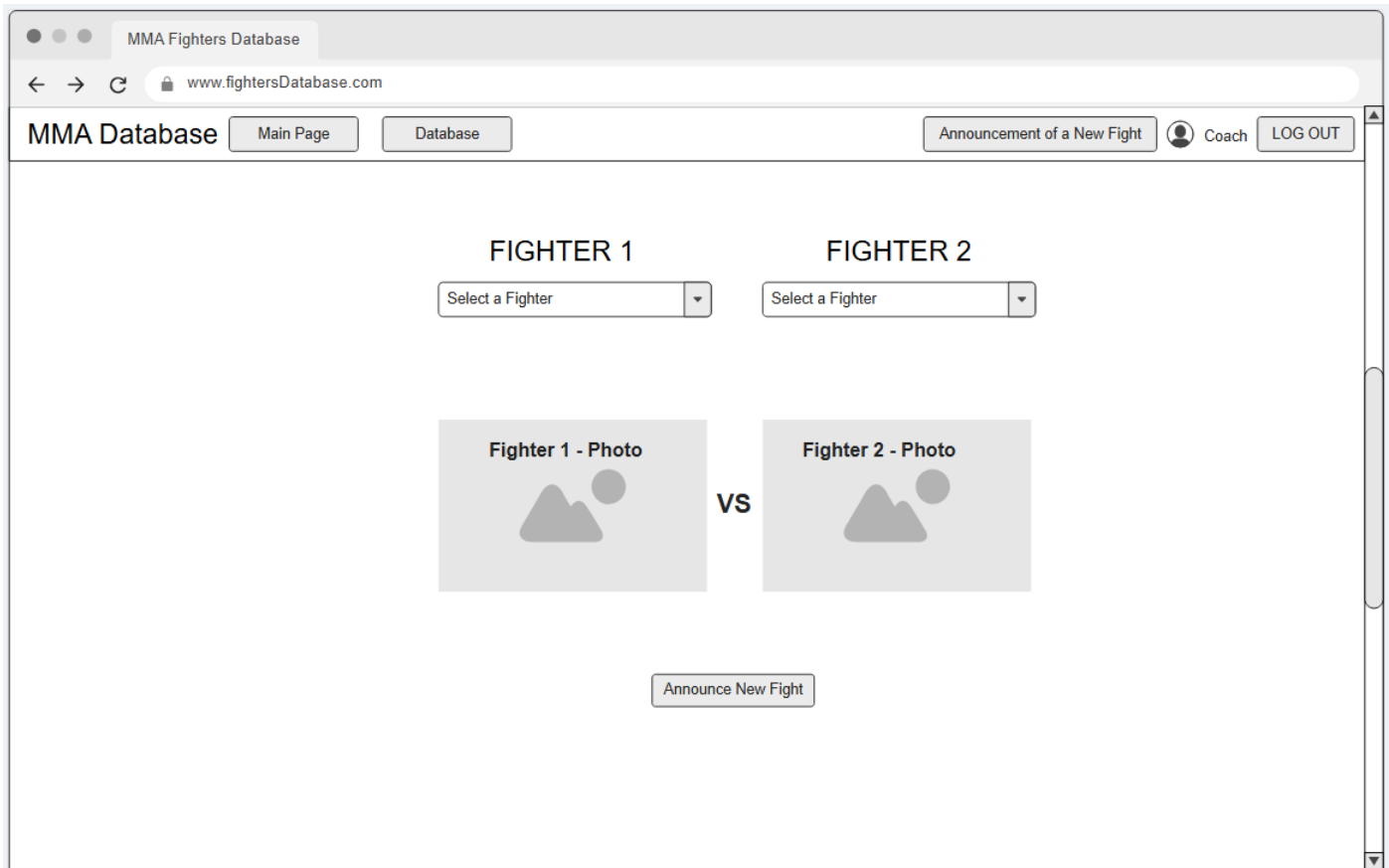
První představa o rozložení systému a volba platformem:

