

ZÁVĚREČNÁ STUDIJNÍ PRÁCE dokumentace

Webová aplikace pro správu fotbalového týmu



Autor: Jakub Šoltys
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2025/26

Poděkování

Chtěl bych poděkovat panu učiteli Mgr. Marku Lučnému za cenné rady a pomoc s projektem a Ing. Petru Grussmannovi za objektivní názory.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 5. 1. 2026

.....
Podpis autora

Abstrakt

Tato práce se věnuje vytvoření webové aplikace SQUADRA, která slouží jako komplexní nástroj pro správu fotbalových týmů. Hlavním cílem bylo vyvinout platformu, která umožní trenérům efektivně organizovat sportovní činnost a hráčům poskytnout okamžitý přístup ke všem důležitým informacím. Aplikace nabízí širokou škálu funkcí pro každodenní chod klubu, počínaje plnou kontrolou nad plánováním tréninků a zápasů, které může trenér v reálném čase vytvářet, upravovat nebo mazat. Hráči vidí naplánované akce ihned po jejich zveřejnění a mohou jednoduše hlasovat o své účasti, díky čemuž má trenér okamžitý přehled o aktuální docházce. Systém dále umožňuje detailní zápis výsledků proběhlých utkání včetně evidence vstřelených gólů a udělených karet. Veškerá tato data se automaticky propisují do celkových statistik, takže hráči i trenéři mohou sledovat dlouhodobý vývoj výkonnosti jednotlivců i celého týmu v přehledných tabulkách. Výsledkem je intuitivní a responzivní systém, který digitalizuje administrativu spojenou s vedením fotbalového týmu a umožňuje všem členům sledovat sportovní pokroky na jednom místě.

Klíčová slova

Správa fotbalového týmu, tréninky, zápasy, docházka, statistiky, webová aplikace

Abstract

This thesis focuses on the development of a web application called SQUADRA, which serves as a comprehensive tool for managing football teams. The primary objective was to develop a platform that enables coaches to efficiently organize sporting activities while providing players with immediate access to all essential information. The application offers a wide range of features for the day-to-day operation of a club, starting with full control over the planning of practices and matches, which the coach can create, edit, or delete in real time. Players can view scheduled events immediately after they are published and can easily vote on their attendance, giving the coach an instant overview of current availability. Furthermore, the system allows for detailed recording of match results, including goal scorers and disciplinary cards. All such data is automatically processed into comprehensive statistics, allowing both players and coaches to track the long-term performance development of individuals and the entire team through clear tables. The result is an intuitive and responsive system that digitizes the administration associated with managing a football team and allows all members to monitor sporting progress in one place.

Keywords

Football team management, practice, matches, attendance, statistics, web application

Obsah

Úvod	2
1 Teoretická východiska a analýza projektu	3
1.1 Účel aplikace	3
1.2 Architektura databáze	3
1.3 Počáteční zkušenosti	4
2 Využité technologie	5
2.1 Django framework	5
2.2 PostgreSQL	5
2.3 Tailwind CSS	5
2.4 Charts.js	5
2.5 Font Awesome	6
2.6 Django Authentication System	6
2.7 Python	6
2.8 JavaScript	6
2.9 Docker	6
3 Pracovní postupy	7
3.1 Založení projektu	7
3.2 Databázová struktura	8
3.3 Tvorba aplikační logiky a šablon	9
3.4 Vizuální zpracování a stylování	9
3.5 Zpracování formulářů a uživatelských vstupů	10
3.6 Autentizace a autorizace uživatelů	11
4 Funkční procesy aplikace	13
4.1 Dashboard	13
4.2 Nastavení a úpravy účtu	13
4.3 Statistiky	13
4.4 Vytváření zápasů a tréninků	14
4.5 Hlasování o účasti	14
4.6 Zobrazení docházky	15
4.7 Zapisování výsledku zápasu	15
4.8 Trenér a více týmů	16
5 Shrnutí a výsledky práce	17

5.1	Funkce aplikace	17
5.2	Splněné a nesplněné cíle	18

ÚVOD

V současném profesionálním sportu již nestačí pouze tvrdě trénovat; klíčem k maximálním výsledkům je precizní analýza a systematické plánování. Trenérství je disciplína založená na datech, kde správné rozvržení tréninků a zápasů rozhoduje o tom, zda sportovec dosáhne svého vrcholu. Zatímco dříve se používaly zprávy přes různé aplikace, dnešní doba vyžaduje lepší nástroje, které dokážou surová data přeměnit v přehledné a srozumitelné informace.

Cílem mé maturitní práce bylo vyvinout komplexní webovou aplikaci, která fotbalistům a trenérům tento proces usnadní. Systém umožňuje nejen bezpečný management uživatelských účtů, ale především přehlednou evidenci tréninků a zápasů. Klíčovými prvky aplikace je přehled historie aktivit a sledování statistik i výkonnosti, které se vizualizují i pomocí dynamických grafů. Kladl jsem velký důraz na plnou responzivitu, aby byla aplikace plnohodnotně použitelná jak na počítači, tak na mobilních zařízeních.