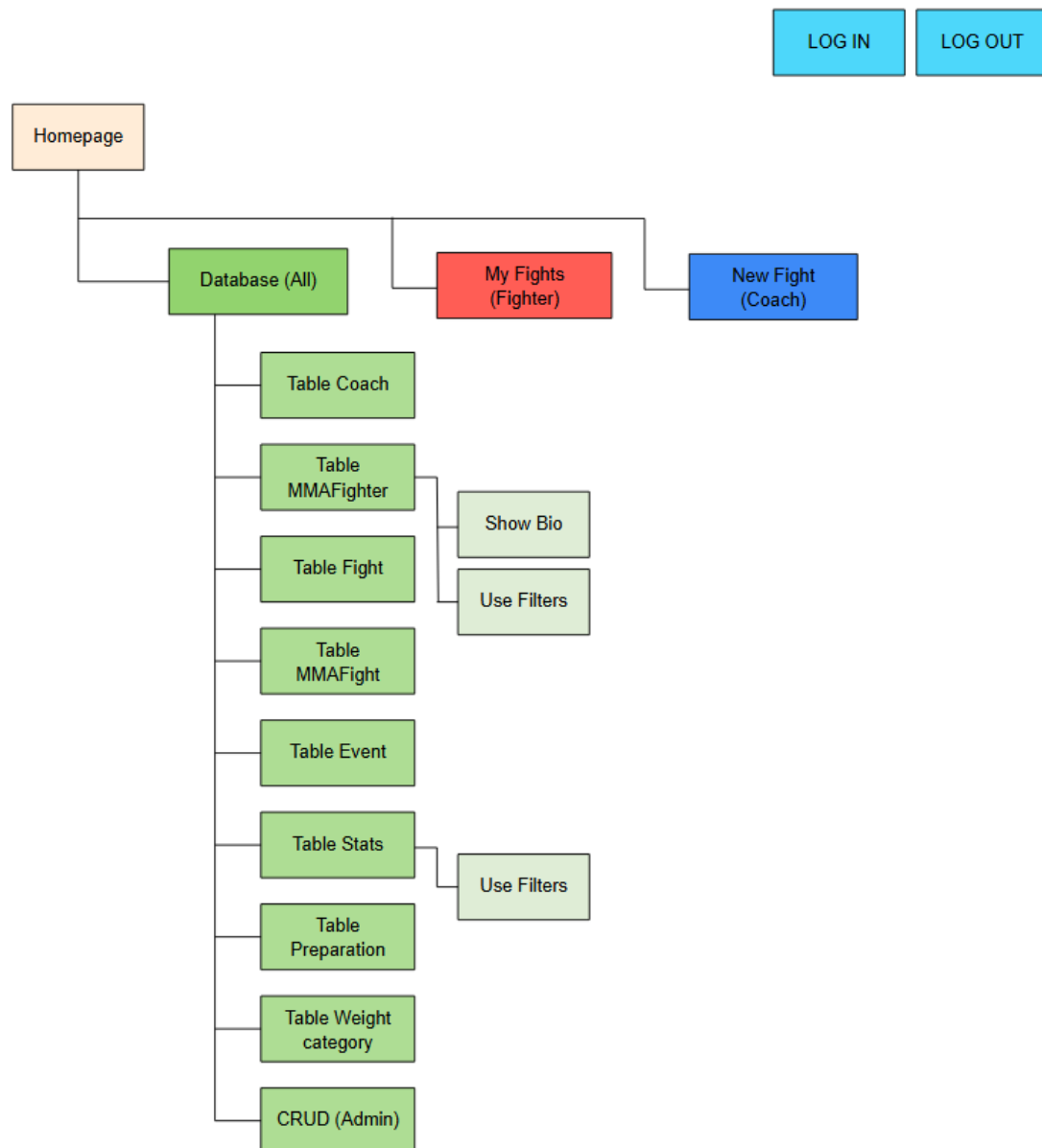
Technologie:

- **Backend** → Java (Spring).
- **Frontend** → JS, CSS (Bootstrap), HTML.
- **Databáze** → MySQL.
- **Platformy** → Web jako primární platforma, cíleno na mobilní i desktopová zařízení.