V rámci této dokumentace podrobně popisují celý proces vývoje. Začínám architekturou databázového modelu, pokračuji přes implementaci uživatelské logiky a formulářů až po tvorbu vizualizačních nástrojů. Závěrečná část je věnována frontendovému řešení, stylování a zajištění adaptivity pro různé typy obrazovek.

1 TEORETICKÁ VÝCHODISKA A ANALÝZA PROJEKTU

1.1 ÚČEL APLIKACE

V současné době existuje na trhu celá řada pokročilých platform zaměřených na správu fotbalových klubů. Většina těchto profesionálních řešení je však primárně cílena na velké organizace a prvoligové týmy, což se často odráží v jejich vysoké pořizovací ceně, náročné administraci nebo přílišné robustnosti, která může být pro menší týmy náročná. Hlavním záměrem mého projektu bylo tuto bariéru prolomit a vytvořit aplikaci, která bude univerzálně dostupná pro každou úroveň fotbalu. Chtěl jsem nabídnout funkční nástroj, který najde své uplatnění jak v profesionálnějších týmech, tak i v nižších regionálních soutěžích, kde je kladen důraz především na jednoduchost, přehlednost a rychlou evidenci tréninkového procesu bez nutnosti složitého školení personálu.

1.2 ARCHITEKTURA DATABÁZE

Moderní webové aplikace jsou stavěny jako vícevrstvé systémy, které zajišťují efektivní komunikaci mezi uživatelem a daty. Moje aplikace využívá architekturu Client-Server, kde na straně klienta stojí webový prohlížeč a na straně serveru běží aplikační logika propojená s databází. Systém je postaven na návrhovém vzoru MVT (Model-View-Template), což je specifická variace rozšířeného vzoru MVC (Model-View-Controller). Tento princip funguje na rozdělení aplikace do tří spolupracujících vrstev:

- Model (Datová vrstva): Tato část definuje strukturu dat. Každý model v kódu odpovídá tabulce v databázi. Díky technologii ORM aplikace komunikuje s databází pomocí objektů.
- View (Logická vrstva): Přijímá požadavky od uživatele, rozhoduje, jaká data je potřeba vytáhnout z Modelu, provádí nad nimi výpočty a výsledek posílá dál k zobrazení.
- Template (Prezentační vrstva): Jedná se o šablony psané v HTML doplněné o speciální značky, které umožňují dynamicky zobrazovat data poslaná z View. Tato vrstva se stará o to, aby uživatel viděl přehledné prostředí.

1.3 POČÁTEČNÍ ZKUŠENOSTI

V tomto projektu vycházím z praktických dovedností, které jsem se naučil v průběhu studia. Již ve třetím ročníku jsem pracoval na vývoji jednoduché webové aplikace postavené na čistém frameworku Django. Tato předchozí zkušenost mi umožnila pochopit základní principy fungování MVT architektury, manipulaci s databázovými modely a základy šablonovacího systému.

Díky znalosti těchto základům jsem se mohl při tvorbě maturitního projektu soustředit na komplexnější výzvy, jako je logika docházkového systému, vizualizace výkonnostních dat pomocí grafů nebo funkčnost účtů uživatelů. Počáteční znalost frameworku Django mi tak dala náskok, který byl ve vývoji této aplikace nezbytný.

2 VYUŽITÉ TECHNOLOGIE

2.1 DJANGO

Hlavním technologickým pilířem aplikace je Django, vysokoúrovňový webový framework napsaný v jazyce Python. Django využívá architekturu MVT a disponuje výkonným nástrojem ORM, který umožňuje efektivní správu databáze pomocí Python objektů bez nutnosti přímého psaní SQL dotazů.

2.2 POSTGRESQL

Jako databázový systém jsem zvolil PostgreSQL, což je pokročilá open-source objektově-relační databáze. Tento systém disponuje velkou spolehlivostí, integritou dat a schopností efektivně zpracovávat komplexní dotazy nad rozsáhlými soubory dat. PostgreSQL plně podporuje standardy SQL a v kombinaci s frameworkm Django nabízí vynikající výkon při správě relačních vazeb mezi uživateli, týmy a jejich statistikami. Pro projekt tohoto typu dostačující.

2.3 TAILWIND CSS

Pro stylování uživatelského rozhraní jsem využil moderní framework Tailwind CSS. Na rozdíl od tradičních frameworků s předpřipravenými komponentami mi Tailwind umožnil tvořit design přímo v HTML kódu pomocí nízkoúrovňových tříd. Tento přístup výrazně urychlil vývoj responzivního vzhledu, zajistil vysokou konzistence grafických prvků.

2.4 CHARTS.JS

Pro generování grafů jsem použil Chart.js, což je populární JavaScriptová knihovna určená pro tvorbu responzivních grafů. Tato technologie mi umožnila transformovat data z databáze (např. počet gólů na zápas) do přehledné grafické podoby. Knihovna využívá element HTML5 Canvas, díky čemuž je vykreslování grafů rychlé a plynulé i na mobilních zařízeních, přičemž nabízí širokou škálu prvků, jako jsou sloupcové, koláčové nebo spojnicové grafy.

2.5 FONT AWESOME

Pro zvýšení přehlednosti a vylepšení vzhledu jsem použil knihovnu Font Awesome. Poskytuje sadu vektorových ikon, které v aplikaci slouží jako jednoduché symboly pro tlačítka a nadpisů. Použití SVG formátu zajišťuje ostrost ikon při jakémkoliv rozlišení a minimální dopad na rychlosť načítání stránek.

2.6 DJANGO AUTHENTICATION SYSTEM

Pro správu uživatelů využívám vestavěný autentizační systém frameworku Django. Tento modul řeší kompletní proces přihlašování, odhlášení a bezpečné ukládání uživatelských dat. Klíčovým prvkem je automatické hashování hesel pomocí algoritmu PBKDF2, který zajišťuje, že citlivé údaje nejsou v databázi čitelné ani v případě jejího úniku.

2.7 PYTHON

Celá aplikace je postavena na programovacím jazyce Python, který je oceňován pro svou vysokou čitelnost, čistou syntaxi a efektivitu vývoje. V projektu slouží jako základní kámen pro veškerou logiku na straně serveru. Python jsem zvolil především kvůli mé znalosti tohoto jazyka z mého studia a rozsáhlému ekosystému knihoven, což mi umožnilo snadno používat pokročilé funkce, jako jsou třeba automatické výpočty statistik docházky.

2.8 JAVASCRIPT

Jazyk JavaScript tvoří interaktivní vrstvu aplikace na straně klienta. V mém projektu slouží především k dynamickému ovládání prvků uživatelského rozhraní bez nutnosti znova načítání celé stránky a k zajištění funkčnosti zapisování zápasů. JavaScript také využívám pro inicializaci a konfiguraci grafů v knihovně Chart.js.

2.9 DOCKER

Pro zajištění konzistentního vývojového a produkčního prostředí jsem využil platformu Docker. Ta umožňuje izolovat aplikaci a její závislosti do kontejnerů, což zajišťuje, že systém běží identicky na jakémkoliv zařízení.

3 PRACOVNÍ POSTUPY

3.1 ZALOŽENÍ PROJEKTU

Prvním krokem realizace bylo vytvoření projektové struktury a nastavení izolovaného vývojového prostředí. V nově vytvořené složce projektu jsem inicializoval virtuální prostředí, což je standardní postup pro zajištění nezávislosti knihoven projektu na globálním nastavení systému. Následovala instalace jazyka Python a frameworku Django pomocí správce balíčků *pip*.

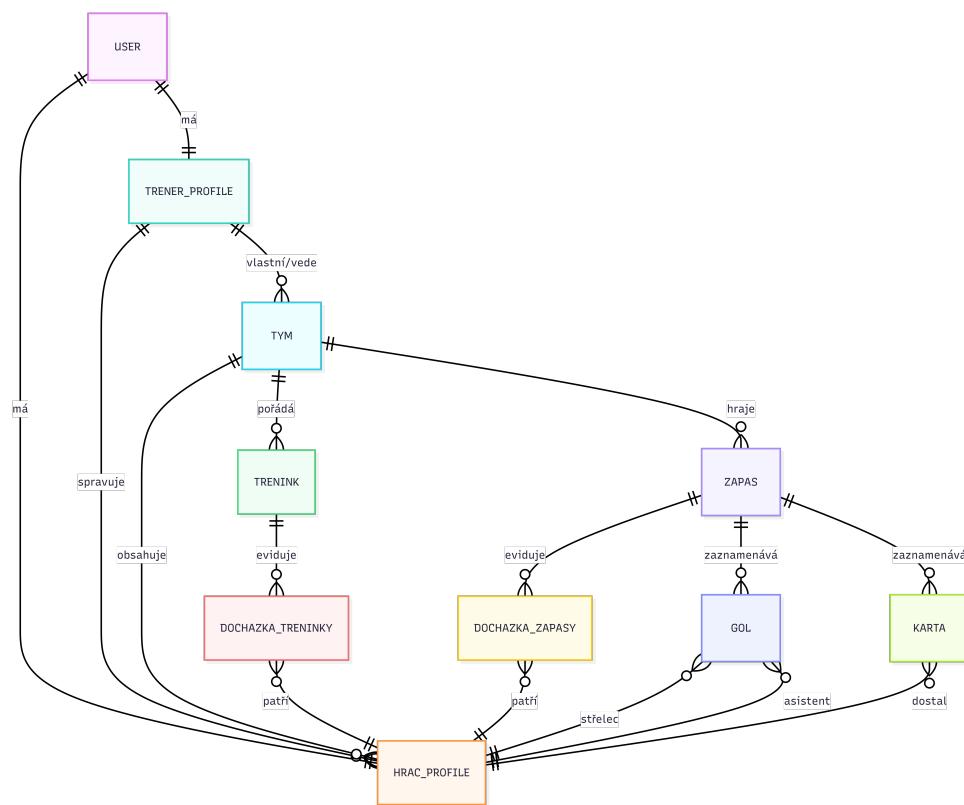
Poté následovala základní konfigurace frameworku, která zahrnovala:

- Inicializace projektu: Pomocí příkazu *django-admin startproject* byla vytvořena základní kostra aplikace.
- Vytvoření superuživatele: Skrze příkaz *createsuperuser* jsem založil administrátorský účet pro přístup do interní správy systému.
- Zprovoznění hlavní stránky: Vytvořil jsem první pohled ve *views.py* a v šabloně *index.html* jsem definoval domovskou stránku aplikace.

Vzhledem k tomu, že jsem se se všemi těmito postupy podrobně seznámil již v předchozích ročnících, proběhla tato fáze bez jakýchkoli potíží.

3.2 DATABÁZOVÁ STRUKTURA

Návrh databáze je srdcem celého projektu. K její realizaci jsem využil relační model, který je v rámci frameworku Django definován přímo v souborech `models.py`.



Obrázek 3.1: ER diagram databázové struktury

Jakmile byly definovány všechny třídy a vztahy v souboru `models.py`, využil jsem vestavěný systém migrací. Proces proběhl ve dvou krocích: nejprve příkazem `makemigrations`, který analyzoval změny v kódu a vytvořil instrukční soubory, a následně příkazem `migrate`. Tímto krokem se nadefinovaná struktura fyzicky propsalá do databáze PostgreSQL. Výsledkem je plně funkční a optimalizovaná databázová struktura, která je připravena na ukládání reálných dat aplikace.

3.3 TVORBA APLIKAČNÍ LOGIKY A ŠABLON

V této fázi jsem propojil databázový model s uživatelským rozhraním. Využil jsem k tomu architekturu MVT, kde hlavní logiku zpracovávají pohledy v souboru *views.py*.

3.3.1 Aplikační logika

V této části jsem se zaměřil na vývoj backendové logiky, která slouží jako spojovací článek mezi databází a uživatelským rozhraním. Pro obsluhu požadavků jsem využil kombinaci funkcionálních pohledů a tříd. Kladl jsem důraz na přehlednost a bezpečnost kódu.

Klíčové prvky, které jsem v rámci *views.py* implementoval:

- Distribuce dat: Naprogramoval jsem logiku, která efektivně získává potřebné informace z databáze a předává je ve formě kontextu do šablon k vykreslení.
- Zpracování dat: Naprogramoval jsem funkce pro výpočet statistik, jako je procentuální úspěšnost docházky nebo tabulky střelců.
- Dynamické dotazy: Využil jsem Django ORM pro efektivní filtrování dat, například zobrazení pouze nadcházejících tréninků.

3.3.2 Šablony

Pro uživatelské rozhraní jsem vytvořil systém HTML šablon. Využil jsem Django Template Language, který mi umožnil vkládat logiku (cykly, podmínky) přímo do HTML kódu.

- Základní šablona (*base.html*): Obsahuje společné prvky. Ostatní stránky z ní dědí pomocí bloku `{% extends %}`, což zajišťuje jednotný vzhled a snadnou údržbu. Stejně tak i *navbar.html* nebo *footer.html*
- Práce s daty: Například pomocí cyklů `{% for %}` dynamicky generuje tabulky s výsledky zápasů nebo seznamy hráčů na základě dat dodaných z *views.py*.

3.4 VIZUÁLNÍ ZPRACOVÁNÍ A STYLOVÁNÍ

Pro dosažení moderního vzhledu a plné responzivity aplikace jsem zvolil framework Tailwind CSS. Na rozdíl od klasického psaní CSS pravidel jsem využíval utility třídy přímo v HTML šablonách, což výrazně urychlilo proces stylování a zajistilo konzistenci napříč celým systémem.

3.5 ZPRACOVÁNÍ FORMULÁŘŮ A UŽIVATELSKÝCH VSTUPŮ

Formuláře jsou klíčovým prvkem pro interakci uživatele se systémem, ať už jde o přihlášení, zadávání výsledků zápasů nebo vytváření nových tréninků. Pro jejich realizaci jsem využil vlastní systém Django Forms.