Wireframes and SiteNav



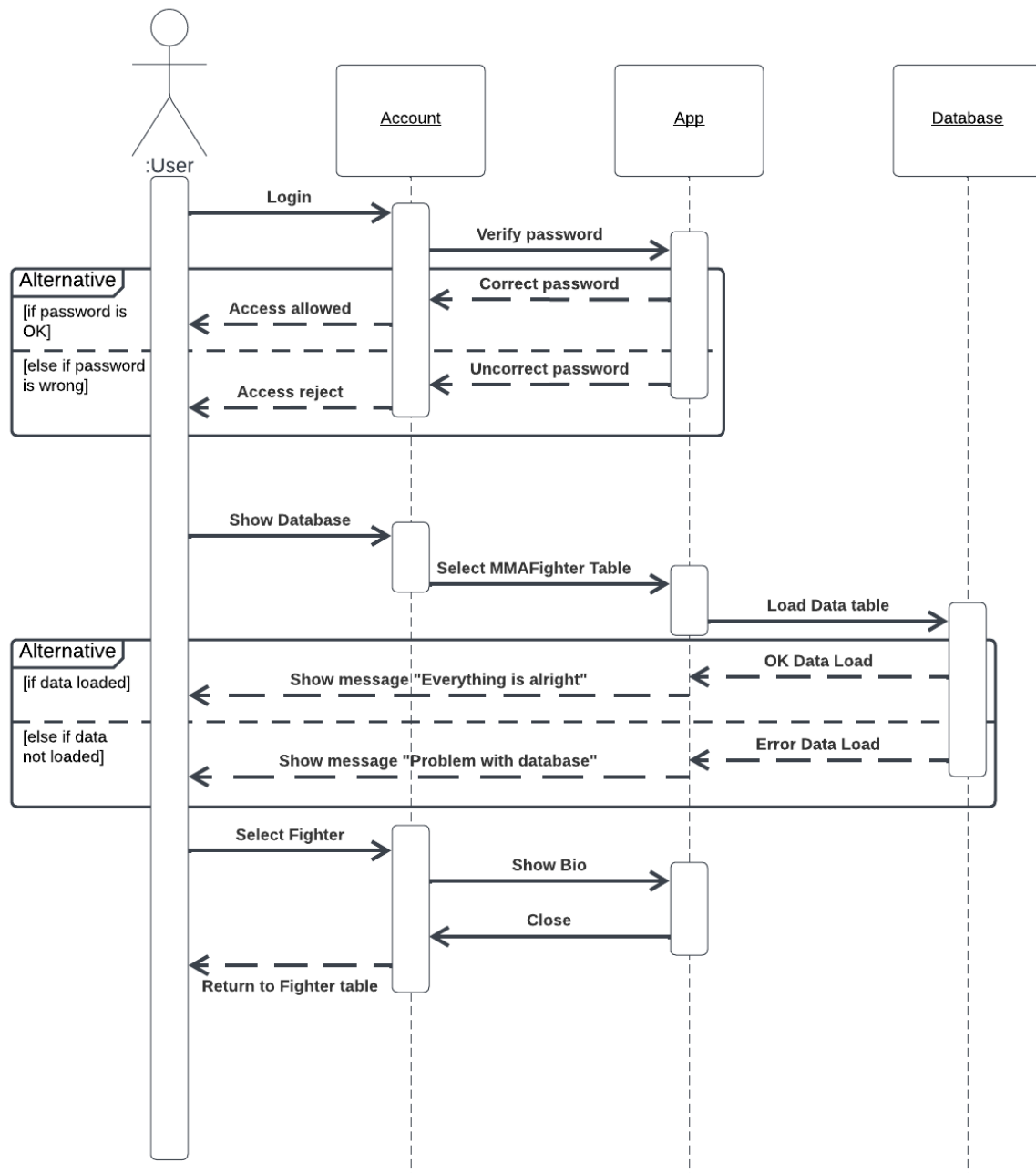




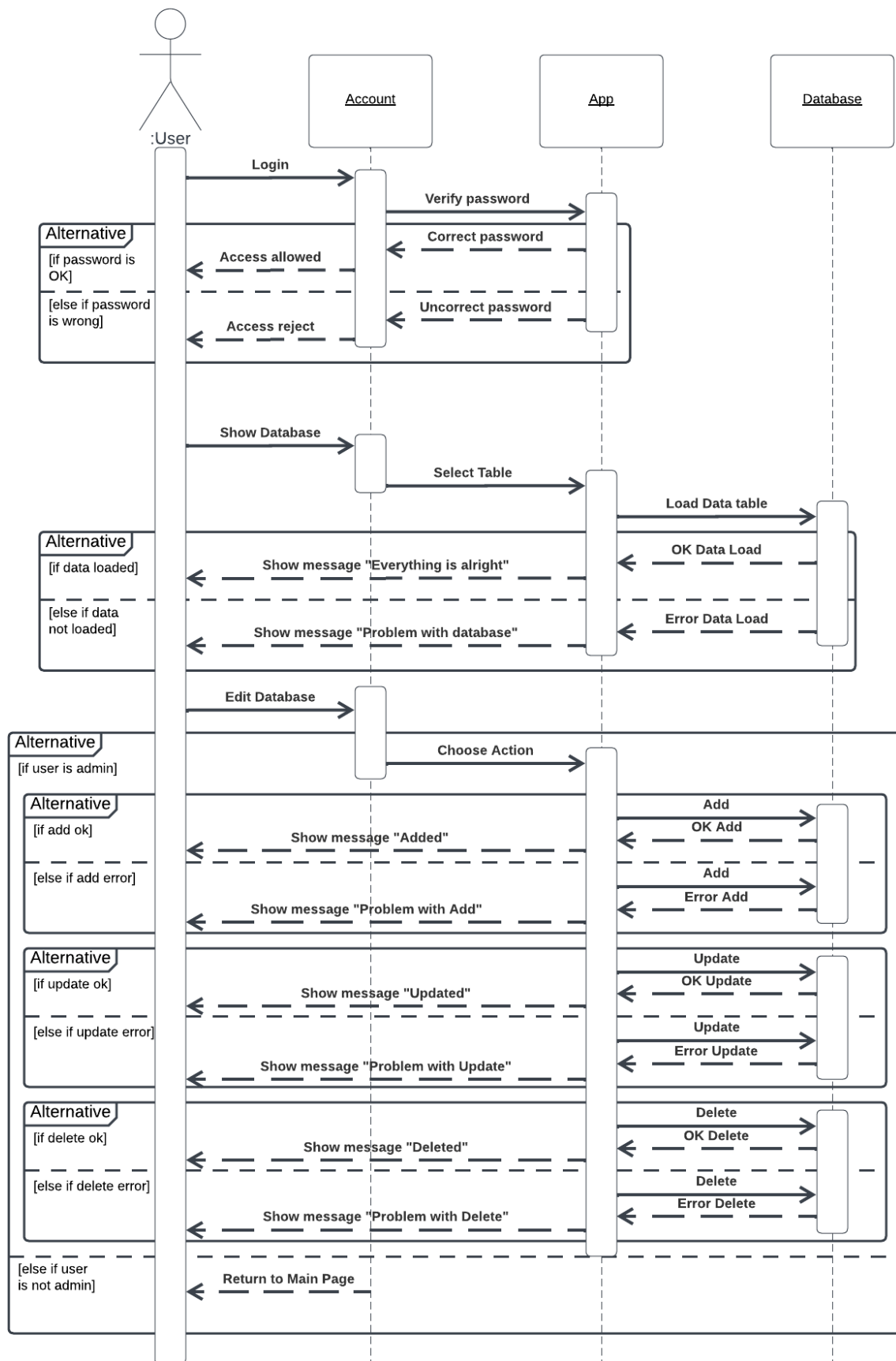
Class diagram, Sequence diagram and Design patterns