- Automatická validace: Django samo kontroluje, zda jsou zadaná data ve správném formátu (např. platné datum, číselné hodnoty u gólů), což předchází chybám v databázi.
- Bezpečnost: Každý formulář obsahuje CSRF token, který chrání aplikaci před útoky typu Cross-Site Request Forgery.
- Propojení s modely: Pro většinu funkcí jsem použil ModelForm, který automaticky generuje pole formuláře podle definice v databázi, což mi ušetřilo čas a eliminovalo duplicitu kódu.

Všechny formuláře se vytvářejí v souboru *forms.py*.

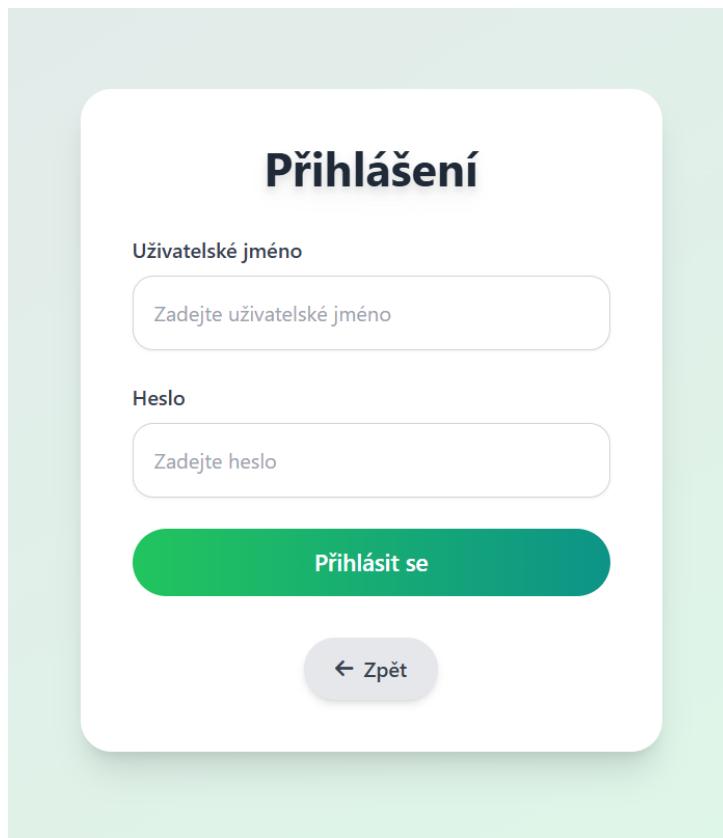
3.6 AUTENTIZACE A AUTORIZACE UŘIVATELŮ

Zajištění bezpečnosti dat a správné rozdělení uživatelských rolí je prioritou systému. Pro tento účel jsem využil vestavěný autentizační systém frameworku Django, který jsem přizpůsobil potřebám interního loginu sportovního klubu.

3.6.1 Proces autentizace

Systém nevyužívá registraci třetích stran; účty vytváří výhradně administrátor v interním rozhraní. Uživatelé se přihlašují pomocí přiděleného uživatelského jména a hesla.

- Přihlašovací cyklus: Funkce *login_view* zpracovává požadavky typu POST, validuje údaje pomocí LoginForm a skrze metodu *authenticate()* ověřuje identitu uživatele proti databázi. Po úspěšném ověření funkce *login()* vytvoří session a uloží uživatele do objektu *request.user*.



Obrázek 3.2: Přihlášení

- První přihlášení: Implementoval jsem speciální logiku pro nově vytvořené účty. Pokud je u uživatele příznak *first_login=True*, je automaticky přesměrován na pohled *first_login_view*.

Zde musí uživatel doplnit své osobní údaje do profilu, čímž se příznak změní na False a účet se aktivuje.

- Odhlášení: Proces *logout_view* bezpečně ukončí session a přesměruje uživatele na úvodní stránku.

3.6.2 Řízení přístupu a autorizace

Autorizace určuje, k jakým datům a funkcím má přihlášený uživatel oprávnění. V projektu využívám model RBAC (Role-Based Access Control) rozdelený do tří úrovní:

- Admin: Má plný přístup k Django Admin rozhraní, kde spravuje, vytváří a maže uživatelské účty.
- Trenér: Má přístup k trenérskému dashboardu. Může spravovat pouze své týmy, vytvářet tréninky, zápasy a prohlížet statistiky svých hráčů.
- Hráč: Má přístup k hráčskému dashboardu, kde vidí informace o svém týmu, vlastní statistiky a může hlasovat o účasti na událostech.