Sequence diagram

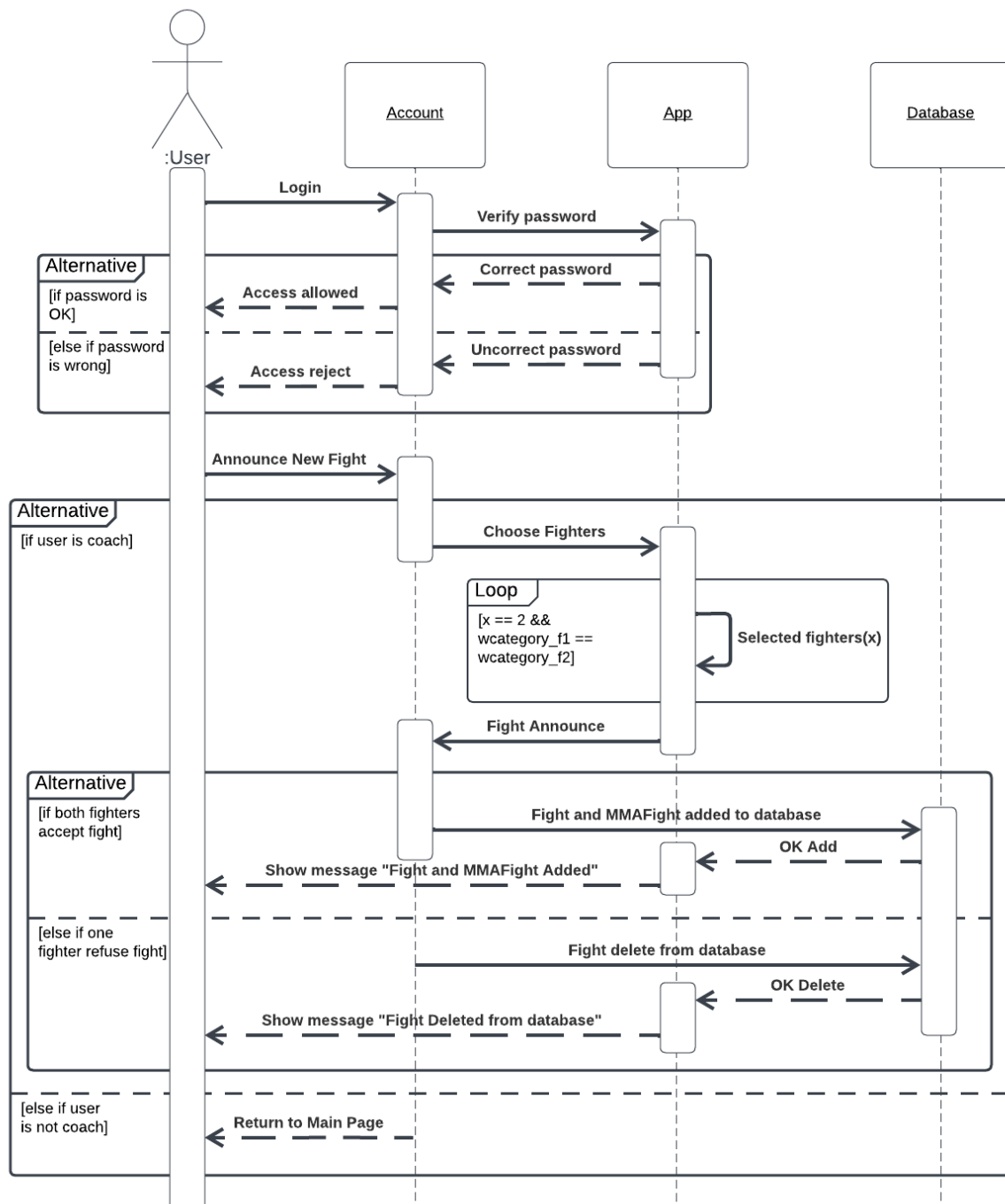
#1 získání BIA fightera



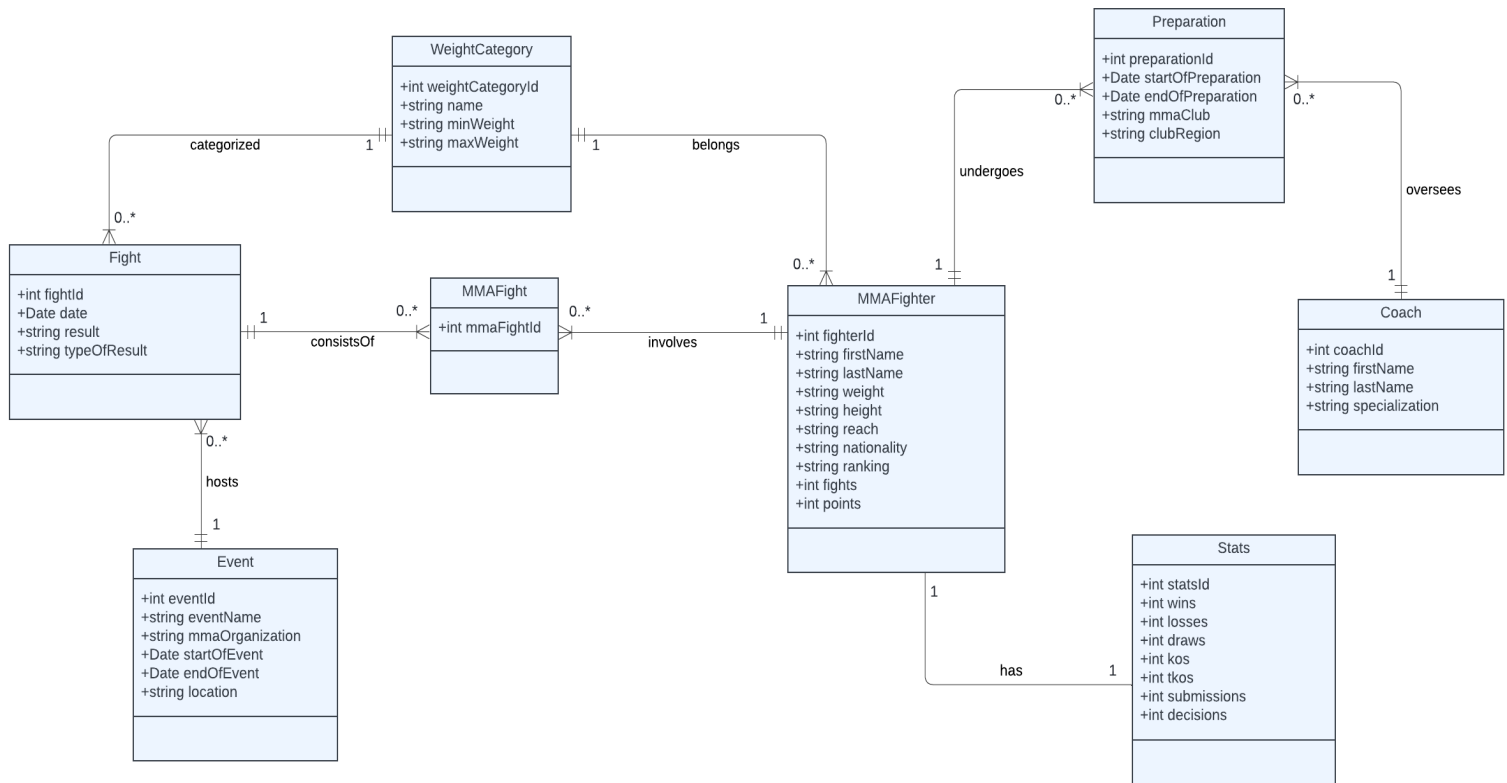
#2 CRUD funkce v databázi



#3 CRUD funkce v databázi



Class diagram



Design patterns

Repository Pattern
 Service Layer Pattern
 Controller Pattern (MVC)
 Factory Method Pattern
 Observer Pattern

Architecture and Diagram Component

Architecture

Frontend

- **HTML/CSS/JavaScript** – Implementace uživatelského rozhraní, vykreslování UI, logika na straně klienta.

Backend

- **Models** – Entitní třídy reprezentující databázové tabulky.
- **Controllers** – Vstupní body aplikace pro komunikaci s Frontendem.
- **Repositories** – Přístup k databázi prostřednictvím SQL dotazů.
- **Services** – Obsahují aplikační business logiku, volají repository a integrují data.

Database

- **MYSQL** – Úložiště perzistentních dat, tabulky odpovídají modelům.

Component connection

- Frontend ↔ Backend: Komunikace přes controllery.
- Backend ↔ Databáze: Komunikace skrze repository vrstvy, které pracují s modely neboli entitami.

Diagram Component