Pro zabezpečení jednotlivých funkcí v kódu využívám:

- Dekorátor *@login_required*: Zajišťuje, že daná stránka je přístupná pouze přihlášeným uživatelům.
- Kontrola vlastnictví zdroje: V kritických pohledech (např. editace tréninku) využívám metodu *get_object_or_404*, která ověřuje, zda editovaný objekt skutečně patří danému trenérovi. Tím je zabráněno neoprávněné manipulaci s daty jiných týmů.
- Session-based autorizace: Pro trenéry spravující více týmů využívám *request.session* k uložení ID aktuálně vybraného týmu, což zajišťuje persistenci volby napříč různými stránkami aplikace.

3.6.3 Bezpečnostní standardy

Aplikace splňuje moderní standardy zabezpečení webových aplikací:

- Hashování hesel: Hesla nejsou nikdy ukládána v prostém textu, ale jsou hashována pomocí algoritmu PBKDF2.
- Validace hesel: Systém vynucuje minimální délku a složitost hesla.
- Ochrana proti útokům: Django automaticky chrání aplikaci proti CSRF pomocí tokenů ve formulářích.

4 FUNKČNÍ PROCESY APLIKACE

4.1 DASHBOARD

Dashboard představuje klíčový prvek uživatelského rozhraní, který slouží jako dynamický rozcestník a poskytuje uživateli okamžitý přehled o nejdůležitějším dění v týmu ihned po přihlášení. Celý obsah je generován v reálném čase a přizpůsobuje se roli přihlášeného uživatele, přičemž prioritu mají časově nejbližší události a aktuální data.

4.2 NASTAVENÍ A ÚPRAVY ÚČTU

Prostřednictvím ovládacích prvků může uživatel podrobně prohlížet a aktualizovat své profilové údaje, spravovat zabezpečení účtu pomocí změny hesla nebo bezpečně ukončit aktivní relaci. Díky propojení s databázovým modelem se veškeré úpravy okamžitě promítají do celého systému, což zaručuje aktuálnost informací pro trenéry i spoluhráče.

4.3 STATISTIKY

Tato část aplikace slouží k tomu, aby všichni v týmu měli jasnou představu o tom, jak se jim daří. Data se v grafech a tabulkách mění podle toho, kdo je zrovna přihlášený. Hráč po přihlášení vidí hlavně svůj vlastní profil – tedy úspěšnost v zápasech, jakou má docházku na tréninky a kolik dal v sezóně gólů. Kromě toho jsou k dispozici i celkové statistiky týmu, kde je vidět, jak se skupině daří jako celku, například jaká je průměrná účast na trénincích nebo kolik gólů tým průměrně nastřílí za zápas. Trenér má pak nejširší přístup; může si rozkliknout statistiky kteréhokoli svého hráče.

4.4 VYTVAŘENÍ ZÁPASŮ A TRÉNINKŮ

Celý proces organizace začíná u trenéra, který skrze jednoduchý formulář zadá klíčové údaje, jako je datum, čas a místo konání události. Jakmile trenér formulář potvrdí, systém data okamžitě zpracuje a nově vytvořený zápas nebo trénink se v reálném čase zobrazí všem hráčům. Hráči tak mají k informacím okamžitý přístup a mohou rovnou hlasovat o své účasti, zatímco trenér má ihned k dispozici přehlednou listinu pro sledování docházky. Tato automatizace zajišťuje, že se informace k hráčům dostanou bez prodlení a bez nutnosti další komunikace přes jiné kanály.

Obrázek 4.1: Vytváření zápasu

4.5 HLASOVÁNÍ O ÚČASTI

Pro hráče je v systému připraven jednoduchý způsob, jak dát trenérovi vědět, zda na akci dorazí. U každého nadcházejícího tréninku nebo zápasu mají na výběr dvě rychlé možnosti: „Přijdu“ nebo „Nepřijdu“. Výběr se do databáze uloží okamžitě po kliknutí, bez nutnosti cokoli dalšího potvrzovat. Pokud si to hráč rozmyslí, může svou volbu kdykoliv změnit nebo ji úplně zrušit, čímž se záznam z databáze opět vymaže. Trenér díky tomu v reálném čase vidí aktuální soupisku pro daný den.

4.6 ZOBRAZENÍ DOCHÁZKY

Pro trenéra je klíčová stránka s detailem docházky, kde má k dispozici kompletní seznam všech svých hráčů. U každého jména přehledně vidí aktuální stav: zda hráč účast potvrdil, omluvil se, nebo zatím vůbec nehlásoval. Tato tabulka funguje dynamicky a automaticky se aktualizuje podle toho, jak hráči postupně odesílají své odpovědi. Trenér tak nemusí ručně nic sčítat, protože má na jednom místě okamžitý a přesný přehled o tom, v jakém složení se tým na zápas nebo trénink sejde.

4.7 ZAPISOVÁNÍ VÝSLEDKU ZÁPASU

Jakmile zápas skončí, systém na to trenéra automaticky upozorní přímo na jeho nástěnce i v celkovém přehledu zápasů nápisem i červeným obtáhnutím bloku zápasu. U daného utkání se objeví tlačítko pro zápis výsledků, které trenéra přenese do detailního formuláře. Zde může ke každému vstřelenému gólu přiřadit konkrétního střelce, asistenta, čas a jeho typ. Podobně lze evidovat i udělené karty, u nichž se zaznamenává jméno hráče, minuta a typ karty. U soupeře se pro zjednodušení zadává pouze konečný počet vstřelených branek.

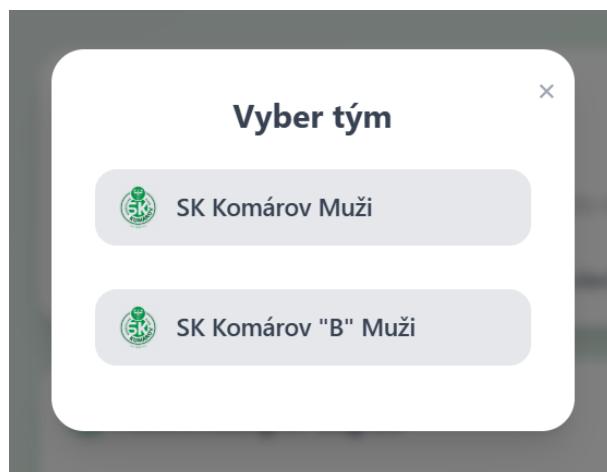
Po potvrzení se všechna data ihned uloží do databáze a výsledek zápasu se okamžitě zobrazí všem hráčům i trenérovi. Velkou výhodou je, že systém v tento moment automaticky přepočítá veškeré individuální i týmové statistiky. Góly, asistence i karty se tak ihned promítnou do profilů hráčů a tabulek úspěšnosti, aniž by bylo nutné cokoli dalšího ručně zpracovávat.

Výsledky vašeho týmu		Výsledky soupeře	
Góly		3	
1. 41' Matěj Binar	<button>Smažat</button>		
2. 43' Patrik Maculík	<button>Smažat</button>		
3. 51' Vojtěch Pavlik	<button>Smažat</button>		
<input type="button" value="Přidat"/> <input type="button" value="Zrušit"/>		Normální	74
Žluté a červené karty <input type="button" value="+ Přidat kartu"/>			
<input type="button" value="Uložit výsledek"/>			

Obrázek 4.2: Zapisování výsledku

4.8 TRENÉR A VÍCE TÝMŮ

Pro trenéry, kteří vedou více než jednu věkovou kategorii nebo skupinu, obsahuje aplikace funkci pro rychlé přepínání mezi týmy. V horní části rozhraní si může trenér jednoduše zvolit, který ze svých týmů chce aktuálně spravovat. Po výběru se celý systém – včetně dashboardu, zápasů a statistik – okamžitě přizpůsobí a zobrazí data příslušná pouze pro daný tým. Tato funkce zajišťuje přehlednost a zabraňuje záměně informací mezi různými skupinami hráčů.



Obrázek 4.3: Výběr týmu

5 SHRNUTÍ A VÝSLEDKY PRÁCE

5.1 FUNKCE APLIKACE

Funkce aplikace začíná u manažera týmu (administrátora). Ten má výhradní přístup do administračního rozhraní, kde zakládá profily pro všechny své hráče i trenéry a definuje strukturu klubu vytvořením jednotlivých týmů podle věkových kategorií. Prozatím pouze v administrátorském prostředí.

Jakmile manažer účty vytvoří, uživatelé se do systému přihlašují pomocí přiděleného jména a hesla. Po prvním vstupu si každý člen může v nastavení zvolit vlastní bezpečné heslo a doplnit své osobní údaje. Tím se aktivuje plnohodnotný přístup k funkcím aplikace:

- Pro trenéry: Možnost zakládat a spravovat tréninky či zápasy a sledovat docházku a statistiky svého týmu.
- Pro hráče: Přehled o nadcházejících událostech, jednoduché hlasování o účasti a přehled svých statistik.
- Pro celý tým: Automatické zaznamenávání výsledků, statistik a historie všech aktivit na jednom místě.

5.2 SPLNĚNÉ A NESPLNĚNÉ CÍLE

Hlavním cílem této práce bylo vytvořit funkční webovou aplikaci, která by usnadnila organizaci tréninků, zápasů a sledování výkonů, a to především týmům v nižších soutěžích. Tento záměr se podařilo naplnit. Výsledkem je jednoduchý systém, který pokrývá celý proces od plánování událostí až po automatické generování statistik. Velkým přínosem je plná responzivita rozhraní, díky které mohou uživatelé s aplikací pohodlně pracovat na počítači i na mobilních telefonech přímo na hřišti.

Při zpětném pohledu však vnímám i určité nedostatky, které jsou dány rozsahem projektu. Mezi ty hlavní patří:

- Administrační prostředí: Současná správa uživatelů a týmů probíhá v základním rozhraní, které by si zasloužilo uživatelsky přívětivější zpracování.
- Struktura dat: Chybí pevné rozdělení historie do jednotlivých sezón a systém zatím neumožňuje, aby byl jeden hráč zaregistrován ve více týmech současně.
- Technický dluh: Jako u každého většího projektu vidím prostor pro zlepšení čistoty kódu a jeho optimalizaci.

Téma správy fotbalových klubů nabízí téměř nekonečné možnosti rozšiřování, at' už jde o pokročilé exporty dat, automatické notifikace nebo detailnější analýzu výkonů. Práce na této aplikaci pro mě byla obrovským přínosem. Naučil jsem se pracovat s novými technologiemi a řešit komplexní logické celky. Věřím, že tato verze je pevným základem, na kterém lze v budoucnu dále stavět.

ZÁVĚR

Cílem této práce bylo navrhnout a realizovat webovou aplikaci SQUADRA, která by digitalizovala a zjednodušila každodenní problémy amatérského fotbalového klubu. Výsledkem je funkční platforma, která efektivně propojuje trenéry a hráče skrze moduly správy týmu, docházky a statistik.

Vytvořený systém řeší nejčastější problémy malých klubů – roztríštěnost informací a náročné ruční zpracovávání statistik. Díky responzivnímu designu a okamžité synchronizaci dat s databází získává trenér do rukou nástroj, který mu šetří čas a umožňuje se plně soustředit na sportovní stránku vedení týmu.

Práce na tomto projektu pro mě byla významnou zkušeností. Musel jsem se vypořádat s návrhem komplexní databázové struktury a zajistit, aby bylo uživatelské rozhraní přívětivé i pro technicky méně zdatné uživatele. I když systém nabízí prostor pro další rozšiřování, například vlastní administrátorské prostředí, současná verze představuje stabilní a plně použitelný produkt. Věřím, že aplikace SQUADRA najde své uplatnění v praxi a stane se užitečným pomocníkem pro efektivnější fungování sportovních kolektivů.

Aplikace je zálohovaná na GitHub adrese <https://github.com/kubasoltys/Maturitni-projekt>.

LITERATURA

- [1] DOKULIL, Jakub. *Šablona pro psaní SOČ v programu L^AT_EX* [online]. Brno, 2020 [cit. 2026-01-03]. Dostupné z: https://github.com/Kubiczek36/SOC_sablon
- [2] *Střední škola průmyslová a umělecká Opava* [online]. [cit. 2026-01-03]. Dostupné z: <https://www.sspu-opava.cz>
- [3] DJANGO SOFTWARE FOUNDATION. *User authentication in Django* [online]. [cit. 2026-01-03]. Dostupné z: <https://docs.djangoproject.com/en/5.2/topics/auth/>
- [4] POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL Tutorial* [online]. [cit. 2026-01-03]. Dostupné z: <https://www.postgresql.org/docs/current/tutorial.html>
- [5] TAILWIND LABS INC. *Tailwind CSS Documentation* [online]. [cit. 2026-01-03]. Dostupné z: <https://tailwindcss.com/>
- [6] DOCKER INC. *How to Dockerize a Django App* [online]. [cit. 2026-01-03]. Dostupné z: <https://www.docker.com/blog/how-to-dockerize-django-app/>
- [7] CHART.JS. *Chart.js Documentation and Samples* [online]. [cit. 2026-01-03]. Dostupné z: <https://www.chartjs.org/docs/latest/>
- [8] FONTAWESOME. *Font Awesome Docs* [online]. [cit. 2026-01-03]. Dostupné z: <https://fontawesome.com/docs>
- [9] W3SCHOOLS. *Online Web Tutorials and CSS Modals* [online]. [cit. 2026-01-03]. Dostupné z: <https://www.w3schools.com/>
- [10] HERMAN, Michael. *Django Custom User Model*. TestDriven.io [online]. [cit. 2026-01-03]. Dostupné z: <https://testdriven.io/blog/django-custom-user-model/>
- [11] VERY ACADEMY. *Django PostgreSQL Migration from SQLite* [online video]. YouTube, 2020 [cit. 2026-01-03]. Dostupné z: <https://www.youtube.com/watch?v=ZgRkGfoy2nE>

- [12] CODE WITH TOMI. *Docker With Django Tutorial* [online video]. YouTube, 2021 [cit. 2026-01-03]. Dostupné z: <https://www.youtube.com/watch?v=BoM-7VMdo7s>
- [13] MERMAID.JS. *Mermaid: Diagramming and charting tool* [online]. [cit. 2026-01-03]. Dostupné z: <https://mermaid.js.org/>

Seznam obrázků

3.1	ER diagram databázové struktury	8
3.2	Přihlášení	11
4.1	Vytváření zápasu	14
4.2	Zapisování výsledku	15
4.3	Výběr týmu	16